2009 Ninth International Conference on Intelligent Systems Design and Applications

A New Scheme for Vision Based Flying Vehicle Detection Using Motion Flow Vectors Classification

Ali Taimori
Electrical Eng. Dept., Tech. and Eng. Faculty, Shahed University
Tehran, Iran
taimori@shahed.ac.ir

Alireza Behrad
Electrical Eng. Dept., Tech. and Eng. Faculty, Shahed University
Tehran, Iran
behrad@shahed.ac.ir

Samira Sabouri
Tech. and Eng. Faculty, S. and R. Branch, Islamic Azad University
Tehran, Iran
sabouri.s@srbiau.ac.ir

***Abstract*—This paper presents a vision based scheme for detecting flying vehicle using a new feature extraction and correspondence algorithm as well as a motion flow vectors classifier. The base of detection is to classify the motion flow vectors of object and scene at two video sequences from a mobile monocular CCD camera. For this purpose, we introduce a method to extract robust features from fuzzified edges at first frame. Then, correspondence features are approximated at second video frame by a multi resolution feature matching processing based on edge Gaussian pyramids. In next stage, the estimated motion flow vectors classify into two object and scene classes using a supervised machine learning method based on MLPs neural network. In final step, the flying vehicle localize by approximating the contour of object based on a convex hull algorithm. Experimental results demonstrate that the proposed method has proper stability and reliability especially for the detection of aerial vehicle in applications with mobile camera.**

Keywords-feature extraction and correspondence; flying vehicle detection; fuzzy sets theory; MLPs neural network; optical flow .

I. INTRODUCTION

The video based detection of flying vehicles is an interest topic in the vision based control of aerial systems which may be utilized in aerial bounds surveillance; flying vehicle tracking; electronic warfare; the navigation of flying robot, missile, micro flying, and unmanned aircrafts etc. For detection, tracking and/or recognition of flying vehicles, some methods have presented up to now. In this context, invisible spectrum based methods like RAdio Detection And Ranging (RADAR) and LIght Detection And Ranging (LIDAR), visible spectrum (with the wave-length of range 380 to 780 nm) based algorithms such as thermal and infrared imaging camera systems, Global Positioning System (GPS) based methods, and the fusion of visible and invisible methods are practical. These methods may be utilized dependent upon the amount of distance to target.

In this paper, we have focused on visible based methods [1-6] and have implemented a vision based scheme for flying vehicle detection. In [1], we presented a flying vehicle tracking method in which the incipient position of target determines manually; then the target tracks by optimizing the mesh energy functions at video sequences. Redding et al. [2] suggested a vision based method for flying vehicle detection. They utilized the pixel information of target, the measuring both of position and altitude of Unmanned Aerial Vehicle (UAV) as well as angle of camera position to obtain the global coordinate of target and regions around it. Kehoe et al. [3] introduced a monocular vision system for estimating the state of aircraft in which the dense optical flow of tracked feature points has utilized for estimating angular rates and wind-axis angels. Jianchao [4] suggested a method for flying vehicle navigation using a video camera and altimeter. For approximating the 3-D motion of camera in that algorithm, geometrical transformation between two consecutive frames has utilized. Betser et al. [5] used the active contour and extended Kalman filter for the contour extraction and tracking of flying vehicles. The mentioned research, in fact, has employed local vision information to control the flying vehicles in a feedback loop. Ha et al. [6] introduced an approach for real time tracking of the flying targets via the combination of geometric active contour model and optical flow.

In the above methods, features information and their correspondences in image are utilized generally for flying vehicle localization and tracking. Therefore, problems like different atmospheric conditions, illumination change, outlier data, dynamic scene due to the movement of camera and so forth may cause injurious effects on feature extraction and correspondence algorithms. In this paper, we present a motion based method to detect the flying vehicle in the visible spectrum in which have tried to address some of the mentioned problems by developing new methodologies in features extraction, correspondence and classification domain. The proposed method can also utilize in image acquisition systems with the both low and high altitude. The main contributions of the paper summarize as:

- Proposing a robust and illumination insensitive feature extractor from fuzzified edges that has so-called Good Edge Features To Detect (GEFTD).
- Introducing a Multi Resolution Feature Matching (MRFM) processing to find correspondence points

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.147

using edge Gaussian pyramids from the images with fuzzified edges.

- Localizing flying vehicle by a MLPs neural network based flow vectors classifier.

The reminder of the paper is organized as follows: section II introduces our flying vehicle detection scheme. In its subsection, we describe feature extraction, multi resolution feature matching, and motion flow vectors classifier. Section III depicts empirical results that include the intermediate and final results of the proposed algorithm. Finally, we conclude the paper in section IV.

II. PROPOSED METHOD

As mentioned in section I, the proposed method is a vision based approach in which accurate optical flows are estimated at first by means of a new feature extraction and correspondence algorithm; then flow vectors classify by a MLPs neural network to detect the flying vehicle. In the algorithm, we first present a robust edge feature extractor algorithm to find interest points at the first video sequence, t, based on fuzzy sets theory [7]. At subsequent video frame, $t+dt$, correspondence points are approximated using a multi resolution processing with sub-pixel accuracy based on edge Gaussian pyramids in which is $dt \geq 1$ and its amount obtains dependent upon the frame rate of video camera and the movement speed of target and camera. It is important to note that the edge Gaussian pyramids determine by down-sampling the edge contours not total image. Then, the estimated motion flow vectors classify into two object and scene classes via a MLPs neural network. In order to determine the contour of target, we have fitted a convex hull on correspondence points belonging to the flying vehicle that is based on our previous work [1].

A. Feature Extraction

The edges of flying vehicles can include appropriate information for detecting. For this purpose, we have utilized edge information in image to extract interest features. In the GEFTD method, we have tried to opt strong interest points in order to detect the aerial vehicle from the fuzzified edges. These interest points contain the local edge and corner features that belong to either object or scene. For this purpose, first edge features are detected at frame t by means of the Canny edge detector algorithm [8]. Then, the edge fuzzifier procedure is employed to fuzzify each edge feature via a triangular membership function [7], [9]. Therefore, the neighbors of an edge point get a fuzzy belonging degree based on uncertainty principal in the fuzzy sets theory. In the edge fuzzifier procedure, the edge image, $\mathbf{I}_e$, is scanned using a square window with the fuzzy width of ω to produce the fuzzified edge image, $\mathbf{I}_f$. The fuzzified edge image which is an intensity image will calculate as follows:

$$\mathbf{I}_f(x+dx, y+dy) = \max\left(\mathbf{I}_e(x+dx, y+dy), 1-\frac{\max(|dx|,|dy|)}{\omega+1}\right), \quad (1)$$

in which (x, y) is the coordinate of central point of the window at frame t, and dx and dy are limited in the closed range of $[-\omega/2, +\omega/2]$. Producing fuzzified edges using the edge fuzzifier procedure have two benefits; firstly, it may be a chance to select an appropriate feature in the vicinities of an edge point because of their similarity. This case is proper where an uncertainty exist. Secondly, the space of process in image is mapped from binary scale to gray scale to be able to utilize the Kanade-Lucas-Tomasi (KLT) algorithm [10] for extracting good edge features. Fig. 1 illustrates a 1-D pixel array in which an edge pixel has fuzzified using a triangular membership function. The proposed GEFTD algorithm is expressed in Figure 2.

In the next stage of our flying vehicle detection algorithm, correspondence points will be obtained by means of the proposed MRFM processing at frame $t+dt$.

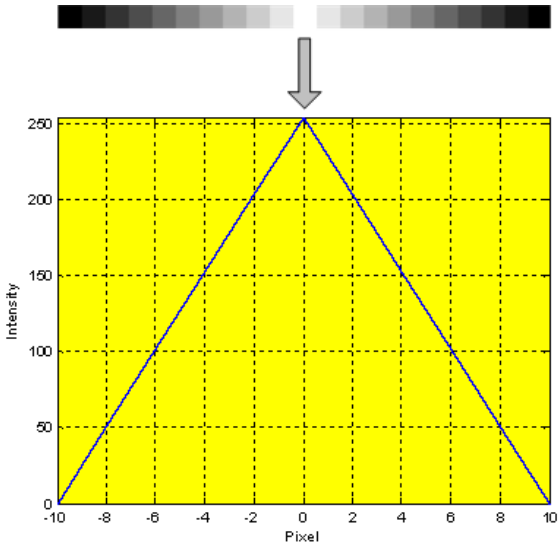

Figure 1. The fuzzification of an edge pixel.

1. (Input)
- ❶ Get the video frame t.

2. (Detect edge features via the Canny edge detector)

3. (Find edge contours using the connected component analysis)

4. (Filter each edge contour by Gaussian smoothing filter. In subsection II-*B*, we will explain about the manner of edge contours filtering.)

5. (Fuzzify each edge contour with the mentioned edge fuzzifier method)
- ❶ The output of the algorithm in this step is a gray scale image with fuzzified edges.

6. (Choose interest points, i.e. GEFTD, by means of the KLT feature extractor algorithm. By tuning the parameters of the KLT algorithm [10], we will select m features.)

7. (Output)
- ❶ The output of the algorithm is the interest points matrix with m extracted features:

$$\mathbf{IP} = \begin{pmatrix} ip_1(x) & ip_2(x) & \cdots & ip_m(x) \\ ip_1(y) & ip_2(y) & \cdots & ip_m(y) \end{pmatrix}.$$

Figure 2. The Proposed GEFTD algorithm.

B. Multi Resolution Feature Matching

We have proposed a new feature matching method based on fuzzified edge Gaussian pyramids to determine correspondence points at frame $t+dt$ with sub-pixel accuracy. For this purpose, edge pyramids have obtained in L levels (let us to nominate them as level 0 to $L-1$). For instance, different edge pyramid levels for one edge contour in image are determined as follows; in case:

$$r_{n0} = \left(x_{n0}(u), y_{n0}(u)\right), \tag{2}$$

is n^{th} edge contour parametric vector from pyramid level 0 (the level 0 of pyramid has the most image resolution and the level $L-1$ of pyramid has the least image resolution); then, two processes are performed to estimate correspondence parametric vector in the level k^{th} of pyramid, r_{nk}. First, 1-D signals smooth and then down-sampling process performs as follows:

$$X_{n0}(u,\sigma) = x_{n0}(u) \otimes g(u,\sigma), \tag{3}$$

$$Y_{n0}(u,\sigma) = y_{n0}(u) \otimes g(u,\sigma), \tag{4}$$

$$R_{n0}(u,\sigma) = \left(X_{n0}(u,\sigma), Y_{n0}(u,\sigma)\right), \tag{5}$$

$$r_{nk} = \frac{R_{n0}(2^k u,\sigma)}{2^k} = \left(\frac{X_{n0}(2^k u,\sigma)}{2^k}, \frac{Y_{n0}(2^k u,\sigma)}{2^k}\right), \tag{6}$$

where $\otimes$ represents the convolution operator and σ is the standard deviation of 1-D smoothing Gaussian filter. The value of σ must opt so that to be able to reduce the effect of noise over the signal as well as preserve significant details in edge contours. The proposed MRFM algorithm is expressed in Figure 3.

C. Motion Flow Vectors Classifier

After determining correspondence points, motion flow vectors can also draw (see Figure 6 (d)). The origin and end of a motion vector are the feature and correspondence points, respectively. The estimated local motion flow vectors at video sequences can classify into two object flow and scene flow classes. The magnitude of scene flows will be approximately zero if the video camera is fixed. In this situation, identifying the moving object is uncomplicated. Classifying flow vectors will be intricate if the both camera and object are moving in image acquisition. In the applications with mobile camera, Behrad and Motamedi [12] introduced a statistical approach based on the planar Affine transformation for locating moving objects in dynamic scenes. In mentioned method, the condition of object detection in image is $S_{\text{Object}} \le 0.5 S_{\text{Scene}}$, in which S_{Object} and S_{Scene} are the surface of object and scene in image, respectively. Here, a supervised machine learning algorithm based on MLPs neural network has employed to classify flow vectors without applying such limitation. In this method, first a MLPs network is trained with N different video samples. Then the intelligent machine is able to prognosticate input patterns. Each training sample includes two video frames (frame t and frame $t+dt$) that have captured from a scene with moving flying vehicle. The network has trained, in fact, with a 4-D feature space. This feature space contains both the feature points coordinate, (x, y), at frame t and the correspondence points coordinate, (x', y'), at frame $t+dt$, in which feature space is normalized to image dimension for reducing sensitivity to the size of training and testing images. For constructing the network, the observations matrix, $\mathbf{X}$, is organized as:

$$\mathbf{X} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\right), \tag{7}$$

where $\mathbf{x}^{(i)}$ is the i^{th} observation vector as follows:

$$\mathbf{x}^{(i)} = \left(\frac{x_1}{h} \cdots \frac{x_p}{h}, \frac{y_1}{w} \cdots \frac{y_p}{w}, \frac{x'_1}{h} \cdots \frac{x'_p}{h}, \frac{y'_1}{w} \cdots \frac{y'_p}{w}\right)^{\mathrm{T}}, \tag{8}$$

in which h and w are the height and width of video frames, and p is a variable that initialize in the feature selection algorithm. Fig. 4 describes the feature selection algorithm for training and testing the classifier. Our reason of choosing the mentioned feature space is that the weights of MLPs network be updated by passing some epochs based on the both magnitude and direction of the extracted motion flow vectors in each training record. In this algorithm, outlier flow vectors will remove by inspecting the status array [11], [13]. Flow vectors classify into two individual object and scene classes in the output of MPLs network by thresholding, T_{out}.

III. Empirical Results

As mentioned, we have utilized motion information between two video frames from a mobile camera to detect the flying vehicle. The proposed algorithm implemented using the Microsoft Visual C++ 8.0 compiler and tested with an Intel Pentium-IV 3.2GHz PC. We have tried to utilize different video data with diversification in atmospheric conditions, image acquisitions and dynamic scenes. In the experiments, we have been constructed a dataset including 135 video records. We have also examined the effect of constant illumination change on the algorithm. The all of training and testing records grab via different mobile monocular Day & Night CCD camera. The target and camera are moving in videos. Thus, the motion models of target and camera are different. In such situations, using simple and traditional methods like segmentation approaches based on frames differencing and thresholding are not responsible for analyzing motion. Therefore, the proposed method has high performance.

1. (Input)

❶ Get two video frames (e.g. frame t and $t+dt$).

2. (Calculate edge Gaussian pyramids in L levels for each frame)

3. (Estimate correspondence points in the highest level of pyramid, i.e. level $L-1$ using Lucas-Kanade (LK) optical flow algorithm [11]. The initial interest points matrix in level $L-1$ at frame t is:)

❶ $\mathbf{IP}_{L-1} = \begin{pmatrix} \frac{ip_1(x)}{2^{L-1}} & \frac{ip_2(x)}{2^{L-1}} & \cdots & \frac{ip_m(x)}{2^{L-1}} \\ \frac{ip_1(y)}{2^{L-1}} & \frac{ip_2(y)}{2^{L-1}} & \cdots & \frac{ip_m(y)}{2^{L-1}} \end{pmatrix}$;

❷ The obtained correspondence points matrix in level $L-1$ at frame $t+dt$ is:

$$\mathbf{CP} = \begin{pmatrix} cp_1(x) & cp_2(x) & \cdots & cp_m(x) \\ cp_1(y) & cp_2(y) & \cdots & cp_m(y) \end{pmatrix}.$$

4. (Consider the following matrix as an initial guess in level $L-2$ and calculate correspondence points again:)

❶ $\mathbf{CP} = \begin{pmatrix} cp_1(x)\times 2 & cp_2(x)\times 2 & \cdots & cp_m(x)\times 2 \\ cp_1(y)\times 2 & cp_2(y)\times 2 & \cdots & cp_m(y)\times 2 \end{pmatrix}$.

5. (Repeat steps 3 and 4 descendingly to reach the most level of resolution, i.e. pyramid level 0)

6. (Output)

❶ The output of the algorithm is correspondence points result in the fuzzified edges with sub-pixel accuracy.

Figure 3. The proposed MRFM algorithm.

A. Fuzzified Edge Gaussian Pyramids

As mentioned in subsection II-*B*, an algorithm with multi resolution processing based on fuzzified edge Gaussian pyramids has utilized to find correspondence points at frame $t+dt$. In Figure 5, results of edge Gaussian pyramids are shown in three levels for a typical aerial vehicle. In this sample, image acquisition has performed in a far distance. Figs. 5 (b) and 5 (c) exemplify edge Gaussian pyramids and fuzzified edge Gaussian pyramids, respectively. In experiments, the fuzzy width, ω, in the edge fuzzifier and the standard deviation of smoothing filter, σ, have been chosen equal to 5 and 3, respectively. The feature extractor performs in level 0 of fuzzified edge Gaussian pyramids at video frame t.

B. Estimating Motion Flow Vectors

After finding correspondence points, motion flow vectors can approximate. Fig. 6 depicts results of estimating motion flow vectors for three flying vehicles. The desired outputs to learn the network are determined based on flow vectors information. Video records have been grabbed from disparate situations; for instance, first sample shows an aircraft in cloudy sky; second sample exhibits a fighter Jet during maneuver in clear sky that has zoomed in over the object; third sample shows an aircraft in a high textured background.

C. Results of Motion Flow Vectors Classifier

In the proposed method, we have been separated the constructed dataset into two train-sets and test-sets including 100 and 35 video records, respectively ($N=100$). We have designated 25 pairs of feature including feature and correspondence points to learn the network initially, which have extracted from each video record ($p=25$). Therefore, the network has 100 input nodes. The MLPs network has also constructed by two hidden layers, in which each layer has formed with 20 neurons. We have utilized the standard error back-propagation algorithm to train the network. The applied activation transfer function is the log sigmoid in neurons.

1. (Input)

❶ Get the m set of correspondence points (as initial correspondence points) at video frame $t+dt$ and their relevant features at video frame t.

2. (Initialize the variable p so that is $m \geq \text{Floor}(4p/3)$)

3. (Remove outlier flow vectors from list)

❶ $temp = 0$;

❷ for $i = 1,2,\ldots,m$

if($status[i] == 1$)

$FinalFeatures[temp] = InitialFeatures[i]$;

$FinalCorrespondences[temp] = InitialCorrespondences[i]$;

$temp++$;

end if

end for i.

4. (Estimate Normalized Cross Correlation [14] between features and their correspondences)

5. (Sort similarity values in ascending order)

6. (Select the first p features and their correspondences with the most similarity)

7. (Output)

❶ The output of the algorithm is p prepared flow vectors for training as well as testing the flow vectors classifier.

Figure 4. The feature selection algorithm.

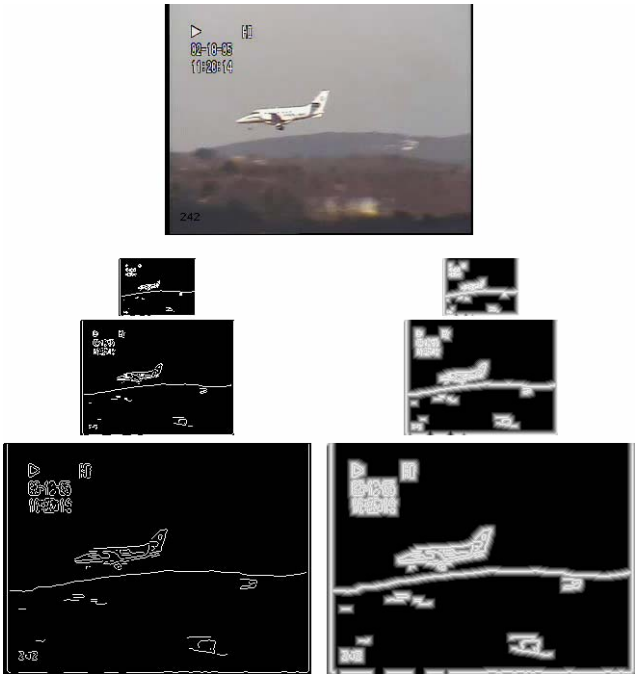

Figure 5. Images from up to down and left to right: (a) original image, (b) edge Gaussian pyramids, and (c) fuzzified edge Gaussian pyramids.

Figure 6. Images from left to right: (a) video frame t, (b) video frame $t+dt$, (c) the results of feature extraction from fuzzified edges, and (d) estimating flow vectors.

Fig. 7 shows the results of flow vectors classifier for four aerial vehicles. In the algorithm, the threshold T_{out} has chosen equal to 0.86. We have fitted a convex hull on classified features that have chosen as the flying vehicle features in the output of the network.

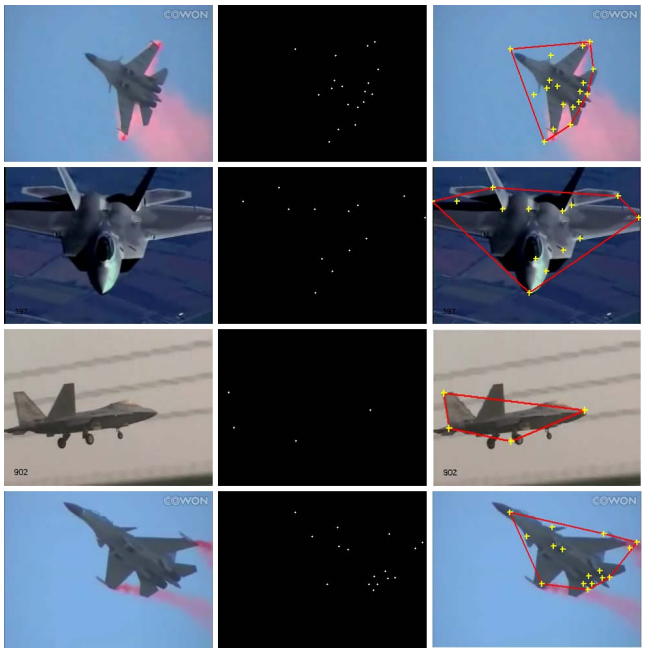

Figure 7. Images from left to right: (a) video frame $t+dt$, (b) the output results of classifier, and (c) the localization of target by fitting convex hull algorithm.

D. *Performance Evaluation*

In order to evaluate the performance of the proposed method we have utilized a criterion to measure the Detection Error (DE) which its definition is compatible with the performance evaluation definitions by Kasturi et al. [15]. For this purpose, the DE in terms of percent can estimate by means of:

$$DE(\%) = 100 \times \left(\frac{S(C \cup F) - S(C \cap F)}{S(C \cup F)} \right), \qquad (9)$$

in which C is the estimation of flying vehicle contour that has determined based on the convex hull fitting, F is the actual contour of flying vehicle and operator $S(\cdot)$ represents the surface of closed contour. It is important to note that the actual surface of flying vehicle has manually characterized by user in order to calculate DE. In Table I, we have measured DE in video records that have shown in Figure 7, respectively. As is obvious in Table I, DE in video sample 1 is more that others.

TABLE I. ESTIMATING *DE* FOR THE SHOWN FLYING VEHICLES IN FIGURE 7

Video Records	$DE(\%)$
Video sample 1	12.11
Video sample 2	9.53
Video sample 3	10.07
Video sample 4	8.69

IV. CONCLUSION

In this paper, we have suggested a vision based approach to detect flying vehicle. The base of detection is to estimate motion information at two video sequences from a mobile CCD camera. In our method, we have presented a new feature extraction and correspondence algorithm to reduce the sensitivity against the constant and uniform illumination change by processing on the fuzzified edge images than processing on ordinary gray level images. The proposed algorithm is computationally efficient and more complex than methods like segmentation approaches based on frames differencing, image thresholding and background modeling. The proposed algorithm can also utilize in vision based aerial vehicle tracking to estimate both the kernel and contour of object; whereas methods such as Kalman filter and mean shift algorithm determine only the kernel of object. Target states like location, position, size, the form of object and so on can take into account as the expected outputs of the proposed detection system for high levels purposes in the control of flying vehicles.

REFERENCES

[1] A. Taimori, and S. Sabouri, "Vision Based Flying Vehicle Tracking," IEEE International Conference on Systems, Man, and Cybernetics, pp. 330–335, 2008.

[2] J.D. Redding, T.W. McLain, R.W. Beard, and C.N. Taylor, "Vision-Based Target Localization from a Fixed-Wing Miniature Air Vehicle," Proceedings of the IEEE American Control Conference, 2006.

[3] J.J. Kehoe, R.S. Causey, A. Arvai, and R. Lind, "Partial Aircraft State Estimation from Optical Flow Using Non-Model-Based Optimization," Proceedings of the IEEE American Control Conference, 2006.

[4] Y. Jianchao, "A New Scheme of Vision Based Navigation for Flying Vehicles-Concept Study and Experiment Evaluation," 7th IEEE International Conference on Control, Automation, Robotics and Vision, vol. 2, pp. 643–648, 2002.

[5] A. Betser, P. Vela, and A. Tannenbaum, "Automatic Tracking of Flying Vehicles Using Geodesic Snakes and Kalman Filtering," 43th IEEE International Conference on Decision and Control, vol. 2, pp. 1649–1654, 2004.

[6] J. Ha, C. Alvino, G. Pryor, M. Niethammer, E. Johnson, and A. Tannenbaum, "Active Contours and Optical Flow for Automatic Tracking of Flying Vehicles," Proceedings of the IEEE American Control Conference, pp. 3441–3446, 2004.

[7] W. Siler, and J. J. Buckley, "Fuzzy Expert Systems and Fuzzy Reasoning," John Wiley & Sons, Inc., 2005.

[8] R. C. Gonzalez, and R. E. Woods, "Digital Image Processing," Prentice Hall, Inc., 3rd Edition, 2008.

[9] A.R. Behrad, S.A. Motamedi, K. Madani, and M. Esnaashari, "A New Algorithm for Target Tracking Using Fuzzy-Edge-Based Feature Matching and Robust Statistic," IEEE International Conference on Image Processing, vol. 1, pp. 577–580, 2002.

[10] J. Shi, and C. Tomasi, "Good Features to Track," IEEE International Conference on Computer Vision and Pattern Recognition, pp. 593–600, 1994.

[11] G. Bradeski, and A. Kaehler, "Learning OpenCV," O'Reilly Media, Inc., pp. 329–334, 2008.

[12] A.R. Behrad, and S. A. 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.

[13] J-Y. Bouguet, "Pyramidal Implementation of Lucas Kanade Feature Tracker Description of the Algorithm," Intel Corporation, Microprocessor Research Labs, OpenCV Documentation, 2001.

[14] E. Trucco, and A. Verri, "Introductory Techniques for 3-D Computer Vision," Prentice Hall, Inc., 1998.

[15] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "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, vol. 31, no. 2, pp. 319–336, 2009.