

Efficient silhouette-based contour tracking using local information

Ajoy Mondal · Susmita Ghosh · Ashish Ghosh

Published online: 11 December 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this article, we present an algorithm that can efficiently track the contour extracted from silhouette of the moving object of a given video sequence using local neighborhood information and fuzzy k-nearest-neighbor classifier. To classify each unlabeled sample in the *target* frame, instead of considering the whole training set, a subset of it is considered depending on the amount of motion of the object between immediate previous two consecutive frames. This technique makes the classification process faster and may increase the classification accuracy. Classification of the unlabeled samples in the *target* frame provides object (silhouette of the object) and background (non-object) regions. Transition pixels from the non-object region to the object silhouette and vice versa are treated as the boundary or contour pixels of the object. Contour or boundary of the object is extracted by connecting the boundary pixels and the object is tracked with this contour in the *target* frame. We show a realization of the proposed method and demonstrate it on eight benchmark video sequences. The effectiveness of the proposed method is established by comparing it with six state of the art contour tracking techniques, both qualitatively and quantitatively.

Keywords Fuzzy k-nearest-neighbor classifier · Boundary pixels · Contour tracking · Motion

1 Introduction

Moving object tracking from a given video sequence has a great deal of interest in computer vision and has been an active research area for the last few decades (Yilmaz et al. 2006; Bovic 2000; Tekalp 1995). Moving object tracking can be defined as the problem of estimating the trajectory of the object in the image plane as it moves around a scene. Tracking is widely used in automated surveillance (Yilmaz et al. 2006; Bovic 2000; Tekalp 1995), traffic monitoring (Yilmaz et al. 2006; Tang et al. 2007), vehicle navigation (Yilmaz et al. 2006), event detection (Bovic 2000; Tekalp 1995), dynamic scene analysis (Bovic 2000; Tekalp 1995), activity-based human recognition (Bovic 2000; Tekalp 1995), etc. Object tracking is a challenging task because of the presence of noise in the image frames, complex object motion, partial or full object occlusion, background clutter, illumination variation in scene, complex object shapes (Yilmaz et al. 2006; Tang et al. 2007), etc. In the literature, there exist many object tracking techniques (Yilmaz et al. 2006) to handle these problems.

An object can be defined by its boundary. Boundary of the object can be represented by a contour. Contour is used to track objects having irregular shapes, which change their shapes and sizes during movement. Kass et al. (1988) proposed a snake model to represent a contour of an object and is used to track the object. Active contour model was used to track objects in Nguyen et al. (2002), Yilmaz et al. (2004), and Allili and Ziou (2008). Several modifications (Cohen 1991; Lefèvre et al. 2002; Xu et al. 1998; Luo et al. 2003; Chiverton et al. 2012; Aitfares et al. 2013; Ning et al. 2013)

Communicated by V. Loia.

A. Mondal · A. Ghosh (✉)
Machine Intelligence Unit, Indian Statistical Institute,
Kolkata 700 108, India
e-mail: ash@isical.ac.in

A. Mondal
e-mail: ajoy.mondal83@gmail.com

S. Ghosh
Department of Computer Science and Engineering, Jadavpur
University, Kolkata 700 032, India
e-mail: susmitaghoshju@gmail.com

of snake active contour model are proposed in the literature. All these methods are parametric.

Due to the problems for estimating parameters, in the literature, non-parametric approaches, geometric active contour models (Caselles 1995; Osher and Sethian 1998; Malladi et al. 1995; Caselles et al. 1997; Paragios et al. 2004; Brox et al. 2010) were developed to track objects. Paragios et al. (2004) proposed an edge-driven bidirectional geometric flow to extract boundary of an object; whereas, Brox et al. (2010) proposed the use of color, texture, and motion in a level set-based segmentation and tracking method. Disadvantages of all parametric and non-parametric active contour methods are (i) the initial contour must in general be close to the true boundary of object, otherwise it is likely to converge to wrong result, (ii) active contour methods have difficulties progressing into boundary concavities.

Recently, tracking is being viewed as a classification problem (Avidan 2004) and a classifier is trained to distinguish the object from the background. Instead of trying to build a complex model to describe the object, classification-based trackers seek a decision boundary that can separate the object and the background. Complex object dynamics is also not a problem since tracking becomes an on-line detection problem which makes no assumptions of where the object could be. To adapt object appearance changes, tracking methods update the decision boundary instead of the object appearance model (Tang et al. 2007).

Several techniques (Avidan 2004; Collins et al. 2005; Grabner et al. 2006; Avidan 2007; Santner et al. 2010; Babenko et al. 2011) using classifiers were proposed to track objects. In all classifier-based object tracking methods, an object in the *candidate* frame is represented by a rectangle or an ellipse or a circle. If the shape of the object is irregular and changes during movement then some parts of background are treated as part of the object in the *candidate* frame which degrades the classification accuracy and also the tracking accuracy in the *target* frame. If the size of the object is large, then number of labeled samples and unlabeled samples are also more which increases the complexity of the classifier-based tracking algorithms.

In this article, we present an algorithm that can efficiently track the contour extracted from silhouette of the moving object from a given video sequence using local neighborhood information and fuzzy k -nearest-neighbor classifier. It assigns class membership to an unlabeled sample rather than assigning the sample to a particular class. Advantage is that no arbitrary assignments are needed. The algorithm is able to give better classification if overlapping is present in the data set. In principle any classifier which learns in one shot (no incremental learning) can be used. In order to take care of the fuzziness present in the image pixels fuzzy classifier is used.

In the *candidate* frame, the moving object is represented by its silhouette as a candidate model opposed to a rectangular (or an elliptical or a circular) tracker as in conventional tracking algorithms. For classification, the features, namely 8-bin Histograms of Oriented Gradients (HOG) along with R , G , and B corresponding to each pixel of the image frame, are extracted. Training set (labeled samples) is manually generated from the *candidate* frame, and the test set (unlabeled samples) is automatically generated from the *target* frame. In the present work, to classify each unlabeled sample in the *target* frame, instead of considering the whole training set, a subset of training set is considered. The training subset is generated with the labeled samples within the spatial neighborhood (window) at the corresponding location in the *candidate* frame of an unlabeled sample in the *target* frame. The neighborhood (window size) is related to the amount of motion between immediate previous two consecutive frames. This technique makes the classification process faster and may increase the classification accuracy. After classification of unlabeled samples in the *target* frame, we have two regions: object silhouette and background. Transition pixels from the non-object (background) region to the object silhouette and vice versa are treated as the boundary or contour pixels of the object. Connecting the boundary pixels, contour, or boundary of the object is extracted in the *target* frame. Hence, the object is tracked with its contour or boundary in the *target* frame.

To evaluate the performance of the proposed algorithm, the tracking results are compared with ground truth contour (boundary) tracker. Here, area-based metric [Average Tracking Accuracy (ATA)] and F-measure are used to measure the performance. Tracking results obtained by the proposed method are compared with those of snake active contour tracking (Snake, Kass et al. 1988), gradient vector flow fast geometric active contour tracking (GVF, Paragios et al. 2004), level set-based active contour tracking (Level set Brox et al. 2010), automatic bootstrapping and tracking of object contours (ABTOC, Chiverton et al. 2012), hybrid region and interest points-based active contour for object tracking (HRIPBACOT, Aitfares et al. 2013), joint registration and active contour segmentation for object tracking (JRACSOT, Ning et al. 2013), and the contour tracking algorithm (fuzzy k -NN). It is found that the proposed method provides better results for seven among eight different video sequences. It is also found that the proposed technique takes less computational time to track the contour for seven video sequences than all other methods.

Organization of the article is as follows. Section 2 provides a description for selecting two components: HOG and fuzzy k -NN classifier. Section 3 describes the proposed method for efficient silhouette-based contour tracking. Experimental results are presented in Sect. 4 and finally, conclusive remarks are provided in Sect. 6.

2 Background

Here, we highlight the usefulness of HOG and fuzzy k-NN for the proposed work.

2.1 HOG

HOG have been widely used as features for hand gesture recognition (Freeman and Roth 1995), face recognition (Levi and Weiss 2004), human detection (Dalal and Triggs 2005), pedestrian detection (Suard et al. 2006), and object tracking (Avidan 2007). It captures the local object appearance, shape and is invariant to local geometric and photometric *transformations: translation* or rotation (Dalal and Triggs 2005). It is also largely invariant to global illumination changes (Levi and Weiss 2004). To extract these features, an image frame is divided into overlapping small spatial regions called cells. For each cell, accumulate a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. For better illumination invariance, shadowing, etc., cells histograms are locally normalized. All cells in the block are normalized. The normalized blocks are referred to as HOG descriptors.

As HOG has the above-mentioned advantages, it has been used as features for tracking moving objects in the present work. Each pixel of the image frame is represented by 11 dimensional feature vector (namely R , G , B , and 8-bin HOG).

2.2 Fuzzy k-nearest neighbor classifier

Fuzzy k-NN (Keller et al. 1985) assigns class membership for an unlabeled sample rather than assigning it to a particular class (not like k-NN and some other hard classifiers). It is a single-shot learner and reduces computational time for classification of unlabeled samples compared to iterative learning-based classifiers (such as multi-layer Perceptron, Radial Basis Function network, etc.). In the present work, real-life data sets are used in which samples of two different classes (object and background) are overlapped. Figure 1 shows a 3D plot of labeled samples in different sub (feature)-spaces. From this figure, it is clear that samples of two classes are very much overlapped in subspaces. It is claimed that samples are overlapped in the feature space also. Thus it is expected that any fuzzy classifier will provide better classification than any non-fuzzy classifier. In the proposed work, for classifying different test patterns (unlabeled samples), different training subsets are considered. This technique increases the computational cost much in case of iterative learning-based classifiers. Since, fuzzy k-NN is single-shot learner, this technique reduces the computational time for classification. For the above-mentioned advantages, fuzzy k-NN clas-

sifier is used to separate the object from the background in the proposed work.

3 Proposed method for efficient silhouette-based contour tracking

In this article, we present an algorithm that can efficiently track the contour extracted from silhouette of the moving object from a given video sequence using local neighborhood information and fuzzy k-NN. We have considered the silhouette of the moving object as a candidate model in the *candidate* frame opposed to a rectangular (or an elliptical or a circular) tracker as in conventional tracking algorithm. A fuzzy k-NN classifier is used to distinguish the object from the background in the *target* frame. The features namely, 8-bin HOG, R , G , and B corresponding to each pixel of the image frame are extracted for classification. Training set (labeled samples) is manually generated from the *candidate* frame, and the test set (unlabeled samples) is automatically generated from the *target* frame. In the present work, instead of considering the whole training samples, a subset of training set is considered to classify an unlabeled sample in the *target* frame. A heuristic approach based on the motion between the immediate previous two frames is proposed to generate the training subset. Classification of unlabeled samples provides two regions: object (silhouette) and background (non-object). Transition pixels from the non-object region to the object silhouette and vice versa are treated as the boundary or contour pixels of the object. The contour or boundary of the object is extracted from its silhouette in the *target* frame. Hence, the object is tracked with its contour or boundary in the *target* frame. Here, we describe the proposed method in an algorithmic form (Algorithm 1).

In the following section, each part of the algorithm is discussed in more detail.

3.1 Acquiring labeled samples

In the present work, object is represented by its silhouette in the *candidate* frame. Figure 2a shows that the object is represented by its silhouette in the *candidate* frame. Figure 2b displays extracted contour or boundary of the *target object* from its silhouette in the *candidate* frame. Labeled samples are manually generated from the *candidate* frame. Pixels within the object silhouette belong to positive (object) class and pixels outside the silhouette of the object but inside the rectangle (which is double in size with respect to height and width of the object) belong to negative (background) class. Figure 3a represents the object silhouette and the rectangle in the *candidate* frame. Here, we assume that possible movement of the object is within that rectangle in the *target* frame (Avidan 2007). Each pixel (labeled sample) is represented by an

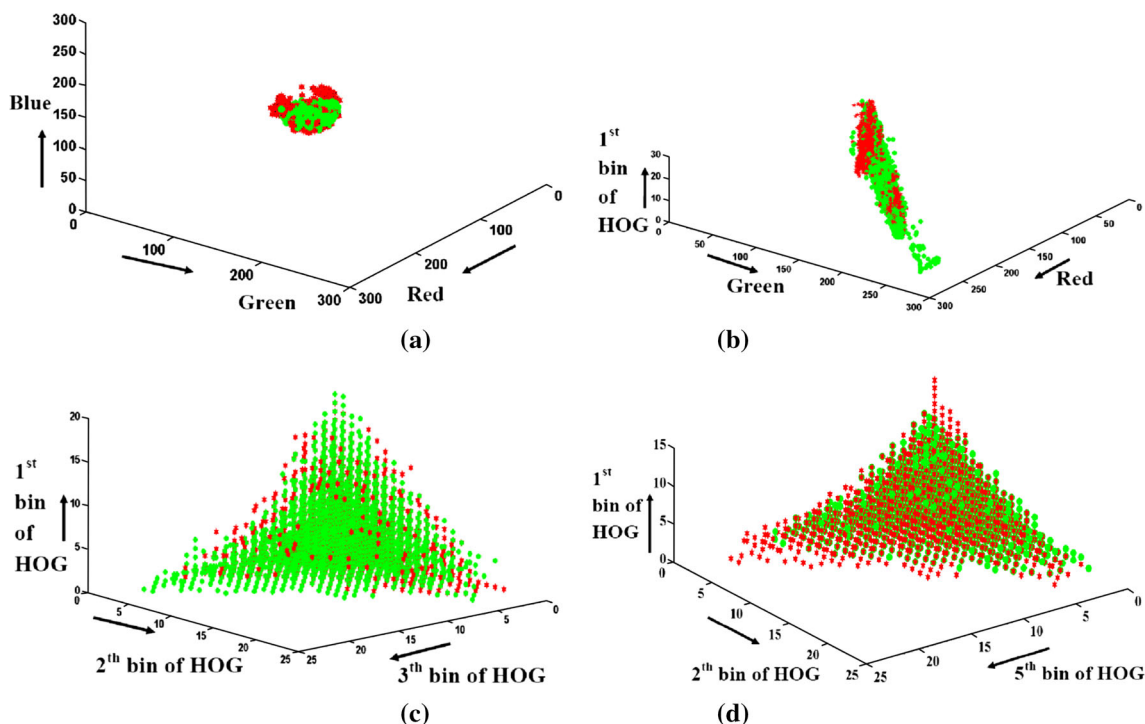


Fig. 1 Labeled samples of two classes are plotted in sub-feature space



Fig. 2 Run-Side2 video frames: **a** object is represented by its silhouette in the candidate frame, **b** extracted object contour from its silhouette in the candidate frame

extracted 11-dimensional feature vector (namely 8-bin HOG along with R , G , and B).

3.2 Generate unlabeled samples

Unlabeled samples are automatically generated from the target frame. Pixels (within the corresponding rectangle of the candidate frame) in the target frame are considered as unlabeled samples. Figure 3b shows the corresponding rectangle of the candidate frame in the target frame. For each pixel, the features 8-bin HOG along with R , G , and B are extracted.

Each unlabeled sample is represented by an extracted 11-dimensional feature vector.

3.3 Rough estimation of motion between two consecutive frames

This method will provide a rough estimate of the amount of motion of an object in two consecutive frames. Depending on it, the training sub-set is generated. Let BM_1 , BM_2 , UM_1 , UM_2 , LM_1 , LM_2 , RM_1 , RM_2 , C_1 , and C_2 be bottom most point, upper most point, left most point, right most

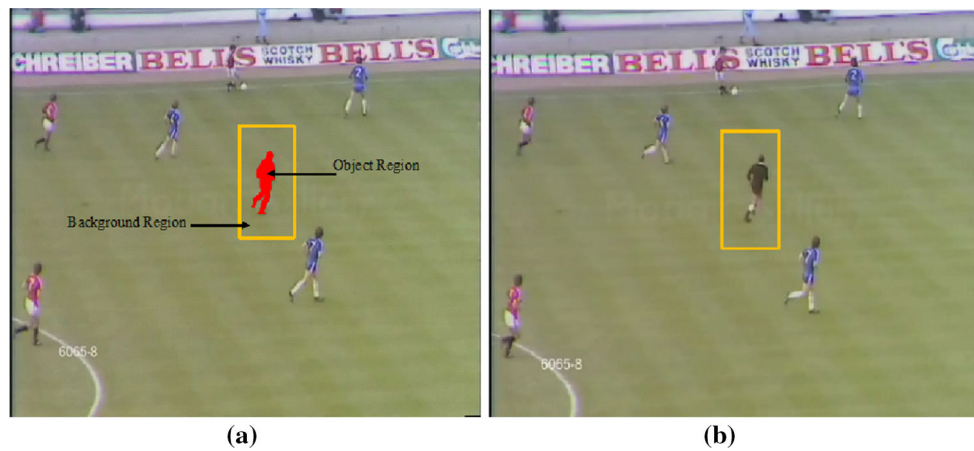


Fig. 3 Run-Side2 video frames: **a** object in the *candidate* frame, **b** corresponding rectangle in the *target* frame

Algorithm 1 Efficient Silhouette-based Contour Tracking using Local Information

- 1: Input: P number of video frames I_1, I_2, \dots, I_P . I_1 and I_2 are the candidate frames in which object information is known. Silhouettes s_1 and s_2 represent the objects in the *candidate* frames or the candidate models, I_1 and I_2 , respectively. r_1 and r_2 represent the extracted contours from the silhouettes
- 2: Output: Silhouettes s_3, \dots, s_P and the extracted contours r_3, \dots, r_P from object silhouettes containing tracked object corresponding to video frames I_3, \dots, I_P , respectively
- 3: Consider $i=2$ as a *candidate* frame and $i+1$ as the *target* frame
- 4: Extract features: 8-bin HOG, R, G, and B of each pixel of *candidate* frame I_i
- 5: Generate training set from the *candidate* frame I_i
- 6: Extract features: 8-bin HOG, R, G, and B of each pixel of *target* frame I_{i+1}
- 7: Automatically generate the test set from the *target* frame I_{i+1} .
- 8: Calculate the amount of motion (M) between two consecutive I_{i-1} and I_i frames using Eqs. 1 and 2
- 9: Classify unlabeled samples in frame I_{i+1} using fuzzy k-NN classifier with local information
- 10: The object is tracked by its contour extracted from the object silhouette.
- 11: update $i=i+1$
- 12: Repeat Steps 6-11 until all frames are processed.

point, and center of the boundary of first two consecutive image frames. Figure 4 shows the contour or boundary of the object in consecutive image frames of *Run-Side2* video sequence. Amount (rough estimate) of motion (M) between two consecutive image frames is estimated as

$$M = \max\{|BM_1 - BM_2|, |UM_1 - UM_2|, \\ \times |LM_1 - LM_2|, |RM_1 - RM_2|, |C_1 - C_2|\}. \quad (1)$$

In a scene, different parts of an object may have different amounts of motions. Here, max operation is used to estimate the largest motion among all the pixels within the object. Since the movement of the object is not constant throughout

the sequence, the amount of motion between any two consecutive frames may not be constant (due to camera motion or object deformation). Hence, the amount (rough estimate) of motion between any two consecutive frames is modified as follows:

$$\bar{M} = \gamma M, \quad (2)$$

where $\gamma \in \mathbb{R}$ (Ghosh et al. 2012) is a camera (or object deformation) parameter. The γ value may be different for different (*current*) frames.

3.4 Classification of unlabeled samples

To classify the unlabeled samples of the *target* frame, a fuzzy k-NN classifier is considered. For classifying each unlabeled sample in the *target* frame, instead of considering the whole training set a subset of the training set is considered. The training subset is generated with the labeled samples from the corresponding spatial neighborhood (in the *candidate* frame) of an unlabeled sample in the *target* frame. The size of the neighborhood ($\bar{M} \times \bar{M}$) is obtained using Eq. 2, depending on the amount of motion between immediate previous two frames. This mechanism reduces the time complexity for classifying the unlabeled samples and may increase the classification accuracy.

3.5 Contour extraction and object tracking

After classification of unlabeled samples in the *target* frame, we have two regions: object and background. Figure 5a shows two regions: object and background. Sometimes the proposed method produces disconnected regions (after classification). In such a scenario, a connected component analysis scheme (Gonzalez and Woods 2008) is followed to get connected/compact object and background regions. The object region represents the silhouette of the object. Figure 5b shows the

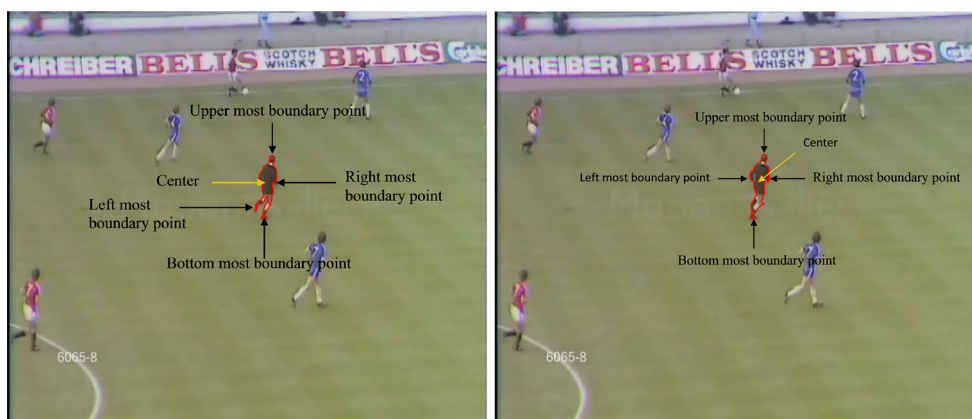


Fig. 4 Object contours in two consecutive frames of *Run-Side2* video sequence

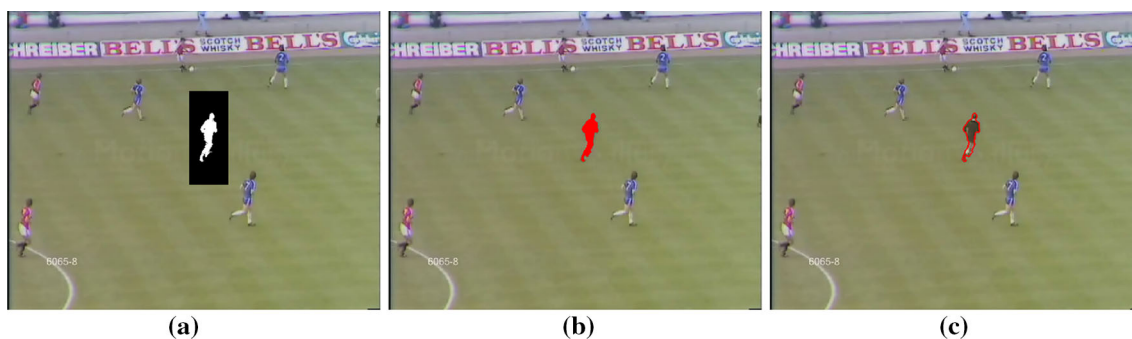


Fig. 5 *Run-Side2* video frames: **a** classification result of the third frame, **b** object silhouette in the third frame, **c** extracted contour in the third frame

object silhouette in the *target* frame. Transition pixels from the non-object region to the object silhouette and vice versa are treated as the boundary or contour pixels of the object. Connecting the boundary pixels, contour, or boundary of the object is extracted in the *target* frame. Hence, the object is tracked with its contour or boundary in the *target* frame. Figure 5c shows the extracted contour of an object in the *target* frame.

4 Experimental results and analysis

To test the effectiveness of the proposed method, eight benchmark video sequences are considered. For all the video sequences, the initial object silhouettes (extracted boundary trackers in at least two consecutive frames) are supplied manually. For the subsequent frames, the tracked object (of the previous frame) is considered as an initialization (which is done automatically by the proposed algorithm). The tracking results obtained using the proposed method are compared with six state-of-the-art contour tracking techniques and another proposed method. Since one particular tracking algorithm may not be suitable for all the data sets and also different tracking algorithms may give different results for a

single data set, it is necessary to measure the performance of tracking algorithms quantitatively. Two measures are used to quantify the performance of the tracking algorithm.

4.1 Evaluation measure

In the present article, area-based metric (Baumann et al. 2008; Kasturi et al. 2009) and F-measure (Lazarevic McManus et al. 2008) are used to measure the performance of the proposed tracking algorithm in a quantitative manner. The area-based metric is computed using the spatio-temporal overlap between the boundary of the object to the ground truth and the boundary of the object generated by the method in the target frame. ATA (Baumann et al. 2008; Kasturi et al. 2009) is used to measure the tracking performance over the whole video sequence. The trajectories obtained by the manually constructed ground truth and the obtained results are compared based on ATA. For a good object tracking system, the ATA measure should be very close to one.

The F-measure, or effectiveness measure (Lazarevic McManus et al. 2008), characterizes the performance of classification in precision-recall space, and is defined as the weighted harmonic mean of precision (P) and recall (R) metrics. For good classification, it should be very close to one.

4.2 Data sets

In the present work, eight different video sequences are considered to test the effectiveness of the proposed algorithm. Among these video sequences, *Diving-Side*, *Run-Side1*, and *Run-Side2* are taken from *UCF Sports Actions Dataset*,¹ *Clip602*, and *Clip680* from *Reef Video Clip Database*,² *Waving-Tree* from *Microsoft Wallflower Dataset*,³ *Traffic-Sequence1* from *Center for Research in Computer Vision*⁴ and *Traffic-Sequence2* from *Universitat Karlsruhe*.⁵ Since the main objective of the present work is to handle the challenges for tracking contour of a moving object from video sequences, where an object in the scene changes its shape, have illumination variation, with complex and clutter background, and unclear boundary with better accuracy, we have chosen video sequences with objects possessing these sort of characteristics.

In *Diving-Side* video sequence, a woman is diving into water. It is difficult to track the boundary of the woman. During diving, the shape of the woman is changed and due to large motion, boundary of the woman is also getting blurred. In *Run-Side1* video sequence, one person is running on a way. Our objective is to track that person. During his run time, the person changes his shape. Similar kinds of objects are running together. Some parts of background are also similar to some parts of the object (person).

Run-Side2 is a football sequence. In this sequence, players for two teams and referee are moving. Our aim is to track only the referee. As the color of the referee's leg is very much similar to that of background and the referee has blurred boundary due to large motion, it is difficult to track the contour of the referee. On the other hand, one big fish is moving under the water in *Clip602* video sequence. This is a complex video sequence as many similar kinds of small fishes are moving around a big fish and so many under-water plants are also moving due to wave of water. Similarly one big fish is moving in under-water *Clip680* video sequence. It is difficult to distinguish the fish from its surrounding background which affects the contour tracking.

In *Waving-Tree* video sequence, we want to track the person who is walking in front of the waving tree in the background. This waving tree makes tracking of the person difficult. *Traffic-Sequence1* is a dense traffic sequence in which several cars are moving with different velocities. Tracking of a particular vehicle becomes challenging as other vehicles act as clutter for it. *Traffic-Sequence2* is also a highway

traffic video sequence that contains vehicles which are moving with different speeds in different directions. Our objective is to track a particular vehicle. The major challenges for tracking the vehicle are the problems of snow clutter in the background which are similar to the moving objects.

4.3 Analysis of results

As mentioned earlier, eight different video sequences are considered to test the effectiveness of the proposed method. Figures 6, 7, 8, 9, 10, 11, 12, and 13 show the contour tracking results of *Diving-Side*, *Run-Side1*, *Run-Side2*, *Clip602*, *Clip680*, *Waving-Tree*, *Traffic-Sequence1*, and *Traffic-Sequence2* video sequences using different methods. From Fig. 6a, it is seen that the contour of the woman is not properly tracked by Snake (Kass et al. 1988). The head portion of the woman is detected as part of the background in most of the frames. As few prominent edges are also present in the background, more parts of the background are also included as part of the object during contour tracking. Due to the large motion of the hands, end parts of her fingers are treated as background and some background parts are treated as part of the object during tracking using GVF (Paragios et al. 2004). But it tracks the woman better than by Snake (Kass et al. 1988) model. Only the end parts of fingers of the woman are not properly tracked by Paragios et al. (2004). Figure 6b shows the contour tracking result using GVF (Paragios et al. 2004). Contour tracking results produced by Level set (Brox et al. 2010) are given in Fig. 6c. From this figure, clearly Level set (Brox et al. 2010) provides similar results as compared to GVF (Paragios et al. 2004). Only difference is that fewer parts of background are detected as the object during contour tracking. On the other hand, ABTOC (Chiveron et al. 2012) method is also unable to properly track the contour as the background contains more prominent edges. Figure 6d shows the tracking results using ABTOC (Chiveron et al. 2012) method. However, HRIPBACOT (Aitfares et al. 2013) method tried to track the contour properly but in case of large motion, it is also unable to track properly as depicted in Fig. 6e. Figure 6f shows the tracking results obtained by JRACSOT (Ning et al. 2013) method. This figure highlights that JRACSOT method tracks in a better way than ABTOC and HRIPBACOT methods as similarity measure is used to match the detected regions. During classification using fuzzy k-NN, only end part of the woman's fingers is treated as background due to large motion of it. Fuzzy k-NN tracks the woman with better accuracy excluding few end parts of the fingers. The extracted contours (contour tracking results) from its silhouettes using fuzzy k-NN are presented in Fig. 6g. On the other hand, the contour of the woman is better tracked by the proposed method than all other methods. Figure 6h shows the extracted contour tracking results using the proposed method. From this figure, it is clear that the end parts of the finger are

¹ <http://server.cs.ucf.edu/vision/data.html>.

² <http://www.reefvid.org/>.

³ <http://research.microsoft.com/enus/um/people/jckrumm/wallflower/testimages.htm>.

⁴ <http://crcv.ucf.edu/data/tracking.php/>.

⁵ http://i21www.ira.uka.de/image_sequences/.

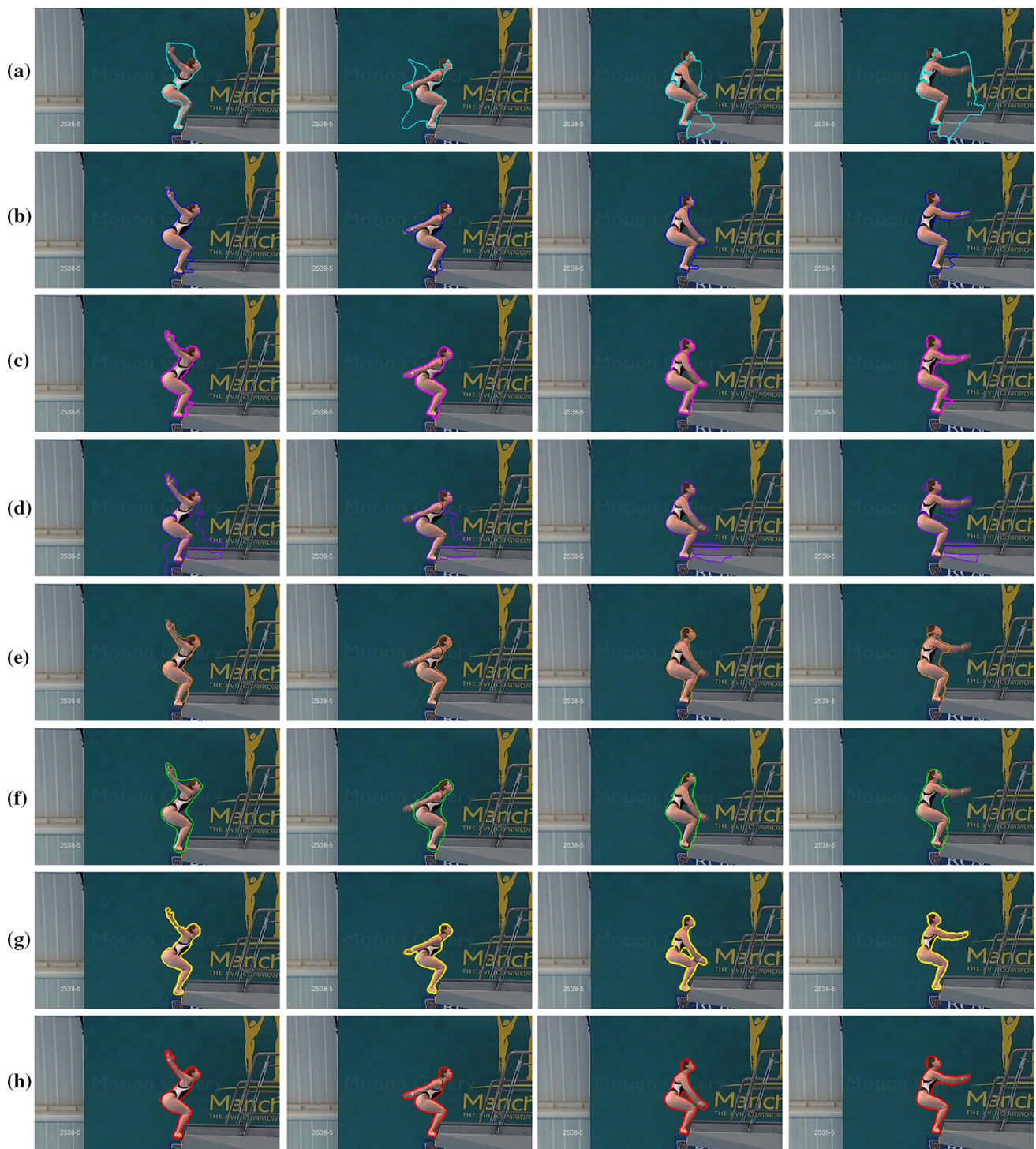


Fig. 6 Diving-Side video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC

(Chiverton et al. 2012). **e** Tracking results by HRIPBACOT (Aitfares et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

properly tracked (as a part of the object) by the proposed method. It is evident that the proposed technique can track the minute details of the object as well.

Figures 7a–h and 8a–h show the contour tracking of *Run-Side1* and *Run-Side2* video sequences using Snake (Kass et

al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), JRACSOT (Ning et al. 2013), fuzzy k-NN, and the proposed method, correspondingly. From these figures it is also seen that the proposed method produced

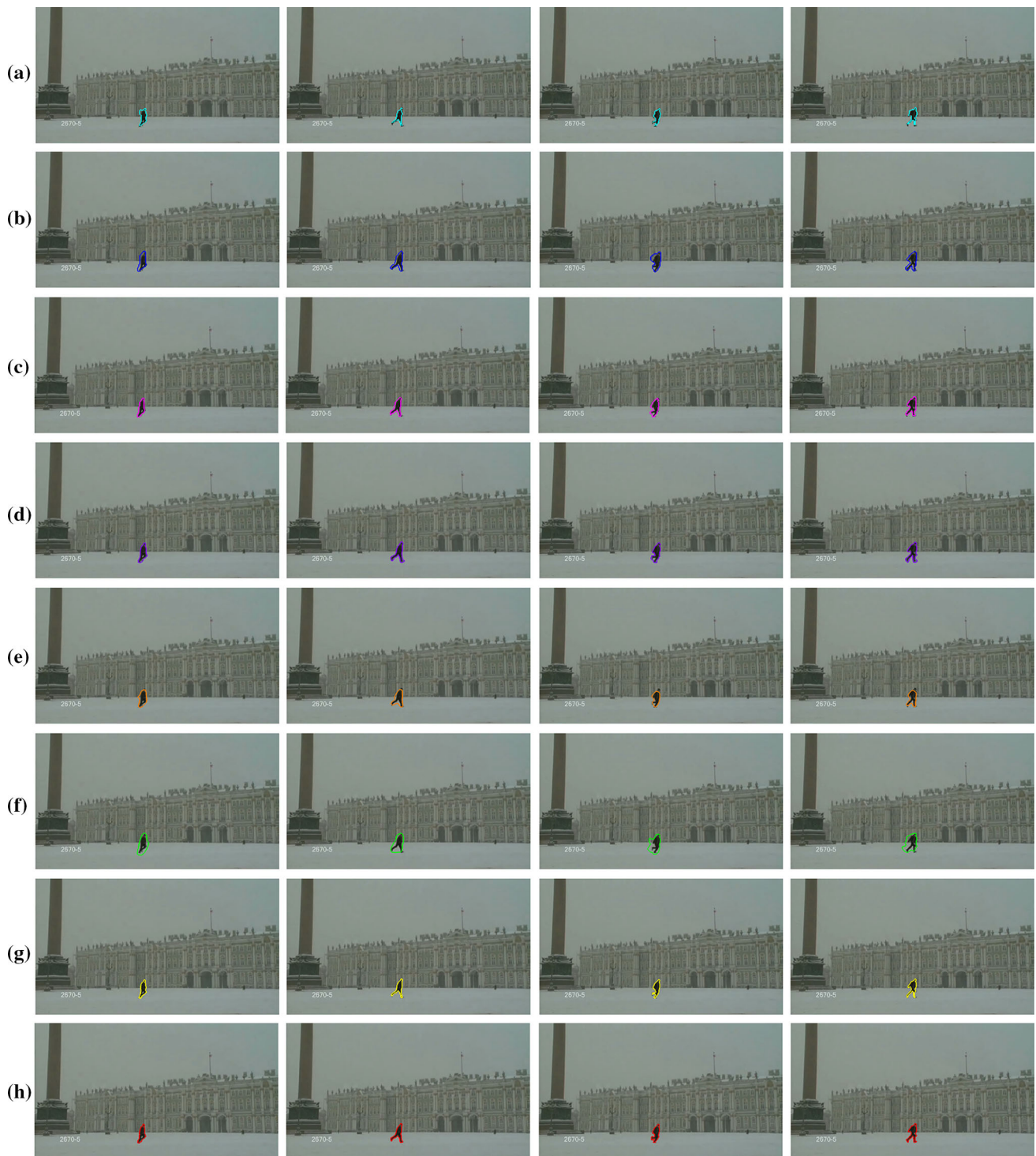


Fig. 7 *Run-Side1* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC

(Chiverton et al. 2012). **e** Tracking results by HRIPBACOT (Aitfares et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

better tracking results than all other methods for both the above-mentioned video sequences. It is concluded from the results of the above two video sequences that the proposed method is also able to track small objects properly.

Now we visually analyze the tracking results of *Clip602* video sequence. From Fig. 9a–c it is seen that Snake (Kass et al. 1988), GVF (Paragios et al. 2004), and Level set (Brox et al. 2010) included some small fishes within the contour



Fig. 8 *Run-Side2* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIPBACOT (Aitfares

et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

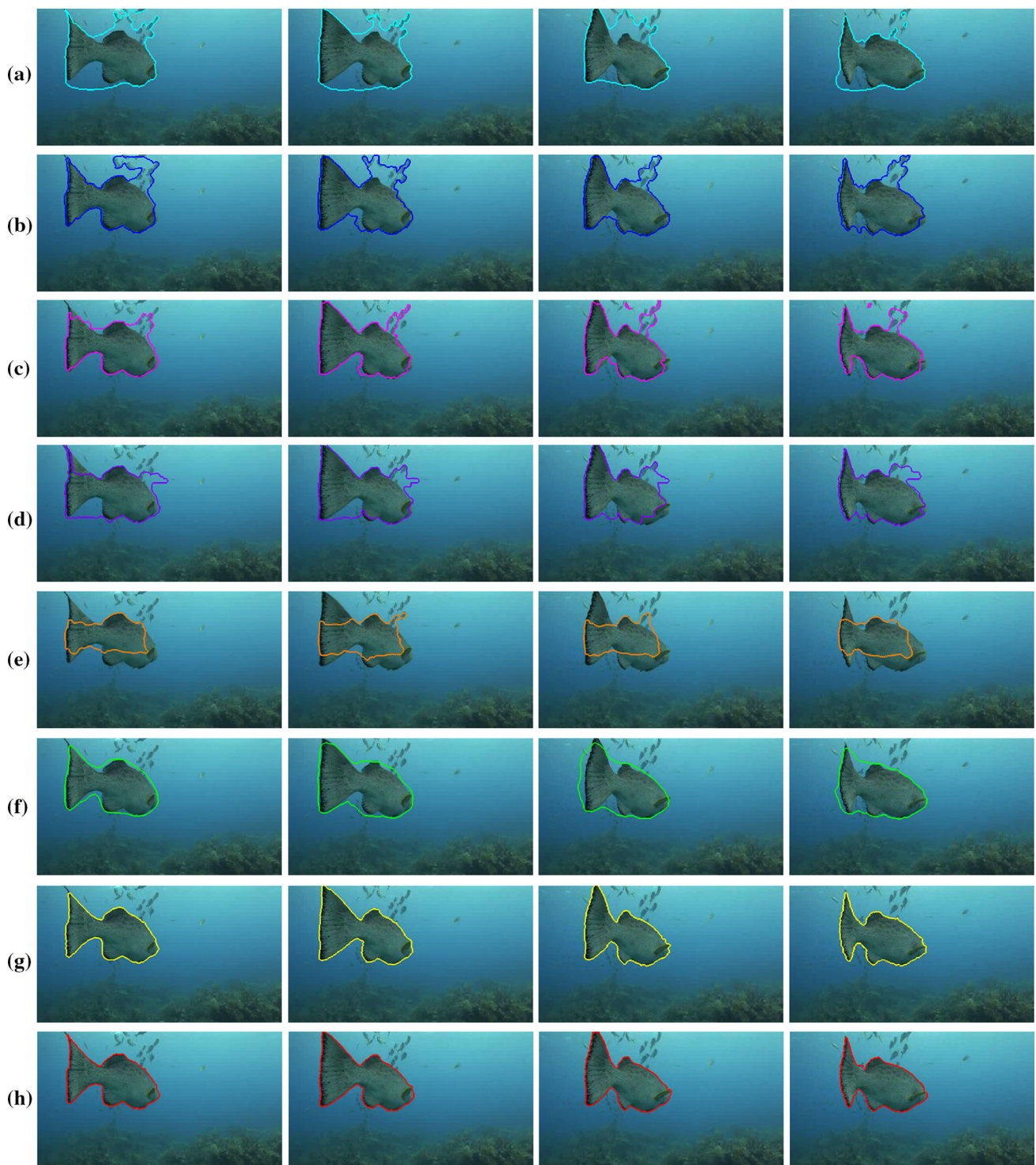


Fig. 9 *Clip602* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIPBACOT (Aitfares et

al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

of the big fish. Since, each small fish has its own contour, during energy minimization, they stuck to the contour of some small fishes. From these figures, it is also found that

Level set (Brox et al. 2010) provides better results than Snake (Kass et al. 1988) and GVF (Paragios et al. 2004). Results obtained using ABTOC (Chiverton et al. 2012),

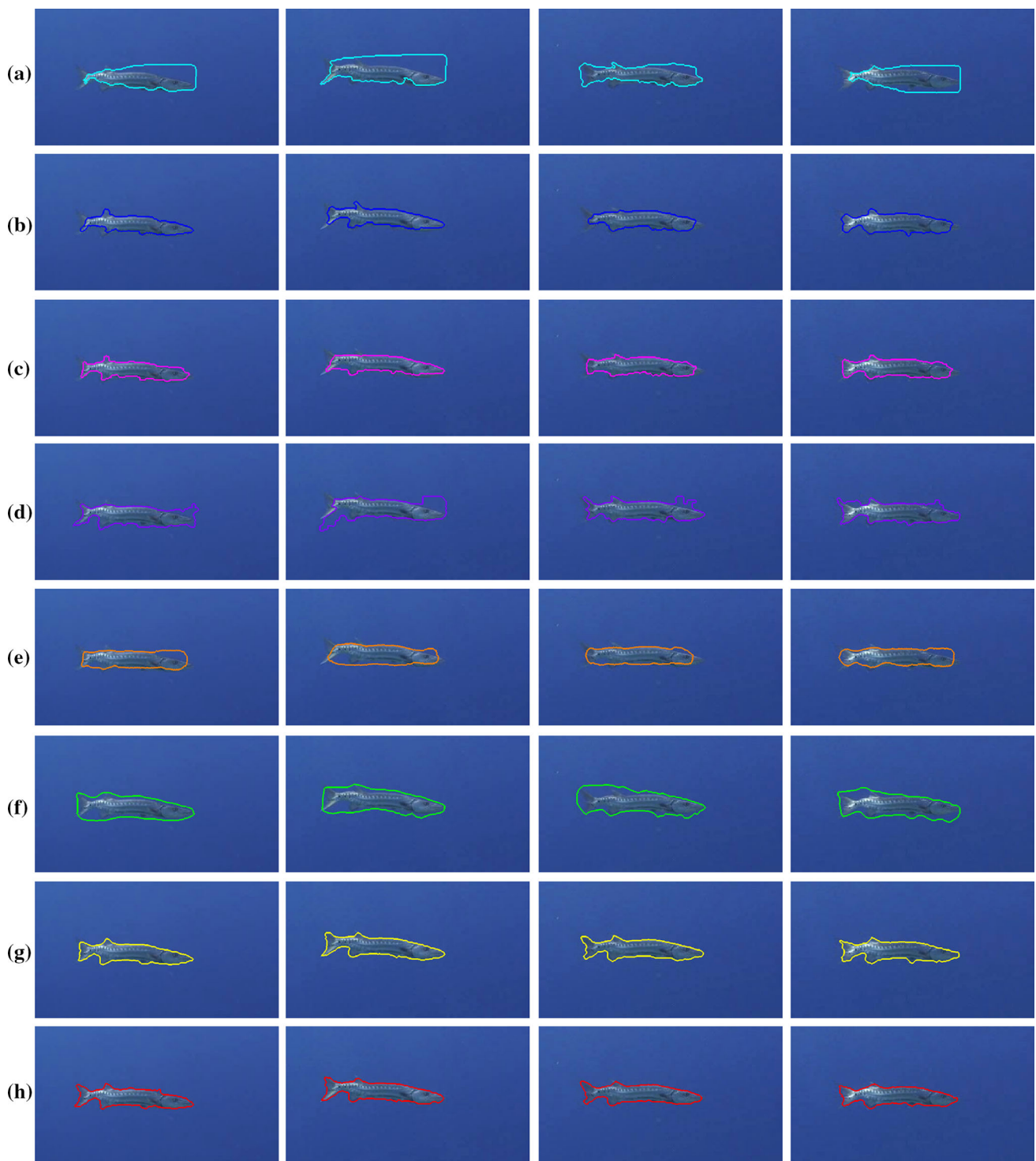


Fig. 10 *Clip680* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIPBACOT (Aitfares et

al. 2013), and JRACSOT (Ning et al. 2013) techniques are displayed in Fig. 9d, e and f, respectively. From this figure it is evident that JRACSOT produces better results than other methods. Figure 9g, h

al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

shows the extracted contours using fuzzy k-NN and the proposed method, respectively. From Fig. 9h it is clear that the proposed method detected the contour of the big fish properly.

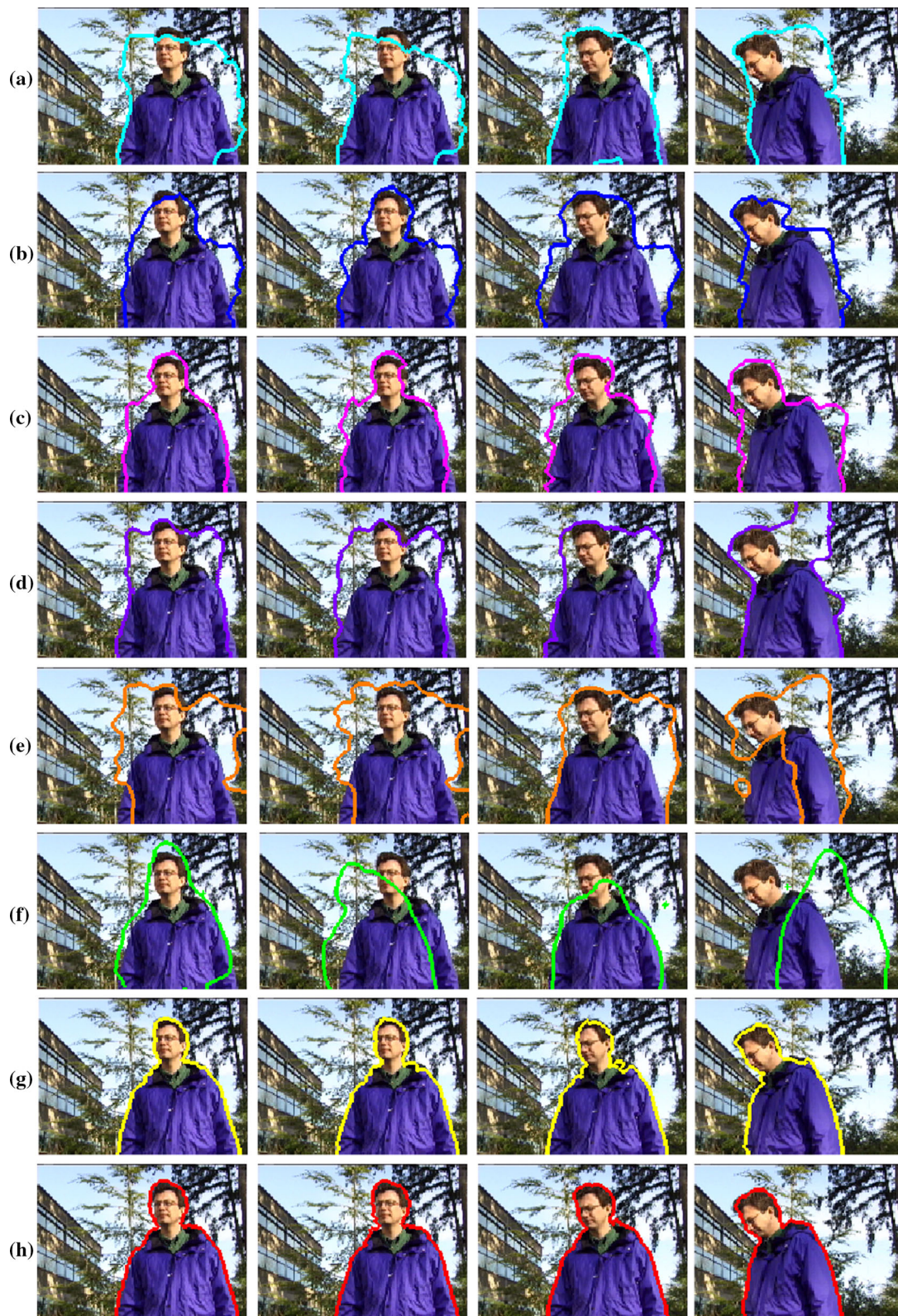


Fig. 11 *Waving-Tree* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIPBA-

COT (Aitfares et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method



Fig. 12 *Traffic-Sequence1* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIP-

BACOT (Aitfares et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

Figure 10a–h shows the contour tracking results of *Clip680* using Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013),

JRACSOT (Ning et al. 2013), fuzzy k-NN, and the proposed method, correspondingly. From these results, it is found that fuzzy k-NN produces better result than all other methods.

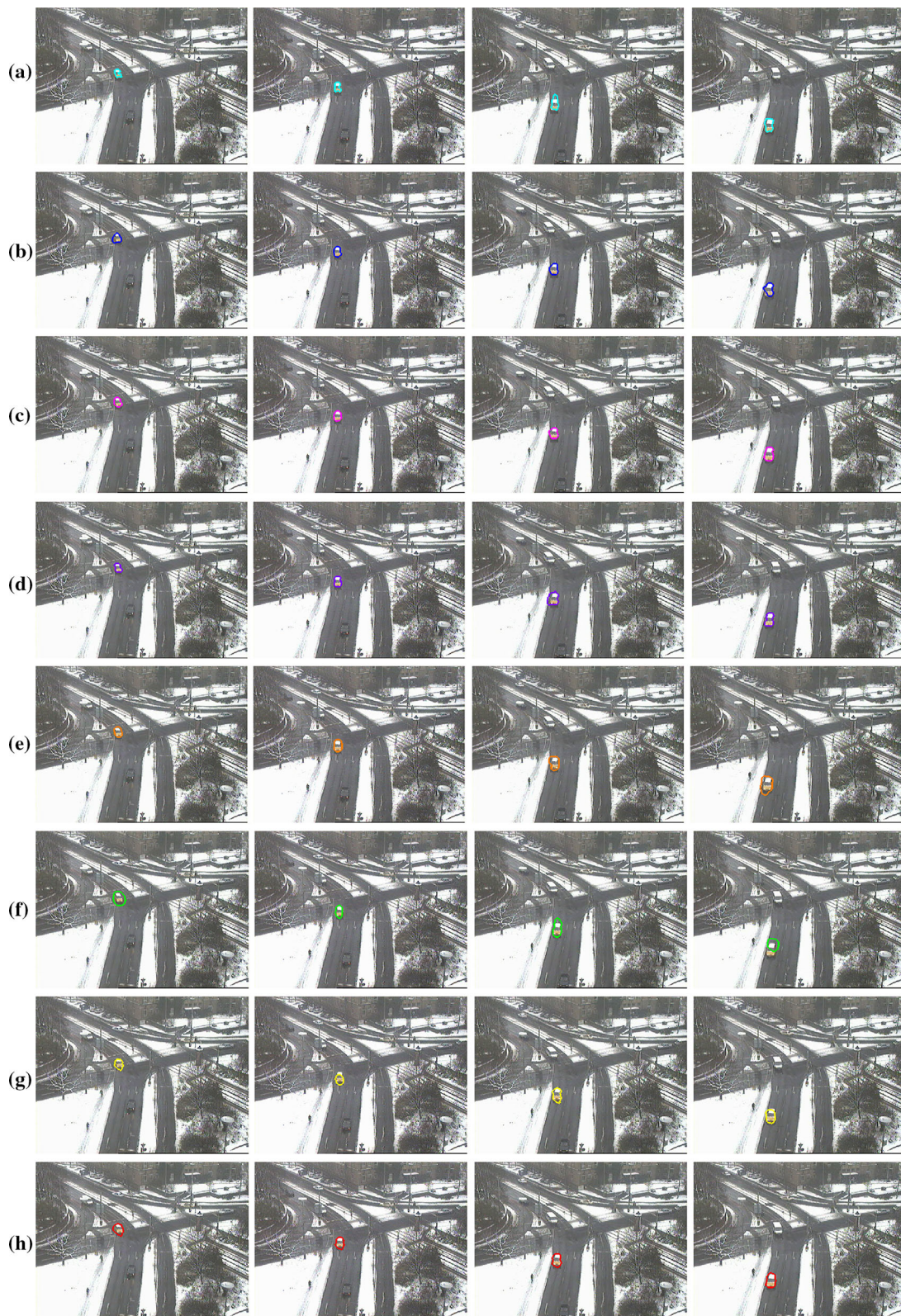


Fig. 13 *Traffic-Sequence2* video sequence. **a** Tracking results by Snake (Kass et al. 1988). **b** Tracking results by GVF (Paragios et al. 2004). **c** Tracking results by Level set (Brox et al. 2010). **d** Tracking results by ABTOC (Chiverton et al. 2012). **e** Tracking results by HRIP-

BACOT (Aitfares et al. 2013). **f** Tracking results by JRACSOT (Ning et al. 2013). **g** Extracted contours by fuzzy k-NN. **h** Extracted contours by the proposed method

Figure 11a displays the tracking results of *Waving-Tree* using Snake (Kass et al. 1988). From this figure, it is seen that Snake (Kass et al. 1988) is unable to provide proper contour of the person. It detects few parts of the head as part of the background for most of the frames. During the contour evolution, it gets stuck at the background as it contains more prominent edges. More parts of the background are included as object within the contour. Figure 11b, c shows the contour results obtained using GVF (Paragios et al. 2004) and Level set (Brox et al. 2010), respectively. These figures indicate that Level set (Brox et al. 2010) produces better contour of the person than Snake (Kass et al. 1988) and GVF (Paragios et al. 2004). However due to the problem of getting stuck to local minima of active contour, ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013) and JRACSOT (Ning et al. 2013) methods are unable to provide good results which are displayed in Fig. 11d, e, and f, respectively. On the other hand, fuzzy k-NN classifies few parts of the head of the person as the background for few number of frames and tracks the person with better accuracy excluding only few parts of the head. Figure 11g shows the object contours (contour tracking results) using fuzzy k-NN. The contour of the person is better tracked by the proposed method than all other methods. Figure 11h shows the extracted contour tracking results using the proposed method. From these figures, it is clear that head of the person is properly tracked (as a part of the object) by the proposed method. It is evident that the proposed technique can track the minute details of the object as well.

Figure 12a–h displays the contour tracking results of *Traffic-Sequence1* using Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), JRACSOT (Ning et al. 2013), fuzzy k-NN, and the proposed method, correspondingly. From these results, it is found that the proposed method produces better results than all other techniques.

Figure 13a–h shows the contour tracking results of *Traffic-Sequence2* using Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), JRACSOT (Ning et al. 2013), fuzzy k-NN, and the proposed method, correspondingly. These figures indicate that Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), and JRACSOT (Ning et al. 2013) techniques tried to track the vehicle. But the boundary (contour) of the vehicle is not properly defined and also the background contains clutter which forces these algorithms to stuck at local minimum and provide unsatisfactory results. It is also observed that fuzzy k-NN is also unable to properly track the vehicle due to the background clutter. On the other hand, in the proposed method, the local information guides fuzzy

k-NN during the classification process and is able to provide better classification and tracking results.

It is already discussed in Sect. 5 that ATA and F-measure are used to measure the performance of different methods quantitatively. Tables 1 and 2 display the ATA and average F-measure for contour tracking using different methods. For a good object tracking system, the ATA measure and F-measure should be very close to one. From these tables, it is clear that the proposed method provides maximum ATA and F-measure values for seven video (*Diving-Side*, *Run-Side1*, *Run-Side2*, *Clip602*, *Waving-Tree*, *Traffic-Sequence1*, and *Traffic-Sequence2*) sequences. The maximum ATA and F-measure values are displayed in bold font in the tables. On the other hand, Snake (Kass et al. 1988) provides minimum ATA values for all the sequences. Also, Snake (Kass et al. 1988) provides minimum F-measure values for five video sequences. The minimum ATA and F-measure values are displayed in italic font in the tables. From the quantitative measures of Tables 1 and 2, it is seen that the proposed method provides better results for seven video sequences than all other methods. Figures 14 and 15 display the graphical representation of ATA and average F-measure of eight video sequences using different methods. From these figures, it is also verified that the proposed method provides better contour tracking accuracy for seven video sequences among eight sequences than all other methods.

4.4 Computational complexity

Suppose each image frame contains P number of pixels. Let the cost for calculating HOG features for each pixel be C_{hog} . Therefore, total cost for calculating HOG of each frame is $P \times C_{\text{hog}}$. Now

$$P \times C_{\text{hog}} = P \times M_{\text{hog}} \times W_{\text{hog}},$$

where M_{hog} is the number of histogram bins and W_{hog} is the considered window size. To generate each training pattern, it needs constant cost; let it be C_{training} . Therefore, total cost for generating the training set is $P_1 \times C_{\text{training}}$, where P_1 is the number of training patterns in the training set and $P_1 \leq P$. Let the constant cost C_{test} be required to generate each test pattern. Therefore, total cost for generating test set is $P_1 \times C_{\text{test}}$, where P_1 is the number of test patterns in the test set and is equal to the number of training patterns.

In Step 8 of Algorithm 1, to estimate the motion using Eqs. 1 and 2, it needs constant cost C_{n1} and C_{n2} , respectively. Let $W_n \times k$ cost be required to classify each unlabeled pixel (sample) using fuzzy k-NN, where W_n is the number of pixels in the considered neighborhood and k is the number of nearest neighbors. Thus, total cost required for classification of all the test patterns is $P_1 \times W_n \times k$. To extract the contour it also needs constant cost; let it be C_{contour} .

Table 1 Performance measure using *Area Based Metric (ATA)*

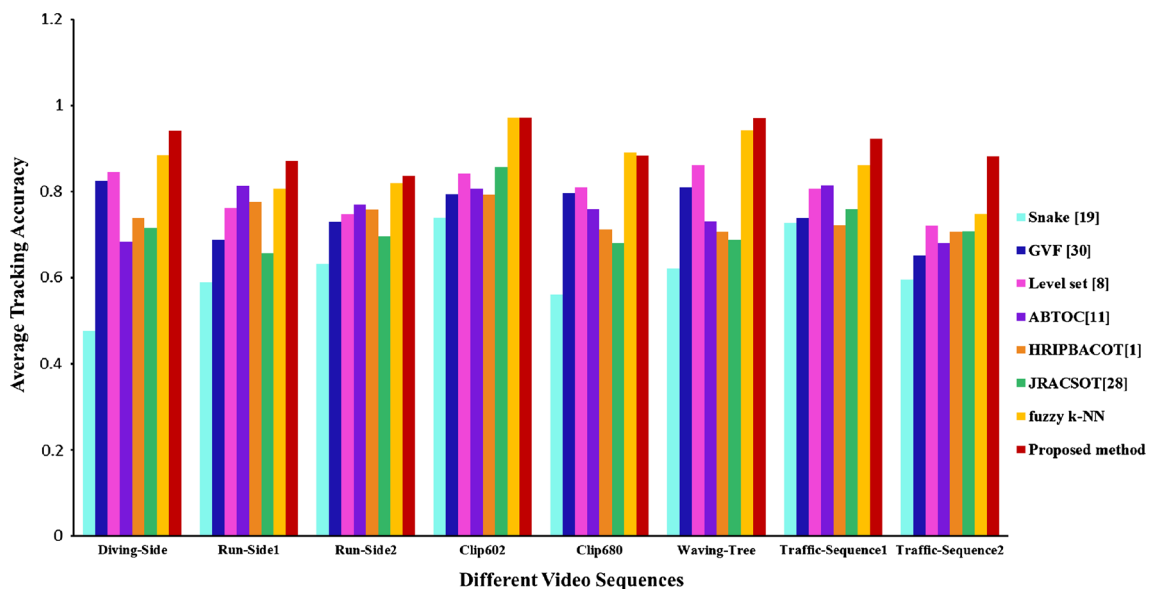
Considered video sequences	Snake (Kass et al. 1988)	GVF (Paragios et al. 2004)	Level set (Brox et al. 2010)	ABTOC (Chiverton et al. 2012)	HRIPBACOT (Aitfares et al. 2013)	JRACSOT (Ning et al. 2013)	Fuzzy k-NN	Proposed method
<i>Diving-Side</i>	0.476032	0.826055	0.846026	0.683698	0.738374	0.715744	0.885300	0.941375
<i>Run-Side1</i>	0.590227	0.688129	0.762253	0.813165	0.776124	0.65697	0.806152	0.870950
<i>Run-Side2</i>	0.632597	0.729215	0.747285	0.76937	0.758024	0.695769	0.819727	0.836584
<i>Clip602</i>	0.739950	0.794567	0.841966	0.807038	0.793484	0.856773	0.971688	0.972585
<i>Clip680</i>	0.560396	0.796381	0.809659	0.75928	0.712112	0.680586	0.890021	0.884271
<i>Waving-Tree</i>	0.620667	0.809785	0.861067	0.730142	0.706817	0.688276	0.942517	0.970961
<i>Traffic-Sequence1</i>	0.727617	0.738523	0.807140	0.814511	0.732112	0.759428	0.861577	0.922790
<i>Traffic-Sequence2</i>	0.595556	0.650901	0.721030	0.680096	0.70724	0.707745	0.748760	0.881423

Bold values indicate better results than other methods and italic values indicate worst results among different methods

Table 2 Average F-measure

Considered video sequences	Snake (Kass et al. 1988)	GVF (Paragios et al. 2004)	Level set (Brox et al. 2010)	ABTOC (Chiverton et al. 2012)	HRIPBACOT (Aitfares et al. 2013)	JRACSOT (Ning et al. 2013)	Fuzzy k-NN	Proposed method
<i>Diving-Side</i>	0.844324	0.934051	0.929787	0.855743	0.893686	0.876846	0.938203	0.974678
<i>Run-Side1</i>	0.759186	0.861288	0.888845	0.906125	0.893183	0.837679	0.892649	0.932256
<i>Run-Side2</i>	0.841831	0.857736	0.883554	0.874636	0.859831	0.84036	0.903028	0.916003
<i>Clip602</i>	0.980655	0.973058	0.962507	0.94654	0.932248	0.921341	0.985845	0.986307
<i>Clip680</i>	0.833163	0.886669	0.894970	0.845891	0.815713	0.804877	0.969388	0.966391
<i>Waving-Tree</i>	0.772582	0.928278	0.944540	0.82183	0.807543	0.790438	0.970833	0.985778
<i>Traffic-Sequence1</i>	0.842334	0.849598	0.869598	0.858573	0.82306	0.830737	0.893324	0.959845
<i>Traffic-Sequence2</i>	0.716253	0.837905	0.902690	0.840705	0.873519	0.873185	0.910932	0.957045

Bold values indicate better results than other methods and italic values indicate worst results among different methods

**Fig. 14** ATA of different video sequences using different methods

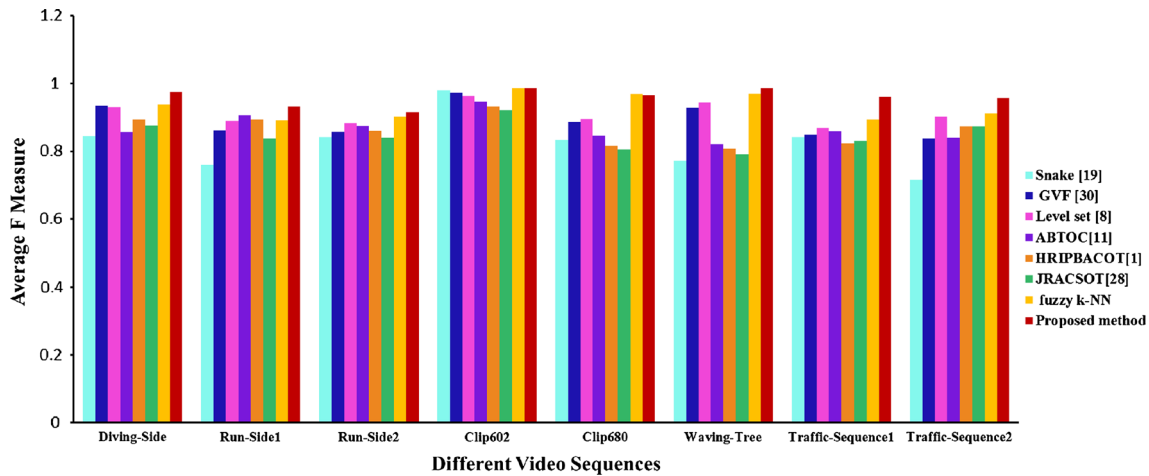


Fig. 15 Average F-measure of different video sequences using different methods

Hence, total cost for the proposed method is

$$\begin{aligned}
 &= P \times C_{hog} + P_1 \times C_{training} + P_1 \times C_{test} + C_{n1} + C_{n2} \\
 &\quad + P_1 \times W_n \times k + C_{contour} \\
 &= P \times M_{hog} \times W_{hog} + P_1 \times C_{training} + P_1 \times C_{test} \\
 &\quad + C_{n1} + C_{n2} + P_1 \times W_n \times k \\
 &\quad + C_{contour}, \quad \text{where } C_{hog} = M_{hog} \times W_{hog} \\
 &= P \times M_{hog} \times W_{hog} + P_1 \times C_{training} + P_1 \times C_{test} + P_1 \\
 &\quad \times W_n \times k + C_{n1} + C_{n2} + C_{contour} \\
 &= P \times M_{hog} \times W_{hog} + P_1 \times C_{training} \\
 &\quad + P_1 \times C_{test} + P_1 \times W_n \times k \\
 &\quad + C_{total}, \quad \text{where } C_{total} = C_{n1} + C_{n2} + C_{contour} \\
 &\leq P \times M_{hog} \times W_{hog} + P \times C_{training} \\
 &\quad + P \times C_{test} + P \times W_n \times k, \text{ since } P_1 \leq P \\
 &= P(M_{hog} \times W_{hog} + C_{training} + C_{test} + W_n \times k) \\
 &= P(M_{hog} \times W_{hog} + W_n \times k + C_t), \\
 &\quad \text{where } C_t = C_{training} + C_{test} \\
 &\leq P(M_{hog} \times P + P \times k), \text{ since } W_{hog} \ll P \text{ and } W_n \ll P \\
 &= P^2(M_{hog} + k).
 \end{aligned}$$

Since for a particular video sequence, both M_{hog} and k are constants, the computational complexity of the proposed algorithm is $O(P^2)$, where P is the number of pixels in the given image frame.

Experiments through simulation have been conducted using visual C++ on a machine with Intel(R) Core(TM)i3 CPU 3.20 GHz and 4.0 GB RAM. Average execution time required for tracking the object of the given video sequences using various methods is provided in Table 3. From the table, it is clear that fuzzy k-NN consumes maximum time for tracking *Diving-Side*, *Run-Side2*, *Clip602*, *Clip680*, *Waving-Tree*, *Traffic-Sequence1*, and *Traffic-Sequence2* video sequences.

The maximum average execution time for the contour tracking of all considered video sequences is displayed in italic font in the table. The proposed method requires the least amount of average execution time for tracking *Run-Side1*, *Run-Side2*, *Clip602*, *Clip680*, *Waving-Tree*, *Traffic-Sequence1*, and *Traffic-Sequence2* video sequences. The least amount of average computational time is given in bold font in the table. Since the object movement between frames is very large for *Diving-Side* video sequence, the proposed method failed to reduce the computational time. Figure 16 shows the graphical representation of average computational time for contour tracking using different methods. From this figure, it is also seen that the proposed method consumes the least amount of computational time for tracking.

Although the proposed method requires least amount of average execution time for tracking contour of the object for seven video sequences than other methods, it may not be applicable for on-line application. The proposed algorithm may be useful for specific off-line applications, such as visual scene analysis, event analysis in surveillance, video annotation, and video motion capture.

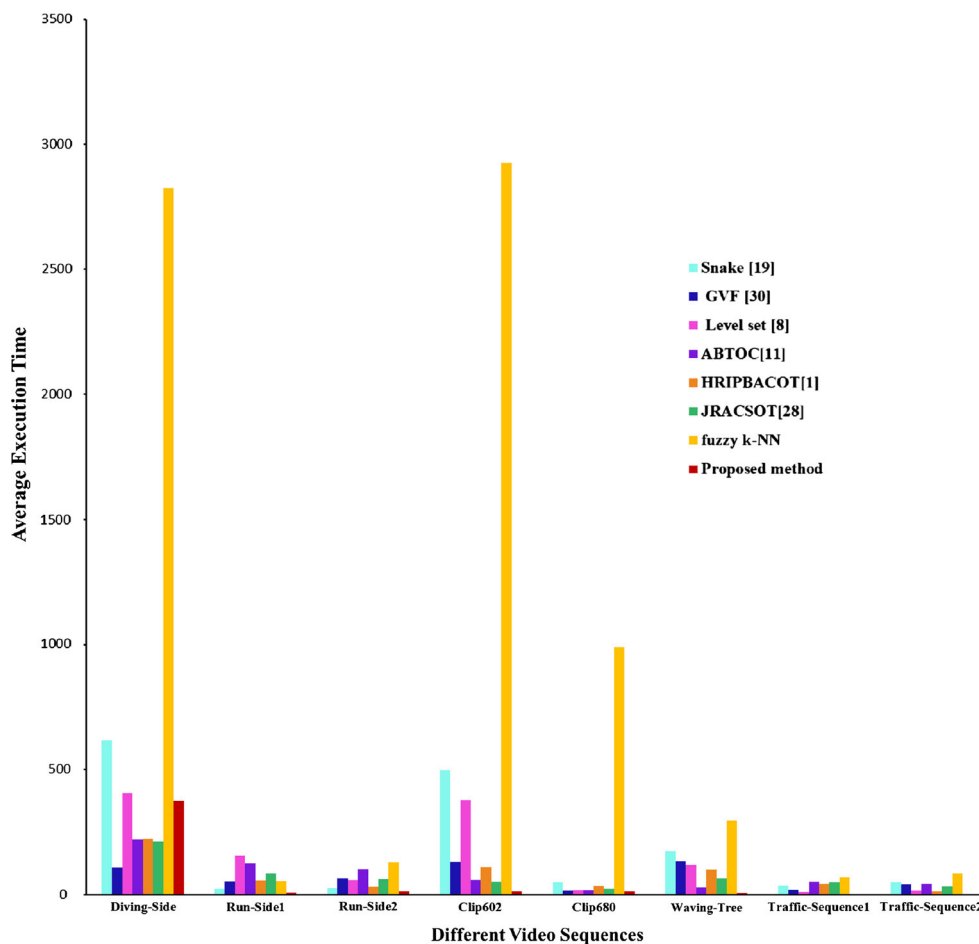
5 Discussion and future work

The above analysis of results indicates that Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), and JRACSOT (Ning et al. 2013) techniques are unable to produce good tracking results if the background contains more prominent edges. All these are contour evolving techniques and extract contour when energy is minimum. During energy minimization, if the background contains more prominent edges they may stuck at these edges. On the other hand, fuzzy k-NN classifies the pixels into

Table 3 Average execution time (second)

Considered video sequences	Snake (Kass et al. 1988)	GVF (Paragios et al. 2004)	Level set (Brox et al. 2010)	ABTOC (Chiverton et al. 2012)	HRIPBACOT (Aitfares et al. 2013)	JRACSOT (Ning et al. 2013)	Fuzzy k-NN	Proposed method
<i>Diving-Side</i>	617.628	108.937	405.356	220.665	222.112	212.37	2824.973	375.730
<i>Run-Side1</i>	22.692	53.046	<i>155.911</i>	124.717	57.082	84.372	54.598	8.459
<i>Run-Side2</i>	27.287	64.376	58.725	101.412	31.247	63.253	<i>130.598</i>	12.932
<i>Clip602</i>	496.477	130.878	377.296	59.979	110.197	51.997	2923.845	13.693
<i>Clip680</i>	50.063	16.812	19.027	17.976	35.863	23.337	<i>989.560</i>	12.642
<i>Waving-Tree</i>	173.910	132.812	119.027	29.393	100.185	64.857	<i>295.442</i>	7.159
<i>Traffic-Sequence1</i>	36.195	20.721	11.590	51.769	43.316	49.661	<i>70.560</i>	2.850
<i>Traffic-Sequence2</i>	50.063	42.342	16.467	43.392	13.103	33.857	<i>84.631</i>	2.142

Bold values indicate minimum average execution time and italic values indicate maximum average execution time among different methods

**Fig. 16** Average execution time of different video sequences using different methods

object and non-object (background). Due to large motion of the object or the camera, the boundary of the object is also blurred. Fuzzy k-NN uses global information to classify pixels of the *target* frame. Due to the blurred boundary and the global information, a few parts of the object boundary are

not properly classified by fuzzy k-NN and unable to produce better contour. However, in the proposed method, instead of the global information, the local information based on the object motion between two consecutive frames is considered to classify pixels in the *target* frame. As this local informa-

tion depends on the object motion, it helps the fuzzy k-NN to provide better classification and contour extraction. The proposed method is for tracking the contour of single object. In future, we plan to extend it to track contours of multiple objects and also to track the contours for occluded objects from a given video sequence.

6 Conclusions

In this article, we have proposed an algorithm that can efficiently track the contour, extracted from the silhouette of the moving object from a given video sequence using local neighborhood information and fuzzy k-NN algorithm. In the proposed work, to classify each unlabeled sample in the *target* frame, instead of considering the whole training set, a subset of training set is considered. A heuristic technique based on the amount of motion of the object between two consecutive frames is proposed to generate training subset from the *candidate* frame at the corresponding neighborhood of each unlabeled sample in the *target* frame. This technique makes the classification process faster and may increase the classification accuracy. From the results, it is found that the proposed method produced better contour tracking results than those obtained using Snake (Kass et al. 1988), GVF (Paragios et al. 2004), Level set (Brox et al. 2010), ABTOC (Chiverton et al. 2012), HRIPBACOT (Aitfares et al. 2013), JRACSOT (Ning et al. 2013), and fuzzy k-NN for seven video sequences. It is also found that the proposed method required least amount of computational time for tracking the given video sequences.

Acknowledgments The authors like to thank the reviewers for their thorough and constructive comments, which helped a lot to enhance the quality of the manuscript. Funding by U. S. Army through the project “Processing and Analysis of Aircraft Images with Machine Learning Techniques for Locating Objects of Interest” (Contract No. FA5209-08-P-0241) is also gratefully acknowledged.

References

- Aitfares W, Bouyakhf E, Herbulot A, Regragui F, Devy M (2013) Hybrid region and interest points-based active contour for object tracking. *Appl Math Sci* 7(118):5879–5899
- Allili MS, Ziou D (2008) Object tracking in videos using adaptive mixture models and active contours. *Neurocomputing* 71:2001–2011
- Avidan S (2004) Support vector tracking. *IEEE Trans Pattern Anal Mach Intell* 26(8):1064–1072
- Avidan S (2007) Ensemble tracking. *IEEE Trans Pattern Anal Mach Intell* 29(2):261–271
- Babenko B, Yang M, Belongie S (2011) Robust object tracking with online multiple instance learning. *IEEE Trans Pattern Anal Mach Intell* 33(8):1619–1632
- Baumann A, Boltz M, Ebling J, Koenig M, Loos HS, Merkel M, Niem W, Warzelhan JK, Yu J (2008) A review and comparison of measures for automatic video surveillance systems. *EURASIP J Image Video Process* 2008(824726):1–30
- Bovic AL (2000) *Image and video processing*. Academic Press, New York
- Brox T, Rousson M, Deriche RD, Weickert J (2010) Colour, texture, and motion in level set based segmentation and tracking. *Image Vis Comput* 28(3):376–390
- Caselles V (1995) Geometric models for active contours. In: *International conference on image processing*, vol 3, pp 9–12
- Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. *Int J Comput Vis* 22(1):61–79
- Chiverton J, Xie X, Mirmehdi M (2012) Automatic bootstrapping and tracking of object contours. *IEEE Trans Image Process* 21(3):1231–1245
- Cohen LD (1991) On active contour models and balloons. *Image Underst* 53(2):211–218
- Collins RT, Liu Y, Leordeanu M (2005) Online selection of discriminative tracking features. *IEEE Trans Pattern Anal Mach Intell* 27(10):1631–1643
- Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Proceedings of the 2005 IEEE Computer Society conference on computer vision and pattern recognition*, vol 1, pp 886–893
- Freeman WT, Roth M (1995) Orientation histograms for hand gesture recognition. In: *IEEE international workshop on automatic face and gesture recognition*, pp 296–301
- Ghosh A, Subudhi BN, Ghosh S (2012) Object detection from videos captured by moving camera by fuzzy edge incorporated Markov Random Field and local histogram matching. *IEEE Trans Circuits Syst Video Technol* 2(8):1127–1135
- Gonzalez RF, Woods RE (2008) *Digital image processing*. Pearson Education, Singapore
- Grabner H, Grabner M, Bischof H (2006) Real-time tracking via on-line boosting. In: *Proceedings of the British machine vision conference*, vol 1, pp 47–56
- Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1(4):321–331
- Kasturi R, Goldgof D, Soundararajan P, Manohar V, Garofolo J, Bowers R, Boonstra M, Korzhova V, Zhang J (2009) Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics and protocol. *IEEE Trans Pattern Anal Mach Intell* 31(2):319–336
- Keller JM, Gray MR, Givens JA (1985) A fuzzy k-nearest neighbor algorithm. *IEEE Trans Syst Man Cybernet* 15(4):580–585
- Lazarevic-McManus N, Renno JR, Makris D, Jones GA (2008) An object-based comparative methodology for motion detection based on the f-measure. *Comput Vis Image Underst* 111(1):74–85
- Lefèvre S, Gerard JP, Piron A, Vincent N (2002) An extended snake model for real-time multiple object tracking. In: *International workshop on advanced concepts for intelligent vision systems*, pp 268–275
- Levi K, Weiss Y (2004) Learning object detection from a small number of examples: the importance of good features. In: *Proceedings of the 2004 IEEE Computer Society conference on computer vision and pattern recognition*, vol 2, pp 53–60
- Luo S, Li R, Ourselin S (2003) A new deformable model using dynamic gradient vector flow and adaptive balloon forces. In: *APRS workshop on digital computing*, pp 9–14
- Malladi R, Sethian JA, Vemuri BC (1995) Shape modeling with front propagation: a level set approach. *IEEE Trans Pattern Anal Mach Intell* 17(2):158–175
- Nguyen HT, Worrington M, van den Boomgaard R, Smeulders A (2002) Tracking non-parameterized object contours in video. *IEEE Trans Image Process* 11(9):1081–1091
- Ning J, Zhang L, Zhang D, Yu W (2013) Joint registration and active contour segmentation for object tracking. *IEEE Trans Circuits Syst Video Technol* 23(9):1589–1597

- Osher S, Sethian JA (1998) Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 79(1):12–49
- Paragios N, Mellina-Gottardo O, Ramesh V (2004) Gradient vector flow fast geometric active contours. *IEEE Trans Pattern Anal Mach Intell* 26(3):402–407
- Santner J, Leistner C, Saffari A, Pock T, Bischof H (2010) PROST: parallel robust online simple tracking. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR), pp 723–730
- Suard F, Rakotomamonjy A, Bensrhair A (2006) Pedestrian detection using infrared images and histograms of oriented gradients. In: IEEE conference on intelligent vehicles, pp 206–212
- Tang F, Brennan S, Zhao Q, Tao H (2007) Co-tracking using semi-supervised support vector machines. In: IEEE 11th international conference on computer vision, pp 1–8
- Tekalp AM (1995) Digital video processing. Prentice Hall, New Jersey
- Xu C, Prince JL (1998) Snakes, shapes, and gradient vector flow. *IEEE Trans Image Process* 7(3):359–369
- Yilmaz A, Li X, Shah M (2004) Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans Pattern Anal Mach Intell* 26(11):1–6
- Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. *ACM Comput Surv* 38(4):1264–1291