# Two-Handed Gesture Tracking Incorporating Template Warping With Static Segmentation

Yu Huang*, Thomas S. Huang*, Heinrich Niemann+
* Beckman Institute, U. of Illinois at Urbana-Champaign, Urbana, IL61801, US
+ Dept. Computer Science, U. of Erlangen-Nuremberg, Erlangen, 91058, Germany
e-mail: {yuhuang, huang}@ifp.uiuc.edu, niemann@informatik.uni-erlangen.de.

**Abstract**

*In this paper we present a template-based method of motion estimation which tracks two-handed-gestures. In our method the gesturing information of temporal motion and spatial luminance is fully utilized. The dominant motion of the detected region corresponding to the tracked object (some hand or the head) is computed. Using this result the object template is warped to yield a prediction template. Incorporated with static segmentation using the watershed algorithm the warped template will be updated through comparison of each sub-region with the prediction template. Tracking results for a set of two-handed command gestures are given to demonstrate its performance.*

## 1. Introduction

Motion estimation and tracking of human body parts is a challenging but critical task for modelling, recognition and interpretation of human behaviors. In recent years there has been an increased interest in trying to understand *hand gestures*, that is, some meaningful or intentional movement of the human arms and hands [5,10,12]. Hand gestures provide a useful interface for humans to interact with others as well as computers. There are two types of gesture interaction: communicative gestures work as a symbolic language and manipulative gestures provide multi-dimensional control. However, the latter prevails in the current use for HCI.

We can divide gestures into static gestures (hand postures) and dynamic gestures. Indeed the hand motion conveys as much meaning as their posture does. For a gesture interpretation system, there are four main components: gesture modeling, gesture analysis, gesture recognition and gesture-based application systems [2,4]. Here we focus on dynamic gesture analysis, i.e. gesture tracking and its motion estimation. Human gestures, especially communicative, naturally employ movements of both hands. In fact two-handed gestures have the stronger expression capability. Our gestures are defined to be some two-handed command gestures, such as “forward“, “backward“, “left“, “right“, “begin” and “stop“ etc.

Normally it is supposed researchers have finished the process of gesture modelling before undergoing gesture analysis. Existing approaches of gesture modelling consist of the 3D model-based and the appearance-based methods. In this paper, we will go the latter way.

### 1.1 Related Work

Some papers on two-handed gesture analysis are addressed below:

Brand et.al [1] also did work on two-handed gesture recognition using a self-calibrating stereo blob tracking system. They propose a Coupled Hidden Markov Models (CHMMs) to model and classify complex gestures.

Cutler and Turk [2] developed a real-time gesture recognition system, in which they mainly estimated optic flow and segmented it into different motion blobs. So the gesture features come from these blobs, such as relative motion and size.

Hong et. al [4] ever realized a gesture learning and recognition system. The center positions of the head and both hands in the image are used as features, which are located by a skin detection and tracking technique.

Yang and Ahuja [12] put forward an apporach for gesture tracking (head and both hands) in an image sequence. They perform statis segmentation and skin detection, then affine motion parameters are estimated after region matching between consecutive frames.

Imagawa et. al [5] construct a sign language recognition system using information from both hands. The defined gesture features include global features, such as hand movement and location, and local features, such as hand shape and orientation. Anyway, the employed method for extraction of those features is only tracking of skin regions and clustering.

Sherrah and Gong [9] show a method of tracking the head and two hands using Bayesian inference from a single 2D view.

Based on a Bayesian Belief Networks (BBNs), they reason about the body parts through fusing other visual cues such as skin color, motion and local intensity orientation with contextual knowledge.

Utsumi and Ohya [10] ever proposed a method to track 3D position, posture and shapes of human hands from multiple-viewpoint images. The constructed system can alleviate the self-occlusion and hand-hand occlusion by employing multiple-view and viewpoint selection mechanism. The gesture features are obtained from hand silhouettes and results of their distance transform.

### 1.2 Overview

In this paper, we propose a template-based method to undergo simultaneously motion estimation and tracking of two hands and the head. This method fully utilizes the information of temporal motion and spatial luminance. Dominant motion of the tracked object (the head, the left or right hand) is calculated by a robust IWLS method. Using the estimated motion parameters the object template is warped to give a prediction template. Results of static segmentation are incorporated to modify the warped template, and the warping errors are utilized to help classification of some doubtful sub-regions around the border of the prediction template.

The next section will describe the general framework of our gesture tracking method for each object (the head, the left or right hand). In Section 3 some experimental results are presented. Conclusions are given in Section 4.

## 2. General Framework of Gesture Tracking

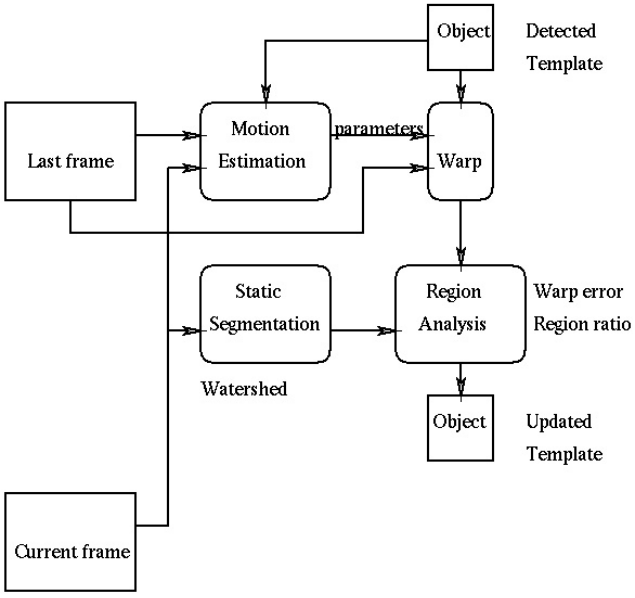



Figure 1 Flow Chart of the Gesture Tracking Algorithm

We assume the head and two hands have been detected in the first frame using some skin color detection technique [12], now the tracking process starts. Though we regard the head and each hand as independent objects, the tracking procedures are the same. The flow chart of our method is illustrated in Figure 1: In Module "Motion Estimation" we calculate dominant motion of the detected region corresponding to the tracked object based on a parametric motion model, this result will be fed into Module "Warp" to register the last frame to the current frame; Meanwhile in Module "Static Segmentation" a watershed algorithm is employed on the current frame to generate many sub-regions (i.e. watershed segments), here different objects are assumed labeled into different regions; In Module "Region Analysis" comparison of each sub-region with the warped template is performed to refine the object template in the current frame, combined with warping error analysis in each sub-region. The details will be explained in the following subsections.

The trend of our framework is comparable to [6]: their method utilized several kinds of edge maps from motion, intensity and prediction to update the object contour, instead we deal with the object template by motion, intensity and region. Recently in [8] a new idea "active blobs" was put forward: In their method the non-rigid deformation was represented in terms of eigenvectors of a finite element method; The photometric variation is still considered by adding new brightness and contrast terms in the optimization; They used a modified Delaunay refinement algorithm to construct a consistent triangular mesh, instead we use the watershed algorithm to generate small watershed segments helpful for region deformation in tracking.

### 2.1 Dominant Motion Estimation

Here we describe the problem as follows: the interframe motion is defined as

$$f(\mathbf{x}, t+1) = f(\mathbf{x} - \mathbf{u}(\mathbf{x};\mathbf{a}), t), \qquad (1)$$

with $f(\mathbf{x}, t)$ as the brightness function in time instant $t$, $\mathbf{x} = (x, y)$ as the coordinate of the image pixel, and $\mathbf{u}(\mathbf{x};\mathbf{a})$ as the motion vector. Without loss of generality, we simply select affine transform as the motion model,

$$\mathbf{u}(\mathbf{x};\mathbf{a}) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ a_3 + a_4 x + a_5 y \end{bmatrix}, \qquad (2)$$

where $\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5)^{\mathrm{T}}$ are the parameters of the affine model. So, dominant motion estimation of the given region $R$ is formulated as the following robust M-estimator,

$$\min_{(u,v)} E_D = \sum_{(x,y)\in R} r(uf_x + vf_y + f_t, s), \qquad (3)$$

here $f_x$, $f_y$, $f_t$ is partial derivatives of brightness function with respect to $x$, $y$ and $t$, and the $r$ - function can be chosen as

the Geman-McClure function as

$$r(x,s) = \frac{x^2}{x^2 + s^2}, \tag{4}$$

with $s$ as the scale parameter.

To solve the problem, there are two different ways to find robustly the motion parameters: one is gradient-based, like the Levenberg-Marquardt method in [8], another is least squares-based, such as the Iterative Weighted Least Squares (IWLS) method. Here we use the IWLS method shown in Equation (5),

$$\begin{bmatrix} \sum wf_x^2 & \sum wxf_x^2 & \sum wyf_x^2 & \sum wf_x f_y & \sum wxf_x f_y & \sum wyf_x f_y \\ \sum wxf_x^2 & \sum wx^2 f_x^2 & \sum wxyf_x^2 & \sum wxf_x f_y & \sum wx^2 f_x f_y & \sum wxyf_x f_y \\ \sum wyf_x^2 & \sum wxyf_x^2 & \sum wy^2 f_x^2 & \sum wyf_x f_y & \sum wxyf_x f_y & \sum wy^2 f_x f_y \\ \sum wf_x f_y & \sum wxf_x f_y & \sum wyf_x f_y & \sum wf_y^2 & \sum wxf_y^2 & \sum wyf_y^2 \\ \sum wxf_x f_y & \sum wx^2 f_x f_y & \sum wxyf_x f_y & \sum wxf_y^2 & \sum wx^2 f_y^2 & \sum wxyf_y^2 \\ \sum wyf_x f_y & \sum wxyf_x f_y & \sum wy^2 f_x f_y & \sum wyf_y^2 & \sum wxyf_y^2 & \sum wy^2 f_y^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} -\sum wf_x f_t \\ -\sum wxf_x f_t \\ -\sum wyf_x f_t \\ -\sum wf_t f_y \\ -\sum wxf_t f_y \\ -\sum wyf_t f_y \end{bmatrix}, \tag{5}$$

where $w(r) = y(r)/r$, with derivative $y(r) = dr(r)/dr$ and error $r = uf_x + vf_y + f_t$. The estimate of $s$ is given by a robust measure as

$$s = 1.4826 median_i |r_i|. \tag{6}$$

The algorithm begins by constructing the Gaussian pyramid (we set up three levels). At the coarse level motion is initially set to zero. The number of iterations is chosen as 10. When the estimated parameters are interpolated into the next level, these parameters are used to warp (realized by bilinear interpolation) the last frame to the current frame. In the current level only the change in the paramenters are estimated in the iterative update scheme.

## 2.2 Static Segmentation by the Watershed Algorithm

In static segmentation, the watershed algorithm of mathematical morphology is a powerful method. It regards the gradient magnitude image as a landscape where the brightness values correspond to the elevation. Areas where raindrops would drain to the same minimum are denoted as catchment basins, and the lines separating adjacent catchment basins are called dividing lines or watersheds.

Early watershed algorithms are developed to process digital elevation models and are based on local neighborhood operations on square grids. Some approaches use ``immersion simulations`` to identify watershed segments by flooding the image with water starting at intensity minima [11]. Improved gradient following methods are devised to overcome plateaus and square pixel grids [3]. Here we use the former method for segmentation of the current frame.

A severe drawback to the computation of watershed algorithm is over-segmentation. Normally watershed merging is needed when this algorithm is performed. But here over-segmentation is welcome, so during tracking we omit the merging process, which saves some computation costs.

## 2.3 Template Warping and Updating

When the motion parameters have been computed by the algorithm shown in Section 2.1, we warp the detected region (or template) of the tracked object from the last frame to the current frame. Then the warped template is used to determine which watershed segments enter the template according to the following measure: Given that the number of pixels belonging to the warped template in the subregion (watershed segment) $R_i$ is $Cp_i$ and the number of all pixels in $R_i$ is $C_i$, a ratio $r_i$ is computed as

$$r_i = Cp_i / C_i. \tag{7}$$

Based on this measure, we discuss further the classfication problem of each subregion in these following situations:

1) When $r_i \geq$ r0 (in this paper r0 = 0.9), we classify $R_i$ as part of the final object template;

2) When r0 > $r_i \geq$ r1 (here r1 = 0.4), another measure as MAE (Mean Absolute Error) of difference between the warped frame and the current frame is taken into account,

$$M_i = \sum_{\mathbf{x} \in R_i} \left| f(\mathbf{x}, t+1) - f^w(\mathbf{x}, t) \right| / C_i. \tag{8}$$

where $f^w(\mathbf{x}, t)$ is the warped image of $f(\mathbf{x}, t)$ using the estimated dominant motion parameters; If the warped error $M_i$ of $R_i$ is smaller enough (less than a given threshold, for instance, 10), $R_i$ is still regarded as part of the updated template; Otherwise, we exclude $R_i$ out of the object region.

3) When $r_i$ < r1, $R_i$ will NOT be included in the updated template.

Figure 2 give an illustration to this process: 2(a) and 2(b) are a pair of consecutive images in the sequence, the region in red is the detected object in the last frame, the region in pink is the real object in the current frame. For sake of simplicity, we assume the detected object is quivalent to the real object. 2(c) is the warped object using the estimated motion parameters, and 2(d) is static segmentation of the current frame (the watersheds drawn with blue color). In 2(e) the warped template (enclosed with green lines), watershed segments and the real object are superimposed in order to illustrate clearly the comparison operation. The subregion enclosed with red lines belongs to the first case, the

subregion with yellow lines belongs to the second case, and the subregion with brown lines should belong to the third case. Finally the updated template of the object in the current frame is shown in 2(f).

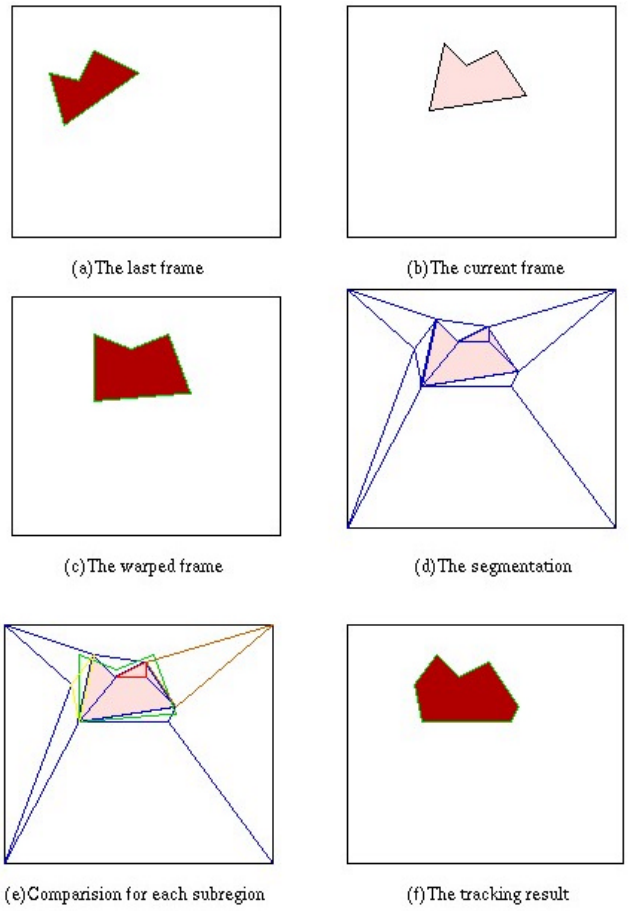



Figure 2 Illustration for the Procedure of Object Tracking

In summary, the process of template warping gives a prediction to the object new position, the comparison of each subregion with the warped template can refine the prediction. In our experiments, it is found the warping error analysis is efficient to avoid some misclassification of small regions near the tracked object in the cluttered background. A Kalman filter could be considered to smooth the estimation of motion parameters based on a simple kinematic model [10].

## 2.4 Two-Handed Gesture Tracking

Normally the template-based method has the capability to handle *small* partial hand-hand occlusions and self-occlusions, which usually appear in two-handed gestures. But, some assumptions are still needed in advance because of limitations of the computer vision algorithms:

1) Two hands must be aimed at the viewing camera, which eliminates heavy occlusions; both hands should avoid occlusions with each other and with the head.

2) The heavy deformation of the hands, like the opening and closing actions of the palm, will make our method invalid; So we ask the state of the palm (either opening or closing) unchanged during tracking;

3) Distance between the hand and the camera is large enough compared with its depth change, so the affine model is able to grasp the motion of hands and the head.

We define a small set of two-handed command gestures, including gestures like “forward“, “backward“, “left“, “right“, “begin” and “stop“ etc., illustrated in Figure 3. The initial posture is the same shown in 3(a), and the final postures of those gestures are illustrated respectively in 3(b)-3(g).

(a) Initial posture

(b) Left

(c) Right

(d) Backward

(e) Forward

(f) Start

(g) Stop

Figure 3 Defined two-handed gestures

It is possible both hands move too close to the head during the gesturing process. In fact, the template-based tracking method possesses certain ability to handle this case even with *small* occlusions. Here we assume this *slight* occlusion only happens in a short period of time (or in a few frames). To deal with heavy occlusion, a multiple-viewpoints scheme like [10] has to be assorted.

In order to show the tracking results clearly, we use an ellipse [7] to approximate the tracked object region in all the experiments

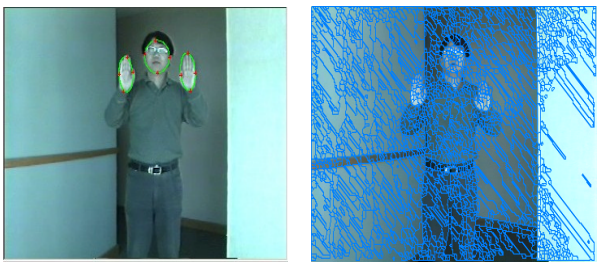
Anyway, those shape parameters of both hands can be also regarded as spatial patterns of postures for a gesture at some time instant, meanwhile the centroid of each ellipse will reflect the two-handed gesture trajectories. All this data will be useful for two-handed gesture recognition.

### 2.5 Some Notations

Yang's method [12] is similar to our procedure, but the difference lies in that during tracking we don't consider region correspondence between consecutive frames (which is not solved robustly in computer vision) and skin color detection. And, what Yang's method didn't utilize is information from template warping and sub-region analysis.

## 3. Experiment Results

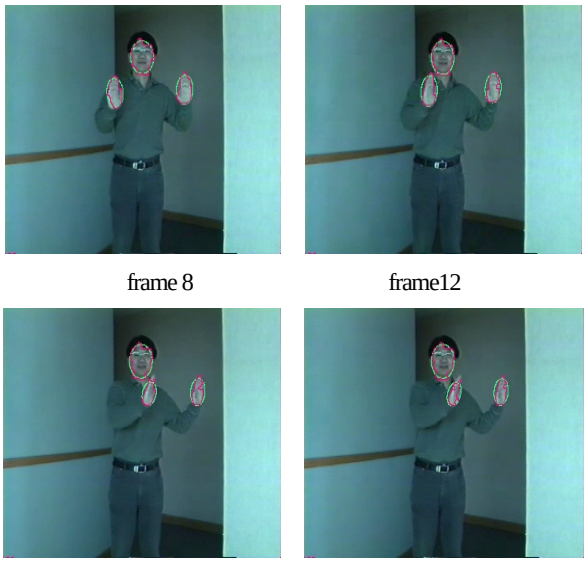
We realize this approach in Visual C++ in Pentium II 400M. Now the processing speed is about 3 seconds per frame if the image size is 350x240. At the initialization, we manually put an ellipse on each hand and the head respectively. Figure 4 gives the initial ellipse fitting manually and its over-segmentation using the watershed algorithm. Those small watershed segments will contribute to template update after the warping operation. The tracking procedure then starts from the marked ellipse region.



(a) ellipse fitting (in green) (b) oversegmentation (in blue)

Figure 4 Tracking initialization

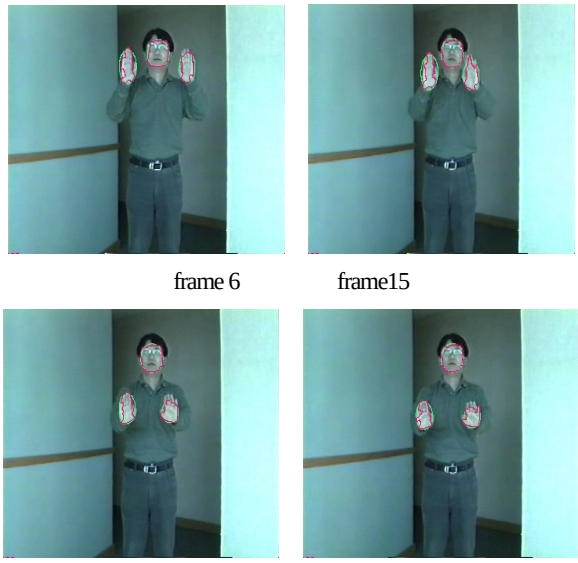
Figure 5-8 gives the results from the "Left", "Forward", "Right" and "Stop" gesture sequence respectively (normally in a sequence there are 20-50 frames). We illustrate simultaneously the tracked region contours (in red) and corresponding ellipses (in green) on each frame. In each sequence, there are some time instants while the face is closer to some hand. If only using the skin color information, it is very hard to distinguish the skin regions of the head and those of the hand, and the region correspondence between consecutive frames like [12] will fail in this case. Our method yields good tracking results, and both the motion and shape parameters for dynamic gesture recognition are obtained too.



frame 8 frame12

frame15 frame18

Figure 5 results from the " Left" gesture sequence



frame 6 frame15

frame 23 frame 28

Figure 6 Tracking results for the " Forward" gesture sequence

## 4. Conclusion

In this paper we have proposed a two-hands-gesture tracking method, its character is full utilization of the object spatio-temporal features in tracking. . The template warping only gives a prediction to the tracked feature position, and the comparison of each sub-region with the warped are able to modify the prediction result. Like [12], we have estimated the motion (affine) parameters, shape features (ellipse fitting) and obtained (region centroid) trajectories of both hands with respect to the head. All the information could be useful for two-handed dynamic gesture recognition.

The disadvantages of our method are also clear in the experiments. First, we rely on the motion estimation of the tracked object; Even though the IWLS method is more stable, we still confronted the divergence in nonlinear iterations. Second, while we introduce the static segmentation result our method has strong dependence on the performance of the employed watershed algorithm. We expect the images in the sequence are with enough resolution, and the motion blur are also not welcome.

In future, we plan to consider the variations of illumination during tracking [8], which is also an important factor in tracking.

## Acknowledgement

This research is supported partially by the NSF Grants EIA-99-75019 and IIS-00-85980.

## References

[1] Brand, M et.al.'Coupled Hidden Markov Models for complex action recognition', IEEE CVPR'97, pp 994 –999, 1997.

[2] Cutler R, Turk M. "View-based Interpretation of Real-time Optical Flow for Gesture Recognition", IEEE International Conference on Automatic Face and Gesture Recognition. 1998.

[3] Gauch J, 'Image segmentation and analysis via multiscale gradient watershed hierarchies', *IEEE T-IP*, 8(1): 69-79, 1999.

[4] Hong P et.al. "Gesture modeling and recognition using finite state machines", IEEE International Conference on Automatic Face and Gesture Recognition, pp410 –415, 2000.

[5] Imagawa I et. al, 'Recognition of local features for camera-based sign language recognition system', Proc. ICPR'00, pp849-853, 2000.

[6] Nguyen H., Worring M., "Multifeature object tracking using a model-free approach", IEEE CVPR, pp 145 –150, 2000.

[7] Saber E, Tekalp A, 'Face detection and facial feature extraction using color, shape and symmetry-based cost functions', ICPR'96, pp654-658, 1996.

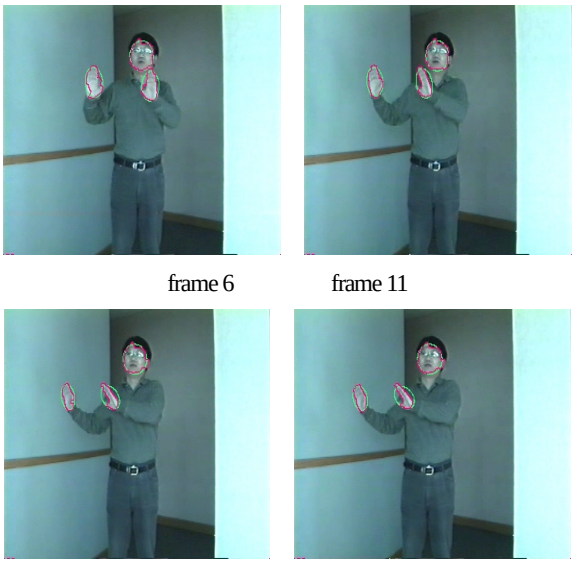
[8] Sclaroff S. and Isidoro J., "Active blobs", ICCV'98, 1998.

[9] Sherrah J, Gong S., 'Tracking discontinuous motion using Bayesian inference', ECCV'00, 2000.

[10] Utsumi A., Ohya J. 'Multiple-hand-gesture tracking using multiple cameras', pp473-478, IEEE CVPR'99.

[11] Vincent L, Soille, 'Watersheds in digital spaces: an efficient algorithm based on immersion simulations', *IEEE T-PAMI*, 13(6): 583-589, 1991.

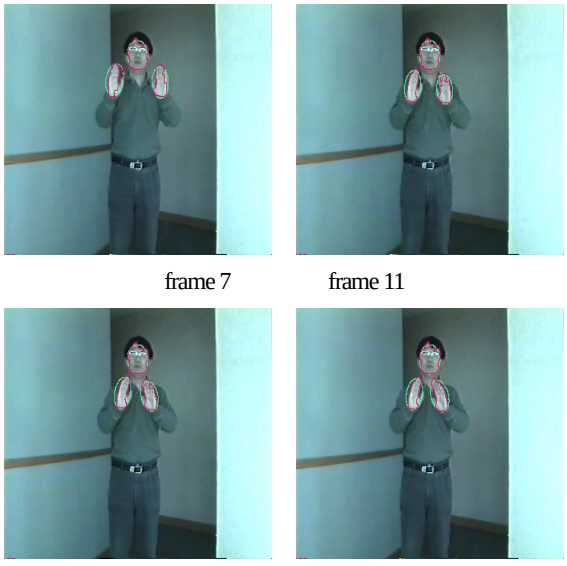
[12] Yang M, Ahuja N, 'Recognizing hand gesture using motion trajectories', CVPR'99, pp892-897, 1999.



frame 6 frame 11

frame 19 frame 24

Figure 7 Tracking results for the " Right" gesture sequence



frame 7 frame 11

frame 14 frame 19

Figure 8 Tracking results for the " Stop" gesture sequence