

Research Article

Deformable Contour-Based Maneuvering Flying Vehicle Tracking in Color Video Sequences

Samira Sabouri,¹ Alireza Behrad,² and Hassan Ghassemlian³

¹ Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran 14778-93855, Iran

² Faculty of Engineering, Shahed University, Tehran 18651-33191, Iran

³ School of Electrical and Computer Engineering, Tarbiat Modares University, Tehran 14115-143, Iran

Correspondence should be addressed to Alireza Behrad; behrad@shahed.ac.ir

Received 23 October 2012; Accepted 11 December 2012

Academic Editors: A. Gasteratos, C.-C. Han, D. P. Mukherjee, A. Prati, and J. M. Tavares

Copyright © 2013 Samira Sabouri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new method for the tracking of maneuvering flying vehicles using a deformable contour model in color video sequences. The proposed approach concentrates on targets with maneuvering motion in sky, which involves fundamental aspect change stemmed from 3D rotation of the target or video camera. In order to segment and track the aircraft in a video, at first, the target contour is initialized manually in a key frame, and then it is matched and tracked automatically in the subsequent frames. Generally active contour models employ a set of energy functions based on edge, texture, color, and shape features. Afterwards, objective function is minimized iteratively to track the target contour. In the proposed algorithm, we employ game of life cellular automaton to manage snake pixels' (snaxels') deformation in each epoch of minimization procedure. Furthermore, to cope with the large aspect change of aircraft, a Gaussian model has been taken into account to represent the target color in RGB space. To compensate for changes in luminance and chrominance ingredients of the target, the prior distribution function is dynamically updated during tracking. The proposed algorithm is evaluated using the collected dataset, and the expected probability of tracking error is calculated. Experimental results show positive results for the proposed algorithm.

1. Introduction

The video-based locating and tracking of flying vehicles is an interesting issue in the visual control of aerial systems, which may be employed in aerial surveillance, the navigation of flying robot, missile, microflying, unmanned aircraft, and so forth. In order to localize, track, and recognize flying vehicles, some approaches have been presented recently. In this context, four main state-of-the-art methodologies are well known and applicable including (1) invisible spectrum-based methods like radio detection and ranging (RADAR) or light detection and ranging (LIDAR); (2) visible spectrum-based approaches [1–8] such as existing algorithms in infrared and thermal imaging systems in the wavelength range of 380 nm to 780 nm and even more in far infrared case; (3) global positioning system (GPS-) based methods; and (4) combination of visible and invisible spectrum-based methods. Feasibility of these categories is mostly dependent upon the distance of the imaging system to the target of interest. Furthermore,

each flying vehicle has a set of flight specifications such as the minimum and maximum speed, maneuvering capability, flight board, and so on, whose data may help to estimate the position of the target accurately.

Irrespective of available information, the paper emphasizes on flying vehicle tracking (FVT) in color video sequences. In this domain, we concentrate especially on maneuvering aircraft with large aspect change, which is considered to be a challenging issue. Considering the high deformation of target contour in this application, utilizing deformable surfaces such as active contour [9–11] and active mesh [1, 12] models seems to be a proper choice.

1.1. Related Work. In [1], a method for flying vehicle tracking was introduced in monochrome video sequences. In this method, the incipient location of the target was determined manually; then, the target was tracked automatically by optimizing the mesh energy functions. In [2], a vision-based scheme for automatic locating of a flying vehicle was

presented by means of extracting fuzzified edges features and matching edge pyramids as well as a motion flow vector classifier based on multilayer perceptrons (MLP) neural network. Ha et al. [3] introduced a method for real time tracking of flying targets via the combination of the geometric active contour model and optical flow algorithm. Haker et al. [4] used a segmentation method based on the adaptive thresholding and Bayesian statistic approach to identify and track target location. In [10], an active contour model for vehicle target tracking was utilized. To deal with the high aspect change problem, they handled the motion model around the vehicle to reduce the tracking error in monochrome video frames. Yilmaz [13] suggested an object tracking method based on an asymmetric kernel and mean shift approach in which the scale and the orientation of the kernel were changed adaptively. Molloy and Whelan [14] introduced an active mesh system for the motion tracking of rigid objects to alleviate one of the main problems in active contour models, that is, the snake initialization. Jian and Jie [15] proposed an approach to track small objects in infrared streams using the mean shift estimator algorithm. In [16], a method based on edge features extraction and matching as well as Kalman filter was suggested to detect and track mostly rigid objects like the land vehicles in video sequences grabbed by moving video cameras. Their method determined camcorder motion model by using the planar affine transformation. Lee et al. [17] suggested a deformable contour model based on frames difference map. In [18, 19], an approach for flying vehicle tracking based on the snake model was proposed. They utilized Kalman filter and an energy function the so-called electric fields, to handle large displacement of the target during tracking.

1.2. Motivation. Basically, target rotation around three roll, yaw, and pitch axes results in maneuvering motion of flying vehicle in sky. Additionally, the 3D rotation of aircraft or camera causes aspect change of the target during tracking process. Figure 1 shows aspect change problem in a typical video sequence. In the targets with aspect change some parts of the target appear during the video frames and some other parts disappear. The aspect change may create changes in the luminance and chrominance components of the target. Therefore, traditional optical flow approaches [20] or method based on feature matching [21] and model based approaches [16] fail in tracking targets with large aspect changes. Furthermore, because of maneuvering characteristics of the target, the path of target cannot be determined using algorithms based on estimation or data association [22]. In the paper, we focus on tracking aircraft targets with maneuvering motion and full aspect change and design a new scheme to alleviate general problems in this area.

Among different methods in literature for visual maneuvering aircraft locating and tracking, the approaches such as [10, 16] take into account partial aspect change. However, their efficiency was not proven for target with full aspect change where the target shape and view completely change. In this context, other issues such as different atmospheric conditions, change in luminance and chrominance, image noise, dynamic scene due to the motion of the camera,



FIGURE 1: Video streams of a typical aircraft with aspect change due to maneuvering motion of the target.

dwindling the size of target, and visibility reduction of the scene are challenging. Some of these problems like the change in luminance and chrominance and the size of the target are substantially originated due to the aspect change phenomenon.

To test the algorithm with video files in the mentioned situations and provide a standard and informative dataset, the collected dataset includes videos with various conditions in addition to the aspect change phenomenon.

To cope with these problems, we propose an algorithm based on the active contour model for tracking maneuvering target with full aspect change. The algorithm includes the following structural features.

- (i) A deformable active contour model is designed to track maneuvering aircraft target with full aspect change.
- (ii) A new set of external energy functions are defined in RGB color space to enhance tracking efficiency.
- (iii) Game of life (GoL) cellular automaton is proposed to manage, arrange, and smooth snake pixels (snaxels) deformation in each epoch of minimization.
- (iv) A parametric, multivariate, and unimodal Gaussian model is utilized to dynamically update color distribution of the target of interest in color video frames.

The rest of the paper is organized as follows: Section 2 summarizes the suggested method, and in its subsections, we discuss the deformable contour model, energy minimization and GoL-based contour updating, respectively. Experimental results appear in Section 3, and we conclude the paper in Section 4.

2. Proposed Method

Figure 2 depicts the block scheme of the proposed aircraft tracking algorithm. In the proposed algorithm, firstly the outer contour of the target is localized manually in the initial key frame. Then, a parametric, multivariate, and unimodal Gaussian model is calculated based on the central limit theorem (CLT) to represent target color distribution in RGB space. After contour initialization, the tracking algorithm starts by employing the parameters of the single gaussian model (SGM), that is, mean vector and covariance matrix, to find the location of the target contour in the current frame. Thereafter, the objective energy function is defined and optimized by means of a constrained greedy minimization procedure. To manage snaxels' deformation, we introduce a

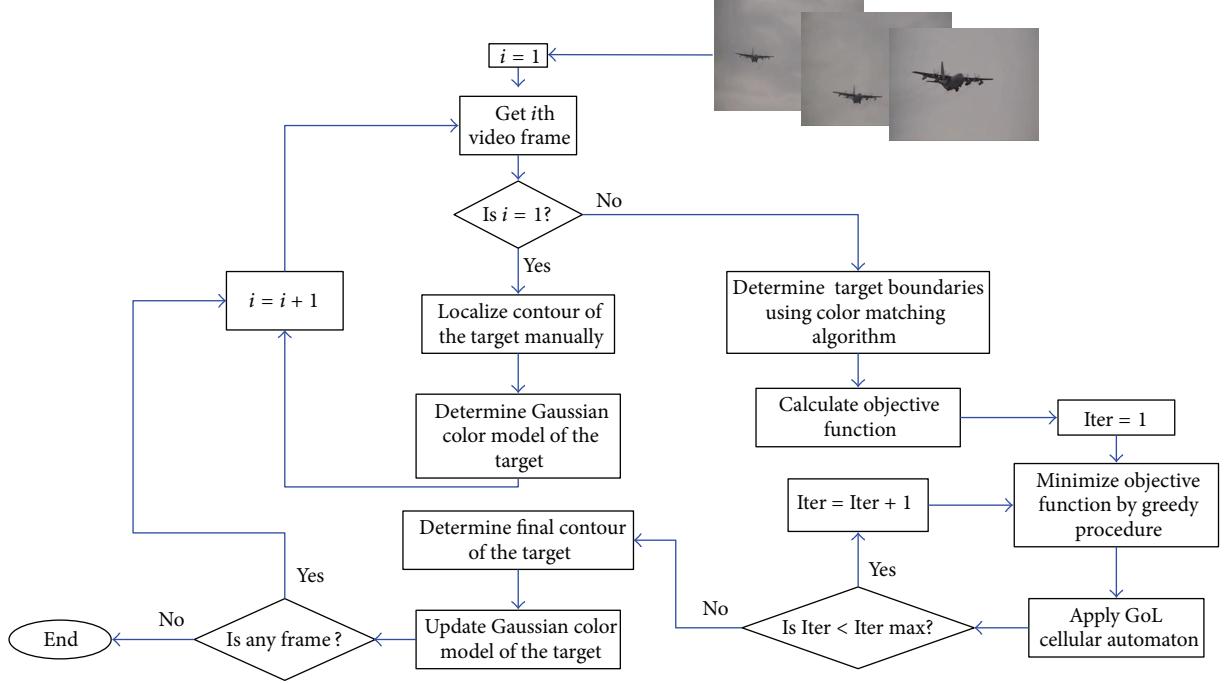


FIGURE 2: Block scheme of the proposed FVT in color video sequences.

GoL cellular automaton which is utilized in each epoch of the optimization routine. In order to deal with the aspect change phenomenon appropriately, the dynamic parametric model is updated after fixing control points and/or reaching the maximum number of iterations in each frame.

2.1. Determining Target Boundaries Using Color Matching Algorithm. In the proposed method, a 2D active contour model is utilized to represent the outer boundary of the target during tracking algorithm. The structure of deformable contour is organized by superposition of predefined energy functions. These functions are minimized using an optimization algorithm.

One of the difficulties in tracking targets with high speed using active contour is the problem of local minimums in the energy minimization algorithm. Therefore, we utilize color distribution of the target to find the initial position of the target. For the first frame of the video, we localize the contour of the target manually. This leads to identify the region of interest (ROI) of the target. Figure 3 shows the ROI of a target and the histograms of color channels.

To determine the initial position of the contour in subsequent frames, a search region (SR) is defined to track the flying vehicle by matching and deforming the contour model. We consider the coordinates of SR as $(x_{\min} - dx, y_{\min} - dy)$ and $(x_{\max} + dx, y_{\max} + dy)$, where

$$\begin{aligned} x_{\min} &= \min_{x \in \text{RoI}} (x), & x_{\max} &= \max_{x \in \text{RoI}} (x), \\ y_{\min} &= \min_{y \in \text{RoI}} (y), & y_{\max} &= \max_{y \in \text{RoI}} (y). \end{aligned} \quad (1)$$

The vector $(dx, dy)^T$ is a confident margin for the target displacement. This margin may be estimated based on the interframe motion of the target. It can be assumed constant, if the speed of aircraft is approximately uniform. In our simulations, we consider the confident margin as $(20, 20)$.

To determine the initial location of the contour in frame t , a parametric, unimodal Gaussian model is considered to represent color distribution of the target in RGB space. The color distribution of the target is represented by a multivariate SGM as

$$p^t(\mathbf{F}^t(\mathbf{x})) = N(\mathbf{F}^t(\mathbf{x}); \boldsymbol{\mu}^{t-1}(\mathbf{x}); \boldsymbol{\Sigma}^{t-1}(\mathbf{x})), \quad (2)$$

where $\mathbf{F}^t(\mathbf{x}) = [F_R^t(\mathbf{x}), F_G^t(\mathbf{x}), F_B^t(\mathbf{x})]^T$ is the color vector, t is the frame index, \mathbf{x} defines pixels coordinates, and the N function represents a multivariate normal distribution as follows:

$$\begin{aligned} N(\mathbf{F}^t(\mathbf{x}); \boldsymbol{\mu}^{t-1}(\mathbf{x}); \boldsymbol{\Sigma}^{t-1}(\mathbf{x})) \\ = c \exp \left(-\frac{1}{2} (\mathbf{F}^t(\mathbf{x}) - \boldsymbol{\mu}^{t-1}(\mathbf{x}))^T \right. \\ \times \left. (\boldsymbol{\Sigma}^{t-1}(\mathbf{x}))^{-1} (\mathbf{F}^t(\mathbf{x}) - \boldsymbol{\mu}^{t-1}(\mathbf{x})) \right), \quad (3) \\ c = \frac{1}{((2\pi)^{d/2} |\boldsymbol{\Sigma}^{t-1}(\mathbf{x})|^{1/2})}. \end{aligned}$$

Here d is the dimension of the color space, which is assumed to be 3 for RGB color space. The mean vector $\boldsymbol{\mu}^{t-1}(\mathbf{x})$ and covariance matrix $\boldsymbol{\Sigma}^{t-1}(\mathbf{x})$ are determined by the color

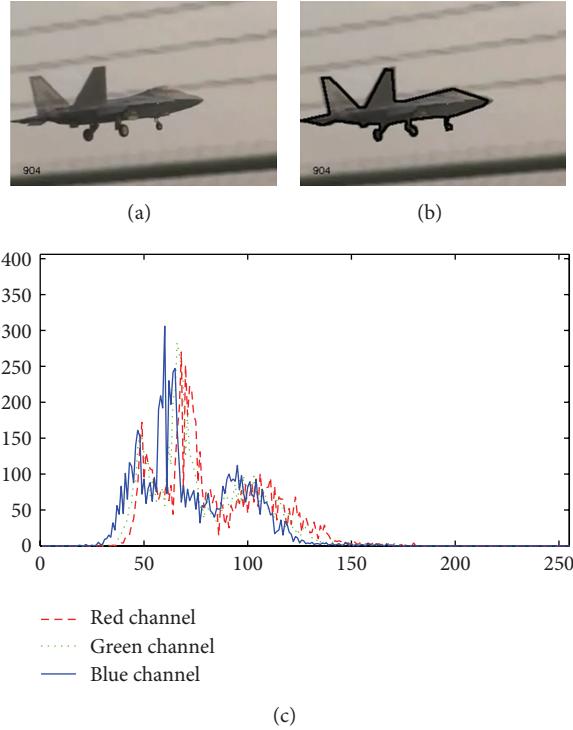


FIGURE 3: Determining initial target contour manually: (a) an initial key frame, (b) target contour determined manually, and (c) histograms of color channels for ROI.

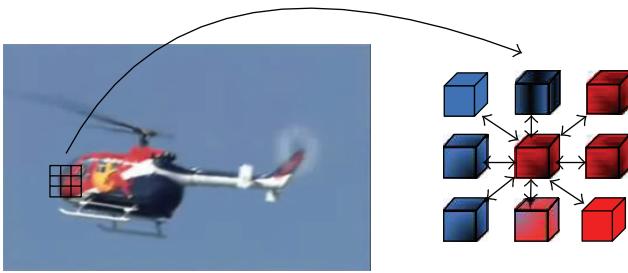


FIGURE 4: Moore 9-neighborhood for a typical snaxel.

information of pixels located inside the contour of the target in the frame $t - 1$ as

$$\begin{aligned} R &= \left[\mathbf{F}_R^{t-1}(\mathbf{x}_1), \dots, \mathbf{F}_R^{t-1}(\mathbf{x}_N) \right]^T, \\ G &= \left[\mathbf{F}_G^{t-1}(\mathbf{x}_1), \dots, \mathbf{F}_G^{t-1}(\mathbf{x}_N) \right]^T, \\ B &= \left[\mathbf{F}_B^{t-1}(\mathbf{x}_1), \dots, \mathbf{F}_B^{t-1}(\mathbf{x}_N) \right]^T, \\ \boldsymbol{\mu}^{t-1}(\mathbf{x}) &= \begin{pmatrix} \mu_R \\ \mu_G \\ \mu_B \end{pmatrix}, \quad \boldsymbol{\Sigma}^{t-1}(\mathbf{x}) = \begin{pmatrix} \sigma_R^2 & \sigma_{RG} & \sigma_{RB} \\ \sigma_{RG} & \sigma_G^2 & \sigma_{GB} \\ \sigma_{RB} & \sigma_{GB} & \sigma_B^2 \end{pmatrix}. \end{aligned} \quad (4)$$

The initial location of the target contour in the current frame is determined using a color matching algorithm. The algorithm includes the following stages.

- (1) Determine the color matching threshold, T_C , by means of prior color distribution as

$$T_C = \frac{\xi}{255} \lceil 255 \max(\lambda^{1/2}) \rceil, \quad \lambda = \text{diag}(\boldsymbol{\Sigma}^{t-1}(\mathbf{x})), \quad (5)$$

where the parameter ξ is a constant number in the range of $[0, 1]$, and the vector λ includes the diagonal elements of matrix $\boldsymbol{\Sigma}^{t-1}(\mathbf{x})$. The notation $\lceil \cdot \rceil$ denotes the ceiling function.

- (2) Determine pixel color consistency for all pixels in SR in the current frame by measuring the Mahalanobis distance:

$$D_M(\mathbf{x}) = \left((\mathbf{F}^t(\mathbf{x}) - \boldsymbol{\mu}^{t-1})^T (\boldsymbol{\Sigma}^{t-1})^{-1} (\mathbf{F}^t(\mathbf{x}) - \boldsymbol{\mu}^{t-1}) \right)^{1/2}, \quad (6)$$

$\mathbf{x} \in \text{SR}$.

- (3) Constitute binary color consistency image I_{CC} , by applying a threshold on $D_M(\mathbf{x})$:

$$I_{CC}(\mathbf{x}) = \begin{cases} 1, & D_M(\mathbf{x}) < T_C, \\ 0, & \text{o.w.} \end{cases} \quad (7)$$

where “1” values show image pixels, which are consistent with target color distribution.

- (4) Suppress small regions in I_{CC} by applying a size-filtering algorithm. In the size filtering, distinct patches that have an area less than the threshold, T_A , are omitted:

$$T_A = \lceil \delta \cdot (h \times w) \rceil, \quad (8)$$

in which the factor δ is equal to 0.01, and the parameters h and w denote the height and width of SR, respectively.

- (5) Extract the outer boundary of consistent pixel in I_{CC} as the initial contour position in the current frame. The final contour of the target is determined using 2D active contour model.

2.2. Active Contour Energy. The active contour model in the search region is considered as a closed contour with n -ordered snaxels, $\mathbf{V} = [\mathbf{v}_0, \dots, \mathbf{v}_{n-1}]$, in which $\mathbf{v}_i = (x_i \ y_i)^T$ denotes the i th control point or snaxel. The contour energy

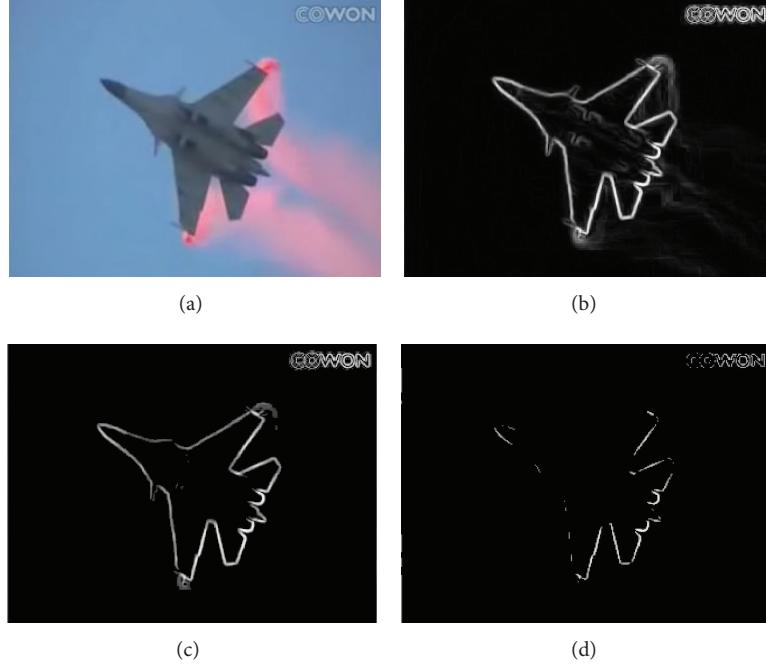


FIGURE 5: Edge energy function for different T_G values. (a) Original frame, (b) edge energy for $T_G = 0$, (c) edge energy for $T_G = 0.2$, and (d) edge energy for $T_G = 0.5$.

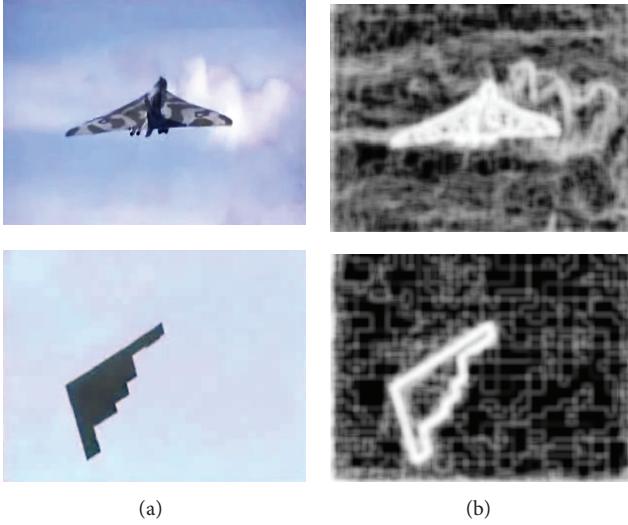


FIGURE 6: Color entropy of sample aircraft images. (a) Original images frames, (b) color entropy images.

in the active contour model is defined as the sum of energy functions for different snaxels of the contour:

$$E_{\text{contour}} = \sum_{i=0}^{n-1} E_{\text{contour}}(\mathbf{v}_i). \quad (9)$$

The snaxel energy consists of two parts including internal and external energies:

$$E_{\text{contour}}(\mathbf{v}_i) = E_{\text{internal}}(\mathbf{v}_i) + E_{\text{external}}(\mathbf{v}_i). \quad (10)$$

The internal energy function defines shape characteristics of the contour and is defined using the following [23]:

$$E_{\text{internal}}(\mathbf{v}_i) = E_{\text{co}}(\mathbf{v}_i) + E_{\text{cu}}(\mathbf{v}_i), \quad (11)$$

where $E_{\text{co}}(\mathbf{v}_i)$ and $E_{\text{cu}}(\mathbf{v}_i)$ are the normalized continuity and curvature energies for the snaxel \mathbf{v}_i , respectively. The continuity and curvature energy functions are defined as

$$\begin{aligned} E_{\text{co}}(\mathbf{v}_i) &= \frac{\tilde{E}_{\text{co}}(\mathbf{v}_i) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{co}}(\mathbf{v}))}{\max_{\mathbf{v} \in M}(\tilde{E}_{\text{co}}(\mathbf{v})) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{co}}(\mathbf{v}))}, \\ \tilde{E}_{\text{co}}(\mathbf{v}_i) &= (\bar{h} - |\mathbf{v}_i - \mathbf{v}_{i-1}|)^2, \\ E_{\text{cu}}(\mathbf{v}_i) &= \frac{\tilde{E}_{\text{cu}}(\mathbf{v}_i) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{cu}}(\mathbf{v}))}{\max_{\mathbf{v} \in M}(\tilde{E}_{\text{cu}}(\mathbf{v})) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{cu}}(\mathbf{v}))}, \\ \tilde{E}_{\text{cu}}(\mathbf{v}_i) &= |(\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1})|^2. \end{aligned} \quad (12)$$

Here, the domain M describes Moore 9-neighborhood as shown in Figure 4, and the parameter \bar{h} is the average distance between two neighboring snaxels. The parameter is updated at the beginning of each epoch of energy minimization routine.

We define the external energy function based on image features such as edge, color, and texture to cope with the problem of large aspect change. Therefore, the external energy function for snaxel \mathbf{v}_i is defined as

$$E_{\text{external}}(\mathbf{v}_i) = E_{\text{ed}}(\mathbf{v}_i) + E_{\text{si}}(\mathbf{v}_i) + E_{\text{te}}(\mathbf{v}_i). \quad (13)$$

The first term of external energy represents the edge energy. The edge energy attracts the target contour toward pixels with

```

(1) For Iter = 1 to Iter_max = 5
(2) For i = 0 to n - 1
    (i) Select ith snaxel of the deformable contour.
    (ii) Calculate energy function for the ith snaxel and its Moore 9-neighborhood.
    (iii) Find location with minimum energy if there is only one local minimum in Moore 9-neighborhood. Otherwise,
          find the point with shortest path. The path is measured based on Euclidian distance.
    End for i.
(3) Move snaxels to new locations.
(4) Apply the proposed GoL cellular automaton as it will be explained in the next subsection.
(5) Update characteristics of deformable contour including coordinates of new control points as well as the number of existing
    snaxels, n.
End for Iter.

```

PSEUDOCODE 1: The pseudo code for energy minimization algorithm and the proposed GoL cellular automaton.

TABLE 1: The probability of tracking error in terms of percent for four different approaches.

Video samples of MAVDB database	The proposed method		Method of [11]		Method of [17]		Method of [18]	
	AA	CM	AA	CM	AA	CM	AA	CM
Video number 2 (Figure 11(f))	6.42	0.49	12	0.71	10	0.45	7.33	0.42
Video number 16 (Figure 14(d))	25.07	0.98	5	0.36	6.5	0.16	6	0.22
Video number 33 (Figure 11(k))	3.17	0.34	12.65	0.92	33.3	1.4	45	1.56
Video number 46 (Figure 11(i))	7.26	0.55	30	1.8	23	0.89	16.4	0.77
Total videos	13.37	0.87	31.14	2.72	23.8	1.75	19.1	1.09

strong edges or pixels with large image gradients. In order to neglect noisy or weak edges, we define edge energy as

$$E_{\text{ed}}(\mathbf{v}_i) = \begin{cases} -\tilde{E}_{\text{gr}}(\mathbf{v}_i), & \tilde{E}_{\text{gr}}(\mathbf{v}_i) > T_G, \\ 0, & \tilde{E}_{\text{gr}}(\mathbf{v}_i) \leq T_G, \end{cases} \quad (14)$$

where $\tilde{E}_{\text{gr}}(\mathbf{v}_i)$ defines the color gradient at the snaxel \mathbf{v}_i and the gradient threshold $T_G = 0.2$ is employed to remove weak edges. To calculate color gradient, different color channels, $\mathbf{F}_L^t(\mathbf{x})$, for all $L \in \{R, G, B\}$, are smoothed using 2D Gaussian kernel, and image derivatives in x and y directions are calculated using Sobel operators. Then, we calculate the parameters of the color gradient as follows [24]:

$$\begin{aligned} g_{xx} &= \left| \frac{\partial \mathbf{F}_R^t(\mathbf{x})}{\partial x} \right|^2 + \left| \frac{\partial \mathbf{F}_G^t(\mathbf{x})}{\partial x} \right|^2 + \left| \frac{\partial \mathbf{F}_B^t(\mathbf{x})}{\partial x} \right|^2, \\ g_{yy} &= \left| \frac{\partial \mathbf{F}_R^t(\mathbf{x})}{\partial y} \right|^2 + \left| \frac{\partial \mathbf{F}_G^t(\mathbf{x})}{\partial y} \right|^2 + \left| \frac{\partial \mathbf{F}_B^t(\mathbf{x})}{\partial y} \right|^2, \\ g_{xy} &= \frac{\partial \mathbf{F}_R^t(\mathbf{x})}{\partial x} \cdot \frac{\partial \mathbf{F}_R^t(\mathbf{x})}{\partial y} + \frac{\partial \mathbf{F}_G^t(\mathbf{x})}{\partial x} \cdot \frac{\partial \mathbf{F}_G^t(\mathbf{x})}{\partial y} \\ &\quad + \frac{\partial \mathbf{F}_B^t(\mathbf{x})}{\partial x} \cdot \frac{\partial \mathbf{F}_B^t(\mathbf{x})}{\partial y}, \end{aligned} \quad (15)$$

$$\phi_1 = \frac{1}{2} \tan^{-1} \left(\frac{2g_{xy}}{g_{xx} - g_{yy}} \right), \quad \phi_2 = \phi_1 + \frac{\pi}{2},$$

where $\partial \mathbf{F}_L^t(\mathbf{x})/\partial x$ and $\partial \mathbf{F}_L^t(\mathbf{x})/\partial y$, for all $L \in \{R, G, B\}$ define the derivatives of color components and φ_k , for all $k \in \{1, 2\}$ represents gradients directions.

We define normalized gradient energy, $\tilde{E}_{\text{gr}}(\mathbf{v}_i)$, as

$$\begin{aligned} E'_k &= (0.5((g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos(2\phi_k) \\ &\quad + 2g_{xy} \sin(2\phi_k))), \quad \forall k \in \{1, 2\}, \\ E'_{\text{gr}} &= \max \{E'_1, E'_2\}, \end{aligned} \quad (16)$$

$$\tilde{E}_{\text{gr}}(\mathbf{v}_i) = \frac{E'_{\text{gr}}(\mathbf{v}_i) - \min_{\mathbf{v} \in M}(E'_{\text{gr}}(\mathbf{v}))}{\max_{\mathbf{v} \in M}(E'_{\text{gr}}(\mathbf{v})) - \min_{\mathbf{v} \in M}(E'_{\text{gr}}(\mathbf{v}))}.$$

Figure 5 illustrates the results of the calculated edge energy for different T_G values.

The second term of the external energy is the similarity energy, E_{si} . We define the similarity energy to attract a snaxel toward an image location with the same color distribution of the snaxel in the previous frame. To measure the similarity, the Mahalanobis distance in RGB color space is employed. We utilize the snaxel \mathbf{v}_i and its Moore 9-neighborhood to calculate the color distribution of the snaxel. For the control point \mathbf{v}_i , the normalized similarity energy is obtained as

$$E_{\text{si}}(\mathbf{v}_i) = \frac{\tilde{E}_{\text{si}}(\mathbf{v}_i) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{si}}(\mathbf{v}))}{\max_{\mathbf{v} \in M}(\tilde{E}_{\text{si}}(\mathbf{v})) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{si}}(\mathbf{v}))}, \quad (17)$$

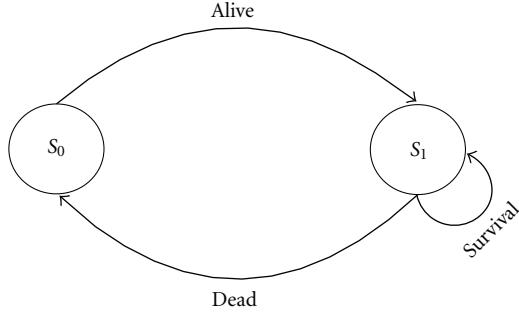


FIGURE 7: The state diagram of the proposed GoL cellular automaton.

where $\tilde{E}_{\text{si}}(\mathbf{v}_i)$ is defined as

$$\tilde{E}_{\text{si}}(\mathbf{v}_i) = \left((\mathbf{F}^t(\mathbf{v}_i) - \boldsymbol{\mu}^{t-1})^T (\boldsymbol{\Sigma}^{t-1})^{-1} (\mathbf{F}^t(\mathbf{v}_i) - \boldsymbol{\mu}^{t-1}) \right)^{1/2}. \quad (18)$$

The third term of external energy, which is called texture energy, employs image texture to define energy function. This energy is based on the entropy of color channels. The entropy is a method to measure the content of information. Pixels in the target boundary have higher information content or entropy in comparison with pixels located inside the target or sky background in SR. Therefore, by utilizing the texture energy we aim at attracting contour snaxels toward points with higher information content.

To measure texture energy, we first calculate the entropy for each individual color channel using the following equation:

$$\mathbf{H}_L = \sum_{i=0}^{k-1} P_L(I_L^i) \log_2 \left(\frac{1}{P_L(I_L^i)} \right), \quad \forall L \in \{R, G, B\}, \quad (19)$$

where I_L^i is a random variable representing i th intensity level in color channel L , and P_L denotes its probability mass function for color channel L which is determined using image histogram in Moore 9- neighborhood.

The normalized texture energy based on the entropy of color channels, for the snaxel \mathbf{v}_i , is defined as

$$E_{\text{te}}(\mathbf{v}_i) = \frac{\tilde{E}_{\text{te}}(\mathbf{v}_i) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{te}}(\mathbf{v}))}{\max_{\mathbf{v} \in M}(\tilde{E}_{\text{te}}(\mathbf{v})) - \min_{\mathbf{v} \in M}(\tilde{E}_{\text{te}}(\mathbf{v}))}, \quad (20)$$

where color entropy image $\tilde{E}_{\text{te}}(\mathbf{v}_i)$ is obtained as

$$\tilde{E}_{\text{te}}(\mathbf{v}_i) = -(\mathbf{H}_R(\mathbf{v}_i) + \mathbf{H}_G(\mathbf{v}_i) + \mathbf{H}_B(\mathbf{v}_i)). \quad (21)$$

Here $\mathbf{H}_R(\mathbf{v}_i)$, $\mathbf{H}_G(\mathbf{v}_i)$, and $\mathbf{H}_B(\mathbf{v}_i)$ denote the entropy of R , G , and B color channels, respectively. The minus sign in (21) is used to minimize the texture energy in the areas with high information content.

Figure 6 illustrates two typical aircraft images and their color entropy images. As it is shown in the figure, the target area and its boundary reveal higher entropy values.

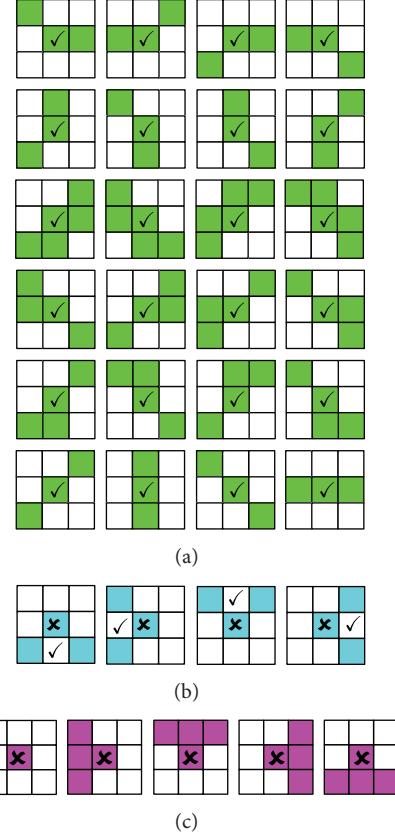


FIGURE 8: Defined structures and rules for the proposed GOL. (a) Continuance of life structures, (b) death and birth rules, and (c) death structures. Symbols \checkmark and \times represent life and death at the next generation.

2.3. Energy Minimization. After defining energy function for the active contour, the objective energy function defined in (9) is minimized iteratively by means of a constrained greedy optimization routine in order to fit the deformable contour on the target precisely. Here, the constrained term denotes a set of conditions, which are applied during the minimization algorithm. The pseudocode of energy minimization procedure is given in Pseudocode 1 in which we apply the proposed GoL cellular automaton after each epoch of energy minimization algorithm.

2.4. Game-of-Life-Based Snaxel Updating. Both target and camera motion like 3D rotation and translation and zooming by the video camera result in changing shape and outer boundary of the aircraft. It is obvious that dynamic motion of snaxels toward the outer boundary of aircraft due to energy minimization routine causes some irregular deformation in the contour. Consequently, the number of control points should be changed. To regularize and smooth snaxels of the contour after energy minimization, we propose a GoL cellular automaton.

GoL cellular automata are substantially a type of two dimensional cellular automata [25]. Fundamentally,

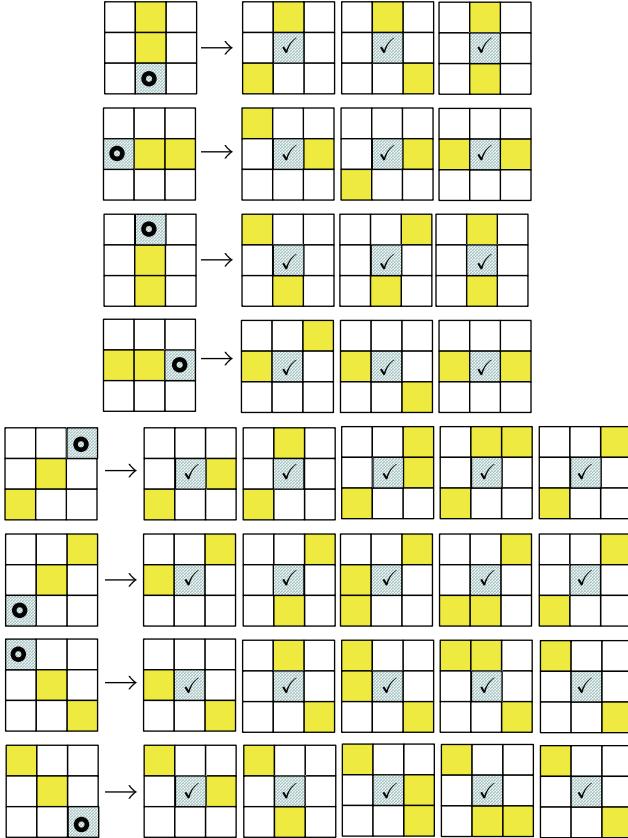


FIGURE 9: The rule of birth in the proposed GoL cellular automaton. The symbol \circ denotes cell that should be checked independently.

automata are simple agents with three main features including uniformity, synchronicity, and locality. Different cellular automata follow special rules to gain a distinct goal. Here, we design a set of rules for GoL to handle changes in snaxels of active contour model. According to the defined rules in the proposed automaton, some of snaxels, that is, alive cells, may be dead or remain without any change during evolution. It is also possible to create some new snaxels in the next generation. In the proposed cellular automaton, the neighborhood radius, r , is set to 1 with the 8-cell Moore neighborhood, and the number of generations, g , is considered to be 7. We consider two states, $N_Q = 2$, with the set of states $Q = \{S_0, S_1\}$, in which the states S_0 and S_1 represent dead and alive cells, respectively. Figure 7 shows the state diagram of the proposed cellular automaton. In the proposed GoL cellular automaton, the next state of a snaxel is determined based on the following rules.

- (i) The rule of continuance of life: a live central cell will survive if its Moore 8-neighborhood is matched with one of the states in Figure 8(a).
- (ii) The rule of death and birth: a live central cell dies, and its adjacent dead cell that is located between two alive cells becomes alive at the next generation when a correspondence to one of the states in Figure 8(b) is found.

(iii) The rule of death: a live central cell becomes dead if its Moore 8-neighborhood is matched with one of the states in Figure 8(c).

(iv) The rule of birth: a boundary dead cell with a live central cell that itself has one alive cell in its Moore 8-neighborhood will be checked for alive state. In this case, we check Moore 8-neighborhood of the dead cell; in the case of a match with one of the structures in Figure 9, the central dead cell is marked as alive in the next generation.

It is important to note that the total number of possible structure, N_s , for $N_Q = 2$, $r = 1$, and the Moore 9-neighborhood is determined as

$$N_s = \sum_{i=0}^9 \binom{9}{i} \equiv N_Q^{(2r+1)^2} = 512. \quad (22)$$

In 2D active contour domain, it is not necessary to consider all the structure. Therefore, we define rules for only necessary structures.

Figure 10 shows simulation results for five initial states up to 6 generations. The results demonstrate that after maximum 4 generations, snaxels are stable. The cellular automaton in the first row of Figure 10 represents an ideal case for a contour whose shape is preserved during evolution; whereas other cellular automata have some irregularity and discontinuity because of the energy minimization algorithm. These contours are smoothed and regularized using the proposed GoL cellular automaton.

3. Experimental Results

The proposed FVT algorithm was implemented using a MATLAB program and tested using a Pentium-IV desktop Personal Computer (PC) with 2.8 GHz CPU and 512 MB RAM. For maneuvering aircraft tracking purposes, we provided an informative dataset called Maneuvering Aircraft Video DataBase (MAVDB). The database includes 72 video clips captured by diverse monocular, moving CCD camcorders. The dataset is freely available for academic application. Time duration of different videos in MAVDB varies from 1.2 s to 51.64 s with the frame rate of 15 Hz or 30 Hz. MAVDB include maneuvering targets in different conditions such as large aspect change, 3D rotation around different axis, targets with change in size, rigid or piecewise rigid flying vehicles, smoky aircrafts, cloudy atmospheric conditions, and illumination change.

Due to the presence of a relative similarity between noise and rain drops and/or ice crystals, we added Gaussian as well as salt and pepper noises to some video streams to simulate different atmospheric conditions especially rainy and snowy situations.

In Figure 11, the results of the proposed FVT algorithm on eleven sample video clips of MAVDB are shown.

Figure 12 depicts results of the proposed method on three noisy video clips. In Figures 12(a) and 12(b), we test our FVT under additive white Gaussian noise (AWGN) with normal distributions of $N(0, 0.01)$ and $N(0, 0.001)$,

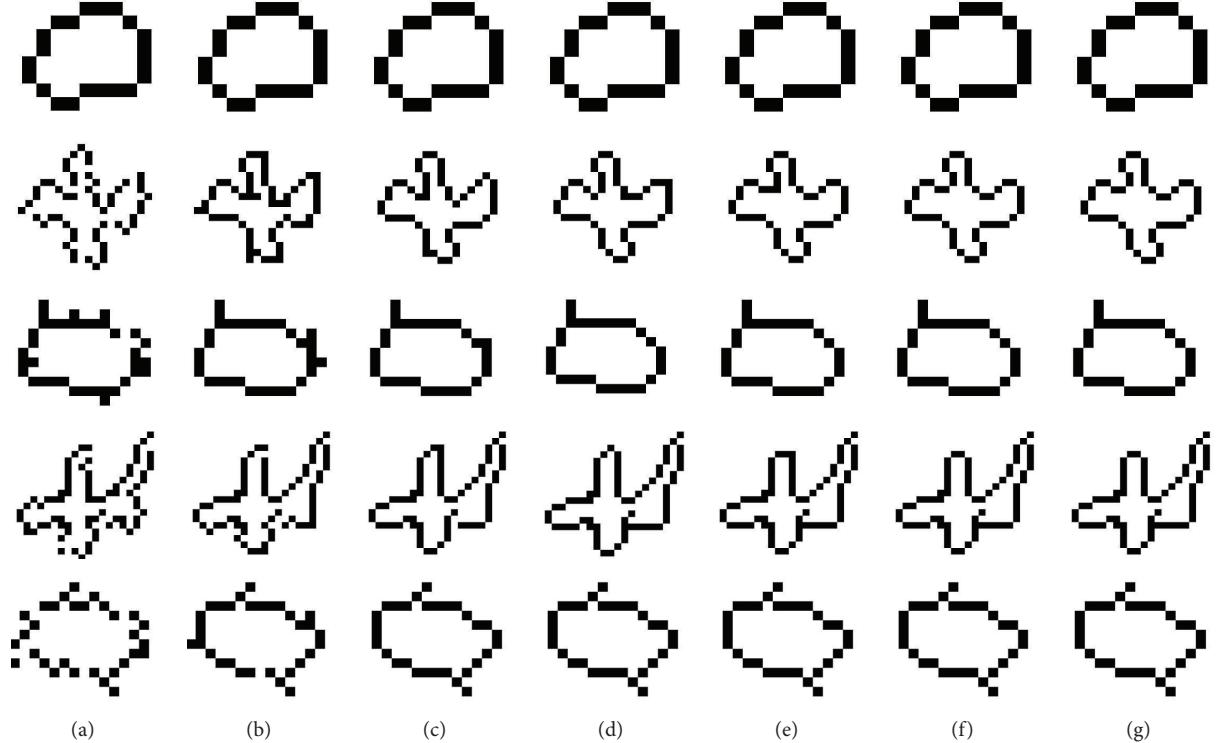


FIGURE 10: Simulation results of applying the proposed GoL cellular automaton on five initial states, (a) initial state which is obtained from energy minimization step, (b) the result of the first generation, (c) the result of the second generation, (d) the result of the third generation, (e) the result of the fourth generation, (f) the result of the fifth generation, and (g) the result of the sixth generation.

respectively. Figure 12(c) shows the results of the proposed tracking algorithm for video sequence with 12.5% additive salt and pepper noise (ASPN). The video sequence of Figure 12(a) shows a smoky fighter jet with maneuvering motion, and Figure 12(b) is a synthetic airplane video with large aspect change. In Figure 13, we have also plotted the trajectory for the true centroid (center of gravity (CoG)) and the tracked CoG of the target in Figure 12(a). These results demonstrate the robustness of our method in noisy video sequences.

Figure 14 depicts the results of the proposed FVT approach on four videos of MAVDB that seem to be more challenging than other videos. In the video sequences of Figure 14(a), a relatively abrupt change in statistical characteristics of color distribution exists. Figure 14(b) shows video sequence of a helicopter with two dominant red and black colors. In video sequence of Figure 14(c), the direct sunlight changes and saturates the intensity values for some parts of the target. The background of the video is also changed to be cloudy in some frames. Figure 14(d) shows the video sequence of an aircraft with textured surface. Obviously, in such situations, the probability of the tracking error increases and target contour fluctuates during tracking process. However instabilities are controlled and compensated because of the dynamic structure of the proposed tracking method as it is shown in Figure 14(d).

In order to measure the performance of the proposed FVT algorithm, we have measured the expected probability of the tracking error using two different methods based on

pixel-based performance evaluation (PE) algorithm. For this purpose, two criteria are considered and measured including alignment amount (AA) and confusion matrix (CM).

In order to measure the performance based on the AA criterion, we have defined the probability of error for a video frame as

$$P(\text{error}) = 1 - \frac{\bigcap_{i=1}^2 \mathbf{F}_i}{\bigcup_{i=1}^2 \mathbf{F}_i}, \quad (23)$$

where \mathbf{F}_1 and \mathbf{F}_2 are ground truth (GT) and predicted result (PR) frames, respectively. Therefore, in a video clip containing N_f frames, the mean error is calculated as follows:

$$\mu_e = E\{P(\text{error})\} = 1 - \frac{1}{N_f} \sum_{n=1}^{N_f} \frac{P(\text{GT}_n, \text{PR}_n)}{P(\text{GT}_n) + P(\text{PR}_n) - P(\text{GT}_n, \text{PR}_n)}, \quad (24)$$

where the operator $E\{\cdot\}$ denotes the mathematical expectation.

The second performance evaluation method is based on a 2×2 confusion matrix, \mathbf{C} . To evaluate performance in a frame in this method, we construct the confusion matrix as follows:

$$\mathbf{C} = \begin{pmatrix} P(\text{OB} | \text{OB}) & P(\text{NO} | \text{OB}) \\ P(\text{OB} | \text{NO}) & P(\text{NO} | \text{NO}) \end{pmatrix} = \begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix}, \quad (25)$$

where the notations OB and NO stand for object and non-object, respectively, and the elements TP , FP , FN , and TN



FIGURE 11: Continued.

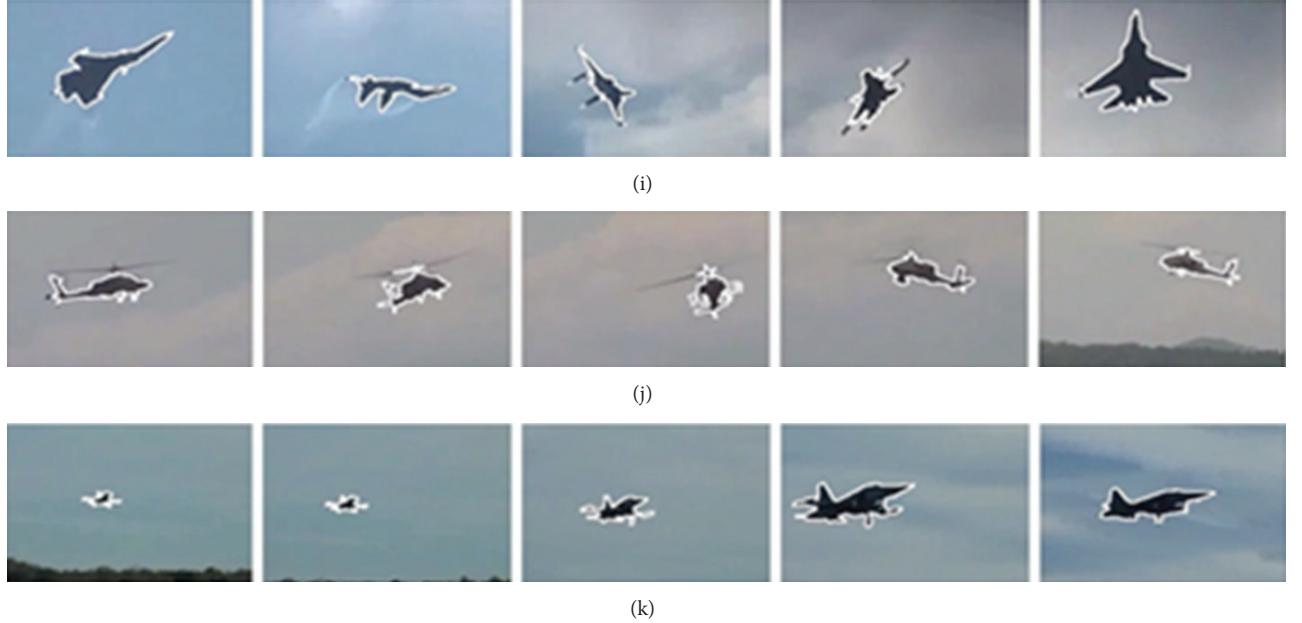
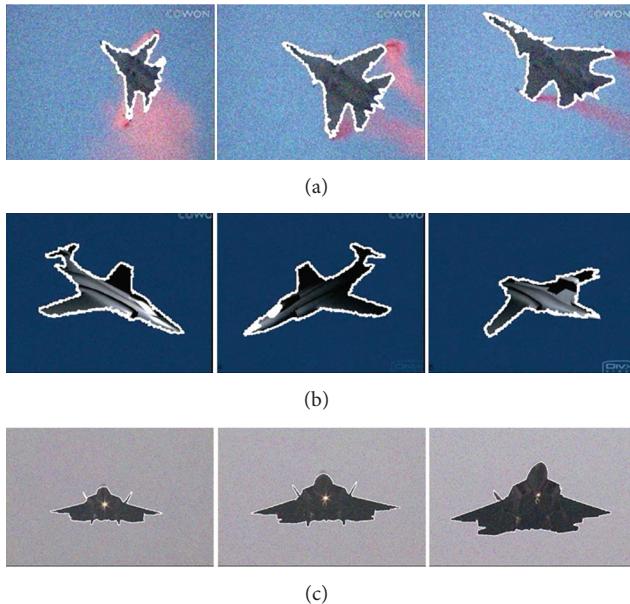


FIGURE 11: The results of the proposed FVT algorithm on eleven sample video clips.

FIGURE 12: The efficiency of the proposed algorithm for noisy video sequences, (a) video clip with AWGN of $N(0, 0.01)$, (b) video clip with AWGN of $N(0, 0.001)$, and (c) video clip with ASPN and the density of 12.5%.

stand for true positive, false positive, false negative and true negative, respectively. In performance evaluation based on the confusion matrix, the expected probability of error for a video clip including N_f frames is determined as

$$\mu_e = E \{P(\text{error})\} = 1 - \frac{\sum_{i=1}^2 \mathbf{C}_t(i, i)}{\sum_{i=1}^2 \sum_{j=1}^2 \mathbf{C}_t(i, j)}, \quad (26)$$

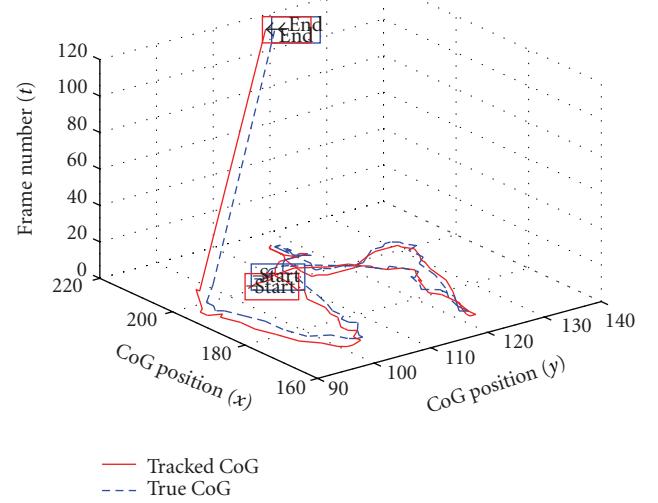


FIGURE 13: The trajectory for the true and tracked centroid of the video clip in Figure 12(a).

where the total confusion matrix, \mathbf{C}_t , is calculated by using the following equation:

$$\mathbf{C}_t = \frac{1}{N_f \cdot (h_v \times w_v)} \cdot \begin{pmatrix} \sum_{n=1}^{N_f} \text{TP}_n & \sum_{n=1}^{N_f} \text{FN}_n \\ \sum_{n=1}^{N_f} \text{FP}_n & \sum_{n=1}^{N_f} \text{TN}_n \end{pmatrix}. \quad (27)$$

In the above equation h_v and w_v are the height and width of the video frames.

Figure 15 exemplifies the method to determine GT and PR in a typical video frame in order to measure the performance

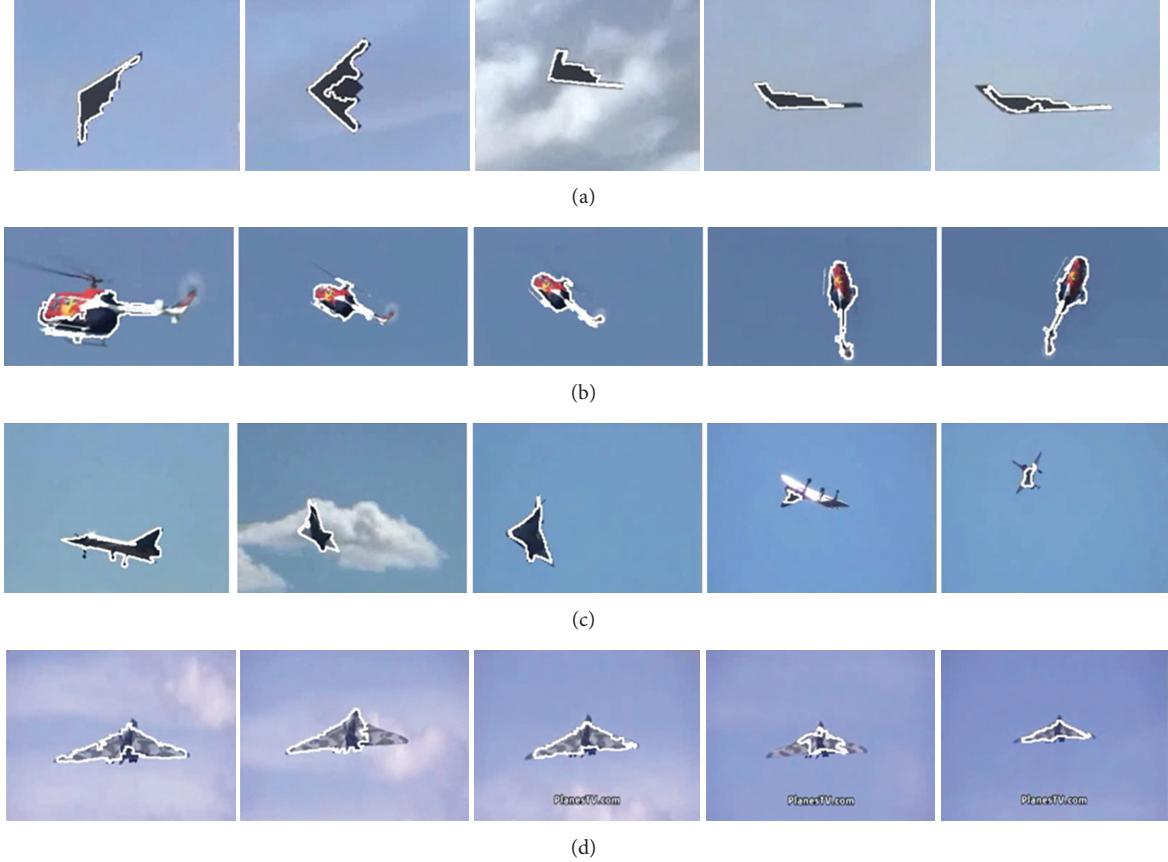


FIGURE 14: The results of the proposed FVT on four challenging video streams. (a) Video with relatively abrupt change in statistical characteristics of color distribution, (b) video sequence of a helicopter with two dominant red and black colors, (c) video with the direct sunlight which changes and saturates the intensity values for some parts of the target, and (d) video sequences of an aircraft with textured surface.

for both AA and CM methods. Considering high volume of frames in database, we calculate GT and PR with the interval of 10 frames in our experiments. Figure 16 illustrates the probability of error in terms of frame number for video streams shown in Figure 14(d) for $\xi = 0.13$. As it is shown in this figure, both curves show that the probability of error is not accumulative and is controlled during the tracking process. Figure 17 shows the expected probability of error in terms of ξ for video sequences shown in Figure 14(d).

To compare the results of the proposed algorithm with those of other methods, we also implemented the active contour method by Williams and Shah [11], parametric active contour method by Lee et al. [17], and the method presented by Torkan and Behrad [18, 19]. We implemented and tested different methods using the collected MAVDB dataset. Table 1 shows the expected probability of tracking error for different implemented algorithms using AA and CM methods. Table 1 shows that the expected probability of errors for the proposed algorithm using all videos in database are 13.37% and 0.87%, based on AA and CM methods, respectively. The table shows that the proposed algorithm has enhanced the accuracy up to 17.77% and 1.85% based on AA and CM methods, respectively.

4. Conclusions

In this paper, we presented a new algorithm for visual maneuvering aircraft tracking in sky based on the deformable contour model in color video sequences. The main aim of the algorithm is to cope with the maneuvering motion and large aspect change of targets stemmed from 3-D rotation of targets in sky or camera. Furthermore, we examined other challenging issues like noisy streams, different atmospheric conditions, and smoky aircraft. To test the proposed algorithm, we collected different video clips in a dataset called MAVDB. We implemented some existing method and tested the algorithm with the collected dataset. To evaluate performance of the proposed method, we measured the probability of tracking error based on two AA and CM criteria. The experimental results demonstrated that the proposed algorithm has reduced the probability of tracking error considerably. In error analysis for the proposed method, we perceived that, during tracking, most of the errors originated from changing the texture of target and its background. Fortunately, the dynamic update strategy in the proposed tracking method

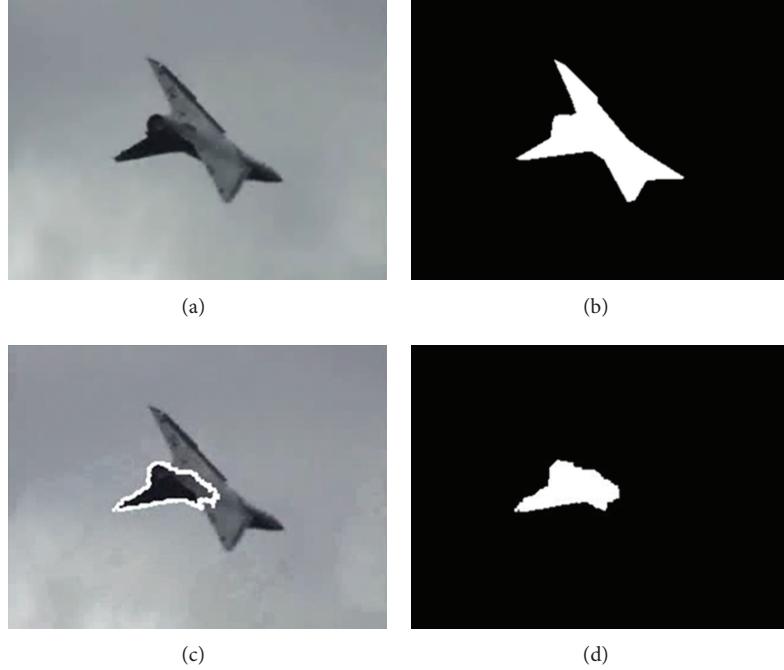


FIGURE 15: Calculation of PR and GT images. (a) Original frame, (b) GT of the original frame, (c) the contour obtained using the proposed FVT algorithm, and (d) the calculated PR image.

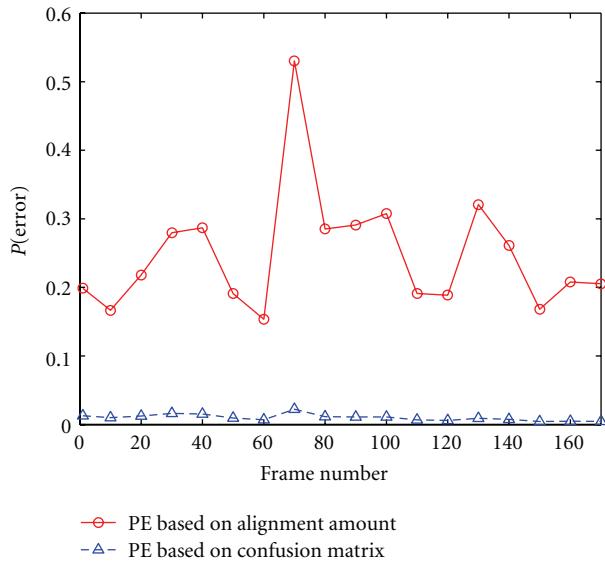


FIGURE 16: The plot of the probability of error in terms of frame number for video streams shown in Figure 14(d).

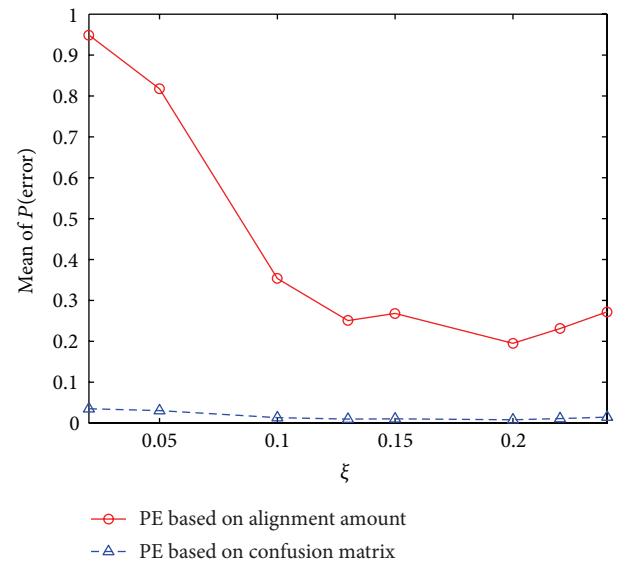


FIGURE 17: The plot of the expected probability of error in terms of ξ for video sequences shown in Figure 14(d).

controls errors and prevent error accumulation as shown in Figure 14. For future works, we are going to focus on some other challenging issues such as occlusion, complex, and cluttered background, which may occur during take-off and landing.

Acknowledgment

The authors would like to sincerely thank A. Taimori, Ph.D. candidate of System Communication in the Tehran Science and Research Branch, Islamic Azad University, Iran, for his valuable suggestions that have helped the to improve the quality of this paper.

References

- [1] A. Taimori and S. Sabouri, "Vision based flying vehicle tracking," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 330–335, Singapore, October 2008.
- [2] A. Taimori, A. Behrad, and S. Sabouri, "A new scheme for vision based flying vehicle detection using motion flow vectors classification," in *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA '09)*, pp. 175–180, Pisa, Italy, December 2009.
- [3] J. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active contours and optical flow for automatic tracking of flying vehicles," in *Proceedings of the American Control Conference (AAC '04)*, pp. 3441–3446, July 2004.
- [4] S. Haker, G. Sapiro, A. Tannenbaum, and D. Washburn, "Missile tracking using knowledge-based adaptive thresholding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '01)*, pp. 786–789, October 2001.
- [5] J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target localization from a fixed-wing miniature air vehicle," in *Proceedings of the American Control Conference (AAC '06)*, pp. 2862–2867, June 2006.
- [6] J. J. Kehoe, R. S. Causey, A. Arvai, and R. Lind, "Partial aircraft state estimation from optical flow using non-model-based optimization," in *Proceedings of the American Control Conference (AAC '06)*, pp. 2868–2873, June 2006.
- [7] Y. Jianchao, "A new scheme of vision based navigation for flying vehicles—concept study and experiment evaluation," in *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision (ICARC '02)*, pp. 643–648, December 2002.
- [8] A. Betser, P. Vela, and A. Tannenbaum, "Automatic tracking of flying vehicles using geodesic snakes and kalman filtering," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 1649–1654, December 2004.
- [9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1989.
- [10] B. Jamasbi, S. A. Motamedi, and A. Behrad, "Contour tracking of targets with large aspect change," *Journal of Multimedia*, vol. 2, no. 6, pp. 7–14, 2007.
- [11] D. J. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP: Image Understanding*, vol. 55, no. 1, pp. 14–26, 1992.
- [12] Y. Wang and O. Lee, "Active mesh—a feature seeking and tracking image sequence representation scheme," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 610–624, 1994.
- [13] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, Minneapolis, Minn, USA, June 2007.
- [14] D. Molloy and P. F. Whelan, "Active-Meshes," *Pattern Recognition Letters*, vol. 21, pp. 1071–1080, 2000.
- [15] C. Jian and Y. Jie, "Real-time infrared object tracking based on mean shift," in *Proceedings of the 9th Iberoamerican Congress on Pattern Recognition (CIARP '04)*, pp. 45–52, Puebla, Mexico, 2004.
- [16] A. Behrad and S. Ahmad Motamedi, "Moving target detection and tracking using edge features detection and matching," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 12, pp. 2764–2774, 2003.
- [17] B. H. Lee, I. Choi, and G. J. Jeon, "Motion-based boundary tracking of moving object using parametric active contour model," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 1, pp. 355–363, 2007.
- [18] S. Torkan and A. Behrad, "A new contour based tracking algorithm using improved greedy snake," in *18th Iranian Conference on Electrical Engineering (ICEE '10)*, pp. 150–155, Isfahan, Iran, May 2010.
- [19] S. Torkan, *Aerial target tracking using energy minimization methods [M.S. thesis]*, Malek-e-Ashtar University of Technology, 2009.
- [20] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. v, pp. 490–498, 1992.
- [21] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly Media, 2008.
- [22] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.
- [23] E. Trucco and A. Verri, *Introductory Techniques For 3-D Computer Vision*, Prentice Hall, New York, NY, USA, 1998.
- [24] S. Dizenzo, "A Note on the Gradient of a Multi-Image," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 116–125, 1986.
- [25] J. L. Schiff, *Cellular Automata a Discrete View of the World*, John Wiley & Sons, New York, NY, USA, 2008.

