

Wearable Virtual Tablet: Fingertip Drawing on a Portable Plane-object using an Active-Infrared Camera

Norimichi Ukita and Masatsugu Kidode
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama-cho 8916-5, Ikoma, Nara, Japan
ukita@is.aist-nara.ac.jp and kidode@is.aist-nara.ac.jp

ABSTRACT

We propose the *Wearable Virtual Tablet* (WVT), where a user can draw a locus on a common object with a plane surface (e.g., a notebook and a magazine) with a fingertip. Our previous WVT[1], however, could not work on a plane surface with complicated texture patterns: Since our WVT employs an active-infrared camera and the reflected infrared rays vary depending on patterns on a plane surface, it is difficult to estimate the motions of a fingertip and a plane surface from an observed infrared-image. In this paper, we propose a method to detect and track their motions without interference from colored patterns on a plane surface. (1) To find the region of a plane object in the observed image, four edge lines that compose a rectangular object can be easily extracted by employing the properties of an active-infrared camera. (2) To precisely determine the position of a fingertip, we utilize a simple finger model that corresponds to a finger edge independent of its posture. (3) The system can distinguish whether or not a fingertip touches a plane object by analyzing image intensities in the edge region of the fingertip.

Categories and Subject Descriptors

H [5]: 2: Input devices and strategies

General Terms

Algorithms

Keywords

Finger-drawing interface, Active-infrared camera, Wearable computer.

1. INTRODUCTION

Our broad objective is to realize wearable interfaces for expanding our information activities in daily life. We often have to input various kinds of information into a computer

when using it. As wearable technology advances, several systems using portable interfaces have been developed:

FingeRing[2]: By typing with fingertip actions that are detected by ring shaped sensors on each finger, a user can use a virtual keyboard.

WristCam[3]: With a camera attached under a user's wrist, finger motions are observed. This allows us to unobtrusively input gestures to a computer.

HIT-Wear[4]: With a camera mounted near a user's viewpoint, the shape/gesture of the hand/fingers are recognized. Then, a selective menu is superimposed on the hand region as seen on the observed image displayed on a user's HMD (Head Mounted Display).

Gesture pendant[5]: By emitting infrared rays from a camera system and observing their reflections, the system can easily detect the regions of the user's hands and classify the motions of the hands into predefined gestures.

While these interfaces are useful for specifying predefined characters and commands, it is impossible to input free contents, for example, not only characters but also arbitrary pictures and symbols.

Although the following free-content input systems have been developed, they have some disadvantages as a convenient wearable interface:

Pen device with physical sensors[6]: A user holds a pen device with gyro and acceleration sensors. With these sensors, the 2D locus of the pen device in the air can be tracked. This system, however, has the following disadvantages: (1) the user has to always carry around the pen device, (2) the user has to turn on/off a mechanical button for switching the input state from on/off to off/on, and (3) there is no tactile sensation while writing¹.

Fingertip writing with a single stroke[7]: A user writes a character with a single stroke by moving the fingertip in the air. The finger motion is observed by the head-mounted camera, and the 2D fingertip locus is displayed on the HMD. This system has an essential problem; the user has to write a character with a single stroke. To stop the input, therefore, the user has

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUT'04, January 13–16, 2004, Madeira, Funchal, Portugal.
Copyright 2004 ACM 1-58113-815-6/04/0001 ...\$5.00.

¹[8] reported that the sense of touch allows us to write and type efficiently

to get the finger out of the visual field of the camera. This troublesome operation makes it impractical.

To solve the problems in these systems, we have proposed the *Wearable Virtual Tablet (WVT)*, where a user can draw an imaginary locus with the fingertip on a common plane surface such as a notebook or a magazine. In this system, when the fingertip touches the plane surface, its locus is regarded as input. The WVT provides the following functions:

- The ability of drawing an imaginary and arbitrary locus on a plane surface using the fingertip.
- The ability of switching the input state from on/off to off/on without any mechanical switches.

In our previous WVT[1], an active-infrared camera is employed to simplify image processing and acquire 3D information of observed objects; a CCD camera captures an infrared ray emitted from infrared-LEDs near the camera.

While the active-infrared camera allows us to easily estimate the motions of a fingertip and a plane surface, the observed infrared ray is affected by several features of an object (e.g., material and color). The previous WVT using an active-infrared camera, therefore, could not work with a plane surface with complicated texture patterns: Since the reflectance of an infrared ray varies depending on gray patterns on a plane surface, these patterns make it difficult to estimate the motions of the fingertip moving on the plane.

We, therefore, improved the method for detecting and tracking a fingertip and a plane surface. Our improved method is as follows: (1) To find the region of a plane surface in the observed image, four edge lines that compose a rectangle are extracted. (2) To precisely determine the position of a fingertip in real time, we utilize a simple finger model that corresponds to the finger edge independent of its posture. (3) The system can distinguish whether or not the fingertip touches the plane surface by analyzing image intensities along the edge region of the fingertip.

2. WEARABLE VIRTUAL TABLET AND ITS ARCHITECTURE

The WVT system consists of a wearable camera and a HMD as shown in Fig.1. With this system, we can draw an arbitrary locus on a portable plane surface even while freely walking and moving our body. The position of a fingertip on the input surface is detected and tracked in the observed image. Its locus is superimposed on the current observed image displayed in the HMD. This locus allows a user to draw the locus continuously.

The following technical problems have to be solved in order to realize the WVT:

- Problem 1** Determine an input surface.
- Problem 2** Detect and track a user’s fingertip.
- Problem 3** Distinguish whether or not a user touches the input surface.
- Problem 4** Superimpose the locus of the fingertip on the current observed image.

In this paper, to make the problem 1 easy, an arbitrary rectangular surface held in a user’s hand is regarded as an

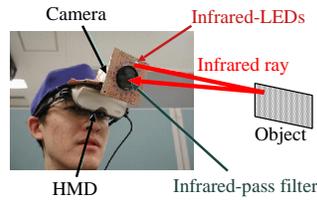


Figure 1: Architecture of the WVT: an active-infrared camera and an HMD.

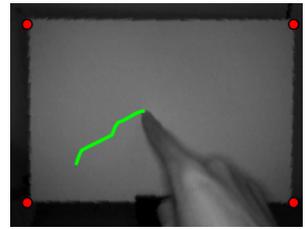


Figure 2: Fingertip drawing using the WVT.

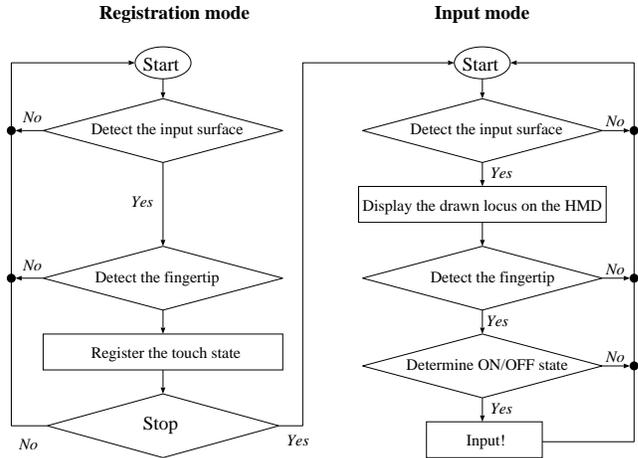


Figure 3: Flowchart of the WVT.

input surface. The system has to detect the four corners of this input surface to determine the location of the input area in the image observed by the wearable camera. The active-infrared camera we used[1] as the wearable camera will be mentioned later. Its properties drastically simplify detection of an input surface as well as estimation of fingertip motion (problem 2); the infrared image enables the system to easily detect an input surface and a user’s finger without being confused by a complicated background scene and the texture patterns on the input surface. Furthermore, the gray-scale information in the infrared image allows the system to estimate whether or not the fingertip touches the input surface, thus solving problem 3. Problem 4 can be solved by translating the fingertip locus from the square buffer image to the region of the input surface in the current image; based on the geometric configuration among the four corners of the input surface, the projection between two tetragons formed by the current region of the input surface and the buffer image can be computed.

Our active-infrared camera consists of a conventional CCD camera, infrared-LEDs and infrared-pass filter as shown in Fig.1; this camera can only capture infrared rays as a gray-scale image (Fig.4 (a)). A strong ray is represented as an area of high intensity in the observed image. Since the infrared light source is near the CCD camera, the camera mainly captures the reflections of the infrared rays emitted by this light source.

3. FINGERTIP DRAWING USING AN ACTIVE-INFRARED CAMERA

Figure3 shows the flowchart of the WVT. The WVT has two functional modes:

Registration mode: To determine whether or not a user has touched the input surface, the distribution of gray values of the pixels around the fingertip is registered in advance. For this registration, problems 1 and 2 described in Sec.2 have to be solved.

Input mode: After the registration mode, a user starts drawing. In this mode, problems 1, 2, 3 and 4 have to be solved.

In the WVT (Fig.3), the following processes are executed:

Detect the input surface The four corners of an input surface are detected.

Detect the fingertip The fingertip of a user is detected.

Register the touch state Image information around the fingertip is registered when a user touches the input surface and registered.

Display the drawn locus on the HMD The input locus drawn by a user is continuously displayed on the HMD.

Determine ON/OFF state Based on the image information registered in the function “Register the touch state”, the system estimates whether or not a user touches an input surface.

In what follows, we describe how to implement the above functions.

3.1 Detecting and Tracking a Virtual Input Surface

In our system, the four corners of an input surface have to be detected to determine its area and posture. For this purpose, we suppose that the only objects in proximity to the camera² are an input surface and a user’s hand when he/she uses the WVT.

In general, it is difficult to extract the regions of close-objects from a color image (Fig.4 (a)); In an example shown in Fig.4 (a), many background objects are observed in the captured image and make close-object extraction difficult. On the other hand, Figure4 (b) shows the observed infrared image. The intensity of the reflected infrared ray depends on the distance from the light source to an object in the scene. By employing this property, the regions of close-objects can be easily extracted from the observed infrared image (Fig.4 (c)) without a complicated method for 3D depth reconstruction. We call this image a *close-object image*.

While the WVT is working, the regions of close-objects are extracted from the infrared image (as shown in Fig.5 (a)). These regions include an input surface and a user’s hand. To realize the WVT, these regions have to be detected severally; the WVT system searches for the four corners of the input surface and the fingertip of a user. Edge detection is useful for discriminating between the regions of observed

²Hereafter, we refer to an object in front of and near the camera as a *close-object*.

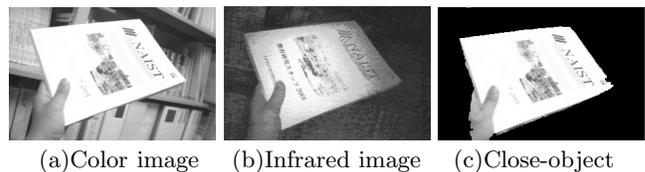


Figure 4: Close-object extraction using an active-infrared camera.

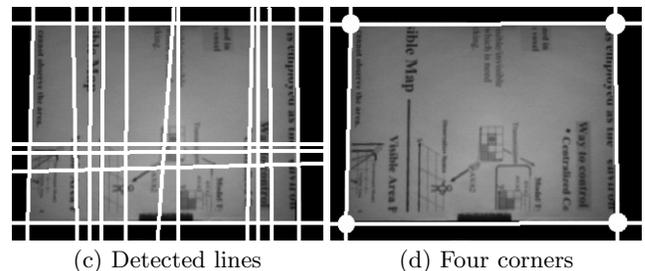
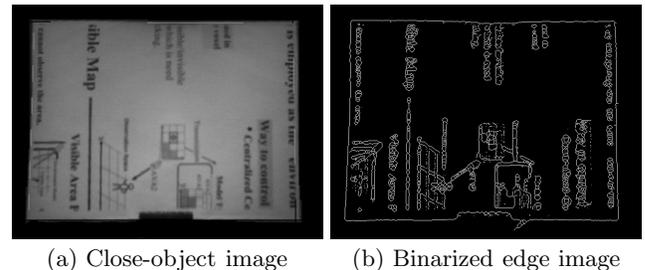


Figure 5: Detecting the corners of an input surface.

objects. We apply the *Sobel operator*³ to the close-object image to generate a *gradient image* whose pixel value is the derivative of local image values. Image binarization⁴ and edge thinning are applied to the gradient image in turn in order to obtain an *edge image* (Fig.5 (b)). With these procedures, the boundary lines of objects can be detected. Note that other edge lines (e.g., texture patterns and shadows on an observed object) are also included in the edge image. To estimate the four sides of the input surface, the fast Hough transform[9]⁵ is then executed for detecting straight lines. Since the Hough transform processes each edge pixel, edge thinning is effective for reducing processing time.

If the input surface has texture patterns that generate edge lines (as shown in Fig.5 (c)), the system has to extract the four sides from the multiple detected lines. This problem also can be solved by employing the properties of the active-infrared camera as follows:

1. Scan the edge image from the left side of the image to

³The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that correspond to edges.

⁴Each pixel value in a gradient image is binarized with the pre-defined threshold.

⁵The image taken by an actual camera is affected by *radial distortion*[10] that deforms the observed image geometrically. This results in difficulty in detecting a straight line in the image. We, therefore, correct the distortion in the image by employing the method proposed in [10].

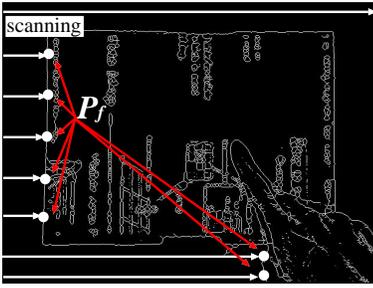


Figure 6: Scanning edge points P_f s from the left side of the image to the right.

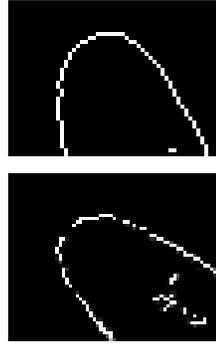


Figure 7: Edge line of a fingertip.

the right. The edge point detected first is denoted by $P_{f_i} (i = 1, \dots)$ as shown in Fig.6. Note that not only the surface but also a user's hand holding it (right-bottom part in Fig.6) are observed in this image. This scanning is also executed from the right to the left, from the top to the bottom, and the bottom to the top. Every detected point must correspond to the four sides of the input surface or the boundary edge of a user's hand.

2. Calculate the distance from each P_{f_i} to every line detected by the Hough transform. If the distance from P_{f_x} to the line L_y is small enough, P_{f_x} is voted to L_y .
3. The four lines corresponding to the four highest vote-getters are considered to be the four sides of the input surface.

The four lines are then determined. The intersection points of these lines are considered to be the four corners (Fig.5 (d)).

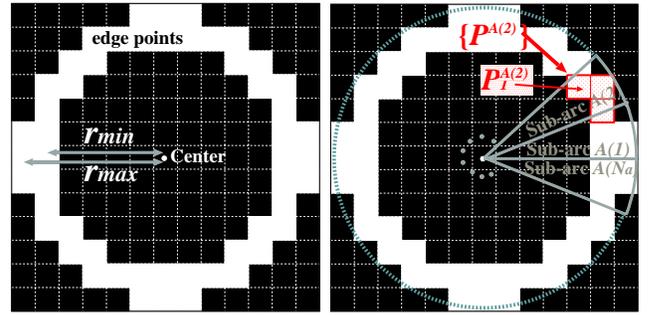
With the above procedures, the system can detect a surface independence of its position and posture when a user holds the input surface.

3.2 Tracking a Fingertip using Arc Detection

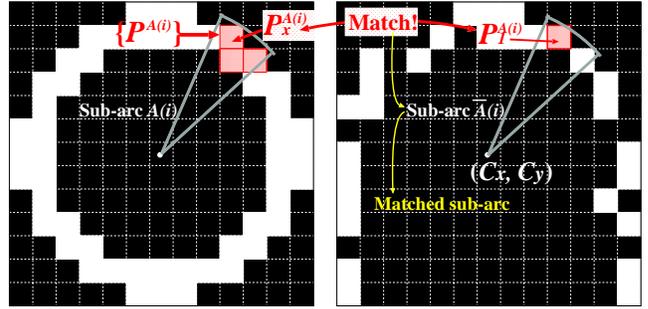
The system searches for the position of a fingertip in the edge image. In [11], the circle-template matching method is proposed for finding fingertips observed in the binarized image; a user's hand is extracted from the observed image. This method works very well when a user's hand is precisely extracted. In an image captured by our active-infrared camera, however, an input surface is observed as well as a user's hand as a close-object and it is difficult to discriminate between the regions of these two objects. This results in difficulty in applying the circle-template matching method proposed in [11] to our WVT system.

A fingertip can be modeled as a semicircle/arc. Most circle/arc detection methods search an edge image for a target (see [12], for example). Fig.7 shows examples of a fingertip in the edge image. From these images, we can see the following problems:

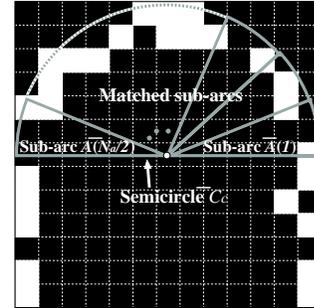
Problem 1 A boundary line is slightly different from part of a true circle, that is, a detected line might be a bent arc.



(1) Lookup-table window (2) Edge point in a sub-arc



Lookup-table window Window W in the edge image
(3) Comparison for matching



(4) Semicircle

Figure 8: Fingertip detection based on arc detection.

Problem 2 The posture and radius of an arc change depending on the position and posture of a fingertip.

Problem 3 An edge line may break due to the failure of edge detection.

The previously proposed methods (e.g., [12]) cope with these problems by iterative computation and robust statistics methods. While these methods can find and detect the precise position and posture of a large target, their computational time is excessive and they are unsuitable for detecting a small target. These characteristics render it unsuitable with our system because (1) for a user to employ the WVT pleasantly, image processing has to finish in real time and 2) the size of a fingertip is quite small in the image (about 10×10 [pixels] in the example shown in Fig.6).

We propose the arc detection method based on analyzing edge lines within sub-arcs. The scheme for detecting a fingertip with the proposed method is as follows:

Step 1 Generate a lookup-table window of fingertip edges in advance. Fig.8 (1) illustrates an example. The edge points in the lookup-table window are inserted between the minimum and maximum radiuses (denoted by r_{min} and r_{max} , respectively) of a fingertip arc observed in an image⁶.

Step 2 Divide the edge points in the lookup-table window into N_a sub-arcs $\{A(1), A(2), \dots, A(N_a)\}$, each of whose apex is the center of the lookup-table window as illustrated in Fig.8 (2). The set of edge points in the sub-arc A_i , where $i \in \{1, \dots, N_a\}$, is described as $\{P_p^{A_i} | p \in \{1, \dots, N^{A_i}\}$, where N^{A_i} denotes the number of edge pixels in the sub-arc A_i . This step is also achieved before using the WVT.

Step 3 Let (C_x, C_y) be the center of the square window W with side r_{max} [pixel] long in the observed edge image. If the number of edge points in W exceeds a predefined threshold, go to **Step 4**.

Step 4 Check whether or not the sub-arc \bar{A}_i in the window W includes the edge point whose coordinates are identical to $P_x^{A_i}$, where $x \in \{1, \dots, N^{A_i}\}$. A sub-arc that satisfies this condition is called a matched sub-arc. Fig.8 (3) illustrates an example.

Step 5 Let the combination of sub-arcs $\{\bar{A}_c, \bar{A}_c + 1, \bar{A}_c + \frac{N_a}{2} - 1\}$ compose a semicircle \bar{C}_c . If all of these sub-arcs are the matched sub-arcs, \bar{C}_c is considered to be a detected semicircle in the observed image as shown in Fig.8 (4).

Step 6 Execute the above Steps 3, 4 and 5 for all the pixels within the region of the input surface.

Step 7 To select a fingertip region from a group of detected semicircles, the circle-template matching method proposed in [11] is employed; (1) the window image of each semicircle is binarized by employing [13] and (2) each binarized window image is compared with the circle-template image and the one with the highest correlation value is selected as a fingertip. The center of the selected semicircle is considered to be the position of the fingertip.

Since our template model of a fingertip can be designed so that it accepts a variety of bent arcs with various sizes, the above problems 1 and 2 are solved. In the Step 5, all pixels in a sub-arc are checked whether or not the observed edge image is matched with the template model. This procedure keeps the high correlation between the observed image and the template model even if the edge line of a fingertip breaks, i.e., in the case of the problem 3.

In addition, for fast processing and stable detection, the above semicircle detection in the Step 6 is restricted by the following rule; we assume that a user points his/her fingertip upward in the observed image while drawing as shown in Figures 2 and 5. As a result, the number of semicircles, each of which can be a candidate of a fingertip, decreases.

Note that when a user holds a surface in his/her hand, its fingertip(s) is observed within the surface. To discriminate between the fingertips holding the plane surface and drawing

⁶ r_{max} and r_{min} are determined based on the actual observed image in advance.

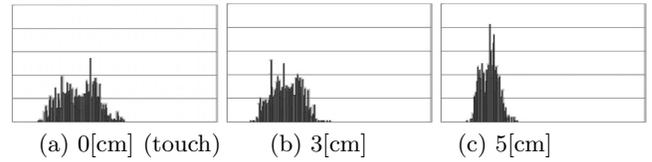


Figure 9: Histograms of gray values around a fingertip in a gradient image (Vertical axis: gray value of each pixel, Horizontal axis: frequency).

an input, the system has to 1) detect the fingertip holding the plane surface before he/she starts drawing and 2) regard it as a non-input fingertip.

With the above scheme, the WVT can detect and track the fingertip of a user in real time.

3.3 Discrimination between Input and Non-input States

As mentioned before, the system can estimate the geometric configuration among observed objects based on depth-dependent information included in an infrared image; gray values in an infrared image are varied in accordance with the distance between the active-infrared camera and an observed object. By employing this property, we estimate whether or not a user touches the input surface with his/her fingertip.

Although gray values in an infrared image include depth-dependent information, they vary not only depending on the distance but also depending on various other factors, for example, their materials and the posture of an input surface. This results in difficulty in investigating the relationship between the distance and gray values at every situation in advance. Accordingly, in our system, a user registers the difference between observed gray values in the case of touching and not touching the plane surface actually used as the input surface before he/she uses the WVT system with this plane surface.

The above registration is practically implemented as follows. To register the variation of gray values around the fingertip in the case of touch, the system observes an input surface while a user moves his/her finger on it. During this operation, the system detects the fingertip and acquires the histogram of gray values in the gradient image, which is the derivative of the local image values and generated by applying the Sobel operator to the original observed image.

Figure9 shows an example of the acquired histogram of gradient values around a fingertip. This histogram is generated from all pixels with the exception of a finger region in a square window whose center is the detected fingertip. For comparison, we also show examples of the gradient-value histograms in the case of non-touch (Fig.9 (b) and (c)), both of which was observed in the same condition, i.e., 1) the geometric configuration (i.e., position and posture configuration) between the camera and the input surface was not changed and 2) the fingertip was projected onto the same position of the image coordinate system. The distributions of gradient values shown in Fig.9 (a), (b) and (c) are different from each other because gray values on and around the boundary of a finger change depending on the following factors:

- A finger's shadow: The shade and size of a finger's shadow vary according to the distance between the

finger and the input surface. The reflectance of an infrared ray is attenuated by the dark shadow.

Segen[14] also proposed the method that estimates the 3D position and posture of a finger by analyzing a finger's shadow. Since the finger and its shadow have to be detected separately in this method, it is impossible to estimate the 3D information of a finger when it touches an object (because its shadow cannot be observed). We, therefore, cannot employ this method for implementing the WVT.

- Distance between finger and input surface regions: The distances from the camera to a finger and to an input surface change the gray values of their regions. In other words, the gray values between their regions represent the distance between them.

We characterize each distribution as a variance of gradient values. In these experimental results, as the distance between the fingertip and the input surface becomes closer, the variance of gradient values becomes greater. In this case, therefore, the system has to consider that a fingertip touches an input surface when the variance of observed gradient values is larger than a predefined threshold. The smallest variance acquired during the above operation is regarded as the threshold. Note that the relationship between the touch and non-touch states may be reversed depending on observed variances in those states, namely the closer the distance between the fingertip and the input surface becomes, the smaller the variance becomes. In this case, the largest variance is considered to be the threshold.

In the former WVT[1], a histogram of gradient values is generated from pixels in a square window as described above. In this square window, however, there may exist several edge lines caused by texture patterns of the input surface. These edge lines seriously corrupt the relationship between the variances observed when a finger touches and is apart from an input surface.

To determine this relationship without being interfered by texture patterns of an input surface, we scan pixels along the boundary edge of a finger in a square window and then calculate the variance of gradient values of these pixels. The pixel values along the boundary edge of a finger are scarcely affected by edge lines on an input surface. As a result, the WVT system can distinguish whether or not a finger touches an input surface even if the surface has texture patterns with complicated edge lines.

The geometric configuration between a camera and an input surface changes while a user moves his/her head and hand. The gray values of an input surface, then, vary depending on this geometric configuration. That is, a constant threshold fails to detect the input state if the input surface moves. To solve this problem, we adjust the threshold depending on the average of the gray values of the input surface region around a fingertip. An example of the relationship between the threshold and the gray values is shown in Fig.10. The horizontal and vertical axes show the average of the gray values and the variance of the gradient values, respectively. This graph was obtained by observing the fingertip on the input surface while the position of the fingertip and the geometric configuration between the camera and the input surface were changed. This procedure is done in the registration mode. Suppose that the smallest variances at each horizontal point (i.e., the lower boundary

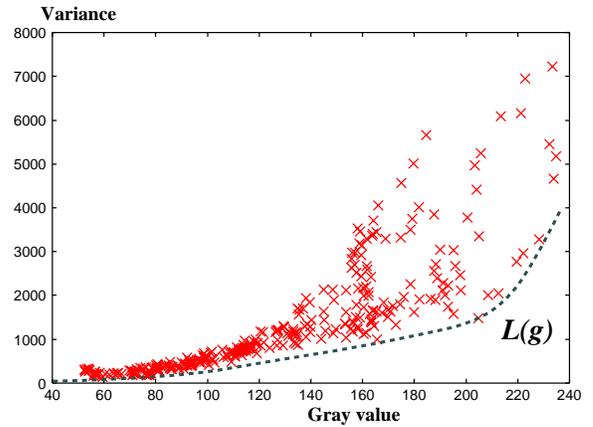


Figure 10: Variable threshold $L(g)$ which are determined depending on the gray value of the observed input surface.

of the observed values) are represented by the function $L(g)$ as illustrated in Fig.10. When the system is in the input mode, the system considers a fingertip as touching an input surface if the variance of observed values along the boundary edge of a finger is above $L(g)$.

3.4 Continuous Superimposed Display of the Input Locus

To correctly transform the fingertip locus between different observed images, we have to estimate the 3D geometric configuration between the camera and the input surface. This procedure, however, requires a complicated reconstruction of 3D information.

In the proposed system, therefore, the approximate transformation is employed: for the transformation between the current image and the image buffer for recording the fingertip locus, the following approximate equation is employed:

$$P(u, v) = (1 - v)((1 - u)P_{00} + uP_{10}) + v((1 - u)P_{01} + uP_{11}), \quad (1)$$

where $0 \leq u \leq 1$, $0 \leq v \leq 1$ and P_{00} , P_{01} , P_{10} and P_{11} denote a 2D point. This equation transforms a 2D point in the square (denoted by B_s) determined by $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$ to a 2D point in an arbitrary tetragon determined by P_{00} , P_{01} , P_{10} and P_{11} . We regard B_s and $(P_{00}, P_{01}, P_{10}, P_{11})$ as the image buffer and the coordinates of the four vertices of the input surface, respectively.

The continuous superimposition of a fingertip locus on the HMD is implemented by the following procedures:

- When the new input result (i.e., the fingertip position on the input surface) is obtained, it is projected to the image buffer by the inverse projection of Equation (1).
- At every captured timing, the locus in the image buffer is re-projected onto the observed image by employing Equation (1).

4. EXPERIMENTS

We conducted experiments to verify that our proposed system works well. Our system consists of a PC (PentiumIII 1.5GHz), a wearable active-infrared camera and a HMD.

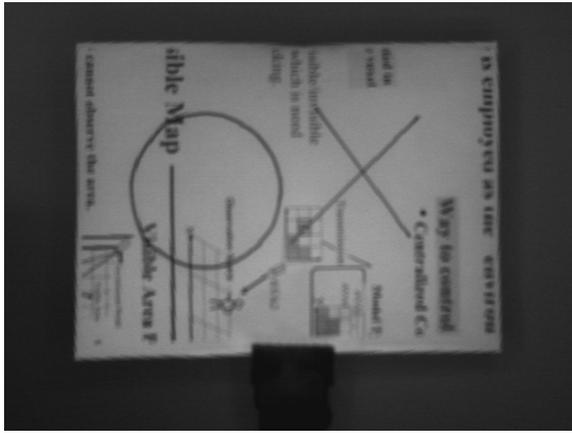


Figure 11: Surface used as an input board.

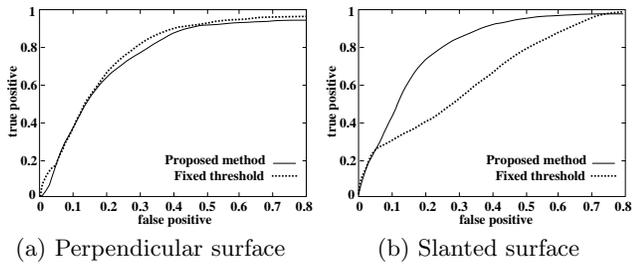


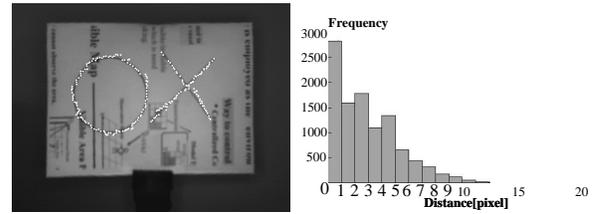
Figure 12: ROC curves in two experiments.

The active-infrared camera system consists of a SONY XC-EI50 with an infrared-pass filter and 24 small infrared-LEDs. The size of a captured image is 640×480 . With these resources, the system could capture and process images at about 0.1[sec] intervals (10[frames/sec]) on average. In all the experiments, we used a sheet of A4-size paper shown in Fig.11 as the input surface. To verify the performance of the proposed system, reference figures (i.e., the circle and the cross) were drawn on the surface.

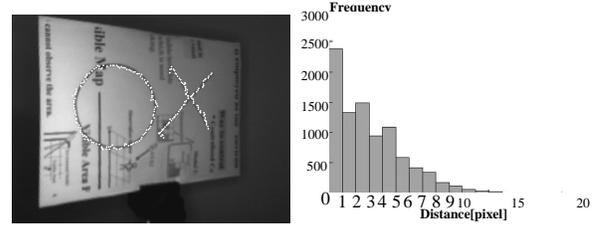
In what follows, we show three experimental results, which shows 1) the correctness of the discrimination between input and non-input states, 2) the input accuracy, and 3) the effectiveness of the superimposed display.

4.1 Correctness of the discrimination between input and non-input states

A user tracked the reference figure on the input surface with the fingertip. We assume that the ground truth of the drawn fingertip-locus was identical to the reference figure. If the distance between the position of the detected fingertip and the reference figure is smaller than 5[pixels] when the system regards the fingertip as touching the input surface, this detected position is considered to be the correct input data. Otherwise, this detected position is considered to be the error input data. The former and latter mean “the system correctly detects the input state when a user touches the input surface” and “the system incorrectly detects the input data when a user does not touch the input surface”, respectively. To verify whether or not the system correctly discriminates between the input and non-input states, we

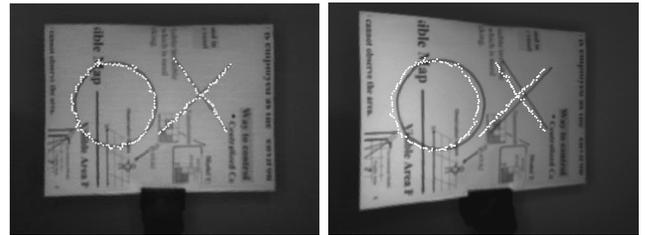


(a) Displayed superimposition (b) Histogram



(c) Displayed superimposition (d) Histogram

Figure 13: Input accuracy. Upper: Perpendicular input surface, Lower: Slanted input surface.



(a) Perpendicular surface (b) Slanted input surface

Figure 14: Superimposition while moving the plane surface.

evaluated the following two rates in the ROC curve:

True positive: The rate of correct inputs.

False positive: The rate of error inputs.

We evaluated these values in the proposed adaptive threshold method (the solid line in Fig.12) and the fixed threshold (the dotted line in Fig.12). Figures 12 (a) and (b) show the results for the surface perpendicular to the optical axis of the camera and the slanted surface, respectively. Especially for input in the slanted surface, the proposed method provided better results. We can, therefore, confirm that the proposed adaptive threshold method is required for a wearable computing environment.

4.2 Accuracy of input data

We verified the distance accuracy of the input data by evaluating the error distance between the reference figure and the drawn locus. The user tracked the reference figure with the fingertip as well as in the above experiment. Figures 13 (a) and (c) show the drawn loci. Figures 13 (b) and (d) show the histograms whose horizontal and vertical axes indicate the error distance and the number of the drawn points. In the perpendicular surface, the average, median and variance values of the error distance were 3.06, 2.83 and

5.64, respectively. In the slanted surface, the results were 3.15, 2.83 and 6.15. In both cases, the rates of the drawn points, whose error distances were within 5[pixel], were over 90%. We consider this accuracy to be enough for practical use.

4.3 Effectiveness of the superimposed display

To verify the effectiveness of the superimposed display, we projected the reference figure from the perpendicular surface (shown in Fig.14 (a)) to the slanted surface (shown in Fig.14 (b)) through the buffer image. The projected figure did not completely coincide with the reference figure on the slanted surface. We could, however, continuously draw in the slanted input surface while visually checking the geometric configuration between the projected figure and the newly input locus.

5. CONCLUDING REMARKS

By employing the properties of an active-infrared camera, we developed the wearable virtual tablet. A user can draw an arbitrary locus on a rectangular surface held by his/her hand. Since the drawn locus is displayed on the HMD and its shape is dynamically adjusted depending on motions of the user and input surface, the user can continuously utilize the WVT in a wearable computing environment.

We mention the following advantages of the WVT.

- The WVT consists of a camera and a HMD, both of which are general in a wearable computing environment.
- Although a rectangular object is required as the input surface, various objects can be used. We, therefore, need not to carry any special input surface.
- We can use the WVT intuitively without any training.

The WVT can be used instead of common input-interfaces as follows.

Keyboard By employing a method for hand-written character recognition, every character (including numerals, alphabets, Chinese characters, and so on) can be inputted. We will confirm the effectiveness of character input (e.g., recognition rate and the max number of characters that a user can input in a single image) with the WVT.

Mouse By superimposing the display image of a PC on the area of the input surface, a user can point at the display image. With this ability, he/she can easily execute several mouse operations such as clicking and dragging. Confirming the effectiveness of these operations will be also included in our future work.

This research is supported by Core Research for Evolutional Science and Technology (CREST) Program “Advanced Media Technology for Everyday Living” of Japan Science and Technology Agency (JST).

6. REFERENCES

- [1] A. Terabe, N. Ukita, Y. Kono, and M. Kidode, “Wearable Virtual Tablet: Fingertip Drawing Interface using an Active-Infrared Camera”, *Proc. of Workshop on Machine Vision Applications 2002*, pp.98–101, 2002.
- [2] M. Fukumoto and Y. Tonomura, “Body Coupled FingerRing: Wireless Wearable Keyboard”, *ACM CHI’97 Proceedings*, pp.147-154, 1997.
- [3] L. T. Cheng, J. Robinson and A. Vardy, “The Wristcam as Input Device”, in *Proc. of International Symposium on Wearable Computing (ISWC 99)*, pp.199–202, 1999.
- [4] H. Sasaki, T. Kuroda, Y. Manabe and K. Chihara, “HIT-Wear: A Menu System Superimposing on a Human Hand for Wearable Computers”, in *Proc. of International Conference on Artificial Reality and Teleexistence (ICAT 99)*, pp.146–153, 1999.
- [5] T. Starner, *et al.*, “The Gesture Pendant: A Self-illuminating, Wearable Infrared Computer Vision System for Home Automation Control and Medical Monitoring”, in *Proc. of International Symposium on Wearable Computers 2000*, pp.87–94, 2000.
- [6] Y. Yamamoto and I. Shiio, “A Simple AR System for Casual Communication”, in *Proc. of Workshop on Interactive Systems and Software (WISS 2000)*, pp.117–124, 2000. (Written in Japanese)
- [7] Y. Muraoka and T. Sonoda, “A Letter Input System of HandWriting Gesture - A User Interface for Wearable Computers -”, in *Proc of Interaction 2001*, pp.3–10, 2001. (Written in Japanese)
- [8] J. K. Hahn, J. L. Sibert and R. W. Lindeman, “Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments”, in *Conference on Human Factors in Computing Systems (CHI 99)*, pp.64–71, 1999.
- [9] H. Koshimizu and M. Numada, “On a Basic Consideration of the Warp Model of Hough Transform”, in *Proc of Machine Vision Application (MVA ’92)*, pp.7–9, 1992.
- [10] R. Y. Tsai: “A efficient and accurate camera calibration technique for 3D machine vision”, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 86)*, pp.364–374, 1986.
- [11] K. Oka, Y. Sato, and H. Koike: “Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems”, in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002)*, pp.429–434, 2002.
- [12] N. Guil and E. L. Zapata: “Lower order circle and ellipse Hough Transform”, *Pattern Recognition*, Vol.30, No.10, pp.1729–1744, 1997.
- [13] N. Otsu: “A Threshold Selection Method from Gray-Level Histograms”, *IEEE Transaction on Systems, Man and Cybernetics*, SMC-9, No.1, pp.62–66, 1979.
- [14] S. Kumar and J. Segen: “Shadow Gestures: 3D Hand Pose Estimation using a Single Camera”, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR99)*, pp.479–485, 1999.