https://doi.org/10.20965/jaciii.2004.p0140

**Paper:**

# Pointing Device Based on Estimation of Trajectory and Shape of a Human Hand in a Monocular Image Sequence

**Satoru Odo**[*,**], **and Kiyoshi Hoshino**[***]

[*]Faculty of Humanities, Okinawa University
555 Kokuba, Naha, Okinawa 902–8521, Japan
E-mail: odo@okinawa-u.ac.jp
[**]Faculty of Engineering, University of the Ryukyus
1 Senbaru, Nishihara-cho, Nakagami-gun, Okinawa 903–0213, Japan
[***]Institute of Engineering Mechanics and Systems
University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[Received August 15, 2003; accepted December 1, 2003]

**Pointing devices are essential components of graphical user interfaces, and the mouse in particular is widely used because of its intuitive and easy operation. Since it must be directly touched by the user, however, the mouse is restricted in the locations where it can be used. A pointing device consists of a pointing mechanism and a switching mechanism, so the use of a noncontact device to carry out these actions should remove the restriction concerning location. In this study, we investigate the construction of a pointing device that does not impart a feeling of restraint or awkwardness, which estimates the user's hand shape and position from images captured by a monocular camera, a noncontact device. In this system, the captured image is transformed from a Cartesian coordinate system to a log-polar system to reduce image data and computational cost, and achieve real-time operation without using special hardware other than a regular camera. Higher order local autocorrelation features of the log-polar coordinate space were used to achieve robustness against background change and hand rotation. In addition to direct pointing, the ability to recognize gestures from the hand's motion trajectory was incorporated to achieve more comfortable user-computer interaction. In experiments using a system consisting of a regular computer and digital video camera, tracking of the hand and estimation of symbolic signs from extracted frames was stable at a practical average speed of 30 ms per frame.**

**Keywords:** pointing device, monocular camera, log-polar mapping, higher order local autocorrelation, learning vector quantization

## 1. Introduction

Pointing devices such as the mouse, track ball, and tablet are essential components of graphical user interfaces (GUI). The mouse in particular is used widely because of its intuitive and easy operation. Yet issues remain regarding these pointing devices as a person-machine interface for achieving natural communication, namely, that they must be directly touched by the operator and so must lie within reach of the user's hands.

Computer use is ubiquitous, with user needs for interfaces with better operability and natural handling. Pointing devices for use in such environments must not constrain the user spatially. In mobile environments, in particular, it is becoming difficult to install pointing devices based on direct contact by the user's finger or a stylus because of the reduced device surface area that comes with the trend in computer downsizing.

As a pointing device consists of a pointing mechanism and a switching mechanism, using the user's movements to realize these mechanisms should free the device from the limitations above. Several methods have been proposed for pointing devices based on the user's movements, such as the EMG control device[1] which employs electrical signals of the arm muscles' contractions, a leg operating device[2] in which the operator uses both legs to tilt a board, and the "ubi-finger"[3] which combines three sensor systems. There also have been techniques[4, 5] proposed for gesture recognition in which sensors such as data gloves, magnetic motion capture, acceleration sensors, and joint angle detection sensors are placed on the user to measure body movement. Although high-speed, stable processing and high measurement precision can be achieved by direct placement of such sensors, along with the use of dedicated hardware, they tend to give the user a feeling of restraint or awkwardness in handling and are not always comfortable.

Noncontact methods such as position measurement based on phase differences of ultrasonic waves[6] or the use of multiple cameras[7] require the use of special equipment, making them unsuitable for the general user. One study proposes the use of lower-order moment features of the tracking region captured by a regular camera[8], yet the amount of information contained in lower-order moments is limited, namely, target size and spread in different directions, so when the target rotates, the re-

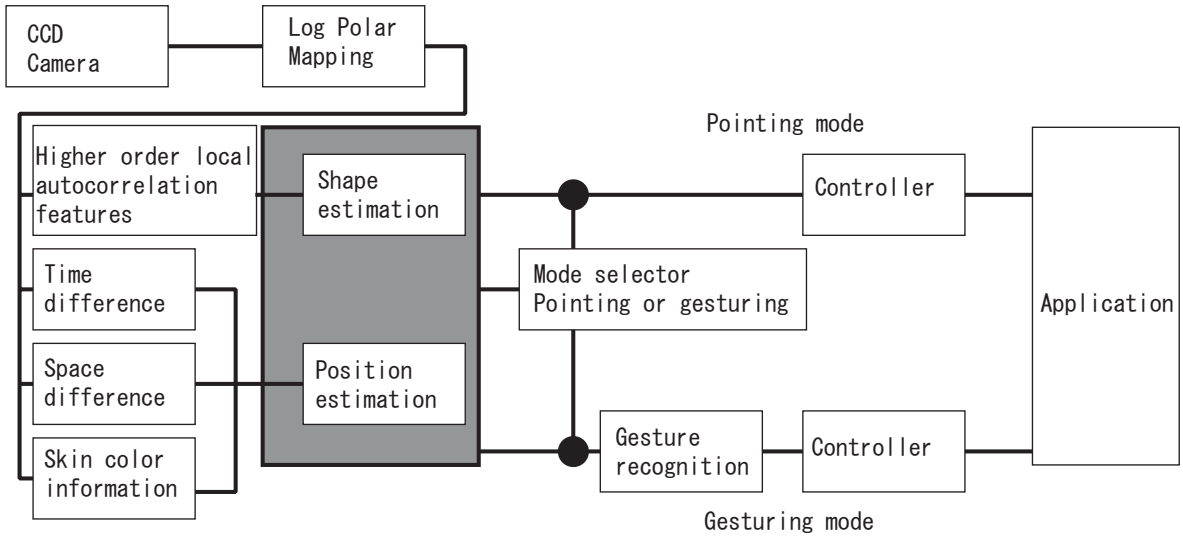


**Fig. 1.** Process flow.

sulting change in the spread direction can lead to recognition failure, and they cannot be used for complex shapes.

Thus, a pointing device that the general user can use comfortably must be noncontact, be able to carry out real-time processing, allow free settings as to its installment or extra hardware, and be sufficiently compact and lightweight, and reasonably priced.

In this study, we study a pointing device that does not impart a feeling of restraint or awkwardness, that estimates the user's hand shape and position from images captured by a monocular camera, a noncontact device.

The captured image is transformed from a Cartesian coordinate system to a log-polar system to reduce image data and computational cost, and achieve real-time operability without using special hardware other than a regular camera. Higher order local autocorrelation features of the log-polar coordinate space were used to achieve robustness against background change and hand rotation. In addition to direct pointing, the ability to recognize gestures from the hand's motion trajectory was incorporated to achieve more comfortable user-computer interaction.

Gesture recognition is used to realize a mouse-like function based on hand-finger movements, specifically as a computer input device. Therefore, there are likely to be fewer erroneous operations when gesture recognition takes place only when the user actually intends to carry out an input operation, instead of having the computer recognize any arbitrary movement. For this reason, we had the user make an "enter" hand shape in front of the camera to turn on/off the mouse-like function.

## 2. System Configuration

### 2.1. Gesture Recognition Algorithm

Each frame of time-series images captured by a stationary monocular camera is transformed into log-polar coordinate images using log-polar mapping (LPM)[9].

The advantages of LPM are that high resolution and a wide working field are obtained using relatively few pixels, while scaling invariance and rotational invariance against the center of transformation are realized. Furthermore, the smaller amount of image data can cut down on the computation time required for image processing. Its shortcoming, however, is unsuitability for dynamic visual processing when uneven sampling causes the image shape to change considerably with translation[10].

Several methods have been proposed to solve this problem. In one, translation is obtained from the optical flow acquired from image sequences in log-polar coordinates when the direction of motion changes periodically with $2\pi$[11]. Another method[12] extracts translation-invariant parameters using an exponential chirp transform[13], which is the equivalent in log-polar coordinates of 2D Fourier transformation. The former method is based on the use of a motion vector, with the image is assumed to move without deformation. If the target object moves a long distance, however, the corresponding image on log-polar coordinates is considerably distorted so an inaccurate motion vector is obtained, preventing correct translation parameters from being obtained. The latter method requires more processing time than conventional transformation and is unsuitable for real-time processing.

In the proposed system, a contour image is generated from the LPM image using time difference, space difference, and skin color information, after which the centroid of the contour image is used to estimate the position of the hand region. The destination of the hand region is estimated from data on past positions. For recognition of the hand shape, higher order local autocorrelation features are computed from the hand region extracted based on skin color information, and then used by a neural network that employs learning vector quantization. This procedure constitutes pointing. The hand region position is

tracked for gesture recognition, so two hand operations — pointing and gesturing — are executed consecutively. The process flow is shown in **Fig.1**. Mode selection between pointing and gesturing is done by the display of a preset hand shape.

Because the translation distance is computed without complex computation such as those required for chirp transform, processing is speeded up. The position is estimated from the centroid, from which detailed information on shape has been omitted, so there is less likelihood of poor tracking precision caused by drastic changes in the image. In addition, color information extracted from the skin color region and the background difference is used to eliminate background objects with similar color information, thus allowing the target object to be extracted properly.

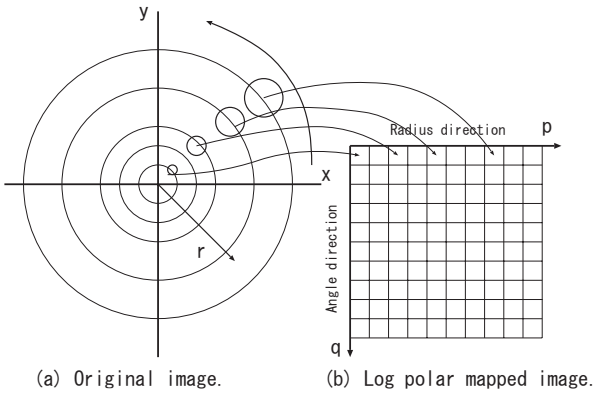


**Fig. 2.** Log-polar mapping from the cartesian plane to the log-polar plane.

### 2.2. Generation of LPM Images

Coordinates $I(x,y)$ of the Cartesian image are assumed to form complex plane $Z$. A point on this complex plane is expressed by $z = x + iy$. Similarly, coordinates $L(p,q)$ of LPM are assumed to form complex plane $W$, on which a point is expressed by $w = p + iq$. LPM is then given by the following expression:

$$w = \log(z + \alpha) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where $\alpha$ is an offset to prevent singularity at the origin.

Original image $I$ is decomposed by LPM into angular and radial components. As shown in **Fig.2**, logarithmic sampling in the radial direction causes information at peripheral areas to be rough compared to the central area. High resolution is maintained in the center, while resolution decreases as the periphery is approached, so overall spatial information is captured roughly. The amount of data and hence processing time are thus drastically reduced. As an example of LPM, the image shown in **Fig.3(a)** is resampled using the points shown in **Fig.3(b)**, which results in **Fig.3(c)**. The inverse mapping results in **Fig.3(d)**.

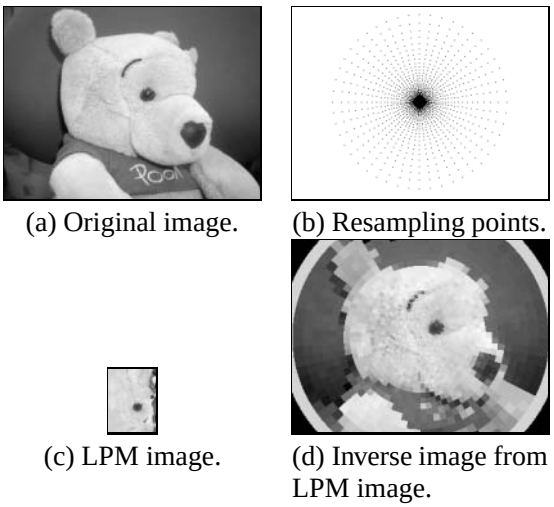


**Fig. 3.** Example of LPM.

The one-to-one correspondence between pixels on the original image and those on the LPM image must be computed to carry out LPM on the input image. While some methods achieve high-speed processing by installing hardware to execute LPM[14, 15], the present system employs software to carry out LPM to not burden the user. Pixel correspondence was determined in preprocessing, since there is no image size change in the present application, and used to generate a lookup table to facilitate mapping. This simplifies mapping and reduces computation time.

### 2.3. Estimation of Hand Region Position

The time difference, space difference, and skin color information are used to extract and track the user's hand. The time difference can be used to separate the background and the movement region easily, since the movement region in the time-series images corresponds to observed changes in luminance over time. Being dependent on the difference in luminance between the background and movement region, results obtained from the time difference are easily affected by changes in room illumination. Thus, edge information obtained from space differences, which are little affected by room illumination, is used in addition, since those areas of the image where great luminance changes occur lie near the edge of movement. Skin color information is used because the present system's purpose is to extract the hand region.

Here, we describe how the hand-finger region is extracted from the input image. Denoting by $L(p,q,t)$ the LPM image generated from the input image at time $t$, the image obtained by taking the time difference of $L(p,q,t)$, denoted by $L_1(p,q,t)$, that obtained by taking the space difference, denoted by $L_2(p,q,t)$, and the skin color region, denoted by $L_3(p,q,t)$, are computed as follows:

Time difference image $L_1(p,q,t)$ is obtained by taking the difference between two consecutive frames, as given in (2). This separates the stationary and moving regions

at time $t$.

$$L_1(p,q,t) = \begin{cases} 1, & |L(p,q,t) - L(p,q,t-1)| \geq th_t \\ 0, & \text{otherwise} \end{cases} \quad . \quad (2)$$

where $th_t$ is the threshold for determining whether there is a change in luminance.

Space difference $L_2(p,q,t)$ is obtained by applying the $3 \times 3$ Sobel filter shown in (3) to the image at time $t$ to extract the edge in the image.

$$L_2(p,q,t) = \begin{cases} 1, & \sqrt{L_{HS}(p,q,t)^2 - L_{VS}(p,q,t)^2} \geq th_s \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $th_s$ is the threshold for determining whether an edge exists, and $L_2(p,q,t)$, $L_{HS}(p,q,t)$, and $L_{VS}(p,q,t)$ are the space difference, the value obtained by applying the $p$-direction Sobel operator, and value obtained by applying the $q$-direction Sobel operator.

The input image is expressed in RGB color mode, where color values are easily affected by brightness due to the high correlation between them. Color information is thus converted in our system to $L^*u^*v^*$ mode (CIE 1976), which has a one-to-one correspondence with the RGB system and is not affected by changes in brightness. Mean $M(\bar{u},\bar{v})$ and variance-covariance matrix $\mathbf{C}$ of the skin color region are then obtained on the $u$-$v$ plane. Skin color region $L_3(p,q,t)$ is thus given by (4). Although this results also in the extraction of skin colored regions present in the background, such as a wall or cardboard box, they are eliminated by logically multiplying the difference image described above since such background regions remain stationary.

$$L_3(p,q,t) = \begin{cases} 1, & (L_c - M)^T \mathbf{C}^{-1} (L_c - M) \geq th_c \\ 0, & \text{otherwise} \end{cases} \quad . \quad . \quad (4)$$

where $th_c$ is the threshold for determining skin color.

From information obtained from eqs.(2) to (4), the contour image is given as follows:

$$L_d(p,q,t) = \begin{cases} 1, & \sum_{i=1}^{3} L_i(p,q,t) = 3 \\ 0, & \text{otherwise} \end{cases} \quad . \quad . \quad . \quad (5)$$

By computing the centroid of contour image $L_d$, position $p_{xy}(t)$ of the hand in the input image $I$ at time $t$ is obtained.

### 2.4. Motion Estimation of Hand Region

Here we discuss tracking for the position of the centroid of the hand region. We denote the centroid position of the hand region at time $t$ by $p_{xy}(t)$, displacement velocity of the hand region by $v(t)$, acceleration by $a(t)$, and the centroid position at time $t$ estimated from that at time $t-1$ by $\hat{p}_{xy}(t)$. Initially at $t = 0$, the centroid position and the estimated centroid position of the hand region are both set at the center of the captured image, with velocity and acceleration both assumed to be zero.

Estimated centroid position $\hat{p}_{xy}(t)$ at time $t$ is given by (6). This then becomes the center of the log-polar coordinate space for mapping the Cartesian image onto an LPM image.

$$\hat{p}_{xy}(t) = p_{xy}(t-1) + v(t-1) \cdot \Delta t \quad . \quad . \quad . \quad . \quad . \quad . \quad (6)$$

where $\Delta t$ is the frame interval.

When centroid position $p_{xy}(t)$ of the actual hand region does not coincide with the estimated centroid position $\hat{p}_{xy}(t)$, we assume that acceleration and displacement velocity $v(t)$ are given by (7) and (8) between the time $t-1$ and $t$.

$$a(t-1) = \frac{2}{(\Delta t)^2} (\hat{p}_{xy}(t) - p_{xy}(t)) \quad . \quad . \quad . \quad . \quad (7)$$

$$v(t) = v(t-1) + a(t-1) \cdot \Delta t \quad . \quad . \quad . \quad . \quad (8)$$

### 2.5. Estimation of Hand Shape

#### 2.5.1. Extraction of Hand Region

When extracting the hand region from LPM image $L$, skin color information is first used to select a region based on (4) and skin color regions are labeled, then the region with the largest area is defined as the hand region. There are cases, however, when shadows may exist when extracting the skin color region from an image, as shown in **Fig.4(a)**, because of the relative positions of the hand and room illumination, which causes the hand region to be incompletely extracted, as shown in **Fig.4(b)**. In the present system, the LPM image is scanned radially outward after extraction of the largest skin-color region to include the entire skin color region as in **Fig.4(c)**. Although some background noise is introduced, this ensures that there are no parts missing from the hand region. After edge enhancement of the image in **Fig.4(c)**, higher order local autocorrelation features are extracted.

#### 2.5.2. Computation of Higher Order Local Autocorrelation Features

Higher order local autocorrelation features are image features[16] proposed by Otsu et al. for image recognition and measurement. Among higher order autocorrelation functions[17], defined by (9), local ones are computed for pixels at the reference point and its vicinity.

$$x^N(a_1, a_2, \ldots, a_N) = \int f(r) f(r+a_1) \ldots f(r+a_N) dr \quad . \quad . \quad . \quad . \quad (9)$$

where $f(r)$ denotes the luminosity of pixels at position $r$, $N$ the order, and $(a_1, a_2, \ldots, a_N)$ the direction of displacement.

Because the correlation between adjacent pixels is considered important when treating natural images, displacement directions are limited to a local region consisting of $3 \times 3$ pixels centered at reference point $r$, and higher order autocorrelation features up to the second order are obtained. Eliminating features that remain equivalent in translation, we obtain the 35 features shown in **Fig.5**,

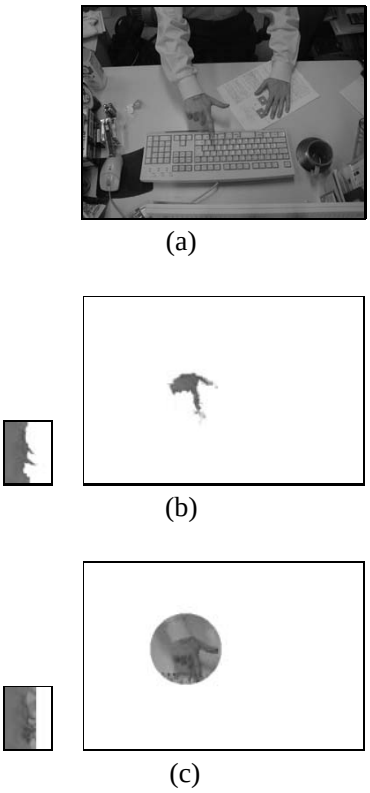

**Fig. 4.** Extraction of hand region.

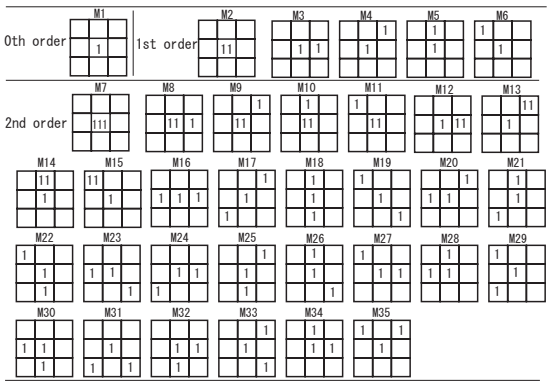


**Fig. 5.** Local patterns to obtain higher order local autocorrelation features.

where '1' represent corresponding pixels in the local pattern. Each feature is computed by adding to all pixels the product of values of the corresponding pixels in the local pattern.

Because higher order local autocorrelation features have the advantage of being translation-invariant, their extraction from the LPM image yields features that are invariant to rotation and scaling.

### 2.5.3. Learning by Learning Vector Quantization

Rather than using quantitative techniques such as multiple regression or principle component analysis for pattern recognition, it suffices to employ qualitative methods of analysis such as cluster analysis or discriminant analysis. Multiple regression analysis, which is a statistical method, often yields unsatisfactory estimation results when dependent and independent variables are related nonlinear by. Statistical methods assume pattern linearity, whereas neural networks are known for treating nonlinear patterns. There are two broad categories of neural networks: hierarchical and competitive. For pattern classification, we use learning vector quantization (LVQ), which is competitive and capable of powerful pattern classification based on a simple algorithm. Being specifically developed to deal with statistical pattern recognition, particularly that involving higher-dimensional probabilistic data with considerable noise, LVQ greatly reduces the amount of computation, compared to conventional statistical methods, and provides nearly optimal recognition accuracy based on Bayes classification rules[19].

The selection of LVQ is based on the following considerations: Hierarchical neural networks have such shortcomings as 1) recognition is treated as a black box, 2) causes of recognition error are difficult to establish, 3) learning requires considerable time, and 4) there is no well established methodology for determining the number of middle level neurons. In contrast, competitive neural networks consist of just two levels — input and output, cluster classification is easily done even when the input has a high dimension, and causal explanations are easily found between input and output. In this study, which presumes a real environment in which the user is expected to discard difficult-to-recognize or confusing patterns and add easy-to-use, readily recognizable patterns, it is preferable to keep training and recognition time short. Although the support vector machine (SVM) was also considered as a candidate due to its high recognition rate, its training requires considerable time so, when it is used as a user interface, the user may be forced to wait while the system is being trained every time a new pattern is added. SVM is a binary classifier and thus unable to optimize classification functions that take into account multiple classes, while no clear method has been established for selecting the kernel trick suitable to the problem at hand. It was therefore decided not to use SVM for this study.

LVQ is an extension of Kohonen's self-organizing map[19] that incorporates supervised learning. After assigning categories to input vectors $x$ and coupling weight vectors $m$, categories are compared between groups. Distances are reduced between input and coupling weight vectors with matching categories, while those not matching are increased. This operation forms a theoretically optimal Bayes classifier boundary. The network configuration is shown in **Fig.6**.

Several LVQ algorithms have been proposed including LVQ1 and its improved versions, LVQ2, LVQ3, and optimized learning rate LVQ1 (OLVQ1)[19, 20]. OLVQ1 is LVQ1 in which coupling weight vectors $m_i$ are assigned learning rates $\alpha_i(t)$. In this study, we use OLVQ1 for its fast learning.

OLVQ1 follows the procedure below.

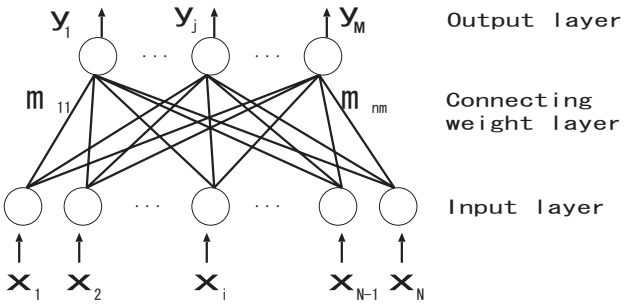


**Fig. 6.** Basic structure of LVQ.

1 The mean of learning vectors belonging to a certain category is given as the initial value of weights between input and output layers.

2 Input vectors $x = (x_1, x_2, x_3, \ldots, x_n)$ are entered into the input layer.

3 In the output layer, distances between weight vector $m_i$ for neuron $i$ and input vectors $x$ are computed as follows:

$$c = \arg\min_i \{||x - m_i||\}. \quad \ldots\ldots\ldots \quad (10)$$

4 $m_i$ with the smallest distance to $x$ is determined to be winning vector $m_c$.

5 Weight vectors are renewed using

$$\begin{aligned} &m_c(t+1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)] \\ &\qquad \text{if } x \text{ is classified correctly,} \\ &m_c(t+1) = m_c(t) - \alpha_c(t)[x(t) - m_c(t)] \\ &\qquad \text{if the classification of } x \text{ is incorrect,} \\ &m_i(t+1) = m_i(t) \qquad \text{for} \quad i \neq c, \end{aligned} \quad (11)$$

where $t$ is time and $\alpha_c$ is the learning coefficient given by

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \quad \ldots\ldots \quad (12)$$

where $s(t)$ is $+1$ when the classification is correct, $-1$ when incorrect.

6 When the learning cycle reaches a set limit, process is terminated. Otherwise, the procedure is repeated from step 2.

In OLVQ1 learning, connecting weights are adjusted so the winning vector approaches learning vectors if it belongs to the correct class, but moves further away otherwise.

### 2.6. Gesture Estimation

For hand-finger gesture recognition, continuous movements of the hand are divided into gesturing and other periods. In one study, spotting recognition is carried out by continuous dynamic programming without specifying the gesturing period[18], while another study uses segmentation for those moments in time when hand-finger motion is minimized[21].

Since gesture recognition is used in the present system as a computer input device, there are likely to be fewer erroneous operations when gesture recognition takes place only when the user actually intends to carry out an input operation, rather than having the computer recognize arbitrary movement.

Therefore, the beginning or ending of gesturing is defined as the point when the user's hand shape matches a preregistered "enter key" gesture when hand movements are minimized, so the interval constitutes the gesturing period. Gestures are then matched by simple dynamic processing. It is normally considered difficult to precisely detect the moment when hand motion is minimized when estimating the gesturing period from a series of images. Our system achieves this by using the hand shape information, i.e., whether it agrees with a pre-registered shape, in addition to detection of minimal hand motion.

The trajectory vector obtained from the hand-finger trajectory tracked during the gesturing period is used as feature vector **S** used in gesture estimation. Denoting by $p_{xy}(t) = (x_t, y_t)$ the hand position at time $t$, feature vector $s(t)$ at time $t$ is given by:

$$\begin{aligned} s(t) &= (v(t), \theta(t)) \quad \ldots\ldots\ldots \quad (13) \\ v(t) &= (x_t - x_{t-1}, y_t - y_{t-1}), \\ \theta(t) &= \cos^{-1}\left(\frac{v(t)\cdot v(t-1)}{|v(t)||v(t-1)|}\right). \end{aligned}$$

Gesture estimation is done by dynamic processing (DP) matching to compute the cost of feature vector $\{\mathbf{S} : s_1(v,\theta), s_2(v,\theta), \ldots, s_i(v,\theta), \ldots, s_N(v,\theta)\}$ and reference feature vector $\{\mathbf{T} : t_1(v,\theta), t_2(v,\theta), \ldots, t_j(v,\theta), \ldots, t_M(v,\theta)\}$ in the dictionary, and then taking the one that minimizes cost.

## 3. Experiment for Evaluation

Recent mouse devices are equipped with multiple operational keys. Thus, in addition to the conventional operations of pointing and the right and left mouse buttons, hand shapes corresponding to those additional operational keys are needed to realize a mouse-like input function. The present system thus uses the 10 hand shapes shown in **Fig.7**. The user matches the hand shapes to the operational keys during actual run time.

In a previous study[22], we discussed the relation between LPM resolution and hand shape recognition rates, reporting that LPM images have a processing advantage over Cartesian images. For comparison, an evaluative experiment was conducted using the same conditions as in

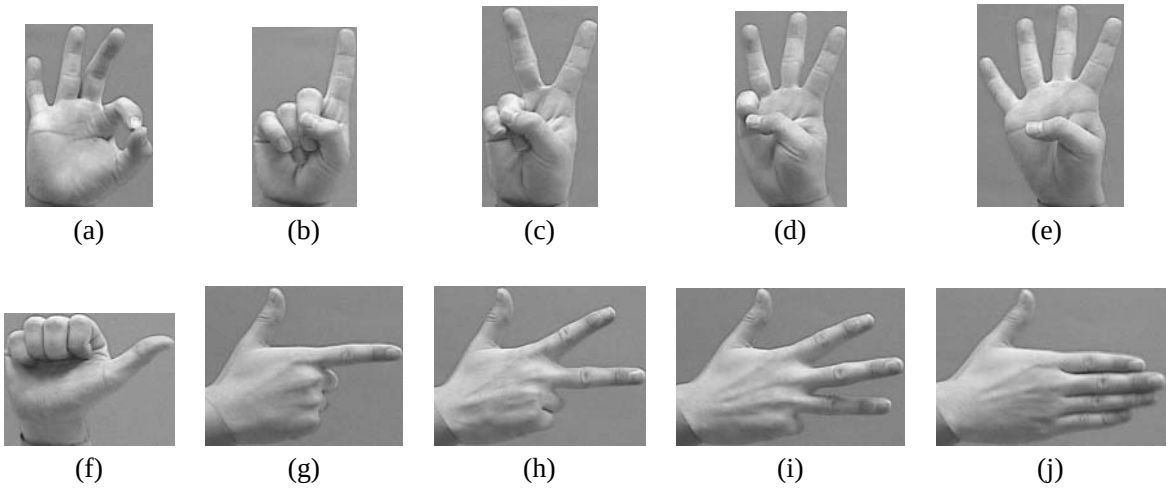


**Fig. 7.** Recognizable class of finger shapes.

the previous study[22]. Since the previous system dealt with four hand shapes, the four shapes (**Fig.7(b), (c), (f)**, and **(j)**) that were also used in the previous system were uesed in the experiment. Compared to the present system, the previous one uesed a different method to extract higher order local autocorrelation features (25 dimensions extracted from the entire LPM image) and a different recognition method (multiple regression).

Based on a resolution of $360 \times 240$ pixels in the original image, we uesed LPM images with resolutions of $120 \times 120$ pixels, $60 \times 60$ pixels, $40 \times 40$ pixels, and $30 \times 30$ pixels, and analyzed the effects.

Using a Sony digital video camera (DVC), images were captured centered at the position of the user's raised hand and at a range so that the hand would fill the scene. Four users were asked to make the 10 hand postures shown in **Fig.7**, but slightly tilted either to the left or right. Captured images were transmitted to the PC at a $360 \times 240$ resolution via IEEE1394, 200 frames were captured per hand shape pattern per user to make a total of 8,000 frames (200 frames $\times$ 4 users $\times$ 10 hand shapes).

The hand region was extracted from each image using a rectangular outer boundary frame and reduced in scale to obtain five differently sized hand images. These were then combined with a monochrome background image so the background center coincides with the centroid of the hand image, thus obtaining a total of 40,000 images (5 hand sizes $\times$ 8,000 frames). Based on a reference size of 100%, which corresponds to the hand size when the user's upper body fills the camera image, hand images of 50%, 75%, 100%, 125%, and 150% were used. **Fig.8** shows an example of the target image.

Hold-out was used for training OLVQ1, where the entire sample set was divided into two subsets, one for training and the other for evaluation. The two sample sets were then interchanged and the average of the results taken to obtain a mutually calibrated recognition rate.

A neural network trained with OLVQ1 was used to obtain recognition rates when the image was combined with the complex background as shown in **Fig.9**.

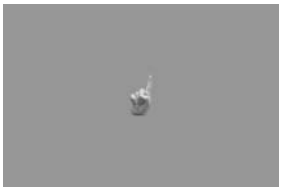

**Fig. 8.** Example of 100%-size hand-finger image used in experiment.

**Fig. 9.** Example of 100%-size hand-finger image combined with complex background used in experiment.

**Tables 1** and **2** show the results of our previous study[22] and the present one, where numbers 1, 2, 3, 4, and 5 for hand size correspond to 50%, 75%, 100%, 125%, and 150% and resolutions represent those of the LPM image.

Measures uesed in the present system to improve the recognition rate include elimination of the background region, which is unnecessary for recognition; the extraction of higher order local autocorrelation features from prebinary edges possessing multivalue information, rather than binary edges; and the use of the nonlinear classification function of OLVQ1, in which categories in feature space are separated nonlinearly instead of multiple regression analysis, which is a linear classification function. As a result, compared to the average recognition rate of 74.5% in

**Table 1.** Recognition rates for various hand sizes when resampling is in log polar coordinates (from [22]).

| Hand size / Resolution | 1 | 2 | 3 | 4 | 5 | Mean | Standard deviation |
|---|---|---|---|---|---|---|---|
| 120 120 | 70.6% | 83.8% | 76.5% | 82.3% | 79.0% | 78.4% | 5.2 |
| 60 60 | 80.4% | 74.1% | 78.7% | 70.9% | 61.0% | 73.0% | 7.7 |
| 40 40 | 82.1% | 79.6% | 59.6% | 67.1% | 66.0% | 70.9% | 9.6 |
| 30 30 | 79.7% | 80.7% | 71.0% | 77.9% | 69.0% | 75.7% | 5.3 |

**Table 2.** Recognition rates for various hand sizes when resampling is in log polar coordinates (Present system).

| Hand size / Resolution | 1 | 2 | 3 | 4 | 5 | Mean | Standard deviation |
|---|---|---|---|---|---|---|---|
| 120 120 | 84.1% | 91.5% | 92.3% | 93.6% | 90.8% | 90.5% | 3.7 |
| 60 60 | 87.3% | 95.5% | 95.4% | 94.1% | 90.3% | 92.5% | 3.6 |
| 40 40 | 88.4% | 95.2% | 94.9% | 93.8% | 92.4% | 92.9% | 2.8 |
| 30 30 | 88.6% | 95.4% | 95.1% | 95.3% | 91.6% | 93.2% | 3.0 |

**Table 3.** Recognition rates for various hand sizes using 10 recognition patterns when resampling is in log polar coordinates.

| Hand size / Resolution | 1 | 2 | 3 | 4 | 5 | Mean | Standard deviation |
|---|---|---|---|---|---|---|---|
| 120 120 | 78.5% | 81.7% | 75.7% | 74.9% | 74.3% | 77.0% | 3.1 |
| 60 60 | 74.5% | 78.1% | 73.9% | 77.0% | 77.4% | 76.2% | 1.9 |
| 40 40 | 72.9% | 79.5% | 76.3% | 77.9% | 80.9% | 77.5% | 3.1 |
| 30 30 | 74.4% | 80.5% | 78.9% | 79.4% | 83.1% | 79.3% | 3.2 |

the previous study, an average recognition rate of 82.7% (an increase of 8.2 percentage points) was attained by isolating the hand region from the background, 86.9% (an increase of 4.2 percentage points) by increasing the higher order local autocorrelation features from 25 to 35 dimensions, and 92.3% (an increase of 5.4 percentage points) by using the OLVQ1 learning algorithm. Combined together, the present system achieves a 17.8 percentage point improvement over the previous system. Standard deviations are smaller, indicating that there is less dispersion in recognition rates caused by variations in hand shape. The results thus demonstrate the validity of the present system.

Results of using the present system to recognize the 10 hand shapes in **Fig.7** are shown in **Table 3**. The average was a satisfactory 77.5% recognition rate.

Although the highest recognition rate is obtained using LPM images of $30 \times 30$ resolution, this resolution can give rise to considerable tracking error, sometimes even causing tracking failure. Since proper pointing cannot take place if tracking failure occurs, $60 \times 60$ is chosen as the setting for LPM image resolution. An experiment to evaluate the tracking performance at $60 \times 60$ resolution, using an animation of a moving sphere with a 15-pixel diameter, yielded a mean error of 6.2 pixels, with a standard deviation of 2.9, and a maximum error of 11.2 pixels[22]. Assuming an allowable maximum error roughly equivalent to the size of the moving object, we can state that tracking performance is satisfactory for regular pointing operations by the user. We then conducted an experiment using a software application that incorporated mouse functions based on the present method. As shown in **Fig.10**, a DVC was positioned to capture the user's hands from above at a distance of 120 cm so the captured image would consist of a rectangular area by 50 cm vertically and 70 cm horizontal.

**Fig. 10.** Experiment setup.

The image captured by the DVC was transmitted via an IEEE1394 interface to a PC (Intel Pentium III 500 MHz). The experiment took place indoors under normal room illumination, with a $360 \times 240$ pixel image size, and 256 hues each for RGB. Prior to the experiment, the four basic operations of pointing, right click, left click, and mode switching were matched with hand shapes, and gestures were registered for the gesturing operation mode.

The results of hand shape recognition were displayed

on the screen in the shape of a mouse cursor, which served to notify the user of the recognition results and enabled corrections easily whenever recognition failure took take. In this experiment, we were able to achieve an average speed of 30 ms per frame, which is satisfactory for practical use.

## 4. Conclusions

In this paper, we proposed a method to estimate hand gestures from input images obtained by a monocular camera, which as a noncontact sensor does not impart to the user a feeling of restraint or awkwardness. The sequential image is transformed from a Cartesian coordinate system to a log-polar coordinate system, and time difference, space difference, and color information are used to extract the hand region. Hand shape is recognized by a neural network in which higher order local autocorrelation features in log-polar coordinate space are learned by OLVQ1. Aimed at realizing a comfortable user-computer interface, the system incorporates a pointing function to achieve direct operation and the ability to recognize symbolic signs from hand motion trajectories. In experiments using a system consisting of a regular computer and DVC, tracking of the hand region and estimation of symbolic signs from extracted frames took place stably at a practical average speed of 30 ms per frame.

**References:**


[1] Toshio Tsuji, Osamu Fukuda, Mitsuru Murakami and Makoto Kaneko, "An EMG controlled pointing device using a neural network," Transactions of the Society of Instrument and Control Engineers, (in Japanese), vol.37, no.5, pp.425–431, 2001.

[2] Yuichiro Kume and Akira Inoue, "Feasibility of feet-operated pointing device," The Journal of the Institute of Image Information and Television Engineers, (in Japanese), vol.54, no.6, pp.871–874, 2000.

[3] Koji Tsukada and Michiaki Yasumura, "Ubi-Finger: Gesture input device for mobile use," IPSJ Journal, (in Japanese), vol.43, no.12, pp.3675–3684, 2002.

[4] Jun'ichi Miyao, "Pedagogy based on sign language word characteristics for a sign language learning system," IEICE Transactions D–I, (in Japanese), vol.J83–D–I, no.10, pp.1120–1128, 2000.

[5] Hideyuki Sawada and Shuji Hashimoto,"Gesture recognition using acceleration sensor and its application for musical performance control," IEICE Transactions A, (in Japanese), vol.79–A, no.2, pp.452–459, 1996.

[6] Hidetoshi Nonaka and Tsutomu Date, "Pointing device using supersonic position measurement," Transactions of the Society of Instrument and Control Engineers, (in Japanese), vol.29, no.7, pp.735–744, 1993.

[7] Hiroki Watanabe, Hitoshi Hongo, Mamoru Yasumoto and Kazuhiko Yamamoto, "Estimation of omni-directional pointing gestures using multiple cameras," The transactions of the institute of electrical engineers of Japan, (in Japanese), vol.121, no.9, pp.1388–1394, 2001.

[8] Ryo Takamatsu and Makoto Sato, "Pointing device based on tracking and recognition of hand with local moments," The transactions of human interface society, (in Japanese), vol.1, no.1, pp.45–52, 1999.

[9] E.L. Schwartz, "Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding," Vision Research, vol.20, no.8, pp.645–668, 1980.

[10] Richard Wallace, Ping-Wen Ong, Ben Bederson, and Eric L. Schwartz, "Space Variant Image Processing," International Juarnal of Computer Vision, vol.13, no.1, pp.71–90, 1994.

[11] Noboru Okajima, Hiroki Nitta and Wataru Mitsuhashi, "Motion Estimation and Target Tracking in The Log-Polar Geometry," Technical Digest of the 17th Sensor Symposium, pp.381–384, 2000.

[12] Tomonori Nonaka and Wataru Mitsuhashi, "Design for a foveal sensor and application for application for pattern recognition," Technical Reports of IEICE, (in Japanese), EID2000–313, pp.77–82, 2001.

[13] Giorgio Bonmassar, and Eric L. Schwartz, "Space-Variant Fourier Analysis: The Exponential Chirp Transform," IEEE Pattern Analysis and Machine Vision, vol.19, no.10, pp.1080–1089, 1997.

[14] Yoshikazu Suematsu and Hironao Yamada, "A wide angle vision sensor with fovea - Design of distortion lens -," Transactions of the Society of Instrument and Control Engineers, (in Japanese), pp.1556–1563, vol.31, no.10, 1995.

[15] Sohta Shimizu, Yoshikazu Suematsu and Shigeto Yahata, "Wide-angle vision sensor with high-distortion lens (Detection of camera location and gaze direction based on the two-parallel-algorithm)," Journal of the Japan Society of Mechanical Engineers, Series C, (in Japanese), pp.4257–4263, vol.63, no.616, 1997.

[16] N. Otsu and T. Kurita, "A new scheme for practical, flexible and inteligent vision systems," Proc.IAPR Workshop on Computer Vision, pp.431–435, 1988.

[17] J.A. Mclaughlin and J. Raviv, "Nth-order autocorrelations in pattern recognition," Information and Control, vol.12, pp.121–142, 1968.

[18] Takuichi Nishimura and Toshiro Mukai, "Adaptation to gesture performers by an on-line teaching system for spotting recognition of gestures from a time-varying image," Systems and Computers in Japan, pp.39–47, vol.31, no.1, 2000.

[19] Teuvo Kohonen, "Self-Organizing Maps," Springer Series in Information Sciences, vol.30, 1995.

[20] Teuvo Kohonen, "Self-Organizatin and Associative Memory," Springer Series in Information Sciences, vol.8, 1984.

[21] Hirohiko Sagawa and Masaru Takeuchi, "A segmentation method of hand gesture for sign language recognition," Proceedings of the human interface symposium '99, (in Japanese), pp.749–754, 1999.

[22] Satoru Odo, Kiyoshi Hoshino, "Hand shape recognitionusing higher order local autocorrelation features in log polar coordinate space," 286–292, Journal of Robotics and Mechatronics, vol.15, no.3, 2003.

**Name:**
Satoru Odo

**Affiliation:**
Doctor Candidate, University of Ryukyus
Lecturer, Okinawa University

**Address:**
555 Kokuba, Naha, Okinawa 902–8521, Japan

**Brief Biographical History:**
1998- Doctor Candidate at University of Ryukyus
2003- Lecturer at Okinawa University

**Main Works:**
• "Hand shape recognition using higher order local autocorrelation features in log polar coordinate space," Journal of Robotics and Mechatronics, vol.15, no.3, pp.286–292, 2003.

**Membership in Learned Societies:**
• The Institute of Electronics, Information and Communication Engineerings (IEICE)
• The Institute of Image Electronics Engineers of Japan (IIEEJ)
• Human Interface Society (HIS)

**Name:**
Kiyoshi Hoshino

**Affiliation:**
Associate Professor, Institute of Engineering Mechanics and Systems, University of Tsukuba

**Address:**
Tsukuba, Ibaraki 305–8573, Japan

**Brief Biographical History:**
1993- Assistant Professor at Tokyo Medical and Dental University
1995- Associate Professor at University of the Ryukyus
1998- Senior Researcher of PRESTO, Japan Science and Technology Corporation (JST)
2002- Project Leader of SORST-JST
2002- Associate Professor at University of Tsukuba

**Main Works:**
• "Interpolation and extrapolation of human 3-D movement for the emergent of dance motions," IEEE Intl. Sympo. Intelligent Signal Processing and Communication Systems, pp.23 8–242 (2002).

**Membership in Learned Societies:**
• The Robotics Society of Japan (RSJ)
• The Institute of Electronics, Information and Communication Engineerings (IEICE)
• Japanese Society for Medical and Biological Engineering (JSMBE)