Sequential image analysis using pulse coupled neural networks for pre-processing

ODO Satoru

Faculty of Humanities, Okinawa University

Abstract We propose a mouse-like function for estimating hand shape from input images with a monocular camera, with which a computer user feels no restraint or awkwardness. Our system involves conversion of sequential images from Cartesian coordinates to log-polar coordinates. Pulse Couple Neural Network (PCNN) is used to extract the hand region, because PCNN has superior segmentation ability. Recognition of the hand shape is carried out by the competitive neural network using higher order local autocorrelation features of log-polar Coordinate space. Mouse-like functions are realized with the hand shape and motion trajectory. Compared to conventional Cartesian coordinates, conversion to log-polar coordinates enables us to reduce image date and computation time, remove the variability by the scaling, and improve antinoise characteristics.

1. INTRODUCTION

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.

Some types of pointing devices are available with contact sensors: such as the EMG control device, a leg operating device, three sensors combined system for the user's movement [1]~[3]. 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.

On the other hand, there is another type of pointing devices using non-contact sensors: ultrasonic waves sensor, multiple cameras system [4], [5]. However, these systems require the use of special equipment, making them unsuitable for the general user.

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 light-weight, 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. 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) [6]. 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 [7]~[9].

In the proposed system, a contour image is generated from the LPM image using pulse-coupled

neural network and skin color information. 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.

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.

2.2 Pulse-Coupled Neural Network

The PCNN is a digital simulation of the cat's visual cortex. It has a lot of good properties for image processing, pattern recoginition [10].

A PCNN neuron contains two main compartments: the Feeding and Linking compartments. Each of these communicates with neighbouring neurons through the synaptic weights **M** and **W** respectively. Each retains its previous state but with a decay factor. Only the Feeding compartment receives the input stimulus, **S**. The values of these two compartments are determind by,

$$F_{ij}(t) = e^{-\alpha_F \Delta t} F_{ij}(t-1) + S_{ij} + VF \sum_{kl} M_{ijkl} Y_{kl}(t-1) \tag{1}$$

$$L_{ij}(t) = e^{-\alpha_L \Delta t} L_{ij}(t-1) + VL \sum_{kl} W_{ijkl} Y_{kl}(t-1) \tag{2}$$

where F_{ij} is the Feeding compartment of the i-th j-th neuron embedded in a 2D array of neurons, and L_{ij} is the corresponding Linking compartment $Y'_{kl}s$ are the outputs of neurons from a previous iteration $(n-1)$. Both compartments have a memory of the previous state, which decays in time by the exponent term. The constants V_F and V_L are normalising constants. If the receptivefields of **M** and **W** change then these constants are used to scale the resultant correlation to prevent saturation.

The state of these two compartments are combined in a second order fashion to create the internal state of the neuron, **U**. The combination is controlled by the linking strength, β. The internal activity is calculated by,

$$U_{ij}(t) = F_{ij}(t)\left(1 + \beta L_{ij}(t)\right) \tag{3}$$

The internal state of the neuron is compared to a dynamic threshold, θ, to produce the output, **Y**, by

$$Y_{ij}(t) = \begin{cases} 1 if U_{ij}(t) > Q_{ij}(t) \\ 0 Otherwise \end{cases} \tag{4}$$

The threshold is dynamic in that when the neuron fires $(Y > \theta)$ the threshold then significantly increases its value. This value then decays until the neuron fires again. This process is described by,

$$\Theta_{ij}(t) = e^{-\alpha_\Theta \Delta t} \Theta_{ij}(t-1) + V Y_{ij}(t) \tag{5}$$

where V_θ is a large constant that is generally more than an order of magnitude greater than the average value of **U**.

2.3 Log Polar Mapped 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:

$$w = \log(z + \alpha) \tag{6}$$

where α is an offset to prevent singularity at the origin.

Original image I is decomposed by LPM into angular and radial components. 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.

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 [11], [12], the present system employs software to carry out LPM to not burden the user.

2.4 Estimation of Hand Shape

2.4.1 Extraction of Hand Region

When using segmentation the method of accumulated response from the PCNN is used. When extracting the hand region from input image, skin color information is first used to select a region 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, because of the relative positions of the hand and room illumination, which causes the hand region to be incompletely extracted.

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. Although some background noise is introduced, this ensures that there are no parts missing from the hand region. After edge enhancement of the image, higher order local autocorrelation features are extracted.

2.4.2 Computation of Higher Order Local Autocorrelation Features

Higher order local autocorrelation features are image features proposed by Otsu et al [13]. for image recognition and measurement. Among higher order autocorrelation functions, defined by Eq.(7),local ones are computed for pixels at the reference point and its vicinity.

$$x^N(a_1, a_2, \ldots, a_N) = \int f(r)f(r + a_1)\ldots f(r + a_N)dr \tag{7}$$

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 3×3 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. 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.

2.4.3 Learning by Learning Vector Quantization

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.

Several LVQ algorithms have been proposed including LVQ1 and its improved versions, LVQ2, LVQ3, and optimized learning rate LVQ1 (OLVQ1) [14]. OLVQ1 is LVQ1 in which coupling weight vectors are assigned learning rates. In this study, we use OLVQ1 for its fast learning. 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.

2.5 Gesture Estimation

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 used in gesture estimation.

3. EXPERIMENT FOR EVALUATION

To realize mouse-like functions, the system must distinguish between "pointing" the left and right mouse buttons. Together with the hand shape used as the "enter" key, the system must be able to recognize at least four classes. For the present system, the four hand shapes are used.

Based on a resolution of 360 × 240 pixels in the original image, we uesed LPM images with resolutions of 120 × 120 pixels, 60 × 60 pixels, 40 × 40 pixels, and 30 × 30 pixels, and analyzed the effects. Using a 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 4 hand postures, but slightly tilted either to the left or right. Captured images were transmitted to the PC at a 360 × 240 resolution via IEEE1394, 200 frames were captured per hand shape pattern per user to make a total of 3,200 frames ($200frames \times 4users \times 4handshapes$). 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 16,000 images ($5handsizes \times 3,200frames$). 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.

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 a result, We obtain an overall average recognition rate of 92.3%. Standard deviations are small, 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 [15].

We then conducted an experiment using a software application that incorporated mouse functions based on the present method. 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.

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 × 240 pixel image size, and 256 hues each for RGB. Prior to the experiment, the four basic perations 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.

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, Pulse Couple Neural Networks 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. Currently, it is a problem that this model takes somewhat a long time.

References

[1] O. Fukuda, J. Arita, and T. Tsuji, 2003 "An EMG-controlled omnidirectional pointing device using a HMM-based neural network," Proceedings of the IEEE International Joint Conference on Neural Networks, pp.3195-3200.

[2] Y. Kume, and A. Inoue, 2000 "Feasibility of feet-operated pointing device," The Journal of the Institute of Image Information and Television Engineers, vol.54, no.6, pp.871-874.

[3] K. Tsukada, and M. Yasumura, 2002 "Ubi-Finger: Gesture Input Device for Mobile Use," Proceedings of APCHI 2002, vol.1, pp.388-400.

[4] H. Nonaka, and T. Date, 1993 "Pointing device using supersonic position measurement," Transactions of the Society of Instrument and Control Engineers, vol.29, no.7, pp.735-744.

[5] H. Watanabe, H. Hongo, M. Yasumoto, and K. Yamamoto, 2001 "Estimation of omni-directional pointing gestures using multiple cameras," The transactions of the institute of electrical engineers of Japan, vol.121, no.9, pp.1388-1394.

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

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

[8] N. Okajima, H. Nitta, and W. Mitsuhashi, 2000 "Motion Estimation and Target Tracking in The Log-Polar Geometry," Technical Digest of the 17th Sensor Symposium, pp.381-384.

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

[10] R. Eckhorn, H.J. Reitboech, M. Arndt, P. Dicke, 1990 "Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Corlez," Neural Comp, vol.2, pp293-307.

[11] Y. Suematsu, and H. Yamada, 1995 "A wide angle vision sensor with fovea - Design of distortion lens -," Transactions of the Society of Instrument and Control Engineers, pp.1556-1563, vol.31, no.10.

[12] S. Shimizu, Y. Suematsu, and S. Yahata, 1997 "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, pp.4257-4263, vol.63, no.616.

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

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

[15] S. Odo, and K. Hoshino, 2004, "Pointing device based on estimation of trajectory and shape of a human hand in a monocular image sequence," Journal of advanced computational intelligence and intelligent informatics, vol.8, no.2, pp.140-149.