Int J Comput Vis (2017) 122:116–148
DOI 10.1007/s11263-016-0959-5

Partially Camouflaged Object Tracking using Modified Probabilistic Neural Network and Fuzzy Energy based Active Contour

Ajoy Mondal[1] · **Susmita Ghosh**[2] · **Ashish Ghosh**[1]

Received: 23 September 2015 / Accepted: 20 September 2016 / Published online: 4 October 2016

Abstract Various problems in object detection and tracking have attracted researchers to develop methodologies for solving these problems. Occurrence of camouflage is one of such challenges that makes object detection and tracking problems more complex. However, less attention has been given to detect and track camouflaged objects due to complexity of the problem. In this article, we propose a tracking-by-detection algorithm to detect and track camouflaged objects. To increase separability between the camouflaged object and the background, we propose to integrate features (CIELab, histogram of orientation gradients and locally adaptive ternary pattern) from multi-cue (color, shape and texture) to represent a camouflaged object. A probabilistic neural network (PNN) is modified to construct an efficient discriminative appearance model for detecting camouflaged objects in video sequences. A large number of training patterns (many could be redundant) are reduced based on motion of the object in the modified PNN. The modified PNN makes the detection process faster and also increases the detection accuracy. Due to high visual similarity among the camouflaged object and the background, the boundary of camouflaged object is not well defined (i.e., boundary may be smooth and/or discontinuous). In this context, a robust fuzzy energy based active contour model using both global and local information is proposed to extract contour (boundary) of the detected camouflaged object for tracking. We show a realization of the proposed method and demonstrate its performance (both quantitatively and qualitatively) with respect to state-of-the-art techniques on several challenging sequences. Analysis of results concludes that the proposed technique can track camouflaged (fully or partially) objects as well as objects in various complex environments in a better way as compare to the existing ones.

Keywords Camouflage · Multi-cue · Probabilistic neural network · Fuzzy energy and visual similarity

1 Introduction

Moving object[1] 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; Li et al. 2013; Yang and Nevatia 2014; Zhang et al. 2015). It can be defined as the problem of estimating the trajectory of an object of interest in a video sequence. It is widely used in automated surveillance system (Yilmaz et al. 2006; Maggio and Cavallaro 2011), traffic monitoring (Yilmaz et al. 2006), vehicle navigation (Yilmaz et al. 2006), activity recognition and anomaly detection (Li et al. 2013), medical application and biological research (Mag-

[1] In this article, the words object and target are used interchangeably.

Communicated by T.E. Boult.

Electronic supplementary material The online version of this article (doi:10.1007/s11263-016-0959-5) contains supplementary material, which is available to authorized users.

✉ Ashish Ghosh
ash@isical.ac.in

Ajoy Mondal
ajoy.mondal83@gmail.com

Susmita Ghosh
susmitaghoshju@gmail.com

[1] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India

[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

gio and Cavallaro 2011; Bandouch et al. 2012), etc. It is a challenging task because of camouflaged objects, presence of noise in the image frames, complex object motion, partial or full object occlusion, background clutter, illumination variation, complex object shapes, large variation in object scale and orientation, etc (Yilmaz et al. 2006; Li et al. 2013; Lascio et al. 2013; Tang et al. 2014; Zhang et al. 2015). In literature, various tracking techniques (Avidan 2007; Ross et al. 2008; Santner et al. 2010; Kwon and Lee 2010; Babenko et al. 2011; Hare et al. 2011; Sevilla-Lara and Learned-Miller 2012; Jia et al. 2012; Kalal et al. 2012; Zhang et al. 2013; Yang and Nevatia 2014; Zhong et al. 2014; Yang et al. 2014; Zhang et al. 2015; Henriques et al. 2015) are developed to handle such problems and there exists no single algorithm which can successfully handle all these factors.

1.1 Review on Visual Tracking

Recently, visual object tracking has been posed as a tracking-by-detection problem (Avidan 2007), where statistical modeling is dynamically performed for detecting an object. Various statistical (generative and discriminative) appearance modelings are very popular in the literature to track an object. The main focus of generative appearance model is to accurately fit the data from the object class. In practice, it is very difficult to verify the correctness of the specified model (Li et al. 2013). It also provides local optimal solutions due to the (sub optimal) parameters. Optimal results may be obtained by introducing online updating mechanism. However, during updating, it incrementally learns only the representation of foreground object ignoring the influence of the background. As a result, they often suffer from the distractions caused by the background regions with similar appearance to the object class (Li et al. 2013). On the other hand, discriminative appearance models pose object tracking as a binary classification problem. Its aim is to find out a decision boundary which maximizes the separability between the object and its surroundings. To cope with changes in the object appearance as well as the background (due to both the object and the camera motions), the tracker needs to be adaptive. Thus, they can achieve effective and efficient performances (Li et al. 2013).

In literature, several techniques (Avidan 2007; Ross et al. 2008; Kwon and Lee 2010; Babenko et al. 2011; Hare et al. 2011; Sevilla-Lara and Learned-Miller 2012; Jia et al. 2012; Kalal et al. 2012; Zhang et al. 2013; Yang and Nevatia 2014; Zhong et al. 2014; Yang et al. 2014; Zhang et al. 2015; Henriques et al. 2015) have been developed based on discriminative, generative and combination of both discriminative & generative appearance models. Among them, VTD (Kwon and Lee 2010), ASLA (Jia et al. 2012), MTT (Zhang et al. 2013) and SCM (Zhong et al. 2014) are based on generative appearance model. Sparse representations are used in all these techniques. These representations are effective to account for appearance changes (e.g., partial occlusion, pose changes, etc). All these methods provide good results for video sequences having such complexities. However, local sparse representation [used in VTD (Kwon and Lee 2010), ASLA (Jia et al. 2012) and SCM (Zhong et al. 2014)] is more effective than global sparse representation [used in MTT (Zhang et al. 2013)]. Particle filter is used to locate the object in the next frame in all these methods. Incase of fast moving object, these methods produce unsatisfactory results.

On the other hand, several algorithms (Avidan 2007; Santner et al. 2010; Kwon and Lee 2010; Babenko et al. 2011; Hare et al. 2011; Sevilla-Lara and Learned-Miller 2012; Kalal et al. 2012; Zhang et al. 2013) based on discriminative appearance models have been developed for visual tracking. In all these approaches, the discriminative appearance models are considered as base classifiers to distinguish the object from its surrounding.

The ensemble tracker (Avidan 2007) poses tracking as a pixel based binary classification problem. Changes in the object appearance can be handled by adding new weak classifiers to the ensemble system. Although this method is able to distinguish the object from its background, it is unable to handle the changes in pose, scale and orientation of the object, full & long time occlusion and background clutter. To reduce the drift problem, Babenko et al. (2011) proposed an online multiple instance learning for object tracking. This technique can take care of partial occlusion, scale change; but is unable to cope with full or long time occlusion, large variation in scale and orientation.

In these tracking-by-detection techniques, appearance models are adaptive to take into account changes caused by object motion, illumination variation and occlusion. From the estimated object location, a set of binary labeled training samples is generated to update the classifier online. However, the generated training set is not always correct (with respect to the object and the background classes), and reduces tracking accuracy. To solve this problem, Hare et al. proposed structure output tracking with kernels (Struck) (Hare et al. 2011) which integrates learning and tracking, avoiding the need for ad-hoc update strategies. In Kalal et al. (2012), proposed a novel tracking algorithm with appearance learning approach which combines nearest neighbor (NN) classifier with median of particle optical flow tracker. It is able to deal with appearance change, illumination variation, pose change, scale variation and background clutter.

Most of the trackers use high-level appearance structures or low-level cues for detecting and tracking objects. In Yang et al. (2014), proposed a discriminative appearance model based on superpixels, to distinguish the target and

the background with mid-level cues. Target location in the next frame is estimated by maximizing a posterior probability based on the object background confidence map. It can handle heavy occlusion and recover the tracker from drift problem. Though all the discussed methods are able to track object in several complex environments, they are computationally more expensive. In Henriques et al. (2015), proposed kernelized correlation filter (KCF) based on fast Fourier transforms to distingue object from its surroundings. It is faster than all other existing tracking algorithms with better performance.

From the above discussion, it is highlighted that the behaviors of all these techniques are not explicitly mentioned to track camouflaged objects. Occurrence of camouflage makes object tracking more complex (Copeland et al. 1997; Singh et al. 2013).

1.2 Review on Feature based Robust Visual Representation of Target (Object)

Visual representation reflects the statistical characteristics of object appearance (Li et al. 2013; Maggio and Cavallaro 2011). It plays an important role in object detection and tracking scenario. Various features are proposed or considered to represent objects for detection and tracking (Challa et al. 2011; Maggio and Cavallaro 2011). Features generated from single-cue (like color or shape or texture or motion) is considered in detection and tracking techniques (Li et al. 2013; Challa et al. 2011; Maggio and Cavallaro 2011). On the other hand, features generated from multi-cue (like color, shape, texture and motion) have also been widely used in detection and tracking system (Wang and Yagi 2008; Ning et al. 2009; Lan et al. 2014). It is observed that multi-cue is better than single-cue in the context of object detection and tracking (Wang and Yagi 2008; Ning et al. 2009; Lan et al. 2014). In Wang and Yagi (2008), color and shape features are integrated to track objects in mean shift framework. The authors considered color histogram in three color spaces RGB, HSV, normalized rg and histogram of gradient orientation to describe shape. This method is time consuming and is unable to track objects under several conditions like illumination variation, similar background objects (clutter) and camouflaged object. To track objects in cluttered background, Ning et al. (2009) integrated histograms of color and texture in mean shift framework. It provides better results than the mean shift tracking using only color histogram. However, this method is not robust for several conditions like illumination variation, pose variation and camouflaged objects; and is also time consuming. In Lan et al. (2014), particle filter based multi-cue (e.g. color and shape) tracking framework is proposed to increase the tracking performance in several complex environments like illumination variation, occlusion and pose variation. Basically, in all these methods, histograms generated from multi-cue are integrated for tracking objects.

1.3 Review on Camouflaged Object Detection and Tracking

Camouflage is an attempt to conceal the signature of a target into the background (Copeland et al. 1997; Gretzmacher et al. 1998). In other words, camouflage is the ability of prey to hide themselves from predators by changing their body patterns, texture and coloration as per environment's texture. Work related to camouflage can be roughly divided into two major areas: (i) camouflage assessment and design, (ii) camouflage detection (Copeland et al. 1997; Gretzmacher et al. 1998; Singh et al. 2013. Camouflage design system artificially creates camouflage images (Du et al. 2012); whereas, camouflage detection is used to extract the target from the background. It basically discriminates the camouflaged foreground object from background (Copeland et al. 1997; Gretzmacher et al. 1998; Singh et al. 2013. Camouflage detection systems have many potential applications including (i) preserving wildlife, (ii) target detection in the battle field, (iii) defect detection during manufacturing, (iv) identification of duplicate products during logistics, etc.

In literature, various techniques (Harville et al. 2001; Boult et al. 2001; KaewTrakulPong and Bowden 2003; Huang and Jiang 2005; Chandesa et al. 2009; Conte et al. 2009; Hou and Li 2011; Malathi and Bhuyan 2013) are developed to detect camouflaged objects. Harville et al. (2001)) proposed a foreground segmentation technique using both color and depth information to detect camouflaged objects. Information loss occurs due to the projection of 3D scene into 2D picture. Here, depth information is considered to reduce the information loss for 3D to 2D projection. Use of depth information increases the detection accuracy and also increases the computational cost. In this direction, Boult et al. (2001) developed a camouflaged object detection technique using background subtraction with consideration of two thresholds. Here, larger threshold is considered to detect pixels which are certainly in the foreground. Whereas, smaller threshold is used to detect uncertain pixels (i.e., pixels which are part of either background or camouflaged object). Then quasi connected component is taken into consideration to detect camouflaged target. The detection accuracy is highly dependent on threshold values. Selection of proper threshold value itself is a problem. For slow moving objects, this method fails to detect objects. In KaewTrakulPong and Bowden (2003), Pong and Bowden proposed a technique with the help of stochastic process to handle camouflage. This algorithm assumes that camouflage occurs when the new observation can not be associated with an existing task. Basically, performance of this method depends on color and

motion of the object. Here, color is used to distinguish the camouflaged object and its surrounding whereas motion is considered to separate out moving objects and static parts of the frame. Hence, both these information help to detect moving camouflaged object. Sometimes, this method may fail for an object which is very much similar to its background and background contains more non-static objects.

Huang and Jiang (2005) devised a method to track camouflaged objects using sequential execution of two techniques. An iterative region consolidation operator is used to fill the gaps introduced by camouflage. Then active contour model is built during tracking to capture the actual shape of the target. Basically, performance of this method relies on the inter-frame difference. If the object has slow motion, it is very difficult to localize the object using iterative weighted region consolidation operator. Hence, tracking may fail for sequences containing slow moving and uniform colored objects. To detect camouflaged object, Chandesa et al. (2009) proposed an algorithm based on particle filter, in which Gaussian mixture model of particle distribution is considered to investigate the effect of camouflage on the particle set representing the object. This method works well on occluded object but not for camouflaged object. Though this method works well it needs object information (a priori) to execute the algorithm. Conte et al. (2009) proposed an algorithm to detect partially camouflaged people. They have used background subtraction technique to detect different parts of people. Then grouping of different parts is performed based on the shape of targets. This method is unable to provide satisfactory results for objects other than human. Also it is not always possible to extract the actual shape of the object by combining different detected parts of the object.

In this direction, Hou and Li (2011) developed an algorithm to track mobile camouflaged objects based on optical flow model. Optical flow model is used to detect motion patterns corresponding to the camouflaged object and the background. Based on magnitude and location of the optical flow, motion patterns are clustered to detect the camouflaged object. Thereafter, Kalman filter is considered to improve the detection accuracy. However, the accuracy of this technique highly depends on the result of optical flow. For slow moving objects and objects with camera motion, this method fails to provide good results. Recently, Malathi and Bhuyan (2013) developed a background subtraction scheme using multi-view information to detect camouflaged object. Here, pixels corresponding to the background are quantized into codebooks which are sufficient to represent a background. Codebooks extracted from each frame corresponding to each camera are combined to construct a background and the objects are detected by subtracting constructed background from the frame. This method provided good results for occluded objects however, unable to provide satisfactory results for camouflaged objects.

1.4 Review on Fuzzy Active Contour

Kass et al. (1988) introduced the active contour (snake) model (ACM) to detect and track objects. The main idea behind the ACM is based on deformation of an initial curve so that it evolves towards the object boundary under some constraints. It has the ability to generate closed parametric curve (Cremers et al. 2007). However, it has some drawbacks such as: (i) segmentation results highly depend on initial contour position, (ii) it is easily deteriorated by noise present in the given image. Several approaches (Caselles et al. 1997; Yezzi et al. 1997; Cohen and Kimmel 1997; Li et al. 2010) based on ACMs have been developed to overcome such drawbacks to some extend.

Active contour and its variants are basically "edge based" models which use image gradient to attract the contour towards the object boundaries. As these models rely on edge information, sometimes it may fail to extract contour when the gradient of the object boundaries is not well defined for noisy, blurred or even discontinuous edged images.

Different from "edge based" active contours, "region based" active contour models (Chan and Vese 2001; Lankton and Tannenbaum 2008; Krinidis and Chatzis 2009; Zhang et al. 2010; Tran et al. 2014; Wu et al. 2015) using region information like intensity, color and texture features are also introduced in the literature.

In general, the "region based" models use image information from not only the evolving contours but also from the image statistics inside and outside the contour. Chan and Vese (2001) proposed an active contour (Chen–Vese) model which depends on the region based energy function, inspired by Mumford-Shush function (Mumford and Shah 1989). In this model, two regions inside and outside the contour in the image are assumed to be homogeneous. Here, the global image (statistics) information is used in the energy function to transform it into the level set formulation. The Chen-Vese model as well as other region based models are able to deal with images having weak boundaries. In addition, they are less sensitive to the initial contour position as compared to the "edge based" model.

Some modifications in the Chen–Vese model have been proposed in the literature (Lankton and Tannenbaum 2008; Krinidis and Chatzis 2009; Zhang et al. 2010; Talu 2013) mainly to reduce the high computational cost. In Krinidis and Chatzis (2009), Krinidis and Chatzis proposed a "region based" active contour model to segment images. This model uses global image information which is able to segment images with blur and discontinuous boundaries and is robust to the initial position of the contour. In this model, they assumed that the image is approximated with two regions of piece-wise constant intensities inside and outside the contour. However, this model is unable to produce good

segmentation for images containing intensity inhomogeneity or non-homogeneity.

In fact, real life images contain intensity inhomogeneity or non-homogeneity which makes the segmentation a more challenging problem. To segment such kinds of images, region based active contours using local image (statistics) information are proposed in the literature (Lankton and Tannenbaum 2008; Zhang et al. 2010). However, all these models are unable to produce good results for highly blurred, noisy and intensity inhomogeneity images.

To overcome such drawbacks, "region based" active contours using both global and local information are developed in the literature (Krinidis and Krinidis 2012; Shyu et al. 2012). In both these methods, local information, based on only the spatial distance is considered and they are not robust to high level of blur, noise and intensity inhomogeneity or non-homogeneity present in the images.

1.5 Contribution and Basic Workflow of the Proposed Algorithm

In this article, a tracking-by-detection algorithm is proposed to detect and track camouflaged objects in video scenes. The proposed work has three main contributions:

(i) Integration of (CIELab, histogram of orientation gradients and locally adaptive ternary pattern) features from multi-cue (color, shape and texture) to increase separability between the camouflaged object and the background.
(ii) A discriminative appearance model is proposed by modifying a probabilistic neural network to detect camouflaged objects.
(iii) A fuzzy energy based active contour model using both the global and local information is proposed to track camouflaged objects.

When visual characteristics (like color or texture or shape or all of them) of the objects are too similar to the background, camouflage occurs (Singh et al. 2013). The objective of a camouflage detection system is to separate the camouflaged object from background. In such cases, features generated from single-cue (like color or texture or shape or motion) is not sufficient to extract camouflaged objects from background because the underlying phenomenon for occurrence of camouflage is not known. In this context, integration of features generated from multi-cue (color, texture, shape, motion, etc.) may increase the separability between the camouflaged objects and background. As our goal is to detect and track the camouflaged objects, (CIELab, histogram of orientation gradients and locally adaptive ternary pattern) features from multi-cue (color, shape and texture) are integrated here to increase the separability between the camouflaged objects and the background.

The probabilistic neural network (PNN) (Specht 1990) developed by Specht has became an efficient tool for solving many pattern classification and computer vision problems (Musavi et al. 1994; Yan et al. 2011; Hao et al. 2007) due to its easy, non iterative training process and statistical foundation in Bayesian estimation theory. It has the ability to produce good classification results in complex classification problems. Recently, object tracking is posed as a binary classification (single object) problem (Avidan 2007). Occurrence of camouflage makes object detection and tracking problems more complex. In such complex scenario, to detect an object, PNN may be considered.

A discriminative appearance model is proposed by modifying a probabilistic neural network to detect camouflaged object in the target candidate frame. In PNN, the pattern layer contains all training patterns which may be redundant and it increases the network structure. This large network structure introduces two problems, (i) high computational cost is incurred to classify an unknown pattern because the amount of computation necessary to classify an unknown pattern is proportional to the size of the network (Traven 1991; Mao et al. 2000), (ii) due to the large network architecture, classifier is highly biased to the training patterns and exhibit poor generalization capacities to unseen data (i.e. overfitting problem) (Bishop 1995; Traven 1991; Mao et al. 2000). To solve these problems of PNN, one algorithm based on motion of the object is proposed to select training patterns (subset of training patterns). The proposed training pattern selection technique makes the classification process faster due to the reduced network architecture and increases the classification accuracy due to the consideration of training subset with local information instead of global information. Therefore, the proposed PNN provides better detection of the camouflaged objects.

Since camouflaged object is visually very similar to the background, boundary of the camouflaged object is not well defined (i.e., the boundary may be smooth and /or discontinuous). Due to motion of the object and the camera, boundary of the object in the frames extracted from a video sequence is blurred. However, the object regions in real life video sequence usually contain high intensity/color inhomogeneity or non-homogeneity. To extract boundary of such kinds of objects in images, fuzzy active contour models are proposed in literature (Krinidis and Chatzis 2009; Wu et al. 2015). All these models use only global information and are unable to extract contours in such environment due to high visual similarity between the camouflaged object and the background. In this article, a robust fuzzy energy based active contour model using both global and local information is proposed to extract the contour (boundary) of the detected

camouflage object for tracking. The contour is evolved iteratively with object detection results produced by the modified PNN as an initialization for the proposed fuzzy active contour model. Finally, camouflaged object is tracked by its contour.

To test the effectiveness of the proposed algorithm, investigation was carried out on 200 challenging video sequences containing (full or partial) camouflaged object, illumination variation, scale change, rotation, shape deformation, background clutter, pose change, partial or full occlusion, etc. Tracking results of the proposed method are compared with six state-of-the-art tracking algorithms. Evaluation measures such as average center location error, average tracking accuracy (area overlap), precision plot and success plot are considered to evaluate the performance of the proposed technique. From the analysis (both quantitative and qualitative) of results, it is found that the proposed method performs favorably against state-of-the-art techniques on several sequences having different complexities.

Rest of the article is organized as follows. Section 2 describes the proposed technique for camouflaged object tracking. Experimental results and analysis are presented in Sect. 3. Discussion and future work are described in Sect. 4. Conclusive remarks are put in Sect. 5.

2 Proposed Technique for Partially Camouflaged Object Tracking

In this article, a tracking-by-detection algorithm has been developed to detect and track camouflaged objects. Color (CIELab), shape (histogram of orientation gradients) and textural (locally adaptive ternary pattern) features are integrated to represent the camouflaged object in the target frame. A discriminative appearance model is proposed by modifying a PNN to detect camouflaged objects in the target candidate frame. Finally, camouflaged objects are tracked in the target candidate frame by the proposed fuzzy energy based active contour model. Figure 1 displays flowchart of the proposed tracking algorithm. Each part of the proposed tracking algorithm is described in the following subsections.

2.1 Multi-cue Feature based Robust Visual Representation

Camouflage occurs due to high visual similarity (with respect to color or texture or shape or all of them) among an object and its surroundings (Singh et al. 2013). The objective of a camouflaged object detection technique is to extract the camouflaged object from the background. In this context, robust features are needed to detect the camouflaged object from its surroundings. As the objective of the present article is to detect and track camouflaged objects, robust features

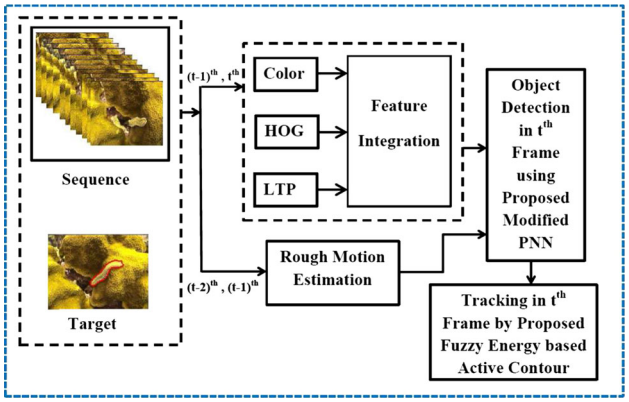


Fig. 1 Flowchart of the proposed tracking algorithm

are needed to increase the separability between the camouflaged object and the background. In this article, color (CIELab), shape (histogram of orientation gradients) and textural features (locally adaptive ternary pattern) are integrated to increase the separability between the camouflaged object and the background.

2.1.1 Color Feature

In this article, color of the object is represented by CIELab color space. It is perceptually uniformed and is better to separate lightness from color compared to other color spaces (Du et al. 2012). It also indicates proper changes in the direction of human color perception. Its components are lightness of the color (L^*) and two color-opponent dimensions (a^* and b^*). The nonlinearity among L^*, a^* and b^* are intended to mimic the nonlinear response of the human eye (Du et al. 2012). Furthermore, uniform changes of components in the $L^*a^*b^*$ color space aim to correspond to uniform changes in the perceived color.

2.1.2 Shape Feature based on Histogram of Orientation Gradients

It has been widely used as a key feature 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). A shape is described by an orientation histogram which is computed based on image derivatives in x and y directions. The gradients at the point $\mathbf{X}$ in the image frame I can be calculated by convolving the Sobel mask (Gonzalez and Woods 2008) with the image. The gradients

are defined as: $I_x(\mathbf{X}) = D_x * I(\mathbf{X})$ and $I_y(\mathbf{X}) = D_y * I(\mathbf{X})$, where D_x and D_y are Sobel masks in x and y directions, respectively. The magnitude of the gradient at the point $\mathbf{X}$ is computed as $G(\mathbf{X}) = \sqrt{I_x^2(\mathbf{X}) + I_y^2(\mathbf{X})}$. The orientation of gradient of that pixel is given by $\theta(\mathbf{X}) = \arctan\left(\frac{I_y(\mathbf{X})}{I_x(\mathbf{X})}\right)$. The orientations are quantized into 8 bins. Each orientation is weighted and assigned to one of the two adjacent bins according to its distance from the bin centers. For more details one can go through Dalal and Triggs (2005).

2.1.3 Locally Adaptive Texture Feature

Various statistical descriptors have been proposed to measure image texture in literature (Haralick et al. 1973; Ojala et al. 1996; Tan and Triggs 2010). Among them, local ternary pattern (LTP) (Tan and Triggs 2010) is quite popular to solve various computer vision problems. As visual appearance of the camouflaged target is very similar to the background, local adaptive ternary pattern (LATP) (Akhloufi and Bendada 2010) is considered to describe image texture in this article. LTP is an extension of local binary pattern (LBP) (Ojala et al. 1996). Here, instead of central pixel value of a neighborhood (small image region), thresholding is done based on user defined value Th. Any pixel value within a neighborhood laying within the interval of $[-Th,\ Th]$ of the central pixel is assigned a value 0. The pixel values above Th when compared to the central pixel are assigned to 1 and ones below $-Th$ when compared to the central pixel are assigned to -1. Thus, LTP of a pixel X is computed as

$$LTP(X) = \begin{cases} 1 & if \quad I(Y) - I(X) \geq Th \\ 0 & if \quad -Th \leq I(Y) - I(X) \leq Th \\ -1 & if \quad I(Y) - I(X) \leq -Th \end{cases}, \quad (1)$$

where Th is a user defined threshold value, $I(\mathbf{X})$ and $I(\mathbf{Y})$ are intensity values of the central pixel and neighborhood pixels, respectively. LTP codes are more resistant to noise, but no longer strictly invariant to gray-level transformations. The LTP encoding procedure is illustrated in Fig. 2. Here the threshold Th was set to 5, so the tolerance interval is [95, 105]. Each ternary pattern is split into its positive and negative halves as illustrated in Fig. 3; these are called upper and lower patterns, respectively. However, performance of LTP depends on threshold value. Akhloufi and Bendada

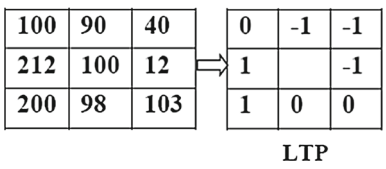


Fig. 2 Illustration of the basic LTP operator

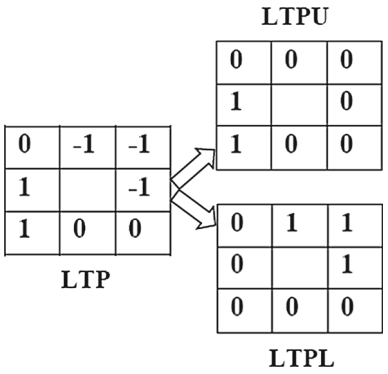


Fig. 3 Splitting a LTP code into positive (LTPU) and negative (LTPL) codes

(2010) proposed a locally adaptive ternary pattern (LATP) for multi-spectral face recognition. LATP code is extracted as

$$LATP(\mathbf{X}) = \begin{cases} 1 & if \quad I(\mathbf{Y}) \geq (\mu_\eta + k\sigma_\eta) \\ 0 & if \quad |I(\mathbf{Y})| < (\mu_\eta + k\sigma_\eta) \\ -1 & if \quad I(\mathbf{Y}) \leq (\mu_\eta - k\sigma_\eta), \end{cases} \quad (2)$$

where μ_η and σ_η refer to the mean and standard deviation of pixel values in the neighborhood and k is a user defined constant.

In this article, the normalized color (CIELab), shape (HOG) and texture (LTPU and LTPL) features are integrated to generate d-dimensional feature vectors. Then each frame is transformed into d-dimensional features space. Now each pixel $\mathbf{X}^t$ of the tth frame is considered as a pattern $I\left(\mathbf{x}^t\right) \in \Re^d$ and is represented by $I\left(\mathbf{x}^t\right) = \left(f_{x_1^t}, f_{x_2^t}, .., f_{x_i^t}, .., f_{x_d^t}\right)$.

Figures 4, 5 and 6 highlight the object detection results in the target candidate frame using the proposed PNN (discussed in the next section) with smoothness parameter $\sigma = 0.1$ for all cases with different features and their various combinations for three video sequences. From Fig. 4, it is observed that texture feature individually is better than color and shape features for detecting camouflaged object in Clip152 sequence. In case of combination of two features, color & texture provide better results. However, combination of all features color, shape and texture provides best results among individual features and their other combinations for Clip152 sequence. On the other hand, for Clip730 sequence, color feature (individually) produces better detection result than shape and texture features. In case of combination of features, color & texture produce better results than other combinations of two features (i.e., color & shape and texture & shape). However, combination of color, shape and texture provides best results among individual features and their other combinations. In case of Clip187 sequence, shape feature is individually better than color and texture features. Among combinations of two features, texture & shape pro-

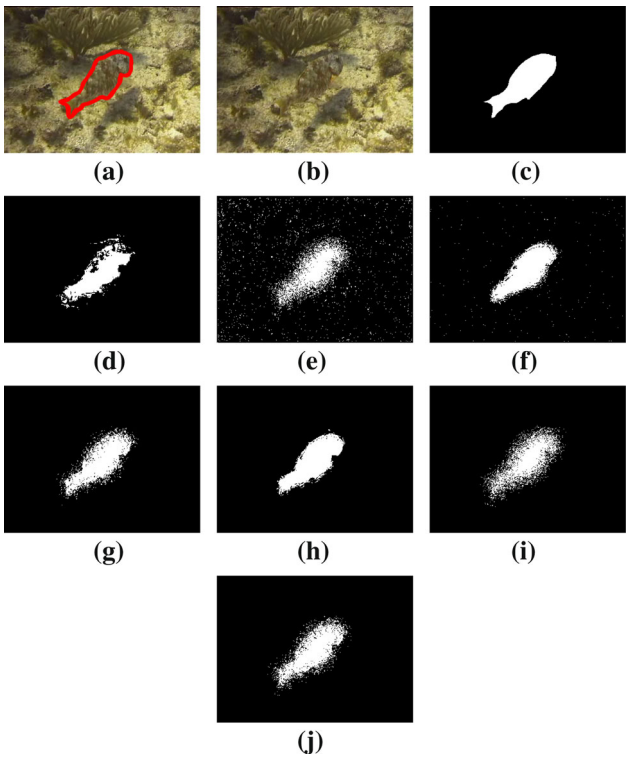


Fig. 4 Clip152 sequence **a** target frame; **b** target candidate frame; **c** ground truth. Object detection using the proposed modifying PNN ($\sigma = 0.1$) with consideration of **d** color feature; **e** shape feature; **f** texture feature; integration of **g** color and shape features; **h** color and texture features; **i** shape and texture features and **j** color, shape and texture features

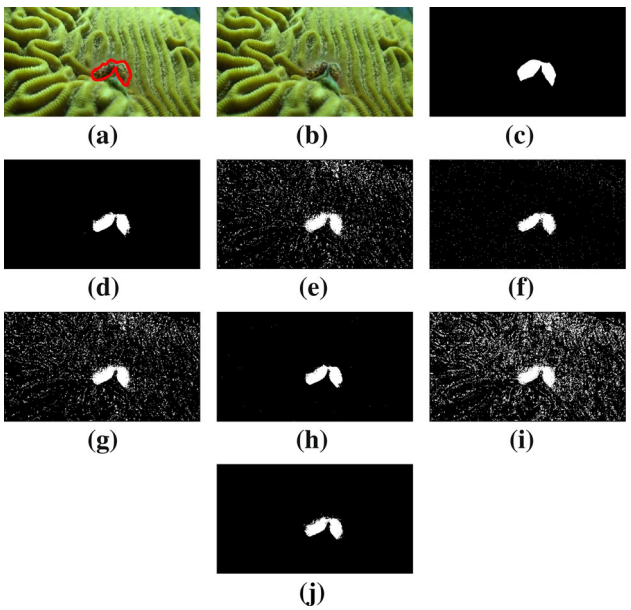


Fig. 5 Clip730 sequence **a** target frame; **b** target candidate frame; **c** ground truth. Object detection using the proposed modifying PNN ($\sigma = 0.1$) with consideration of **d** color feature; **e** shape feature; **f** texture feature; integration of **g** color and shape features; **h** color and texture features; **i** shape and texture features and **j** color, shape and texture features

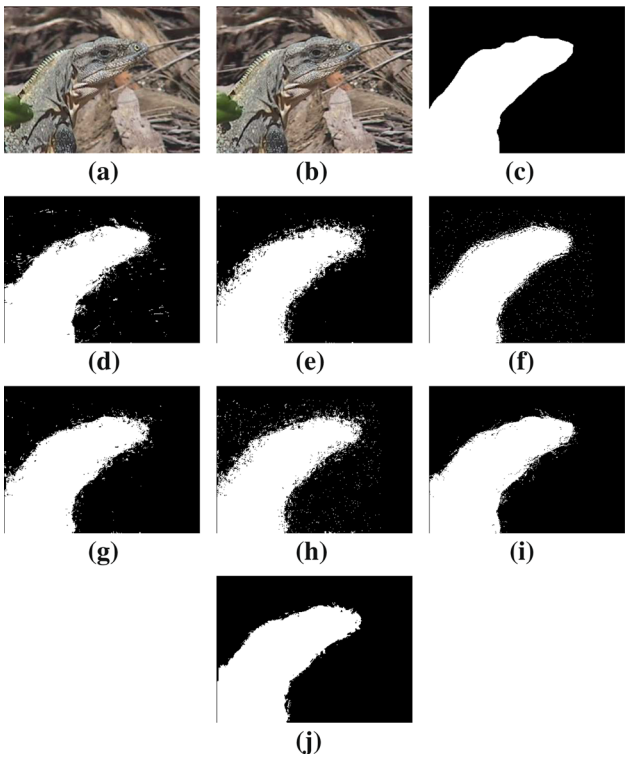


Fig. 6 Clip187 sequence **a** target frame; **b** target candidate frame; **c** ground truth. Object detection using the proposed modifying PNN ($\sigma = 0.1$) with consideration of **d** color feature; **e** shape feature; **f** texture feature; integration of (**g**) color and shape features; **h** color and texture features; **i** shape and texture features and **j** color, shape and texture features

vide better results. However, combination of color, shape and texture provides the best result. From these figures, it is also observed that some features and their combinations produce better detection results for some sequences and for other sequences, these are unable to provide good results. On the other hand, integration of color, shape and texture features provides better object detection results than individual features and their other combinations for the camouflaged sequences. These examples highlight the superiority of integration of features from multi-cue for detecting camouflaged objects. Quantitative measures for the importance of the proposed features integration for detecting camouflaged objects are provided in Table 1. It shows that the average object detection accuracy (Metz 1978) of different considered sequences using different features and their various combinations. This table also highlights that the proposed integration of features from multi-cue, provides better results.

2.2 Rough Estimation of Object Motion

In this article, object motion between two consecutive frames is roughly estimated in a similar way as discussed in Mondal et al. (2014). Suppose, contours of the camouflaged object in

Table 1 Average object detection accuracy based on various features

Different features and their combinations							
Sequences	C	S	T	CS	CT	TS	CST
Clip152	0.9800	0.9589	*0.9819*	0.9841	**0.9869**	0.9732	***0.9908***
Clip730	*0.9769*	0.9200	0.9719	0.9121	**0.9779**	0.8469	***0.9779***
Clip187	0.6657	*0.6717*	0.6707	0.9004	0.8278	**0.9699**	***0.9745***

C color; *S* shape; *T* texture; *CS* color and shape; *CT* color and texture; *TS* texture and shape; *CST* color, shape and texture; *italics* indicates best among individual features; *bold* indicates best among combination of two features; *bolditalics* indicates best among all individual features and their combinations

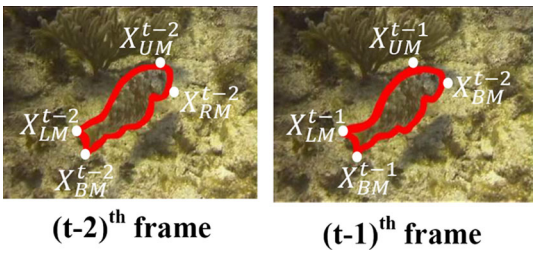


Fig. 7 Estimation of rough motion between $(t-2)$th and $(t-1)$th frames

two consecutive ($(t-2)$th and $(t-1)$th) frames are available and motion of the camouflaged object between $(t-1)$th and tth frames need to be estimated. Let $\mathbf{X}_{BM}^{t-2}, \mathbf{X}_{BM}^{t-1}, \mathbf{X}_{UM}^{t-2}, \mathbf{X}_{UM}^{t-1}$, $\mathbf{X}_{LM}^{t-2}, \mathbf{X}_{LM}^{t-1}, \mathbf{X}_{RM}^{t-2}$ and $\mathbf{X}_{RM}^{t-1}$ be the bottom most, upper most, left most and right most points on the contours of the object in the $(t-2)$th and the $(t-1)$th frames, respectively (see Fig. 7).

Motion (M) of the object between these two frames is roughly estimated as:

$$M = \max\{\|\mathbf{X}_{BM}^{t-2} - \mathbf{X}_{BM}^{t-1}\|,\ \|\mathbf{X}_{UM}^{t-2} - \mathbf{X}_{UM}^{t-1}\|, \|\mathbf{X}_{LM}^{t-2} - \mathbf{X}_{LM}^{t-1}\|, \|\mathbf{X}_{RM}^{t-2} - \mathbf{X}_{RM}^{t-1}\|\}, \quad (3)$$

where $\|.\|$ is Euclidean distance. Since movement of the object is not constant throughout the sequence, motion of the object between the $(t-2)$th and $(t-1)$th frames is not similar for the $(t-1)$th and tth frames. Hence, motion of the object between the $(t-1)$th and the tth frames is modified as

$$\bar{M} = \gamma * M, \quad (4)$$

where $\gamma \in \mathbb{R}$ is a camera or object deformation parameter.

2.3 Camouflaged Object Detection Using Proposed Modified Probabilistic Neural Network based Discriminative Appearance Model

Probabilistic neural network (PNN) (Specht 1990) developed by Specht, has became an efficient tool for solving many pattern classification and computer vision problems (Musavi et al. 1994; Yan et al. 2011; Hao et al. 2007) due to its easy, non iterative training process and statistical foundation in Bayesian estimation theory. PNN consists of four (input, pattern, summation and decision) layers with neurons. Pattern layer of PNN consists of all training samples including redundant samples. Inclusion of redundant training samples may lead to a large architecture. The large network structure introduces two problems: (i) high computational cost is required to classify an unknown sample because the amount of computation necessary to classify an unknown sample is proportional to the size of the network (Traven 1991; Mao et al. 2000), (ii) due to the large network architecture, classifier is highly biased to the training samples and exhibit poor generalization capacities to the unseen data (i.e. overfitting problem) (Bishop 1995; Traven 1991; Mao et al. 2000). Some algorithms for reduction of redundant training samples have been proposed in literature (Burrascano 1990; Traven 1991; Raghu and Yegnanarayana 1998; Mao et al. 2000). In this work, we have also proposed a technique based on motion of the object between two consecutive frames to reduce redundant training samples for classifying unlabeled samples. Here, instead of considering the whole training set, a subset (generated based on motion of object) of training set is considered to classify each unlabeled sample. For different unlabeled samples, different training subsets (consist of equal number of training samples) are generated. As PNN has non iterative training process and consideration of training subset reduces the architecture of modified PNN, the proposed discriminative appearance model takes less computational time than the original PNN to classify the unlabeled samples. Moreover, the proposed modified PNN considers local information instead of global information to classify unlabeled samples, thereby increasing the classification accuracy.

2.3.1 Generation of Training and Testing Samples

To construct an appearance model for both the object and the background, prior knowledge about the label of each pixel (or pattern or sample) can be learned from a training (i.e., tar-

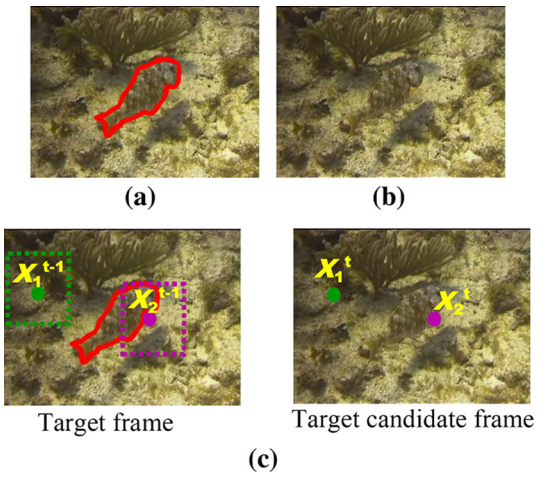


Fig. 8 **a** Target frame; **b** target candidate frame; **c** generation of training subset for each unlabeled sample of Clip152 sequence

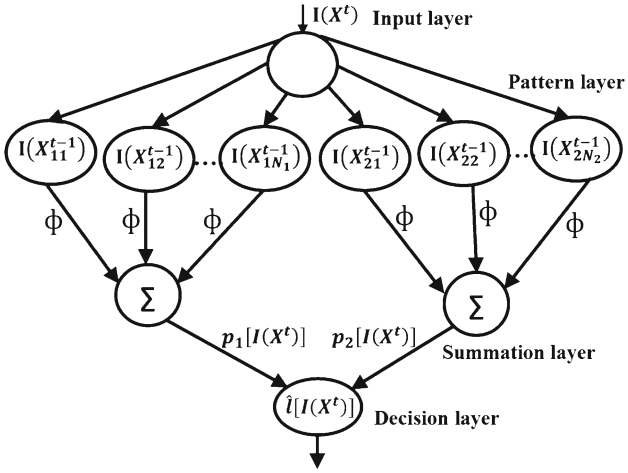


Fig. 9 Architecture of the modified PNN

get)[2] frame. By convention, the pixels within the contour of the object are termed as positive pixels and pixels outside the contour are treated as negative pixels. Positive and negative pixels in d-dimensional feature space are termed as positive and negative samples, respectively. Training set consists of these positive and negative samples. Figure 8a shows the target frame where the object is bounded by its (red colored) contour.

Test set is generated from the test (i.e., target candidate)[3] frame. All pixels of the test frame in d-dimensional feature space are called test samples. Figure 8b displays the target candidate frame from which the moving object needs to be detected.

In this investigation, to classify each unlabeled sample in the tth frame, instead of considering the whole training set generated from the $(t-1)$th frame, a subset of training set is considered Mondal et al. (2014). The training subset for each test sample is generated with the labeled samples within a spatial neighborhood (window) at the corresponding location in the $(t-1)$th frame of an unlabeled sample in the tth frame. Figure 8c depicts the generation of training subset for each unlabeled sample in the target candidate frame. Here, to classify unlabeled sample, corresponding to pixel $\mathbf{X}_1^t$ in the tth frame, the training subset is generated from the green colored dotted rectangle in the $(t-1)$th frame. $\mathbf{X}_1^{t-1}$ in the $(t-1)$th frame is the corresponding pixel of $\mathbf{X}_1^t$ at the tth frame. For different unlabeled samples, different training subsets are generated. A heuristic approach based on motion ($\bar{M}$) of the object is proposed to generate the training subset. The neighborhood (window) size is taken as $\bar{M} \times \bar{M}$.

[2] In this article, training, target, $(t-1)$th and previous frame are used interchangeably.

[3] In this article, test, target candidate, tth and current frame are used interchangeably.

Use of local training subset reduces the number of neurons in the pattern layer of the modified PNN. Smaller number of nodes in the pattern layer minimizes the architecture of the modified PNN. Figure 9 shows the architecture of the modified PNN. It consists of several interconnected neurons organized in successive layers. Input layer does not perform any computation, it only distributes the input patterns to the neurons in the pattern layer. After receiving input (unlabeled) pattern $I\left(\mathbf{x}^t\right)$ from the input layer, neuron (corresponding to labeled patterns) $I\left(\mathbf{x}_{ij}^{t-1}\right)$ at the pattern layer computes its output as

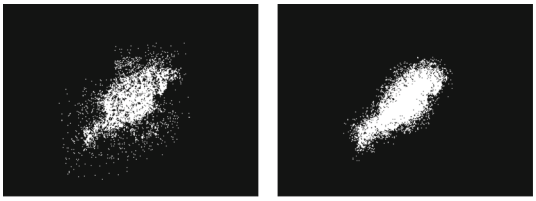
$$\phi_{ij}\left[I\left(\mathbf{X}^t\right)\right] = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma^d}\left[e^{-\frac{\left[I(\mathbf{x}^t)-I\left(\mathbf{x}_{ij}^{t-1}\right)\right]^T\left[I(\mathbf{x}^t)-I\left(\mathbf{x}_{ij}^{t-1}\right)\right]}{2\sigma^2}}\right], \tag{5}$$

where d is the dimension of the feature vector representing a pattern, $I\left(\mathbf{X}_{ij}^{t-1}\right)$ is the jth training pattern of the ith class in the $(t-1)$th frame, $I\left(\mathbf{X}^t\right)$ is the test pattern in the tth frame, σ is the smoothing parameter and $\mathbf{x}_{ij}^{t-1} \in \eta_{\mathbf{x}^t}^t$, where $\eta_{\mathbf{x}^t}^t$ is the temporal neighborhood of $\mathbf{x}^t$.

The neurons at the summation layer compute the maximum likelihood of the unlabeled pattern $I\left(\mathbf{x}^t\right)$ being classified into $l_i;\ i = 1,\ 2$ by averaging the output of all neurons which belong to the same class. Therefore

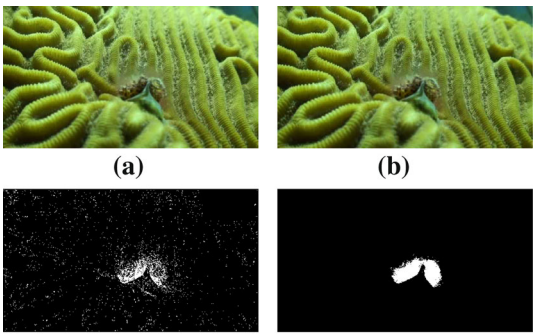
$$p_i\left[I\left(\mathbf{X}^t\right)\right] = \frac{1}{N_i}\sum_{j=1}^{N_i}\phi_{ij}\left[I\left(\mathbf{X}^t\right)\right], \tag{6}$$

where N_i denotes the total number of training samples of class $l_i;\ i = 1,\ 2$. The neuron at the decision layer classifies the pattern $I\left(\mathbf{x}^t\right)$ according to Bayes' decision rule based on output of all neurons in the summation layer. Thus, class label of the pattern $I\left(\mathbf{x}^t\right)$ is estimated as



(a) (b)

Fig. 10 Detection of camouflaged object in the target candidate frame of Clip152 sequence using **a** PNN (considering whole training set) and **b** the proposed modified PNN (considering local training subset)



(a) (b) (c) (d)

Fig. 11 **a** Target frame; **b** target candidate frame. Detection of camouflaged object in the target candidate frame of Clip730 sequence using **c** PNN (considering whole training set) and **d** the proposed modified PNN (considering local training subset)

$$\hat{l}\left[I\left(X^t\right)\right] = \max\left\{p_i\left[I\left(X^t\right)\right]\right\},\ i = 1, 2. \tag{7}$$

All pixels of the tth frame are thus classified as either object or background; hence, detects the object.

Figure 10a shows the object detection result of Clip152 sequence using PNN with consideration of all training samples and it takes 2.77 seconds to detect object in a single frame. Figure 10b displays the detected object of Clip152 sequence using the modified PNN with consideration of local training subset and it takes 0.06 seconds to detect the object in a single frame. Similarly, Fig. 11c, d display the detected camouflaged object in the target candidate frame of Clip730 sequence with consideration of all training samples and local training subset; and take 4.46 and 0.07 seconds for executing a single frame, respectively. Both these examples highlight the efficiency of the proposed discriminative appearance model by modifying PNN for detecting the camouflaged objects in the target candidate frame. Table 2 shows the detection accuracy based on (whole and local) training set. This table also highlights that the modified PNN (consideration of local training set) produces better result than PNN (consideration of whole training set) in camouflaged condition. Description of these datasets are given in Table 3. To extract the contour of the camouflaged object or to track such an object, a fuzzy energy based active contour with global and local information is proposed. The proposed fuzzy energy based active contour is evolved with the detected object by the modified PNN as an initialization. Object tracking using the proposed fuzzy energy based active contour model is explained in the following subsection.

Table 2 Average (object detection) accuracy based on training patterns

Sequences	All training patterns	Selected training patterns
Clip152	0.9641	***0.9820***
Clip730	0.9687	***0.9939***

BoldItalics indicates best results

2.4 Camouflaged Object Tracking using the Proposed Global and Local Fuzzy Energy based Active Contour Model

2.4.1 Existing Global Fuzzy Energy based Active Contour Model

Fuzzy active contour model proposed by (Krinidis and Chatzis 2009) is formulated below.

Let an image be $I(\mathbf{X}) : \Omega \rightarrow \Re^d$, where $\Omega \subset \Re^2$ is the image domain and d is the dimension of feature vector $I(\mathbf{X})$. Let C be a closed contour in the domain Ω which separates Ω into two regions: $\Omega_1 = inside\,(C)$ and $\Omega_2 = outside\,(C)$. The general form of energy function which approximates the given image $I(\mathbf{X})$ by two regions over Ω, defined in Krinidis and Chatzis (2009) is given as

$$\begin{aligned} F\left(C, c_1, c_2, u\right) = {} & \lambda_1 \beta \int_{\Omega} \left[u\left(\mathbf{X}\right)\right]^m \left\|I\left(\mathbf{X}\right) - c_1\right\|^2 d\mathbf{X} \\ & + \lambda_2 \beta \int_{\Omega} \left[1 - u\left(\mathbf{X}\right)\right]^m \left\|I\left(\mathbf{X}\right) - c_2\right\|^2 d\mathbf{X} \\ & + \mu . L\left(C\right); \end{aligned} \tag{8}$$

where $c_1 \in \Re^d$ and $c_2 \in \Re^d$ (depending on C) are the average prototypes of the regions for the given image inside and outside the contour C. The membership function $u\left(\mathbf{X}\right) \in [0, 1]$ is the membership of a pixel $\mathbf{X}$ for belonging to the inside of C. $m > 1$ is a weight exponent which controls the degree of 'fuzziness' of each membership value.

2.4.2 Proposed Global and Local Fuzzy Energy based Active Contour Model

In this article, we proposed a new fuzzy energy function by incorporating local energy term in the existing global fuzzy energy based active contour model (Krinidis and Chatzis

2009) defined in Eq. (8). Basically, local energy term is calculated based on local spatial constraint and local feature constraint. The proposed fuzzy energy based function that takes both global and local information to track the contour of the detected object. After transforming tth frame into the normalized d-dimensional feature space, we have $I(\mathbf{X}^t) : \Omega \rightarrow \Re^d$, where $\Omega \subset \Re^2$ is the domain of the image frame and d is the dimension of feature vector $I(\mathbf{X}^t)$. Let C be a closed contour in the domain Ω which separates Ω into two regions: $\Omega_1 = inside\,(C)$ and $\Omega_2 = outside\,(C)$. The general form of the proposed energy function when tth frame $I(\mathbf{X}^t)$ is approximated by two regions over Ω, is

$$\begin{aligned} F(C, c_1, c_2, u) = {} & \lambda_1 \beta \int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m \left\| I(\mathbf{X}^t) - c_1 \right\|^2 d\mathbf{X}^t \\ & + \lambda_2 \beta \int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m \left\| I(\mathbf{X}^t) - c_2 \right\|^2 d\mathbf{X}^t \\ & + \lambda_1 (1-\beta) \int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m h_1 \, d\mathbf{X}^t \\ & + \lambda_2 (1-\beta) \int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m h_2 \, d\mathbf{X}^t \\ & + \mu . L(C); \end{aligned} \tag{9}$$

with

$$h_1 = \int_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m \left\| I(\mathbf{Y}^t) - c_1 \right\|^2 d\mathbf{Y}^t,$$

$$h_2 = \int_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[u\left(\mathbf{Y}^t\right)\right]^m \left\| I(\mathbf{Y}^t) - c_2 \right\|^2 d\mathbf{Y}^t,$$

where $c_1 \in \Re^d$ and $c_2 \in \Re^d$ (depending on C) are the average prototypes of the regions for the tth frame inside and outside the contour C. The membership function $u\left(\mathbf{X}^t\right) \in [0, 1]$ is the membership of a pixel $\mathbf{X}^t$ for belonging to inside of C. $m > 1$ is a weight exponent which controls the degree of 'fuzziness' of each membership value. The term $I(\mathbf{Y}^t)$ represents the feature values of a pixel $\mathbf{Y}^t$ in a spatial neighborhood of a pixel $\mathbf{X}^t$. $W_{\mathbf{X}^t\mathbf{Y}^t}$ is the weight of $\mathbf{Y}^t$ pixel in a local neighborhood of pixel $\mathbf{X}^t$. $\lambda_1 > 0$ and $\lambda_2 > 0$ are two fixed parameters; $0 \leq \beta \leq 1$ is a constant to control the influence of both the global energy and local energy. $\mu \geq 0$ is the length parameter of the curve C and it can be chosen by the user. Effect of the length parameter for the images without any artifacts was briefly discussed in Pan et al. (2006) and He and Osher (2007). This parameter is not important for such kind of images. However, in case of noisy images, the length parameter can be useful and is briefly discussed in Krinidis and Chatzis (2009) and Li et al. (2008).

Here, the weight $W_{\mathbf{X}^t\mathbf{Y}^t}$ depends on both the local spatial constraint and local feature constraint. For each pixel $\mathbf{X}^t \in \Re^2$, the local spatial constraint reflects the damping extent of neighbors with the spatial distances from the central pixel and is defined as

$$W^{SC}_{\mathbf{X}^t\mathbf{Y}^t} = \frac{1}{1 + d_{\mathbf{X}^t\mathbf{Y}^t}} \tag{10}$$

with $d_{\mathbf{X}^t\mathbf{Y}^t} = \|\mathbf{X}^t - \mathbf{Y}^t\|$, $\mathbf{Y}^t \in \eta_{\mathbf{X}^t}$, where $\eta_{\mathbf{X}^t}$ is the spatial neighborhood (local window) of $\mathbf{X}^t$. The spatial constraint ensures the influence of pixels within the local window. It can be changed according to their distances from the central pixel. With the help of spatial constraint, more local information is incorporated in the proposed energy model.

The local feature constraint is defined as

$$W^{FC}_{\mathbf{X}^t\mathbf{Y}^t} = \exp\left\{ \frac{-\left\| I(\mathbf{X}^t) - I(\mathbf{Y}^t) \right\|^2}{\sum_{\mathbf{Y}^t \in \eta_{\mathbf{X}^t}} \left\| I(\mathbf{X}^t) - I(\mathbf{Y}^t) \right\|^2} \right\}, \tag{11}$$

where $I(\mathbf{X}^t)$ is the d-dimensional feature vector of the central pixel $\mathbf{X}^t$ within a spatial local window and $I(\mathbf{Y}^t)$ is the d-dimensional feature vector of $\mathbf{Y}^t$ pixels in the same window. $\eta_{\mathbf{X}^t}$ is the neighborhood of pixel $\mathbf{X}^t$. The denominator is a function of local density surrounding the central pixel and its value reflects the feature homogeneity degree of that local window. When its value is small, the local window is more homogeneous and vice versa. Eq. (11) indicates that when the feature vector $I(\mathbf{Y}^t)$ is close to $I(\mathbf{X}^t)$, $W^{FC}_{\mathbf{X}^t\mathbf{Y}^t}$ is large and vice versa. The value of $W^{FC}_{\mathbf{X}^t\mathbf{Y}^t}$ can be changed with different feature values of the pixels over a frame and thus it reflects the damping extent in the feature values. Now, the weight $W_{\mathbf{X}^t\mathbf{Y}^t}$ based on both the local spatial constraint and local feature constraint is defined as

$$W_{\mathbf{X}^t\mathbf{Y}^t} = W^{SC}_{\mathbf{X}^t\mathbf{Y}^t} * W^{FC}_{\mathbf{X}^t\mathbf{Y}^t}. \tag{12}$$

The energy function [in Eq. (9)] can be minimized when the contour C is exactly on the object boundary and c_1 and c_2 optimally approximate the tth frame into two regions inside and outside the contour C, respectively. To obtain the entire object boundary, a contour C has to be found which minimizes the proposed energy function given in Eq. (9) over all pixels $\mathbf{X}^t \in \Omega$. In addition, sometimes to smooth the contour C, one penalty term i.e., length $L(C)$ of the contour C is considered for the proposed energy function.

The energy function given in Eq. (9) will be minimized when $C = C_O$, i.e.,

$$C_O = \arg\min_{C} F(C, c_1, c_2, u). \tag{13}$$

Fig. 12 Camouflaged object is tracked in the target candidate frame of Clip152 sequence using the proposed fuzzy energy based active contour

Fig. 13 Camouflaged object is tracked in the target candidate frame of Clip730 sequence using the proposed fuzzy energy based active contour

Therefore, the curve C_O is the extracted contour in the tth frame and the object is tracked with the extracted contour in the tth frame. Figures 12 and 13 show the tracking results of camouflaged object in the tth frame of Clip152 and Clip730 sequences, respectively, using the proposed fuzzy energy based active contour.

2.4.3 Pseudo Level Set Formulation

The proposed fuzzy energy function is formulated using a pseudo level set formulation as considered in Krinidis and Chatzis (2009) and Osher and Sethian (1988) based on the membership values. The curve $C \subset \Omega$ is implicitly represented by the pseudo zero level set of Lipschitz function $u(\mathbf{X}^t) : \Omega \rightarrow \Re$, such that

$$\begin{cases} C = \left\{\mathbf{X}^t \in \Omega : u\left(\mathbf{X}^t\right) = 0.5\right\}, \\ inside\,(C) = \left\{\mathbf{X}^t \in \Omega : u\left(\mathbf{X}^t\right) > 0.5\right\}, \\ outside\,(C) = \left\{\mathbf{X}^t \in \Omega : u\left(\mathbf{X}^t\right) < 0.5\right\}. \end{cases} \tag{14}$$

Here, it is noted that the regularization or the penalty term of the proposed energy function $F(C, c_1, c_2, u)$ given in Eq. (9) is $L(C) = \int_{\Omega} \left|\nabla H\left(u\left(\mathbf{X}^t\right) - 0.5\right)\right| d\mathbf{X}^t$, where $H(s)$ is a heaviside function (Osher and Sethian 1988). Therefore, the energy function in Eq. (9) can be rewritten as

$$\begin{aligned} F(C, c_1, c_2, u) &= \lambda_1 \beta \int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m \left\| I(\mathbf{X}^t) - c_1 \right\|^2 d\mathbf{X}^t \\ &+ \lambda_2 \beta \int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m \left\| I(\mathbf{X}^t) - c_2 \right\|^2 d\mathbf{X}^t \\ &+ \lambda_1 (1-\beta) \int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m h_1 \, d\mathbf{X}^t \\ &+ \lambda_2 (1-\beta) \int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m h_2 \, d\mathbf{X}^t \\ &+ \mu \int_{\Omega} \left|\nabla H\left(u\left(\mathbf{X}^t\right) - 0.5\right)\right| d\mathbf{X}^t. \end{aligned} \tag{15}$$

The values of u for the pixels located inside the contour C are different from the corresponding values for the pixels located outside the contour, when the energy function is minimized. However, the values of u for the pixels located inside the contour C are similar. This is the same for the pixels located outside the contour C.

Now keeping u fixed and minimizing the energy function $F(C, c_1, c_2, u)$ [in Eq. (15)] with respect to c_1 and c_2, we get the following equations:

$$\begin{aligned} &\int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m \left[I\left(\mathbf{X}^t\right) - c_1\right] d\mathbf{X}^t = 0 \\ &\Rightarrow c_1 = \frac{\int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m I\left(\mathbf{X}^t\right) d\mathbf{X}^t}{\int_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m d\mathbf{X}^t}, \end{aligned} \tag{16}$$

and

$$\begin{aligned} &\int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m \left[I\left(\mathbf{X}^t\right) - c_2\right] d\mathbf{X}^t = 0 \\ &\Rightarrow c_2 = \frac{\int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m I\left(\mathbf{X}^t\right) d\mathbf{X}^t}{\int_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m d\mathbf{X}^t}. \end{aligned} \tag{17}$$

Again, keeping c_1 and c_2 fixed and minimizing the energy function $F(C, c_1, c_2, u)$ [given in Eq. (15)] with respect to $u(\mathbf{X}^t)$, we get

$$\begin{aligned} &\lambda_1 \beta m \left[u\left(\mathbf{X}^t\right)\right]^{m-1} \left\| I(\mathbf{X}^t) - c_1 \right\|^2 \\ &- \lambda_2 \beta m \left[1 - u\left(\mathbf{X}^t\right)\right]^{m-1} \left\| I(\mathbf{X}^t) - c_2 \right\|^2 \\ &+ \lambda_1 (1-\beta) m \left[u\left(\mathbf{X}^t\right)\right]^{m-1} h_1 \\ &- \lambda_2 (1-\beta) m \left[1 - u\left(\mathbf{X}^t\right)\right]^{m-1} h_2 = 0 \\ &\Rightarrow u\left(\mathbf{X}^t\right) = \frac{1}{1 + \left(\frac{\lambda_1 \left\{\beta \left\| I(\mathbf{X}^t) - c_1 \right\|^2 + (1-\beta) h_1\right\}}{\lambda_2 \left\{\beta \left\| I(\mathbf{X}^t) - c_2 \right\|^2 + (1-\beta) h_2\right\}}\right)^{\frac{1}{m-1}}}, \end{aligned} \tag{18}$$

where h_1, h_2 are similar as defined in Eq. (9).

For simplicity, without loss of generality, the above minimization [Eq. (18)] has been done without considering the length term (i.e., $\mu = 0$).

2.4.4 Convexity of the Proposed Energy Model

The contour is iteratively evolved with an initialization (object detection result of the modified PNN). When the energy function defined in Eq. (15) is minimum, the optimal contour is obtained. If the energy function is convex, then it is easy to find the global optimal solution (contour) with an iterative way. Therefore, in this section, we will prove that the proposed energy function F in Eq. (15) is convex with respect to u.

Proof is given in Appendix 1.

2.4.5 Numerical Approximation

The proposed energy function [given in Eq. (15] as a minimization problem) can be solved by the gradient descent approach which is derived from the Euler–Lagrange equation (Fox 1988). But it is slow for convergence (Krinidis and Chatzis 2009). In this article, a fast numerical approach proposed by Song and Chan (2002) and developed by Krinidis and Chatzis (2009), is used. Instead of solving the Euler-Lagrange equation, this method calculates and verifies whether the energy decreases when the membership value of any arbitrary pixel on the tth frame changes or vice versa.

Now for a pixel $\mathbf{X}^t \in \Omega$ of the tth frame, the feature value of the pixel $\mathbf{X}^t$ is I_o and the corresponding degree of membership is u_o. Assume that the degree of membership of the pixel $\mathbf{X}^t$ is changed to a new value u_n which is calculated using Eq. (18). Let ΔF be the difference between the new and old energies when the degree of membership of pixel $\mathbf{X}^t$ is changed. Then ΔF (detail derivation is given in Appendix 2) is calculated as follows:

$$
\begin{aligned}
\Delta F = {} & \lambda_1 \beta S_1 a_1 \|I_o - c_1\|^2 + \lambda_2 \beta S_2 a_2 \|I_o - c_2\|^2 \\
& - 2\lambda_1 (1-\beta) S_1 \|I_o - c_1\| b_1 \\
& + \lambda_1 (1-\beta) g_1 \left[\|c_1 + S_1 (\|I_o - c_1\|)\|^2 - \|c_1\|^2\right] \\
& + \lambda_1 (1-\beta) \left([u_n]^m - [u_o]^m\right) \\
& \left\{\sum_{\Omega} W_{o\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m \|I(\mathbf{Y}^t) - c_1 - S_1 (\|I_o - c_1\|)\|^2\right\} \\
& + \lambda_1 (1-\beta) \left([u_n]^m - [u_o]^m\right) \\
& \left\{\sum_{\Omega} W_{o\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m \|I_o - c_1\|^2 [1 - S_1]^2\right\} \\
& - 2\lambda_2 (1-\beta) S_2 \|I_o - c_2\| b_2 \\
& + \lambda_2 (1-\beta) g_2 \left[\|c_2 + S_2 (\|I_o - c_2\|)\|^2 - \|c_2\|^2\right] \\
& + \lambda_2 (1-\beta) \left[(1-u_n)^m - (1-u_o)^m\right] \\
& \left\{\sum_{\Omega} W_{o\mathbf{Y}^t} \left[u\left(\mathbf{Y}^t\right)\right]^m \|I(\mathbf{Y}^t) - c_2 - S_2 (\|I_o - c_2\|)\|^2\right\} \\
& + \lambda_2 (1-\beta) \left[(1-u_n)^m - (1-u_o)^m\right] \\
& \left\{\sum_{\Omega} W_{o\mathbf{Y}^t} \left[u\left(\mathbf{Y}^t\right)\right]^m \|I_o - c_2\|^2 [1 - S_2]^2\right\}, \qquad (19)
\end{aligned}
$$

where

$$
a_1 = \sum_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m,\ a_2 = \sum_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m,
$$

$$
S_1 = \frac{[u_n]^m - [u_o]^m}{\sum_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m + [u_n]^m - [u_o]^m},
$$

$$
S_2 = \frac{(1-u_n)^m - (1-u_o)^m}{\sum_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m + (1-u_n)^m - (1-u_o)^m},
$$

$$
b_1 = \sum_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m \left\{\sum_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m I(\mathbf{Y}^t)\right\},
$$

$$
b_2 = \sum_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m \left\{\sum_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[u\left(\mathbf{Y}^t\right)\right]^m I(\mathbf{Y}^t)\right\},
$$

$$
g_1 = \sum_{\Omega} \left[u\left(\mathbf{X}^t\right)\right]^m \left\{\sum_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m\right\},
$$

$$
g_2 = \sum_{\Omega} \left[1 - u\left(\mathbf{X}^t\right)\right]^m \left\{\sum_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[u\left(\mathbf{Y}^t\right)\right]^m\right\}.
$$

2.5 Online Update to Handle Drift Problem

To capture the changes due to both camera and object motions, the proposed modified PNN based discriminative appearance model needs to be updated. In this work, after tracking the object in the tth frame, new training (samples) information generated from the tth frame is used to update the proposed discriminative model and to detect and track the object in the next $(t+1)$th frame. Since every time, the proposed discriminative appearance model is updated with new information, it is able to reduce the propagation of drift.

2.6 Computational Complexity

Let N be the total number of patterns (corresponding to pixels) in the tth frame. Let the cost for calculating HOG feature for each pixel be f_{hog}. Therefore, total cost for calculating HOG of each frame is $N * f_{hog}$. Now

$$
N * f_{hog} = N * B_{hog} * W_{hog} + f_{n_1},
$$

where B_{hog} is the number of considered histogram bins and W_{hog} is the considered window size. Again let the cost for calculating LTP feature for each pixel be f_{ltp}. Therefore, total cost for generating LTP of a frame is $N * f_{ltp}$. Now

$$N * f_{ltp} = N(f_a * W_{ltp} + f_{n_2}),$$

where W_{ltp} is the considered window size. Therefore, let the total cost required to generate 13-dimensional feature vectors for a frame be $f_{feature}$. We have

$$f_{feature} = N * f_{hog} + N * f_{ltp} + f_{n_3},$$

where f_{n_3} is a constant cost. To estimate the motion of the object between the $(t-1)$th and the tth frames using Eqs. (3) and (4), they need constant costs f_{n_4} and f_{n_5}, respectively. To classify each test pattern in the tth frame, let it require f_{pnn} cost. Then we have

$$f_{pnn} = N_1 * f_{n_6} + N_1 * f_{n_7} + f_{n_8},$$

where N_1 is the number of training samples and $N_1 << N$. Therefore, total cost required to detect an object in the tth frame is

$$N * f_{pnn} = N(N_1 * f_{n_6} + N_1 * f_{n_7} + f_{n_8}).$$

In the proposed fuzzy active contour model, let W be the size of the considered neighborhood and T be the total number of iterations required to converge, then the total cost required for fuzzy active contour is

$$f_{facm} = N * W * T * f_b + f_{n_9}.$$

Therefore, total cost required for the proposed tracking algorithm is

$$\begin{aligned} f_t &= f_{feature} + N * f_{pnn} + f_{facm} + f_{n_4} + f_{n_5} \\ &= N * f_{hog} + N * f_{ltp} + f_{n_3} \\ &\quad + N(N_1 * f_{n_6} + N_1 * f_{n_7} + f_{n_8}) \\ &\quad + N * W * T * f_b + f_{n_9} + f_{n_4} + f_{n_5} \\ &= N * B_{hog} * W_{hog} + f_{n_1} + N(f_a * W_{ltp} + f_{n_2}) + f_{n_3} \\ &\quad + N(N_1 * f_{n_6} + N_1 * f_{n_7} + f_{n_8}) \\ &\quad + N * W * T * f_b + f_{n_9} + f_{n_4} + f_{n_5} \\ &\leq N^2, N \gg N_1 \text{ and } N \gg T. \end{aligned}$$

For all the experiments, B_{hog}, W_{hog}, W_{ltp} and W are constants. Therefore, time complexity of the proposed tracking algorithm is

$$f_t = \mathrm{O}\left(N^2\right).$$

Here we also provide the proposed tracking method in an algorithmic form.

Algorithm 1 : Proposed Algorithm

Input: $I(\mathbf{X}^{t-2})$ and $I(\mathbf{X}^{t-1})$ with object contours $C(\mathbf{X}^{t-2})$ and $C(\mathbf{X}^{t-1})$.
Output: Contour $C(\mathbf{X}^t)$ in the tth frame.

for ($t = 3$ to end of the sequence)
1: Transform both $I(\mathbf{X}^{t-1})$ and $I(\mathbf{X}^t)$ into d-dimensional feature space.
2: Calculate motion ($\bar{M}$) of the object between the $(t-2)$th and the $(t-1)$th frames using Eqs. (3) and (4).
3: Detect object in the tth frame using the proposed modified PNN.
4: Extract contour $C(\mathbf{X}^t)$ of the object in the tth frame using the proposed fuzzy active contour model.
(i): For initial partition, set up $u(\mathbf{X}^t) > 0.5$ for one part (detected object region) and $u(\mathbf{X}^t) < 0.5$ for the other part (i.e., outside the detected object region).
(ii): Compute $c_1(\mathbf{X}^t)$ and $c_2(\mathbf{X}^t)$ using Eqs. (16) and (17).
(iii): Assume that feature value of the current pixel $\mathbf{X}^t$ is $I_o(\mathbf{X}^t)$ and $u_o(\mathbf{X}^t)$ is its degree of membership value. Then the new degree of membership value $u_n(\mathbf{X}^t)$ is calculated by using Eq. (18) for the current pixel $\mathbf{X}^t$. The difference between new and old energies ΔF is calculated by using Eq. (19).
If $\Delta F < 0$, then $u_o(\mathbf{X}^t)$ is changed by $u_n(\mathbf{X}^t)$, otherwise keep the old value $u_o(\mathbf{X}^t)$.
(iv): Repeat step (iii) using Jacobi iteration to calculate total energy F for the whole tth frame.
(v): Repeat steps (ii)–(iv) till the total energy F remains unchanged.
end

3 Experimental Results and Analysis

In this section, the experimental setups and analysis (both quantitative and qualitative) of results are presented.

3.1 Experimental Setups

We utilized 8 bin HOG, LTPL and LTPU textural features extracted only on the lightness L component of CIELab color space. Here, HOG is 8 dimensional feature vector, color is 3 dimensional feature vector, each of LTPL and LTPU is 1 dimensional feature vector. Therefore, dimension of the required feature vector [combination of color (3), texture (2) and shape (8)] is 13. In all experiments, we set W_{hog}=5, W_{ltp}=3 and W=5. To collect the training set in the initialization step, contours of the target in the 1st and 2nd frames are located by manually cropping. The detected object using the proposed modified PNN in the tth frame is considered to initialize the fuzzy active contour model. Here, camera or object deformation parameter γ is experimentally set to 15 for all experiments. The smoothing parameter σ in PNN is considered in the range [0.1, 5.0]. Parameters λ_1 and λ_2 both are set to 1 and β within the range [0, 1]. All simulations have been done on a machine with Intel(R) Core(TM)i3 CPU 3.20 GHz, 4.0 GB RAM and fedora operating system.

The proposed algorithm is evaluated on 200 (50 sequences from Visual Tracker Benchmark,[4] 100 sequences from Reef Video Clip Database,[5] 30 sequences from ALOV++[6] and 20 sequences from VOT2015 Challenge)[7] challenging video

[4] http://cvlab.hanyang.ac.kr/tracker_benchmark/indexhtml.
[5] http://www.reefvid.org/.
[6] http://crcv.ucf.edu/data/ALOV++/.
[7] http://www.votchallenge.net/vot2015/dataset.html.

sequences. These sequences include most challenging factors in visual tracking like fully or partially camouflaged object, complex background, illumination variation, fast motion, large variation in scale, motion blur, rotation, non-rigid object deformation, partial or full occlusion, low contrast, etc. Due to the page constraint, details about some of the considered video sequences are given in Table 3. The tracking results obtained using the proposed method were compared with recent popular existing tracking algorithms: DFT,[8] ASLA,[9] MTT,[10] SCM,[11] SPT,[12] KCF[13] and two camouflaged object tracking algorithms: weighted region consolidation (WRC) (Huang and Jiang 2005) and optical flow and Kalman filter (OFKF) technique (Hou and Li 2011). Here, the proposed method provides object contour (boundary) whereas all existing considered techniques (except WRC) use rectangular trackers. Therefore, to make a fair comparison between the proposed method and all existing considered techniques, rectangular tracker is used with the contour information of the proposed method. However, due to page constraint, results of some (30) sequences are provided in this article. More results (obtained by WRC and OFKF techniques) can be found in the supplementary material.

Table 3 Details about some of the considered sequences

Sequences	Frame resolution	Total number of frames
Bmx	1280 × 692	76
Butterfly	640 × 480	151
Crossing	1210 ×681	131
Iceskate	480 ×270	661
Octopus	1280 ×270	291
Sphere	480 ×360	201
Light23	480 ×360	238
Specularity8	1280 ×720	119
Shape14	1280 ×720	299
Motionc1	640 ×480	196
Confusion35	720 ×576	109
Low contrast13	1920 ×1080	421
Moving camera15	854 ×480	250
Zooming camera25	1920 ×1080	120
Clip730	512 × 288	453
Clip152	384 × 288	198
Clip187	384 × 288	407
Clip284	384 × 288	694
Clip442	384 × 288	1182
Clip498	384 × 288	166
Clip233	384 × 288	89
Clip790	360 × 288	186
Clip796	360 × 288	1781
Clip249	384 × 288	183
Trans	640 × 332	124
Faceocc1	352 ×288	892
Carscale	640 ×272	252
Doll	400 ×300	3872
Jogging	352 ×288	307
Singer1	624 ×352	351
Trellis	320 ×240	569

3.2 Evaluation Methodology

In this article, precision plot, success plot Babenko et al. (2011), Henriques et al. (2015), Wu et al. (2013), average center location error and average tracking accuracy (ATA)Kasturi et al. (2009) are considered to quantitatively evaluate the performance of the proposed algorithm for tracking single object. In case of multiple objects tracking, multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) Kasturi et al. (2009) are also considered for quantitative analysis.

3.2.1 Precision Plot

Center location error (Kasturi et al. 2009) is defined as the Euclidean distance between the center location of the tracked target and the ground truth. Then the average center location error over all the frames of a sequence is considered to summarize the overall performance for that sequence. However, when the tracker loses the target, the output location can be random and the average center location error may be unable to measure the tracking performance correctly (Babenko et al. 2011; Wu et al. 2013). Recently, the precision plot (Babenko et al. 2011; Henriques et al. 2015; Wu et al. 2013) has been considered to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold value (with respect to the ground truth). In this article, the threshold value is set to 20 pixels (Babenko et al. 2011; Wu et al. 2013).

3.2.2 Success Plot

Another evaluation metric is the bounding box (area) overlap. For a given tracked bounding box A and the ground truth bounding box B, the overlap score is defined as $S = \frac{|A \cap B|}{|A \cup B|}$, where $\cap$ and $\cup$ represent the intersection and union of two

[8] http://people.cs.umass.edu/~lsevilla/20trackingDF.html.

[9] http://faculty.ucmerced.edu/mhyang/project/cvpr12_jia_project.htm.

[10] http://sites.google.com/site/zhangtianzhu2012/publications.

[11] http://faculty.ucmerced.edu/mhyang/project/cvpr12_scm.htm.

[12] http://www.umiacs.umd.edu/~fyang/spt.html.

[13] http://home.isr.uc.pt/~henriques/circulant/.

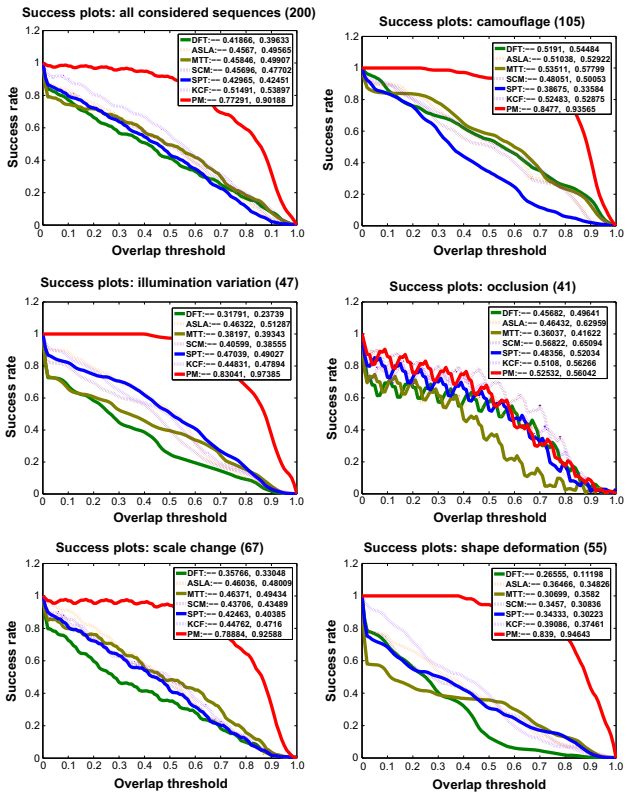


Fig. 14 Each plot corresponds to one-pass evaluation (OPE). In the legend of success plots, the first value is AUC score and the last one is the success rate value at threshold 0.5 (Color figure online)

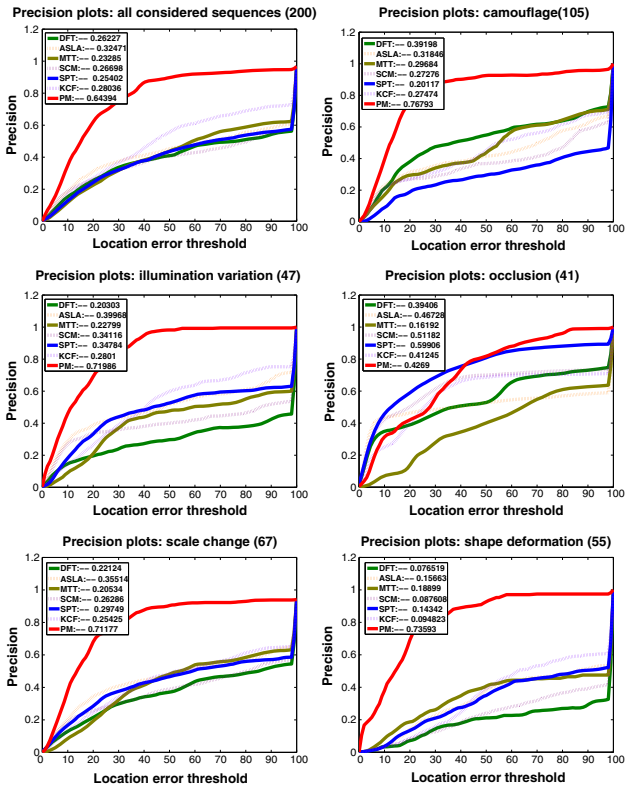


Fig. 15 Each plot corresponds to one-pass evaluation (OPE). In the legend of precision plots, the value indicates precision value at threshold (20 pixels) (Color figure online)

regions, respectively and |.| denotes the number of pixels in the region. To measure the performance of an algorithm on a sequence of frames, we count the number of successful frames whose overlap score S is larger than the given threshold value. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. Success rate value at a specific threshold (e.g., 0.5) may not always be the representative for evaluating tracker performance (Wu et al. 2013). Instead of that, area under curve (AUC) (Wu et al. 2013) of each success plot is considered to evaluate the performance of the tracking algorithms.

3.3 Quantitative Analysis

The overall performances of the proposed method and all other existing techniques are summarized by the success and precision plots as shown in Figs. 14 and 15, respectively. For success plots, AUC score is used to summarize the performances of different algorithms, while for precision plots, center location error at threshold value (20) is considered. In the precision plots, the performances of some techniques are different from the performances of algorithms in the success plots. Since success and precision plots are based on different metrics which measure different characteristics of the trackers. However, AUC score of the success plot measures the overall performance which is more accurate than the center location error at threshold value of the precision plot (Wu et al. 2013). In this article, performances of different algorithms are analyzed based on success plots; however the precision plots can be considered as auxiliary.

From the success plots of all the considered (200) sequences, it is seen that the average performance of KCF is better than all other existing (SCM, SPT, DFT, ASLA and MTT) techniques. This is due to the use of robust appearance model (circulant structure) and update mechanism. ASLA, SCM and MTT have similar (almost equal) performance. Sparse representations are considered in all these techniques. On the other hand, SPT outperforms DFT by 2.5 % but below KCF by 19 %. However, the proposed method performs better than all existing techniques. Tables 4 and 5 also highlight similar findings. More plots can be found in supplementary material.

3.3.1 Quantitative Analysis Based on Attributes

Here, the performances of the proposed algorithm and other existing techniques are analyzed based on several attributes

Table 4 Average tracking accuracy (area overlap) of all considered (200) sequences

Sequences	DFT	ASLA	MTT	SCM	SPT	KCF	PM
200	0.4129	0.4501	*0.4504*	0.4501	0.4429	**0.5098**	***0.7750***

BoldItalics indicates best value, *bold* indicates second best, *italics* indicates third best

Table 5 Average center location error of all considered (200) sequences

Sequences	DFT	ASLA	MTT	SCM	SPT	KCF	PM
200	66.715	61.778	69.024	62.366	*60.294*	**40.822**	***10.441***

BoldItalics indicates best value, *bold* indicates second best, *italics* indicates third best

like camouflage, illumination variation, occlusion, scale change, shape deformation, background clutter, fast motion, pose change, etc. Due to the page constraint, analysis based on some of them (attributes) are discussed here. In case of camouflage, MTT produces better results than all other existing techniques. SPT produces worst results due to segmentation. However, the proposed method obtains the best results among all existing algorithms. Though SPT works well than other state-of-the-art techniques for sequences having illumination variations, but the proposed method provides the best results. In case of occlusion, ASLA, SCM, MTT all performed better due to sparse representation. Among them, SCM provides better results than ASLA and MTT because it uses hybrid (both generative and discriminative) appearance model based on local sparse representation. However, KCF outperforms others while handling occlusion due to kernelized correlation filter based on discrete Fourier transform which is robust under occlusion. On the other hand, the proposed method is unable to detect and track object under full occlusion. But it produces better results for partial occluded objects. Due to partial occlusion, the AUC shows that the proposed method provides acceptable results under occlusion. On the other hand, MTT outperforms for sequences having objects with large scale variations. Among all these techniques, the proposed method obtains the best results because it extracts contour of the object, so scale change can not create a problem for extracting contours. Tables 6 and 7 display the center location errors and average tracking accuracies of some considered sequences for all techniques. These tables also highlight similar findings.

3.4 Qualitative Analysis

3.4.1 Scale Change and Pose Variation

We evaluate the performances of all the trackers on sequences where objects undergo large scale and pose changes. First, we compare the performances of trackers in terms of handling large scale variation. Carscale and Doll sequences of Fig. 16, Crossing sequence of Fig. 17, Low contrast and Light sequences of Fig. 18, Clip249, Clip284, Clip442, Clip730 sequences of Fig. 19. The target undergoes large scale change in these images due to object movement, whereas Singer1 of Fig. 16, Moving camera and Zooming camera of Fig. 18, the target changes its scale due to camera motion (or zooming). From these figures, it is shown that ASLA and SCM tried to adapt the scale but was unable to adapt actual scale of the object, whereas DFT, MTT, KCF techniques failed to adapt large scale change of the object. SPT provided better results than other existing techniques for sequences where objects underwent scale changes, but failed for large scale variation. On the other hand, as the proposed method extracts contour of the object, it is able to properly track the object with large scale variation.

In the Trans sequence of Fig. 16, the Bmx, Iceskate and Octopus sequences of Fig. 17, the objects change their appearances significantly with their scales. The Trellis sequence of Fig. 16 changes its pose under illumination variation condition. In Clip790 and Clip152 of Fig. 19, the objects change their pose in camouflaged environment. In these sequences, KCF and SPT tried to track the object properly, but due to scale change and occurrence of camouflage, after some frames, they fail to provide good results. However, the proposed method uses local information to detect the object, and thus pose change does not create a problem for the proposed method.

3.4.2 Shape Deformation

The Trans sequence of Fig. 16, Octopus, Iceskate sequences of Fig.17 and Shape sequence of Fig. 17 show examples where drastic shape deformation occurs, tracking algorithms using holistic appearance models are unlikely to perform well (e.g., MTT and KCF). Other trackers such as ASLA, SCM, SPT also generate inaccurate results and their tracking windows cover a small portion of the object with a few number of successfully tracked frames. On the other hand, shape deformation can not create a problem for the proposed method for extracting contour of such objects. Hence, the proposed method can track shape deformable objects successfully.

Table 6 Center location error

Sequences	DFT	ASLA	MTT	SCM	SPT	KCF	PM
Bmx	80.931404	*74.183272*	93.527788	86.425588	**57.242464**	79.103122	***4.420175***
Butterfly	*39.400818*	289.603366	300.665582	189.480262	195.836494	**24.302096**	***4.500000***
Crossing	*22.658566*	138.828635	26.915897	51.428637	35.390508	**21.700795**	***4.374453***
Iceskate	*39.578972*	128.544565	196.458901	132.942693	177.144333	**20.487457**	***20.357091***
Octopus	*59.626939*	**46.071459**	124.365045	61.708927	94.089470	62.336120	***6.670759***
Sphere	50.751903	133.314856	152.023315	97.857628	***16.831229***	*19.580526*	**18.971748**
Light	163.502168	17.315019	*15.772800*	186.808959	16.450604	**12.955346**	***3.704055***
Specularity	129.222652	43.072263	**14.677605**	74.712240	63.722877	*16.649455*	***14.522926***
Shape	195.110579	*13.611480*	**13.378924**	16.809748	27.200487	36.303328	***12.708082***
Motionc	73.601444	*23.368815*	***13.177996***	41.463503	41.078864	28.393178	**13.227517**
Confusion	7.912099	10.284466	5.964288	*5.163981*	13.702894	***4.508166***	**4.834151**
Low contrast	64.455336	45.932675	33.128893	*30.408819*	108.949187	**25.565931**	***6.128211***
Moving camera	225.216114	***7.651350***	32.740981	21.050286	*12.944514*	21.745703	**11.347053**
Zooming camera	*13.935250*	73.807708	**11.885161**	99.897294	41.697262	22.216277	***6.164569***
Clip730	**8.068376**	*12.068048*	17.595768	28.570511	41.438430	92.027791	***5.471521***
Clip152	100.761389	195.092011	168.013769	172.403973	**41.188876**	*94.604120*	***38.154898***
Clip187	**25.809414**	86.794811	157.218992	90.562039	102.678089	*64.981330*	***4.845389***
Clip284	*30.290749*	35.593709	34.037194	66.725242	65.170559	**26.443014**	***12.219050***
Clip442	85.235805	95.701956	*50.343628*	83.312501	75.672177	**48.166680**	***7.877701***
Clip498	33.489636	*14.699226*	**10.673971**	17.003137	20.333979	17.846120	***5.516157***
Clip233	**12.524848**	34.792787	25.246487	40.189297	40.455687	*22.395450*	***8.110710***
Clip790	49.445279	49.055761	**29.931174**	47.763538	204.462591	*39.698874*	***12.262466***
Clip796	*2.579731*	***2.199148***	4.968583	**2.310338**	65.141472	2.853344	2.719945
Clip249	29.794407	30.931084	*26.985849*	28.795526	**6.698778**	27.347306	***4.967117***
Trans	268.343388	39.333038	177.186029	*42.258811*	**11.310398**	168.902250	***10.315606***
Faceocc1	23.587613	78.167291	29.753680	**12.474893**	40.854845	***11.930419***	*22.141050*
Carscale	75.751940	*24.529978*	83.543614	33.052647	**18.080219**	76.641049	***3.949226***
Jogging	**27.772805**	98.316951	93.187874	98.407995	***1.989573***	*90.302107*	95.719804
Singer1	18.784432	**3.442082**	69.227230	***2.956170***	155.814957	13.301354	*3.622780*
Trellis	42.245188	7.536595	56.221727	***6.294823***	*6.989458*	17.342230	**6.334139**

BoldItalics indicates best value, *bold* indicates second best value and *italics* indicates third best

3.4.3 Camouflaged Object

The performances of the trackers are evaluated on sequences containing camouflaged objects. Clip187, Clip233, Clip498, Clip796 sequences of Fig. 19 have camouflaged objects with slow motion. From these figures, it is seen that all these trackers try to successfully track the camouflaged objects. However, MTT, ASLA, SCM perform better than KCF and SPT as local sparse representation is considered for appearance modeling. The proposed method provides the best result among them. In case of Clip152, Clip249, clip284, Clip442, Clip730 sequences of Fig. 19 which contain camouflaged object with scale variation and in Clip790 sequence of Fig. 19 containing camouflaged object which changes it appearance. All the existing trackers try to properly track the objects; but due to scale and appearance changes, after few frames they fail to track the object properly. On the other hand, the proposed method successfully tracks the object properly in such complex environment.

3.4.4 Occlusion

We evaluate the performance of tracking algorithms including the proposed technique on sequences where the objects undergo partial or heavy occlusions several times. In the Carscale sequence of Fig. 16, the object undergoes partial occlusion; whereas in Faceocc1 and Jogging sequences of Fig. 16, the object undergoes heavy occlusion. All existing trackers try to successfully track the occluded object. But due to fast motion and scale change, they fail to successfully track the objects after a few number of frames (e.g., Carscale#252 Jogging#307). Incase of partial occlusion, the

Table 7 Average tracking accuracy

Sequences	DFT	ASLA	MTT	SCM	SPT	KCF	PM
Bmx	0.198994	*0.206489*	0.141809	0.171547	**0.230473**	0.196422	***0.928814***
Butterfly	*0.297064*	0.006683	0.002663	0.019546	0.041330	**0.409660**	***0.909362***
Crossing	*0.497065*	0.094643	**0.533404**	0.328756	0.467039	0.493698	***0.554065***
Iceskate	*0.346022*	0.174792	0.062168	0.093337	0.091373	**0.526962**	***0.568468***
Octopus	*0.286723*	**0.328906**	0.205906	0.258324	0.108520	0.246128	***0.861250***
Sphere	0.372123	0.105290	0.056870	0.181065	*0.512848*	**0.656769**	***0.683004***
Light	0.036534	0.501686	**0.685550**	0.030745	0.666188	*0.668888*	***0.880991***
Specularity	0.516896	0.670966	**0.813402**	0.606887	0.671278	*0.700844*	***0.885334***
Shape	0.257372	0.623641	**0.788649**	*0.739798*	0.563029	0.648203	***0.837811***
Motionc	0.455324	*0.658334*	**0.686720**	0.602468	0.541738	0.616924	***0.872202***
Confusion	0.859137	0.798740	**0.875350**	0.852384	0.780385	***0.894058***	*0.870293*
Low contrast	0.482756	0.570736	0.647150	**0.678614**	0.132290	*0.651701*	***0.914405***
Moving camera	0.012042	0.617392	0.461850	**0.625365**	***0.763915***	0.468915	*0.619226*
Zooming camera	*0.536343*	0.384562	**0.573494**	0.382404	0.515667	0.523091	***0.860374***
Clip730	0.289308	*0.433153*	**0.505263**	0.345981	0.297017	0.255194	***0.851651***
Clip152	*0.135127*	0.039828	0.052065	0.048638	**0.281886**	0.161266	***0.558524***
Clip187	**0.467013**	*0.364163*	0.141122	0.351481	0.337268	0.350641	***0.936252***
Clip284	*0.674343*	0.630456	0.670926	0.504710	0.385538	**0.713322**	***0.855343***
Clip442	0.311125	*0.479023*	**0.514813**	0.454602	0.346182	0.329878	***0.903900***
Clip498	0.617117	*0.742313*	**0.765959**	0.704595	0.537662	0.734047	***0.863951***
Clip233	**0.775553**	0.659731	0.715032	0.616157	0.602272	*0.739076*	***0.905239***
Clip790	0.505272	*0.559807*	0.548776	**0.562473**	0.108290	0.547848	***0.859924***
Clip796	***0.939205***	0.835671	*0.908414*	0.837906	0.487918	**0.938508**	0.891181
Clip249	0.505731	0.382954	**0.547316**	0.407107	*0.529238*	0.526891	***0.854965***
Trans	0.009384	*0.546260*	0.246554	0.528747	**0.810588**	0.092383	***0.811549***
Faceocc1	*0.688809*	0.298117	0.580104	**0.749549**	0.573050	***0.795908***	0.375153
Carscale	0.417602	**0.551030**	0.240763	*0.433879*	0.429623	0.418725	***0.819184***
Jogging	**0.201439**	0.160345	0.122781	0.150745	***0.744170***	*0.184589*	0.125540
Singer1	0.357793	*0.437791*	0.160772	**0.496452**	0.170572	0.361830	***0.695146***
Trellis	0.369432	*0.664774*	0.279005	***0.752783***	0.598859	0.489068	**0.671104**

Bolditalics indicates best value, *bold* indicates second best value and *italics* indicates third best

proposed method successfully tracks the object. But for full occlusion, the proposed method fails to track the object due to loss of object information.

3.5 Execution Time Analysis

Table 8 summaries the average execution time for tracking objects in a single frame using different techniques. From the table, it is seen that KCF is faster among all algorithms. Whereas SCM technique takes more time to track an object in a single frame. DFT and ASLA take less average execution time than the proposed method. On the other hand, the proposed method takes less execution time than MTT, SCM and SPT. Therefore, the proposed method needs moderate amount of time to track the object in a single frame.

4 Discussion and Future Work

In this work, it is established that the proposed integrated feature set from multi-cue is able to increase the separability between the camouflaged object and its surroundings. The proposed modified PNN requires small training samples rather than the whole training set and can better detect the camouflaged object efficiently. The proposed global and local fuzzy energy based active contour model uses the region information, is able to extract the contour of the detected object and finally track the object properly.

4.1 Extension to Multiple Objects Tracking

Though the tracking results reported in this article highlight only a single object, it can also be extended to multiple

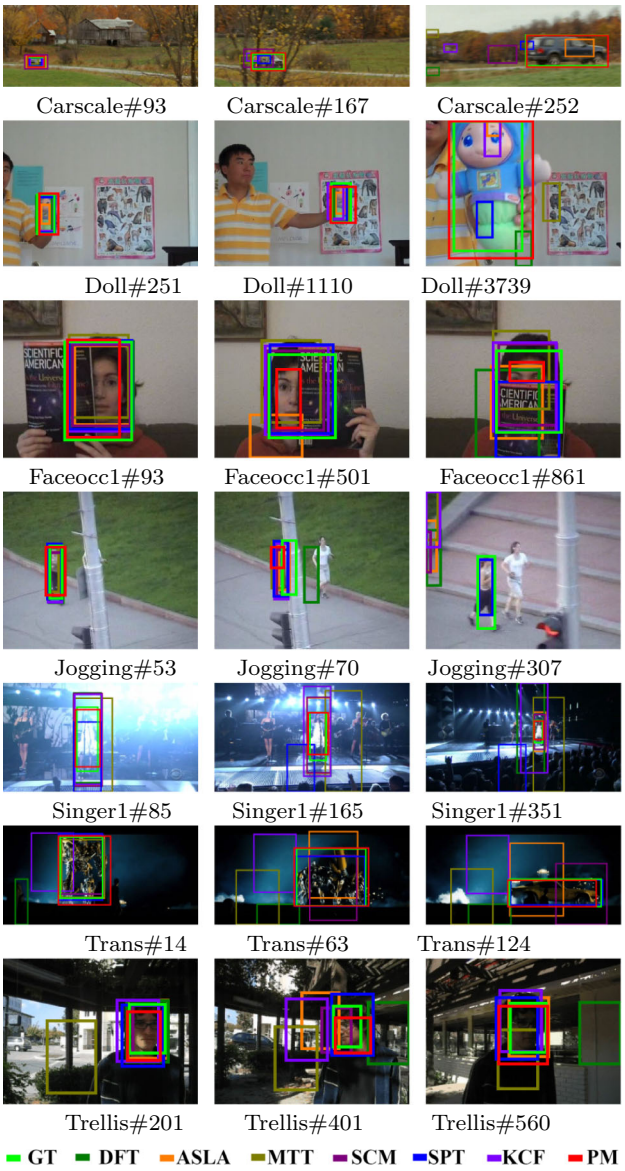


Fig. 16 Tracking results on Visual Tracker Benchmark sequences (Color figure online)

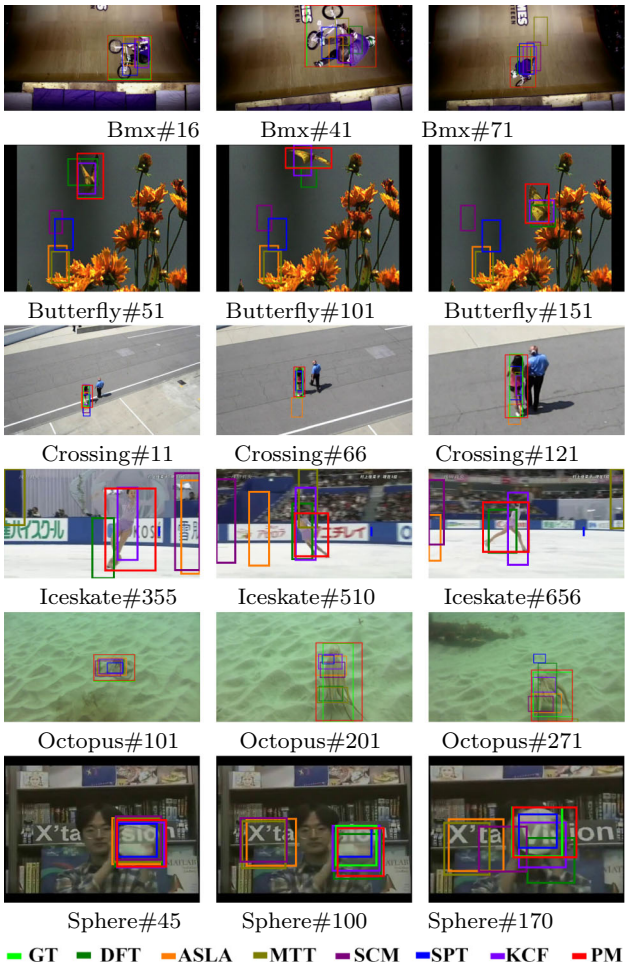


Fig. 17 Tracking results on VOT2015 Challenge sequences (Color figure online)

objects tracking by considering multi-class (multiple objects and background) classification problem instead of two-class (object and background) classification (Desai et al. 2011). After detecting objects using the modified PNN in the tth frame, contours of the detected objects are extracted by the proposed fuzzy energy based active contour model and track the objects. Here, we assume that the sequence contains R numbers of different objects $l_{obj1}, l_{obj2}, .., l_{objR}$ and a background l_{bg}. Initially objects are given by the user. Similar energy function given in Eq. (15) of each object is defined. By minimizing energy functions, contours of the objects are extracted and the objects are tracked. As an illustration, some results of Clip796 sequence are given in Fig. 20. From this figure, it is seen that the proposed method provides acceptable results for tracking multiple objects. To measure the performance (quantitatively) of the proposed algorithm for multiple objects tracking, two values MOTA and MOTP are calculated. For this sequence, MOTA=1 & MOTP=0.8418 and these values indicate that the proposed method can provide acceptable results for multiple objects tracking also.

4.2 Effect of Smoothing Parameter σ

Accuracy of PNN depends on smoothing parameter σ (Specht 1990). In this work, σ is chosen experimentally. For different values of σ, different (object detection) results are obtained. Figures 21 and 22 display camouflaged object detection results of Clip152 and Clip249 sequences, respectively, using the proposed modified PNN with different values of smoothing parameter σ. From these figures, it is observed that for large value of $\sigma > 5$ (Figs. 21c, d and 22c, d), objects are not detected properly. More parts of the object are detected as background. It is also observed that for small values of $\sigma < 0.1$ (Figs. 21i, 22i), objects are detected

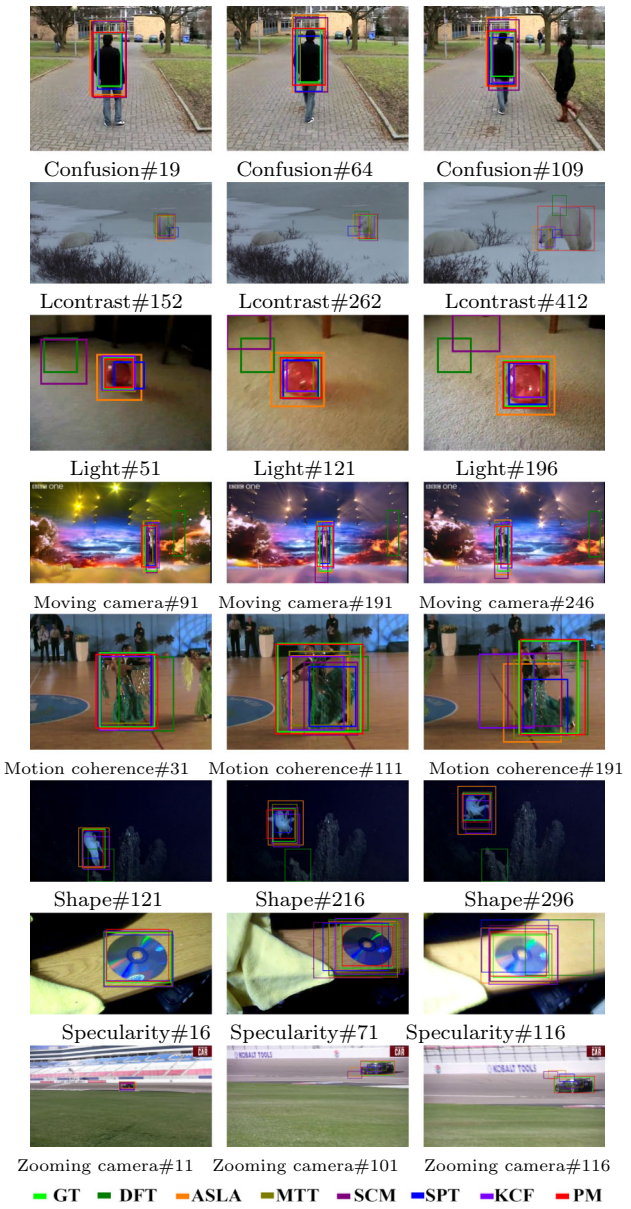


Fig. 18 Tracking results on ALOV++ sequences (Color figure online)

along with more background pixels. For other values of $\sigma \in [0.1,\ 5.0]$, similar kinds of results are obtained (Figs. 21e, f, g, h and 22e, f, g, h). Table 9 shows the detection accuracy of the proposed method for different values of σ. This table also highlights the effect of parameter σ for object detection.

4.3 Effect of Parameter β

In this section, the role of parameter β for controlling the weights of global and local energies (or forces or information or terms) of the proposed fuzzy energy function is discussed. Here, β is $0 \leq \beta \leq 1$. If a lower value of β is chosen, the global term contributes less than the local term in the

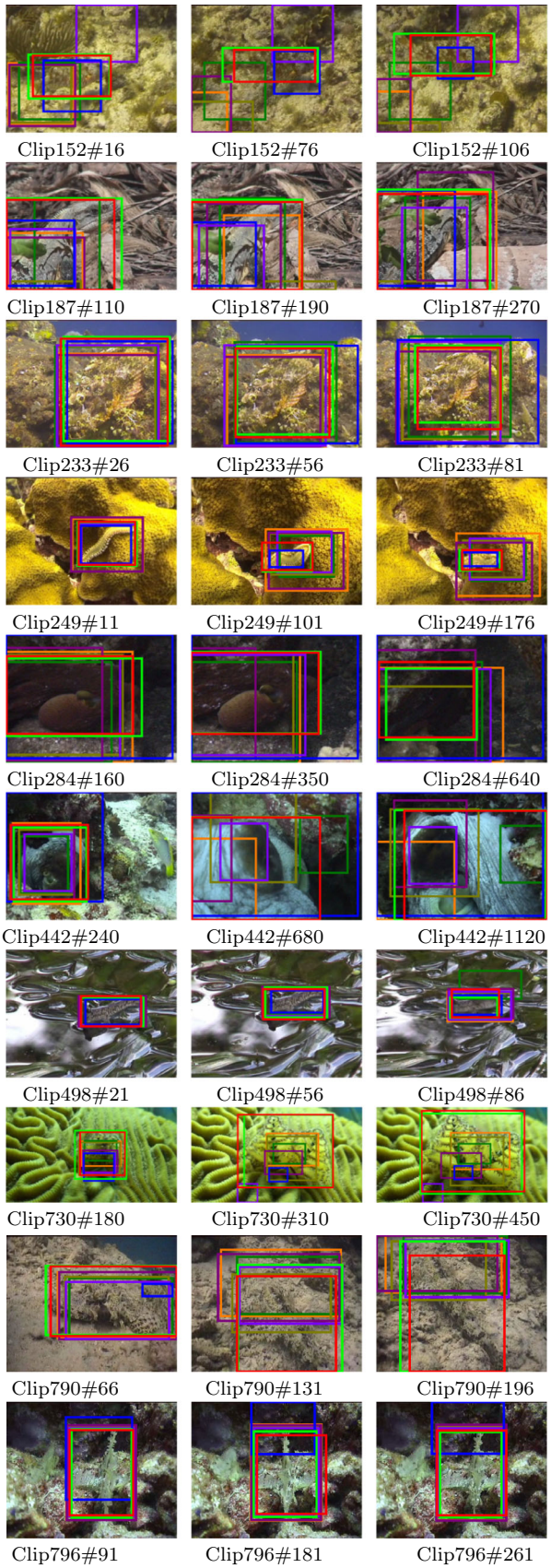


Fig. 19 Tracking results on Reef Video Clip sequences (Color figure online)

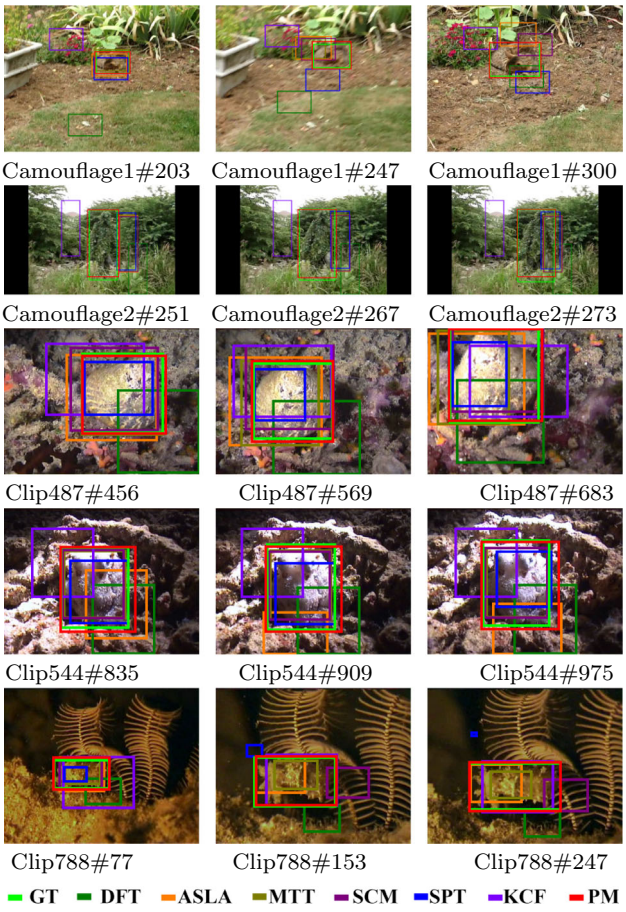


Fig. 19 continued

Table 8 Average execution time (seconds) for tracking object in a single frame

Sequences	DFT	ASLA	MTT	SCM	SPT	KCF	PM
200	**0.1793**	*0.4450*	1.0429	4.1442	2.5735	***0.0051***	0.5321

Bolditalics indicates best value, *bold* indicates second best, *italics* indicates third best

proposed fuzzy energy function. On the other hand, higher value dfbr extracting exact boundary of the object in the tth frame. Moderate value of β ($= 0.5$) indicates that the global and local terms have equal contribution in the proposed fuzzy energy function.

To visually illustrate the effect of parameter β, tracking results of Clip249 sequence for different values of β are given in Fig. 23. From this figure (Fig. 23d), it is seen that tracking result is not proper due to more contribution of the global term than the local term and is unable to extract the boundary of the objects. Figure 23f, h, j highlight that similar tracking results are obtained with different values of β=0.7, 0.5, 0.1, 0.01. Detection accuracy and tracking accuracy for different values of parameter β are summarized in Tables 10 and 11, respectively. Table 10 highlights that similar object detection

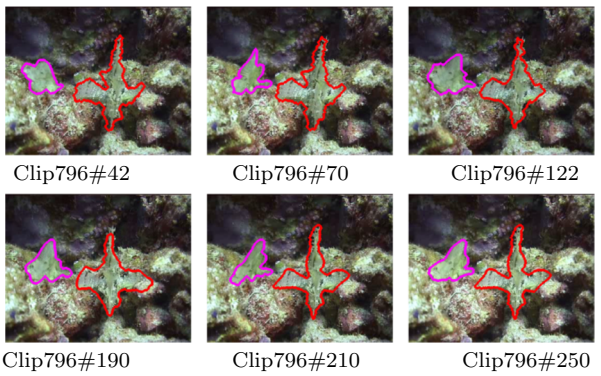


Fig. 20 Tracking results of multiple objects on Clip796 sequence

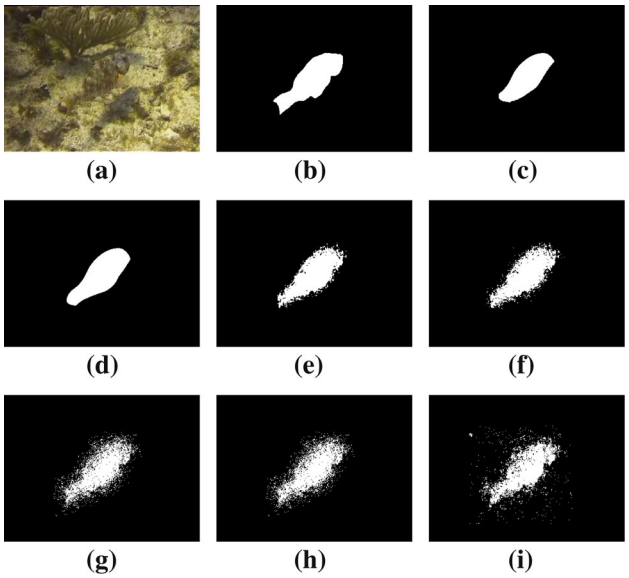


Fig. 21 **a** Original frame; **b** ground truth; object detection in Clip152 sequence using the modified PNN with **c** $\sigma = 100.0$; **d** $\sigma = 30.0$; **e** $\sigma = 5.0$; **f** $\sigma = 1.0$; **g** $\sigma = 0.5$; **h** $\sigma = 0.1$ and **i** $\sigma = 0.05$

result is obtained by varying the value of parameter β as it does not have any role for detection whereas Table 11 indicates that different tracking results are obtained by varying values of β. However, in this work we chose $\beta = 0.5$ for all experiments.

4.4 Effect of Parameters λ_1 and λ_2

Two parameters λ_1 and λ_2 in the proposed energy function [given in Eq. (15)] are the weights of integrals over regions Ω_1 [i.e., inside (C)] and Ω_2 [i.e., outside (C)] of the contour C. For all experiments, we consider λ_1=λ_2 to balance the impotence of each side during the minimization of the proposed energy function given in Eq. (15). If we consider λ_1 to be greater than λ_2, then the region inside the contour i.e., Ω_1 is given more importance than the outside region of the contour i.e., Ω_2 and vice versa. Figure 24 displays

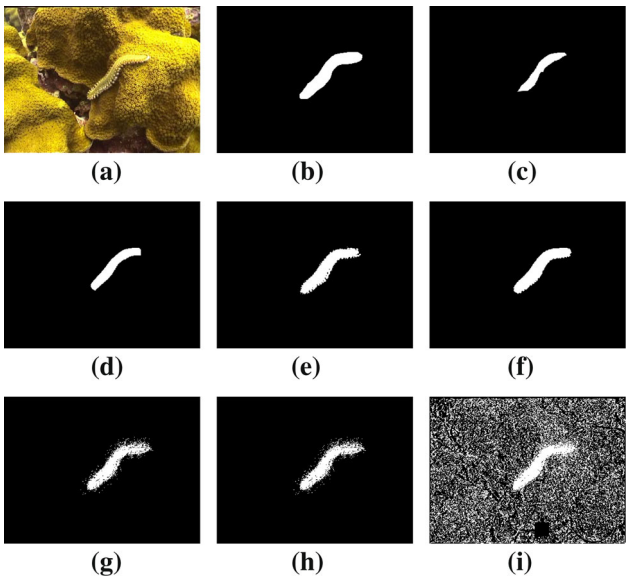


Fig. 22 **a** Original frame; **b** ground truth; object detection in Clip249 sequence using the modified PNN with **c** $\sigma = 100.0$; **d** $\sigma = 30.0$; **e** $\sigma = 5.0$; **f** $\sigma = 1.0$; **g** $\sigma = 0.5$; **h** $\sigma = 0.1$ and **i** $\sigma = 0.05$

Table 9 Object detection accuracy for different values of σ

Different values of σ							
Sequence	$\sigma = 100.0$	$\sigma = 30.0$	$\sigma = 5.0$	$\sigma = 1.0$	$\sigma = 0.5$	$\sigma = 0.1$	$\sigma = 0.05$
Clip152	0.58592	0.6879	0.9886	0.9866	0.9753	0.9763	0.7845
Clip249	0.6928	0.7923	0.9953	0.9956	0.9926	0.9922	0.6907

the tracking results for different values of λ_1 and λ_2. From these figures, it is observed that when $\lambda_1 > \lambda_2$, contour is moved towards the inner region of the target (see Fig. 24d). When $\lambda_1 < \lambda_2$, contour is moved away from the exact object boundary. For visual illustration, see Figs. 24f, h, j. On the other hand, for $\lambda_1 = \lambda_2$, the exact contour of the object is obtained. Figure 24(l) displays the tracking result when $\lambda_1 = \lambda_2$. Tables 12 and 13 summarize the detection accuracy and tracking accuracy for different values of λ_1 and λ_2, respectively. Table 12 highlights that for different values of λ_1 and λ_2, similar object detection results are obtained. Whereas Table 13 highlights that the best tracking result is obtained when λ_1=λ_2. In our work, $\lambda_1 = \lambda_2 = 1$ is considered for all experiments.

4.5 Occlusion Handling

Several factors affect the performance of tracking algorithm. Occlusion is one of such factor. Two types of occlusion may occur: (i) occlusions between two or more objects of interests; (ii) occlusions between an object of interest and an object of the background. Both of them may be either partial or full. Here, behavior of the proposed algorithm for handling last problem is discussed. In the proposed method, modified PNN (pixel based classification technique) detects the target

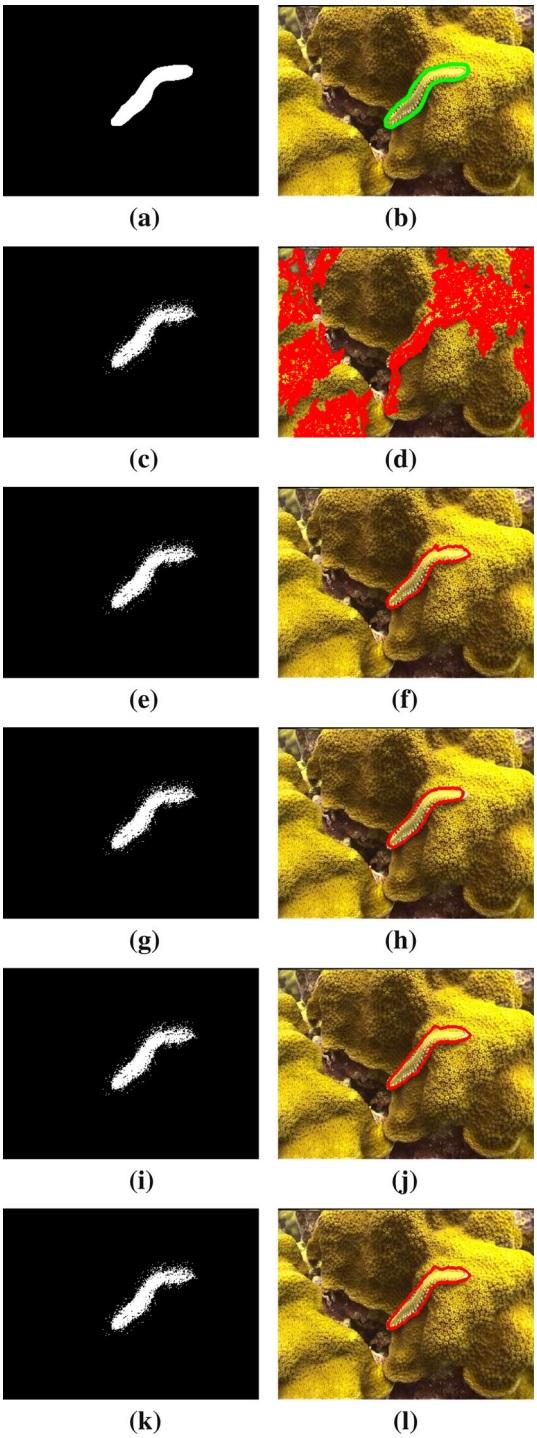


Fig. 23 **a** and **b** Ground truth object detection and tracker; Results of camouflaged object detection and tracking in Clip249 sequence using the modified PNN and the proposed fuzzy active contour with **c** and **d** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.9$; **e** and **f** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.7$; **g** and **h** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.5$; **i** and **j** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.1$; **k** and **l** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.01$

Table 10 Object detection accuracy for different values of β

Different values of β					
Sequence	β=0.9	β=0.7	β=0.5	β=0.1	β=0.01
Clip249	0.9926	0.9926	0.9926	0.9926	0.9926

Table 11 Tracking accuracy for different values of β

Different values of β					
Sequence	β=0.9	β=0.7	β=0.5	β=0.1	β=0.01
Clip249	0.0719	0.8375	0.8352	0.8355	0.8314

and fuzzy active contour extracts the contour of the target. The proposed method can track objects with partial occlusion (less than 20 %). Figure 25a shows the performance of the proposed algorithm for tracking (less than 20 %) occluded object. For partial occlusion (more than 20 %) and full occlusion, the proposed technique fails to track the object due to inaccurate detection results and loss of object information. Figure 25b, c show the poor performance of the proposed technique for tracking occluded (more than 20% for partial and full) objects. Table 14 shows the quantitative measures for performance of the proposed method for occluded object tracking. Similar observation is found from this table.

4.6 Future Work

The proposed technique considers only three features for integrating. In future, motion information may be taken into account to generate more features. In this work, all features are given equal weights during integration. Weights based on importance of each feature may be provided during integration in future. We will also consider some other color spaces other than CIElab (developed based on human eye or biological vision system) in future.

Though, in this work σ is chosen experimentally, in literature, several techniques (Mao et al. 2000; Zhong et al. 2007; Kusy and Kluska 2013) are proposed to select proper value of σ. σ may be selected automatically by developing some optimization technique.

The proposed method does not explicitly consider occlusion problem. Attention may be given to develop, some objective functions based on object detection results (obtained by the modified PNN) to detect occurrence of occlusion automatically. Due to occlusion, object information is not available in that frame. So prior information i.e., shape of object (from previous frame) may be considered as object information in the frame where occlusion occurred. To estimate the location of the target, particle filter can be considered. Therefore, use of shape prior and particle filter with the proposed method may help handle occlusion problem in a better way.

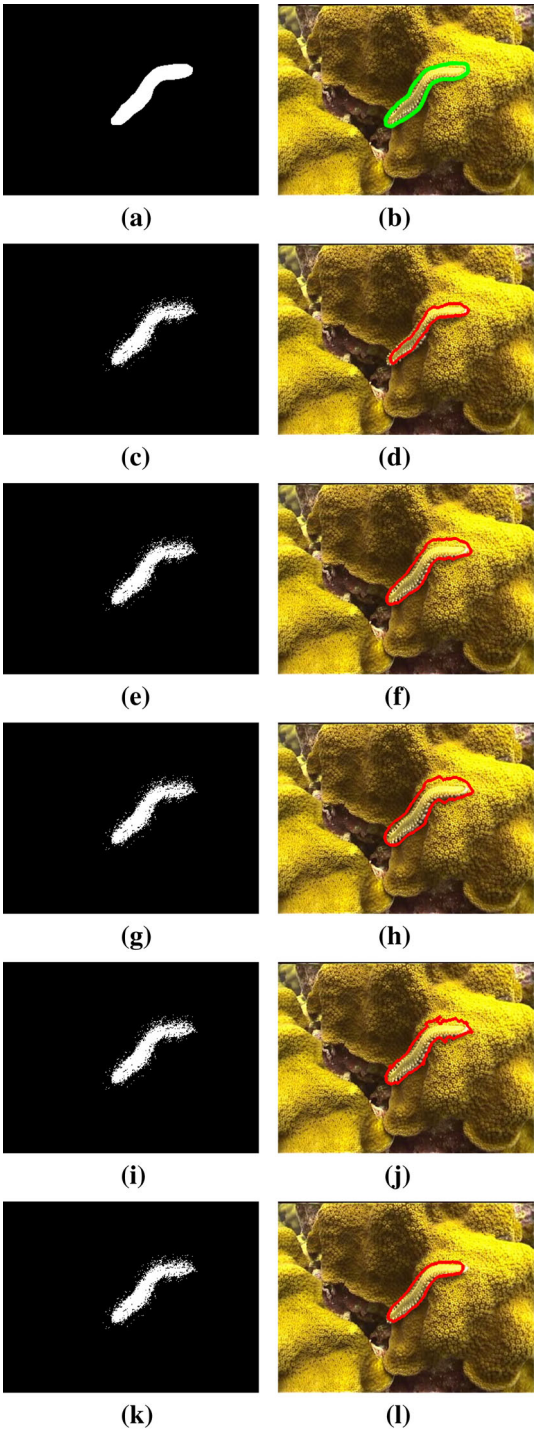


Fig. 24 **a** and **b** Ground truth object detection and tracker; camouflaged object detection and tracking in Clip249 sequence using the modified PNN and the proposed fuzzy active contour with **c** and **d** $\sigma = 0.1$, $\lambda_1 = 1.1$, $\lambda_2 = 1$, $\beta = 0.5$; **e** and **f** $\sigma = 0.1$, $\lambda_1 = 1$, $\lambda_2 = 1.1$, $\beta = 0.5$; **g** and **h** $\sigma = 0.1$, $\lambda_1 = 1$, $\lambda_2 = 1.3$, $\beta = 0.5$; **i** and **j** $\sigma = 0.1$, $\lambda_1 = 1$, $\lambda_2 = 1.5$, $\beta = 0.5$; **k** and **l** $\sigma = 0.1$, $\lambda_1 = \lambda_2 = 1$, $\beta = 0.5$

Table 12 Object detection accuracy based on L_1: λ_1=1.1 & λ_2=1.0; L_2: λ_1=1.0 & λ_2=1.1; L_3: λ_1=1.0 & λ_2=1.3; L_4: λ_1=1.0 & λ_2=1.5 and L_5: λ_1=1.0 & λ_2 =1.0

Different values of λ_1 and λ_2					
Sequence	L_1	L_2	L_3	L_4	L_5
Clip249	0.9926	0.9926	0.9926	0.9926	0.9926

Table 13 Tracking accuracy based on L_1: λ_1=1.1 & λ_2=1.0; L_2: λ_1=1.0 & λ_2=1.1; L_3: λ_1=1.0 & λ_2=1.3; L_4: λ_1=1.0 & λ_2=1.5 and L_5: λ_1=1.0 & λ_2 =1.0

Different values of λ_1 and λ_2					
Sequence	L_1	L_2	L_3	L_4	L_5
Clip249	0.7879	0.6938	0.6059	0.6912	***0.8532***

BoldItalics indicates best result

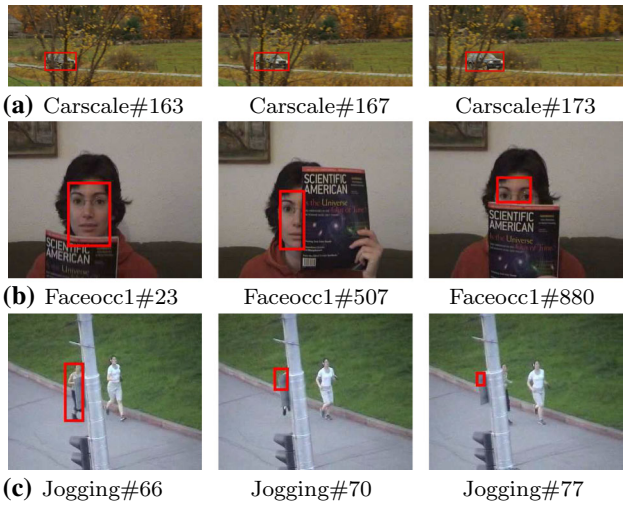


Fig. 25 Performance of the proposed algorithm for tracking occluded object

Table 14 Tracking accuracy of the occluded object using the proposed method

Sequence	Average center location	Average tracking accuracy
Carscale	3.9492	0.8192
Faceocc1	22.1411	0.3752
Jogging	15.7198	0.3955

5 Conclusions

In this article, a tracking-by-detection algorithm is proposed to detect and track camouflaged objects. We propose to represent an object in terms of feature vectors of color, shape and texture which increases the separability between the camouflaged object and background. Object is detected by the modified probabilistic neural network and finally tracking is done by devising fuzzy energy based active contour model. The proposed technique is able to track objects under camouflaged as well as several other (illumination variation, scale change, rotation, pose change) complex environments. Experimental results and analysis demonstrate that the proposed method outperforms state-of-the-art techniques of object detection for various complex (including camouflaged) environments.

Acknowledgements The authors like to thank the editor and 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.

Appendix 1

The proposed energy function F in Eq. (15) is convex with respect to u.

Proof:
Let

$$F_A(\mathbf{X}^t) = \int_{\Omega} \left[u(\mathbf{X}^t)\right]^m \left\| I(\mathbf{X}^t) - C_1 \right\|^2 d\mathbf{X}^t,\ \mathbf{X}^t \in \Omega \equiv \Re^2 \tag{20}$$

and

$$f_A(\mathbf{X}^t) = \left[u(\mathbf{X}^t)\right]^m \left\| I(\mathbf{X}^t) - C_1 \right\|^2, \tag{21}$$

where $f_A : \Omega \to \Re$. Therefore, $F_A(\mathbf{X}^t) = \int_{\Omega} f_A(\mathbf{X}^t) d\mathbf{X}^t$.

Now let $\mathbf{X}_1 \equiv (x_1, y_1)$ and $\mathbf{X}_2 \equiv (x_2, y_2) \in \Omega$. For any $\theta \in [0, 1]$, we have

$$\begin{aligned}
&\theta \mathbf{X}_1 + (1-\theta)\mathbf{X}_2 \\
&= \theta\,(x_1, y_1) + (1-\theta)\,(x_2, y_2) \\
&= (\theta x_1 + (1-\theta)x_2, \theta y_1 + (1-\theta)y_2) \\
&= (\theta\,(x_1 - x_2) + x_2, \theta\,(y_1 - y_2) + y_2) \in \Omega \equiv \Re^2.
\end{aligned}$$

Since $\theta\,(x_1 - x_2) + x_2 \in \Re$ as $x_1, x_2 \in \Re$ and $\theta \in [0, 1]$; and $\theta\,(y_1 - y_2) + y_2 \in \Re$ as $y_1, y_2 \in \Re$ and $\theta \in [0, 1]$. Therefore, the domain of f_A i.e., $\Omega \equiv \Re^2$ is convex.

Differentiating Eq. (21) w. r. t. u, we have

$$\frac{\partial f_A}{\partial u} = m \left[u(\mathbf{X}^t)\right]^{m-1} \left\| I(\mathbf{X}^t) - c_1 \right\|^2.$$

Again differentiating $\frac{\partial f_A}{\partial u}$ w. r. t. u, we have

$$\frac{\partial^2 f_A}{\partial u^2} = m(m-1) \left[u(\mathbf{X}^t)\right]^{m-2} \left\| I(\mathbf{X}^t) - c_1 \right\|^2.$$

Now $\frac{\partial^2 f_A}{\partial u^2} \geq 0$, since $m > 1$, $u(\mathbf{X}^t) \in [0,1]$ and $\left\|I(\mathbf{X}^t) - c_1\right\|^2 \geq 0$.

Since domain of f_A is convex and $\frac{\partial^2 f_A}{\partial u^2} \geq 0$, therefore f_A is convex. Thus, $\forall \mathbf{Y}_1, \mathbf{Y}_2 \in \Omega \equiv \Re^2$ and $\theta \in [0,1]$. The following relation

$$f_A(\theta \mathbf{X}_1 + (1-\theta_2)\mathbf{X}_2) \leq \theta f_A(\mathbf{X}_1) + (1-\theta) f_A(\mathbf{X}_2) \tag{22}$$

holds. Integrating both the sides of Eq.(22), we have

$$\int_{\Omega} f_A(\theta \mathbf{X}_1 + (1-\theta)\mathbf{X}_2) d\mathbf{X}^t \leq \theta_2 \int_{\Omega} f_A(\mathbf{X}_1) d\mathbf{X}^t + (1-\theta) \int_{\Omega} f_A(\mathbf{X}_2) d\mathbf{X}^t. \tag{23}$$

But $\int_{\Omega} f_A(\mathbf{X}^t) d\mathbf{X}^t = F_A(\mathbf{X}^t)$. Therefore, we have

$$F_A(\theta \mathbf{X}_1 + (1-\theta)\mathbf{X}_2) \leq \theta_2 F_A(\mathbf{X}_1) + (1-\theta) F_A(\mathbf{X}_2).$$

Hence, F_A is convex.

Similarly, let

$$F_B(\mathbf{X}^t) = \int_{\Omega} \left[1 - u(\mathbf{X}^t)\right]^m \left\|I(\mathbf{X}^t) - c_2\right\|^2 d\mathbf{X}^t, \tag{24}$$

and

$$f_B(\mathbf{X}^t) = \left[1 - u(\mathbf{X}^t)\right]^m \left\|I(\mathbf{X}^t) - c_2\right\|^2, \tag{25}$$

where $f_B : \Omega \to \Re$. Therefore $F_B(\mathbf{X}^t) = \int_{\Omega} f_B(\mathbf{X}^t) d\mathbf{X}^t$. In a similar way, we can prove that F_B is also convex.

Again let

$$F_{C_1}(\mathbf{Y}^t) = \int_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u(\mathbf{Y}^t)\right]^m \left\|I(\mathbf{Y}^t) - c_1\right\|^2 d\mathbf{Y}^t, \tag{26}$$

$$F_C(\mathbf{X}^t) = \int_{\Omega} \left[u(\mathbf{X}^t)\right]^m F_{C_1}(\mathbf{Y}^t) d\mathbf{X}^t, \tag{27}$$

$$f_{C_1}(\mathbf{Y}^t) = W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u\left(\mathbf{Y}^t\right)\right]^m \left\|I(\mathbf{Y}^t) - c_1\right\|^2, \tag{28}$$

and

$$f_C(\mathbf{X}^t) = \left[u(\mathbf{X}^t)\right]^m f_{C_1}(\mathbf{Y}^t),\ \mathbf{X}^t, \mathbf{Y}^t \in \Omega \equiv \Re^2. \tag{29}$$

Therefore $F_{C_1}(\mathbf{Y}^t) = \int_{\Omega} f_{C_1}(\mathbf{Y}^t) d\mathbf{Y}^t$ and $F_C(\mathbf{X}^t) = \int_{\Omega} f_C(\mathbf{X}^t) d\mathbf{X}^t$. Here $f_{C_1} : \Omega \to \Re$ and the domain of f_{C_1} is convex.

Now differentiating Eq. (28) w. r. t. u, we have

$$\frac{\partial f_{C_1}}{\partial u} = -m W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u(\mathbf{Y}^t)\right]^{m-1} \left\|I(\mathbf{Y}^t) - c_1\right\|^2.$$

Again differentiating $\frac{\partial f_{C_1}}{\partial u}$ w. r. t. u, we have

$$\frac{\partial^2 f_{C_1}}{\partial u^2} = m(m-1) W_{\mathbf{X}^t\mathbf{Y}^t} \left[1 - u(\mathbf{Y}^t)\right]^{m-2} \left\|I(\mathbf{Y}^t) - c_1\right\|^2.$$

Now $\frac{\partial^2 f_{C_1}}{\partial u^2} \geq 0$, since $m > 1$, $W_{\mathbf{X}^t\mathbf{Y}^t} \geq 0$, $u(\mathbf{Y}^t) \in [0,1]$ and $\left\|I(\mathbf{Y}^t) - c_1\right\|^2 \geq 0$.

Since domain of f_{C_1} is convex and $\frac{\partial^2 f_{C_1}}{\partial u^2} \geq 0$, hence f_{C_1} is convex. Therefore $\forall \mathbf{Y}_1, \mathbf{Y}_2 \in \Omega$ and $\theta \in [0,1]$. The following relation

$$f_{C_1}(\theta \mathbf{Y}_1 + (1-\theta)\mathbf{Y}_2) \leq \theta f_{C_1}(\mathbf{Y}_1) + (1-\theta) f_{C_1}(\mathbf{Y}_2) \tag{30}$$

holds. Integrating both the sides, we have

$$\int_{\Omega} f_{C_1}(\theta \mathbf{Y}_1 + (1-\theta)\mathbf{Y}_2) d\mathbf{Y}^t \leq \theta \int_{\Omega} f_{C_1}(\mathbf{Y}_1) d\mathbf{Y}^t + (1-\theta) \int_{\Omega} f_{C_1}(\mathbf{Y}_2) d\mathbf{Y}^t,\ \forall \mathbf{Y}^t \in \Omega. \tag{31}$$

But $F_{C_1} = \int_{\Omega} f_{C_1}(\mathbf{Y}) d\mathbf{Y}^t$, hence F_{C_1} is convex. Since f_{C_1} is convex and $\left[u(\mathbf{X}^t)\right]^m \geq 0$, then $f_C = \left[u(\mathbf{X}^t)\right]^m f_{C_1}$ is also convex.

Again let $\overline{F_C} = \left[u(\mathbf{X}^t)\right]^m F_{C_1}$. Since F_{C_1} is convex and $\left[u(\mathbf{X}^t)\right]^m \geq 0$, therefore $\overline{F_C}$ is also convex. Therefore $\forall \mathbf{X}_1, \mathbf{X}_2 \in \Omega$ and $\theta \in [0,1]$. The following equation

$$\overline{F_C}(\theta \mathbf{X}_1 + (1-\theta)\mathbf{X}_2) \leq \theta \overline{F_C}(\mathbf{X}_1) + (1-\theta) \overline{F_C}(\mathbf{X}_2) \tag{32}$$

holds. Then integrating both the sides, we have

$$\int_{\Omega} \overline{F_C}(\theta \mathbf{X}_1 + (1-\theta)\mathbf{X}_2) d\mathbf{X}^t \leq \theta \int_{\Omega} \overline{F_C}(\mathbf{X}_1) d\mathbf{X}^t + (1-\theta) \int_{\Omega} \overline{F_C}(\mathbf{X}_2) d\mathbf{X}^t, \forall \mathbf{X}^t \in \Omega. \tag{33}$$

Therefore $F_C = \int_{\Omega} \overline{F_C}(\mathbf{X}) d\mathbf{X}^t = \int_{\Omega} \left[u(\mathbf{X}^t)\right]^m F_{C_1} d\mathbf{X}^t$ is convex. In a similar way, it can be shown that

$$F_D = \int_{\Omega} \left[1 - u(\mathbf{X}^t)\right]^m \left\{ \int_{\Omega} W_{\mathbf{X}^t\mathbf{Y}^t} \left[u(\mathbf{Y}^t)\right]^m \left\| I(\mathbf{Y}^t) - c_2 \right\|^2 d\mathbf{Y}^t \right\} d\mathbf{X}^t \tag{34}$$

is also convex. Since $0 \leq \beta \leq 1$, $1 - \beta \geq 0$, $\lambda_1, \lambda_2 > 0$, then F is the weighted sum of four (F_A, F_B, F_C, and F_D) convex functions. i.e.,

$$F = \lambda_1 \beta F_A + \lambda_2 \beta F_B + \lambda_1 (1-\beta) F_C + \lambda_2 (1-\beta) F_D.$$

Hence, F is convex with respect to u. Therefore, the proposed energy function is convex with respect to membership function $u(\mathbf{X}^t)$. □

Appendix 2

Since an image is discrite in nature, instead of integration, summation is considered here.

Let us assume two prototypes c_1 and c_2 which approximate the image intensity $inside(C)$ and $outside(C)$ the contour C. Thus it can be written as

$$c_1 = \frac{\sum_{\Omega} [u(X)]^m I(X)}{\sum_{\Omega} [u(X)]^m} \tag{35}$$

and

$$c_2 = \frac{\sum_{\Omega} [1 - u(X)]^m I(X)}{\sum_{\Omega} [1 - u(X)]^m}, \tag{36}$$

where $I(X)$ is the intensity value at pixel location X, $u(X)$ is the degree of membership of pixel X inside C, and m is the fuzzifier which determines the fuzziness present in the given image.

Therefore, total fuzzy energy for the whole image can be computed as

$$\begin{aligned} F &= \lambda_1 \beta \sum_{\Omega} [u(X)]^m \left\| I(X) - c_1 \right\|^2 \\ &\quad + \lambda_2 \beta \sum_{\Omega} [1 - u(X)]^m \left\| I(X) - c_2 \right\|^2 \\ &\quad + \lambda_1 (1-\beta) \sum_{\Omega} [u(X)]^m h_1 \\ &\quad + \lambda_2 (1-\beta) \sum_{\Omega} [1 - u(X)]^m h_2 \\ &= F_A + F_B + F_C + F_D. \end{aligned} \tag{37}$$

where

$$\begin{aligned} F_A &= \lambda_1 \beta \sum_{\Omega} [u(X)]^m \left\| I(X) - c_1 \right\|^2, \\ F_B &= \lambda_2 \beta \sum_{\Omega} [1 - u(X)]^m \left\| I(X) - c_2 \right\|^2, \\ F_C &= \lambda_1 (1-\beta) \sum_{\Omega} [u(X)]^m h_1, \\ F_D &= \lambda_2 (1-\beta) \sum_{\Omega} [1 - u(X)]^m h_2, \end{aligned} \tag{38}$$

with

$$\begin{aligned} h_1 &= \sum_{\Omega} W_{XY} [1 - u(Y)]^m \left\| I(Y) - c_1 \right\|^2, \\ h_2 &= \sum_{\Omega} W_{XY} [u(Y)]^m \left\| I(Y) - c_2 \right\|^2. \end{aligned} \tag{39}$$

Let us assume that a pixel $P \in I$ with intensity I_o and degree of membership u_o. If we change the degree of membership of pixel P to the value u_n using Eq. (18), then c_1 and c_2 will be changed to new values $\overline{c_1}$ and $\overline{c_2}$, respectively. The new values of c_1 and c_2 are calculated as

$$\begin{aligned} \overline{c_1} &= \frac{\sum_{\Omega} [\overline{u}(X)]^m I(X)}{\sum_{\Omega} [\overline{u}(X)]^m} \\ &= \frac{\sum_{\Omega} [u(X)]^m I(X) + \left(u_n^m - u_o^m\right) I_o}{\sum_{\Omega} [u(X)]^m + u_n^m - u_o^m} \\ &= \frac{c_1 \sum_{\Omega} [u(X)]^m + \left(u_n^m - u_o^m\right) I_o}{\sum_{\Omega} [u(X)]^m + u_n^m - u_o^m} \\ &= \frac{c_1 a_1 + \left(u_n^m - u_o^m\right) I_o}{a_1 + u_n^m - u_o^m} \\ &= \frac{c_1 \left(a_1 + u_n^m - u_o^m\right) - c_1 \left(u_n^m - u_o^m\right) + \left(u_n^m - u_o^m\right) I_o}{a_1 + u_n^m - u_o^m} \\ &= c_1 + \frac{\left(u_n^m - u_o^m\right) I_o - c_1 \left(u_n^m - u_o^m\right)}{a_1 + u_n^m - u_o^m} \\ &= c_1 + \frac{\left(u_n^m - u_o^m\right) \left\| I_o - c_1 \right\|}{a_1 + u_n^m - u_o^m} \\ &= c_1 + S_1 \left\| I_o - c_1 \right\| \end{aligned} \tag{40}$$

where $a_1 = \sum_{\Omega} [u(X)]^m$ and $S_1 = \frac{u_n^m - u_o^m}{\sum_{\Omega} [u(X)]^m + u_n^m - u_o^m} = \frac{u_n^m - u_o^m}{a_1 + u_n^m - u_o^m}$.

Similarly,

$$\begin{aligned}\overline{c_2} &= \frac{\sum_{\Omega}[1-\overline{u}(X)]^m I(X)}{\sum_{\Omega}[1-\overline{u}(X)]^m} \\ &= c_2 + \frac{\left[(1-u_n)^m-(1-u_o)^m\right]\|I_o-c_2\|}{a_2+(1-u_n)^m-(1-u_o)^m} \\ &= c_2 + S_2\|I_o-c_2\| \end{aligned} \tag{41}$$

where $a_2 = \sum_{\Omega}[1-u(X)]^m$ and $S_2 = \frac{(1-u_n)^m-(1-u_o)^m}{a_2+(1-u_n)^m-(1-u_o)^m}$.

From Eq. (15), it is seen that if degree of membership $u(X)$ is changed, then the energy of the model will also be changed. If F denotes the old energy and $\overline{F}$ denotes the new energy due to changing of degree of membership of the point P, then

$$\overline{F} = \overline{F}_A + \overline{F}_B + \overline{F}_C + \overline{F}_D, \tag{42}$$

with

$$\begin{aligned}\overline{F}_A &= \lambda_1\beta\sum_{\Omega}[\overline{u}(X)]^m\|I(X)-\overline{c_1}\|^2 \\ &= \lambda_1\beta\left\{\sum_{\Omega}[u(X)]^m\|I(X)-\overline{c_1}\|^2\right. \\ &\quad \left.+\left(u_n^m-u_o^m\right)\|I_o-\overline{c_1}\|^2\right\} \\ &= F_A+\lambda_1\beta\|I_o-c_1\|^2\left[a_1+\left(u_n^m-u_o^m\right)\right] \\ &\quad S_1\left(\frac{a_1}{a_1+\left(u_n^m-u_o^m\right)}\right) \\ &= F_A+\lambda_1\beta S_1 a_1\|I_o-c_1\|^2. \end{aligned} \tag{43}$$

Similarly,

$$\overline{F}_B = F_B+\lambda_2\beta a_2 S_2\|I_o-c_2\|^2. \tag{44}$$

Then

$$\begin{aligned}\overline{F}_C &= \lambda_1(1-\beta)\sum_{\Omega}[\overline{u}(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\} \\ &= \lambda_1(1-\beta)\sum_{\Omega}[u(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\} \\ &\quad+\lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\} \\ &\quad+\lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I_o-\overline{c}_1\|^2\right\}. \end{aligned} \tag{45}$$

Let

$$\begin{aligned}\overline{F}_{C_1} &= \lambda_1(1-\beta)\sum_{\Omega}[u(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\}, \\ \overline{F}_{C_2} &= \lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\}, \\ \overline{F}_{C_3} &= \lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I_o-\overline{c}_1\|^2\right\}. \end{aligned} \tag{46}$$

Now,

$$\begin{aligned}\overline{F}_{C_1} &= \lambda_1(1-\beta)\sum_{\Omega}[u(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\} \\ &= F_C+\lambda_1(1-\beta)\sum_{\Omega}[u(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m S_1^2\|I_o-c_1\|^2\right\} \\ &\quad-2\lambda_1(1-\beta)\sum_{\Omega}[u(X)]^m \\ &\quad\left\{\sum_{\Omega}W_{XY}[1-u(Y)]^m\|I(Y)-c_1\|S_1\|I_o-c_1\|\right\}. \end{aligned} \tag{47}$$

$$\begin{aligned}\overline{F}_{C_2} &= \lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I(Y)-\overline{c}_1\|^2\right\} \\ &= \lambda_1(1-\beta)\left(u_n^m-u_o^m\right) \\ &\quad\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I(Y)-c_1-S_1(I_o-c_1)\|^2\right\}. \end{aligned} \tag{48}$$

$$\overline{F}_{C_3} = \lambda_1(1-\beta)\left(u_n^m-u_o^m\right)\left\{\sum_{\Omega}W_{oY}[1-u(Y)]^m\|I_o-\overline{c}_1\|^2\right\}$$

$$= \lambda_1(1-\beta)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I_o - c_1\right\|^2 \left[1-S_1\right]^2\right\}. \quad (49)$$

Therefore, we have $\overline{F}_C = \overline{F}_{C_1} + \overline{F}_{C_2} + \overline{F}_{C_3}$.

$$\begin{aligned}\overline{F}_C = {} & F_C + \lambda_1(1-\beta)\sum_{\Omega}\left[u\left(X\right)\right]^m \left\{\sum_{\Omega} W_{XY}\left[1-u\left(Y\right)\right]^m S_1^2 \left\|I_o - c_1\right\|^2\right\} \\ & - 2\lambda_1(1-\beta)\sum_{\Omega}\left[u\left(X\right)\right]^m \left\{\sum_{\Omega} W_{XY}\left[1-u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_1\right\| S_1 \left\|I_o - c_1\right\|\right\} \\ & + \lambda_1(1-\beta)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_1 - S_1\left(I_o - c_1\right)\right\|^2\right\} \\ & + \lambda_1(1-\beta)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I_o - c_1\right\|^2 \left[1-S_1\right]^2\right\}. \quad (50)\end{aligned}$$

Similarly, we have

$$\begin{aligned}\overline{F}_D = {} & F_D + \lambda_2(1-\beta)\sum_{\Omega}\left[1-u\left(X\right)\right]^m \left\{\sum_{\Omega} W_{XY}\left[u\left(Y\right)\right]^m S_2^2 \left\|I_o - c_2\right\|^2\right\} \\ & - 2\lambda_2(1-\beta)\sum_{\Omega}\left[1-u\left(X\right)\right]^m \left\{\sum_{\Omega} W_{XY}\left[u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_2\right\| S_2 \left\|I_o - c_2\right\|\right\} \\ & + \lambda_2(1-\beta)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_2 - S_2\left(I_o - c_2\right)\right\|^2\right\} \\ & + \lambda_2(1-\beta)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I_o - c_2\right\|^2 \left[1-S_2\right]^2\right\}. \quad (51)\end{aligned}$$

Therefore, we have

$$\begin{aligned}\overline{F} = {} & \overline{F}_A + \overline{F}_B + \overline{F}_C + \overline{F}_D \\ = {} & F + \lambda_1\beta S_1 a_1 \left\|I_o - c_1\right\|^2 + \lambda_2\beta S_2 a_2 \left\|I_o - c_2\right\|^2 \\ & - 2\lambda_1\left(1-\beta\right) S_1 \left\|I_o - c_1\right\| b_1 + \lambda_1\left(1-\beta\right) g_1 \left[\left\|c_1 + S_1\left(I_o - c_1\right)\right\|^2 - \left\|c_1\right\|^2\right] \\ & + \lambda_1\left(1-\beta\right)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_1 - S_1\left(I_o - c_1\right)\right\|^2\right\} \\ & + \lambda_1\left(1-\beta\right)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I_o - c_1\right\|^2 \left[1-S_1\right]^2\right\} \\ & - 2\lambda_2\left(1-\beta\right) S_2 \left\|I_o - c_2\right\| b_2 + \lambda_2\left(1-\beta\right) g_2 \left[\left\|c_2 + S_2\left(I_o - c_2\right)\right\|^2 - \left\|c_2\right\|^2\right] \\ & + \lambda_2\left(1-\beta\right)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_2 - S_2\left(I_o - c_2\right)\right\|^2\right\} \\ & + \lambda_2\left(1-\beta\right)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I_o - c_2\right\|^2 \left[1-S_2\right]^2\right\}. \quad (52)\end{aligned}$$

Thus the energy difference ΔF between old energy (F) and new energy ($\overline{F}$) due to change of degree of membership value can be obtained as

$$\begin{aligned}\Delta F = {} & \overline{F} - F \\ = {} & \lambda_1\beta S_1 a_1 \left\|I_o - c_1\right\|^2 + \lambda_2\beta S_2 a_2 \left\|I_o - c_2\right\|^2 \\ & - 2\lambda_1\left(1-\beta\right) S_1 \left\|I_o - c_1\right\| b_1 + \lambda_1\left(1-\beta\right) g_1 \left[\left\|c_1 + S_1\left(I_o - c_1\right)\right\|^2 - \left\|c_1\right\|^2\right] \\ & + \lambda_1\left(1-\beta\right)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_1 - S_1\left(I_o - c_1\right)\right\|^2\right\} \\ & + \lambda_1\left(1-\beta\right)\left(u_n^m - u_o^m\right) \left\{\sum_{\Omega} W_{oY}\left[1-u\left(Y\right)\right]^m \left\|I_o - c_1\right\|^2 \left[1-S_1\right]^2\right\} \\ & - 2\lambda_2\left(1-\beta\right) S_2 \left\|I_o - c_2\right\| b_2 + \lambda_2\left(1-\beta\right) g_2 \left[\left\|c_2 + S_2\left(I_o - c_2\right)\right\|^2 - \left\|c_2\right\|^2\right] \\ & + \lambda_2\left(1-\beta\right)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I\left(Y\right) - c_2 - S_2\left(I_o - c_2\right)\right\|^2\right\} \\ & + \lambda_2\left(1-\beta\right)\left[(1-u_n)^m - (1-u_o)^m\right] \left\{\sum_{\Omega} W_{oY}\left[u\left(Y\right)\right]^m \left\|I_o - c_2\right\|^2 \left[1-S_2\right]^2\right\}, \quad (53)\end{aligned}$$

where

$$a_1 = \sum_{\Omega}\left[u\left(X\right)\right]^m$$

$$a_2 = \sum_{\Omega}\left[1-u\left(X\right)\right]^m$$

$$S_1 = \frac{u_n^m - u_o^m}{\sum_{\Omega} [u(X)]^m + u_n^m - u_o^m}$$

$$S_2 = \frac{(1-u_n)^m - (1-u_o)^m}{\sum_{\Omega} [1-u(X)]^m + (1-u_n)^m - (1-u_o)^m}$$

$$b_1 = \sum_{\Omega} [u(X)]^m \left\{ \sum_{\Omega} W_{XY} [1-u(Y)]^m I(Y) \right\}$$

$$b_2 = \sum_{\Omega} [1-u(X)]^m \left\{ \sum_{\Omega} W_{XY} [u(Y)]^m I(Y) \right\}$$

$$g_1 = \sum_{\Omega} [u(X)]^m \left\{ \sum_{\Omega} W_{XY} [1-u(Y)]^m \right\}$$

$$g_2 = \sum_{\Omega} [1-u(X)]^m \left\{ \sum_{\Omega} W_{XY} [u(Y)]^m \right\}.$$

References

Akhloufi, M.A., & Bendada, A. (2010). Locally adaptive texture features for multispectral face recognition. In *IEEE International Conference on Systems Man and Cybernetics (SMC)* (pp. 3308–3314).

Avidan, S. (2007). Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(2), 261–271.

Babenko, B., Yang, M., & Belongie, S. (2011). Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1619–1632.

Bandouch, J., Jenkins, O. C., & Beetz, M. (2012). A self-training approach for visual tracking and recognition of complex human activity patterns. *International Journal of Computer Vision*, *99*(2), 166–189.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Boult, T. E., Micheals, R. J., Gao, X., & Eckmann, M. (2001). Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. *Proceedings of the IEEE*, *89*(10), 1382–1402.

Burrascano, P. (1990). Learning vector quantization for the probabilistic neural network. *IEEE transactions on Neural Networks*, *2*(4), 458–461.

Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic active contours. *International journal of Computer Vision*, *22*(1), 61–79.

Challa, S., Morelande, M. R., Musicki, D., & Evans, R. J. (2011). *Fundamentals of object tracking*. Cambridge: Cambridge University Press.

Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, *10*(2), 266–277.

Chandesa, T., Pridmore, T., Bargiela, A. (2009). Detecting occlusion and camouflage during visual tracking. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)* (pp. 468–473).

Cohen, L. D., & Kimmel, R. (1997). Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, *24*(1), 57–78.

Conte, D., Foggia, P., Percannella, G., Tufano, F., Vento, M. (2009). An algorithm for detection of partially camouflaged people. In *6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 340–345).

Copeland, A.C., Trivedi, M.M. (1997). Models and metrics for signature strength evaluation of camouflaged targets. In *AeroSense* (pp. 194–199).

Cremers, D., Rousson, M., & Deriche, R. (2007). A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, *72*(2), 195–215.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, *1*, 886–893.

Desai, C., Ramanan, D., & Fowlkes, C. C. (2011). Discriminative models for multi-class object layout. *International Journal of Computer Vision*, *95*(1), 1–12.

Di Lascio, R., Foggia, P., Percannella, G., Saggese, A., & Vento, M. (2013). A real time algorithm for people tracking using contextual reasoning. *Computer Vision and Image Understanding*, *117*(8), 892–908.

Du, H., Jin, X., Mao, X. (2012). Digital camouflage images using two-scale decomposition. In *Computer Graphics Forum* (pp. 2203–2212).

Fox, C. W. (1988). *An introduction to the calculus of variations*. New York: Dover Publications Inc.

Freeman, W.T., & Roth, M. (1995). Orientation histograms for hand gesture recognition. In *IEEE International Workshop on Automatic Face and Gesture Recognition* (pp. 296–301).

Gonzalez, R. F., & Woods, R. E. (2008). *Digital image processing*. Singapore: Pearson Education.

Gretzmacher, F.M., Ruppert, G.S., Nyberg, S. (1998). Camouflage assessment considering human perception data. In *Aerospace/Defense Sensing and Controls* (pp. 58–67).

Hao, W., Zhang, B., & Tian, W. (2007). Head tracking by means of probabilistic neural networks. *Measurement Science and Technology*, *18*(7), 1999–2009.

Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, *3*(6), 610–621.

Hare, S., Saffari, A., Torr, P.H. (2011). Struck: Structured output tracking with kernels. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 263–270).

Harville, M., Gordon, G., Woodfill, J. (2001). Foreground segmentation using adaptive mixture models in color and depth. In *Proceedings on IEEE Workshop on Detection and Recognition of Events in Video* (pp. 3–11).

He, L., & Osher, S. (2007). Solving the Chan–Vese model by a multiphase level set algorithm based on the topological derivative. In *Proceedings of the 1st International Conference on Scale Space and Variational Methods in Computer Vision, SSVM* (pp. 777–788).

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 583–596.

Hou, J. Y. Y. H. W., & Li, J. (2011). Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*, *15*, 2201–2205.

Huang, Z.Q., & Jiang, Z. (2005). Tracking camouflaged objects with weighted region consolidation. In *Proceedings on Digital Image Computing: Techniques and Applications (DICTA)* (pp. 24–31).

Jia, X., Lu, H., Yang, M.H. (2012). Visual tracking via adaptive structural local sparse appearance model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1822–1829).

KaewTrakulPong, P., & Bowden, R. (2003). A real time adaptive visual surveillance system for tracking low-resolution colour targets

in dynamically changing scenes. *Image and Vision Computing*, *21*(10), 913–929.

Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1409–1422.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331.

Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., et al. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 319–336.

Krinidis, S., & Chatzis, V. (2009). Fuzzy energy-based active contours. *IEEE Transactons on Image Processing*, *18*(12), 2747–2755.

Krinidis, & S., Krinidis, M. (2012). Fuzzy energy-based active contours exploiting local information. In *8th International Conference on Artificial Intelligence Applications and Innovations (AIAI'12)* (pp. 27–30).

Kusy, M., & Kluska, J. (2013). Probabilistic neural network structure reduction for medical data classification. In *Artificial Intelligence and Soft Computing* (pp. 118–129).

Kwon, J., & Lee, K.M. (2010). Visual tracking decomposition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1269–1276).

Lan, X., Ma, A.J., Yuen, P.C. (2014). Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1194–1201).

Lankton, S., & Tannenbaum, A. (2008). Localizing region-based active contours. *IEEE Transactions on Image Processing*, *17*(11), 2029–2039.

Levi, K., & Weiss, Y. (2004). Learning object detection from a small number of examples: the importance of good features. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 53–60.

Li, C., Kao, C. Y., Gore, J. C., & Ding, Z. (2008). Minimization of region-scalable fitting energy for image segmentation. *IEEE Transactions on Image Processing*, *17*(10), 1940–1949.

Li, C., Xu, C., Gui, C., & Fox, M. D. (2010). Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, *19*(12), 3243–3254.

Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., & Hengel, A. V. D. (2013). A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, *4*(4), 1–58.

Maggio, E., & Cavallaro, A. (2011). *Video tracking: Theory and practice*. West Sussex: Wiley.

Malathi, T., & Bhuyan, K.M. (2013). Foreground object detection under camouflage using multiple camera-based codebooks. In *Annual IEEE India Conference (INDICON)* (pp. 1–6).

Mao, K. Z., Tan, K. C., & Ser, W. (2000). Probabilistic neural-network structure determination for pattern classification. *IEEE Transactions on Neural Networks*, *11*(4), 1009–1016.

Metz, & C.E. (1978). Basic principles of ROC analysis. In *Seminars in nuclear medicine* (Vol. 8, pp. 283–298). Elsevier

Mondal, A., Ghosh, S., & Ghosh, A. (2014). Efficient silhouette-based contour tracking using local information. *Soft Computing*, *20*, 1–21.

Mumford, D., & Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, *42*(5), 577–685.

Musavi, M. T., Chan, K. H., Hummels, D. M., & Kalantri, K. (1994). On the generalization ability of neural network classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(6), 659–663.

Ning, J., Zhang, L., Zhang, D., & Wu, C. (2009). Robust object tracking using joint color-texture histogram. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(07), 1245–1263.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, *29*(1), 51–59.

Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: algorithms based on hamilton–jacobi formulation. *Journal of Computational Physics*, *79*(1), 12–49.

Pan, Y., Birdwell, D.J., Djouadi, S.M. (2006). Efficient implementation of the Chan-Vese models without solving PDEs. In *Proceedings of International Workshop on Multimedia Signal Processing* (pp. 350–353).

Raghu, P., & Yegnanarayana, B. (1998). Supervised texture classification using a probabilistic neural network and constraint satisfaction model. *IEEE Transactions on Neural Networks*, *9*(3), 516–522.

Ross, D. A., Lim, J., Lin, R. S., & Yang, M. H. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, *77*(1–3), 125–141.

Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H. (2010). PROST: Parallel robust online simple tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 723–730).

Sevilla-Lara, L., & Learned-Miller, E. (2012). Distribution fields for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1910–1917).

Shyu, K. K., Pham, V. T., Tran, T. T., & Lee, P. L. (2012). Global and local fuzzy energy-based active contours for image segmentation. *Nonlinear Dynamics*, *67*(2), 1559–1578.

Singh, S. K., Dhawale, C. A., & Misra, S. (2013). Survey of object detection methods in camouflaged image. *IERI Procedia*, *4*, 351–357.

Song, B., & Chan, T. (2002). A fast algorithm for level set based optimization. *CAM-UCLA*, *68*, 02–68.

Specht, D. F. (1990). *Neural networks*, *3*(1), 109–118.

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).

Talu, M. F. (2013). ORACM: Online region-based active contour model. *Expert Systems with Applications*, *40*(16), 6233–6240.

Tan, X., & Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, *19*(6), 1635–1650.

Tang, S., Andriluka, M., & Schiele, B. (2014). Detection and tracking of occluded people. *International Journal of Computer Vision*, *110*(1), 58–69.

Tran, T. T., Pham, V. T., & Shyu, K. K. (2014). Image segmentation using fuzzy energy-based active contour with shape prior. *Journal of Visual Communication and Image Representation*, *25*(7), 1732–1745.

Traven, H. G. (1991). A neural network approach to statistical pattern classification by semiparametric' estimation of probability density functions. *IEEE Transactions on Neural Networks*, *2*(3), 366–377.

Wang, J., & Yagi, Y. (2008). Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Transactions on Image Processing*, *17*(2), 235–240.

Wu, Y., Lim, J., Yang, M.H. (2013). Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2411–2418).

Wu, Y., Ma, W., Gong, M., Li, H., & Jiao, L. (2015). Novel fuzzy active contour model with kernel metric for image segmentation. *Applied Soft Computing*, *34*(2015), 301–311.

Yan, W., Weber, C., & Wermter, S. (2011). A hybrid probabilistic neural model for person tracking based on a ceiling-mounted camera. *Journal of Ambient Intelligence and Smart Environments*, *3*(3), 237–252.

Yang, B., & Nevatia, R. (2014). Multi-target tracking by online learning a CRF model of appearance and motion patterns. *International Journal of Computer Vision*, *107*(2), 203–217.

Yang, F., Lu, H., & Yang, M. H. (2014). Robust superpixel tracking. *IEEE Transactions on Image Processing*, *23*(4), 1639–1651.

Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P., & Tannenbaum, A. (1997). A geometric Snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, *16*(2), 199–209.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, *38*(4), 1264–1291.

Zhang, K., Zhang, L., Song, H., & Zhou, W. (2010). Active contours with selective local or global segmentation: a new formulation and level set method. *Image and Vision Computing*, *28*(4), 668–676.

Zhang, T., Ghanem, B., Liu, S., & Ahuja, N. (2013). Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, *101*(2), 367–383.

Zhang, X., Hu, W., Xie, N., Bao, H., & Maybank, S. (2015). A robust tracking system for low frame rate video. *International Journal of Computer Vision*, *115*, 1–26.

Zhong, M., Coggeshall, D., Ghaneie, E., Pope, T., Rivera, M., Georgiopoulos, M., et al. (2007). Gap-based estimation: Choosing the smoothing parameters for probabilistic and general regression neural networks. *Neural Computation*, *19*(10), 2840–2864.

Zhong, W., Lu, H., & Yang, M. H. (2014). Robust object tracking via sparse collaborative appearance model. *IEEE Transactions on Image Processing*, *23*(5), 2356–2368.