Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition 38 (2005) 2373–2384

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Tracking non-rigid objects using probabilistic Hausdorff distance matching☆

Sang-Cheol Park[a], Sung-Hoon Lim[a], Bong-Kee Sin[b], Seong-Whan Lee[a,*]

[a]Department of Computer Science and Engineering, Center for Artificial Vision Research, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-701, Korea

[b]Division of Electronics, Computer and Telematics Engineering, Pukyong National University, Taeyon-dong, Nam-ku, Busan 608-737, Korea

Received 26 November 2003; received in revised form 3 December 2004; accepted 7 January 2005

Abstract

This paper proposes a new method of extracting and tracking a non-rigid object moving against a cluttered background while allowing camera movement. For object extraction we first detect an object using watershed segmentation technique and then extract its contour points by approximating the boundary using the idea of feature point weighting. For object tracking we take the contour to estimate its motion in the next frame by the maximum likelihood method. The position of the object is estimated using a probabilistic Hausdorff measurement while the shape variation is modelled using a modified active contour model. The proposed method is highly tolerant to occlusion. Unless an object is fully occluded during tracking, the result is stable and the method is robust enough for practical application.

Keywords: Object tracking; Active contour; Watershed segmentation; Hausdorff distance

1. Introduction

Thanks to the recent surge of interest in computer-based video surveillance, video-based multimedia service and interactive broadcasting, the problem of tracking objects in video has attracted a lot of researchers around the world. Object tracking is so basic and in high demand that it is an indispensable component in many applications including robot vision, video surveillance, object-based compression, etc. The problem of object tracking can be divided into two subproblems, extraction of a target object and tracking it over time. In general even those subtasks are not easy because of cluttered backgrounds and frequent occlusions. However, we can readily find a large number of studies on diverse methods using a variety of features such as color, edge, optical flow, and contour [1,2].

A target object under tracking involves motion, which we classify into three types; in the increasing difficulty of analysis, they are (1) object motion with a static camera, (2) camera motion over a static scene/object, and (3) simultaneous movement of camera and objects. To speak in terms of applications, the first type of motion has been heavily studied by the students in video surveillance, while the second type has been a prime target of analysis in video compression. The last type of hybrid motion, though more involved, has been an important topic in object-based services, object-based video compression, and interactive video.

This paper discusses an efficient method of extracting the contours of and tracking an interesting object in video from

☆ This research was supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Science and Technology of Korea.

* Corresponding author. Tel.: +82 2 3290 3197; fax: +82 2 926 2168.

E-mail addresses: scpark@image.korea.ac.kr (S.-C. Park), shlim@image.korea.ac.kr (S.-H. Lim), bkshin@pknu.ac.kr (B.-K. Sin), swlee@image.korea.ac.kr (S.-W. Lee).

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.01.015

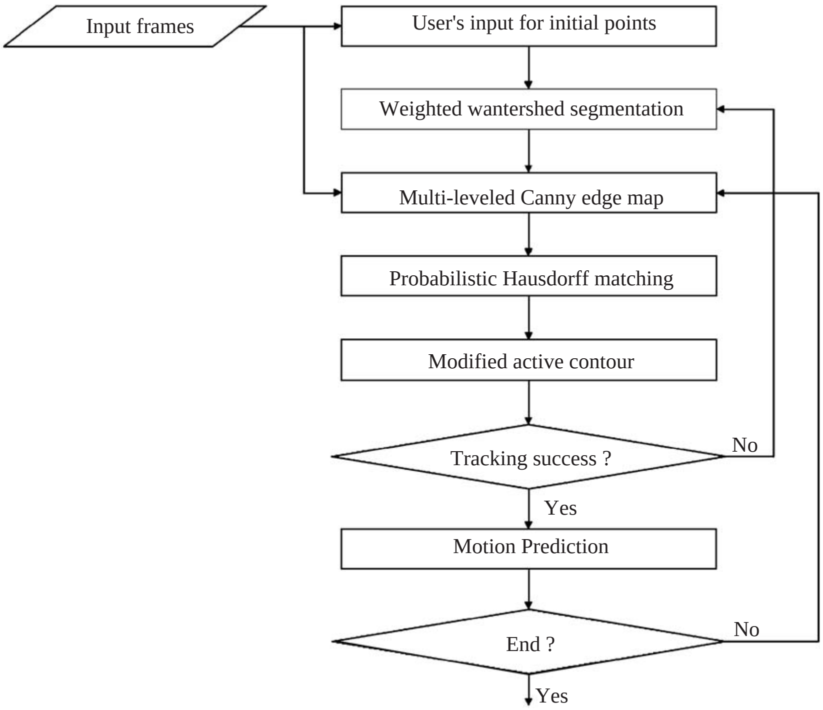

Fig. 1. The block diagram of the proposed method.

a non-static camera. Contour extraction step first estimate the rough contour of an object, and then locate and delineate the exact contour. Since the initial contour estimation starts from a given seed or a rough estimate, it is desired to make the initial error as small as possible. For this, we have developed a modified watershed algorithm based on Gaussian weighted edges. The resulting contour is then used to estimate the motion vectors and locate the objects in the succeeding frames.

In the object tracking step, there are two subtasks, global motion estimation using the Hausdorff matching method and local deformation analysis using the modified active contour model. The overall procedure is illustrated in Fig. 1.

The paper is organized as follows: in Section 2 a number of related works are reviewed briefly with the focus in extracting and tracking non-rigid foreground objects in video. Section 3 describes functional/system components and algorithms. Section 4 provides experimental results along with extraction and tracking examples. Then Section 5 concludes the paper.

2. Related works

This section reviews the representative studies concerning the techniques of extracting and tracking objects and point out their advantages and disadvantages.

2.1. Object extraction

The goal of object extraction is to locate and delineate the area of an interesting object in a given image. The methods for extracting objects differ by the features they use: temporal features, spatial features, or both.

Wang and Adelson [3] proposed a method of using motion vectors while assuming motionless static background. The method can extract any moving objects from the motion of segmented regions. Since, however, it does not have an explicit model for camera movement, the method requires fixing the background and the camera as well.

An alternative type of the object extraction method is to use spatial domain features, typically, active contour model or snake [5]. The active contour model is a highly useful tool for evaluating the boundary of object by minimizing an energy function. Amongst the various solutions are the optimization technique [7], the finite difference method [4], the finite element method [5], the dynamic programming [6].

On the other hand Cohen [8] introduced a related model of the normalized gradient of energy function. Similarly Ivins and Porrill presented the dual Active Region Models [9] with which they obtained two initial contours, inner and outer, of an object, and then minimized their energy functions to locate objects using a slant descent technique. Astrom and Kahl required a user's input as an initial contour of an object [10]. Similarly Zhong and Chang developed a system AMOS

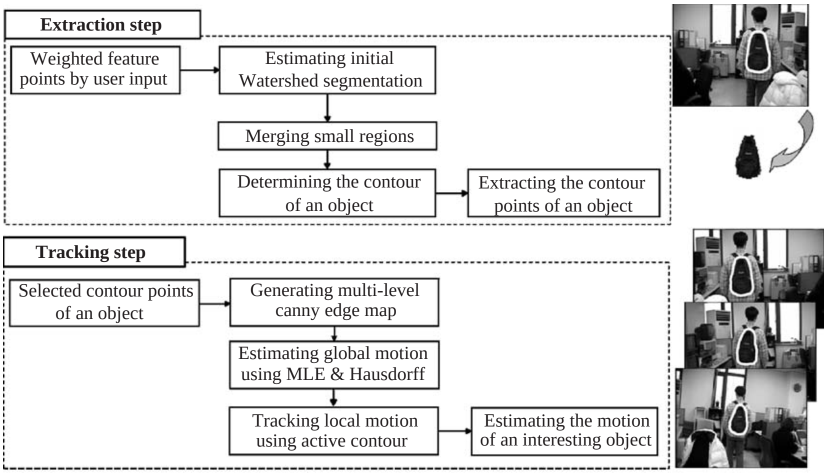

Fig. 2. The overview of the proposed method.

[11] that took user input to refine the contour and predict its motion. These methods have an advantage of producing very accurate object contours, although the requirement of user input is a serious hurdle to achieve a fully automatic system.

There is still another new group of the object extraction methods, collectively called an area-based method. They consider the relation between image pixels. Quite recently, Gu and Lee proposed an object extraction method using the watershed segmentation technique [12]. By the same token Long and Feng [13] proposed dividing an image via dynamic similarity and then extracting an object through the differential decomposition of the divided image in the consecutive frames.

2.2. Object tracking

Object extraction, if efficient and accurate, may entirely obviate the need for tracking objects. It is defined as the task of estimating the motion of an interesting object in video predicting its position in successive frames. What makes the problem so hard and lowers the performance is the presence of partial or full occlusion. Although numerous researchers have struggled with the problem with various proposals, the problem of occlusion still remains essentially unsolved. Typical object tracking algorithms employ the technique of the differential decomposition between two images so as to minimize the Sum of Squared Differences (SSD) [14] and use the object feature points or the contour to improve the tracking accuracy.

Oberti and Regazzoni [15] proposed a method of watching people inside a room. First they detected person's edge. They used it to define the edge area by approximating the person's edge. Then they used it as a tracking object. This method has been successfully applied to tracking indoor objects free of large abrupt motion. With abrupt illumination changes, the complex background and the edge may be lost due to motion blur. In practice, however, this method often failed by falling into local optima. Zhong et al. [16] proposed a slightly different method that considered the amount of occlusion to the deformable shape and minimized it. Nevertheless, they could not solve complete occlusions. Slightly earlier Freedman and Brandstein [17] proposed a method of extracting and tracking the contour of a foreground object moving against cluttered background. They simplified the problem by the shape learning technique.

3. Proposed contour extraction and motion tracking

This section explains the proposed method based on brief theories and algorithms underlying. It starts with a brief review on watershed segmentation. Then it will move on to object extraction features, and then to Hausdorff matching-based object tracking. Refer to Fig. 2 while following the description of the section.

3.1. Watershed segmentation

The watershed transform is a well-known method of choice for image segmentation in the field of mathematical morphology. The watershed transform can be classified as a region-based segmentation approach. To date a number of algorithms have been developed to compute watershed transforms. They can be divided into two classes, one based on the specification of a recursive algorithm by Vincent and Soille [21], and the other based on distance functions by Meyer [18]. The details of the watershed algorithm of this paper is based on the works developed by Nguyen et al. [22], and Vincent and Soille [21]. The algorithm consists of the following steps:

(1) Compute the gradient map by feature points, such as edge pixels, for all pixels in the image.
(2) Sort the feature points according to gradient values.
(3) Select the pixel with the smallest value.
(4) Label the connected region of the pixel with geodesic distance to the neighborhood region.
(5) Repeat steps (3), (4) until all pixels are labeled.
(6) Extract the boundaries of the special labels.
(7) Merge the labels to several regions.

The watershed segmentation of an image is described by the following sequence of definitions

Definition 1. The geodesic distance $d_A(a, b)$ between feature point a and b within A is the minimum paths within A from a to b. If B is a subset of A, we define $d_A(a, B) = min_{b\in B}(d_A(a, b))$.

Definition 2. Let $B \subseteq A$ be partitioned into k connected components B_i, $i=1,\dots,k$. The geodesic region influence determined by the set B_i within A is defined as

$$i_A(B_i) = \{p \in A | \forall_j \in [1,\dots,k] : d_A(p, B_i) < d_A(p, B_j)\}. \quad (1)$$

In addition the set $I_A(B)$ is the union set of the geodesic regions of all connected components of B and defined by

$$I_A(B) = \bigcup_{i=1}^{k} i_A(B_i). \quad (2)$$

Note that the complement of $I_A(B)$ within A is called the skeleton. Now we can introduce the definitions of the watershed segmentation operation, which may be viewed as a generalization of the skeleton by influenced regions.

Let an image f be an element of the space $C(D)$ of real twice continuously differentiable functions on a connected domain D with only isolated critical points.

Definition 3. The topographical distance between a point $p \in D$ and a set $A \subseteq D$ is defined as $T_f(p, A) = \min_{a\in A} T_f(p, a)$. The path with the shortest T_F distance between p and q is a path of the steepest slope.

Definition 4. According to minima $\{m_k\}_{k\in I}$, for some index set I which is the number of small geodesic regions, the catchment basin $CB(m_i)$ of a minimum m_i is defined as the set of points $x \in D$ which are topographically closer to m_i than to any other regional minimum m_j:

$$CB(m_i) = \{x \in D | \forall_j \in I : f(m_i) + T_f(x, m_i) < f(m_j) + T_f(x, m_j)\}. \quad (3)$$

The watershed *WS* of image *f* is the set of points which do not belong to any catchment basin:

$$WS(f) = D \cap \left(\bigcup_{i\in I} CBm_i\right)^c. \quad (4)$$

3.2. Object extraction using the watershed segmentation

The proposed method to work even in images with complicated background. The basic idea lies in feature point weighting in the watershed algorithm.

3.2.1. Initial watershed segmentation

Object tracking in general involves a repeated extraction of an object over a sequence of frames. And the performance of object tracking as well as extraction in the current frame is directly related to that of extraction in the preceding frame. In the case of the first frame it is assumed that the user supplies the initial rough boundary consisting of several key feature points. We believe that this may be improved with an effort focused on the problem.

An important feature of proposed method is giving a Gaussian weight to the initial feature points. This helps removing the need for selecting all detailed feature points of an object. Let $X = (X_1, \dots, X_n)$ be an ordered sequence of feature points $X_i = [x_i, y_i]^t$, $i = 1, \dots, n$. And let us denote the magnitude of the displacement from the origin to X be $m(X)$, and the Gaussian weight for the feature point X as $G(X)$. Then the weighted magnitude of the displacement vector to X for the watershed segmentation is given by

$$M_{new}(X_i) = G(X_i) * M(X_i),$$

$$G(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (5)$$

where $*$ denotes 2D convolution operation. The watershed algorithm proceeds according to Eq. (5) with a sequence of weighted feature points. The watershed segmentation algorithm is a region-based technique. The result is a set of image regions dividing the input image.

3.2.2. Region merging

Since the watershed segmentation in general shows numerous small chaotic regions, there naturally arises a need to merge them into a single connected component which will assumedly define the object boundary. Fig. 3 illustrates the overall object extraction process, in which a numerous fragmental regions are produced and then merged into a single complete contour. In the region merging step, we simply merge the regions one by one until we are left with only two regions, foreground and background. In each iteration, the smallest region is found and merged into the nearest region. The choice of the target neighbor is based on the minimum gradient magnitude. And the final result consists of an object region and background region.

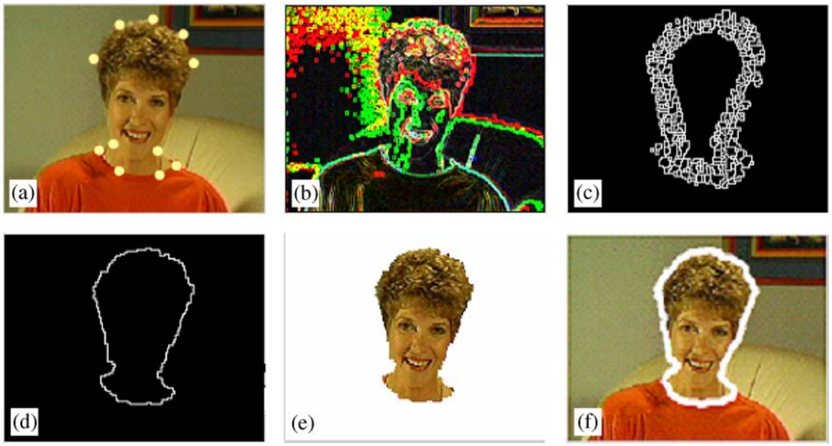

Fig. 3. Watershed segmentation process steps, (a) an input image and selected feature points, (b) an edge image, (c) initial small regions, (d) merged region, (e) segmented result and (f) extracted object.

3.3. Contour tracking

An object in an image is defined by a contour. A contour is sufficient for our task. The proposed tracking method takes the contour generated in preceding step to analyze the object contour's motion. The motion estimate is based on probabilistic Hausdorff distance measurement between the previous contour and all possible current candidate regions.

3.3.1. Motion tracking with Hausdorff distance

For the candidate feature set in the next frame, we use the feature map which is calculated by multi-level canny edge detector (see Fig. 4).

These level edge maps are used to calculate the Hausdorff distance, then each level has the different weight. If we use three level edge map, the real distance can be calculated by the following equation:

$$HD = H_{L1}(M, R) * \omega_1 + H_{L2}(M, R) * \omega_2 + H_{L3}(M, R) * \omega_3, \qquad (6)$$

where ω_i is the weight coefficient and H_{L1}, H_{L2} and H_{L3}, are the Hausdorff distance calculated from the first, second, and third level edge map, respectively.

The Hausdorff matching score is the Hausdorff distance between two finite feature point sets [20]. The Hausdorff distance is used to measure the degree of similarity between the extracted contour and a search region in video frames. Let us assume that we are given a sequence of reference feature points, $M=(m_1, m_2, \ldots, m_k)$ computed in the preceding frame, and let a new sequence of image feature points, $R=(r_1, r_2, \ldots, r_n)$ from the current frame. The Hausdorff distance between M and R can be written by

$$H(M, R) = \max(h(M, R), h(R, M)), \qquad (7)$$

where

$$h(M, R) = \max_{m\in M} \min_{r\in R} \|m - r\|, \qquad (8)$$

where $\| \ \|$ is the distance between two points, e.g., Euclidean norm. This distance is sensitive to noise and occlusion, we extend the function $h(M, R)$ so that the modified Hausdorff can measure the partial distance between a reference feature and an image feature. We use the likelihood function to find the position of the object.

First, to estimate the position of the model using the maximum likelihood, we define the set of distance measurements between reference feature points and input image feature points. Let $d(m, R)$ be the partial distance, we use some measurements of the partial distance. We consider the following distance measurements:

$$D_1(M, R) = \min_{m\in M} d(m, R), \qquad (9)$$

$$D_2(M, R) = {}^{75}K^{th}_{m\in M} d(m, R), \qquad (10)$$

$$D_3(M, R) = {}^{90}K^{th}_{m\in M} d(m, R), \qquad (11)$$

where the partial distance is defined as $d(m, R) = \min_{r\in R} \|m - r\|$ and ${}^xK^{th}_{m\in M}$ represents the K^{th} ranked distance that $K/m_k = x\%$ [20].

$$D_4(M, R) = \min_{m\in M} \sum_W [I(m) - I(R)]^2, \qquad (12)$$

$$D_5(M, R) = \min_{m\in M} \sum_W |I(m) - I(R)|. \qquad (13)$$

There are several distance measurements to estimate the partial distance, e.g., Euclidean distance, city-block distance, the distance transform, the sum of squared difference, optical-flow feature tracking method which can be used for the distance measurement defined as Eq. (12).

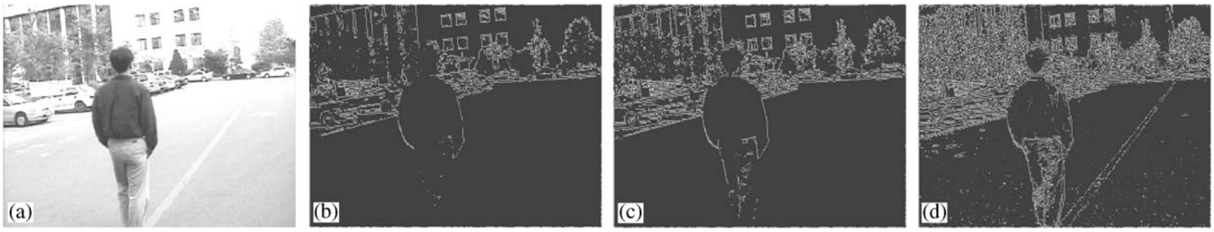

Fig. 4. Multi-level canny edge: (a) an input frame, (b) the 1st level edge, (c) the 2nd level edge and (d) the 3rd level edge.

3.3.2. Probabilistic Hausdorff measurement

Let s be the position of an object in the current frame, then distance measurements are denoted $D_1(s), D_2(s), \ldots, D_l(s)$. We use these distances to match maximum likelihood. In order to estimate the contour motion problem, we use the Olson's maximum-likelihood matching method [19]. Then the joint probability of the distance measurements $D_i(s)$ can be written as

$$p(D_1(s), \ldots, D_m(s)|s) = \prod_{i=1}^{l} p(D_i(s)), \tag{14}$$

where $p(D_i(s))$ is the probability density function of normalized distance of $D_i(s)$ at the position s, that is, $p(D_i(s)) = (1 - D_i(s))$, p must be maximized by Eq. (14). The likelihood for p is formulated as the product of the prior probability of the position s and the probability in Eq. (15):

$$L(s) = p(s) \prod_{i=1}^{m} p(D_i(s)). \tag{15}$$

For convenience, we will take the logarithm of Eq. (15) as follows:

$$\ln L(t) = \ln p(t) + \sum_{i=1}^{m} \ln p(D_i(t)). \tag{16}$$

We search s that maximizes this likelihood function, then this result uses to estimate the global motion for an interesting object.

3.3.3. Active contour motion model

The conventional active contour model consists of two terms, internal and external energy terms. The discrete energy function in the active contour model is defined as follows:

$$E(t) = \sum_t [E_{int}(t) + E_{ext}(t)]. \tag{17}$$

In this paper we propose a modified active contour motion model that adds an external motion term. The proposed energy function contains four terms: continuity, curvature, image force, and motion estimation confidence as follows:

$$\begin{aligned} E(t) = \sum_i [&\alpha E_{cont}(v_i(t)) + \beta E_{curv}(v_i(t)) \\ &+ \gamma E_{img}(v_i(t)) + \eta E_{match}(v_i(t))], \end{aligned} \tag{18}$$

where contour energy functions, continuity energy, $E_{cont}(v(t))$, curvature energy, $E_{curv}(v(t))$, image force (edge/gradient), $E_{img}(v(t))$ and motion estimation confidence, $E_{match}(v(t))$. This last term in the above equation measures the motion variation between feature points in current frame and those in the previous frame. It can be defined as

$$E_{macth}(t) = \sum_{i}^{n} |\bar{v}(t) - v_i(t)|^2, \tag{19}$$

where $\bar{v}$ is the average motion vector at time t, v_i is the motion vector of the ith feature point in frame t.

The position of the contour must be predicted for the efficient tracking. To predict the position of the contour in the next frame, we calculate the following prediction energy, the predicted position of the contour at time $t+1$

$$\begin{aligned} E_{pre}(t+1) = E_{real}(t) + \sum_{v_i \in V} [&\varepsilon \cdot (E_{real}^{v_i}(t) - E_{real}^{v_i}(t-1)) \\ &+ (1-\varepsilon) \cdot (E_{real}^{v_i}(t) - E_{pre}(t))], \end{aligned} \tag{20}$$

where $E_{pre}(t+1)$ is the predicted position of the contour at time $t+1$ and $E_{real}(t)$ is the real energy term of the moved contour at time t and ε is the matching rate between $E_{real}(t)$ and $E_{pre}(t)$. We proposed the probabilistic method to extract and to track an object using watershed and Hausdorff matching.

4. Experimental results and analysis

We have proposed the object tracking method using the modified Hausdorff distance matching and active contour algorithm. It has been tested over a set of VHS video clips recorded. The background includes streets, cars, indoor laboratory scenes. All the deformable targets are humans movement freely. The test images have been in monochrome MPEG-1 format with the dimension of 320×240. For comparison of the performance, we will refer to the method of Kass's active contour [5].

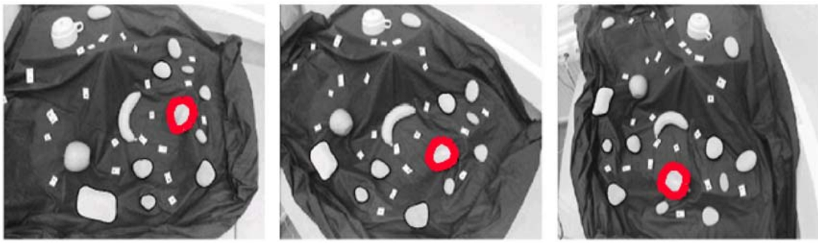

Fig. 5. Astrom and Kahl's test images [10].

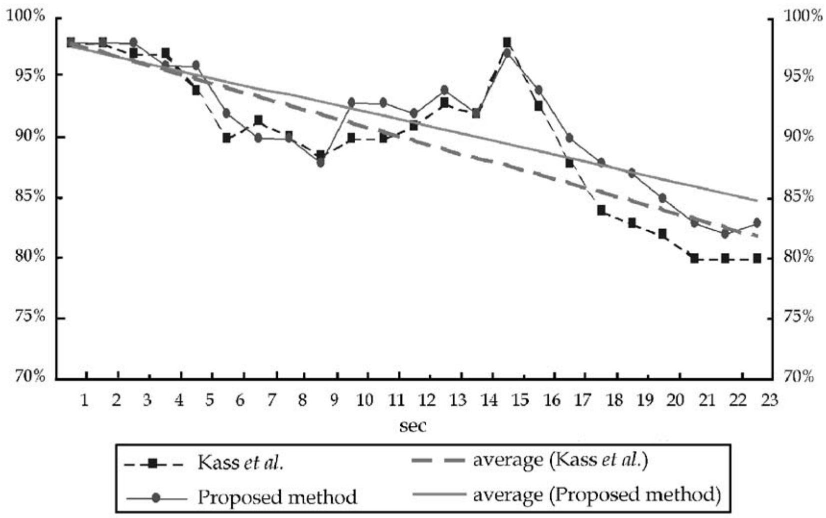

Fig. 6. The comparison to the tracking performance using Fig. 5.

4.1. Simple tracking

Fig. 5 shows the three frames of a video sequence used in the work of Kass et al. [5]. We took the clip to compare the tracking performance with their method. The background of the video is relatively simple. The target objects include a cup, a pimento, a banana, an archetype and several polygonal objects. They are stationary, but the camera moves, apparently panning, tilting and zooming.

For the tracking performance $tr(t)$ at time t, we calculated the ratio of the number of pixels of the real area $np_r(t)$ to that of the segmented area $np_e(t)$. Also, for occlusion rate, $occ(t)$, at time, t, we used the pixel difference between pixels of the real area $np_r(t)$ and that of the occluded area $np_o(t)$.

$$tr(t) = \frac{np_e(t)}{np_r(t)}, \tag{21}$$

$$occ(t) = \frac{np_o(t)}{np_r(t)}. \tag{22}$$

Fig. 6 shows the value of $tr(t)$ over the sequence of the video clip. The two straight lines downward show the performance degradation averaged over the entire sequence.

4.2. Tracking with occlusion

The next set of tests concerns the performance of tracking under occasional occlusions, partially or fully (see Fig. 7). When an object is occluded by a nearer object, then the tracking problem becomes harder. Therefore it is natural to consider the degree of occlusion alongside the tracking rate. We define a simple complementary measure, the occlusion rate $occ(t)$ that denotes the percentage of the area occluded:

Test using videos involving occlusion was carried out over several samples. The result is summarized graphically in Fig. 8. The average tracking rate of Kass's method is 68%, while that of the proposed method is 93%. As it can be seen in Fig. 8, the object tracking rate falls as the target get occluded by another object. The difference occurs when the degree of fall becomes significant. In the case of Kass's degradation starts as soon as the occlusion starts; on the other hand the proposed method begins to fail only when there is a substantial overlap of over 65%.

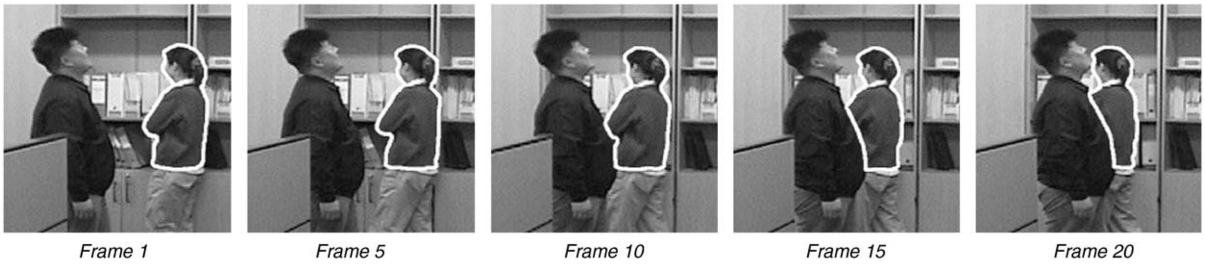

Fig. 7. Tracking experiment with a partial occlusion.

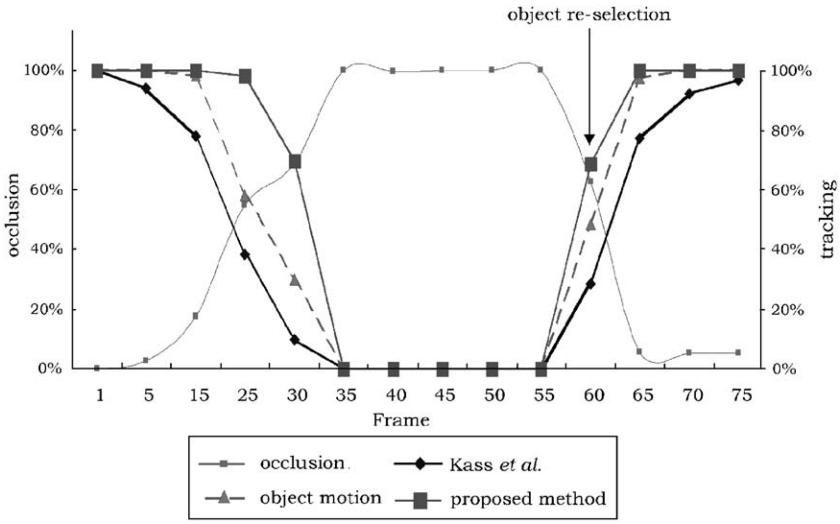

Fig. 8. Tracking rates changing with partial and full occlusion.

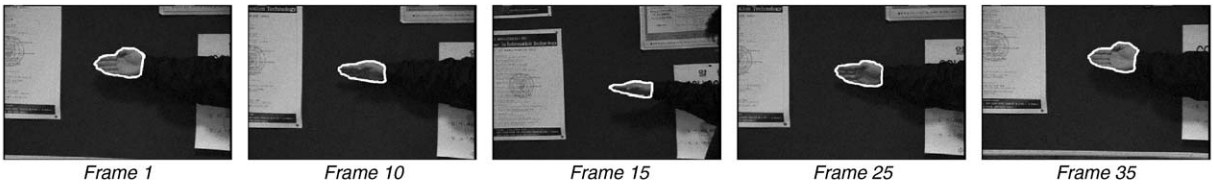

Fig. 9. Tracking result for a hand.

4.3. Tracking a deformable object

This experiment was tested to measure the object tracking to the degree of images using the affine transformation. The target object's shape changes by time. In this test the target object changes shape or the outer boundary over time. Fig. 9 shows the experimental result for the deformable object in the consecutive frames.

4.4. Tracking a human face

For another test, we took a face against background of studio using announcer's face motion images; in this condition, not only could we saved time required to process texture, but also the quality of tracking results was very good. Fig. 10 shows a typical case to which the tracking module was applied.

4.5. Long sequence tracking performance

In realistic settings, the time span of moving object tracking tends to increase beyond a few or tens of frames. It is especially true when the camera moves to follow a moving target object. Hence a practical system is required to operate in extended sequences of video. The current experiment involves long sequence video data.

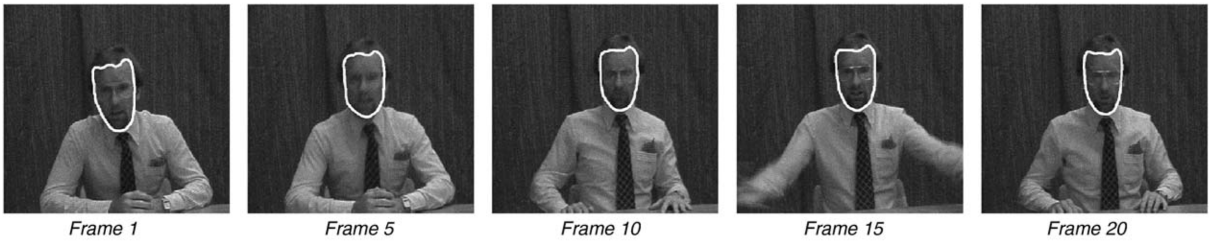

Fig. 10. Tracking an announcer's face in studio.

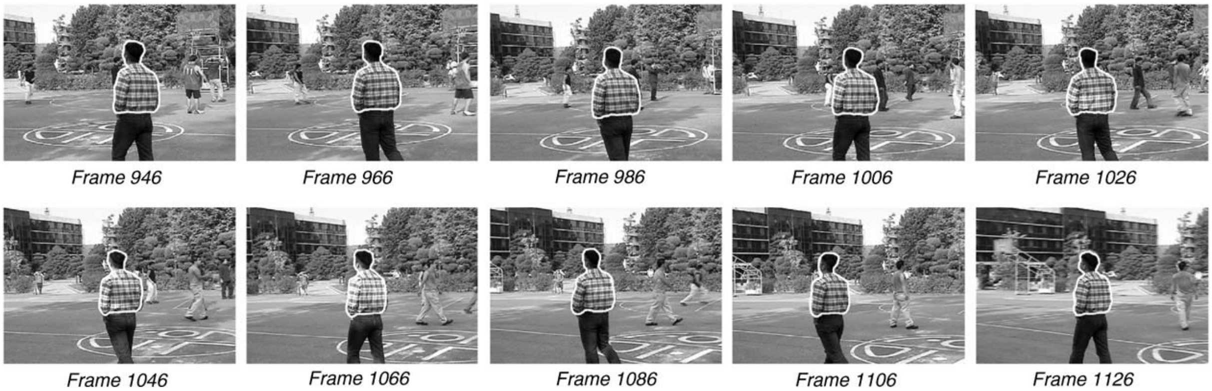

Fig. 11. Human body tracking in a parking lot.

Table 1
Experiment over a long sequence video

	Average number of mis-tracked pixels	Tracking rate (%)
Snake model	255	88
Adaptive template	287	86
Proposed method	174	93

Fig. 11 shows selected samples from a video of 180 frames (numbered from 946 to 1126) where the human upper body is the target of tracking. The object boundary in the first frame was manually given by a user. It consists of a sequence of feature points appropriately defining the boundary. Note that the boundaries of subsequent tracking frames are highly accurate. The performance of tracking over the long sequences is summarized in Table 1 and Fig. 12.

The proposed method records 5% over the standard snake and template models. Moreover the accuracy degrades much slowly compared with those of snake and template.

The object boundary was manually selected by a user in the first frame, several feature points of a object were simply selected by a user. Furthermore, a human body sequence captured in our lab was used to check the performance experiment in the different environment.

People in motion usually involves changes in pose or face direction. Such variations do not pose any problem in the proposed method. Fig. 13 shows a successful result under the pose variation of the man and Fig. 14 shows the experimental result under the size variation of the target object. In order to compare the experimental results quantitatively, we measured the successful tracking rate and the mistaken pixels of tracking process from each frame in Fig. 12.

5. Conclusions and future works

This paper proposed an effective contour tracking method for a deformable object which was described by a set of feature points. The proposed method works successfully in spite of non-static camera and non-linear deformation over time with cluttered backgrounds. The extracting result has shown that the proposed method separates well target foreground objects from the background by analyzing the background texture.

Through a series of experimental results, we could confirm that the proposed method was quite stable and produced good results under object shape deformation and non-static camera motion. One problem with the implementation of the proposed system is that it sometimes fails if an object moves too fast. Another problem is that its processing time

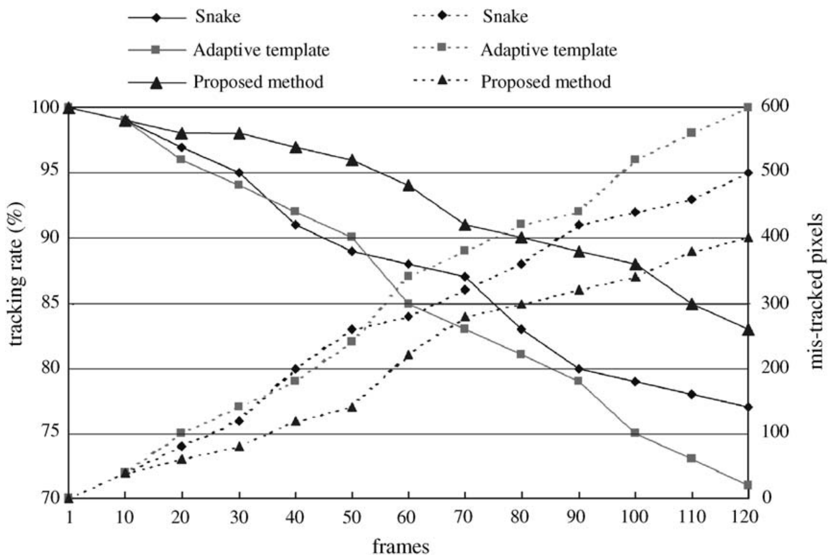

Fig. 12. Tracking rate and mis-tracked pixels for each frame.

Fig. 13. The experimental result of a person tracking in a room.

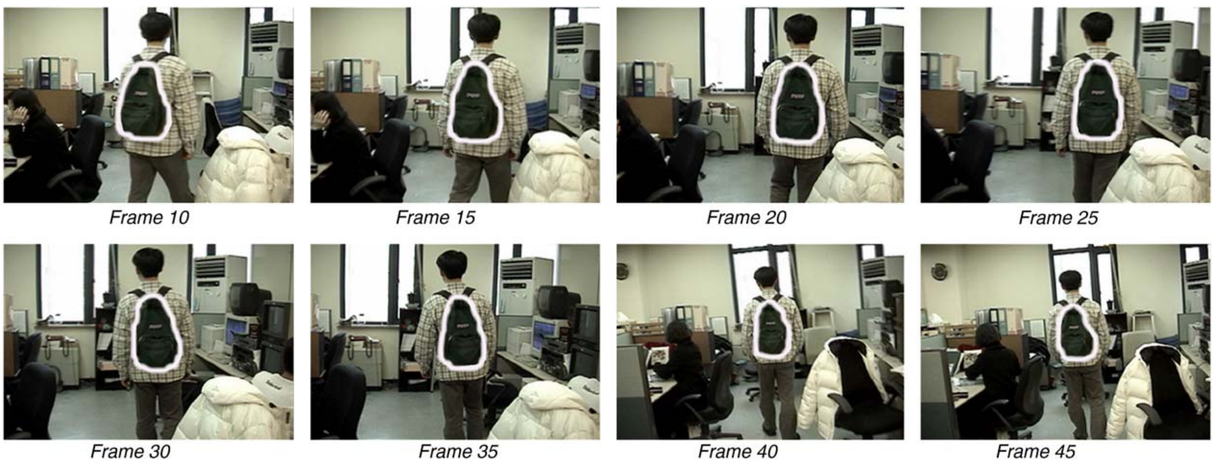

Fig. 14. The experimental result of a knapsack tracking.

is so slow and, in its current form is not suited to realtime application. We believe that these problems can easily be overcome. The real issue of the paper is the performance and robustness of extracting and tracking deformable objects.

References

[1] K. Sobottka, I. Pitas, Segmentation and tracking of faces in color images, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, vol. 1, October 1996, pp. 236–241.

[2] F. Leymarie, M.D. Levine, Tracking deformable objects in the plane using an active contour model, IEEE Trans. Pattern Anal. Mach. Intel. 15 (6) (1996) 617–634.

[3] J.Y.A. Wang, E.H. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3 (5) (1994) 625–638.

[4] D.H. Ballard, C.M. Brown, Computer vision, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[5] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active Contour Models, in: Proceedings of the International Conference on Computer Vision, London, England, vol. 1, June 1987, pp. 259–268.

[6] K. Cheung, T. Lee, R.T. Chin, Boundary detection by artificial neural network, in: Proceedings of the International Joint Conference on Neural Networks, Nagoya, Japan, vol. 2, October 1993, pp. 1189–1194.

[7] L.D. Cohen, I. Cohen, Finite-element methods for active contour models and balloons for 2-D and 3-D images, IEEE Trans. Pattern Anal. Mach. Intel. 15 (11) (1993) 1131–1147.

[8] L.D. Cohen, On active contour models and balloons, Comput. Vision Graph. Image Process.: Image Understand. 53 (2) (1991) 211–218.

[9] J. Ivins, J. Porrill, Active region models for segmenting textures and colours, Image Vision Comput. 13 (2) (1994) 431–438.

[10] K. Astrom, F. Kahl, Motion estimation in image sequences using the deformation of apparent contours, IEEE Trans. Pattern Anal. Mach. Intel. 21 (2) (1999) 114–127.

[11] D. Zhong, S.F. Chang, AMOS: an active system for MPEG-4 video object segmentation, in: Proceedings of the International Conference on Image Processing, Chicago, IL, USA, vol. 2, October 1998, pp. 647–651.

[12] C. Gu, M.C. Lee, Semiautomatic segmentation and tracking of semantic video objects, IEEE Trans. Circuits Syst. Video Technol. 8 (5) (1998) 572–584.

[13] F. Long, D. Feng, H. Peng, W.C. Siu, Extracting semantic video objects, IEEE Comput. Graph. Appl. 21 (1) (2001) 48–54.

[14] G.D. Hager, P.N. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. Pattern Anal. Mach. Intel. 20 (10) (1998) 1025–1039.

[15] F. Oberti, C. Regazzoni, Adaptive tracking of multiple non-rigid objects in cluttered scenes, in: Proceedings of the International Conference on Pattern Recognition, Barcelona, Spain, vol. 3, September 2000, pp. 1096–1099

[16] Y. Zhong, A.K. Jain, M.P. Dubuisson-Jolly, Object tracking using deformable templates, IEEE Trans. Pattern Anal. Mach. Intel. 22 (5) (2000) 544–549.

[17] D. Freedman, M.S. Brandstein, Methods of global optimization in the tracking of contours, in: Proceedings of the Annual Asilomar Conference on Signals Systems and Computers, Pacific Grove, Canada, vol. 1, October 1999, pp. 725–729.

[18] F. Meyer, Topographic distance and watershed lines, Signal Process. 38 (1994) 113–125.

[19] C.F. Olson, Maximum-likelihood image matching, IEEE Trans. Pattern Anal. Mach. Intel. 24 (6) (2002) 853–857.

[20] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Computing images using the Hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intel. 15 (9) (1993) 850–863.

[21] L. Vincent, P. Soille, Watershed in digital spaces: an efficient algorithm based on immersion simulations, IEEE Trans. Pattern Anal. Mach. Intel. 13 (1991) 583–598.

[22] H.T. Nguyen, M. Worring, R. Boongaard, Watersnakes: energy-driven watershed segmentation, IEEE Trans. Pattern Anal. Machine Intel. 25 (3) (2003) 330–342.

About the Author—SANG-CHEOL PARK was born in Mokpo, Korea, in 1971. He received his B.S. and M.S. degrees in computer science from Chonnam National University, Kwangju, Korea, in 1996 and 1998. He received a Ph.D. degree in the Department of Computer Science and Engineering at Korea University, Seoul, Korea in 2004. His research interests include computer vision, multimedia, and augmented reality.

About the Author—SUNG-HOON LIM was born in Seoul, Korea, in 1973. He received his B.S. degree in computer science from Dong-kuk University, Seoul, Korea, in 2000. He received his M.S. degree in Computer Science and Engineering from Korea University, Seoul, Korea, in 2002. His research interests include pattern recognition and video processing.

About the Author—BONG-KEE SIN received B.S. degree in Mineral and Petroleum Engineering from Seoul National University, Seoul, Korea, in 1985, and M.S. degree in Computer Science from Korea Advanced Institute of Science and Technology or KAIST in 1987. Then he had worked for the Software Research Labs of Korea Telecom until February 1999. Between 1991 and 1994, he continued his study for his Ph.D. in computer science in KAIST. In March 1999, he joined the faculty of the Department of Computer Multimedia Engineering in Pukyong National University, Busan, and is now an assistant professor. His general research interest includes character recognition and various applications of statistical pattern recognition methods in general and HMM in particular. His current research focus lies in the analysis and recognition of various sequential data including music signal, genomic sequence, and video.

About the Author—SEONG-WHAN LEE received his B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1984, the M.S. and Ph.D. degrees in computer science from KAIST in 1986 and 1989, respectively. From February 1989 to February 1995, he was an assistant professor in the Department of Computer Science at Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering at Korea University, Seoul, Korea,

as an associate professor, and now he is a full professor. He was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. He also obtained the Outstanding Research Award from the Korea Information Science Society in 1996. He has been the co-Editor-in-chief of the International Journal on Document Analysis and Recognition since 1998 and the associate editor of the Pattern Recognition Journal, the International Journal of Pattern Recognition and Artificial Intelligence, and the International Journal of Computer Processing of Oriental Languages since 1997. He was the Program co-chair of the Sixth International Workshop on Frontiers in Handwriting Recognition, the Second International Conference on Multimodal Interface, the 17th International Conference on Computer Processing of Oriental Languages, the Fifth International Conference on Document Analysis and Recognition, and the Seventh International Conference on Neural Information Processing. He was the workshop co-chair of the Third International Workshop on Document Analysis Systems, the First IEEE International Workshop on Biologically Motivated Computer Vision, the First International Workshop on Pattern Recognition with Support Vector Machines, the Second International Workshop on Biologically Motivated Computer Vision, the 6th IEEE International Conference on Automatic Face and Gesture Recognition. He served on the program committees of several well-known international conferences. He is a fellow of IAPR, a senior member of the IEEE Computer Society and a life member of the Korea Information Science Society and the Oriental Languages Computer Society. His research interests include pattern recognition, computer vision and neural networks. He has more than 200 publications on these areas in international journals and conference proceedings, and authored ten books.