Tangible Mixed Reality Desktop for Digital Media Management

Stefan Winkler, Hang Yu, ZhiYing Zhou

Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576

ABSTRACT

This paper presents a tangible mixed reality desktop that supports gesture-oriented interactions in 3D space. The system is based on computer vision techniques for hand and finger detection, without the need for attaching any devices to the user. The system consists of a pair of stereo cameras that point to a planar surface as the work bench. Using stereo triangulation, the 3D locations and directions of the user's fingers are detected and tracked in the space on and above the surface. Based on our 3D finger tracking technique, we design a few simple multi-finger gestural interactions for digital media management. The system provides a convenient and user-friendly way of manipulating virtual objects in 3D space and supports seamless interactions with physical objects.

1. INTRODUCTION

1.1. Problem Statement

The design of human-computer interfaces is always a challenge. Traditional input functionality via keyboard and mouse is rather inefficient, so multi-modal techniques using vision, sound, tracking sensors and projective displays have been developed to provide abundant input and output channels between human and computer. However, even with these techniques, users are required to adapt to the devices and learn the use of the associated interfaces. Furthermore, the use of such systems still differs significantly from everyday interactions with the physical world. For example, the sorting and arranging of media files on a computer can be very tedious as compared to using photo prints. As an additional impediment, computer work is mostly a solitary experience, and teamwork happens sequentially through a series of incremental changes, because technology does not make it easy for team-mates to work together on a common task.

1.2. Related Work

There has been extensive research conducted on human-computer interfaces to provide tangible user interaction combing the physical and virtual world. These interactions largely rely on the real-time identification of user gestures. We distinguish tracking methods based on specialized sensing equipment from those based on computer vision techniques.

Methods based on specialized sensing equipment are a direct way of tracking the user's actions with embedded electronics. For example, in the systems of DiamondTouch[1] and SmartSkin,[2] a mesh of capacitive electrodes is attached to the interface to track the position of the user's fingers or hands. Based on DiamondTouch, Wu et al. developed a prototype for room furniture arrangement by multiple finger and gestural input.[3] These hardware-based methods can provide accurate and fast interactions, however the specialized devices required inhibit their wide-spread use.

Alternatively, computer vision techniques are employed in the tabletop display community to detect and recognize objects on the interface. DigitalDesk[4] is one of the earliest works in this domain to integrate physical objects and digital information on a desk equipped with a camera and a projector. For example, the number in a physical document pointed to by the user can be read into a projected calculator on the desk.

E-mail of corresponding author: winkler@nus.edu.sg

Perceptive Workbench[5] and PlayAnywhere[6] apply shadow shape analysis to fingertip detection. The position and pointing direction of the fingertips is estimated from the objects' shadow projected on the interface. Infrared cameras are used in several systems to detect the users' fingertips in real time such as EnhancedDesk[7] and Barehands.[8] The advantage of using infrared cameras on gesture recognition and tracking is their robustness to changing lighting condition and complex backgrounds.

The detection of touch location can be achieved by multiple channels, for example stereo cameras. In the system of TouchLight,[9] two cameras are installed behind a semi-transparent screen. The outputs from these two cameras are fused to detect the location of touch on the screen. Similar work can be found in the 4D Touchpad[10] and Visual Touchpad.[11] These works are related to the one proposed in this paper, but their fingertip detection mechanism and the supported interactions are constrained to a planar 2D surface. The advantage of vision-based techniques lies mainly in their non-intrusive and relatively cheap setup.

1.3. Proposed System

In this paper, an intuitive human-computer interface for a tangible mixed-reality desktop is proposed. With vision-based finger tracking techniques, we can support direct interactions from the user's hands. The 3D location and direction of the fingers are tracked using a pair of stereo cameras. This extends the work space from the planar desktop surface to the complete 3D space within the user's (and the cameras') reach, providing users much more freedom of operation. We demonstrate the system's capabilities using digital media management tasks.

1.4. Outline

The paper is organized as follows. Section 2 describes the design of the overall system. Section 3 introduces a few interaction techniques adopted in the application of digital media management. The strengths and weaknesses of our system as well as directions for future work are discussed in Section 4. Section 5 concludes the paper.

2. SYSTEM DESIGN

2.1. Hardware Setup

The hardware setup of our system is similar to Visual Touchpad.[11] As shown in Figure 1, it consists of a desktop with a pair of stereo cameras attached in front of or above the work bench. Interactions between hands and virtual objects are confined to the space above the tabletop which is visible to both cameras. The desktop is covered with white paper to improve finger tracking accuracy (more on this in Section 4).

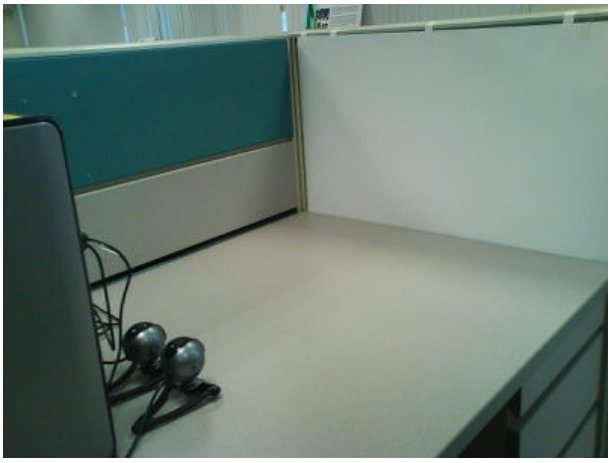

Figure 1. System configuration.

2.2. Finger Tracking in 2D

The technique of finger tracking in planar space is based on the algorithm by Hardenberg.[12] Here we give a brief review of the method. It comprises three major steps, namely hand segmentation, fingertip detection and classification.

2.2.1. Hand Segmentation

This step can be considered a preliminary step as its main objective is to extract the relevant information from the image. Only the areas containing hands are selected as the areas of interest. The technique used in this step is based on image differencing. With image differencing, moving objects in the foreground are extracted by comparing successive frames. To improve the robustness to changing illumination, they modified the technique so that the reference image used for differencing is updated with every new frame:[13]

$$I_r^t(x,y) = \frac{N-1}{N} \times I_r^{t-1}(x,y) + \frac{1}{N} \times I_n(x,y)\,, \tag{1}$$

where I_r^t denotes the reference image at time t, I_n denotes the newly updated image, and N denotes the number of frames. The reference image thus becomes a running average over all previous frames, whose weight decreases with time.

2.2.2. Fingertip Detection

After obtaining the areas of interest containing hands, the next step is to detect the fingertips in these areas. In the algorithm, the fingertips are located according to two properties illustrated in Figure 2:

- The fingertips can be formalized as closed areas with a circular border. The diameter of the circle is determined from the finger width.
- The two sides of a finger can be formalized as a long chain of non-filled pixels between the finger and the outside searching square, and a short chain of filled pixels which represents the finger.

By applying these two properties, the fingers and fingertips can be detected fast and reliably. The locations of the fingertips are determined by the shape property, and the direction of the fingers is obtained by tracking the location of the fingertip to the middle point of the short chain within the searching square (as indicated by the arrow in Figure 2).

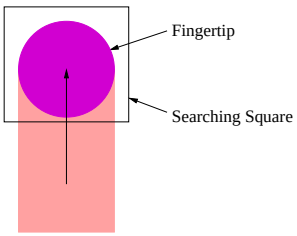

Figure 2. Fingertip detection.

2.2.3. Fingertip Classification and Tracking

Once the fingertips are detected, they are classified according to their relationship with the hand through standard connected component analysis. From the locations and directions of the fingers, the center of the palm can be estimated, and the positions of the fingers relative to each other are determined.

The system does not include a stage for tracking as such; the locations of the fingertips are re-computed for each new frame. However, confining the search space to the areas of interest ensures the efficiency of the processing. The advantage of this method over full-fledged tracking algorithms that keep records of a detected

object's position over time is that it provides more stability in case of fast finger movements. This is useful as the hand can move quite fast in certain interactions thus causing "jumps" over frames.

Figure 3 shows some results on applying the finger detection algorithm to images from the pair of stereo cameras. Each row corresponds to one camera. The left picture in each row shows the averaged background for detection as defined by Eqn. 1. The center picture shows the extracted foreground in the form of the segmented hand shape. The right picture shows the detection of the fingertips, with the location of the tips and the direction of the fingers marked on them.

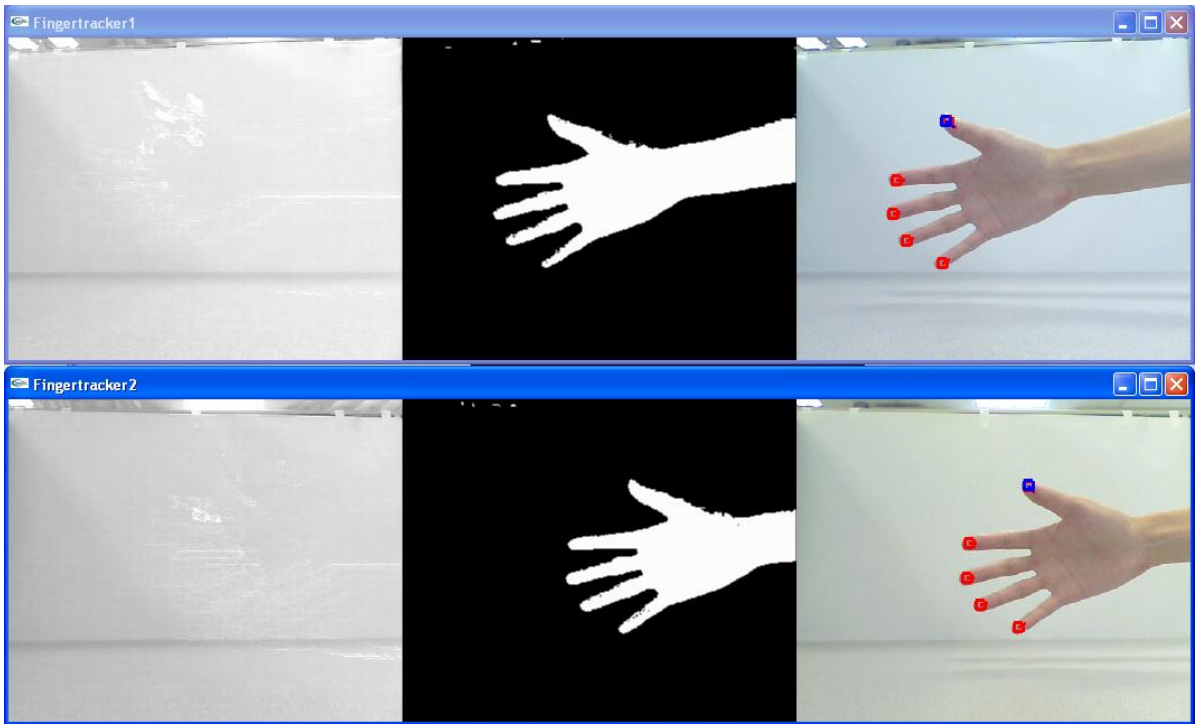

Figure 3. Fingertip detection by the pair of stereo cameras. Top/bottom row: left/right camera. From left to right: Averaged background, segmented foreground, detected fingertips.

2.3. Localization in 3D Space

Using calibrated stereo vision, we can compute the distance of any object in the field of view of both cameras and thus obtain the location of the fingertips in 3D space. This procedure requires a calibrated camera setup and stereo triangulation.

Calibration is essential in a 3D vision-based camera tracking system. It will affect the tracking accuracy directly. We use the Matlab-based Camera Calibration toolbox[14] for this step, from which we obtain the internal and external parameters of the cameras. This is done offline as part of the initialization of the system.

The 3D locations of the fingertips detected from the pair of stereo cameras are calculated by stereo triangulation. Figure 4 illustrates this process. A 3D line in the coordinate system of each camera reference frame is projected onto two 2D lines on each camera image plane. Once we know the positions of the 2D lines and calibration parameters of the stereo cameras, the 3D information of the line is obtained.

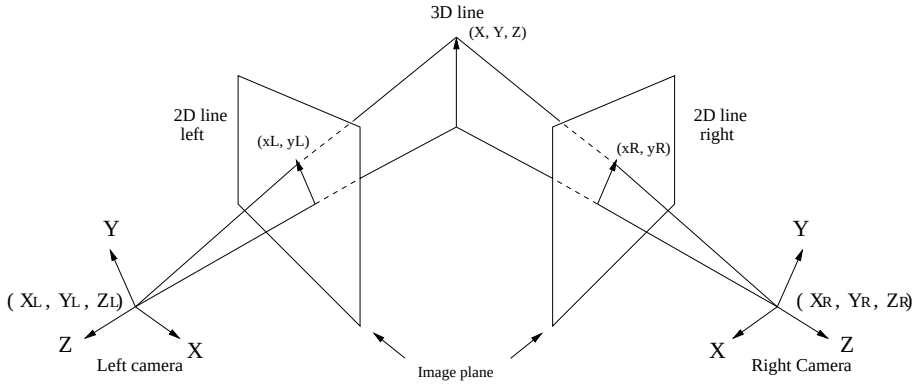

Figure 4. Stereo triangulation from a pair of cameras.

Stereo triangulation is performed as follows.[14] Let $\bar{X}_L = \{X_L\ Y_L\ Z_L\}^T$ and $\bar{X}_R = \{X_R\ Y_R\ Z_R\}^T$ be the coordinates of a point $P(X, Y, Z)$ in the left and right camera reference frames. The relation between these two vectors is given by the following equation:

$$\bar{X}_L = \mathbf{R}\bar{X}_R + \mathbf{T}, \tag{2}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the rotation and translation matrices from right to left camera frames. These two matrices are obtained from stereo camera calibration.

Furthermore, let $\bar{x}_L = \bar{X}_L/Z_L = \{x_L\ y_L\ 1\}^T$ and $\bar{x}_R = \bar{X}_R/Z_R = \{x_R\ y_R\ 1\}^T$ be the coordinates of the perspective projection vectors p_L and p_R of P on the left and right camera image planes. The stereo triangulation problem is to determine $\bar{X}_L$ and $\bar{X}_R$ given $\bar{x}_L$ and $\bar{x}_R$. Eqn. 2 can be written as:

$$Z_L\bar{x}_L = Z_R\mathbf{R}\bar{x}_R + \mathbf{T}, \tag{3}$$

which leads to:

$$\left[\ -\mathbf{R}\bar{x}_R \quad \bar{x}_L\ \right] \begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = \mathbf{T}. \tag{4}$$

The solution for Eqn. 4 can be obtained by least square minimization. Let $\mathcal{A} = \left[\ -\mathbf{R}\bar{x}_R \quad \bar{x}_L\ \right]$:

$$\begin{bmatrix} Z_R \\ Z_L \end{bmatrix} = \left(\mathcal{A}^T\mathcal{A}\right)^{-1}\mathcal{A}^T\mathbf{T}. \tag{5}$$

The explicit expression for Z_R can be expanded from Eqn. 5:

$$Z_R = \frac{\|\bar{x}_L\|^2\langle\bar{\alpha}_R, \mathbf{T}\rangle - \langle\bar{\alpha}_R, \bar{x}_L\rangle\langle\bar{x}_L, \mathbf{T}\rangle}{\|\bar{\alpha}_R\|^2\|\bar{x}_L\|^2 - \langle\bar{\alpha}_R, \bar{x}_L\rangle^2}, \tag{6}$$

where $\bar{\alpha}_R = -\mathbf{R}\bar{x}_R$ and $\langle ., . \rangle$ is the standard scalar product operator. Finally, $X_R = x_R Z_R$ and $Y_R = y_R Z_R$.

2.4. Inserting Virtual Objects

Our goal is to manipulate virtual objects directly using fingers and hand gestures. To aid in the insertion of virtual objects into the real world, we also compute the direction of the fingers in 3D space, in addition to the location of the fingertips. The direction of the fingers is determined by the finger detection algorithm as illustrated above in Figure 2.

Figure 5 shows the results of attaching a digital photo onto a finger in 3D space. The first two pictures in each row show the fingertip detection results from the pair of stereo cameras. The right-most picture shows the effect on overlaying an image on the fingertip. The image is oriented such that the finger direction becomes the normal vector of the image plane.

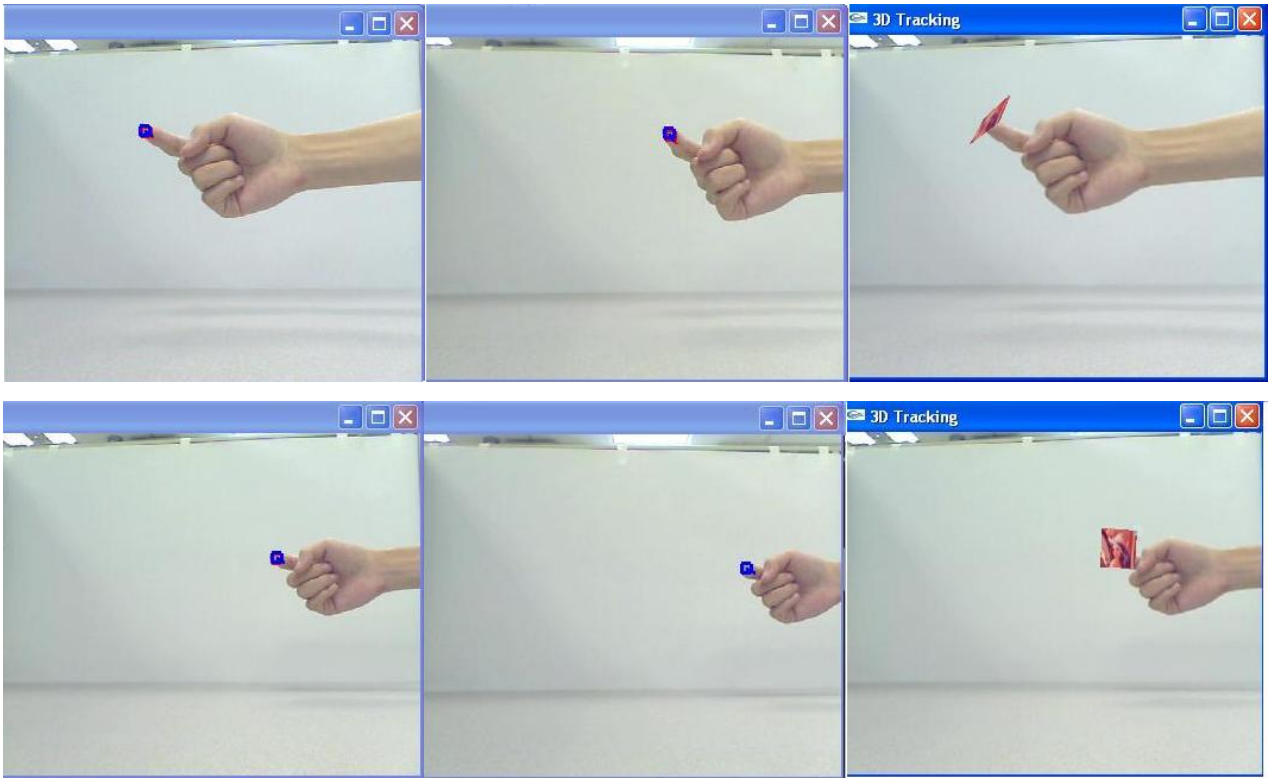

Figure 5. Two examples of attaching a virtual object (an image) to the fingertip. The image plane is chosen perpendicular to the finger direction.

3. INTERACTION TECHNIQUES

We demonstrate our system's capabilities using a few simple digital media management tasks. Four examples of interaction techniques with virtual and real objects are shown in the following:

- **Selecting an object.** The first step of any interaction is to pick an object for processing. Figure 6 shows an example of this operation. The photo closest to the pointing finger is picked and enlarged to mark is as selected.
- **Dragging an object.** Once an object is selected, it can be dragged along by moving the finger. The location and direction of the object is updated from the finger tracking.
- **Resizing an object.** To resize an object, a two-finger gesture is recognized. The magnitude of the adjustment is controlled by the separation between the two fingers. Figure 7 illustrates the operation of enlarging a picture.
- **Deleting an object.** To delete an object, it can be dragged into a physical bin on the table. Figure 8 shows the process of this operation. The paper box acts as the trash bin.

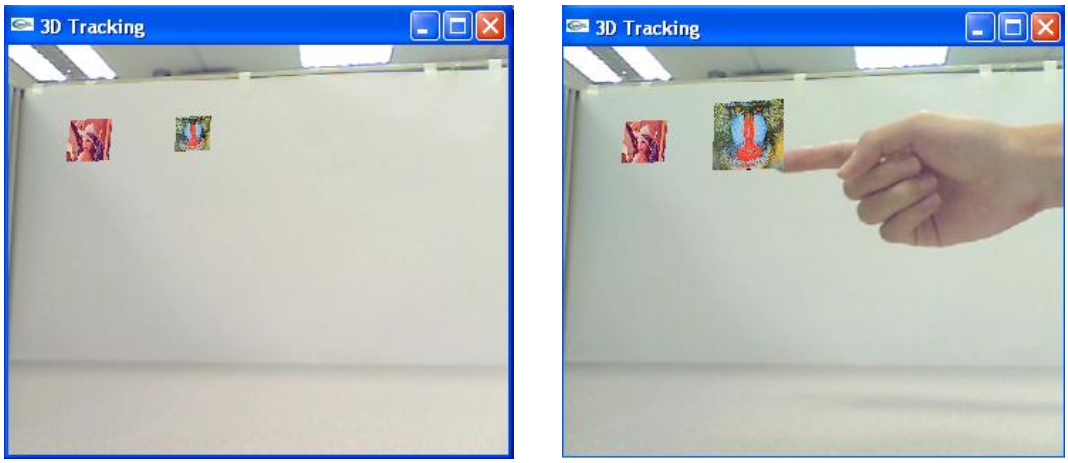

Figure 6. Selecting an image by pointing a finger on the target.

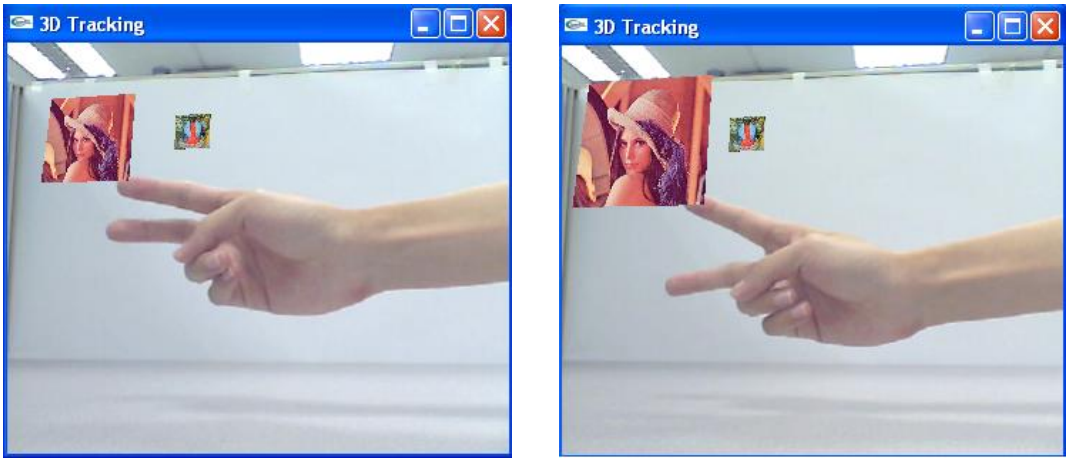

Figure 7. Enlarging an image using two fingers.

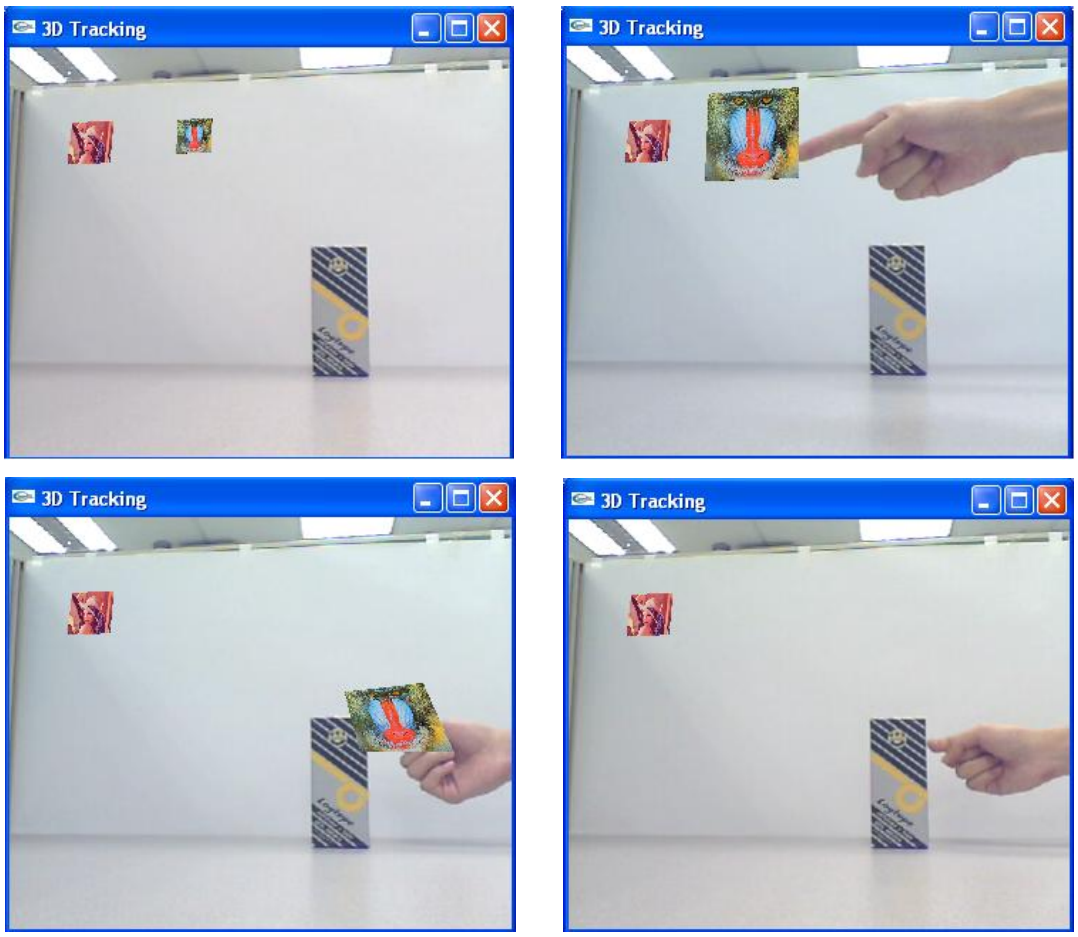

Figure 8. Deleting an image by dragging it into the physical paper bin.

4. DISCUSSION

4.1. Robustness

The fingertip detection algorithm employed in our system is quite robust. It can achieve consistent detection at 28 frames per second on a 352x288 video resolution on a PC configuration with a Pentium D at 2.80 GHz. The high speed is partly due to the absence of a tracking mechanism in the method.

However, there are several sources of instability, in particular the ambient light conditions and the layout of the background. False detection will occur with changing ambient illumination or if there are objects similar in shape to the fingertips in the scene. Meanwhile, dark-colored objects, dark areas in the background, or shadows cast by the user can also interfere with finger detection. The reason is the image differencing between successive frames that is part of the hand segmentation. Dark-colored objects do not provide sufficient contrast to extract the foreground from the background. This is also the reason that we placed a white surface in the background. We are considering the installation of spot lights in well-chosen locations on top of the work space as a possible solution to this problem.

We also found the algorithm to be more robust if the palm of the hand rather than its back is facing the camera. This is due to the fact that the fingernails can cause problems with the detection algorithm in some cases. When the fingers are close to the camera, the fingernails can be wrongly treated as the background instead of part of the fingers, resulting in shortened fingers and incorrect fingertip locations. This seems to be due to the lighter color of the fingernails.

4.2. Display Method

In our proposed system, digital media is processed in an augmented environment in which virtual and real objects are combined. Depending on the hand position and the operation, there can be occlusions of virtual object projections and real objects such as the fingers manipulating the objects. In our method to render the interaction between virtual objects and the real world, the virtual objects are always in front. Alternatively, as we know the precise positions of the virtual objects, we can render them according to their depth with respect to the real objects, such as the fingers. The parts of the virtual objects that are occluded can be omitted from the display.

The tangible interface metaphor, that is, the tabletop in our system setup, can also be implemented with a projection-based display. This kind of setup can provide an intuitive interface for certain applications (e.g. digital media management), is less intrusive than head-mounted displays (HMD's), and facilitates integration into our daily living environments.

4.3. Camera Placement

In the current system configuration (cf. Figure1), the stereo cameras are placed relatively close together in front of a white surface board. The work area for object manipulation is the 3D space between the board and the cameras. There are some other possibilities for camera placement, which can improve system robustness for certain types of interactions, especially with multiple hands or users. For example, the cameras can be mounted on top or to the sides of the desktop, thus enhancing the stereo triangulation accuracy and finger tracking stability.

4.4. Multi-User Support

The system can be extended to support multiple work benches, allowing people to collaboratively manipulate a common set of virtual objects or play multi-player games. These work benches can be distributed and communicate via real-time remote connection.

5. CONCLUSIONS

We introduced a tangible mixed-reality desktop with interaction in 3D space. The interface is controlled using hands and fingers. It is based on an efficient real-time finger detection algorithm. With the detected location and direction of fingertips, we apply stereo triangulation to obtain the 3D parameters of the fingers. This allows us to attach virtual objects to the fingers in 3D space. We demonstrated the capabilities of the proposed interface in a simple digital photo management application. We are currently extending this system to include additional interaction techniques, and developing other applications that make use of 3D finger- and hand-based interaction in intuitive ways.

REFERENCES

1. P. Dietz and D. Leigh, "Diamondtouch: a multi-user touch technology," in *Proc. 14th Annual ACM Symposium on User Interface Software and Technology*, pp. 219–226, ACM Press, (New York, NY, USA), 2001.
2. J. Rekimoto, "SmartSkin: An infrastructure for freehand manipulation on interactive surfaces," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 113–120, 2002.
3. M. Wu and R. Balakrishnan, "Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays," in *Proc. 16th Annual ACM Symposium on User Interface Software and Technology*, pp. 193–202, ACM Press, (New York, NY, USA), 2003.
4. P. Wellner, "Interacting with paper on the DigitalDesk," *Communications of the ACM* **36**(7), pp. 87–96, 1993.
5. T. Starner, B. Leibe, D. Minnen, T. Westyn, A. Hurst, and J. Weeks, "The perceptive workbench: Computer vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks," *Machine Vision and Applications* , pp. 59–71, 2003.
6. A. D. Wilson, "PlayAnywhere: a compact interactive tabletop projection-vision system," in *Proc. 18th Annual ACM Symposium on User Interface Software and Technology*, pp. 83–92, ACM Press, (New York, NY, USA), 2005.
7. H. Koike, Y. Sato, and Y. Kobayashi, "Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system," *ACM Transactions on Computer-Human Interaction* **8**(4), pp. 307–322, 2001.
8. M. Ringel, H. Berg, Y. Jin, and T. Winograd, "Barehands: Implement-free interaction with a wall-mounted display," in *Proc. SIGCHI*, pp. 367–368, 2001.
9. A. D. Wilson, "TouchLight: an imaging touch screen and display for gesture-based interaction," in *Proc. 6th International Conference on Multimodal Interfaces (ICMI)*, pp. 69–76, ACM Press, (New York, NY, USA), 2004.
10. J. J. Corso, D. Burschka, and G. D. Hager, "The 4D Touchpad: Unencumbered HCI With VICs," in *Proc. IEEE Workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (CVPRHCI)*, 2003.
11. S. Malik and J. Laszlo, "Visual touchpad: a two-handed gestural input device," in *Proc. 6th International Conference on Multimodal Interfaces (ICMI)*, pp. 289–296, ACM Press, (New York, NY, USA), 2004.
12. C. von Hardenberg and F. Brard, "Bare-hand human-computer interaction," in *Proc. Workshop on Perceptual User Interfaces*, **15**, pp. 1–8, 2001.
13. J. Q. Stafford-Fraser, *Video-Augmented Environments*. PhD thesis, Gonville & Caius College, University of Cambridge, 1996.
14. J.-Y. Bouguet, "Camera calibration toolbox for matlab." http://www.vision.caltech.edu/bouguetj/calib_doc/, 2004.