MOTION TRACKING WITH REFLECTIONS
3D pointing device with self-calibrating mirror system

Shinichi Fukushige, Hiromasa Suzuki
Department of Precision Enginieering, The University of Tokyo,7-3-1 Hongo Bunkyo, Tokyo, Japan

Keywords: Interactive pointing device, 3D input, reflections, motion tracking.

Abstract: We propose a system that uses a camera and a mirror to input behaviour of a pointer in 3D space. Using direct and reflection images of the pointer obtained from single directional camera input, the system computes the 3D positions and the normal vector of the mirror simultaneously. Although the system can only input the ‘‘relative positions’’ of the pointer, in terms of 3D locations without scale factor, calibration of the mirror orientation is not needed. Thus, the system presents a very simple and inexpensive way of implementing an interaction device.

1 INTRODUCTION

Input devices for processing 3-dimensional (3D) computer-generated models are divided into two types - those with a 2-dimensional (2D) interface and those with a 3D interface. 2D input devices, such as mice, tablets, and touch monitors, are used more widely than 3D input devices because of their simplicity and easy-use. However, because the target is in 3D space, 2D-base-input equipment needs several constrains and restrictions of pointer movements (Sugishita, 1996) (Zeleznik, 1996) (Branco, 1994). Preconditions, for translating 2D input operations into 3D, often hinder the intuitive input operations of designers.

Therefore, recently, various devices have been developed which can directly indicate the position on 3D space.

Currently, however, 3D input devices are not widely used among general users and are not used as general-purpose tools due to their costs and complexity, requiring special sensors for treating magnetism, ultrasonic waves and laser, or having complex structures, such as joint or wire mechanisms or stereo camera systems (Kenneth, 1994) (Sato, 2000) (Smith, 1995) (Turban, 1992).

Stereovision is commonly used to calculate 3D positioning of a pointer by implementing images from more than one single camera (Faugeras, 1993) (Yonemoto, 2002) (Xu, 1996) (Longuet-Higgins, 1981). However, processing multiple video images in real time thus requires large amount of CPU resources or special hardware. Furthermore, these methods involve synchronization and complex computations that usually require an initial calibration phase. Since multiple cameras must be placed at separated positions to ensure full 3D restoration accuracy, it is difficult to miniaturize such systems.

We would like to provide a simple 3D pointing device that users can handle easily and with a feeling of familiarity. This paper proposes a system for assuming the 3D motion of a pointer in real time by inputting a single video image of the pointer tip with a mirror reflection. Conventionally, in order to determine an object's 3D positioning from a single view, the shape and size of the object or multiple markers on it should be recognized simultaneously. And the restoration accuracy of them are low in the direction of the optical axis.

The proposal method is different from the method of Lane et al (Lane, 2001), which also uses a mirror reflections and estimates the ‘‘absolute’’ 3D positions. This method needs manual calibration and must divide the 3D space into a mirror reflection area and an inputting area.

We propose using a mirror system with self calibration which estimates the relative 3D positions of the pointer. ‘‘Relative positions’’ mean that the restored x, y, z coordinates of the pointer include the same unknown parameter regarded as a scale factor.

However, in the 3D pointing usage, the scale factor can be set freely by a user, because the fine

Fukushige S. and Suzuki H. (2006).
MOTION TRACKING WITH REFLECTIONS - 3D pointing device with self-calibrating mirror system.
In *Proceedings of the First International Conference on Computer Vision Theory and Applications*, pages 428-434
DOI: 10.5220/0001377404280434

motion tracking is more important than inputting absolute positioning value in the real world.

2 PRINCIPLE OF 3D MOTION TRACKING FROM SINGLE DIRECTIONAL IMAGES

Under the principle of estimating the 3D motion of a pointer via a single camera images, we use reflections of a mirror plane. We assume the internal camera parameters such as focal length are pre-calibrated. And normalized camera coordinates are used. The image plane of the normalized camera is in the place of unit length, i.e., 1 from a focal point, i.e., the z axis is taken as direction of the optical axis of a camera, and $z = 1$ is the image plane. Any standard camera may be used, because any standard camera image coordinates can be easily converted for normalized image coordinates. We can thus consider the problems of vision using a normalized camera regardless of individual camera parameters (Xu, 1996).

2.1 Self-calibration of the System

Initially, the proposal system estimate the orientation of the mirror by using the 2D positions of direct and reflected pointer images projected on a camera's image plane (see Figure 1).

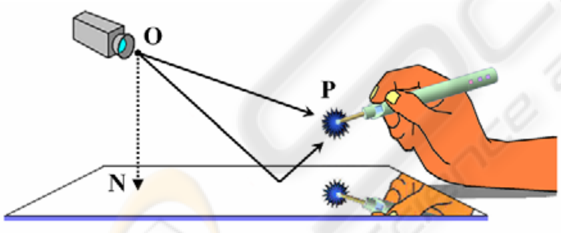

Figure 1: Over view of the proposal system.

The $\mathbf{N} = (N_x, N_y, N_z)$ is the foot point of the camera's focal point $\mathbf{O} = (0,0,0)$ to the mirror plane. And the normal vector of the mirror $\lambda\mathbf{N}$ can be estimated from projected images of the 3D point $\mathbf{P}$ which is the tip of the pointer moving freely in the 3D space. Thus, we can get more than four projected 2D points from more than two 3D points by tracking the movement of $\mathbf{P}$.

Supposing that, at one time, the point is located on $\mathbf{P}_1$, and at other time it is located on $\mathbf{P}_2$ ($\mathbf{P}_1 \neq \mathbf{P}_2$). Then, $\mathbf{m}_1 = (m_{1x}, m_{1y}, 1)$ and $\mathbf{m}_2 = (m_{2x}, m_{2y}, 1)$ are the points projected directly onto the image plane, and $\mathbf{m}_1' = (m_{1x}', m_{1y}', 1)$ and $\mathbf{m}_2' = (m_{2x}', m_{2y}', 1)$ are the points reflected by the mirror and projected onto the image plane from $\mathbf{P}_1$ and $\mathbf{P}_2$ respectively (see Figure 2).

Reflection of light from a mirror is governed by the two Laws of Reflection:
(1) The incident ray, reflected ray and normal at the point of incidence lie on the same plane.
(2) The angle which the incident ray makes with the normal (angle of incidence) is equal to the angle which the reflected ray makes with the normal (angle of reflection).

From the law (1), relation among $\mathbf{m}_1$, $\mathbf{m}_1'$, $\mathbf{m}_2$, $\mathbf{m}_2'$ and $\mathbf{N}$ is written as follows:

$$\alpha_1 \mathbf{m}_1 + \beta_1 \mathbf{m}_1' = \mathbf{N} \quad (1)$$

$$\alpha_2 \mathbf{m}_2 + \beta_2 \mathbf{m}_2' = \mathbf{N} \quad (2)$$

Where, only the $\mathbf{m}_1$, $\mathbf{m}_1'$, $\mathbf{m}_2$ and $\mathbf{m}_2'$ are given value and are on the same plane (z=1). Here the α_1, α_2, β_1 and β_2 are scalar.

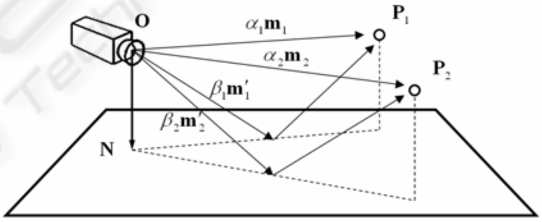

Figure 2: The point P moves from the position P1 to P2.

But, at this time, directly projected points and reflected points have not been distinguished. Using those four 2D points, there can be six straight lines that pass by the every two points and intersect at the three points shown as Figure 3.

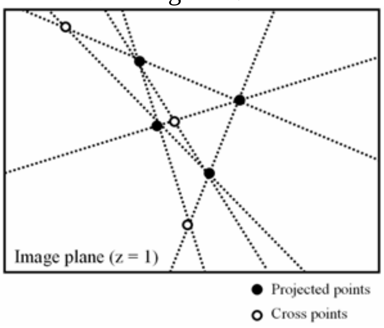

Figure 3: The four projected points and the six straight lines that pass by the every two points.

Two of these six lines are the nodal lines of the image plane and the two planes described as (1) and (2). Thus, in an ideal case, these two lines should always intersect at the same point $\hat{\mathbf{N}} = (N_x/N_z, N_y/N_z, 1)$ which is the intersection point of the image plane and the line extended to $\mathbf{N}$ from the focal point $\mathbf{O}$.

Therefore, one of the three intersection points, whose movement is minimum can be thought as the $\hat{\mathbf{N}}$. And the points near $\hat{\mathbf{N}}$ on the lines is reflection point, and the points far from $\hat{\mathbf{N}}$ can be the directly projected points.

Also $\mathbf{N}$ lies on the intersection line of the two planes (1) and (2). From these formulas, the $\lambda\mathbf{N}$ is determined as follows:

$$\begin{aligned}\lambda N_x &= m_{1x}m_{2x}m'_{2y} - m_{1x}m_{2x}m'_{1y} - m_{1x}m_{2y}m'_{2x} + m_{1x}m'_{1y}m'_{2x} \\ &\quad + m_{1y}m_{2x}m'_{1x} - m_{1y}m'_{1x}m'_{2x} - m_{1x}m'_{1x}r'_{y} + m_{2y}m'_{1x}m'_{2x} \\ \lambda N_y &= m_{1x}m'_{1y}m'_{2y} - m_{1x}m_{2y}m'_{1y} + m_{1y}m_{2x}m'_{2y} + m_{1y}m_{2y}m'_{1x} \\ &\quad - m_{1y}m_{2y}m'_{2x} - m_{1y}m_{1x}m'_{2y} - m_{1x}m'_{1y}m'_{2y} + m_{2y}m'_{1y}m'_{2x} \\ \lambda N_z &= m_{1x}m'_{2y} - m_{1x}m_{2y} + m_{1y}m_{2x} - m_{1y}m'_{2x} \\ &\quad - m_{1x}m'_{1y} + m_{2y}m'_{1x} - m'_{1x}m'_{2y} + m'_{1y}m'_{2x}\end{aligned}$$

The accuracy of the estimated $\lambda\mathbf{N}$ depends on the combination of the four projected points. From the lows of reflection (1), the relation among the four points on the image plane is shown as Figure 4.

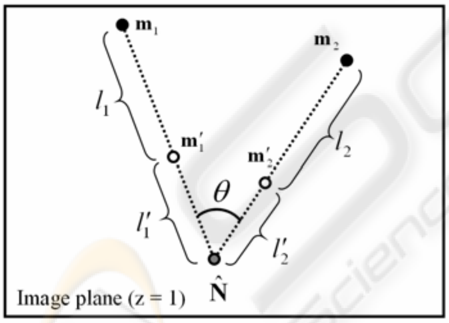

Figure 4: The four projected points on the image plane.

Where, $\hat{\mathbf{N}} = (N_x/N_z, N_y/N_z, 1)$ is the intersection point of the image plane and the line extended to $\mathbf{N}$ from the focal point $\mathbf{O}$. We suppose that the accuracy of the $\lambda\mathbf{N}$ is evaluated with evaluation function E defined by

$$E = \frac{1}{\sin\theta} + \varphi\left(\frac{l'_1}{l_1 + l'_1} + \frac{l'_2}{l_2 + l'_2}\right) \tag{3}$$

Where φ is a constant (7.5 in our experiments), and

$$\sin\theta = \frac{\|(\hat{\mathbf{N}} - \mathbf{m}_1) \times (\hat{\mathbf{N}} - \mathbf{m}_2)\|}{\|\hat{\mathbf{N}} - \mathbf{m}_1\|\|\hat{\mathbf{N}} - \mathbf{m}_2\|}$$

$$l_1 = \|\mathbf{m}_1 - \mathbf{m}'_1\|, \quad l'_1 = \|\mathbf{m}'_1 - \hat{\mathbf{N}}\|$$

$$l_2 = \|\mathbf{m}_2 - \mathbf{m}'_2\|, \quad l'_2 = \|\mathbf{m}'_2 - \hat{\mathbf{N}}\|$$

While calibrating the normal of the mirror, the user ought to move the pointer widely, and the system tracks the four projected points and gets $\mathbf{m}_1$, $\mathbf{m}'_1$, $\mathbf{m}_2$ and $\mathbf{m}'_2$, which minimize E.

2.2 Restoring the 3D Motion of the Pointer From 2D Images

Using the concept of a virtual camera, the camera is set virtually to the opposite side of the mirror from an actual camera. Images reflected by the mirror can be calculated as being shot directly by the virtual camera (see Figure 5). Actually, there is no virtual camera's image plane. However, it can be regarded as overlapping onto the image plane of the actual camera.

We set $\mathbf{C_r}$ as actual camera coordinates and $\mathbf{C_v}$ as virtual camera coordinates.

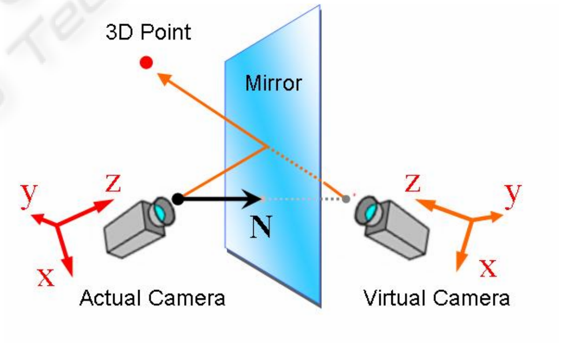

Figure 5: The virtual camera is set to the opposite side of the mirror from the actual camera.

$\mathbf{C_r}$ is field symmetrical with $\mathbf{C_v}$. The relation between two coordinates is thus set as follows:

$$\mathbf{C_r} = \mathbf{R}\mathbf{C_v} + \mathbf{t} \tag{4}$$

$\mathbf{R}$ expresses rotational movement and $\mathbf{t}$ expresses parallel translation movement to $\mathbf{C_r}$ from $\mathbf{C_v}$. Then

$$\mathbf{R} = \mathbf{I} - 2\mathbf{n}\mathbf{n}^T$$

$$= \begin{bmatrix} 1-2n_x^2 & -2n_x n_y & -2n_x n_y \\ -2n_x n_y & 1-2n_y^2 & -2n_y n_x \\ -2n_x n_z & -2n_y n_z & 1-2n_z^2 \end{bmatrix} \qquad (5)$$

$$\mathbf{t} = 2\mathbf{N} \qquad (6)$$

Where, $\mathbf{n} = (n_x, n_y, n_z)$ is the unit normal vector of the mirror, thus $\mathbf{n} = \mathbf{N} / |\mathbf{N}|$.

The 3D position of the pointer $\mathbf{P}$ is calculated with the two points $\mathbf{m} = (m_x, m_y, 1)$, $\mathbf{m}' = (m'_x, m'_y, 1)$ projected onto the image plane. We can consider $\mathbf{m}$ as the point being shot by the actual camera $\mathbf{C_r}$, and can consider $\mathbf{m}'$ as the point of the virtual camera $\mathbf{C_v}$.

The two lines extended toward $\mathbf{m}$ and $\mathbf{m}'$ from each camera's focal point should intersect at the same point $\mathbf{P}$ in an ideal case. From the formula (4), this relation is described:

$$s\mathbf{m} = s'\mathbf{R}\mathbf{m}' + \mathbf{t} \qquad (7)$$

Where, $s\mathbf{m}$ and $s'\mathbf{m}'$ are the 3D positions of $\mathbf{P}$ in the coordinates $\mathbf{C_r}$ and $\mathbf{C_v}$ respectively. When $s'\mathbf{m}'$ is translated to the coordinates $\mathbf{C_r}$, it becomes right side of the formula (7). Although $\mathbf{N}$ is not given, the normal vector of the mirror $\lambda\mathbf{N}$ is predetermined at the self-calibration phase. Then, (7) is changed to

$$s\mathbf{m} = s'\mathbf{R}\mathbf{m}' + \lambda\mathbf{N} \qquad (8)$$

Unfortunately, s and s', which satisfy this formula, may not be found due to errors included in $\lambda\mathbf{N}$ and camera parameters.

This means that the two lines do not always intersect. Then we define the ‘‘intersection point’’ as the centre of the smallest sphere to which both lines are tangential. Consider the case in which the two lines are respectively tangential to the sphere at a node A, $t\mathbf{m}$, and a node B, $t'\mathbf{R}\mathbf{m}' + \lambda\mathbf{N}$ as illustrated in Figure 6.

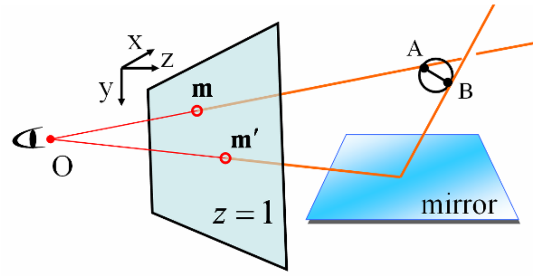

Figure 6: The two lines which tangential to the sphere at a node A and B.

Here the t and t' are scalar.

By defining a unit vector directed from node A to node B as $\mathbf{d}$ and the distance between these two nodes as u, the node B can be determined as

$$t\mathbf{m} + u\mathbf{d} = t'\mathbf{R}\mathbf{m}' + \lambda\mathbf{N} \qquad (9)$$

Since the two lines are perpendicular to line AB, the unit vector $\mathbf{d} = (d_x, d_y, d_z)$ is denoted as

$$\mathbf{d} = \frac{\mathbf{m} \times \mathbf{R}\mathbf{m}'}{\|\mathbf{m} \times \mathbf{R}\mathbf{m}'\|} \qquad (10)$$

Therefore, the three remaining unknowns t, t' and u in (9) can be obtained by solving the matrix expression of:

$$\begin{bmatrix} t \\ u \\ t' \end{bmatrix} = \lambda \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} \begin{bmatrix} m_x & d_x & -(1-2n_x^2)m'_x + 2n_x n_y m'_y + 2n_x n_y m'_z \\ m_y & d_y & 2n_x n_y m'_x - (1-2n_y^2)m'_y + 2n_y n_x m' \\ m_z & d_z & 2n_x n_z m'_x + 2n_y n_z m'_y - (1-2n_z^2)m'_z \end{bmatrix}^{-1} \qquad (11)$$

Finally, the point of intersection, which therefore is assumed to be the location which is the middle point of A and B, can be determined as

$$\mathbf{P} = t\mathbf{m} + \frac{u\mathbf{d}}{2} \qquad (12)$$

The three scalar t, t' and u are all include a remaining unknown λ shown as (11). This can be regarded as a scale factor. We can give free values to λ in the virtual space.

3 EXPERIMENTAL RESULTS

We present the inputting experiments with our proposed system. The internal camera parameters are as follows:

[resolution of camera image] 640 × 480 pixels
[camera focal distance] 3.73
[angle of view] vertical: 31.2°, transverse: 39.6°

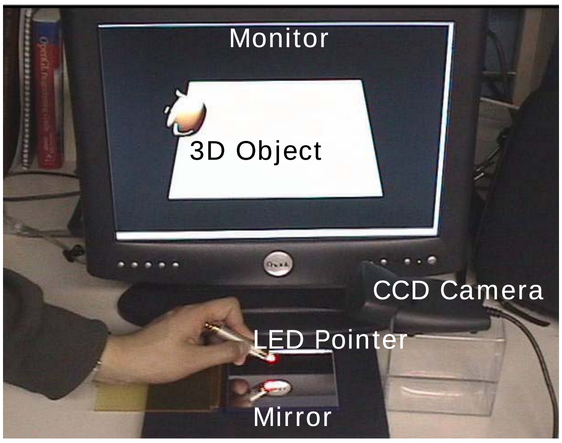

Figure 7: The mirror system with a LED pointer.

We use two types of pointers, i.e., a stylus with a LED at the nib, and the fingertips.

The pointer with the LED uses the properties of a light emitting tool, which include information such as color, lighting, darkness, blinking, etc., of the LED with button operation. Those signals provide variety of operations such as “a click” and “a double click”. A clicking is a single blinking and a double clicking is a continual double blinking within a second.

The gestures of picking and releasing using two fingertips can be recognized by the system, and enable users treat computer-generated objects to be as in the 3D space.

And the system can be used with a big mirror fixed on a wall or a ground and with a camera held by hands, because calibration of the mirror orientation in the camera coordinates is very easy and fast (see Figure 8).

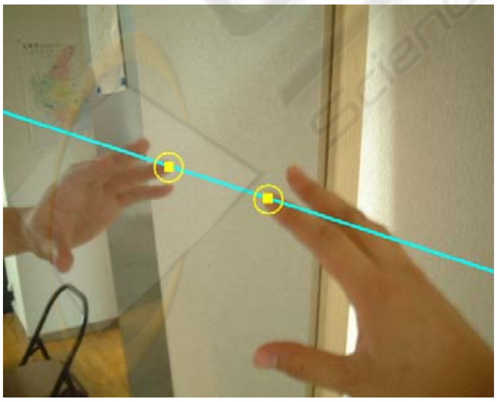

Figure 8: A system with a big mirror fixed on a wall and a hand held camera. (One of the inputting images).

Then, we conduct the input experiment of 3D curves, surfaces, etc (see Figure 9). The proposed method can input fine operation at the tip of a pointer with sufficient balance in all the directions.

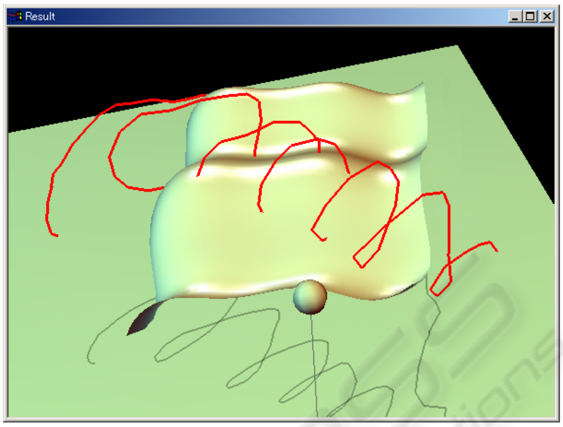

Figure 9: Generating a 3D curve and a 3D surface.

The cost of calculation is comparatively low because only one image obtained from single camera is processed. This enables real time tracking of projected points, restoration of 3D motion and orientation of the mirror by only CPU processing.

3.1 Space Resolution

For 3D pointing devices, the resolution of space in which we input motions reflects more about system performance. The sensitivity to relative movement of pointer operation thus strongly influences the “feel” of use more than absolute positioning accuracy.

We then consider “space resolution” as a criterion showing the ability of how dense the system samples a space.

Space resolution is defined as the minimum distance among the 3D operations of a pointer that the system can recognize.

The resolution in a 2D digital image is, if expanded onto 3D, expressed as a spatial spread of a 4-sided pyramids as shown in (see Figure 10). The 3D resolution is calculated by an intersection area of two 4-sided pyramids that are extended from directly projected image pixels and reflected image pixels. As long as the centre of the pointer tip moves inside the 4-sided pyramids, movement does not appear in the image and it is not recognized by the system.

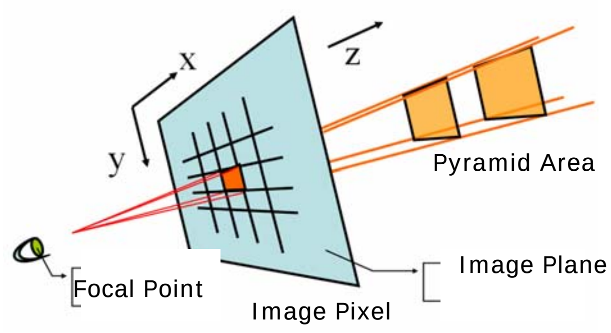

Figure 10: A digital image and an extended pyramid area.

For comparison, we introduce two conventional techniques estimating the 3D position of a pointer via single directional camera input.

1) SPHERICAL MARKER

A spherical marker is at the tip of the pointer, and used for estimating the 3D position from the 2D size of the sphere projected on the image plane.

2) PLURAL MARKERS

Three markers are attached at equal intervals to a pen-like pointer to restore the 3D position of the markers from the 2D position where the three markers are projected to the image plane. The distances between each marker are given.

The resolution of the direction perpendicular to the optical axis (xy direction of an image plane) becomes the same as each of the techniques of spherical and multiple markers. However, the resolution to the optical axis direction is not the same as shown in (see Table 1).

Table 1: Averages of space resolutions.

	X (width)	**Y** (height)	**Z** (depth)
Mirror System	0.13mm	0.13mm	0.31mm
Sphere Marker	0.14mm	0.19mm	0.62mm
Plural Markers	0.13mm	0.20mm	0.48mm

Here, we calculate the average of space resolutions in the area of hemisphere shown as (see Figure 11), and setting conditions of the system are as follows:

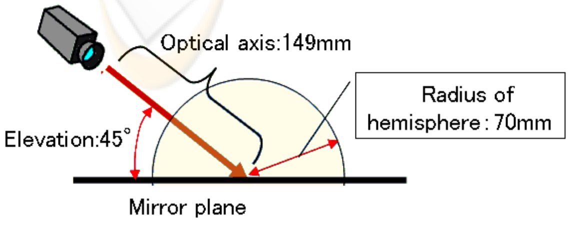

Figure 11: The area of sampling hemisphere.

4 CONCLUSIONS

We have proposed a real-time method for restoring the 3D motion of a pointer by using single directional video input. Conventionally, to obtain an object's 3D position from a single view, the shape and size of an object or plural markers on it had to be recognized simultaneously. Such a method, however, makes restoration accuracy low.

We use mirror images of the pointer to enable us to input fine 3D motion of objects such as a light emitting pointer and fingers. We constructed a simple, compact system as a desktop tool, and the relative locations of the camera and the mirror are self-calibrated. By processing the pointer images, we implemented mouse button functions such as clicking.

The method we have proposed can be constructed using simple, common components such as a camera, a mirror. This makes them applicable in a variety of situations. As shown in (Figure.12), we consider a single desktop tool with a built-in camera.

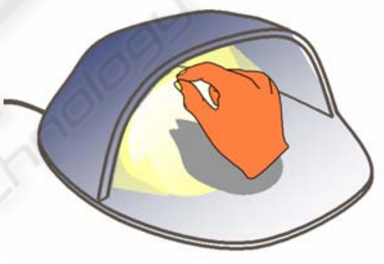

Figure 12: A simplified desktop tool.

The principle of the system can be used for without the stylus tools, recognizing gesture input with 3-dimensional manual operation shown as Figure 13.

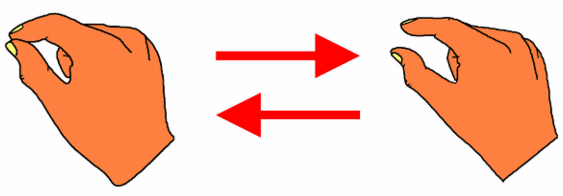

Figure 13: A gesture of picking and releasing.

Our proposal was implemented using gestures such as picking or releasing objects with the fingertip, but more complicated operation is possible using all of the fingers, e.g., for turning or changing a computer-processed object manually.

REFERENCES

Kenneth, J., Massie, T., 1994. The PHANToM haptic interface: A device for probing virtual objects. In *ASME International Mechanical Engineering Expotion and Congress Chicago*.

Sato, M., Koike, Y., 2000. Process division in haptic display system. In *Proceedings Seventh International Conference on Parallel and Distributed Systems.*, pp. 219–224.

Smith, J. R., 1995. Field mice: extracting hand geometry from electric field measurements. In *IBM Systems Journal*, Vol. 35, No. 3&4.

Yonemoto, S., Taniguchi, R., 2002. High-level Human Figure Action Control for Vision-based Real-time Interaction, In *Proc. Asian Conf. on Computer Vision*.

Sugishita, S., Kondo, K., Sato, H., Shimada, S., 1996. Sketch Interpreter for geometric modelling, In *Annals of Numerical Mathematics 3*, pp. 361–372.

Zeleznik, R. C., Herndon, K. P., Hughes, J. F.,1996. SKETCH: An Interface for Sketching3D Scene, In *Proceedings of ACM SIGGRAPH '96*, pp. 163–170, ACM Press.

Branco, V., Costa, A., Ferriera, F. N., 1994. Sketching 3D models with 2D interaction devices, In *Proc. of Eurographics'94*, volume 13, pp. 489–502.

Turban, E., 1992. Expert Systems and Applied Artificial Intelligence, pp. 337-365, Prentice Hall.

Faugeras, O. D., 1993. *Three-dimensional computer vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA.

Xu, G., Zhang, Z., 1996. *Epipoolar Geometry in Stereo, Motion and Object Recognition*. Kluwer Academic Publishers.

Longuet-Higgins, H. C., 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-135.

Lane, J., Lalioti, V., 2001. Interactions with reflections in virtual environments. In *Proc. AFRIGRAPH '01: The 1st international conference on Computer graphics, virtual reality and visualisation,* p87-93.