# Extraction of Spatial-Temporal Features for Vision-Based Gesture Recognition♦

Huang Yu, Xu Guang-you, Zhu Yuan-xin
Department of Computer Science and Technology,
Tsinghua University, Beijing, 100084, P R China

## Abstract

*One of the key problems in a vision-based gesture recognition system is the extraction of spatial-temporal features of gesturing. In this paper we propose an approach of motion-based segmentation to realize this task. We use the direct method cooperated with the robust M-estimator to estimate the affine parameters of gesturing motion, and based on the dominant motion model we extract the gesturing region, i. e. the dominant object. So the spatial-temporal features of gestures have been extracted. Finally, we directly use the dynamic time warping (DTW) method to perform matching of 12 control gestures (6 for "translation" orders, 6 for "rotation" orders). A small demonstration system has been set up to verify our method, in which we can control a panorama image viewer (set by mosaicing a sequence of standard "Garden" images) with recognized gestures instead of the 3-D mouse tool.*

**Key words:** Gesture recognition, dominant motion model, M-estimator, affine transform model.

## 1. Introduction

Human can develop the ability to interpret gestures, and gestural languages have been developed to allow hearing-impaired people to communicate more easily. Gestures are thus a natural and intuitive form of both interaction and communication. Recognizing this, researchers are developing devices that allow gestures to be used as a form of input for human-computer interaction (HCI). Since the use of the special data glove[1] requires users to wear an encumbering device, so many researchers made efforts on vision-based gesture recognition for naturalness of HCI. There are two types of gesture interaction: communicative gestures as a symbolic language and manipulative gestures to provide multi-dimensional control. However, the latter prevails in the current use for HCI.

Until recently, most of the work in this direction has been focused on the recognition of static hand gestures or postures. In recent year, there has been an interest in incorporating the dynamic characteristics of gestures. The rationale is that hand gestures are dynamic actions and the motion of the hands conveys as much meaning as their posture does[2].

According to our points of view, definition and selection of gesture features is more important than choice of recognition algorithms especially as we consider the state-of-the-art of computer vision techniques. So, it looks clear that gesture segmentation is an open (unsolved) problem in the vision-based gesture recognition. For the dynamic gestures, researchers would resort to methods of motion-based segmentation or spatiotemporal segmentation, which research advances can provide motivation to the stage of gesture analysis. In this paper, our innovations are on extraction of spatiotemporal gesture features in the stage of gesture analysis. Actually a small-scale demonstration system has been set up to verify our method.

In the stage of gesture analysis, we estimate the affine motion parameters of the gesturing hand using the direct method. The direct method is also based on the robust M-estimator, cooperated with

♦ This work was supported by "863" High Technology Research & Development of China and Open Foundation of National Key Lab of Pattern Recognition and Artificial Intelligence of China.

image registration and warping[3]. In the iterative process of M-estimator, ‘edge flow’ estimation like that in[4] can afford the initial guess. We not only exploit the temporal redundancy but also decrease the computation cost using motion prediction[5]. Next, we realize gesture segmentation based on the dominant motion model, a sequential processing approach other than the parallel method for mixture models. In fact, we extract not only the motion parameters but also the gesturing region's (i. e. the dominant object) shape parameters. Besides, the image filtering helps to eliminate some isolated pixels or fill some scrap holes in the segmented region[6]. In the stage of gesture interpretation, a modified dynamic time warping (DTW) method[7] is used for spatiotemporal matching between two dynamic gestures having time variance, in which a distance measure merging the spatial and temporal features is defined.

We also develop a small-scale indoor demonstration prototype for vision-based dynamic gesture recognition. The recognized gestures include 12 types of dynamic gestures for motion control: 6 translations as "left, right, up, down, forward, back", and 6 rotations as " roll, pitch, rotate" counterclockwise or clockwise respectively. The recognized gestures are used to control a panorama image (set by mosaicing a sequence of standard "Garden" images) viewer instead of 3-D mouse.

In the following sections, we discuss related work, then addresses the direct method of motion estimation cooperated with robust M-estimator. After describing the dominant motion model-based segmentation method, we introduce our DTW-based gesture recognition method. Finally we give both the system structure and experimental results.

## 2. Related Work

Existing approaches on this area consist of the 3D model-based and the appearance-based methods, which taxonomy comes from the gesture modeling stage[2]. The 3D model-based approaches, classified in complex volumetric models and simple skeletal models, can result in a wider class of gestures and offer more promise. However, this prospect is hindered by complexity of feature extraction and model parameter estimation[8]. The appearance-based methods appear rich and colorful since the extracted features are omnifarious, such as 2D image sequences directly, image eigenvectors, image moments, deformable templates, fingertips and motion vectors etc. After all, these methods are not satisfactory due to some existing problems: Whether the features reflect both the spatial and temporal characteristic of gestures? Whether the features are extracted easily and robustly?

A view-based method[9] is proposed to exploit the set of view models of gestures using normalized correlation. The object in the subsequent input images is tracked, and when the correlation score drops below a predetermined threshold, a new model view is created with current input image. To compare a new input gesture, each frame of the new sequence is correlated with a model view and its score determined. This method is sensitive to backgrounds, and motion information is not extracted enough.

Quek[4] employs a moving edge detector that accentuates moving edges and suppresses stationary ones. The application of the variance constraint effectively smoothed the field of vectors and aligned local fields while allowing variation across the entire image. These vectors are clustered by spatial location and direction. Its recognition results are not given. The drawback is that information about hand shape is lost.

Cui et. al[10] propose a general framework called SHOSLIF-M to learn and recognize hand gestures from image sequences. First, image sequence representing the event is acquired involving motion detection and motion-based visual attention. The temporal window is mapped to a standard temporal length to form a motion clip. Second, the object of interest is framed using a rectangular window from each image in the sequence and then is mapped to a fixated size fovea image. Third, the spatial-temporal event is recognized from the fovea vector. It is pointed out in the paper that this method assumes the background is uniform since it does not deal with segmentation

directly.

## 3. Motion Estimation

### 3.1 The Direct Method for Motion Estimation

Motion analysis in computer vision commonly uses the motion correspondence-based method or the optic flow-based method. The former one confronts the difficulties as feature (point, line, curve or patch) extraction and matching, especially in the case of occurrence of occlusion and disocclusion. The latter one depends on estimation of optic flow where accurate and dense measurements are difficult to achieve. In general, problems with optical flow are the "aperture problem", motion discontinuity, boundary over-smoothing and multiple moving targets.

Recently, the direct method[11] is broadly used for motion estimation where the 2-D (or 3-D) motion transform (or model) is directly computed from the pixels of the intensity images (luminance signals) without feature extraction or optic flow estimation. Research results show that this method outperforms the above two because it is more robust and resilient to noise. Firstly, the interframe motion is defined as

$$I(\mathbf{p},t) = I(\mathbf{p}-\mathbf{u}(\mathbf{p};\theta),t-1), \qquad (3.1)$$

where $\mathbf{p}=(x,y)$ is coordinate of the image pixel, $\mathbf{u}(\mathbf{p};\theta)$ is the motion vector. Obviously amongst the parametric model of interframe motion is essential, here we use the affine motion model (6 parameters) as

$$\mathbf{u}(\mathbf{p};\theta)=\begin{bmatrix}u(x,y)\\ v(x,y)\end{bmatrix}=\begin{bmatrix}a_0+a_1x+a_2y\\ a_3+a_4x+a_5y\end{bmatrix} \qquad (3.2)$$

where $\theta=(a_0,a_1,a_2,a_3,a_4,a_5)^{\mathrm{T}}$ are the parameters of the affine model. This model is reasonable when the range change in gesturing is small enough compared with the range from the camera.

The direct method of motion estimation can be formulated as the following optimization problem,

$$\min_{(u,v)} E_D = \sum_{(x,y)\in R} \rho(uI_x+vI_y+I_t,\sigma), \qquad (3.3)$$

with $I_x, I_y, I_t$ as partial derivatives of brightness function with respect to *x*, *y* and *t*, only $\rho(r_i;\sigma)$ has different forms, its normal form is the square function.

### 3.2 Robust M-Estimator

Model violations such as these results in measurements that can be viewed in a statistical context as outliers, people usually appeal to the field of robust statistics to solve the problem[11]. Actually, robust regression has been popular in image processing and computer vision, in which the M-estimator (called the generalized maximum likelihood estimator) is amongst more efficient and practical although its breakpoint is not bigger than the other robust regression methods. Here we use the M-estimator to assist motion estimation, especially it appears more efficient when we consider the dominant motion model to be the sequential segmentation approach (described in Section 4.1).

The M-estimator needs to solve a nonlinear minimization problem with an iterative algorithm, we adopt a continuation method, the Simultaneous-Over-Relaxation (SOR) [12]. It can be simply written as

$$a_i^{(n+1)} = a_i^{(n)} - \omega\frac{\partial E_D}{T_{a_i}\partial a_i}, \qquad (3.4)$$

with $a_i$ from (3.2), $\omega=1.995$（$0<\omega<2$）, $T_{a_i}$ as the upper bound of the second-order partial derivatives, i. e.

$$T_{a_i} \geq \frac{\partial^2 E_D}{\partial a_i^2}. \qquad (3.5)$$

with $E_D$ as that in (3.3).

## 4. Gesture Segmentation

For a gesture recognition system there are many aspects evaluated such as accuracy, robustness, speed as well as the variability in the number of

different classes of gestures it covers. Gesture segmentation is considered from two different levels. The higher level indicates segmenting gestures from other unintentional hand movements as well as from each other. The lower level regards segmentation of the gesturing object (hand or arm) from its background and foreground. Our work in gesture segmentation is on the lower level, so we mainly recognize the isolated gestures.

### 4.1 The Dominant Motion Model

By now, we can divide methods of motion segmentation into two sets[10]. One set solves the problem by letting multiple models simultaneously compete for the description of the individual motion measurements[11], and the second one excavates out the multiple models sequentially by solving for a dominant model at each stage[13]. For the former method, its difficulties occur at determination of the number of models or uncertainty of mixture models. The latter one may confront puzzles in the case of absence of dominant motion, and it yet lacks competition amongst the motion models.

The latter sequential segmentation procedure segments a single object and computes its motion parameters. After that, attention is given to other objects, i. e. using the dominant model to another object, and so on. The segmented object on each stage is called the dominant object, and its motion as the dominant motion. From the given assumptions to our system environment in Section 6, this method is more suitable method for gesture segmentation. From the view of robust statistics, the pixels for the dominant object belong to inliers of the dominant model.

### 4.2 Segmentation

Once the motion has been determined, we would like to identify the region having this motion. To simplify the problem, a certain technique is used[3], i. e. the consecutive images are registered by warping using the detected dominant motion. Image registration is defined as a mapping between two images both spatially and with respect to intensity[3]. The registration problem is to find the optimal spatial and intensity transformations so that the two images are matched (Normally the intensity transformation is not necessary). Here we use the estimated affine motion parameters to warp the first image to the second image. In fact, the motion of the corresponding region (dominant object) is cancelled after registration, and the segmenting region is stationary in the registered images. So the segmentation problem reduces therefore to identifying the stationary regions in the registered images.

In our computation procedure, a "coarse-to-fine" strategy[11] is used, which advantages include speeding the search for the optimum displacement estimate, increasing computational efficiency and finding better solutions by avoiding local minima. First we construct a three-level Gaussian pyramid, then begin motion estimation from the lowest spatial resolution. Here the initial guess comes from the "edge flow" computation[4]. The new motion parameters is projected onto the next level of the pyramid (scaled as appropriate), then the image at the current time instant is warped towards the image at the next time instant using the current motion vector. The warped image is then used to compute the change value of motion parameters at this level. The process is repeated until down to the finest level.

Since the performance of above algorithms is limited, the identified region of the dominant (gesturing) object are violated by noise, such as with small holes or spare isolated pixels. Then an image filtering technique[6], i. e. pixel labeling is exploited to alleviate this disturbance. Finally we can extract a single connected region as the dominant object region.

### 4.3 Motion Prediction

If we have identified the stroke phrase of gestures, its motion constancy is obvious, so motion prediction is useful. Not only the information present between the two considered frames, but also the information extracted from the previous frames should be exploited. Here we adopt an adaptive

prediction filter[5] to help motion estimation in the next time. Firstly the prediction filter is represented by

$$a_i^{n+1} = (F_i^{n-1})^T \Phi_i^n \text{, i=0~5.} \quad (4.1)$$

with $\Phi_i^n = [a_i^n, a_i^{n-1}]^T$, $a_i^n$ as affine parameters in the time instant $n$, $F_i^n = [w_{1i}^n, w_{2i}^n]^T$ as the weight vector of the filter. Then we recursively search the desired weights $F_i^n$ as follows

$$\begin{cases} X_i^n = \lambda_i^{-1} P_i^{n-1} \Phi_i^n \\ K_i^n = [1 + (\Phi_i^n)^T X_i^n]^{-1} X_i^n \\ \alpha_i^n = a_i^{n+1} - (\hat{F}_{i1}^{n-1})^T \Phi_i^n \\ \hat{F}_i^n = \hat{F}_i^{n-1} + \alpha_i^n K_i^n \\ P_i^n = \lambda_i^{-1} P_i^{n-1} - K_i^n (X_i^n)^T \end{cases} \quad (4.2)$$

with $\lambda_i$ as a constant less than 1. The initial conditions needed for the above recursive procedure are

$$P_i^0 = \delta_i^{-1} I_2, F_i^0 = \mathbf{0}_2, \quad (4.3)$$

where $\delta_i$ is a small positive constant. So we constrcut a recursive prediction method. The prediction motion can be regarded as the initial guess in the future time.

## 5. Gesture Recognition

### 5.1 Spatiotemporal Features of Gesturing

We calculate the following features from image moments as the shape parameters. The image moments are defined as

$$M_{pq} = \frac{1}{N} \sum_{i=1}^{N} u_i^p v_i^q \quad (5.1)$$

where ($u$, $v$) is the coordinate of each pixel, $N$ is number of pixels in the framed region. Then,

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}, \quad (5.2)$$

$$a = \frac{M_{20}}{M_{00}} - x_c^2, b = 2(\frac{M_{11}}{M_{00}} - x_c y_c), c = \frac{M_{02}}{M_{00}} - y_c^2, \quad (5.3)$$

$$\theta = tg(\frac{b}{2(a-c)})^{-1}, \quad (5.4)$$

$$l_1 = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad (5.5)$$

$$l_2 = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}. \quad (5.6)$$

where $\theta$ reflects the orientation angle of the shape, $(l_1, l_2)$ represents the length and width of one rectangle with the same moments[14].

Now $\upsilon = (\theta, l_1 / l_2, l_1)^T$ is defined as the shape vector. Let $g^l = \{I_1, I_2, ..., I_l\}$ be a gesturing sequence, and $\theta_n$, $\upsilon_n$ be the affine motion vector and shape vector respectively between frame $I_n$ and frame $I_{n+1}$, so an extended vector is constituted as $\mathbf{a}_n = (\theta_n, \upsilon_n)^T$. Now we prepare a sequence of extended vectors $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{l-1}\}$ for the following recognition.

### 5.2 DTW for Gesture Matching

Since a gesture maybe is performed slowly or quickly (similar to speech), so the recognition method must be time instance invariant. The DTW (Dynamic Time Warping)[6] is an efficient technique (regarded as a simplified version of HMM) to match a test pattern with a reference pattern if the time scales are not perfectly aligned. Here we use a modified DTW algorithm from [6].

In the DTW method, we need to define a distance measure for two features. Now let one gesture feature be $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, ...\mathbf{a}_i, ...\mathbf{a}_{l-1}\}$, and the other be $\mathbf{A}' = \{\mathbf{a}'_1, \mathbf{a}'_2, ...\mathbf{a}'_i, ...\mathbf{a}'_{m-1}\}$, normally $m \neq l$. Then a similarity measure of two gestures is defined as

$$d(i,j) = d(\mathbf{a}_i, \mathbf{a}'_j) = \sum_{k=1}^{9} w_k p(\alpha_k - \alpha_k')^2 \quad (5.7)$$

with $\mathbf{a} = (\alpha_1, \alpha_2, ...\alpha_9)^T$, $\mathbf{a}' = (\alpha'_1, \alpha'_2, ...\alpha'_9)^T$, where the former 6 parameters are motion vectors and the latter 3 parameters are shape vectors. Meanwhile $w_k$ are weights for every feature component respectively.

## 6. Experiment Results

### 6.1 System Configuration and Outline

In our demonstration system of gesture recognition, a sequence of monocular images is grabbed on-line, catching the isolated single-handed gesture movements. The hardware configuration includes one Pentium II (266MHz, 64M memory), one image grabber card (Rainbow Runner Card, Matrox Corp.) and one Color Camera (Video Hi8 Pro camera, Sony Corp.). Fig. 6.1 shows the procedure of gesture recognition. The recognized gestures are used as special visual input device of an image-based virtual modeling system, i. e. controlling a panorama image (set by mosaicing a sequence of standard "Garden" images) viewer. Some assumptions must be made in advance:

1) Hand gestures must be aimed at the computer vision's systems (i. e. the viewing camera), which eliminates most of 3D occlusion;

2) Make the viewing camera almost fixed (stationary), then define the gesture 3D space in front of the camera;

3) Distance between the hand and the camera is large enough compared with its depth change, so the motion models are valid.

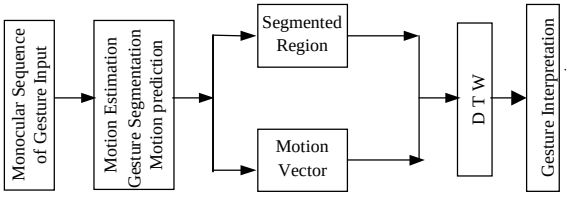



Fig. 6.1 Gesture Recognition outline

The gesture library bears with storing not the original image sequence of gesturing but the extracted discriminative features. We collect multiple samples for each gesture script, then choose one which is the most "similar" to other samples using DTW, where the similarity measure is the same to that in (5.3). The library includes 12 kinds of basic control gestures: 6 for translations as "left, right, up, down, forward, back" and 6 for rotations as "roll, pitch, rotate" with the contrary two directions respectively. Fig. 6.2 shows the procedure of gesture modeling.

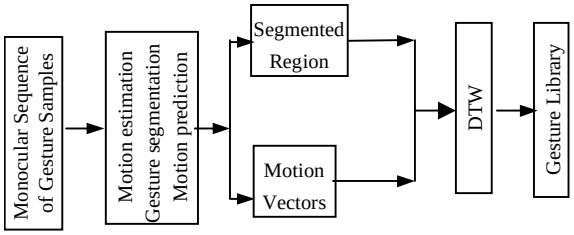



Fig. 6.2 Gesture Modeling outline

The system runs on the Window 95 environment. Here Fig. 6.3 shows the system running interface: the window on the left upper corner shows gesturing intensity images grabbed by a camera on-line, and the right window displays the panorama image viewer controlled by the recognized gesture command. (The “forward, back” control the viewer’s “zoom in, zoom out” process).

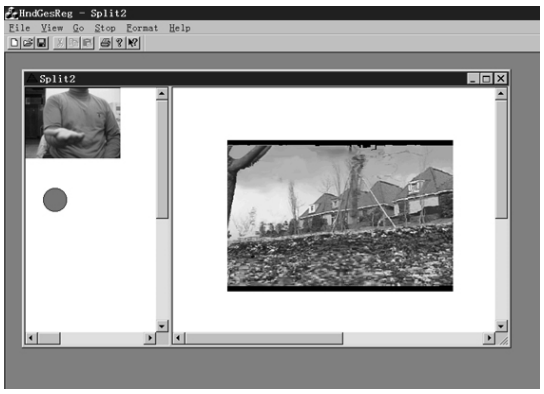



Fig. 6.3 System-running interface on Window 95

### 6.2 Experimental Results

Below Fig. 6.4 is a dynamic gesture sample (the “up” translation command), where (a) and (b) are two consecutive frames, and (c) is the estimation result of dominant motion with robust M-estimator, where the white pixels are inliers and the black pixels are outliers.

(a) (b) (c)

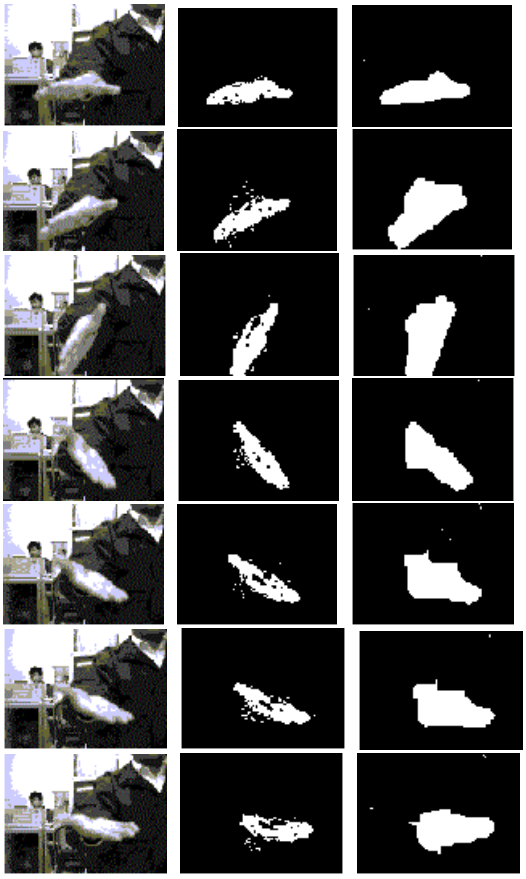
Fig. 6.4 Result of dominant motion estimation

Here Table 6.1 shows the estimation result of affine motion from the “up” gesturing sequence (9 frames).

Table 6.1 Estimation result of affine motion

| | $a_0$ | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ |
|---|---|---|---|---|---|---|
| 1 | -0.3792 | 0.0002 | -0.0010 | -1.6033 | -0.0087 | -0.0184 |
| 2 | -0.4879 | -0.0013 | 0.0041 | -1.7906 | -0.0164 | -0.0064 |
| 3 | 0.2927 | 0.0077 | -0.0057 | -2.1765 | 0.0050 | 0.0016 |
| 4 | -0.4304 | -0.0026 | 0.0016 | -1.4725 | -0.0088 | -0.0296 |
| 5 | 0.1097 | -0.0045 | -0.0051 | -1.5731 | 0.0055 | -0.0032 |
| 6 | 0.0087 | -0.0122 | -0.0132 | -2.5393 | 0.0011 | 0.0133 |
| 7 | -0.1172 | -0.0043 | -0.0126 | -1.2460 | 0.0021 | -0.0089 |
| 8 | -0.3176 | -0.0054 | -0.0076 | -1.2160 | 0.0009 | -0.0154 |

Fig. 6.5 gives a sequence of original frames (a) along with the corresponding extracted dominant moving hand (b) and the filtered regions (c). It records the action of "rotation clockwise" command.



(a) (b) (c)

Fig. 6.5 An example of gesture segmentation

Below Table 6.2 shows the recognition rate on the training set for 12 kinds of gestures. We select 6 persons to do standard gesturing, each person does each gesturing twice. So we collect 144 samples of gesturing altogether. From 12 samples of each gesture we extract one as the standard template and save its shape and motion features in our library. Then we test the recognition performance on the training set, i. e. the 120 original gesturing image sequences for gesturing samples. From Table 6.2, we find that the result for translation gestures is better than that for rotation gestures.

Table 6.2 Recognition rates for 12 kinds of gestures

| Transl. | Up | Down | Left | Right | Forward | Back |
|---|---|---|---|---|---|---|
| Rate | 100% | 100% | 100% | 100% | 100% | 100% |
| Rotat. | Pit_d | Pit_u | Rol_l | Rol_r | Rot_l | Rot_r |
| Rate | 90% | 90% | 90% | 90% | 90% | 90% |

(Notation: here abbreviate "pitch up or down" as "pit_u or pit_d", "roll left or right" as "rol_l or rol_r", and "rotate left or right" as "rot_l or rot_r".)

## 7. Conclusions

In this paper, we mainly discuss how to extract the spatial-temporal features of dynamic gesturing from the monocular frame sequence, viewed by one static camera. Based on the dominant motion model, we propose an efficient method of gesture segmentation in which the direct method cooperated with the robust M-estimator is used for motion estimation. In this paper wee construct an efficient method of representing and extracting the spatiotemporal features of dynamic gestures for vision-based gesture analysis. Compared with these methods in [4,9,10], our method is more efficient for suitable consideration in both the spatial and temporal space. Also we set up a small-scale demonstration system to verify our method, where we use the modified DTW[7] method (it was successful in the isolated speech recognition) to realize the gesture matching with the extracted feature sequences. It is shown the system performance is satisfactory.

## Acknowledgement

We thank Mr. Lin SONG for providing the panorama image viewer, Mr. Zhen WEN for image grabbing and Mr. Haibing REN for computation of image filtering.

## References

[1] Fels S, Hinton G, "Glove-talk -- a neural network interface between a data-glove and a speech synthesizer", IEEE T-NN, 1993, 4(1), pp2-8.

[2] Pavlovic V I et.al, "Visual interpretation of hand gestures for human-computer interaction: a review", IEEE T-PAMI, 1997, 19(7), pp677-695.

[3] Brown L G, "A survey of image registration techniques", ACM Computing Surveys, 1992, 24(4): 325-375.

[4] Quek F., "Eyes in the Interface", Image and Vision Computing, 1995, Vol.13, pp511-525.

[5] Chen L-H, Chang S, "A video tracking system with adaptive predictors", Pattern Recognition, 25(10), 1993.

[6] Rosenfeld A, Digital Picture Processing, Academic Press, New York, 1976.

[7] Haltsonen S, "Improved dynamic time warping methods for discrete utterance recognition", IEEE T-ASSP, 1985, 33(2): 449-450.

[8] Rehg J, Kanade T, "DigitEyes: Vision-based human hand tracking", Technical Report, CMU, 1993.

[9] Darrell T. J., Pentland A. P., "Space-time gestures", IEEE Proc. of ICCVPR, New York, 1993, pp335-340

[10] Cui Y et. al, "Learning-based hand sign recognition using SHOSLIF-M", IEEE Proc. of ICCV95, pp631-636.

[11] Sawhney H. S., Ayer S., "Compact representations of videos through dominant and multiple motion estimation", IEEE T-PAMI, 1996, 18(8), pp814-830.

[12] Black M J, Jepson A D. "Estimation optical flow in segmented images using variable-order parametric models with local deformation". IEEE T-PAMI, 1996, 18(10), pp972-986.

[13] Bergen J R et. al, "A three-frame algorithm for estimating two-component image motion", IEEE T-PAMI, 1992, 14(8), pp886-895.

[14] Horn B K P, Robot Vision, MIT Press, 1986.