We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Interaction Paradigms for Bare-Hand Interaction with Large Displays at a Distance

Kelvin Cheng and Masahiro Takatsuka
HxI Initiative, NICTA & ViSLAB, The University of Sydney
Australia

1. Introduction

Large displays are now more widely used than ever before, yet the computer mouse remains the most common pointing tool for large displays users. The main drawback of the mouse is that it is only a tool for mapping hand movements to on-screen cursor movements, for the manipulation of on-screen objects. Such an indirect mapping is due to differences between input space (usually a horizontal table used by the mouse) and output space (display) (Hinckley, 2003). This indirectness enlarges the Gulf of Execution and the Gulf of Evaluation (Norman, 1986), and thus reduces users' freedom and efficiency.

Much effort has been made on making this interaction more natural and more intuitive for users. One common approach is to use our own hand as the input method, making pointing as easy as pointing to real world objects. Computer vision can be used to detect and track user's pointing gesture and is an active area of research. Computer vision based systems have the advantages of being a non-invasive input technique and do not require a physically touchable surface, which is highly suitable for interaction at a distance or hygienically demanding environments such as public spaces. However, while previous works seemingly employ a similar interaction model, the actual model itself has not been well researched.

This chapter examine the underlying interaction models for bare-hand interaction with large display systems from a distance. Theories and techniques that make interaction with computer display as easy as pointing to real world objects are also explored. We will also investigate the feasibility of using low-cost monocular computer vision to implement such a system.

2. Background & Previous Work

We start off by reviewing important work historically and investigate significant related research in this area and take a close look at the current trend in immersive interaction and the state of the art techniques that researchers around the world have proposed to improve interactions with large surfaces. We focus on non-intrusive techniques where users do not need to wear or hold any special devices, nor will there be wires attached. Users only need to approach the surface and use their bare hand. Most of these use some kind of sensors or

computer vision to detect the hand. We will also present literature that has adopted hand pointing as one of the main approaches to selection.

2.1 Sensitive Surfaces

DiamondTouch (Dietz & Leigh, 2001) is a touch sensitive table from Mitsubishi Electric Research Laboratories (MERL) that can detect and identify multiple and simultaneous users' touches. Antennas are place under the surface of the table each with a different electrical signal. When the user touches the table, a small electrical circuit is completed, by going from a transmitter to the table surface to the user's finger touching the surface, through the user's body and onto a receiver on the users' chair. Users must be seated to operate this device.

SmartSkin (Rekimoto, 2002) is a similar technique proposed by Rekimoto. Instead than using electricity, it relies on capacitive sensing by laying a mash of transmitter and receiver electrodes on the surface. When a conductive object (such as the human hand) comes near the surface, the signal at the crossing point is decreased. This also enables the detection of the multiple hand position. They are also able to detect the hand even when it is not actually touching the surface.

More recently, multi-touch techniques have been widely researched. One example is the use of frustrated total internal reflection (FTIR) to detect multiple fingers on a tabletop display (Han, 2005). This technique makes use of the fact that when light is travelling inside a medium (e.g. acrylic pane), while undergoing total internal reflection, an external material (e.g. human finger) is encountered causing light to escape from the medium. An external camera can capture exactly where the light escapes, thereby detecting position of the fingertip. Multiple fingertips can thus be detected at the same time

2.2 Tracking hands above surfaces

Computer vision can also be used to track different parts of the human user. The most notable is the implementation of DigitalDesk (Wellner, 1993) by Wellner in 1993 who used a projector and a camera looking down from above a normal desk. It allows users to use their finger, detected by using finger motion tracking, and a digitizing pen. It supports computer based interaction with paper documents allowing such tasks as copying numbers from paper to the digital calculator, copying part of a sketch into digital form and moving digitals objects around the table.

In a similar research, rather than aiming a camera at a desk, the camera is directed to a vertical whiteboard. Magic Board (Crowley et al., 2000) used a steerable camera so that screens can be larger than the field of view of the camera. Rather than using motion detection, cross-correlation is used instead, which extracts small regions from an image (e.g. image of a pointing finger) as template for searching in later images.

The SMART board (SMART, 2003) fixes 4 tiny cameras on four corners of its display which can detect any objects that come into contact with the surface or when it hovers above it.

The Everywhere Displays Project uses a "multi-surface interactive display projector" so that it can make any surface in the room interactive (Pinhanez, 2001). It is done by attaching a rotating mirror so that any surface in the room can be projected onto and captured by the camera. Finger detection is performed on the same surface that is being projected generating a "click" event as if it is a computer mouse.

The use of infrared has been used to remove the need for color hand segmentation and background subtraction due to fact that they sometimes fail when scene has a complicated background and dynamic lighting (Oka et al., 2002). Diffused illumination (DI) is another common technique used in HoloWall (Matsushita & Rekimoto, 1997) and Barehands (Ringel & Henry Berg, 2001), where infrared LEDs are emitted from behind the wall as well as a back projected display. An infrared filtered camera is positioned behind as well to detect the presence of hand or finger when it comes near the wall (within 30 cm) which reflects additional infrared light. The Microsoft Surface (Microsoft) is a tabletop display that also uses this technique, in addition, the camera can also recognize tagged physical objects placed on the surface.
EnhancedDesk (Oka et al., 2002) used hand and fingertip tracking on an augmented desk interface (horizontal). An infrared camera is used to detect areas close to the human body temperature. Selection is based on where the fingertip is.
The Perceptive Workbench (Leibe et al., 2000) uses infrared illuminated from the ceiling and a camera under a table. When the user extends their arm over the desk, it casts a shadow which can be picked up by the camera. A second camera fitted on the right side of the table captures a side view of the hand. Combining together, the location and the orientation of the user's deictic (pointing) gesture can be computed. The approach assumes that the user's arm is not overly bent. It fails when shadow from the user's body is casted, as well as when two arm are extended at the same time.
All of these techniques require the user to be at the location they want to point at. This also requires large movements of the arm, as well as pacing across surfaces. They do not work well when surfaces are hard to reach.

2.3 Free Hand Pointing

Perhaps the most direct form of selection at a distance is being able to point at something with your hand without any restrictions. The current trend is to make pointing as easy as pointing to real world objects. Taylor and McCloskey identified that "to indicate accurately the position of an object that is beyond arm's reach we commonly point towards it with an extended arm and index finger" (Taylor & McCloskey, 1988). The pointing gesture belongs to a class of gesture called "deictics". Studies have shown that the pointing gesture is often used to indicate a direction as well as to identify nearby objects (McNeill, 1992). This is an interesting concept because the pointing gesture is natural, intuitive, and easy to use and understand. Indeed, children can use body postures and gestures such as reaching and pointing by about nine months (Lock, 1980).
Here, we will review some of the literature, approaches to pointing and the varied implementations where hand pointing was used for interactive systems. In general these systems have the advantages of having no physically touchable surfaces thereby highly suitable for hygienic demanding environment such as factories or public spaces. Depending on the system setup, this usually allows users to interact with the display wherever they are standing.

2.3.1 Stereoscopic Systems

The pointing finger has been used in many existing systems. A pair of uncalibrated stereo cameras can be used with active contours to track and detect the position and direction of the index finger in a 40cm workspace (Cipolla & Hollinghurst, 1998).

The Hand Pointing System (Takenaka, 1998) developed by Takenaka Corporation uses two cameras attached to the ceiling to recognize the three-dimension position of user's finger. A mouse click is mimicked by using a forward and backward movement of the finger.

The Free-Hand Pointing (Hung et al., 1998) is a similar system that also uses stereo camera to track the user's finger. They first detect and segment the finger with a global search, then the fingertip and finger orientation is determined in a local search, and finally the line from finger to display is extracted.

The PointAt System (Colombo et al., 2003) allows users to walk around freely in a room within a museum while pointing to specific parts of a painting with their hand. Two cameras are setup to detect the presence of a person by using modified background subtraction algorithm as well as skin color detection. The tip of the pointing hand and the head centroid is then extracted. By using visual geometry and stereo triangulation, a pointing line is then deduced. This method can be applied to more than 2 cameras as well, and do not require manual calibration. Dwell clicking is used in this implementation.

Similarly, Nickel and Stiefelhagen (Nickel & Stiefelhagen, 2003) also used a set of stereo cameras to track the user's hand and head to estimate the pointing direction in 3D. Pointing gesture is recognised by using a three-phase model: *Begin* (hand moves from arbitrary position towards pointing position), *Hold* (hand remains motionless while pointing) and *End* (hand moves away from pointing position). They found that adding head orientation detection increases their gesture detection and precision rate. Comparing three approaches to estimate pointing direction, they found that the hand-head line method was the most reliable in estimating the pointing direction (90%). The others were forearm direction and head orientation.

In most case studies above, the design choice for the interaction technique, such as the different hand signal or hand gestures used, was not based on observation from prior user study. These are frequently based solely on the authors' intuition. These are inconsistent across different experiments.

There are a number of problems associated with using two cameras (Takatsuka et al., 2003). One is the reduced acquisition speed as there is a need to process two images entirely to locate the same point. With stereoscopic view, it is not difficult to find out the exact location of a certain object in the scene and is the reason many gesture based computer vision research has been based on stereo cameras. However, there are few studies in literatures that use monocular vision to allow the use of remote hand pointing gesture as well as being non-intrusive.

2.3.2 Monocular Systems

A single camera makes it much easier and simpler to allow real-time computation. Nowadays, most computer users have one web-camera at home, but possessing two or more is less likely.

In the EyeToy game for PlayStation2 (SONY), a motion recognition camera can be placed on top of a large display. A mirror image from the camera is presented on the display, as well

as additional game play information. A selection is made when users place their hands at specific location so that it's on-screen image coincided spatially with on-screen targets.
In the Ishindenshin system (Lertrusdachakul et al., 2005) a small video camera is placed near the center of a large vertical display directly in front of the user. The user is able to keep eye-contact and use the pointing gesture to interact with another user in a video conference. The fingertip location is detected by the camera and its location in x and y coordinates are determined.
A similar method is also used in another study (Eisenstein & Mackay, 2006) to compare the accuracy using two computer-vision based selection techniques (motion sensing and object tracking). They found that both techniques were 100% accurate and that the object tracking technique was significantly fewer errors and took less time.
The Virtual Keypad implementation (Tosas & Li, 2004) detects the user's fingertips position in both the x and y directions for interacting with targets on-screen. It is used much closer to the camera. Note that in all of these systems, the fingertip or hand can only be tracked in 2D, depth is not registered.
In the computer vision community, various researchers have investigated the use of monocular vision to estimate depth (Torralba & Oliva, 2002, Saxena et al., 2007). It has even been suggested that monocular vision is superior in detecting depth than stereo vision due to the fact that "small errors in measuring the direction each camera is pointing have large effect on the calculation (Biever, 2005)". On the other hand, very few examples in the HCI literature have been found using monocular vision to allow the use of remote hand pointing gesture whilst being non-intrusive.
Compared with monocular vision techniques, it is easier to find the exact location of a certain object in the scene using stereoscopic view. The need for a pair of stereo cameras would be eliminated if depth recovery is achievable with a single camera. This in turn transfers the problem from one that involves the hardware to one of software. However, few researchers have investigated interaction methods that rely on monocular computer vision, and where depth information recovery is required.
3DV Systems developed the "ZCam" to detect depth information based on the Time-Of-Flight principle (3DV, 2008). Infrared light are emitted into the environment and a camera captures the depth by sensing the intensity of the reflected infrared light reaching back to the camera. The major drawback of this setup is that it requires potential users to purchase this special camera.
The above mentioned systems used monocular vision to detect the users hand only. However, in order to provide a truly non-intrusive and natural pointing system, the system should also take into account the user's standing position.
The "Peek Thru" implementation does exactly this (Lee et al., 2001). A single camera is used to detect the user's pointing finger and the eye position. However, the detection of the fingertip was deemed too difficult to identify due to the observed occlusion by the user's torso. Users were asked to wear an easy to detect thimble on their fingertip. We feel that this violates the notion of using nothing but our own hand for interaction. Even if the fingertip position was detected using computer vision, this setup only differential between different angels from the camera's view, rather than the exact x and y coordinates.
As can be seen, we can observe numerous attempts to further the current keyboard and mouse interface for the large display.

At this point in time, we are starting the transition from the mouse based UI to that of surface based hand tracking era. They currently exist commercially in the form of public directory touch screen and multi-touch interfaces such as iPhone and Microsoft Surface. We believe that camera based hand pointing is the next frontier in the future of HCI, as users do not even require to touch the display, they are able to interact from a distance. It is time for the computer system to finally adapt to humans, rather than humans adapting to technologies. This research is invaluable for us in developing more natural interaction methods that are easy to setup and use.

3. Pointing Strategies

We begin our investigation by focusing on the use of natural pointing for interacting with the computer.

Although many interactive systems have focused on improving the detection of the users' pointing direction, few have considered the kinds of pointing strategy that is natural to the users and analyzed the accuracy of these strategies provided by the users themselves.

3.1 Background and Related Work

Let's look at the pointing strategies used in recent literature.

The MobiVR system (Rakkolainen, 2003) captures and detects a pointing finger behind a handheld micro display – a non-see-through binocular near-eye microdisplay (taken from a Head Mounted Display) attached to a reconfigured SmartNav infrared camera aiming forward. By attaching an infrared reflector ring on the finger, the user can perform a pointing gesture in front of the device to control a mouse cursor. This makes use of the eye-fingertip strategy, where the pointing direction is extracted from a line starting at the eye and continues to the fingertip. An interesting point here is that the fingertip is behind the near-eye display.

Various researchers have investigated the use of the head-hand line (similar to the eye-fingertip strategy but detecting the whole head and hand rather than specifically the eye or fingertip) for interacting with their systems (Colombo et al., 2003, Nickel & Stiefelhagen, 2003). All of these systems used a set of stereo cameras to track the user's head-hand line to estimate the pointing direction in 3D at a distance of around 2 meters. However, Nickel and Stiefelhagen (Nickel & Stiefelhagen, 2003) also found that adding head orientation detection increases their gesture detection and precision rate. Comparing three approaches to estimate pointing direction, they found that the head-hand line method was the most reliable in estimating the pointing direction (90%). The others were forearm direction (73%) and head orientation (75%). However, their result is based on whether 8 targets in the environment were correctly identified, rather than accurately measuring the accuracy from a specific target. The forearm orientation and head-hand line were extracted through stereo image processing while the head orientation was measured by means of an attached sensor.

There is evidence to suggest that the natural pointing gesture may be estimated to be somewhere between the head-hand line and the arm pointing strategy. Mase (Mase, 1998) used a pair of stereo cameras to determine the fingertip position. In order to extract a virtual line between the finger and a target, the location of the "Virtual Project Origin (VPO)" must be calculated. The VPO varies according to the pointing style of each individual. They made

used of a pre-session calibration and experimental results suggested "the VPO is mostly distributed along the line of the eye on the pointing hand's side and the shoulder".
In an experiment to investigate the effect of vision and proprioception in pointing ("the ability to sense the position, location, orientation, and movement of the body and its parts" (The Gale Group, 2005)), it has been reported that users tend to point (with an extended arm) to the target (about a meter away) by placing their fingertip just lateral to the eye-target line (Taylor & McCloskey, 1988). However, a line extended from the direction of the pointing arm would miss the target. They explained that this may be used to avoid the fingertip occluding the target or, alternatively, influenced by proprioception (which usually results in using the whole arm as a pointer). Their result suggests that the eye-target line is a good predictor of the pointing direction.
It is interesting to note that Olympic pistol marksmen and archers aim their targets by extending their arm and standing side-on to the target, so that the line running from the eye to the target coincide with the line running from the shoulder to the target (Taylor & McCloskey, 1988)
The accuracy estimated by vision systems is only as good as the pointing accuracy provided by the user, and in practice this can be even worse. To make any system more reliable and accurate, one should begin by understanding the pointing strategies adopted by users when pointing, and methods in which they can adopt to point to the best of their ability. Only then should we focus on detecting the hand accurately. It is observed that different systems require different strategies for targeting. There is currently a gap in the literature on systematically describing how and why natural interaction methods work and classifying them based on their accuracy, naturality and how well they capture the exact intention of the user. The above mentioned literature has mainly focused on extracting or detecting different pointing gestures through image processing or by different sensors. However, no known strategies for pointing were recommended which allow users to point naturally and accurately (to best represent their aim). In this context, we investigate how people point naturally and better understand the mechanism and the strategy behind targeting.

3.2 Style of Pointing Gesture

A preliminary experiment was conducted to investigate how people naturally point at objects on a vertical wall in a normal environment and to investigate the style of gestures people adopt when pointing. We hypothesized that there are three main pointing strategies: (1) using a straight arm, (2) using only the forearm and extended fingertip, and (3) placing their fingertip in between the eye and target (line-up).
Nineteen volunteers were recruited for the study but were not told the main objective at the beginning, only that they are required to point at a target using their arm. Subjects were asked to stand at three distances (1.2m, 3m and 5m) away from a wall and point to a target object, 5mm by 5mm, was marked on the wall, 155 cm from the ground (around eye level for most participants). The study was a within-subject study so that each of the subjects had to complete all tasks for each distance. The order of distances was counter-balanced with approximately one third starting at each distance.
In the first task, subjects were asked to point at the target using their preferred hand. They were free to use any strategies they wanted and were not told specifically how they should point. This allows us to observe the kind of strategy used naturally by each subject to point at objects from a distance. In the second task, subjects were specifically asked to point using

their forearm and their fingertip only, limiting the movement of their upper arm. This allows us to observe different variations of the forearm pointing method.
In the final task, subjects were specifically asked to use their whole arm to point while keeping it straight.

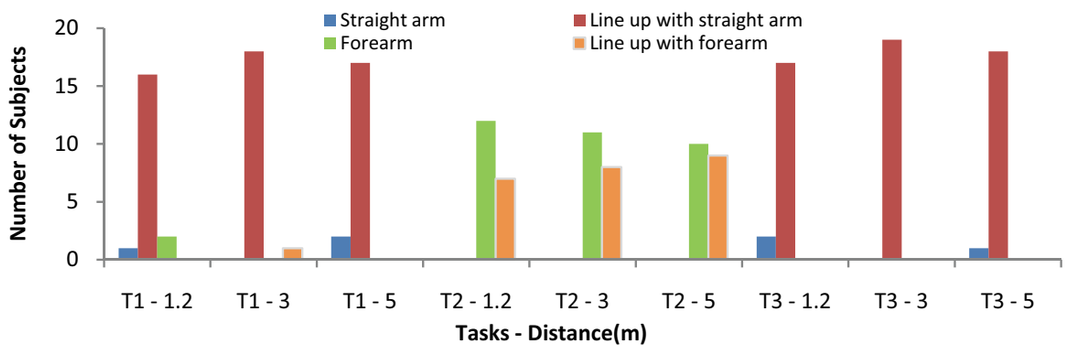


Fig. 1. The number of subjects using different pointing styles for each task/distance combination.

The main observation for task 1 was that almost all subjects used a full arm stretch to point regardless of distance. This may be due to the fact that during full arm pointing, they can see their arm in front of them, which provides a better visual approximation than the limited view of the arm provided with just the forearm.
Except as required in task 2, almost no subject used the forearm pointing method. While this method was praised for the minimal effort required to point, subjects felt that this method was unnatural and awkward.
In task 3, even though users were specifically asked to use full arm pointing, we observed two main strategies used to point at the target. Three subjects used their arm as a pointing instrument where the pointing direction is estimated from the shoulder to their fingertip (Figure 2b), while most users tried to align their fingertip in between the eye and target (Figure 2c) as hypothesized. This observation can also been seen in task 1.

(*a*) (*b*) (*c*)

Fig. 2. (a) Forearm pointing: only using the forearm and fingertip to point, while keeping their upper-arm as close to the body as possible (b) Full arm pointing 1: using the arm as a pointing instrument while keeping the arm as straight as possible (c) Full arm pointing 2: the fingertip is placed between the eye and the target.

Overall, we observed a clear preference for the line-up method when pointing with a straight arm, while the two different forearm pointing methods are roughly equally used. Agreeing with our hypothesis, we did observe three different methods of pointing. The results from this study suggested that the line up pointing method is shown to be the most natural way of pointing to targets. Qualitative measures also suggest that users prefer to use a full arm stretch to point at targets.
Given the small scope, this experiment should only be treated as a preliminary work on this subject. However, this may serve as a basis for further analysis and experimentation with different size of targets.
Having studied the styles of pointing that are natural to the users, we observed informally that the forearm pointing method may be less accurate than both form of full arm pointing. However, further investigation would be required to justify this.

3.3 Pointing Accuracy

In most vision-based interactive system the accuracy of the estimated pointing direction is an essential element to their success. The focus is usually in finding new ways of improving the detection, estimation and tracking the pointing hand or finger, and deduce the pointing direction (Cipolla & Hollinghurst, 1998, Nickel & Stiefelhagen, 2003). However, it is assumed implicitly that the user is always pointing to their desired location all the time, and the systems do not take into account inaccuracies made by the user. A study was, therefore, conducted to investigate the accuracy provided by three common pointing strategies used in previous interactive systems (without the presence of feedback to the user): Forearm, Straight-arm, Line-up methods.
Despite advances in computer vision, there is still no consistent method to segment and extract the pointing direction of the arm. To minimize error introduced by a computer vision system, we detect the pointing direction of the arm by asking subjects to hold a laser pointer in their hand in a way that best represent the extension of their arm direction and which was consistent in all three methods (Figure 3).

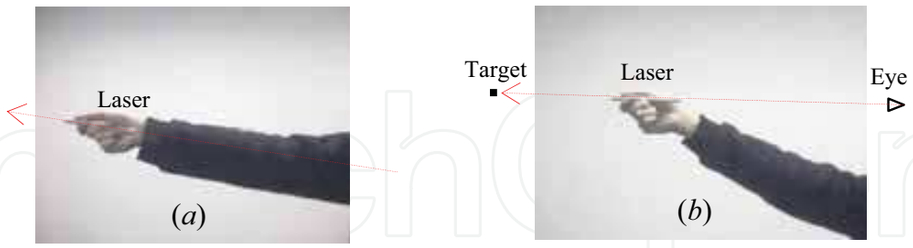


Fig. 3. (*a*) The laser pointer was used to represent the arm's direction (*b*) The laser pointer used with the line-up method, where the target, two ends of the laser pointer and eye are collinear.

Laser dot gave us a way to quantitatively compare the different targeting strategies. Although physically pointing with hand or arm compared to holding the laser in the palm are slightly different, we believe that this difference would be consistent enough so that it would still be comparable relatively between the strategies. In addition, users were not provided feedback from the laser pointer to adjust their accuracy. The effect of distance on

the accuracy of pointing – whether pointing deteriorates as user moves away from the target – was also investigated in this experiment.

Fifteen volunteers participated in this study. A webcam was used to detect a 5x5mm target on a wall and the laser dot produced by the laser pointer. The study was a within-subject study, where each subject performed pointing tasks with all three pointing styles from the three distances: 1, 2 and 3 metres from the target. Three blocks of trails were completed for each of the 9 combinations and the mean position for each combination was determined. The order of pointing styles and distances were counter-balanced. Without turning on the laser, subjects were asked to aim as accurately as possible, and hold the laser pointer in their hand in a way that best represents the direction of their pointing arm (for straight arm and forearm pointing, Figure 3a). For the line-up method, users were asked to place the laser pointer so that both ends of the laser pointer are collinear with the target and the eye (Figure 3b). To prevent subjects receiving feedback from the laser dot, the laser was only turned on after they have taken aim.

Accuracy was measured in terms of the distance between the target and the laser dot produced on the wall. In summary, the experimental design was: 15 subjects x 3 pointing techniques x 3 distances x 3 blocks = 405 pointing trials.

Figure 4 illustrates the mean distance from target for each pointing method at three distances and their interactions for all trials.

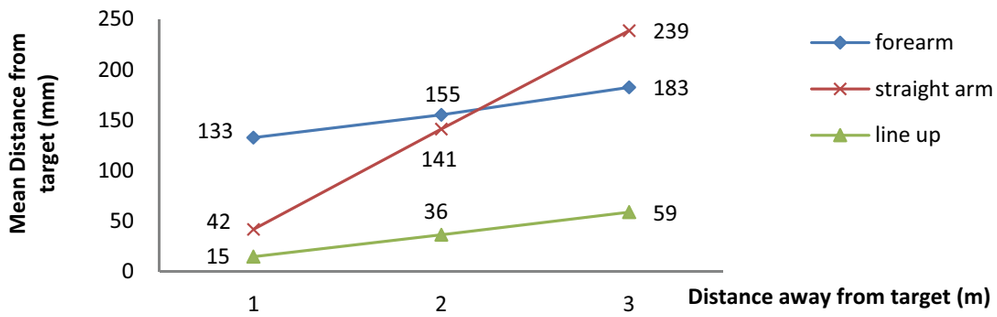


Fig. 4. Mean distance between target and the laser dot position.

A two-way analysis of variance (ANOVA) with repeated measures reveals a significant main effect for pointing technique on accuracy ($F[2,28]=37.97$, $p<0.001$), and for distance on accuracy ($F[2,28]=47.20$, $p<0.001$). We also observed a significant interaction between technique and distance ($F[4,56]=9.879$, $p<0.001$).

Multiple pairwise means comparisons were tested within each pointing technique (table 1) with Bonferroni correction. Trend analyses were also performed on each of the technique. Significance in the linear component signifies a linear increase in accuracy with increasing distance within that particular technique.

Technique	1m vs 2m	2m vs 3m	1m vs 3m	Linear component
Forearm	1.000	0.378	0.055	0.018*
Straight arm	<0.001*	0.002*	<0.001*	<0.001*
Line up	0.003*	0.116	0.014*	0.005*

*Denotes significance at the 0.05 level

Table 1. The significance of multiple pairwise means comparisons within each pointing technique.

Multiple pairwise means comparisons were also performed within each distance to investigate possible differences between each technique (table 2).

Distance	Forearm vs straight arm	Straight arm vs line up	Forearm vs line up
1m	0.001*	<0.001*	<0.001*
2m	1.000	<0.001*	<0.001*
3m	0.182	<0.001*	<0.001*

*Denotes significance at the 0.05 level

Table 2. The significance of multiple pairwise means comparisons within each distance.

The results suggest that the line-up method is the most accurate at all distances. We can also observe a linear increase throughout at a rate of 14.7mm per meter. The forearm pointing technique was consistently less accurate than the line-up method. It is interesting to note the insignificance between the three distances within the forearm pointing method. This may suggest that the pointing method has a high tolerance with increasing distance from target. On the other hand, straight arm pointing method is highly affected by the increase in distance from target. This is illustrated by the significant difference across all distances as well as the high linear increase (65.7mm per meter). Compared to forearm pointing, the accuracy at 2m and 3m is not significant. The only difference between forearm and straight arm pointing is at 1m. This may be due to the higher level of feedback given from the longer arm extension, and that the straight arm pointing resembles the line-up method at close proximity to the target.

From this experiment, we have identified inaccuracies in users pointing performance, which varies depending on the strategy used. We observed that the line-up method is the most accurate pointing method, and that the straight arm method is more accurate than the forearm method only at a distance of one metre. Understanding the natural pointing accuracy can assist future vision-based hand pointing interaction researchers and practitioners to decide the input strategy that best suit their users in completing the required tasks.

From the literature review, we observed that different interactive systems use different strategies for pointing. However, the pointing strategies used have not been systematically studied. Here, we attempt to characterize the mechanism of pointing in terms of their geometric models used in these systems, and in the process, we use the results from our experiments to gain a better understanding of how and why the line-up method is a more accurate pointing method.

3.4 Models for Targeting

We hypothesized that there is a difference between the two strategies of targeting using full arm stretch. To investigate the reason for this difference, we begin by formalizing the

concept of targeting from a geometrical perspective based on our observations and from previous work. We then introduce three models for targeting - the Point, Touch and the dTouch model. It should be noted that the word "pointing" and "targeting" can be used interchangeably to mean the act of aiming (at some target).

3.4.1 Geometrical Configuration

We now define the geometrical configuration in which our models for targeting from a distance will be based. These are the building blocks for the models of targeting.
From a point at the eye, P_e, a line of gaze, l_g, is directed towards a point on a target object, P_o. While a point provided by a pointing mechanism, P_p, guides a line of ray, l_r, to the same target P_o. Figure 5a illustrates this geometrical arrangement.

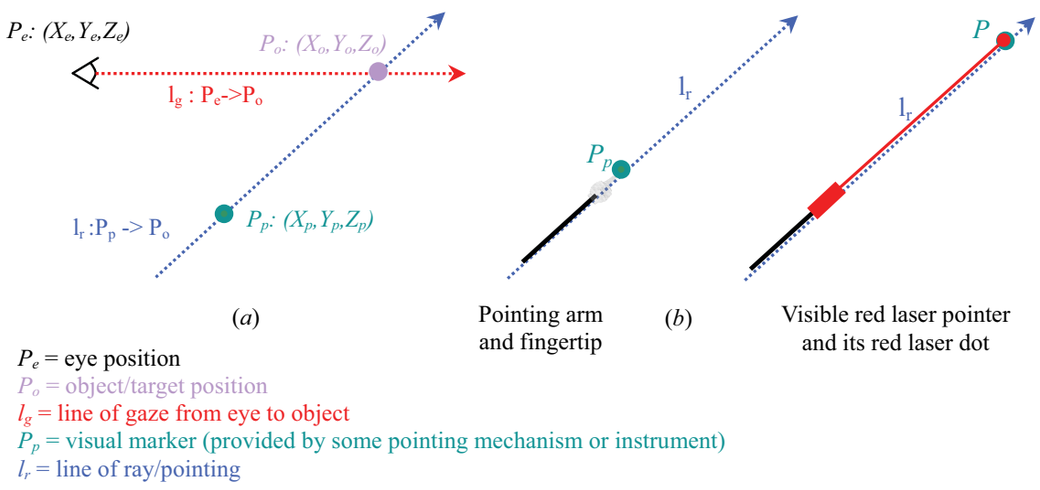


Fig. 5. (a) The general configuration for targeting (b) Examples of visual markers.

The task of targeting is therefore to intersect l_g with l_r at object P_o. A pointing mechanism or visual marker (P_p) may include a variety of pointers that the user holds or use to point. The fingertip (when user is using their arm) and the laser dot produced by a laser pointer are examples of such (Fig. 5b). On the other hand, the arm direction is an example of line of pointing (l_r).
We now distinguish three models that can explain most of the common strategies used in the real environment and in many previous interactive systems.

3.4.2 The Point Model

The Point model describes the occasion when users' point at a target from a distance using their arm or a presentation pointer as the pointing instrument. Targeting is characterized by having the eye gaze, l_g, intersects with the pointing direction provided by the arm, l_r, at the target object, P_o, such that the pointing marker, P_p, doesn't meet at the target object P_o (i.e. $P_o \neq P_p$). Figure 6a illustrates this geometrical arrangement. The task for the user is to use their pointing instrument to approximate a pointing direction that meets the target object.

However, it is only an approximation, rather than precise targeting, since the visual marker is not on the surface of the target to assist the targeting process.
This can be used to model the cases when the arm is fully stretched, when only the forearm is used to point to the target (Nickel & Stiefelhagen, 2003) or when only the fingertips are used, in the case of (Cipolla & Hollinghurst, 1998). This technique is known as *ray casting* for interacting with virtual environments (Bowman & Hodges, 1997). It can also be used to model the straight-arm method and the forearm method that were observed and used in our experiments in this section 3.2 and 3.3. In these cases, the length of the whole arm, forearm or fingertip is used as a pointing instrument P_p to infer a line of pointing l_r towards a target P_o (Figure 6b). This model can also be used to explain the use of an infrared laser pointer (Cheng & Pulo, 2003). In that work, the infrared laser pointer is used to point at an on-screen target (similar to a regular red laser pointer). However, the laser dot is not visible to the user. Cameras are used to capture the infrared laser dot on the display to determine the on-screen target selected. The only visual marker to guide the user to point to the target is the laser pointer itself (and not the laser beam). In this case, the infrared laser pointer is represented by P_p and its inferred pointing direction l_r.

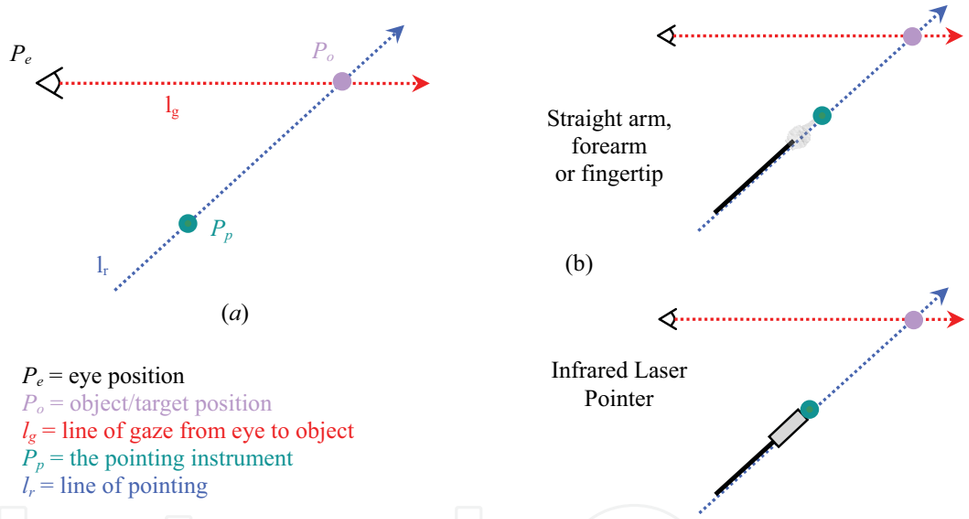


Fig. 6. (a) The Point Model for targeting (b) Examples of techniques that use the Point model.

Pointing using this Point model may be inaccurate, due mainly to the distance between P_p and P_o. Consistent with the results of our accuracy experiment in 3.3, the straight arm pointing method was observed to be more accurate when the subject, and hence the arm and hand (P_p), is close to the target, is at a distance of 1 meter from the target (mean error of 42mm). However, as the user moves further away from the target, the inaccuracy increased dramatically (mean error of 141mm at a distance of 2m). Therefore, it can be seen that accuracy is not guaranteed when pointing techniques which make use of the Point model are used.

3.4.3 The Touch Model

The Touch model describes the occasion when the fingertip is used to physically touch a target object or when a pointing instrument is used and a visual marker is seen on the surface of the object. Targeting is characterized by having the eye gaze, l_g, intersects with the pointing direction provided by the arm, l_r, at the target object, P_o, such that the pointing marker, P_p , meets at the target object P_o (i.e. $P_o = P_p$).

Figure 7 illustrates this geometrical arrangement. With this model, the task for the user is to use a visual marker (e.g. their fingertip or a pointing instrument) to physically makes contact with a target object (more specifically on the surface of the object). This is a form of precise targeting as the visual marker assists the targeting process.

This can be used to model any kinds of touch based interaction including DiamondTouch (Dietz & Leigh, 2001) and SmartSkin (Rekimoto, 2002). The fingertip acts as a pointing instrument P_p and is used to make physical contact with the target P_o (Figure 7b). Other input methods may be used in place of the fingertip, such as a stylus (a pen like object with a tip), to act as the pointing instrument.

This model can also be used to explain the use of a red laser pointer, for example in (Olsen & Nielsen, 2001). The red laser dot produced by the pointer is represented by P_p and its pointing direction l_r. The red laser can be thought of as an extended arm, and the laser dot, the index finger.

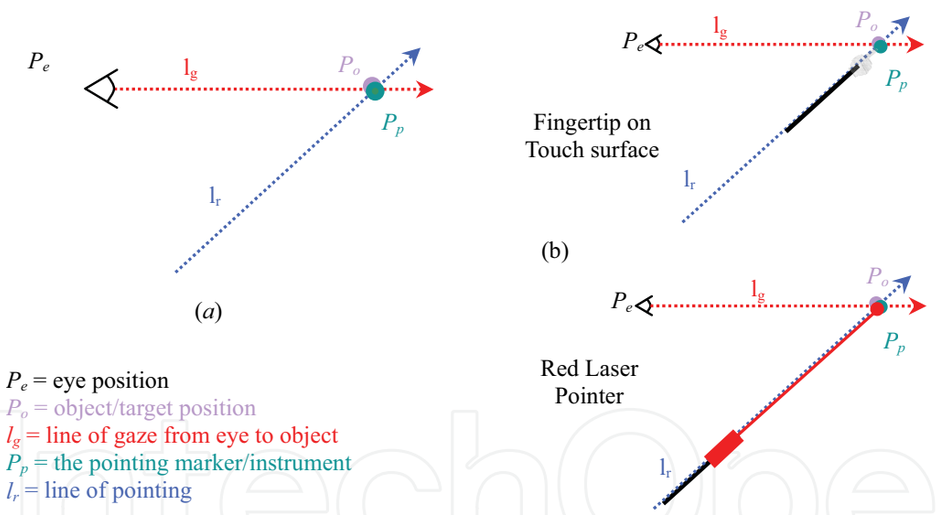


Fig. 7. (a) The Touch Model for targeting (b) Examples of techniques that use the Point model.

Targeting using the Touch model is accurate. This is because the distance between P_p and P_o is zero and they overlap at the same position, which makes targeting a precise task. Unlike the Point model, estimating the direction of pointing, l_r, is not required. Even when users misalign their pointing instrument and the target, the misalignment can be easily observed by the user, allowing readjustment of the position of the pointing instrument. Therefore, it can be seen that accuracy is guaranteed when pointing techniques which make use of the Touch model are used.

3.4.4 The dTouch (distant-Touch) Model

The dTouch model describes the occasion when the fingertip or a visual marker is used to overlap the target object in the user's view, from a distance. It may also be describes as using the fingertip to touch the target object from a distance (from the user's point of view).

Targeting using the dTouch model is characterized by having the eye gaze, l_g, intersects with the pointing direction provided by the arm, l_r, at the pointing marker, P_p, such that the eye, P_e, the pointing maker, P_p, and the target object, P_o, are collinear. P_p may or may not coincide with P_o. Figure 8 illustrates this geometrical arrangement.

With this model, the task for the user is to align a visual marker, P_p, anywhere along the gaze from eye to target, l_g, so that it aligns with the object (as seen from the user's eye). It is not a requirement that the user is located close to the target object. The target may even be unreachable to the user (Figure 8a). In such case, the dTouch model can be considered a remote touch, a touch that occurs from afar (touch interaction without physically touching).

In the case when P_p coincides with P_o (i.e. the fingertip touches the target, Figure 8b), it fits both the Touch model and the dTouch model. The dTouch model can be considered a generalization of the Touch model since the dTouch model can be used to encompass the case when P_p coincide with P_o (defined by the Touch model), as well as other cases where P_p and P_o do not coincide. In other words, the Touch model is a specific case of the dTouch model. The main difference between the Touch model and the dTouch model is the position of the pointing marker, P_p. Even though both models restrict the marker to lie on the line of gaze, l_g, it must be coincident to the target object with the touch model, while it is unrestricted with the dTouch model. The dTouch model can therefore represent a wider selection of pointing techniques than the Touch model.

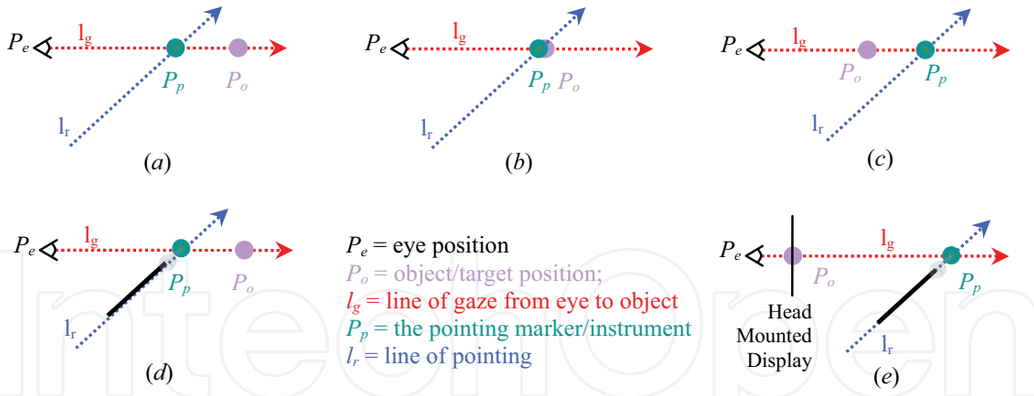


Fig. 8. (a-c) The dTouch model for targeting (d) An example using the eye-fingertip line (e) An example using a head mounted display and finger to interact with virtual object.

The generalized dTouch model can be used to model previous works that uses the eye-fingertip line (Lee et al., 2001) or the head-hand line (Colombo et al., 2003) and the line-up method that were observed and used in our experiments in this section, 3.2 and 3.3. The fingertip or hand act as a pointing instrument, P_p, and is aligned with the target P_o, on the line of gaze. When the user interacts with an object using the dTouch model, the user's intention is realized on the screen target.

This model can also be used to explain the interactions in previous works on head mounted virtual displays (Rakkolainen, 2003, Pierce et al., 1997). In these works, a head mounted display is worn in front of user's eye, and the fingertip is used to interact with the virtual objects. However, the virtual target, P_o, is located between the eye, P_e, and the fingertip, P_p, on the line of gaze, l_r (Figure 8c).

The accuracy of targeting using the dTouch model depends on the distance between P_p and P_o. Due to hand instability by users, the further away the two points are from each other, the larger the amount of hand jitter. However, unlike the Point model, estimating the direction of pointing, l_r, is not required. Users can readjust their position of the pointing instrument when misalignment of the two points is observed by the user.

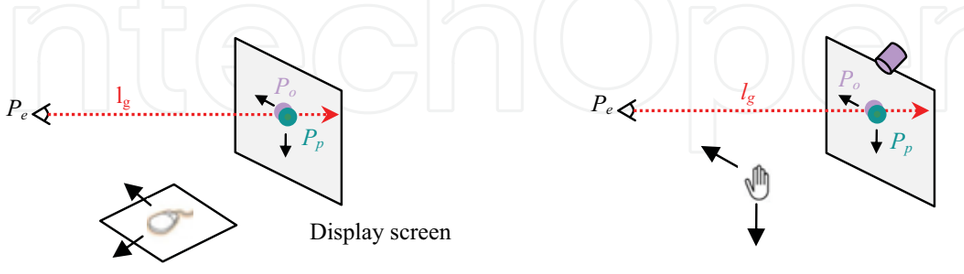
This is consistent with the results observed from our accuracy experiment in 3.3, at a distance of 1 meter the line-up method (mean error of 15 mm) is more accurate than the other two methods based on the Point model (means of 42 and 133mm). This is also consistent with the results from Nickel and Stiefelhagen (Nickel & Stiefelhagen, 2003), where the percentage of targets identified with the head-hand line using our dTouch model (90%) is higher than the forearm line using our Point model (73%). As can be seen, the accuracy of the dTouch model is better than the Point model.

When the distance between the two points is reduced to zero, we can expect a guaranteed accuracy, as with the Touch model.

3.4.5 Indirect Interaction

Even though our Touch model is only relevant when the interaction is direct, the model can actually be used to represent indirect interactions, with only minor modifications. The interaction is indirect when the input space is no longer the same as the output space. The computer mouse is a good example of this. In such cases, the line of ray is no longer a straight line. The input and the output space may certainly have some form of correlation; however, a direct relationship (in terms of physical space) is no longer necessary. The ray of pointing is therefore no longer relevant. Here are some examples:

1) The mouse cursor appearing on the screen can be represented as P_p. When the cursor moves onto an on-screen UI widget, and the mouse is clicked, P_o and P_p coincides. The task for the user here is to align a mouse cursor to the target (indirectly through a mouse) (Figure 9a).



(*a*) Computer mouse (*b*) EyeToy game with hand movement

Fig. 9. Examples of indirect techniques that use the Touch model.

2) Even though remote controllers are discrete input device, they can be thought of in terms of this model as well. The directional keys on the remote controller may be mapped to the physical grid-like space on the display screen. When a directional key is pressed, the on-screen selection indicator (P_p) is moved closer to its intended target (P_o).
3) Yet another example of the use of this model is the EyeToy camera (SONY). The user's hand image displayed on the screen is represented by P_p. The task for the user is to shift their on-screen hand image as close to the target (P_o) as possible (Figure 9b).

Therefore, the Touch model can also be used to model any interactive system that relies on visual feedback, either direct or indirect. However, in our work, we are mainly interested in direct interaction. Interactions where users are not required to hold on to any devices (i.e. no intermediary devices), they are able to perform from a distance. Interaction techniques that use this model do not require the user to touch the screen or be within reach of the output display. They are able to interact remotely.

However, we should still recognize the benefits exhibited when the Touch model is adapted to indirect devices. For example, the computer mouse can provide users with stability, as hand jitter and fatigue will no longer be concerns. It also provides users with a higher degree of accuracy.

In summary, from these models, we have deduced that the Point model does allow direct interaction from a distance but can be highly inaccurate. The Touch model provides high accuracy but does not allow bare-hand interaction from a distance. While the dTouch model provides good accuracy and allows direct hand pointing strategy that we observed to be most natural to the users (eye-fingertip).

Understanding the mechanism of targeting can assist future human-computer interaction researchers and practitioners to decide the input strategy that best suit their users in completing the required tasks. When designing an interactive system that is natural, unintrusive and direct, we recommend that the dTouch model be used as the underlying strategy.

4. dTouch Pointing System - Conceptual Design

To demonstrate the use of the dTouch method, we proposed an example interactive system using the eye-fingertip method that allows direct and non-invasive interaction with a large display at a distance with a single camera. We call this the "dTouch pointing system".

Our goal is to design a natural interactive system for large screen displays that uses only a single camera, which relies on monocular computer vision techniques.

A single camera setup has only gained popularity in recent years in the form of webcam for personal computing and for console gaming (SONY). The main advantage of using a single camera as a basis for designing a new interactive system is the availability of the webcam. Most PC owners will already have one, and they are commonly included in laptop computers. This reduces the need for users to purchase expensive specialized hardware (in the case for a new input device), or the need for a second webcam (in the case for a stereo camera setup) which may otherwise be unnecessary. By using computer vision, users are able to use their natural ability to interact with the computer thus allowing a more enjoyable experience. Such an approach may also allow a wider adoption of new interactive technologies in daily life.

Current monocular systems often use a single camera to detect the position of the user's hand. The x and y coordinates in 2D space are determined. This is used to determine an intended target position on the screen. Interaction, then relies on visual feedback, usually in the form of an on-screen cursor. Although demonstrated to be accurate due to the feedback, this provides only an indirect interaction. The major drawback of this type of interaction is the fixed interaction space in a 2D area. The user is not able to move around freely as they must stay within the same area to interact with the system. To provide a more natural interactive system, we attempt to extract 3D information from the environment and the user.

4.1 Design Overview

We envision that our pointing system would be used in situations where occasional, quick, and convenient interaction is required, such as in a presentation setting or information kiosk, as opposed to highly accurate and continuous mouse movement such as those required for typical personal computing usage.

To provide natural interaction, the system must allow users to point at the display directly using their bare hand. A web-camera is placed above the screen and angled downwards in order to determine the location of the user. To estimate the user's pointing direction, a vector is approximated from the user's eye through the pointing finger to a point of intersection on the display screen. The resulting position on the display would be the user's intended on-screen object. This makes use of the eye-fingertip method in the dTouch model (Figure 10).

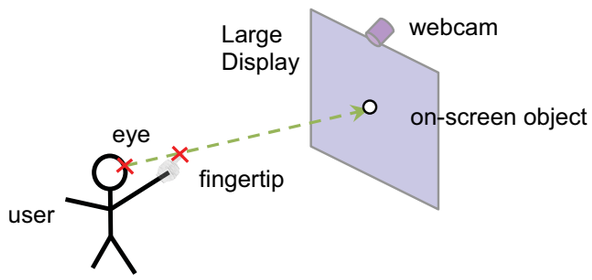


Fig. 10. Overview of the dTouch Interactive Pointing System.

To determine the pointing direction, image processing must be performed to track the eye and fingertip. In effect, we wish to construct a straight line in 3D space which can be used to give us a 2D coordinate on the display screen.

The specific concepts used in this setup are examined.

4.2 The View Frustum

A view frustum is used in computer graphics to define the field of view of a camera, or a region of space which may appear on a computer screen (Figure 11a). The view frustum is the area within a rectangular pyramid intersected by a near plane (for example a computer screen) and a far plane (Wikipedia, 2008). It defines how much of the virtual world the user will see (Sun, 2004). All objects within the view frustum will be captured and displayed on the screen, while objects outside the view frustum are not drawn to improve performance.

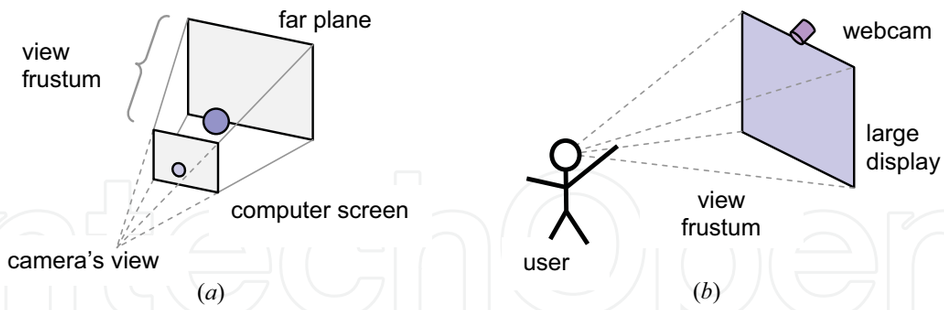


Fig. 11. (a) View frustum in computer graphics. (b) The view frustum.

An interaction volume is an area where the interaction occurs between the user and the system (the display, in our case). In computer vision based interactive systems this area must be within the camera's field of view. Users can use their hand within this area to interact with objects displayed on the screen.

In interactive systems that do not require explicit knowledge of the user's location, the interaction volume is static. To adequately interact with the system, the user must adjust themselves to the volume's location by pacing or reaching out. In addition, because the interaction volume is not visible, users must discover it by trial and error. On the other hand, when the user's location is known, the interaction volume adjusts to the user, and is always in front of the user.

To achieve the latter, as is the case of our method, a camera can be used to detect the face of the user. A view frustum can then be constructed between the origin (at the eye position) and the large display (Figure 11b). The view frustum can therefore be used as a model for approximating the interaction volume. The user can use their hand and fingers within this volume to interact with the display. The view frustum thus defines the interaction area available to the user.

4.3 Virtual Touchscreen

To investigate the integration of the benefits afforded by large displays and the interaction space afforded by the user, we proposed improving interaction with large displays by leveraging the benefits provided by touch screens.

Our idea is to imagine bringing the large display close enough to the user so that it is within arm's length from the user (thereby reachable), and users can imagine a touchscreen right in front of them (Cheng & Takatsuka, 2006). The distance between the user and the virtual objects remains unchanged. This "virtual" touchscreen and the original large display share the same viewing angle within the confines of the view frustum (Figure 12).

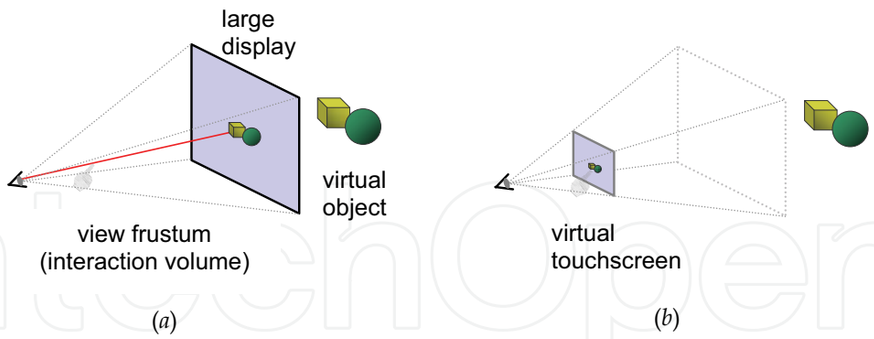


Fig. 12. (a) Pointing at the large display. (b) Pointing at the virtual touchscreen

With this approach, users can use their finger to interact with the virtual touchscreen as if it was a real touchscreen (Figure 13).

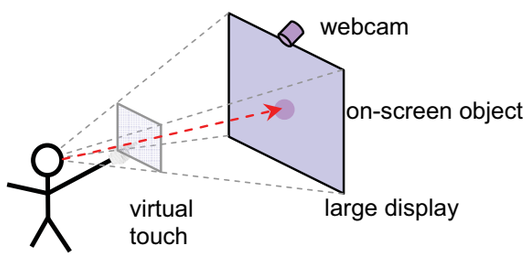


Fig. 13. A user interacting with the large display using their fingertip through a virtual touchscreen at arms' length

The user is therefore restricted to using their fully stretched arm, so that a virtual touchscreen can always be approximated at the end of their fingertip, eliminating the guesswork for the user to find the touchscreen. From the experiment in section 3.3, the majority of subjects were observed to use the full arm stretch and the eye-fingertip method to point at the target. Therefore, rather than feeling constrained, it should be natural for the user to point in our system.

An advantage of this approach is that it provides a more accurate estimation of the fingertip position, as the distance between finger and display can now be estimated from the position of the user. The other advantage is that the virtual touchscreen is re-adjusted as the user moves. In previous interactive systems, the interaction volume, and therefore the virtual touchscreen, is static. To interact with such systems, the user must determine the interaction area by initially guessing and/or through a feedback loop. While the user moves around, the interaction area does not move correspondingly, it is therefore necessary to re-adjust their hand position in order to point to the same target. Conversely, in our approach, as the user's location is known, the virtual screen adjusts to the user accordingly, while staying in front of the user. The user will always be able to "find" the virtual touchscreen as it is always within their view frustum (where the origin is at the user's eye position). As long as the user extends their hand within the bounds of the view frustum (or visibly the large display) they can always interact with the system. By taking into account the users' location,

the dynamic virtual touchscreen enables users to roam around the room and still be able to interact with the display.

4.4 Interaction Model

The moment the user touches a virtual object on the virtual touchscreen, the dTouch model can be used to define this targeting action, as illustrated in Figure 14.

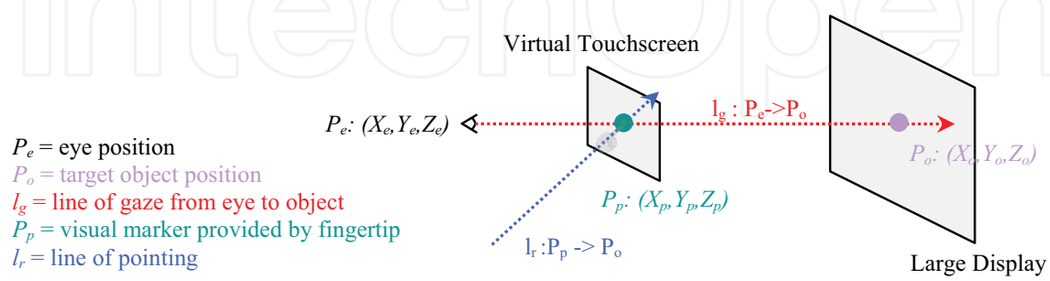


Fig. 14. The dTouch model for targeting with the virtual touchscreen.

In our system, the task for the user is to move their fingertip (P_p) to the line of gaze from eye to target (l_g) so that it aligns with the object (as seen from the user's eye). P_p is also the location of the virtual touchscreen. The object on the large display is indicated by, P_o, which is unreachable to the user, when used at a distance. When the eye, fingertip and on-screen object coincide, the dTouch model is in action and the virtual touchscreen is automatically present. The use of dTouch in this case can be considered a distant touch, a touch that occurs from afar. When the user walks towards the large display and is able to touch it physically, P_p and P_o coincides. The fingertip touches the target. The virtual touchscreen coincides with the large display, making up a large touchscreen. In this case, the Touch model applies. In practice, due to the placement and angle of the camera, the user may not be able to interact with the display at such close proximity, as the fingertip may be out of the camera's view.

4.5 Fingertip Interaction

To select an on-screen object, it is expected that users will use the virtual touchscreen as a normal touch screen where they select objects by using a forward and backward motion. However, it may be difficult for the user to know how far they have to push forward before the system will recognize the selection. Furthermore, since we are only using a single web-camera, it may be difficult to capture small changes in the distance of the fingertip from the image. One possible solution is to use dwell selection, where the user has to stay motionless at a particular position (with a given tolerance) for a specified time (typically around one second).

5. Monocular Positions Estimation

To demonstrate the feasibility of our system design using the dTouch model, a prototype was implemented. In this section, we present the method used for this implementation.

The aim is to produce a method for finding the head and fingertip position in 3D space, as well as the resultant position, all from a 2D camera image. It should be noted that cameras will become cheaper and better with time, however, the contributions here can be used as a basis for further advancement in this field for the years to come.
Our proposed interactive system is designed to be used for large displays together with the use of a webcam. The webcam is positioned on top of the display to detect the users pointing direction. The pointing direction is estimated by two 3D positions in space the eye and the fingertip. As users will be using the dTouch model of pointing with the use of a virtual touchscreen, the first position is the eye position and the second is the fingertip position from the user. To estimate the pointing direction and the final resulting point, we divide the process into three steps:

1) eye position (E) estimation
2) fingertip position (F) estimation
3) resultant point (P) estimation

The first two steps both involve finding two points in 3D space given a single 2D image. In our system, the z-coordinate is the missing information. To acquire extra information, constraints from the environment are often exploited. It is possible to use known size of familiar objects in the scene within an unknown environment (Torralba & Oliva, 2002). We proposed that it is also possible to use the kinematic constraints provided naturally by the human body (Zatsiorsky, 1998). This has the advantage of being more robust to different users and is indifferent to the environment. This is the main approach that we will use in this work.
Face detection is used to detect the position at the dominant eye, while depth (from the camera) is determined from the width of the face (Cheng & Takatsuka, 2005). The fingertip is detected as the lowest skin colour pixel from the view of the camera (Figure 15a). The depth of the fingertip is calculated by intersecting a sphere (where the center is at the user's shoulder, using arm's length as the radius) and a line vector from the center of camera through the detected fingertip in the camera's image plane, and towards the user's fingertip (Figure 15b).

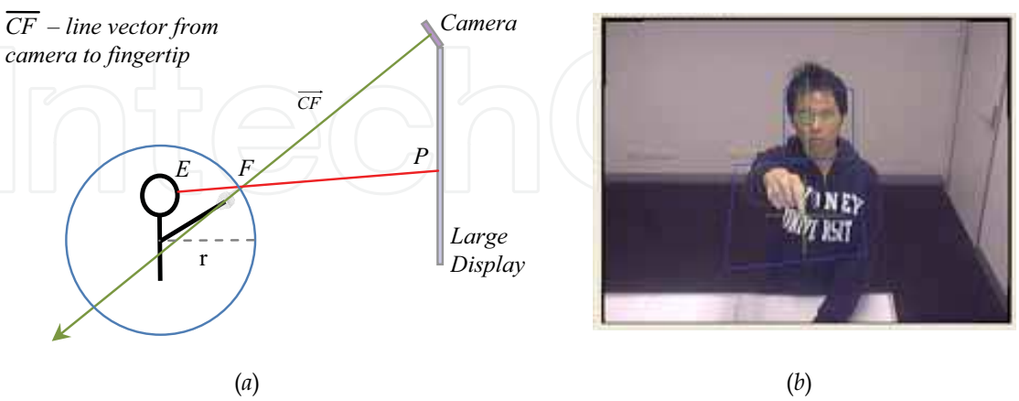


Fig. 15. (a) The fingertip position is estimated by the intersection of the line vector and the arm's sphere (b) The webcam's view highlighting the estimated positions of the eye and fingertip, and the size and position of the virtual touchscreen.

We have also introduced the use of Kalman filtering on the detected eye positions as well as the final estimation to increase stability.

6. Usability Evaluation

A usability experiment was conducted to evaluate this proof-of-concept prototype. We investigated the minimum target size that can be selected using our system with twenty-two volunteers. The body measurements of these subjects were collected and adjusted for their hand preference and eye dominance. They were then asked to stand at a distance of 160cm from the display and performed the calibration process by pointing to the four corners of the display. The display was 81.5cm x 61.5cm with a resolution of 1024x768. The webcam used was Logitech QuickCam Pro 4000 at a resolution of 320x240. A circular target was placed at the center of the display. The user was asked to point to the target for 5 seconds keeping their hand steady. This was done directly after the calibration process so that factors such as change in user's posture were reduced to a minimum. Circular targets were chosen as it provides a uniform acceptable distance from the center of target, compared to square targets. We selected six target size, the smallest being 50 pixels, while the largest 175 pixels in diameter. The effective accuracy required to select the target is half the target size.

Target Size (diameter in px)	50	75	100	125	150	175
Effective pointing accuracy required (distance from target in px)	25	37.5	50	62.5	75	87.5

Table 3. Target size and their corresponding effecting pointing accuracy required

This study was a within-subjects study, where each subject performed the six conditions. Subjects were asked to perform 3 blocks of these trials. The order of the targets was randomized to avoid ordering effects. A time limit of five seconds was imposed on each selection task. Unsuccessful selection occurs at two occasions: 1) when the user had not selected the target after the time limit and 2) when the user was pointing inside the target but had not stabilized enough to activate a dwell.
In summary, the experimental design was: 22 subjects x 6 targets x 3 blocks of trials = 396 pointing trials
The number of successful selection for each target size was added up, giving a total of 66 trials for each target size. We classified the trials into three categories:

In target & within time limit	-	Successful selection
In target but over time limit	-	Borderline
Out of target & over time limit	-	Unsuccessful selection

The case when the trials are in target but went over the time limit was classified as borderline because strictly speaking, they would have selected the target if more time was given. The delay in selection may have been due to a number of reasons:
- hand jitter causing unstable pointing position, thereby increasing time required to dwell
- pointing position may have been inside the target in one frame and outside in the next frame due to hand jitter
- slow hand movement from the subject to begin with

Because of all these uncertainties, we listed these as borderline. Figure 16 shows the percentage of successful and unsuccessful trails for each target size. As can be seen, the number of successful trials increases as the target size increase, while at the same time the number of unsuccessful trials decreases at a similar rate. A peak of 72.7% success was observed at the largest target size. An increase of 19.7% for successful selection was observed from size 50 to size 62.5px, while only a modest increase of 6.1% was observed from size 62.5 to 75px.

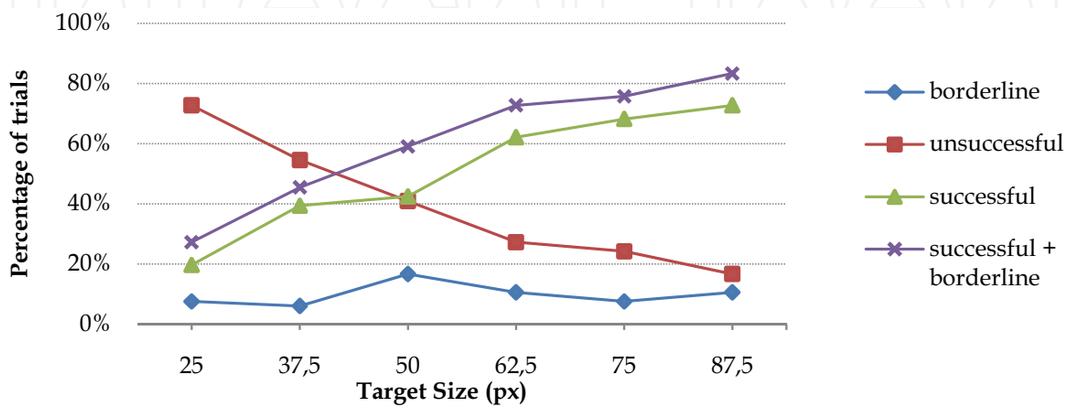


Fig. 16. Percentage of successful and unsuccessful trials for each target

Borderline cases are overall quite low in numbers, on average around 10% of all trials. If borderline cases are included as successful selection, we can observe a peak of 83.3% success rate. With the combined success rate, a slowing down in improvement can again been seen from size 62.5 to 75px (3.1% increase compared to an increases of 13.6% from 50 to 62.5px). This may suggest a target size of 62.5 as an optimal size, as further increase in size does not result in a consistent gain.

We can conclude that a target size of 62.5px (125px diameter) is the most suitable choice for system such as ours. On a 1024x768 screen, one can fit around 8 targets horizontally, and 6 targets vertically, a total of 48 targets. In terms of physical dimensions in our setup, this translates to around 99.5mm for each target both vertically and horizontally. Further investigation may be under taken to refine the target resolution than those used in this evaluation.

6.1 Limitations

With current segmentation techniques, it is still difficult to detect the fingertip from a frontal view. Subjects were asked to lower their fingertip so that it would appear as the lowest skin colour pixel. Many subjects felt that this was awkward and uncomfortable after prolonged use. However, as this style of pointing was used in all conditions, users' ability to use a particular style of pointing for target selection was not impaired.

Users' body postures were restricted as slight deviation would increase system estimation error. This was seen by the user during the calibration phase of the system. These estimation errors were found to have stemmed from the inaccuracy of the face detector used. Improving the face detection algorithm would minimize such errors.

In light of these limitations and constraints, it was felt that the results were not significantly distorted and are sufficient for testing on our prototype.

7. Summary

In this chapter, we have investigated the interaction paradigms for bare-hand interaction at a distance. Through the initial studies, we found that full arm stretch was the most common pointing strategy, while the most accurate strategy was when users line up the target with their eye and fingertip. From the observations, we systematically analysed the various natural pointing strategies and formalized geometric models to explain their differences. The dTouch model was found to give the most accurate strategy and we recommend the use of this model for designing future interactive systems that requires interaction at a distance.
The dTouch interactive system was designed and implemented as an example that uses the dTouch pointing strategy. We exploited geometric constraints in the environment, and from this we were able to use monocular computer vision to allow bare-hand interaction with large displays. The result of the experimental evaluation confirmed that using the dTouch technique and a webcam to recover 3D information for interaction with large display is feasible. Models developed and lessons learnt can assist designers to develop more accurate and natural interactive systems that make use of human's natural pointing behaviours.

8. Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

9. References

3DV Systems, ZCam http://www.3dvsystems.com/

Biever, C. (2005) Why a robot is better with one eye than two *New Scientist.* Reed Business Information Ltd.

Bowman, D. A. & Hodges, L. F. (1997), 'An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments', *in Symposium on Interactive 3D Graphics*, pp 35-38.

Cheng, K. & Pulo, K. (2003), 'Direct Interaction with large-scale display systems using infrared laser tracking devices', *in Proceedings of the Australian Symposium on Information Visualisation (invis '03)*, ACS Inc., Adelaide, pp 67-74.

Cheng, K. & Takatsuka, M. (2005), 'Real-time Monocular Tracking of View Frustum for Large Screen Human-Computer Interaction', *in Proceedings of the 28th Australasian Computer Science Conference*, Estivill-Castro, V., Newcastle, Australia, pp 125-134.

Cheng, K. & Takatsuka, M. (2006), 'Estimating Virtual Touchscreen for Fingertip Interaction with Large Displays', *in Proc. OZCHI 2006*, ACM Press., pp 397-400.

Cipolla, R. & Hollinghurst, N. J. (1998) In *Computer Vision for Human-Machine Interaction*(Eds, Cipolla, R. and Pentland, A.) Cambridge University Press, pp. 97-110.

Colombo, C., Bimbo, A. D. & Valli, A. (2003) Visual Capture and Understanding of Hand Pointing Actions in a 3-D Environment, *IEEE Transactions on Systems, Man, and Cybernetics,* **33(4),** pp 677-686.

Crowley, J. L., Coutaz, J. & Berard, F. (2000) Things that See, *Communications of the ACM,* **43(3),** pp 54-64.

Dietz, P. & Leigh, D. (2001), 'DiamondTouch: A Multi-User Touch Technology', *in Proc. UIST '01*, ACM Press, pp 219-226.

Eisenstein, J. & Mackay, W. (2006), 'Interacting with Communication Appliances: An evaluation of two computer visionbased selection techniques', *in CHI*, ACM Press., Montreal, Quebec, Canada, pp 1111-1114.

Han, J. (2005), 'Low-Cost Multi-Touch Sensing through Fustrated Total Internal Reflection', *in UIST '05*, ACM Press., Seattle, Washington, pp 115-118.

Hinckley, K. (2003) In *The Human-Computer Interaction Handbook*(Eds, Jacko, J. A. and Sears, A.) Lawrence Erlbaum Associates, Inc, pp. 151-168.

Hung, Y.-P., Yang, Y.-S., Chen, Y.-S., Hsieh, I.-B. & Fuh, C.-S. (1998), 'Free-Hand Pointer by Use of an Active Stereo Vision System', *in Proceedings of 14th International Conference on Pattern Recognition (ICPR)*, Brisbane, pp 1244-1246.

Lee, M. S., Weinshall, D., Cohen-Solal, E., Colmenarez, A. & Lyons, D. (2001), 'A computer vision system for on-screen item selection by finger pointing', *in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 1026-1033.

Leibe, B., Starner, T., Ribarsky, W., Wartell, Z., Krum, D., Weeks, J., Singletary, B. & Hodges, L. (2000) Towards Spontaneous Interaction with the Perceptive Workbench, a Semi-Immersive Virtual Environment, *IEEE Computer Graphics and Applications,* **20(6),** pp 54-65.

Lertrusdachakul, T., Taguchi, A., Aoki, T. & Yasuda, H. (2005) Transparent eye contact and gesture videoconference, *International Journal of Wireless and Mobile Computing,* **1(1),** pp 29-37.

Lock, A. (1980), 'Language Development: past, present and future.', *in Bulletin of British Psychological Society 33*, pp 5-8.

Mase, K. (1998) In *Computer Vision for Human-Machine Interaction*(Eds, Cipolla, R. and Pentland, A.) Cambridge University Press, pp. 53-81.

Matsushita, N. & Rekimoto, J. (1997), 'HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall', *in Proc. UIST '97*, ACM Press, pp 209-210.

McNeill, D. (1992) *Hand and mind: what gestures reveal about thought,* University of Chicago Press.

Microsoft Surface http://www.microsoft.com/surface/index.html

Nickel, K. & Stiefelhagen, R. (2003), 'Pointing Gesture Recognition based on 3D-Tracking of Face, Hands and Head-Orientation', *in Proceedings of ICMI '03 International Conference on Multimodal Interfaces*, ACM Press, Vancouver, pp 140-146.

Norman, D. A. (1986) *Cognitive Engineering,* Lawrence Erlbaum Associates, Inc., New Jersey.

Oka, K., Sato, Y. & Koike, H. (2002) Real-Time Fingertip Tracking and Gesture Recognition, *IEEE Computer Graphics and Applications,* **22(6),** pp 64-71.

Olsen, D. R. & Nielsen, T. (2001), 'Laser Pointer Interaction', *in Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'01)*, ACM Press., Seattle, WA, pp 17-22.

Pierce, J., Forsberg, A., Conway, M., Hong, S. & Zeleznik, R. (1997), 'Image Plane Interaction Techniques In 3D Immersive Environments', *in Symposium on Interactive 3D Graphics*, pp 39-43.

Pinhanez, C. (2001), 'Using a Steerable Projector and a Camera to Transform Surfaces Into Interactive Displays', *in CHI 2001*, IBM T J Watson Research Center, pp 369-370.

Rakkolainen, I. (2003), 'MobiVR - A Novel User Interface Concept for Mobile Computing', *in Proceedings of the 4th International Workshop on Mobile Computing*, Rostock, Germany, pp 107-112.

Rekimoto, J. (2002), 'SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces', *in Proc. CHI '02*, ACM Press, pp 113-120.

Ringel, M. & Henry Berg, Y. J., Terry Winograd (2001), 'Barehands: Implement-Free Interaction with a Wall-Mounted Display', *in Proc. of CHI 2001*, ACM Press, pp 367-368.

Saxena, A., Chung, S. H. & Ng, A. Y. (2007) 3-D Depth Reconstruction from a Single Still Image, *International Journal of Computer Vision,* **76(1),** pp 53-69.

DViT: Digital Vision Touch Technology, White Paper, SMART Technologies Inc, 2003 http://www.smarttech.com/dvit/DViT_white_paper.pdf

SONY EyeToy http://www.eyetoy.com

Sun Microsystems Inc., definition of View Frustum, http://java.sun.com/products/java-media/3D/forDevelopers/j3dguide/glossary.doc.html#47353

Takatsuka, M., West, G. A. W., Venkatesh, S. & Caelli, T. M. (2003) Low-cost interactive active range finder, *Machine Vision and Application,* **14(**pp 139-144.

Takenaka Corporation, Input System Which can Remotely Click the Screen Without hand Contact Developed http://www.takenaka.co.jp/takenaka_e/news_e/pr9806/m9806_02_e.htm

Taylor, J. & McCloskey, D. I. (1988) Pointing, *Behavioural Brain Research,* **29(**pp 1-5.

Gale Encyclopedia of Neurological Disorders, The Gale Group, Inc. http://www.answers.com/topic/proprioception

Torralba, A. & Oliva, A. (2002) Depth Estimation from Image Structure, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **24(9),** pp 1226-1238.

Tosas, M. & Li, B. (2004), 'Virtual Touch Screen for Mixed Reality ', *in European Conference on Computer Vision*, Springer Berlin, Prague, pp 72-82.

Wellner, P. (1993) Interacting with Paper on the DigitalDesk, *Communications of the ACM,* **36(7),** pp 87-96.

Wikipedia, the free encyclopedia, definition of View Frustum http://en.wikipedia.org/wiki/View_frustum

Zatsiorsky, V. M. (1998) *Kinematics of Human Motion,* Human Kinetics Publishers.

IntechOpen

Human-Computer Interaction
Edited by Inaki Maurtua

ISBN 978-953-307-022-3
Hard cover, 560 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

In this book the reader will find a collection of 31 papers presenting different facets of Human Computer Interaction, the result of research projects and experiments as well as new approaches to design user interfaces. The book is organized according to the following main topics in a sequential order: new interaction paradigms, multimodality, usability studies on several interaction mechanisms, human factors, universal design and development methodologies and tools.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Kelvin Cheng and Masahiro Takatsuka (2009). Interaction Paradigms for Bare-Hand Interaction with Large Displays at a Distance, Human-Computer Interaction, Inaki Maurtua (Ed.), ISBN: 978-953-307-022-3, InTech, Available from: http://www.intechopen.com/books/human-computer-interaction/interaction-paradigms-for-bare-hand-interaction-with-large-displays-at-a-distance

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen