ELSEVIER

Image and Vision Computing 21 (2003) 623–635

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Target tracking in airborne forward looking infrared imagery

Alper Yilmaz*, Khurram Shafique, Mubarak Shah

Department of Computer Science, Univ. of Central Florida, Orlando, FL 32816-2362, USA

Abstract

In this paper, we propose a robust approach for tracking targets in forward looking infrared (FLIR) imagery taken from an airborne moving platform. First, the targets are detected using fuzzy clustering, edge fusion and local texture energy. The position and the size of the detected targets are then used to initialize the tracking algorithm. For each detected target, intensity and local standard deviation distributions are computed, and tracking is performed by computing the mean-shift vector that minimizes the distance between the kernel distribution for the target in the current frame and the model. In cases when the ego-motion of the sensor causes the target to move more than the operational limits of the tracking module, we perform a multi-resolution global motion compensation using the Gabor responses of the consecutive frames. The decision whether to compensate the sensor ego-motion is based on the distance measure computed from the likelihood of target and candidate distributions. To overcome the problems related to the changes in the target feature distributions, we automatically update the target model. Selection of the new target model is based on the same distance measure that is used for motion compensation. The experiments performed on the AMCOM FLIR data set show the robustness of the proposed method, which combines automatic model update and global motion compensation into one framework.

Keywords: FLIR imagery; Target tracking; Target detection; Global motion compensation; Mean-shift

1. Introduction

Detection and tracking of moving or stationary targets in FLIR imagery are challenging research topics in computer vision. Though a great deal of effort has been expended on detecting and tracking objects in visual images, there has been only limited amount of work on thermal images in the computer vision community.

The thermal images are obtained by sensing the radiation in the infrared (IR) spectrum, which is either emitted or reflected by the object in the scene. Due to this property, IR images can provide information which is not available in visual images. However, in contrast to visual images, the images obtained from an IR sensor have extremely low signal to noise ratio (SNR), which results in limited information for performing detection or tracking tasks. In addition, in airborne forward looking infrared (FLIR) images, non-repeatability of the target signature, competing background clutter, lack of a priori information, high ego-motion of the sensor and the artifacts due to the weather conditions make detecting or tracking targets even harder.

To overcome the shortcomings of the nature of the FLIR imagery, different approaches impose different constraints to provide solutions for a limited number of situations. For instance, many target detection methods require that the targets are hot spots which appear as bright regions in the images [1,2,3]. Similarly, several target tracking algorithms require one or both of the following assumptions to be satisfied: (1) no sensor ego-motion [4] and (2) target features do not drastically change over the course of tracking [3,5,6]. However, in realistic tracking scenarios, neither of these assumptions are applicable, and a robust tracking method must successfully deal with these problems. To the best of our knowledge, there is no such published method that provides a solution to both of these problems in one framework.

In this paper, we present an approach for real-time target tracking in FLIR imagery in the presence of high global motion and changes in target features, i.e. shape and intensity. Moreover, the targets are not required to have constant velocity or acceleration. The proposed tracking algorithm uses the positions and the sizes of targets determined by the target detection scheme. For target detection, we apply steerable filters and compute texture energies of the targets, which are located using a segmentation-based approach. Once the targets are detected, the tracking method employs three modules to

* Corresponding author.

E-mail addresses: yilmaz@cs.ucf.edu (A. Yilmaz), khurram@cs.ucf.edu (K. Shafique), shah@cs.ucf.edu (M. Shah).

0262-8856/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0262-8856(03)00059-3

perform tracking. The first module, which is a modified version of Ref. [7], is based on finding the translation vector in the image space that minimizes the distance between the distributions of the model and the candidate. The distributions are obtained from the intensity and local standard deviation measure of the frames. The local standard deviation measure is obtained in the neighborhood of each pixel in the frame and provides a very good representation of frequency content of the local image structure. Based on the distance measure computed from the target feature distributions, the other two modules compensate for the sensor ego-motion and update the target model. The global motion estimation module uses the multi-resolution scheme of Ref. [8] assuming a planar scene under perspective projection. It uses Gabor filter responses of two consecutive frames to obtain the pseudo-perspective motion parameters.

The remainder of the paper is organized as follows: Section 2 discusses the recent literature on detecting and tracking targets in FLIR imagery. In Section 3, the target detector which is used to initialize the tracking algorithm with the position and the size of the target is described. Section 4 presents a discussion on the tracking problems and gives the details of the tracking algorithm which uses the two modules (1) automatic target model update (Section 4.3), (2) the sensor ego-motion compensation (Section 4.4). The implementation details are outlined in Section 4.5. Finally, experimental results for the proposed tracking method are presented in Section 5 and conclusions are drawn in Section 6.

2. Related work

In this section, we examine some of the representative works reported in the literature on detecting and tracking targets in FLIR imagery. In general, existing methods on IR images work for a limited number of situations due to the constraints imposed on the solution.

For detection of FLIR targets, many methods rely on the ithot spot technique, which assumes that the target IR radiation is much stronger than the radiation of the background and the noise. The goal of the target detectors is then to detect the center of the region with the highest intensity in the image, which is called the ithot spot. The hot spot seekers use various spatial filters to detect targets in the scene. Takken et al.[2] developed a spatial filter based on least mean square (LMS) to maximize the signal to clutter ratio for a known and fixed clutter environment. Chen et al. [1] modeled the underlying clutter and noise after local demeaning as a whitened Gaussian random process, and developed a constant false alarm rate detector using the generalized maximum likelihood ratio.

Temporal filters like Triple Temporal Filter (TTF), Infinite Impulse Response (IIR) and Continuous Wavelet Transform (CWT) have been widely used. Lim et al.[9] have presented a multistage IIR filter for detecting dim point targets. Tzannes[10] presented a Generalized Likelihood Ratio Test (GLRT) solution to detect small (point) targets in a cluttered background when both the target and clutter are moving through the image scene.

Similar to the target detection methods, target tracking approaches also impose constraints on the solution, such as no sensor ego-motion or no target modal change. However, even with these assumptions, the tracking performance of most methods is not convincing. Below, we will briefly summarize commonly cited methods which have attempted to deal with these problems.

To compensate the global motion, Strehl and Aggarwal [5] have used a multi-resolution scheme based on the affine motion model. The affine model has its limitations and for FLIR imagery, which is obtained from an airborne sensor; it is unable to capture the skew, pan and tilt of the planar scene.

Similarly, Shekarforoush and Chellappa [3] first compensate for the sensor ego-motion to stabilize the FLIR sequence, then detect very hot or very cold targets. The stabilization and tracking is based solely on the goodness of the detection and the number of targets, i.e. if the number of targets is not adequate, or there is significant background texture, the system is not able to detect sufficient number of targets. Therefore stabilization fails to correctly register the image.

Braga-Neto and Goutsias [6] have presented a method based on morphological operators for target detection and tracking in FLIR imagery. Their tracker is based on the assumptions that the targets do not vary in size, they are either very hot or very cold spots, and sensor ego-motion is small. However, these assumptions contradict the nature of airborne FLIR imagery.

Davies et al. [4] proposed a multiple target tracker system based on Kalman filters for small targets, which uses the output of the Daubechies wavelet family for FLIR imagery. The method assumes constant acceleration of the target, which is not valid for maneuvering targets. In addition, the method works only for sequences with no global motion.

In this paper, we propose a solution for tracking targets in airborne FLIR imagery, which is motivated by the need to overcome some of the shortcomings of the existing tracking techniques. In contrast to the previous methods on tracking targets in FLIR imagery, we relax the constraints on motion smoothness, brightness constancy and provide a robust target tracking algorithm, which minimizes a multi-objective functionconstructed using intensity and local standard deviation distributions. The tracking algorithm discussed in this paper requires an initialization of the target bounding box in the frame where the target first appears. Section 3 describes this target detection step.

3. Target detection

Detection of targets in the FLIR sequences is a hard problem because of the variability of the appearance of targets due to atmospheric conditions, background, and thermodynamic state of the targets. Most of the time, the background forms similar shapes to those of the actual targets, and the targets become obscured.

Since the theme of this research is target tracking, we perform an initial target detection similar to Ref. [11]. In our implementation, we focused only on hot targets, which appear as bright spots in the scene, having high contrast with the neighboring background. This initial step uses the intensity histogram and partitions the intensity space, while assigning ambiguous regions according to fuzzy c-means clustering. Then, a merging phase is employed which fuses the edge information and brightness constraints [12]. The segmentation results of these initial steps are shown in Fig. 1.

Once the regions are segmented using the outlined method, a confidence measure for each candidate region is computed as the product of two sigmoid functions:

$$C_i = \underbrace{\frac{1}{(1+\mathrm{e}^{-\lambda_1(\mu_f-\mu_1)})}}_{\text{target brightness}} \times \underbrace{\frac{1}{(1+\mathrm{e}^{-\lambda_2(\mu_f-\mu_b-\mu_2)})}}_{\text{contrast with background}} \tag{1}$$

where μ_f and μ_b, respectively, are the means of the foreground and the background of the ith target, λ_1 and λ_2 control the slope of the sigmoids and μ_1 and μ_2 are the offsets of the sigmoids. If a target region is bright with a high contrast with its neighborhood, then Eq. (1) assigns a confidence close to 1, otherwise the confidence will be close to 0. Regions with high confidence are selected as possible targets.

In Fig. 2, we show the target candidates detected using the scheme outlined above. However, as seen from the last example in Fig. 2, contrast and brightness are not always sufficient to characterize correct targets. Following this observation, we perform an additional texture analysis step for each target candidate and find the similarity between the target candidate and its immediate neighborhood. Texture of an image can be analyzed through its spectral content of the subband signals, which are obtained by filtering the image with filter banks. For images, an efficient way to code spectral content is by computing the energy [13]

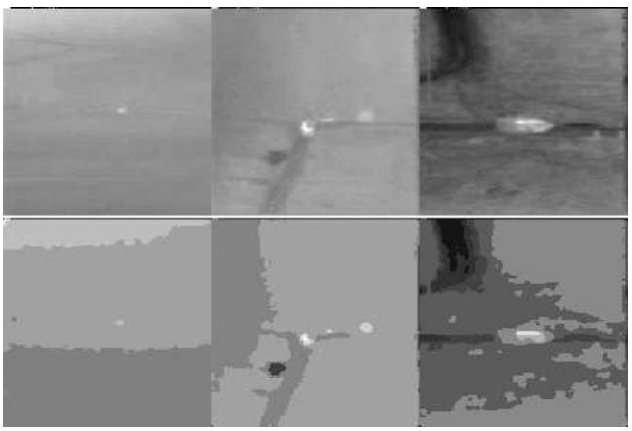

Fig. 1. First row: input images; second row: corresponding segmentation results.

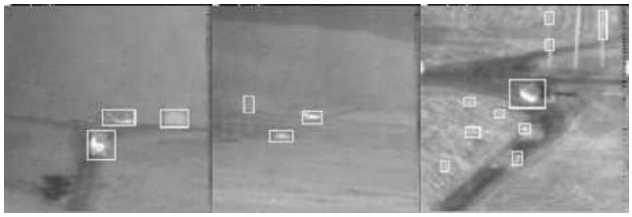

Fig. 2. Target candidates detected using the fuzzy clustering and edge fusion.

$$e_i = \frac{1}{M \cdot N} \sum_{x=1}^{M} \sum_{y=1}^{N} G_i^2(x,y)$$

where N and M are respectively number of rows and columns of the window and $G_i = F_i^* I$ is filter response of image I using filter F_i. To define the texture content of the target, an energy vector is constructed $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_k)^{\mathrm{T}}$ for the target windows. Unless a camouflaged target is present, the texture of target should be different from neighboring regions. To utilize this, eight overlapping windows in the neighborhood of the target are selected and their energy vectors, $E_1, E_2, \ldots, E_8$, are computed. Selection of overlapping windows is done by moving the target window half way in eight directions as shown in Fig. 3.

Texture similarity of the target window and the neighboring window can be computed by Euclidean distance measure:

$$\mathrm{dist} = \min_{i|i\leq 8} \sqrt{\sum_{j=1}^{k} (E(j) - E_i(j))^2}$$

where k is the number of filters. If dist $> \tau$,where τ is a predefined threshold then we confirm that the candidate is a true target, otherwise we label the candidate as false positive.

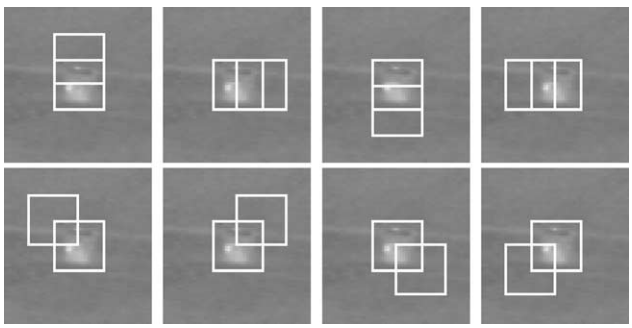

Fig. 3. Selection of rectangular target regions for texture analysis.

In order to obtain better categorization and to eliminate the false positives, we analyzed the performance of the discussed detection scheme using different texture measures including:

- *Law's Texture Measures*: They were proposed in a feature extraction scheme based on gradient operators. This scheme uses 25 masks which are obtained by convolution of five 1-dimensional vectors each representing level, edge, spot, wave and ripple [14].
- *Wavelets*: Wavelets decomposition is obtained by separable filter banks and every decomposition contains information of a specific scale and orientation [15]. We used different families of wavelet: Bi-orthogonal, Haar, Daubechies and Quadrature mirror wavelets.
- *Steerable Filters*: Similar to wavelets, steerable filters are also a tool for multi resolution analysis. This is a type of over-complete wavelet transform whose basis functions are directional derivative operators in different sizes and orientations [16]. We used two different types of steerable filters: SP-3 and SP-5, where SP-3 has 4 and SP-5 has 6 orientation subbands.

For detailed information about these texture measures, we refer the readers to the cited publications.

In order to obtain realistic comparison results, we manually selected 200 target regions and 200 non-target regions from a wide spectrum of FLIR imagery. Then, we computed the energies of each candidate region for the texture measures given above. The categorization performance for these texture measures are shown in Table 1, where the categorization means the target is correctly categorized as the target and the non-target correctly categorized as the non-target. Compared to the others, steerable pyramid type SP-3 had the best performance with 94% correct categorization. In addition, it can also be seen that the performance of Law's Texture Measures is comparable to the steerable filters. Since Law's Texture Measures have low computational cost, they are a good candidate for real-time systems.

Table 1
Correct categorization performance of different methods

Method Name	Perf. (%)
Steerable Filter (SP-3)	94
Steerable Filter (SP-5)	93
Law's texture energy filter	88
Bi-orthogonal wavelet filter	82
Haar wavelet filter	82
Daubechies wavelet filter (Daub-3)	81
Daubechies wavelet filter (Daub-2)	80
Daubechies wavelet filter (Daub-4)	80
Quadrature mirror wavelet filter (QMF-16)	80
Quadrature mirror wavelet filter (QMF-13)	79
Quadrature mirror wavelet filter (QMF-5)	79
Quadrature mirror wavelet filter (QMF-9)	76

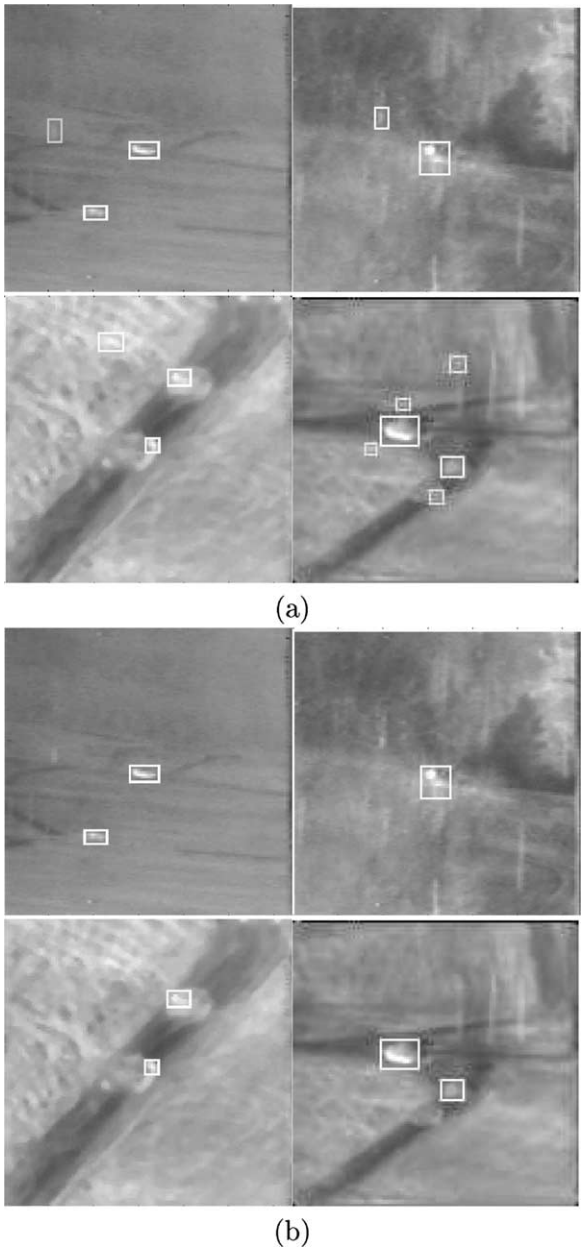

Fig. 4. (a) Candidate target positions, (b) detected targets by eliminating the false positives using Steerable Filters. Note that all true targets are correctly detected.

In Fig. 4(a), we show the target candidates, and in Fig. 4(b) we show detected targets after eliminating the false positives using Steerable Filters.

Once the targets are detected using the described method, we use the positions and the sizes of the targets to initialize the proposed target tracking system, which is detailed in Section 4.

4. Target tracking

We can categorize target tracking approaches into two classes. The first class is the correspondence based approaches, where the moving objects are detected in each frame and then the correspondences between the detected targets in the current and the previous frames are established. In contrast, the second class of approaches require target detection only in the first frame, and a target model, e.g. target template or intensity distribution, is extracted and used in performing tracking in the subsequent frames. There are several ways to generate a target model and perform tracking. A very commonly used approach among researchers is to compute the correlation between the target template and the potential target regions in the next frame and find the best match. However, the search involved in such correlation based methods is very time consuming and they are prone to errors due to changes in the target model.

Here, we follow the 'mean-shift tracking' approach which was proposed by Comaniciu et al. [7]. This method relies on the intensity distributions generated from the target region and computes the translation of the target center in the image space. However, due to the nature of target in closing sequences of FLIR imagery, our system differs from Ref. [7] in two aspects:

–In addition to two-dimensional kernels used to define the density of target in spatial domain, one-dimensional kernel density estimates are used in generation of smooth feature distributions. This is required to have a more realistic distribution model for small targets that appear in closing FLIR sequences.

–We fuse the information obtained from local standard deviation of the target region by computing its kernel density estimate. Due to target's low contrast with the background, this modification is necessary for FLIR imagery.

The local standard deviation of a pixel $\mathbf{x}_i$ in the image is computed from its neighborhood defined by M using

$$S(\mathbf{X}) = \sqrt{\frac{1}{|\mathbf{M}| - 1} \sum_{x_i \in \mathbf{M}} (\mathbf{I}(\mathbf{X}_i) - \mathbf{I}(\mathbf{X}))^2} \tag{2}$$

where $I : \mathbb{N}^{\not\#} \rightarrow \mathbb{N}^{\not\#}$ is the imaging function, $\mathbf{x}_i$ are the spatial location and $|M|$ denotes number of pixels in the neighborhood. In our experiments we select M as a 5×5 window. In Fig. 5, we show the standard deviation image generated using Eq. (2). As can be seen, the target regions are clearly emphasized.

4.1. Tracking model

There are different appearance models for targets in general. For instance, in template matching methods the appearance remains constant and is only good for tracking in very short durations but it performs poorly for longer durations [17], which generally occur in FLIR imagery. Here we use a Bayesian model to cope with small variations in appearance of FLIR targets. Our appearance models are probability density functions (pdf) of both the intensity values and the local standard deviations and are estimated using a kernel density estimation [18]:

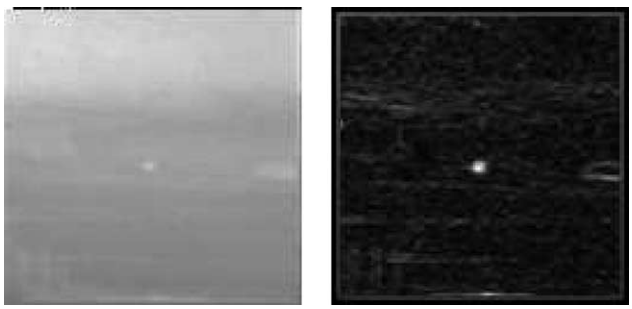

Fig. 5. Left: input image; right: standard deviation image.

$$f_K(\mathbf{m}) = \frac{1}{nhd} \sum_{i=1}^{n} \mathbf{K}(\mathbf{x}_i - \mathbf{m}) \tag{3}$$

where $\mathbf{m}$ is the center of a d-dimensional kernel, n is the number of points inside the kernel and h is bandwidth of the kernel. In the simplest case, kernel density estimation can be generated with a uniform kernel, where the resulting itpdf will be the histogram of data. Other possible kernels include Gaussian kernel, Triangular kernel, Bi-weight kernel, Epanechnikov kernel, etc. Among these kernels, in the continuous domain the Epanechnikov kernel yields the best minimum mean integrated square error between two kernel densities [18].

For images, construction of *pdfs* using density estimation does not incorporate spatial relation of the intensities. To incorporate the spatial relation, the kernel density estimation is defined by cascading two Epanechnikov kernels. The first kernel is used to define spatial relation of the feature through Euclidean distance of its spatial position from the target centroid; ie. we place a two-dimensional kernel centered on the target centroid and kernel values are used as spatial weights. Two-dimensional Epanechnikov kernel, which captures spatial relationship, is given by

$$K_2(\mathbf{x}) = \begin{cases} \frac{2}{\pi h^2}(h^2 - \mathbf{x}^{\mathrm{T}}\mathbf{x}^{\mathrm{T}}) & \mathbf{x}^{\mathrm{T}}\mathbf{x}^{\mathrm{T}} < \mathbf{h}^2 \\ 0 & \text{otherwise} \end{cases}$$

where h is the radius of the kernel [7]. The second kernel is used as a weighing factor in the feature histogram; i.e. we place a one-dimensional kernel centered on the feature value and kernel values are used as weights. One dimensional Epanechnikov kernel is given by

$$K_1(x) = \begin{cases} \frac{3}{4\pi h^3}(h^2 - x^2) & x < h \\ 0 & \text{otherwise} \end{cases}$$

such that $\int_{-h}^{h} K_1(x)\mathrm{d}x = 1$. Using the cascaded kernels, density estimate of feature u for a target can be estimated from

$$P(u) = \frac{\sum_{i=1}^{n} K_1(I(\mathbf{x}_i) - \mathbf{u}) K_2(\mathbf{x}_i - m)}{C} \tag{4}$$

where C is the normalization constant, $C = \sum_{i=1}^{n} K_2(\mathbf{m} - \mathbf{x}_i)$ and the bandwidths, h_1 and h_2 for both kernels are specified separately. Fig. 6 shows density estimation of an target computed using Eq. (4)

Section 4.2 gives details on how we utilize the information obtained from two modalities: intensity and the local standard deviation (Fig. 6).

4.2. Methodology

Assume that the target first appeared in the 0th frame, and $\mathbf{m}_0$ denotes its center. To track the target in the succeeding frames, kernel density estimates of each itbin for both the intensity Q_I and the standard deviation Q_S images are computed using Eq. (4).

Using the target model defined in Section 4.1, one possible way to find the target position in the current frame is to search neighboring regions for a distribution similar to the model computed for different scales of two-dimensional kernel in the neighboring regions [19]. Although such an approach is more stable to changes in the target features compared to template matching based methods, it is still computationally expensive. We will rather locate the target position directly by minimizing the distance between the model and the candidate and model *pdfs* [7], which is defined by

$$d(\mathbf{m}) = \sqrt{1 - \rho(\mathbf{m})} \tag{5}$$

where $\rho(\mathbf{m})$ is the modified Bhattacharya coefficient which fuses the information obtained from two different modalities: intensity and local standard deviation. Considering Eq. (5), the modified Bhattacharya coefficient can also be interpreted as the likelihood measure between the model and the candidate distributions, and is given by

$$p(\mathbf{m}) = \sum_{u=1}^{b} \left(\lambda \underbrace{\sqrt{P_{I_u}(\mathbf{m})\mathbf{Q}_{\mathbf{I_u}}}}_{\text{intensity based}} + (1-\lambda) \underbrace{\sqrt{P_{S_u}(\mathbf{m})\mathbf{Q}_{\mathbf{S_u}}}}_{\text{stdv. based}} \right) \tag{6}$$

where b is the number of bins common to both the intensity and standard deviation distributions and $\lambda \in [0,1]$ is the parameter which balances the contribution of intensity and local standard deviation features. Expanding the likelihood to Taylor series around previous target position $\mathbf{m}_0$ gives

$$\rho(\mathbf{m}) = \rho(\mathbf{m}_0) + \frac{1}{4\mathbf{C}} \sum_{i=1}^{n} \mathbf{K}_2(\mathrm{m} - \mathbf{x}_i)$$

$$\left(\sum_{u=1}^{b} K_1(I(\mathbf{x}_i) - \mathbf{u}) \sqrt{\frac{\mathbf{Q}_{\mathbf{I_u}}}{\mathbf{P}_{\mathbf{I_u}}(\mathbf{m}_0)}} + \sum_{u=1}^{b} K_1(S(\mathbf{x}_i) - \mathbf{u}) \sqrt{\frac{\mathbf{Q}_{\mathbf{S_u}}}{\mathbf{P}_{\mathbf{S_u}}(\mathbf{m}_0)}} \right) \tag{7}$$

where $S(\mathbf{x}_i)$ denotes the normalized standard deviation measure computed using Eq. (2). Discarding the constant terms in Eq. (7), the likelihood of each pixel belonging to the target can be defined by the following function [7]:

$$\psi_i = \sum_{u=1}^{b} K_1(I(\mathbf{x}_i) - \mathbf{u}) \sqrt{\frac{\mathbf{Q}_{\mathbf{I_u}}}{\mathbf{P}_{\mathbf{I_u}}(\mathbf{m}_0)}} + \sum_{u=1}^{b} K_1(S(\mathbf{x}_i) - \mathbf{u}) \sqrt{\frac{\mathbf{Q}_{\mathbf{S_u}}}{\mathbf{P}_{\mathbf{S_u}}(\mathbf{m}_0)}} \tag{8}$$

where $i = 1 \ldots n$. In order words, ψ_i denotes the concentration of the target in the spatial kernel defined by K_2. Here we assume ψ is normalized such that $\sum_{i=1}^{n} \psi_i = 1$. Starting from the center of a differentiable kernel like K_2, an upclimbing scheme can be used to maximize the concentration of the target, and the gradient of the density estimate which is given in Eq. (3), $\hat{\nabla} f$, will point to the new kernel center at every iteration [20,21]. Translating the image origin to kernel center, such that $\mathbf{m} = 0$, The mean-shift vector is given by

$$\mathbf{m}_1 = \mathbf{m}_0 + \frac{4}{\pi \mathbf{h} 4} \sum_{i=1}^{n} \psi_i x_i \tag{9}$$

where $\mathbf{m}_1$ is the new target position.

The scheme outlined above can be used for tracking targets whose features remain constant throughout the sequence. However, in general, targets don't have a constant brightness or contrast and the feature distributions generated from the target can change. In Section 4.3, we propose a solution to overcome this shortcoming.

4.3. Target model update

During the course of tracking, model based tracking methods often suffer from abrupt changes in target model.

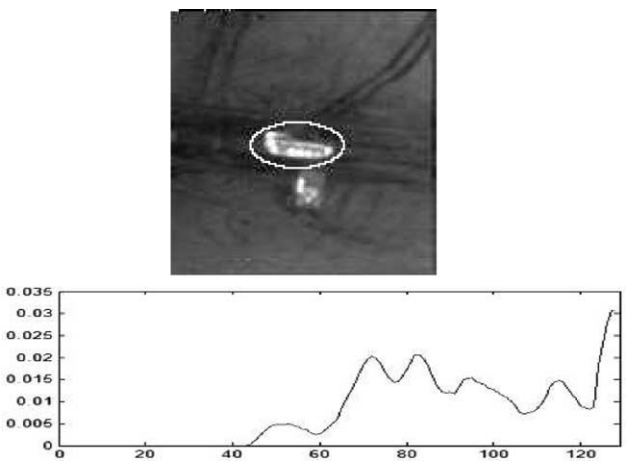

Fig. 6. Source image (top); density estimate from cascaded kernels of Eq. (4) (bottom).

The simplest way to change the target model is to periodically update the feature distributions. However, due to low contrast of the target with its background, the update may not necessarily occur when the target is correctly localized. Another straight forward solution is to change the target model using a constant threshold on the similarity metric used in tracking. For instance, for correlation based methods, the model can be updated if the correlation of the model with the target is higher than the threshold. Similarly, for the method outlined in Section 4.2, the model can be updated if the distance calculated in Eq. (5) is lower than a threshold [22]. The basic problem with using a constant threshold is to select the right value for all the sequences, i.e. a particular threshold may work very well for one sequence, but it may fail for others.

The model can be automatically updated using sequence-specific information about the rate of change of target features, which can be computed using the distance measure given in Eq. (5). In our implementation, the target model refers to the distributions of the target intensity and the local standard deviation measures. The rate of change for target intensity and local standard deviation differs from one sequence to another. To utilize sequence-specific changes of the target features over time, the distribution of the distance (see Fig. 7 an example distribution) is modeled by the Gaussian distribution. The Gaussian distribution parameters, mean μ and standard deviation σ, are updated at each frame using:

$$\mu_k = \frac{(k-1)\mu_{k-1} + d_k}{k} \tag{10}$$

$$\sigma_k^2 = \sigma_{k-1}^2 + (\mu_k - \mu_{k-1})^2 + \frac{d_k - \mu_k}{k-1} \tag{11}$$

where k denotes the current frame number. The decision whether to update the model is made based on the current value of the distance d_k, i.e. if $d_k < m_k - 2\sigma_k$ then target model is updated. This scheme guarantees that the model is updated if the target distribution change is within the acceptable range for a particular sequence. For instance, for a sequence where the target distribution changes very rapidly, such that the μ_k and σ_k are high, the acceptable range for model update will be wide.

In Fig. 8, distances between the model distribution and the distribution for the new target position computed with and without model update are shown. In Fig. 8(a), big jumps in the distance plot are due to the intensity change of the target for some parts of the sequence. These problems are corrected by model update as shown in Fig. 8(b). More results are given in Section 5.

One limitation of the mean-shift based tracking is that at least some part of the target in the next frame should reside inside the kernel [7]. However, targets in airborne FLIR imagery have high motion discontinuities due to sensor ego-motion. In Section 4.4, we describe the proposed approach to overcome the tracking problems due to sensor ego-motion.

4.4. Sensor ego-motion compensation

FLIR sequences obtained via an airborne moving platform suffer from abrupt discontinuities in motion. Because of this, the output of the tracker becomes unreliable, and requires compensation for the sensor ego-motion.

There are several approaches presented in the literature for ego-motion (global) compensation. However, they are not directly applicable to FLIR imagery due to lack of texture and low SNR. For instance, we have noted that motion compensation using the intensity values does not result in good estimation of motion parameters.

Therefore, we employ images filtered by two-dimensional Gabor filter kernels, which are oriented sine-wave gratings that are spatially attenuated by a Gaussian window [23]. Two-dimensional Gabor filters have the form

$$G_i(x,y) = \mathrm{e}^{-\pi[x^2/\alpha^2 + y^2/\beta^2]} \cdot \mathrm{e}^{-2\pi i[u_0 x + v_0 y]} \tag{12}$$

where α and β specify effective width and height, while u_0 and v_0 specify modulation of the filter [24]. In our implementation, we used the real parts of the Gabor responses for four different orientations. The responses of the Gabor filters are then summed and used as the input for the global motion compensation module. In Fig. 9, we show a selected frame from one of the FLIR sequences along with its Gabor response.

To compensate for the global motion, we employ the multi-resolution framework of Ref. [25]. The compensation

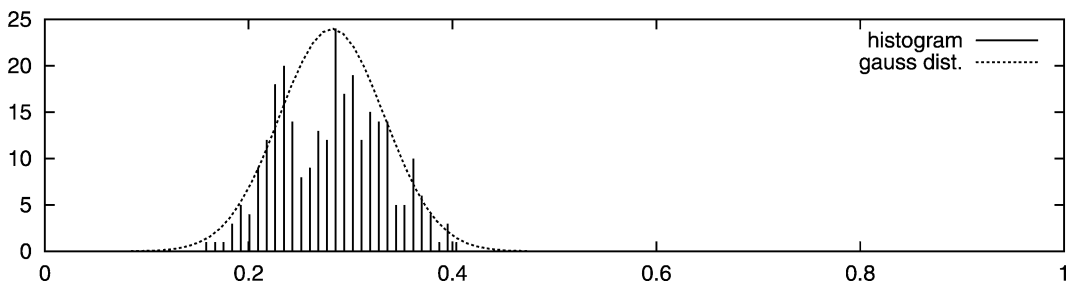

Fig. 7. Histogram of distances calculated using Eq. (5), and Gaussian model fitted to the distribution.

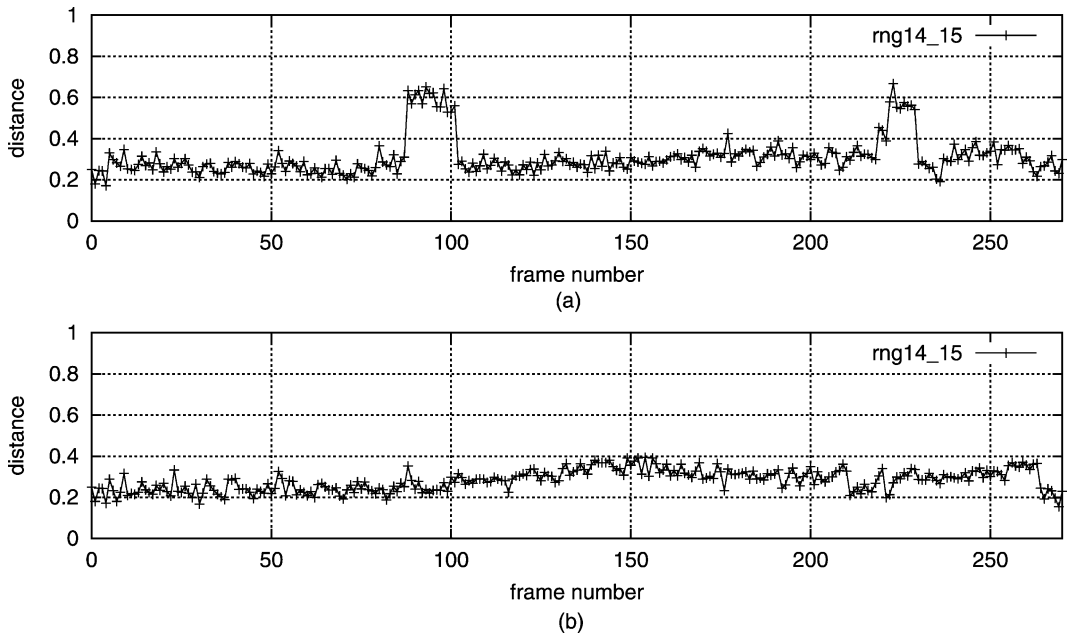

Fig. 8. Distribution of the distances, (a) computed before update model, (b) after model update.

method uses the pseudo perspective motion model given by:

$$u = a_1 + a_2x + a_3y + a_4xy + a_5x^2$$

$$v = a_6 + a_7x + a_8y + a_4y^2 + a_5xy$$

where $a_1 \ldots a_8$ are motion parameters, (x, y) is the position of a point in image space and (u, v) is the optical flow vector. Pseudo perspective motion model provides a better estimate of the motion for the planar scenes in closing FLIR sequences compared to simpler motion models such as the affine model, which fails to detect skew, pan and tilt in planar scenes. Rewriting the pseudo perspective motion of the sensor between two images in the matrix form we have:

$$\mathbf{U} = \mathbf{M}\mathbf{a} \tag{13}$$

where $\mathbf{U} = (\mathbf{u}, \mathbf{v})^{\mathrm{T}}$ is the optical flow, $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8)^T$ and

$$\mathbf{M} = \begin{pmatrix} 1 & x & y & xy & x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & y^2 & xy & 1 & x & y \end{pmatrix}$$

Optical flow can be computed using the optical flow constraint equation given by

$$\mathbf{F}_{\mathbf{X}}^{\mathrm{T}} U = -\mathbf{f}_{\mathrm{t}} \tag{14}$$

where $\mathbf{F}_{\mathbf{X}} = (\mathbf{f}_{\mathbf{x}}\mathbf{f}_{\mathbf{y}})^{\mathrm{T}}$ is the spatial gradient vector and $\mathbf{f}_{\mathrm{t}}$ is the temporal derivative. Combining Eqs. (13) and (14) results in a linear system that can be solved using the least squares method as

$$\left(\sum \mathbf{M}^{\mathbf{T}}\mathbf{F}_{\mathbf{X}}\mathbf{F}_{\mathbf{X}}^{\mathbf{T}}\mathbf{M}\right)\mathbf{a} = -\sum \mathbf{f}_{\mathbf{t}}\mathbf{M}^{\mathbf{T}}\mathbf{F}_{\mathbf{X}}. \tag{15}$$

In Fig. 10, we show two successive frames I_k and I_{k+1}, their difference $I_{k+1} - I_k$, the compensated global motion, i.e. first frame registered onto the second frame, I'_k using Eq. (15), and the difference $I'_k - I_{k+1}$. As it is clearly seen, the frames are correctly registered.

Compensating for sensor ego motion in images lacking adequate gradient information suffers from biased estimation of the motion parameters. Especially for FLIR imagery, background clutter and lack of texture increase the possibility of estimating incorrect parameters. Based on these observations, it is important not to perform motion compensation for every frame. Similar to the scheme described in Section 4.3, we compensate for the global motion if the distance computed using Eq. (5) is $d_k > m_k + 2\sigma_k$, where m_k is the mean and σ_k is the standard deviation of d_i for $i < k$. This scheme guarantees compensation for global motion if the target distribution changed drastically for a particular sequence, such that the tracker fails to locate the target.

Once the projection parameters, $\mathbf{a}$, are computed by solving the system of equation given in Eq. (13), we apply

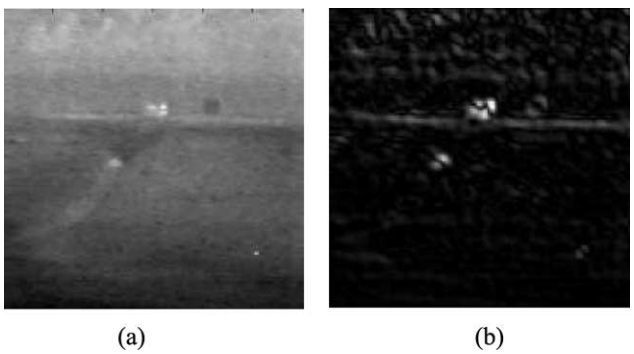

Fig. 9. (a) Sample frame from one of the FLIR sequences, (b) summation of four Gabor responses of the frame in (a).

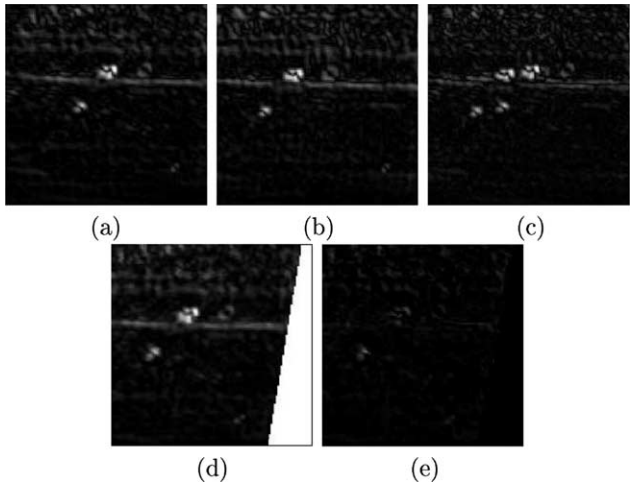

Fig. 10. (a) The reference frame I_k; (b) the current frame I_{k+1}; (c) the difference image obtained using (a) and (b), note the large bright spots due to miss registration; (d) first frame registered onto the second frame, i.e. global motion is compensated; (d) the difference image obtained using (b) and (d).

the transformation to the previous target center and compute the approximate new candidate target center using $\mathbf{m}_k = \mathbf{m}_{k-1} + \mathbf{U}$.

We then perform mean-shift iterations in the neighborhood of the new target position to increase the likelihood of the target model and the new candidate model.

In Section 4.5, we describe the complete algorithm to assist the reader in implementing the proposed detection and tracking method.

4.5. Algorithm

The complete algorithm is composed of two separate parts: the detection part, which is used once for the initialization of the system, and the tracking part. In the algorithm, we assume that the target first appeared at frame 0, and the current frame is k. The model distributions for intensity and standard deviation are denoted by Q_I and Q_S respectively. Similarly, we denote the candidate distributions for intensity and standard deviation for frame $k+1$ by P_I and P_S, respectively. The distance between model and candidate distributions is referred to as d_k.

4.5.1. Detection

1. Compute image intensity histogram for 0th frame and locate peaks and valleys in the histogram.
2. Partition intensity space using thresholds at valleys; intensity values close to valleys are considered ambiguous.
3. Perform fuzzy c-means clustering to assign ambiguous pixels to corresponding partitions and perform region merging based on the edge information to eliminate the false regions [11].
4. Apply Eq. (1)to compute confidence for target candidates and discard targets with small confidences.
5. Use steerable filters and compute the energy vector for each target candidate window, and the overlapping windows in their neighborhoods (see Fig. 3).
6. Compute texture similarity measure using Eq. (2) and discard targets that are similar to their neighborhood.

4.5.2. Tracking

For the detected targets, algorithm executes the following steps to perform tracking at frame k.

1. For the detected targets compute Q_I and Q_S using Eq. (4).
2. Initialize target center at frame k using the previous target center and compute distributions P_I and P_S.
3. Compute the modified mean-shift vector iteratively using Eq. (8) and (9).
4. Compute distance d_k (Eq. (5), and go to step 2 until the change of d_k at each iteration is close to 0.
5. If $d_k < m_{k-1} - 2\sigma_{k-1}$ update Q_I and Q_S else if $d_k > m_{k-1} + 2\sigma_{k-1}$ compensate for global motion.
6. Update the distance distribution parameters m_k and σ_k using Eq. (10) and (11) then return to step 2.

Since each detected target in the scene has its own distribution model, the above tracking algorithm also performs tracking under partial occlusion. We do not address complete occlusions in this paper.

Section 5 describes the experimental setup of the systems and the results obtained for various sequences.

5. Experiments

We have applied the proposed tracking method to the AMCOM FLIR data set. The data set was made available to us in grayscale format and has 41 sequences where each frame in each sequence is 128×128.

The proposed approach was developed using C++ on a Pentium III platform and the current implementation of the algorithm is capable of tracking at 10 fps. On all sequences, the detection algorithm is executed until a target is detected. For dark targets, we manually marked the target and performed tracking. We used 64 bins to construct distributions of the intensity and the standard deviation measures.

In this section, we show the robustness of the system using both the model update and motion compensation modules and present the results on sequences which have low SNR and high global motion. In the figures, rather than showing every 10th or 15th frame, we selected representative frames from these sequences to demonstrate motion of the targets. The positions of the targets in the sequences presented here are visually verified. For video sequences of

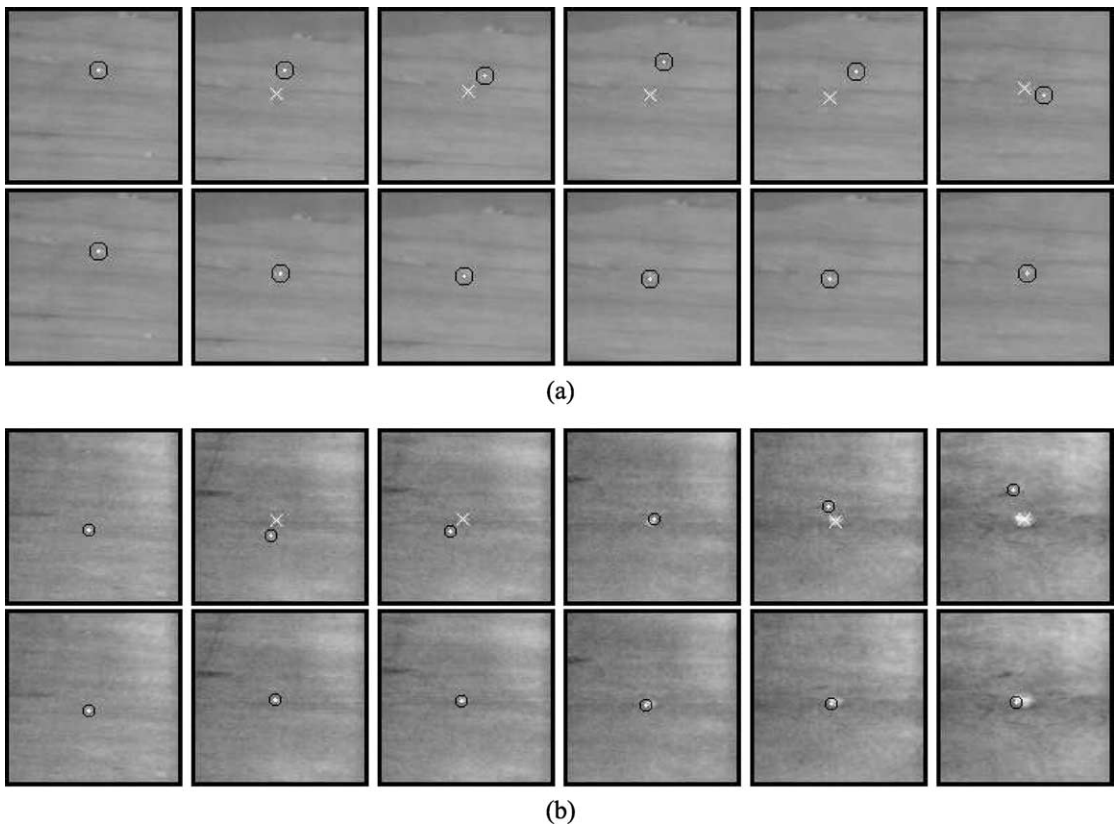

Fig. 11. Tracking results obtained by the proposed method on two sequences. In each part the first row shows results without both the global motion compensation and the model update. The second row shows the results with global motion compensation and the model update. (a) Sequence rng15_20, frames 0, 3, 10, 25, 44 and 54. (b) Sequence rng22_08, frames 0, 20, 50, 99, 169 and 236. The correct target positions are marked by ' × ', whereas the detected target positions are marked by circles.

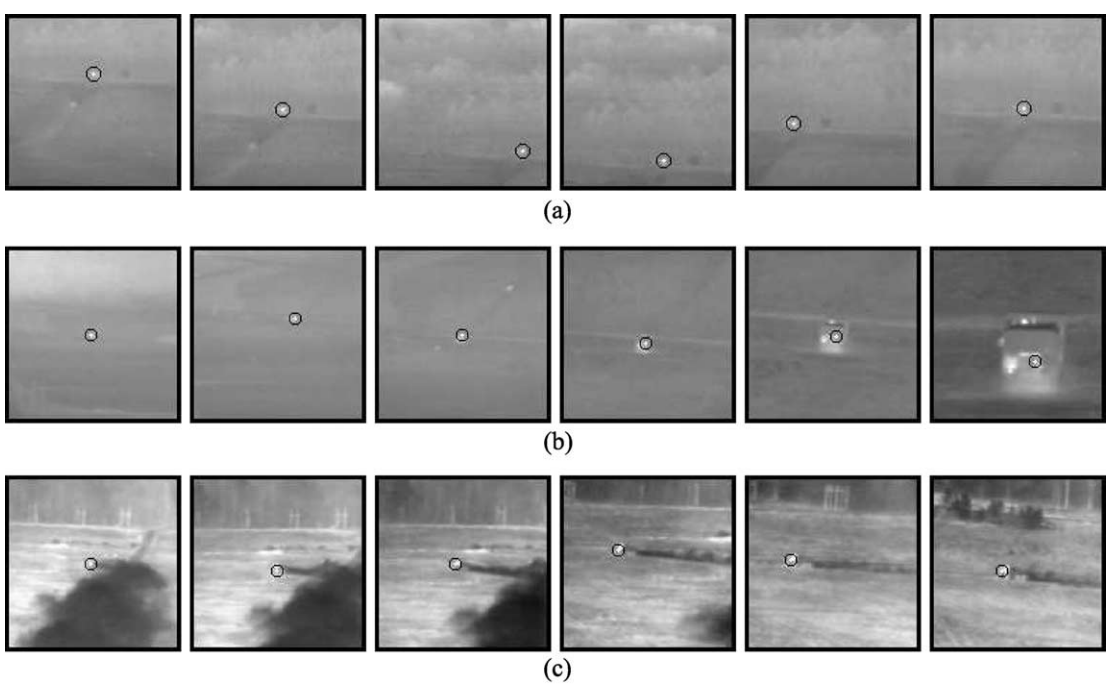

Fig. 12. Tracking results for various sequences. Sequence (a) rng17_01, frames 1, 17, 35, 53, 70 and 115, (b) rng14_15, frames 1, 60, 128, 195, 237 and 271, (c) rng19_07, frames 129, 138, 144, 159, 174 and 189.

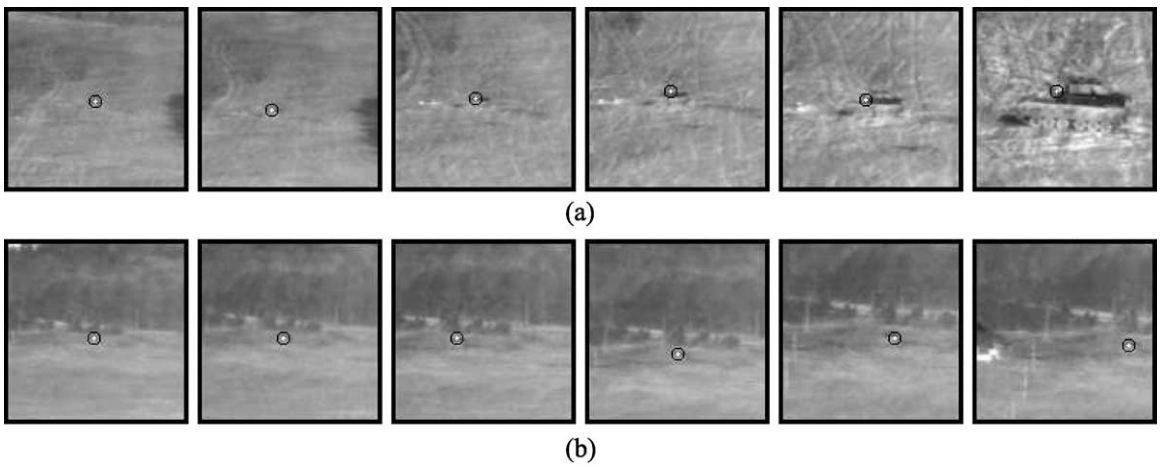

Fig. 13. Target tracking results for cold targets: (a) sequence rng18_07, frames 113, 135, 181, 204, 229 and 259; (b) sequence rng18_03, frames 11, 39, 63, 91, 119, 154 and 170.

tracking results please visit http://www.cs.ucf.edu/~vision/projects/MeanShift/MeanShift.html.

In Fig. 11, we present the results that demonstrate the importance of using global motion compensation and automatic model update for two sequences. In the figure, first rows of parts (a) and (b) show the tracking results where we neither update the target model nor compensate for the global motion. The correct target positions are marked by '×', whereas the detected target positions are marked by circles. Specifically in Fig. 11(a), the sensor ego-motion causes an abrupt change in the target position and the target is lost without global motion compensation; however as can be seen from the second row the target is correctly located using the global motion compensation module. In Fig. 11(b), due to changes in target distribution over time and low SNR of the target, (as it is clear from the first row) the tracker loses the target when it does not perform the model update. However, as shown in the second row the model update

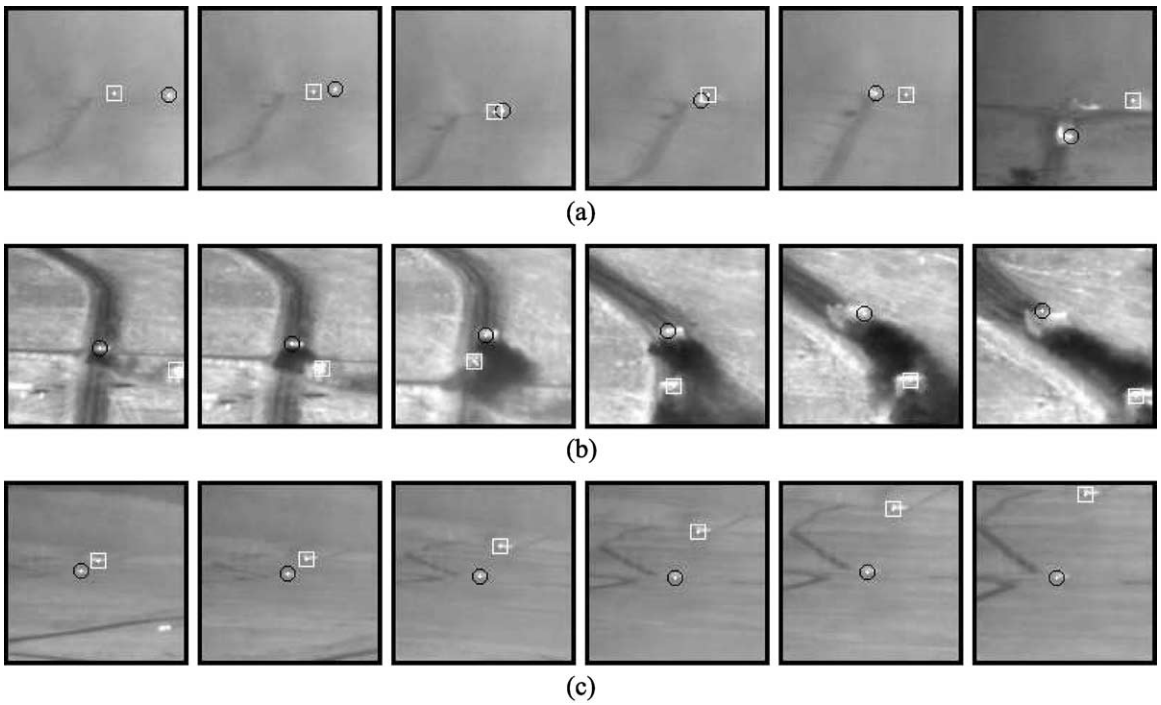

Fig. 14. Tracking results for multiple targets: (a) sequence rng16_07 frames 226, 241, 253, 274, 294 and 386; (b) sequence rng19_NS frames 208, 215, 231, 253, 267 and 274; (c) sequence rng16_18 frames 1, 20, 40, 60, 79 and 99.

improves the performance and the target is correctly tracked.

In Fig. 12, tracking results for sequences *rng17_01* (a), *rng14_15* (b) and *rng19_07* (c) are shown. The tracked target positions are marked by circles. In all three sequences, despite the fact that the targets look very similar to their backgrounds, their positions are correctly located.

In Fig. 13, tracking results for cold targets are shown. Since the detection method only detects ithot targets, the sizes and positions of the targets are manually initialized. In both sequences rng18_07 (part (a)) and rng18_03 (part (b)), the targets have very low contrast with the background and particularly in part (b) neighboring locations hide the target due to high gradient magnitude. Using both the intensity and the local standard deviation measures together improved the tracking performance for both sequences. In particular, for sequence rng18_07 the system was able to track the target successfully even in presence of a very high change in the intensity levels.

To demonstrate multiple target tracking capability, we also applied the tracking method to sequences with multiple targets. In Fig. 14, tracking results are presented for sequences (a) rng16_07, (b) rng19_NS and (c) rng16_18. As seen from the results, the target are correctly tracked. We do not address complete occlusion problems in this paper.

6. Conclusions

We propose a robust approach for tracking targets in airborne FLIR imagery. The tracking method requires the position and the size of the target in the first frame. The target detection scheme, which is used to initialize the tracking algorithm, finds target candidates using fuzzy clustering, edge fusion and texture measures. We employ a texture analysis on these candidates to select the correct targets. The experimental results for 200 target and 200 non-target regions that were manually marked show that 'steerable filters' have better categorization performance compared to 'Law's texture measures' and 'wavelet filters'. Once the targets are detected, the tracking system tracks the targets by finding the translation of the target center in the image space using the intensity and local standard deviation distributions. According to the distribution of the distance calculated from target model and distributions of the new target center, the algorithm decides whether a model update or global motion compensation is necessary. Sensor ego-motion is compensated for two consecutive frames assuming the pseudo-perspective motion model. The results demonstrated on sequences, which have low SNR and high ego motion, show the robustness of the proposed approach for tracking FLIR targets.

Acknowledgements

The authors wish to thank to Richard Sims of US Army AMCOM for providing us FLIR sequences. This research was partially funded by a grant from Lockheed Martin Corporation.

References

[1] J.Y. Chen, I.S. Reed, A detection algorithm for optical targets in clutter, IEEE Transactions on Aerospace and Electronic Systems 23 (1) (1987) 46–59.
[2] M.S. Longmire, E.H. Takken, Lms and matched digital filters for optical clutter suppression, Applied Optics 27 (6) (1988) 1141–1159.
[3] H. Shekarforoush, R. Chellappa, A multi-fractal formalism for stabilization, object detection and tracking in flir sequences, In: IEEE International Conference on Image Processing, vol. 3, 2000.
[4] D. Davies, P. Palmer, Mirmehdi, Detection and tracking of very small low contrast objects, In: Ninth British Machine Vision Conference, September, 1998.
[5] A. Strehl, J.K. Aggarwal, Detecting moving objects in airborne forward looking infra-red sequences, Machine Vision Applications Journal 11 (2000) 267–276.
[6] U. Braga-Neto, J. Goutsias, Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators, In: 33rd Conference of Information Sciences and Systems, March, 1999.
[7] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 142–149.
[8] J.R. Bergen, P. Anandan, K.J. Hanna, R. Hingorani, Hierarchical model-based motion estimation, In: Europen Conference on Computer Vision, 1992, pp. 237–252.
[9] E.T. Lim, S.D. Deshpande, C.W. Chan, R. Venkateswarlu, Adaptive spatial filtering techniques for the detection of targets in infrared imaging seekers, In: Proceedings of SPIE, vol. 4025, 2000, pp. 194–202.
[10] A.P. Tzannes, D.H. Brooks, Detection of point targets in image sequences by hypothesis testing: a temporal test first approach, In: Proceedings of ICASSP, 1999.
[11] Y.W. Lim, S.U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, Pattern Recognition Journal 23 (9) (1990) 935–952.
[12] E. Saber, A.M. Tekalp, G. Bozdagi, Fusion of color and edge information for improved segmentation and edge linking, Image and Vision Computing Journal 15 (1997) 935–952.
[13] G. Strang, T. Nguyen, Wavelets and Filter Banks, Cambridge Press, Wellesley, 1996.
[14] K.I. Laws, Textured image segmentation, PhD Thesis, Electrical Eng., University of Southern California, January 1980.
[15] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7) (1989) 674–693.
[16] W.T. Freeman, E.H. Adelson, The design and use of steerable filters, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (9) (1991) 891–906.
[17] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, Robust online appereance models for visual tracking, In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 415–422.
[18] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[19] A. Yilmaz, M. Shah, Automatic feature detection and pose recovery for faces, In: Asian Conference on Computer Vision, January, 2002, pp. 284–289.
[20] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1995) 790–799.
[21] K. Fukunaga, L.D. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Transactions on Information Theory IT-21 (1975) 32–40.
[22] A. Yilmaz, K.H. Shafique, N. Lobo, X. Li, T. Olson, M. Shah, Target-tracking in flir imagery using mean-shift and global motion compensation, In: IEEE CVPR Workshop on Computer Vision Beyond Visible Spectrum, 2001.
[23] John. G. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, Optical Society of America 2 (1985) 1160–1169.
[24] R.N. Braithwaite, B. Bhanu, Hierarchical gabor filters for object detection in infrared images, In: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 628–631.
[25] M. Irani, P. Anandan, Video indexing based on mosaic representations, Proceedings of IEEE 86 (5) (1998) 905–921.