

RESEARCH

Open Access



Visual contour tracking based on inner-contour model particle filter under complex background

Songxiao Cao^{1*}  and Xuanyin Wang²

Abstract

In this paper, a novel particle filter-based visual contour tracking method is proposed, which uses inner-contour model to track contour object under complex background. The purpose is to achieve effectiveness and robustness against complex background. To that end, the proposed method first utilized Sobel edge detector to detect the edge information along the normal line of the contour. Then, it sampled the inner part of the normal line to get the local color information, which was then combined with the edge information to construct new normal line likelihood. After that, all the inner color information was used to construct global color likelihood. Finally, the edge information, local color information, and global color information were fused into new observation likelihood. Experimental results showed that the proposed method was robust for contours tracking under complex background, and it was also computationally efficient and can run in real-time completely.

Keywords: Contour tracking, Inner-contour model, B-spline, Particle filter

1 Introduction

Nowadays, the application of visual object tracking is becoming more and more important in many fields, such as surveillance, robots, and human-computer interfaces [1]. However, it is still a challenging task to achieve reliable tracking due to cluttered background, occlusion, and different illuminations. To overcome the above-mentioned challenges and achieve robust tracking, a lot of tracking methods have been published during last two decades. Most of these tracking methods use rectangle or other rigid shapes to represent the target, which lose detailed shape and edge information. Furthermore, rectangle or other rigid shapes contain some background pixels outside the real target region, which will reduce the robustness of tracking. To overcome this problem, some researchers use contours to represent deformable targets [2–5].

1.1 Related work

Most contour tracking methods can be grouped into two categories, parametric active contour [6–8] and geometric active contour [5, 9], with different representations of contour curves. In the former, the contour was approximated by an explicit parametric model, typically using a set of control points and B-splines. In the second case, the contour was typically represented by an implicit function, as in the level set method. In general, the parametric contour methods were more efficient, and were thus more suitable for real-time contour tracking.

In order to track contours with non-Gaussian and nonlinear state densities in cluttered video sequences, Isard and Blake [6] introduced the CONDENSATION algorithm. They used B-spline to represent object contours, and particle filters to track the curve parameters given noisy observations. However, in this work, a very simple measurement term was used. Therefore, this method had difficulties in dealing with complex background clutters. To improve the performance under complex background, several methods were proposed. Li and Zhang [3] proposed Unscented Kalman Particle Filter (UKPF) to construct a new observation model, in which Kalman filter and unscented particle filter were

* Correspondence: caosongxiao@cjl.u.edu.cn

¹College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China

Full list of author information is available at the end of the article

used to adopt sub-optimal proposal distributions. Chen [10] proposed a Multicue Hidden Markov Model Unscented Particle Filter (MHMM-UPF) for contour detection and tracking based on multiple visual cues in spatial domain and improved the performance by joint probability matching to reduce background clutter. Although these two methods could improve the tracking performance in some simple environments, they still could not deal with the existence of objects that are similar with the target. Another contour tracking strategy is tracking by segmentation, in which the contours are represented with segmentation masks. For instance, Godec [11] proposed a Hough Tracking (HT) algorithm, which is one of the state-of-the-art contour tracking algorithms in recent years. In this algorithm, the object location was determined with the Hough voting technique, and a mask was obtained with the Grab cut algorithm. However, due to its imposing shape constraints, the HT algorithm is very time-consuming, and is not suitable for real-time object tracking.

1.2 Our approach

In contrast to the above methods, a novel multi-feature fusion approach called inner-contour model was proposed in this paper. This method fused the color feature with gradient feature to construct a new observation model in the particle filter framework to realize robust contour tracking in cluttered background. The proposed method first utilized Sobel edge detector to detect the edge information along the normal line of the contour. Then, it sampled the inner part of the normal line to get the local color information, which was combined with the edge information to construct new normal line likelihood. After that, all the inner color information was

used to construct global color likelihood. Finally, the edge information, local color information, and global color information were fused into new observation likelihood. Experimental results showed that the proposed method was robust for contours tracking under complex background, and it was also computationally efficient and can run in real-time completely. The pipeline of the proposed method is shown in Fig. 1.

1.3 Paper organization

Section 2 summarizes the relevant methods for visual contour tracking; Section 3 deals with the proposed inner-contour model for visual contour tracking; Section 4 introduces the experimental results and compares them with UKPF [3], MHMM-UPF [10], and HT [11]; and Section 5 outlines the conclusions and suggestions for future research.

2 Visual contours tracking based on particle filter

This section briefly overviews the main concepts of the related methods discussed in this paper, including the basic formulae of particle filter for visual tracking and the visual contour observation model for contour appearance representation.

2.1 Particle filter

Particle filter is a Monte Carlo approximation to the optimal Bayesian filter. It provides robust tracking of moving objects in cluttered environment, especially in the case of nonlinear and non-Gaussian problems where the interest lies in the detection and tracking of moving objects. It is a probabilistic framework for sequentially estimating the state of the target, recursively calculating the posterior density $p(s_t | z_{1:t})$ of the current object state s_t .

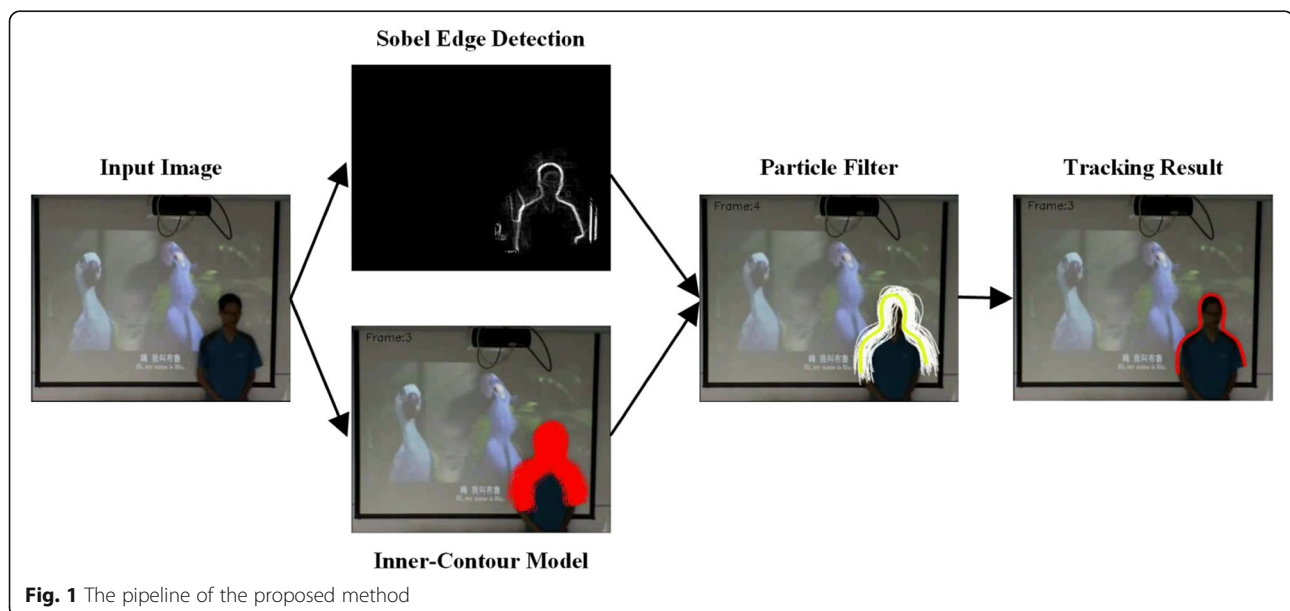


Fig. 1 The pipeline of the proposed method

conditioned on all observations $z_{1:t} = (z_1, z_2, \dots, z_t)$ up to time t . The posterior density $p(s_t | z_{1:t})$ can be obtained recursively in two stages: prediction and update, which are, respectively, written as follows:

$$p(s_t | z_{1:t}) = \frac{p(z_t | s_t) p(s_t | z_{1:t-1})}{p(z_t | z_{1:t-1})} = k_p p(z_t | s_t) p(s_t | z_{1:t-1}) \quad (1)$$

$$p(s_t | z_{1:t-1}) = \int p(s_t | s_{t-1}) p(s_{t-1} | z_{1:t-1}) ds_{t-1} \quad (2)$$

According to formulas (1) and (2), we obtain the following formula:

$$p(s_t | z_{1:t}) = k_p p(z_t | s_t) \int p(s_t | s_{t-1}) p(s_{t-1} | z_{1:t-1}) ds_{t-1} \quad (3)$$

where k_p is a normalizing constant that is independent of s_t , $p(z_t | s_t)$ is the likelihood function, $p(s_t | s_{t-1})$ is the dynamic model, and $p(s_t | z_{1:t-1})$ is the temporal prior over s_t given the prior observations.

The integral in formula (3) has no closed form solution, except in some most basic cases, so the particle filter is used to approximate formula (3) by using a set of weighted particles $\{s_t^{(i)}, w_t^{(i)}\}_{i=1}^n$, and each particle represents a hypothetical state of the object. Under this representation, formula (3) can be approximated as follows:

$$p(s_t | z_{1:t}) \approx k_p p(z_t | s_t) \sum_{i=1}^n w_{t-1}^{(i)} p(s_t^{(i)} | s_{t-1}^{(i)}) \quad (4)$$

where $w_t^{(i)}$ is the weight for particle $s_t^{(i)}$.

To implement a standard PF, a state representation s_t should be identified, in object tracking, which might include locations, scales, and rotations of the object. Moreover, it is necessary to design three distributions: the process dynamical distribution $p(s_t | s_{t-1})$, which describes how the object moves between frames; the proposal distribution $q(s_t | s_{1:t-1}, z_{1:t})$, which is sampled each time the particle distribution updates; and the observation likelihood distribution $p(z_t | s_t)$, which means how the object appears in the video frame. This paper focuses on this likelihood, and will be discussed in detail in the later sections.

The dynamical distribution $p(s_t | s_{t-1})$ can usually be represented as a linear stochastic differential function:

$$s_t = A s_{t-1} + B \omega_{t-1} \quad (5)$$

where A defines the deterministic component of the dynamic model, s_t is the state vector of time t , $\omega_{t-1} \in (0, 1)$ is the system noise, which is usually a uniformly random variable or a multivariate Gaussian random

variable, and B is the propagation distance, indicating the distance the particles can propagate in the next frame.

2.2 Visual contour observation model

In this paper, the visual contour object was modeled as a B-spline curve and was restricted to a shape space proposed by A. Blake and M. Isard [6]. The observation model of the tracking process was based on the model introduced by J. MacCormick [12], and the assumptions and propositions in this reference were adopted as the fundamentals for further derivation. This model can be described briefly as follows.

Giving a candidate contour represented by a B-spline curve, on which a finite number of points are sampled, and then the normal $l_i (i = 1, 2, \dots, m)$ (hereafter called measurement line) to this curve at these points are searched for edge features, all the measurement lines have the same length L . A Sobel edge detector is applied to each measurement line, which is characterized by local maximum. This model made the following hypotheses.

Each feature could correspond to the real edge of the target or clutter feature. The model assumes that all the clutter features are uniformly distributed on the measurement line, and only one edge feature can be detected on each measurement line. The number n of clutter features can be observed on the measurement line with length L obeys a Poisson law with the density $\lambda: bL(n) = e^{-\lambda L} (\lambda L)^n / n!$; there is a fixed possibility q_{01} that the edge feature is not detected; the distribution of the distance between edge feature and contour location of the real object is Gaussian, with zero mean and variance σ^2 .

Based on these hypotheses, the likelihood of the measurement line l_i can be expressed as (see reference [13] for details):

$$p_i(n; z | v = v_n) = e^{-\lambda L} \frac{\lambda^n}{n!} \left(q_{01} + \frac{q_{11}}{\gamma} \sum_{k=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z_k - v_i)^2}{2\sigma^2}} \right) \quad (6)$$

where $q_{11} = 1 - q_{01}$. Given that all the observations of the m measurement line $l_i (i = 1, 2, \dots, m)$ are statistically independent, then the likelihood of the entire contour becomes:

$$p(z | x) = \prod_{i=1}^m p_i(n; z | v = v_i) \quad (7)$$

3 The proposed method

3.1 Inner-contour model

With only using the gradient feature of the input image, the contour tracking algorithms based on general model may have good performance with simple background which does not have much edge features.

However, in highly cluttered environment, the tracker will easily drift to the noise edge feature, which leads to the failure of the tracking process. In order to improve the robustness of contour tracking, it is necessary to introduce other useful features into this model, and naturally fuse all the features to construct new observation likelihood. Inspired by this idea, this paper proposed new observation likelihood which combines the gradient feature and color feature naturally.

To achieve robustness against non-rigidity, rotation, and partial occlusion, the color distribution is a widely used target representation model. In this paper, the color distribution in HSV color space was used to express the color features.

With the contour as the boundary, all the m measurement lines $l_i (i = 1, 2, \dots, m)$ can be separated into two parts, the inner part normal lines and the outer part normal lines, as shown in Fig. 2. The inner part reflects some characteristics of the target, while the outer part is often the clutter background, so using the inner part measurement lines can enhance the observation model during the tracking process.

For a single measurement line l_i , the histogram of the inner part is $h_i = \{h_{i,u}\}_{u=1\dots q}$, while the corresponding reference histogram is $r_i = \{r_{i,u}\}_{u=1\dots q}$, the size of both histograms are $1 \times L/2$, thus the similarity of the two histograms can be measured by the Bhattacharyya distance:

$$d_i = \sqrt{1 - \rho[h_i, r_i]} \quad (8)$$

where $\rho[h_i, r_i] = \sum_{u=1}^q \sqrt{h_{i,u} r_{i,u}}$ is the Bhattacharyya factor. Then, the local color likelihood of l_i is:

$$p_{i,LC}(z|x) = \frac{1}{\sqrt{2\pi}\delta_{LC}} e^{-\frac{d_i^2}{2\delta_{LC}^2}} \quad (9)$$

The new likelihood for the measurement line combine l_i with the gradient feature and local color feature can be written as:

$$p_{l_i} = p_i(n; z|v = v_n) p_{i,LC}(z|x) \quad (10)$$

The likelihood of combining the whole contour with all measurement lines turns to be:

$$p(z|x) = \prod_{i=1}^m p_{l_i} \quad (11)$$

However, only using the local color information of each measurement line cannot provide the overall color distribution information of the contour target. In order to address this problem, all the inner part measurement lines are combined to construct a global color histogram, as shown in Fig. 3.

The histogram of all the m measurement lines $l_i (i = 1, 2, \dots, m)$ is $H = \{H_u\}_{u=1\dots q}$, while the corresponding reference histogram is $Q = \{Q_u\}_{u=1\dots q}$, the size of both histograms is $1 \times L/2$, thus the similarity of the two histograms can be measured by Bhattacharyya distance:

$$D_i = \sqrt{1 - \rho[H_i, Q_i]} \quad (12)$$

where $\rho[H_i, Q_i] = \sum_{u=1}^q \sqrt{H_{i,u} Q_{i,u}}$ is the Bhattacharyya factor. Then, the global color likelihood of the contour is:

$$p_{GC}(z|x) = \frac{1}{\sqrt{2\pi}\delta_{GC}} e^{-\frac{D_i^2}{2\delta_{GC}^2}} \quad (13)$$

The final likelihood of the contour, which combines the gradient information, local color information, and global color information, can be expressed as follows:

$$p(z|x) = p_{GC}(z|x) \prod_{i=1}^m p_{l_i} \quad (14)$$

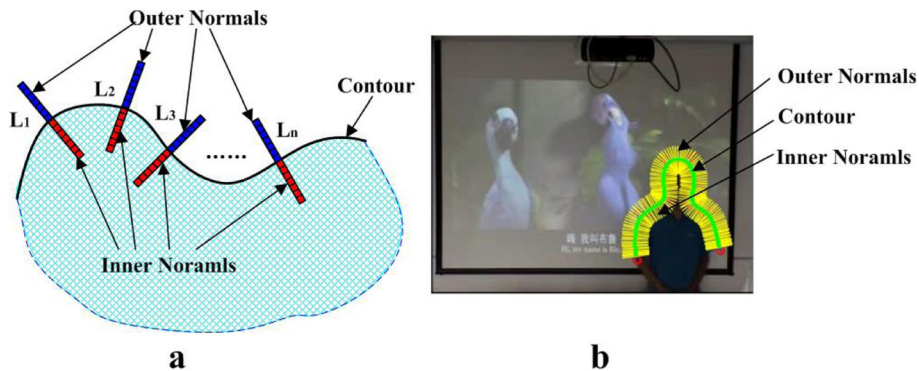


Fig. 2 Inner-contour model

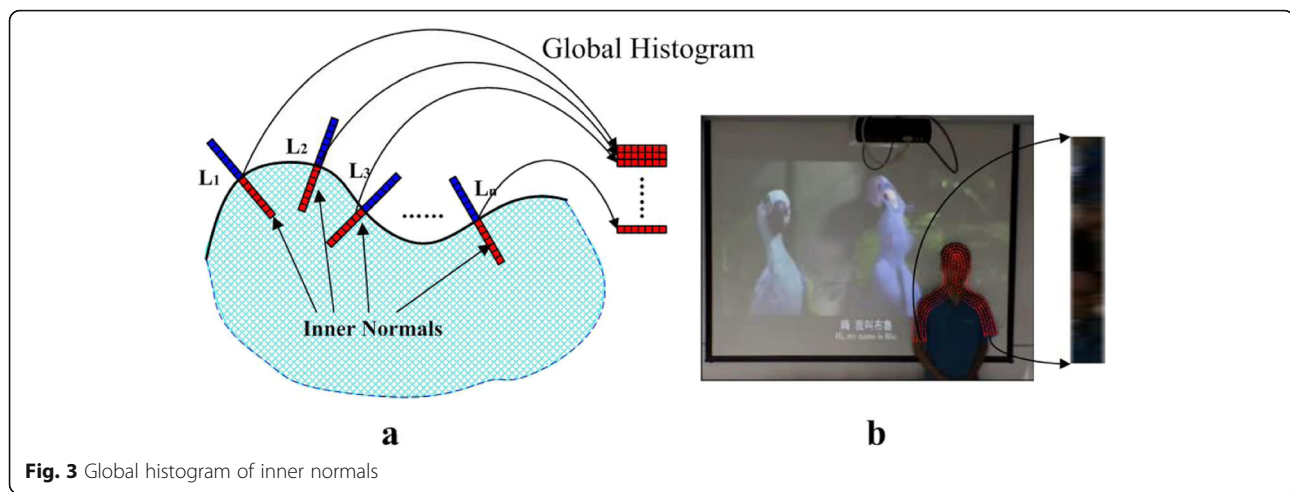


Fig. 3 Global histogram of inner normals

Initializes particles

while frame≠empty **do**

Observation

Calculate the gradient of every normal line of each candidate contour with Sobel

Calculate Bhattacharyya coefficients of every normal line (Eq. (6))

Calculate the local color likelihood of each candidate contour (Eq. (9))

Calculate the global color likelihood of each candidate contour (Eq. (13))

Weigh each sample

Resampling

Calculate the normalized cumulative probabilities

Generate a uniformly distributed random number

Resample each particle and generate new samples

Propagation

Propagate the samples with random walk model (Eq. (5))

Estimation

Estimate the mean state of the set

Calculate the object position and rotation angle

end while

4 Results and discussion

4.1 Experiment setting

To demonstrate the effectiveness and robustness of the proposed tracking scheme, seven different color videos were used in our experiments, six of which were acquired indoors and outdoors with a SONY CCD camera EX-FCB48, and the rest was acquired from the public tracking dataset by Babenko [14]. These videos contained several challenging conditions, such as partial or total occlusion, and similar objects in the background. The contours of the object of basic truth of all tested videos were marked manually frame by frame. For all videos, the target object was manually selected in the first frame.

The control points of three particle filter-based algorithms (UKPF, MHMM-UPF, and the proposed algorithm) were generated randomly according to formula 5, and the dynamic parameters used in the formula were listed in Table 2.

All the algorithms were implemented in C++ using the OpenCV library and run on a 1.8 GHz Pentium Dual-Core CPU, with 2 Gbyte of DDR memory.

The proposed tracking method was compared with UKPF, MHMM-UPF, and HT algorithms, and the

Table 1 Test videos information and parameters used in the experiments

Video	Resolution	Total frames	Number of control points
Hand1	320 × 240	509	13
Hand3	320 × 240	596	15
Body	360 × 288	491	14
Leaf1	320 × 240	147	8
Leaf2	360 × 288	1091	8
Taxi	720 × 576	30	12
David	320 × 240	750	13

Table 2 Parameter values used in the experiments

Video	q_{01}	L	$b(n)$	A	B
Hand1	0.1	20	0.05	(1,1,1)	(8,8,10)
Hand3	0.1	20	0.05	(1,1,1)	(8,8,10)
Body	0.1	20	0.05	(1,1,1)	(8,8,6)
Leaf1	0.1	20	0.05	(1,1,1)	(10,10,15)
Leaf2	0.1	20	0.05	(1,1,1)	(10,10,15)
Taxi	0.1	20	0.05	(1,1,1)	(20,20,15)
David	0.1	20	0.05	(1,1,1)	(10,10,15)

tracking results of each algorithm were marked with different colors to demonstrate the differences. The parameters used in the experiments are shown in Tables 1 and 2.

In the following section, we first presented all the tracking results of the two algorithms in the same tested video with different color curves, and then gave detailed evaluations and comparisons to demonstrate the effectiveness of the algorithms.

4.2 Performance and results overview

For comparison, we first implemented all the algorithms separately, and recorded the related data. Then, we re-drew all the curves in the same video with different colors in order to observe the differences between the algorithms, as shown in Figs. 4, 5, 6, 7, 8, 9, and 10.

The test videos “Hand1” and “Hand3” were used to test the tracking performance in the scenarios of cluttered background, with abundant edge features and obvious affine transformation when the hand was moving. In “Hand1,” due to the existence of collars, cuffs, pockets, and wrinkles in clothes, the edge features of background interference were very noticeable. In “Hand3,” the background was more chaotic and there were many dense edge features, as shown in Fig. 5. Therefore, if only edge feature was used for contour tracking, it would be difficult to achieve stable tracking. As shown in the 285th, 361st, and 373rd frames in Fig. 4, the 291st, 488th, and 514th frames in Fig. 5, the tracking results of the UKPF and MHMM-UPF drifted to the background, while in the previous frames of the tracking result of HT, the contours were not close to the real target, because the HT algorithm needed a progress to segment the target, as shown in the 43rd frame in Fig. 4. When using the proposed inner-contour model, the tracking process was more reliable and robust with the help of local and global color information, as shown by the tracking results of green curves in Figs. 4 and 5. At the same time, the algorithm can respond to the affine deformation of the target in time and accurately in the tracking process. In Fig. 4, the target had an obvious movement close to the camera and then far away from the camera, that is, the process of the target from small to large and then to small, and also the process of rotation with large angle, during all these affine transformations.

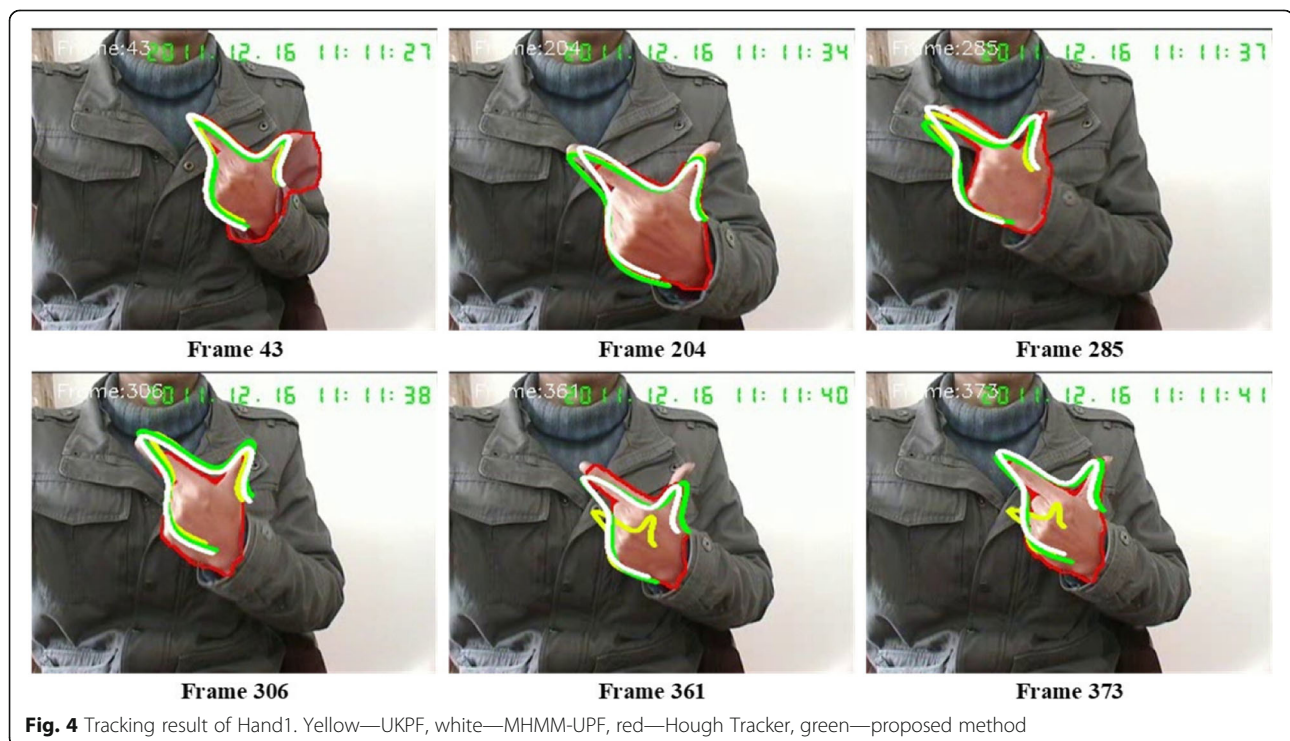
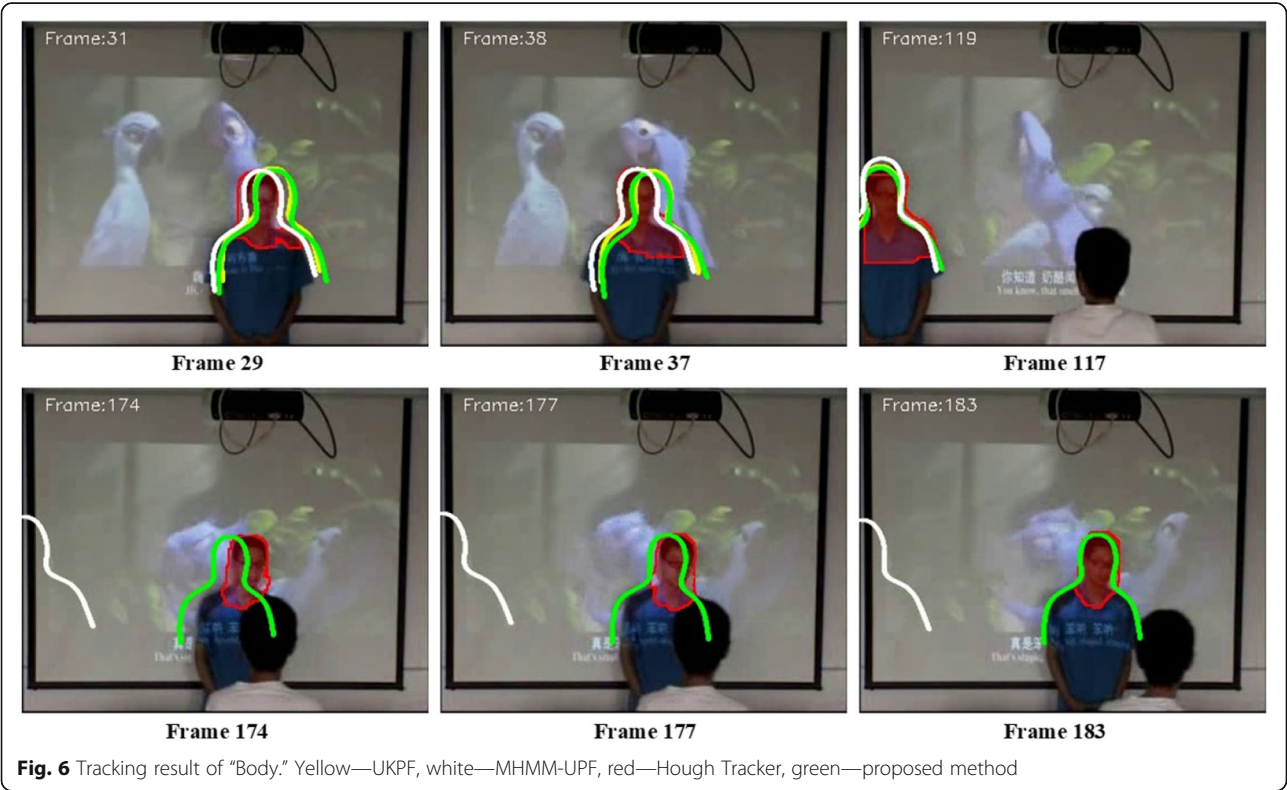
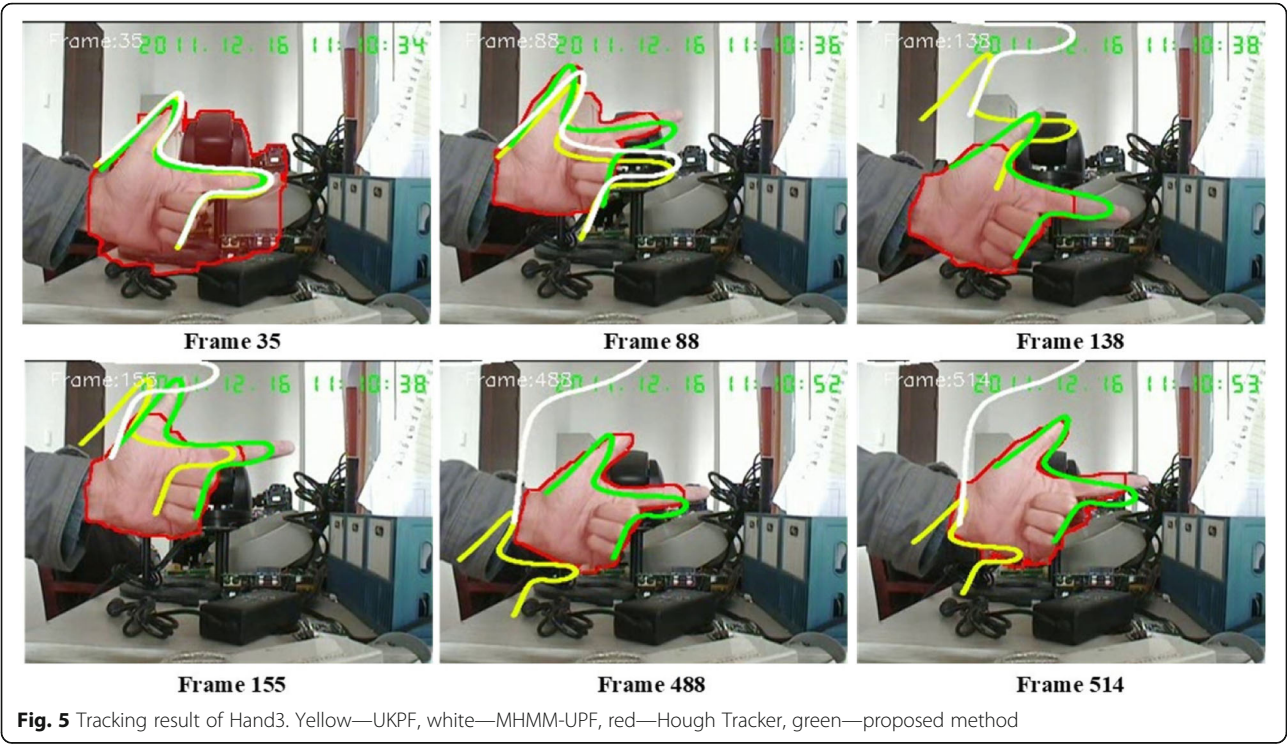


Fig. 4 Tracking result of Hand1. Yellow—UKPF, white—MHMM-UPF, red—Hough Tracker, green—proposed method



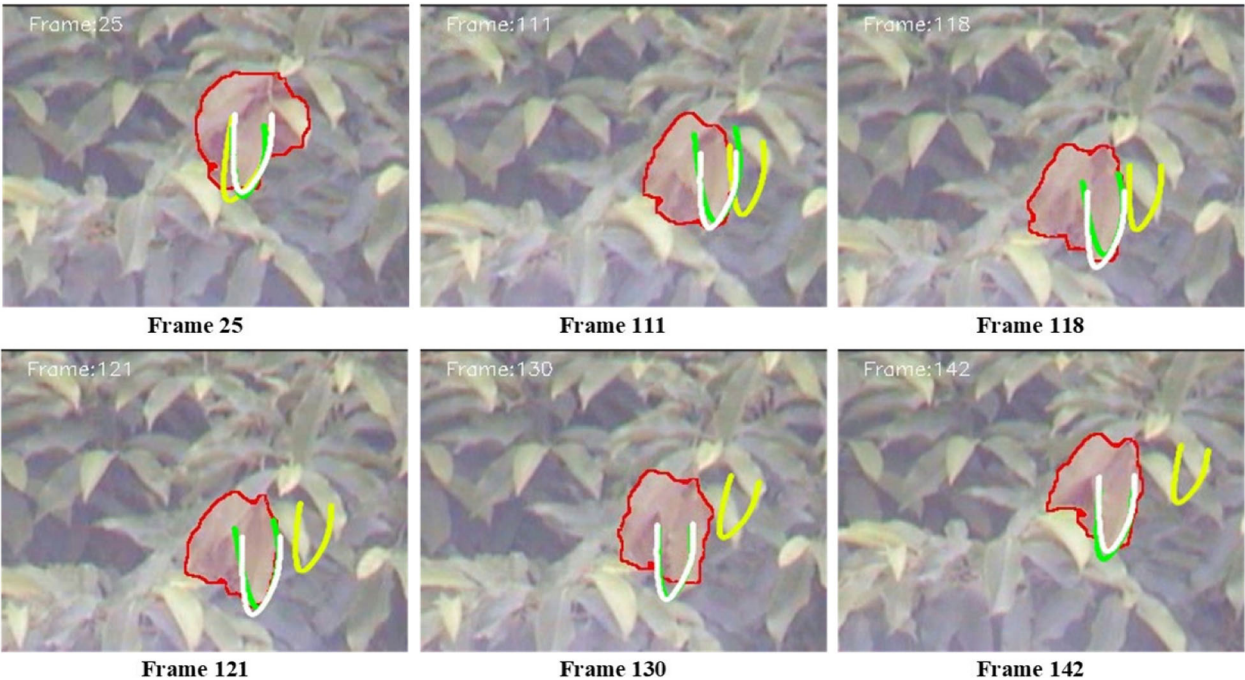


Fig. 7 Tracking result of Leaf1. Yellow—UKPF, white—MHMM-UPF, red—Hough Tracker, green—proposed method

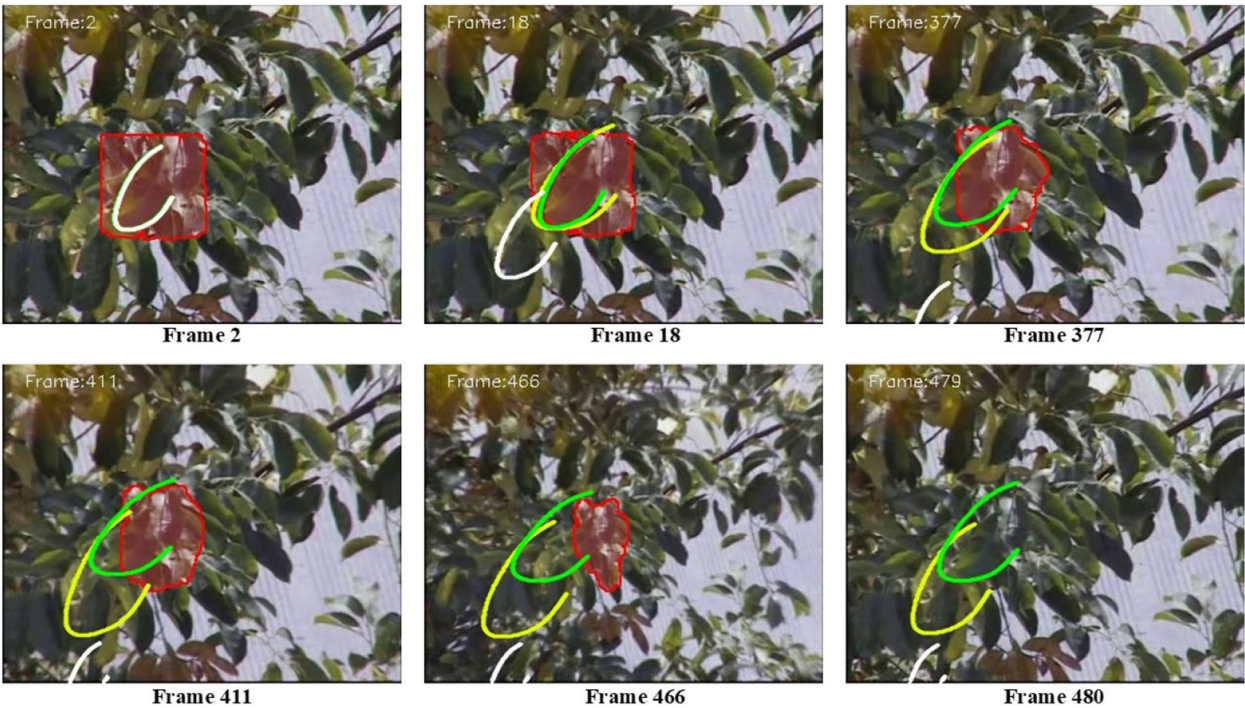


Fig. 8 Tracking result of Leaf2. Yellow—UKPF, white—MHMM-UPF, red—Hough Tracker, green—proposed method

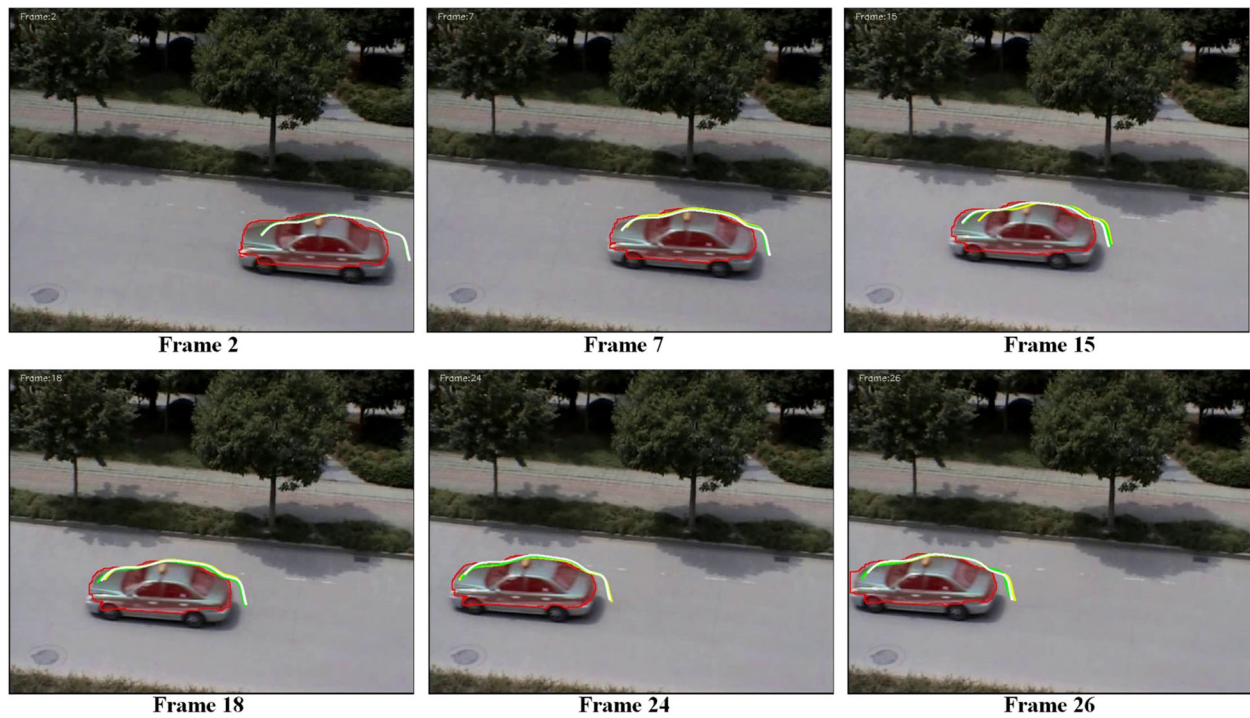


Fig. 9 Tracking result of Taxi. Yellow—UKPF, white—MHMM-UPF, red—Hough Tracker, green—proposed method

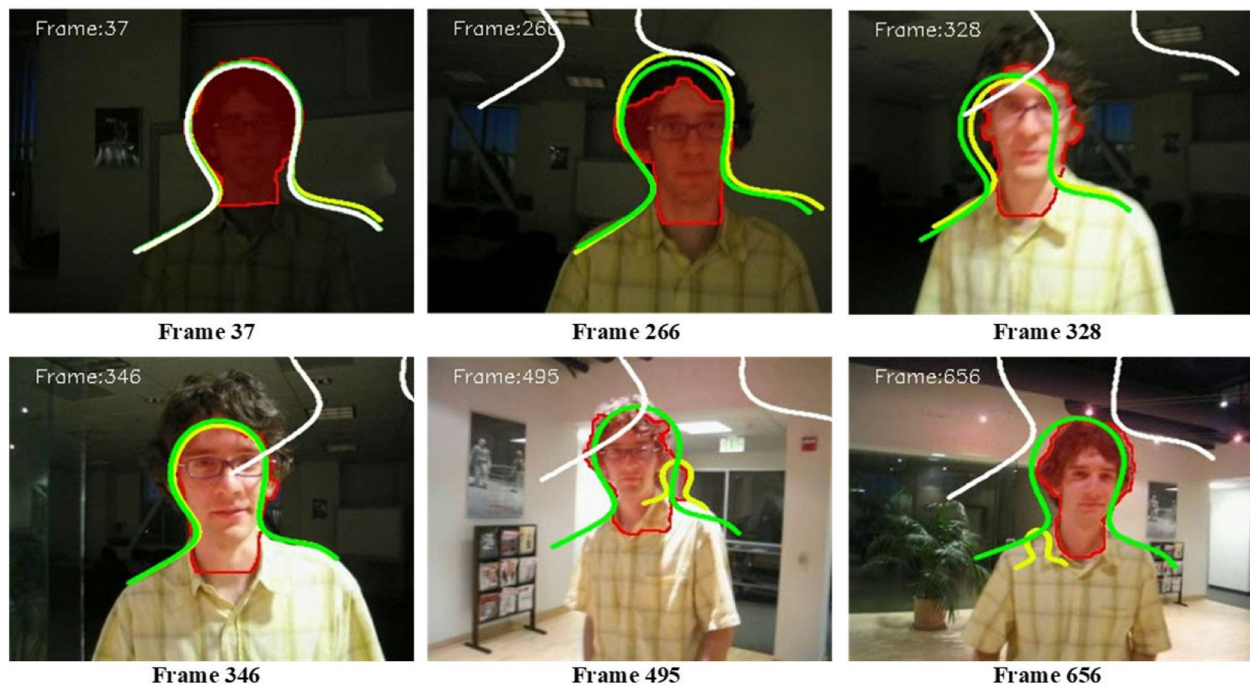


Fig. 10 Tracking result of David. Yellow—UKPF, white—MHMM-UPF, red—Hough Tracker, green—proposed method

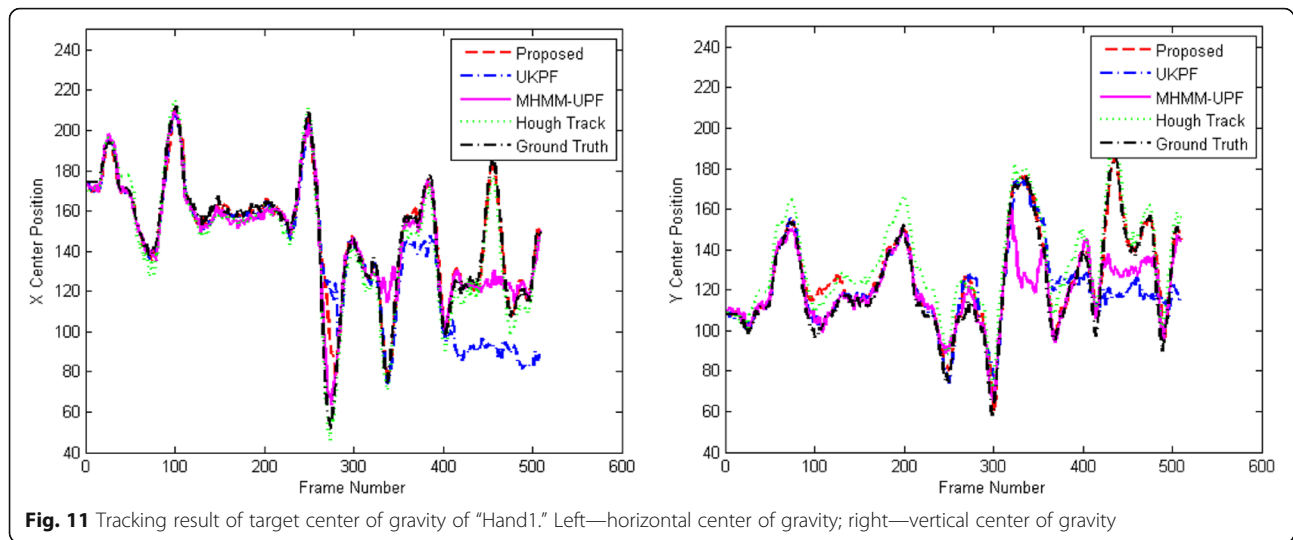


Fig. 11 Tracking result of target center of gravity of “Hand1.” Left—horizontal center of gravity; right—vertical center of gravity

The proposed method always kept good tracking performance, while the UKPF and MHMM-UPF could not.

The test video “Body” was used to test the robustness and effectiveness of the algorithm in the complex dynamic backgrounds with similar target interference and occlusion. In the “Body” video, a cartoon movie was always played on the projection screen of the background; therefore, the edge features and color features of the background were changing simultaneously, and another person walked in front of the target, as shown in Fig. 6. When only edge feature was used to track human contours, it was susceptible to dynamic background and similar target occlusions, resulting in tracking failures, as shown in the 174th, 177th, and 183rd frames of Fig. 6.

When combined with the inner-contour information, because there were significant differences between the clothes worn by the two people, it was easier to distinguish two different targets, even in the case of partial occlusion, so that the target could be accurately tracked. Although the tracking contours of HT were not as accurate as the proposed method, the accuracy of the target center positions was the best among all the test algorithms.

The test videos “Leaf1” and “Leaf 2” were used to test the tracking performance when there were a large number of similar targets in the background. These two sets of test videos were the most challenging, because the target to be tracked was a bunch of completely similar

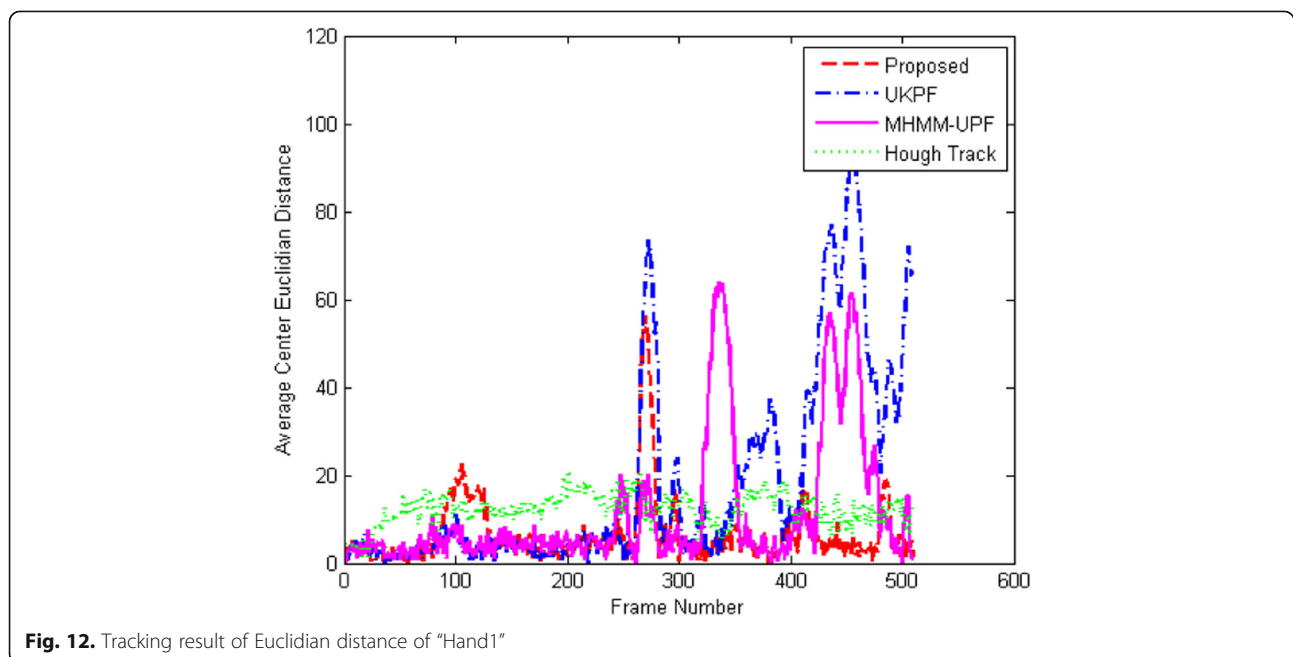


Fig. 12 Tracking result of Euclidian distance of “Hand1”

leaves, which flutter in the wind, and the system dynamic model was more complicated, thus it was difficult to achieve stable tracking. As shown in the 118th, 121st, and 130th frames of Fig. 7, and the 377th, 411th, and 466th frames in Fig. 8, if only the edge information was used, it was easy to track other leaves during intense movement and deformation when the leaves were blown by the wind. When the inner-contour model was adopted, it had better stability under the same conditions, which can not only accurately track the position of the target, but also make correct affine changes with the swing and deformation of the leaves. The HT tracker did not perform well in these two test videos because it was hard to segment the real target among so many similar leaves around the target.

The test videos “Taxi” was used to test the tracking performance when the target moved quickly in a simple background, as shown in Fig. 9. In this test, all the algorithms tracked the target well, and the HT tracker performed better than the other three particle filter-based algorithms, because in this simple background, it was relatively easy to segment the real target precisely.

The test video “David” was a well-known video sequence, which was used to track algorithms to test the robustness when the environment luminance changed significantly. The experimental results showed that, as shown in Fig. 10, the proposed method and HT can track the person well from the first frame to the end, while UKPF and MHMM-UPF lost the targets during the luminance changing.

In order to evaluate the tracking performance differences of various algorithms more objectively and accurately, the Euclidean distance of the center of gravity coordinates, the Euclidean distance of the control points and the algorithm time were used to evaluate the difference of different tracking algorithms in this paper. The center of gravity coordinate Euclidean distance represented the Euclidean distance between the center of gravity of the target contour calculated by the algorithm and the ground truth, which was used to characterize the accuracy of the overall position of the contour. The average Euclidean distance of the control points represented the average value of the Euclidean distance between each control point of the target contour calculated by the algorithm and the ground truth, which was used to characterize the track location accuracy. The algorithm time was a measure of the time spent by each algorithm on tracking different targets. It was used to characterize the execution efficiency and real-time performance of the algorithm.

As shown in Fig. 11, the left side represents the difference between the horizontal position and the ground truth of the center of gravity of the tracking

target contour calculated by each algorithm of the “Hand1” video, and the right side represents the difference between the vertical position and the ground truth. Figure 12 shows the Euclidean distance between the center of gravity of the control points and the ground truth of each frame. From these figures, it can be seen that the proposed algorithm performed best among all the tested algorithms.

Table 3 Average Euclidean distance using different particle numbers

Video	Number of particles	HT	UKPF	MHMM-UPF	Proposed
Hand1	100	12.15	21.86	18.43	9.34
	150		20.47	16.94	9.12
	200		18.78	15.41	8.67
	250		19.01	15.32	9.01
	300		18.95	15.33	8.98
Hand3	100	15.78	32.19	24.66	11.63
	150		30.84	22.90	10.96
	200		25.72	22.37	10.39
	250		26.21	22.17	10.15
	300		25.34	22.10	10.33
Body	100	4.12	12.77	8.12	5.54
	150		11.23	7.95	5.11
	200		10.87	7.84	4.82
	250		10.96	7.89	4.91
	300		11.02	7.83	4.90
Leaf1	100	5.12	12.07	9.34	4.25
	150		11.45	9.06	4.10
	200		10.56	8.76	4.04
	250		10.55	8.12	4.14
	300		10.81	8.23	4.12
Leaf2	100	15.74	40.33	33.59	8.67
	150		35.54	29.96	7.27
	200		30.78	29.07	7.03
	250		30.47	29.24	7.13
	300		29.87	30.63	7.09
Taxi	100	2.43	7.67	7.12	7.16
	150		7.34	6.87	6.76
	200		7.23	6.65	6.51
	250		7.21	6.53	6.59
	300		7.22	6.61	6.55
David	100	10.44	25.45	27.29	11.67
	150		23.64	26.43	10.83
	200		21.98	25.39	10.01
	250		22.01	25.34	10.05
	300		22.03	25.38	10.04

Table 4 Time consumption (ms) of tested algorithms

Video	HT	UKPF	MHMM-UPF	Proposed
Hand1	239	32.71	34.51	33.65
Hand3	178	39.53	42.29	40.26
Body	164	34.21	36.26	36.61
Leaf1	161	16.98	19.45	18.04
Leaf2	170	18.65	20.54	20.08
Taxi	162	44.43	48.29	40.54
David	178	34.15	37.34	35.84

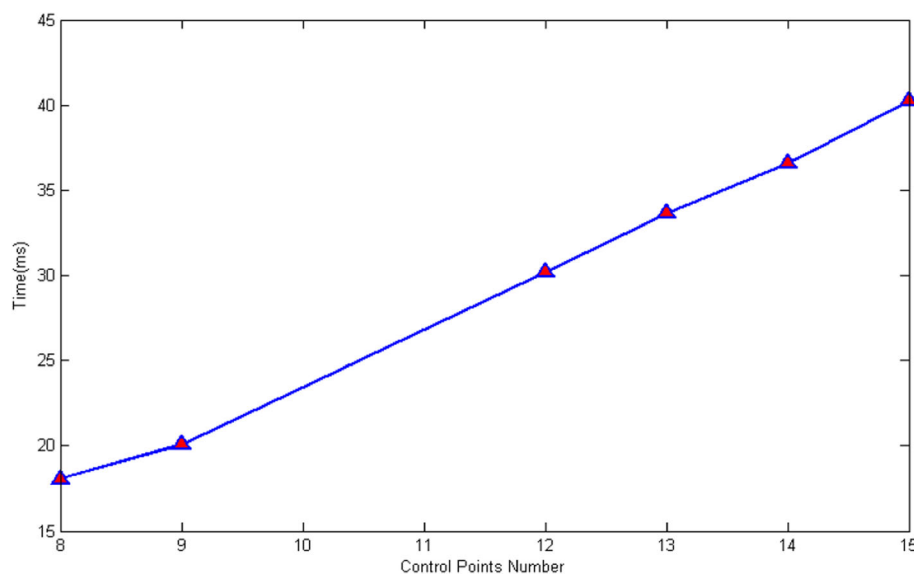
For performance evaluation and comparison, all the three PF-based algorithms (UKPF, MHMM-UPF, and the proposed algorithm) were tested with different numbers of particles, including 100, 150, 200, 250, and 300, and the tracking results are shown in Table 3. It can be found that the tracking performance improves with the increase of the number of particles, and reaches the top when the number of particles is larger than 200.

The experiment result of average Euclidian distance and time consuming of all the test videos can be seen in Table 3. In terms of average Euclidean distance, the performance of the proposed algorithm and HT tracker was better than that of UKPF and MHMM-UPF, and the average Euclidean distance of the proposed algorithm was the smallest in test videos “Hand1,” “Hand3,” “Leaf1,” and “Leaf2,” while the performance of HT tracker was better in test videos “Body,” “Taxi,” and “David” than that of the proposed one.

The difference of Euclidean distance between the four algorithms in the test video “Leaf2” was the most obvious, followed by the videos “Hand1” and “Hand3,”

and the “Taxi” tracking was the least. The reason of this phenomenon was, in the leaf contour tracking video, there were a lot of edge information and some similar targets. Meanwhile, the (Pan-Tilt-Zoom) PTZ camera took intense motion in three degrees of freedom, so when only edge information was used, the calculated weight of each particle may be close in any position, so that the effect of resampling was not obvious, and finally, the target position generated a relatively large deviation. When the color information was introduced, the small difference between the target and the surrounding leaves was used to increase the weight of each particle, and the weight of the edge of the background became smaller. After resampling, more particles moved closer to the target leaves; therefore, the tracking position was more accurate. In the “Body” video, there were few distinct edge features around the body. The edge features could well distinguish the target and the background, so the difference between the two algorithms was very small.

For the perspective of algorithm time-consuming (Table 4), the HT tracker was the most time-consuming one, it took much more time than the other three algorithms due to its complex segment algorithm and imposing shape constraints. Among the other three algorithms except HT, the proposed algorithm was more time-consuming than UKPF, but less than MHMM-UPF; however the difference was very small, about 1 to 2 ms. At the same time, it can be found that the time consuming of the algorithm increased with the increase of the number of control points. The reason was that if only the time

**Fig. 13** The relationship between time consuming and the number of control points of test video “Hand3”

consumption of coordinate transformation of the control points was considered, the algorithm time would be slightly affected by the number of control points due to the representation of the shape space, and the main influencing factor was the time to draw the B-spline curve. In the case of 50 equal divisions between every two control points, the time to draw the B-spline curve increased approximately linearly with the number of control points, as shown in Fig. 13. In the case of 18 control points (using test video “Hand3”), the average algorithm time was 40.26 ms, and the algorithm time could basically meet the real-time requirements, and the real-time performance of the algorithm could be further improved by using some fast spatial-temporal mechanisms and algorithms, as mentioned in literatures [15–17].

5 Conclusions

This paper has presented a method of visual contour tracking based on particle filter for inner-contour model under complex background. This novel method fused the gradient feature, local color feature, and global color feature naturally to achieve robust contour tracking in cluttered environment. Specifically, the proposed algorithm first used Sobel edge detector to detect the edge information along the normal lines of the contour, and then sampled the inner part of the normal lines to get the local color information, which was combined with the edge information to construct new normal line likelihood. After that, all the inner color information was used to construct global color likelihood. Finally, the edge information, local color information and global color information are fused together as new observation likelihood. The experimental results demonstrated that, compared with gradient-only feature method, the proposed algorithm was effective and robust in dealing with cluttered background, and it was also computationally efficient and could run completely in real time.

The proposed algorithm was inspired by the gradient-only contour tracking method (UKPF) and achieved better results, and it would be helpful for other tracking methods that needed to consider multi-cues fusing in cluttered background.

Abbreviations

HT: Hough Tracking; MHMM-UPF: Multicue Hidden Markov Model Unscented Particle Filter; UKPF: Unscented Kalman Particle Filter

Acknowledgements

The most heartfelt gratitude to Ke Xiang for his helpful discussion and feedback on the algorithm, as well as Fangge Lu's for proofreading of English writing.

Authors' contributions

SC implemented the core algorithm, designed all the experiments, addressed the resulting data, and drafted the manuscript. XW participated in the design and construction of the inner-contour model and helped draft the manuscript. All authors have read and approved the final manuscript.

Funding

None

Availability of data and materials

None

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Competing interests

The authors declare that they have no competing interests.

Author details

¹College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China. ²State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China.

Received: 19 February 2019 Accepted: 19 November 2019

Published online: 10 December 2019

References

1. A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey. *ACM Comput. Surv.* **38**(4), 13 (2006)
2. M. Isard, A. Blake, CONDENSATION—conditional density propagation for visual tracking. *Int. J. Comput. Vis.* **29**(1), 5–28 (1998)
3. P. Li, T. Zhang, A.E.C. Pece, Visual contour tracking based on particle filters. *Image Vis. Comput.* **21**(1), 111–123 (2003)
4. A.W.M. Smeulders et al., Visual tracking: an experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1442–1468 (2014)
5. P. Lv et al., Multiple cues-based active contours for target contour tracking under sophisticated background. *Vis. Comput.* **33**(9), 1103–1119 (2016)
6. Isard, M. and A. Blake. Contour tracking by stochastic propagation of conditional density. in *European Conference on Computer Vision*. 1996.
7. N. Peterfreund, Robust tracking of position and velocity with Kalman snakes. *Pami* **22**(6), 564–569 (1999)
8. F.Y. Shih, Z. Kai, Locating object contours in complex background using improved snakes. *Computer Vision & Image Understanding* **105**(2), 93–98 (2007)
9. Chockalingam, P., N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. in *IEEE International Conference on Computer Vision*. 2009.
10. C. Yunqiang, R. Yong, T.S. Huang, Multicue HMM-UKF for real-time contour tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(9), 1525–1529 (2006)
11. M. Godec, P.M. Roth, H. Bischof, Hough-based tracking of non-rigid objects. *Comput. Vis. Image Underst.* **117**(10), 1245–1256 (2013)
12. MacCormick, J. and A. Blake. A probabilistic exclusion principle for tracking multiple objects. in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999.
13. J. Maccormick, *Stochastic algorithms for visual tracking* (2002)
14. Babenko, B., M.H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2009.
15. Yan, C., et al., Cross-modality bridging and knowledge transferring for image understanding. *IEEE Transactions on Multimedia*, 2019: p. 1–1.
16. Yan, C., et al., STAT: spatial-temporal attention mechanism for video captioning. *IEEE Transactions on Multimedia*, 2019: p. 1–1.
17. C. Yan et al., A fast Uyghur text detector for complex background images. *IEEE Transactions on Multimedia* **20**(12), 3389–3398 (2018)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.