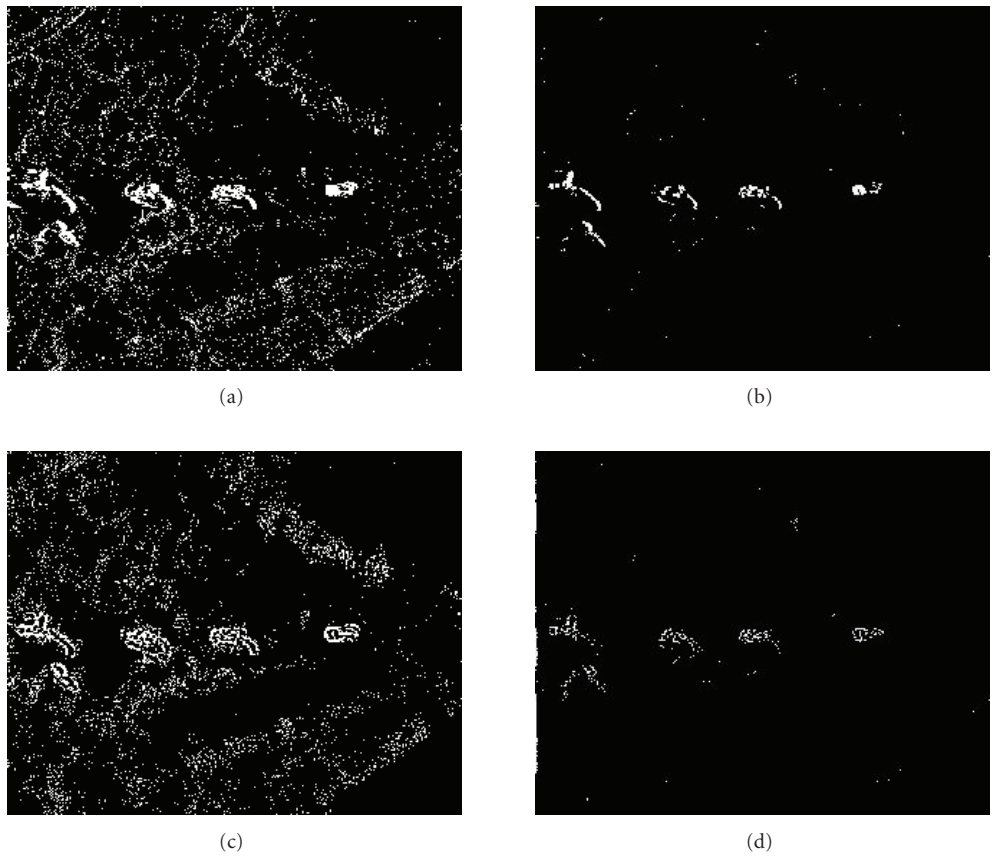


(a) (b) (c) (d)

Figure 8: Binary images generated by applying different thresholding algorithms to frame difference image in Figure 6(d); (a) Fixed threshold = 10; (b) Fixed threshold = 30; (c) Adaptive thresholding by using *mean-C* with a 5×5 window and C is set at 5.0; (d) Gaussian adaptive thresholding with a 5×5 window and C is set at 10.0.

without knowledge about the source of degradation. Many different, often elementary and heuristic methods are used to improve images in some sense. A literature survey is given in [26]. Advanced image enhancement algorithms employ spatial filter, neural network, cellular neural network, and fuzzy filter. However, these methods are computationally heavy. They are not suitable for real-time target detection. In our algorithm, we employ dynamic Gabor filter.

3.2.2. Dynamic Gabor Filter. Gabor function has been recognized as a very useful tool in computer vision and image processing, especially for texture analysis, due to its optimal localization properties in both spatial and frequency domain. There are many publications on its applications since Gabor proposed the 1D Gabor function [27]. The family of 2D Gabor filters was originally presented by Daugman [28] as a framework for understanding the orientation-selective and spatial–frequency-selective receptive field properties of neurons in the brain's visual cortex, and then was further mathematically elaborated [29]. The 2D Gabor function is a harmonic oscillator, composed of a sinusoidal plane wave of a particular frequency and orientation, within a Gaussian envelope. Gabor wavelets are hierarchically arranged, Gaussian-modulated sinusoids. The Gabor-wavelet transform of a two-dimensional visual field generates a four-dimensional field: two of the dimensions are spatial, the other two represent spatial frequency and orientation. A Gabor wavelet is defined as

$$\psi_{\mu,\nu}(z) = \frac{\left\|k_{\mu,\nu}\right\|^2}{\sigma^2} e^{-\|k_{\mu,\nu}\|^2 \times \|z\|^2 / 2\sigma^2} \left[e^{ik_{\mu,\nu} z} - e^{-\sigma^2/2} \right], \quad (18)$$

where $z = (x, y)$ is the point with the horizontal coordinate x and the vertical coordinate y. The parameters μ and ν define the orientation and scale of the Gabor kernel, $\|\cdot\|$ denotes the norm operator, and σ is related to the standard derivation of the Gaussian window in the kernel and determines the ratio of the Gaussian window width to the wavelength. The wave vector $k_{\mu,\nu}$ is defined as follows

$$k_{\mu,\nu} = k_\nu e^{i\phi_\mu}, \quad (19)$$

where $k_\nu = k_{\max}/f^\nu$ and $\phi_\mu = \pi\mu/8$, $k_{\max}$ the maximum frequency, and f^ν is the spatial frequency between kernels in frequency domain.

The Gabor kernels in (18) are all self-similar since they can be generated from one kernel (a mother wavelet) by dilation and rotation via the wave vector $k_{\mu,\nu}$. Each kernel is a product of a Gaussian envelope and a complex plane wave. The first term $e^{ik_{\mu,\nu} z}$ in the square bracket in (18) controls the oscillatory part of the kernel and the second term $e^{-\sigma^2/2}$

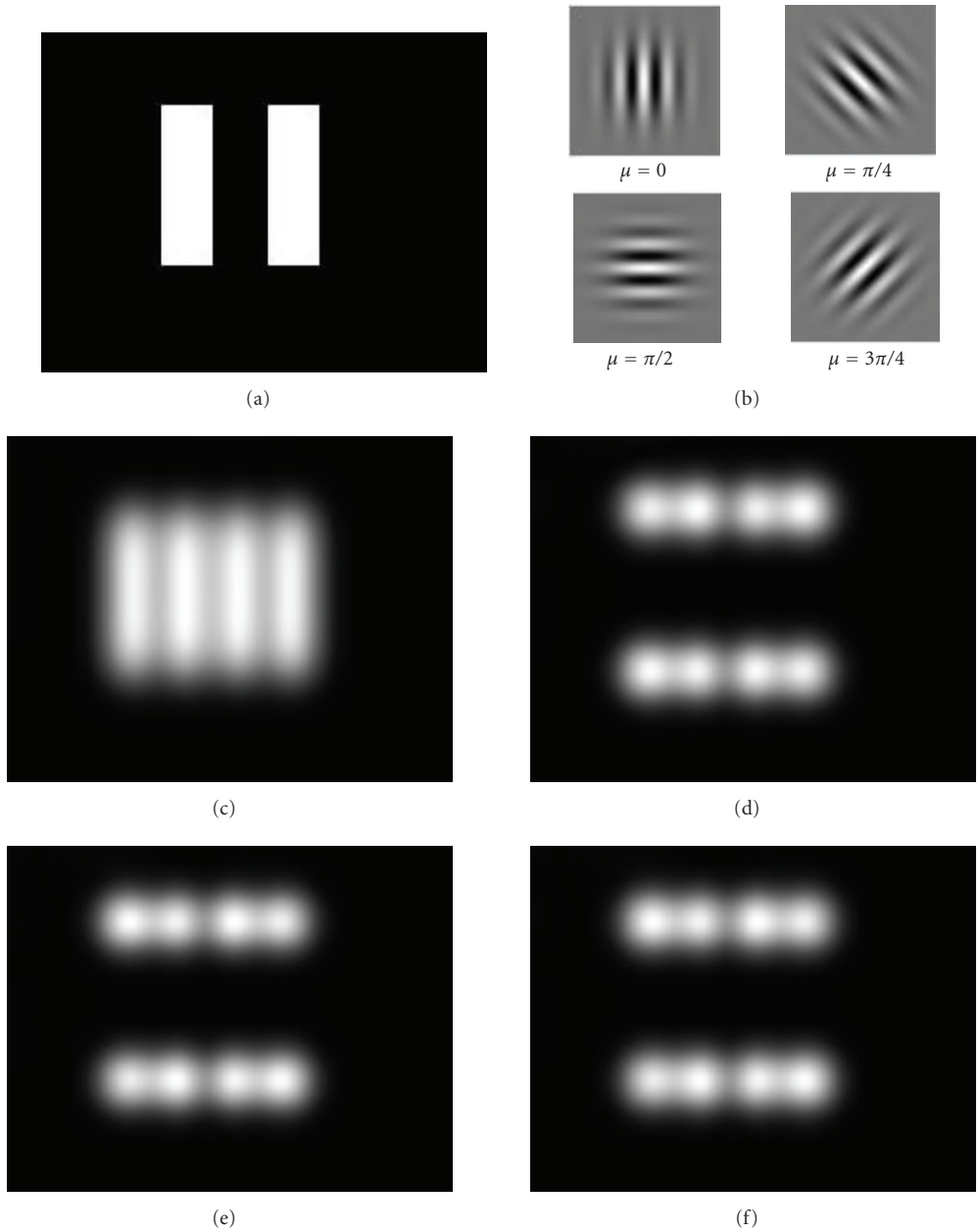


Figure 9: Gabor kernels and Gabor filter responses. (a) Input image; (b) 4 Gabor kernels with $\nu = 3$ and $\sigma = 2\pi$; (c), (d), (e), and (f) Gabor filter response with a Gabor kernel at orientation $\mu = 0$, $\pi/4$, $\pi/2$, and $3\pi/4$, respectively.

compensates for the DC value, thus making the kernel DC-free, that is, the integral $\int \psi_{\mu,\nu}(z)d^2z$ vanishes. Therefore, it is not necessary to consider the DC effect, when the parameter σ is large enough.

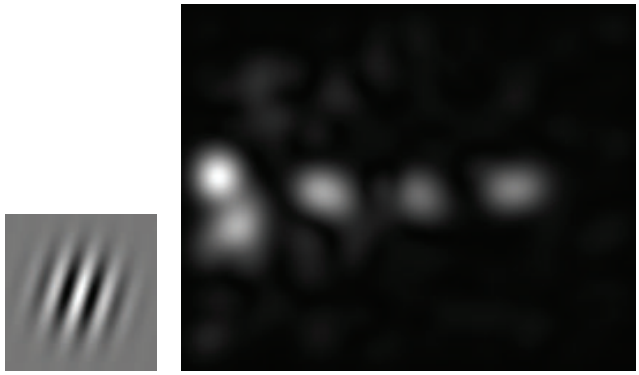
The Gabor filtering of an image I is the convolution of the image I with a Gabor kernel as defined by (18). The convolution image is defined as

$$O_{\mu,\nu}(z) = I(z) * \psi_{\mu,\nu}(z). \tag{20}$$

The response $O_{\mu,\nu}(z)$ to the Gabor kernel $\psi_{\mu,\nu}(z)$ is a complex function with a real part $\mathrm{Re}\{O_{\mu,\nu}(z)\}$ and an imaginary part $\mathrm{Im}\{O_{\mu,\nu}(z)\}$. The magnitude response $\|O_{\mu,\nu}(z)\|$ is expressed as

$$\left\|O_{\mu,\nu}(z)\right\| = \sqrt{\mathrm{Re}\left\{O_{\mu,\nu}(z)\right\}^2 + \mathrm{Im}\left\{O_{\mu,\nu}(z)\right\}^2}. \tag{21}$$

Figure 9(a) shows a synthesized binary image. Figure 9(b) shows four Gabor kernels with $\nu = 3$ and $\sigma = 2\pi$, at orientation $\mu = 0$, $\pi/4$, $\pi/2$, and $3\pi/4$, respectively. The Gabor filter responses are shown in (c), (d), (e), and (f), corresponding to the Gabor kernel at orientation 0, $\pi/4$, $\pi/2$, and $3\pi/4$, accordingly. Here, the interesting result is shown in (c), where the disconnected blobs in (a) are merged into one blob after Gabor filtering. The similar phenomenon



(a) (b)

Figure 10: (a) Dynamic Gabor kernel determined by the optical flow in Figure 6(c); (b) Gabor filter response for the frame difference image in Figure 6(d).

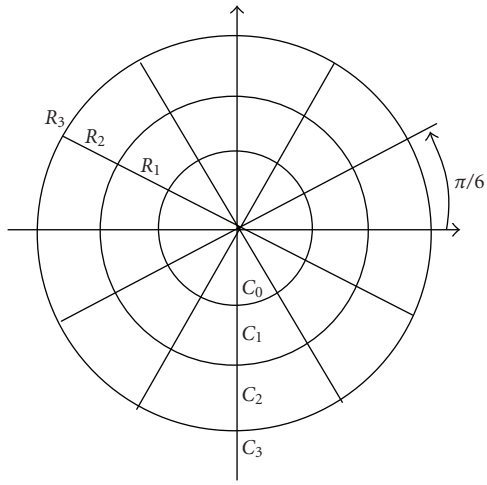


Figure 11: Specular highlight detector.

happens in target detection by frame differencing technique. If the interval between two consecutive frames is too large or if the targets move too fast, the moving targets appear as separate blobs in frame difference image. By carefully choosing the orientation of Gabor filter, separated blobs can be detected as a connected blob from Gabor response. Our algorithm employs this experiment result.

In our algorithm, we fix the following parameters, $k_{\max} = \pi/2$, $\sigma = 2\pi$, $f = \sqrt{2}$, and $\nu = 3$. The orientation μ is dynamically changed according to optical flows from inliers. We call it *dynamic Gabor filter*. The orientation μ is defined as

$$\mu = \frac{1}{K_{\text{in}}} \sum_{i=1}^{K_{\text{in}}} \theta\left(\vec{F}_i^{t't}\right), \tag{22}$$

where $\theta(\vec{F}_i^{t't})$ is the orientation of the optical flow $\vec{F}_i^{t't} \in F_{\text{in}}^{t't}$, and is given by

$$\theta\left(\vec{F}_i^{t't}\right) = \arctan \frac{y_i^{t'} - y_i^t}{x_i^{t'} - x_i^t}. \tag{23}$$

Figure 10(a) shows the dynamic Gabor kernel determined by the optical flows in $F_{\text{in}}^{t't}$ as shown in Figure 6(c). Figure 10(b) shows the Gabor filter response by performing convolution for the frame difference image in Figure 6(d) and the dynamic Gabor kernel in Figure 10(a).

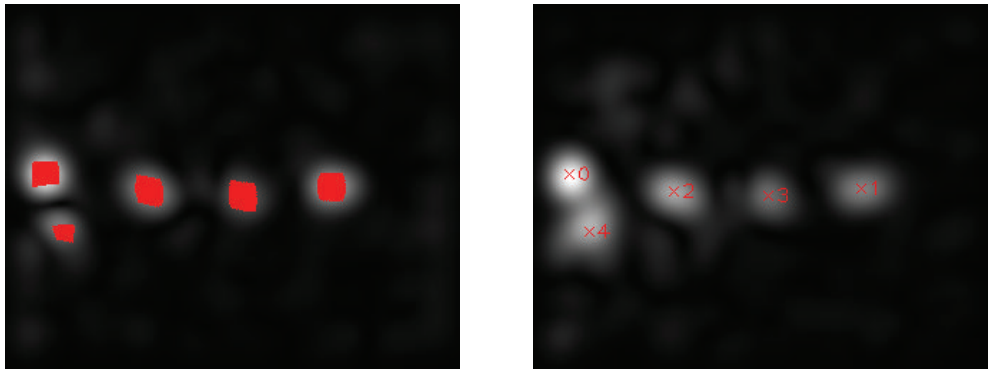
3.3. Specular Highlights Detection. As can be seen in Figure 10(b), the image changes appear as high intensity in the dynamic Gabor filter response. They look like spotlights. The center of the spotlight is brightest, and the brightness on the circular points around the center becomes dim gradually when the circle becomes larger. We call these high intensity *specular highlights*. Therefore, the target detection problem becomes the specular highlight detection problem. Because the intensity of highlights changes for the moving targets (some specular highlights are dimmer than others), the thresholding algorithms cannot detect all specular highlights successfully. Here, we employ the specular highlight detector as shown in Figure 11. The C_0 is the pixel under examination. This detector compares the intensity at C_0 and the intensity of pixels on the circular circles C_1, C_2, and C_3, with radius R_1, R_2, and R_3, respectively. C_1, C_2, and C_3 are sampled at $\pi/6$ interval, hence the detector will only compare the intensity at C_0 and 12 sample points, $C_{j,1}$, $C_{j,2}, \ldots, C_{j,12}$, from each circular circle. Let $G(z)$ denote the dynamic Gabor filter response at z, the discrimination of specular highlights is as follows

$$C_0 \text{ is} \begin{cases} \text{a specular highlight,} \\ \quad \text{iff } G(C_0) \geq G(C_{1,i}) \text{ and } G\left(C_{j,i}\right) \geq G\left(C_{j+1,i}\right), \\ \text{not a specular highlight,} \\ \quad \text{otherwise,} \end{cases} \tag{24}$$

where $j = 1, 2$, and $i = 1, 2, \ldots, 12$.

The specular highlight points detected from the dynamic Gabor filter response in Figure 10(b) are shown in Figure 12(a) by red dots. Note that red dots form several red regions in Figure 12(a). This is caused by the loose condition, " if and only if $G(C_0) \geq G(C_{1,i})$ and $G(C_{j,i}) \geq G(C_{j+1,i})$", in (24). The loose condition is chosen in attempt not to miss the possible specular highlights. These specular highlight points are denoted by $P^h = \{p_1^h, \ldots, p_{K_h}^h\}$, where K_h is the number of specular highlight points. In our algorithm, it is convenient to use the center and radius to represent the location and size of the specular highlights. To obtain the location and the size of specular highlights, $\{p_1^h, \ldots, p_{K_h}^h\}$ are clustered. Let $H_i(c, r)$ denote ith specular highlight, where r is the radius, c the center, and c contains x-coordinate, x_c, and y-coordinate, y_c.

The specular highlights generated above need to be clustered to determine the precise center of the specular spot. Among the clustering algorithms, k-NN (k nearest neighbor) algorithm needs a user predetermined constant k the number of the clusters [30]. It is not applicable to



(a) (b)

Figure 12: (a) Specular highlight points; (b) Specular highlight clustering.

our problem. On the other hand, the mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters [31]. However, the computation is complicated. Similarly, (support vector machine) SVM is another powerful clustering algorithm [32], but it is computationally heavy. In this work, $H_i(c,r)$ is obtained according to the following algorithm.

Specular Highlight Point Clustering Algorithm. The summary of this algorithm is as follows. For a specular highlight point, create a new cluster and consider this point is the center of the newly created cluster. Then, check whether there are other specular highlight points that are close to the current one, according to the predetermined threshold T_h. If yes, those points are also added to the newly created cluster, and the center of the cluster is updated after adding a specular highlight point to the newly created cluster. This process is repeated for all specular highlight points. After this processing, it forms a cluster. Then it chooses the next specular highlight point that is not clustered so far, and repeats the above processing. This processing is repeated until all specular highlight points are clustered. The details are given below.

(1) For $p_j^h \in P^h$, it is considered as the center of $H_i(c,r)$ and it is removed from P^h, added to $H_i(c,r)$, and set $c = p_j^h$ and $M_i = 1$, where M_i is the number of the specular highlight points in $H_i(c,r)$, and both i and j begin from 0, and $H_i(c,r)$ is an empty set initially.

(2) For $p_k^h \in P^h (k \neq j)$, if $\|p_k^h - c\| \leq T_h$, p_k^h is removed from P^h, added to $H_i(c,r)$, and update M_i and the center c according to

$$M_i = M_i + 1, \qquad x_c = \frac{1}{M_i}\sum_{m=1}^{M_i} x_m, \qquad y_c = \frac{1}{M_i}\sum_{m=1}^{M_i} y_m, \tag{25}$$

where T_h is a predetermined threshold value, $(x_m, y_m) \in H_i(c,r)$, (x_c, y_c) is the coordinates of the center c, and $\|p_k^h - c\|$ means the Euclidean distance between the specular highlight point, p_k^h, and the center c.

(3) Repeat step (2) for all specular highlight points in P^h. When this step finishes, $H_i(c,r)$ is obtained, and the radius r is given by

$$r = \max\left\|p_k^h - c\right\|, \tag{26}$$

where $k = 1, 2, \ldots, M_i$,

(4) Update i, and repeat steps (1) to (3) for the left specular highlight points in P^h to search for the next cluster.

(5) Repeat steps (1) to (4) until P^h becomes an empty set.

Let $H_S = \{H_1(c,r), H_2(c,r), \ldots, H_{Ks}(c,r)\}$ represent the detected specular highlights, where K_s is the number of specular highlights. Figure 12(b) shows the clustering result for the specular highlight points in Figure 12(a), where each cluster means a specular highlight. The specular highlights are numbered from 0 to 4, and the centers are marked by a small "x".

3.4. Moving Target Localization

3.4.1. Outlier Clustering. Because outliers are caused by the moving targets, they can be used for moving target localization. Here we employ the observation result that if outliers belong to the same moving targets, they are located closely, in optical flow field. Therefore, the outliers are clustered first. The clustering algorithm for outliers is the same one as described in Section 3.3, but with different clustering threshold T_o. Let $C_{\text{out}} = \{C_1(c,r), C_2(c,r), \ldots, C_{K_o}(c,r)\}$ represent the outliers clusters, where K_o is the number of the clusters. The outlier clustering result for the outliers detected from input images in Figures 6(a) and 6(b) is shown in Figure 6(c) by the purple circles, and the center of each cluster is marked by small "+" in purple. If all outliers are separated correctly, we can say that each cluster corresponds to one or multiple targets. However, this assumption is not always correct. Some moving target may not generate outliers

because outlier separation algorithm may fail or because the displacement of moving target is too small. This case is indicated in Figure 6(c) by the dotted circle in red, where a moving target exists. In the following, we combine both outlier clustering result and specular highlight detection result for moving target localization.

3.4.2. Moving Target Localization Based on Outlier Clustering and Specular Highlights. The discrimination rule for moving target localization based on outlier clustering and specular highlight detection is as follows. For a specular highlight $H_i(c,r) \in H_S$, if its center lies in a outlier cluster $C_k(c,r) \in C_{\text{out}}$ $(i = 1,\ldots,K_s, k = 1,\ldots,K_o)$, it is considered as a target. If its center does not lie in any outlier cluster, the dynamic Gaussian detector is employed, which is described in Section 3.4.3. According to this rule, the specular highlight numbers 0, 1, 3, and 4 in Figure 12(b) are identified as moving targets, and are marked by red circles in Figure 13. The localized targets are represented by its center and radius which is set at T_o (the thresholding for outliers clustering).

3.4.3. Moving Target Localization Based on Dynamic Gaussian Detector. As shown in Figure 12(b), a specular highlight is similar to a two-dimensional (2-D) Gaussian distribution. The moving target localization method described in Section 3.4.2 may fail if the feature point detector, described in Section 3.1.1, does not detect the enough outliers belonging to a moving target. To make the moving target localization robust, we further employ 2-D Gaussian function as a target detector to conduct the secondary moving target localization. (Correspondingly, the method used in Section 3.4.2 is called primary moving target localization.) A general 2-D Gaussian function is given by

$$G(x,y) = Ae^{-[a(x-x_0)^2+b(x-x_0)(y-y_0)+c(y-y_0)^2]}, \tag{27}$$

where

$$\begin{aligned} a &= \left(\frac{\cos\theta}{\sigma_x}\right)^2 + \left(\frac{\sin\theta}{\sigma_y}\right)^2, \\ b &= -\frac{\sin 2\theta}{\sigma_x^2} + \frac{\sin 2\theta}{\sigma_y^2}, \\ c &= \left(\frac{\sin\theta}{\sigma_x}\right)^2 + \left(\frac{\cos\theta}{\sigma_y}\right)^2 \end{aligned} \tag{28}$$

and the coefficient A is the amplitude, (x_0, y_0) is the center, σ_x, σ_y are the x and y spreads of the Gaussian function, and θ is the orientation. Figure 14 shows 2D Gaussian function distribution at orientation $\theta = 0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6$, respectively.

In our algorithm, the detector compares the specular highlight with 2D Gaussian kernel generated according to (27) and (28), and calculates the similarity. The orientation θ of 2-D Gaussian function is determined by the orientation of the specular highlight. Here we call it *dynamic Gaussian detector*. This detector algorithm is as follows.

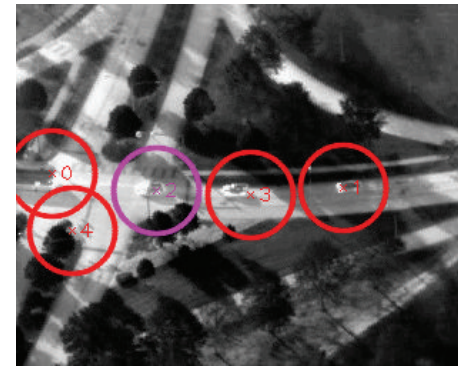


Figure 13: Target localization result.

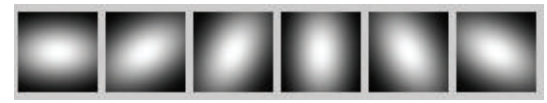


Figure 14: Gaussian kernel at orientation $\theta = 0, \pi/6, \pi/3, \pi/2, 2\pi/3$, and $5\pi/6$, respectively.

Target Localization Algorithm based on Dynamic Gaussian Detector. (1) For $H_i(c,r) \in H_S$ which does not lie in any outlier cluster in C_{out}, extract $W \times W$ image I_{sub} centered at c for this specular highlight, where W is determined by r, and currently is set at $2 \times 1.2 \times r + 1$.

(2) I_{sub} is binarized by fixed threshold, $0.7v_{\max}$, where $v_{\max}$ is the maximal intensity in I_{sub}.

(3) The first principal axis of the binarized image I_{sub} is calculated according to

$$\alpha = \frac{1}{2}\arctan\frac{2m_{11}}{m_{20}-m_{02}}, \tag{29}$$

where

$$m_{pq} = \sum_{x=1}^{W}\sum_{y=1}^{W} I_{\text{sub}}(x,y)(x-x_c)^p(y-y_c)^q \quad (p,q = 1,1;\ 2,0;\ 0,2) \tag{30}$$

is the moment around the centroid (x_c, y_c). x_c and y_c are given by

$$x_c = \frac{m_{10}}{m_{00}}, \qquad y_c = \frac{m_{01}}{m_{00}}, \tag{31}$$

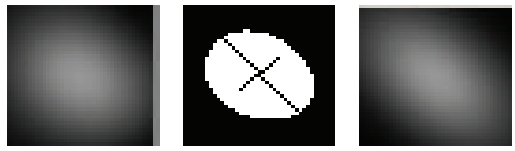
where

$$m_{p'q'} = \sum_{x=1}^{W}\sum_{y=1}^{W} I_{\text{sub}}(x,y)x^{p'}y^{q'} \quad (p',q' = 0,0;\ 1,0;\ 0,12). \tag{32}$$

Note that m_{pq} and $m_{p'q'}$ are the *moment* of order $(p+q)$ for the image I_{sub}, around the center c and origin, respectively. Equations (30) and (31) are the digital expression of the moment. Generally, for a 2D continuous function $f(x,y)$ the moment (sometimes called "raw moment") of

Table 2: Correct detection rate, miss detection rate, and hit rate for the 4 datasets.

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Total number of targets	381	266	287	297
Detected targets	326	221	249	270
Missed targets	55	45	38	27
Correct detection rate	85.6%	83.1%	86.8%	90.90%
Miss detection rate	14.4%	16.9%	13.2%	9.10%
Hit rate	85.9%	81.3%	70.7%	76.60%



(a) (b) (c)

Figure 15: (a) A specular highlight; (b) Principal axis for the specular highlight in (a); (c) Generated Gaussian pattern.

order $(p + q)$ is defined by $m_{pq} = \iint_{-\infty}^{\infty} x^p y^q f(x, y)dxdy$, where $p, q = 0, 1, 2, \ldots$.

(4) α is used as the orientation to generate Gaussian kernel I_G, according to (27), where the Gaussian pattern size is W.

(5) The similarity between I_{sub} and I_G is calculated according to (12) [33], which is rewritten as

$$s = \sum_{k=0}^{W-1} \sum_{k=0}^{W-1} \frac{\left[I_{\text{sub}}(k,l) - \bar{I}_{\text{sub}}\right] \times \left[I_G(k,l) - \bar{I}_G\right]}{W^2 \sqrt{\sigma(I_{\text{sub}}) \times \sigma(I_G)}}. \quad (33)$$

If $s \geq T_G$, $H_i(c, r)$ is considered as a target, where T_G is the predetermined threshold.

(6) Repeat steps (1) to (5) for all specular highlights in H_S, which do not lie in any cluster in C_{out}.

Figure 15(a) shows the image I_{sub} for the specular highlight number 2 in Figure 12(b), which does not lie in any outlier cluster in Figure 6(c). Figure 15(b) shows the binarized specular highlight and the first principal axis by a long black line segment, and the second principal axis by short, and (c) shows the generated Gaussian kernel according to (27).

4. Experiment Results

The entire algorithm described in Section 3 is implemented by using C++ and OpenCV on windows platform. The input image size is 320×256, Δ is set at 2, the outlier clustering threshold T_o at H/6 (H is the image height), the specular highlight point clustering threshold T_h at $2T_o/3$, the similarity threshold T_G at 0.93, and A, σ_x, and σ_y are set at 1, 25.0, and 15.0, respectively. The IR video data from the VIVID datasets provided by the Air Force Research Laboratory is used. Figures 16, 17, and 18 show some experiment results. Figures 16(a) and 16(b) show two consecutive input images, (c) shows the detected optical flows (marked by red line segments) and outlier clustering (marked by purple circles), (d) the generated frame difference, (e) the detected specular highlights, and (f) the detected moving targets marked by red circles.

Figure 17 shows the target detection results at frame 29, 32, 37, 69, 78, and 82, for an input image sequence. Green circles mark the ground truth target positions, labeled manually, red circles means targets detected based on outlier clustering and specular highlights, and purple circles marks the output of the dynamic Gaussian detector. In frame 32, the target number 3 in (a) is missed. In frame 37, the target number 3 in (a) is also missed, and the dynamic Gaussian detector mistakenly detected a specular highlight (marked by purple circle) caused by tree leaves. In frame 69, the system also mistakenly detected a specular highlight caused by tree leaves. However, the system detected a moving target (number 2 in (d)) that was not marked by the human operator. In Frame 78, the system also detected a moving target (number 2 in (e)) which is the ground truth target but is not marked by the human operator. This is a human operator's mistake. In frame 81, the system mistakenly detected a target (number 0 in (f)) and lost one target.

Figure 18 shows target detection results at frame 44, 50, 53, 73, 81, and 84 for another input image sequence. Green circles mark the ground truth target positions, labeled manually, red circles means targets detected based on outlier clustering and specular highlights, and purple circles marks the output of the dynamic Gaussian detector. In frame 44, the dynamic Gaussian detector identified two targets, number 2 and 3, in (a). However, the target number 3 is a false target. In frame 53, the target in the middle was detected as two separated targets. In frame 81 and 84, the system lost one target.

5. Performance Analysis

To evaluate the performance of this algorithm, we selected four image sequences with the significant background as the test data. Each sequence contains 100 frames, and each frame contains two to four moving targets. The ground truth targets are labelled manually. The total number of targets in these 4 datasets is 1231. We examined the correct detection rate, hit rate, and processing time. The hit rate is defined as the ratio for the intersected area of detected target and ground truth target and the area of the ground truth target. The experiments are conducted on a Windows Vista machine mounted with a 2.33 GHz Intel Core 2 CPU and 2 GB main memory. The total average correct detection rate is 86.6%,

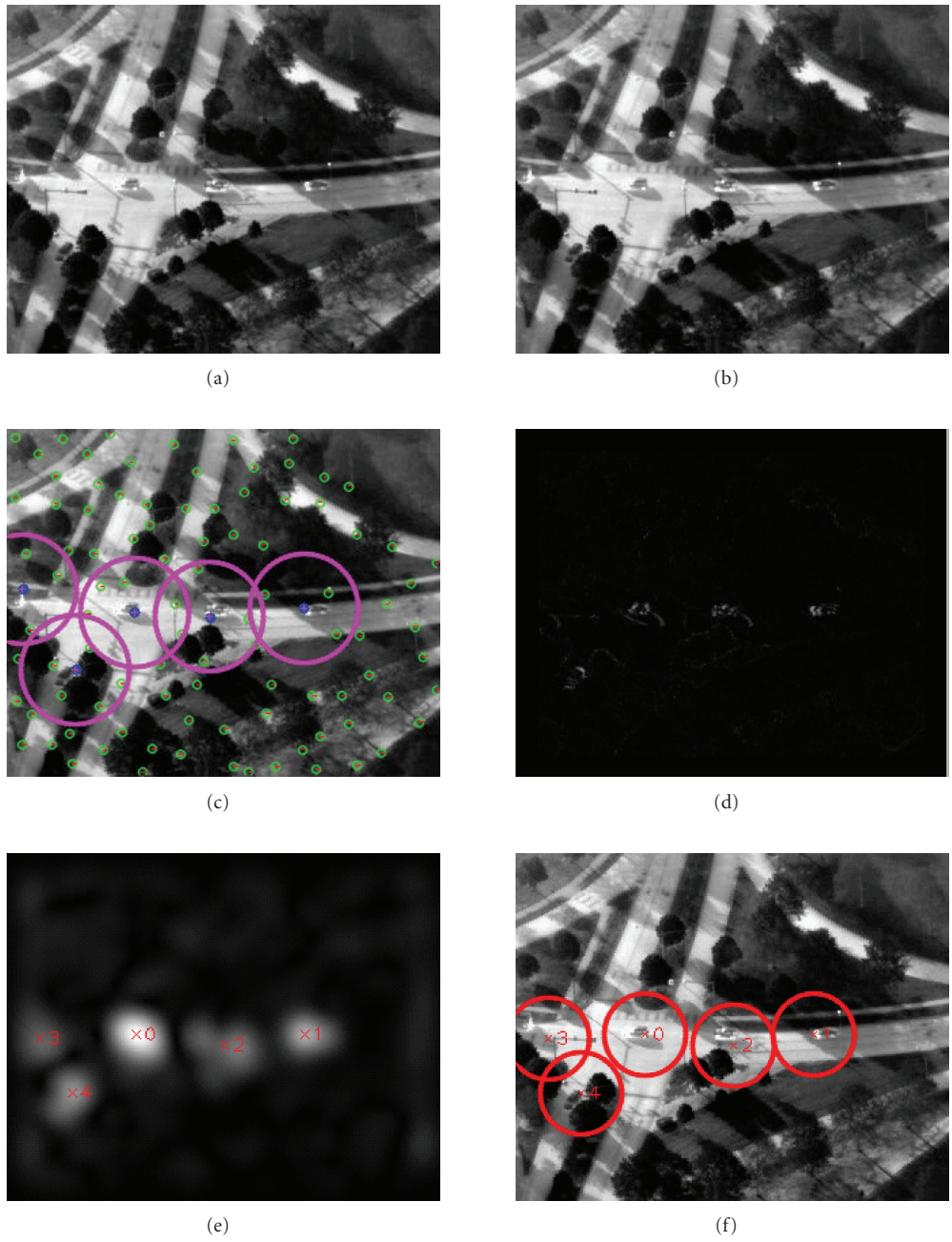
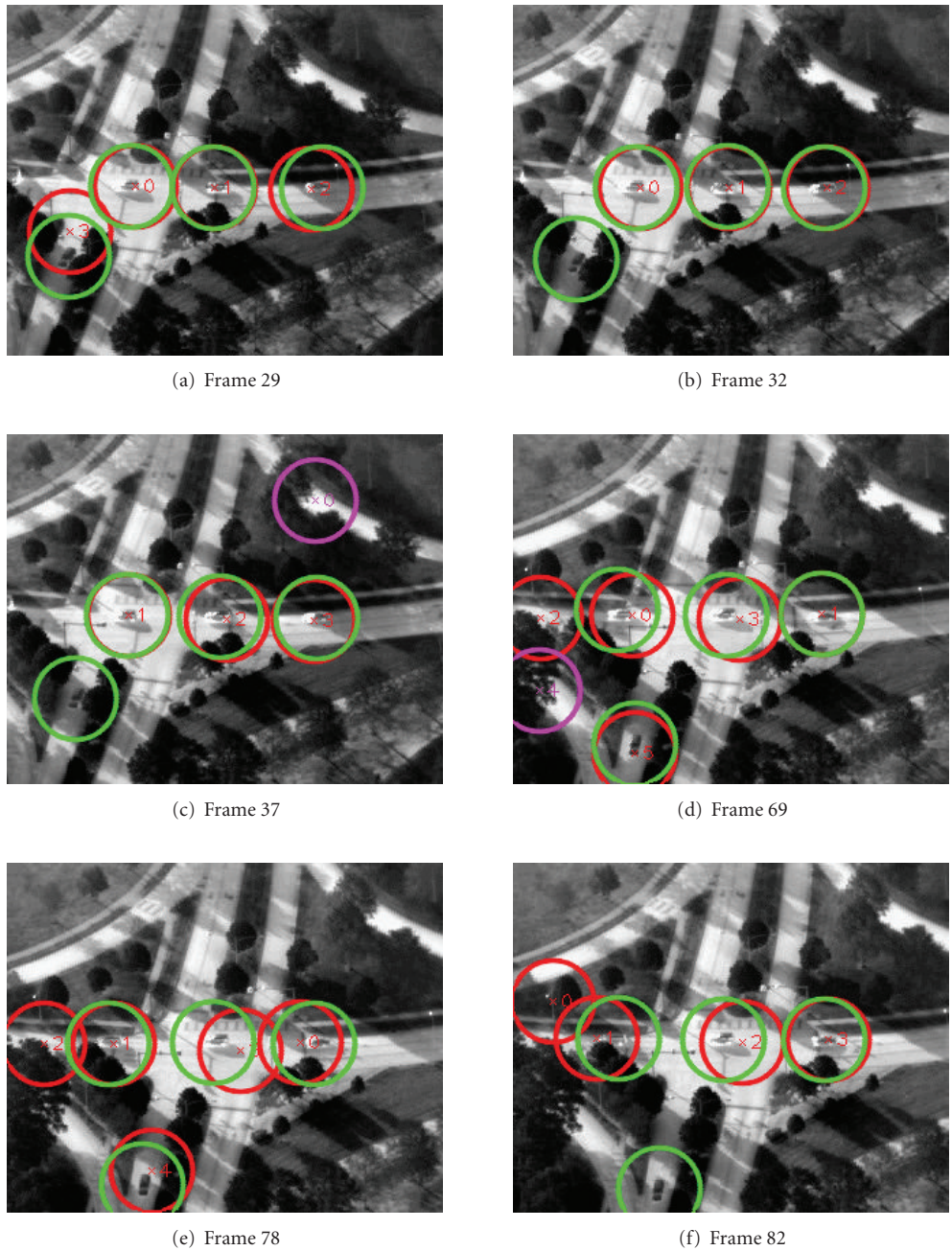


Figure 16: (a) and (b) Two input images; (c) Detected optical flows (marked by red line segments) and outliers clustering (marked by purple circles); (d) Frame difference; (e) Detected specular highlights; (f) Detected moving targets marked by red circles.

and hit rate is 78.6%, respectively. The detail detection results are shown in Table 2. The average processing time is 581 ms/frame. The detailed processing time are shown in Figure 19.

6. Conclusions and Future Works

This paper described a method for multiple moving target detection from airborne IR imagery. It consists of motion compensation, dynamic Gabor filtering, specular highlights detection, and target localization. In motion compensation, the optical flows for two consecutive images are detected from the feature points. The feature points are separated into inliers and outliers, accordingly, the optical flows are also separated into two classes, optical flows belonging to inliers and optical flows belonging to outliers. The optical flows belonging to inliers are used to calculate the global motion model parameters. Here, the Affine model is employed. After the motion model estimation, the frame difference image is generated. Because of difficulties to detect the targets from the frame difference image, we introduce the dynamic Gabor filter. In this step, we use the orientation of the optical



(a) Frame 29 (b) Frame 32

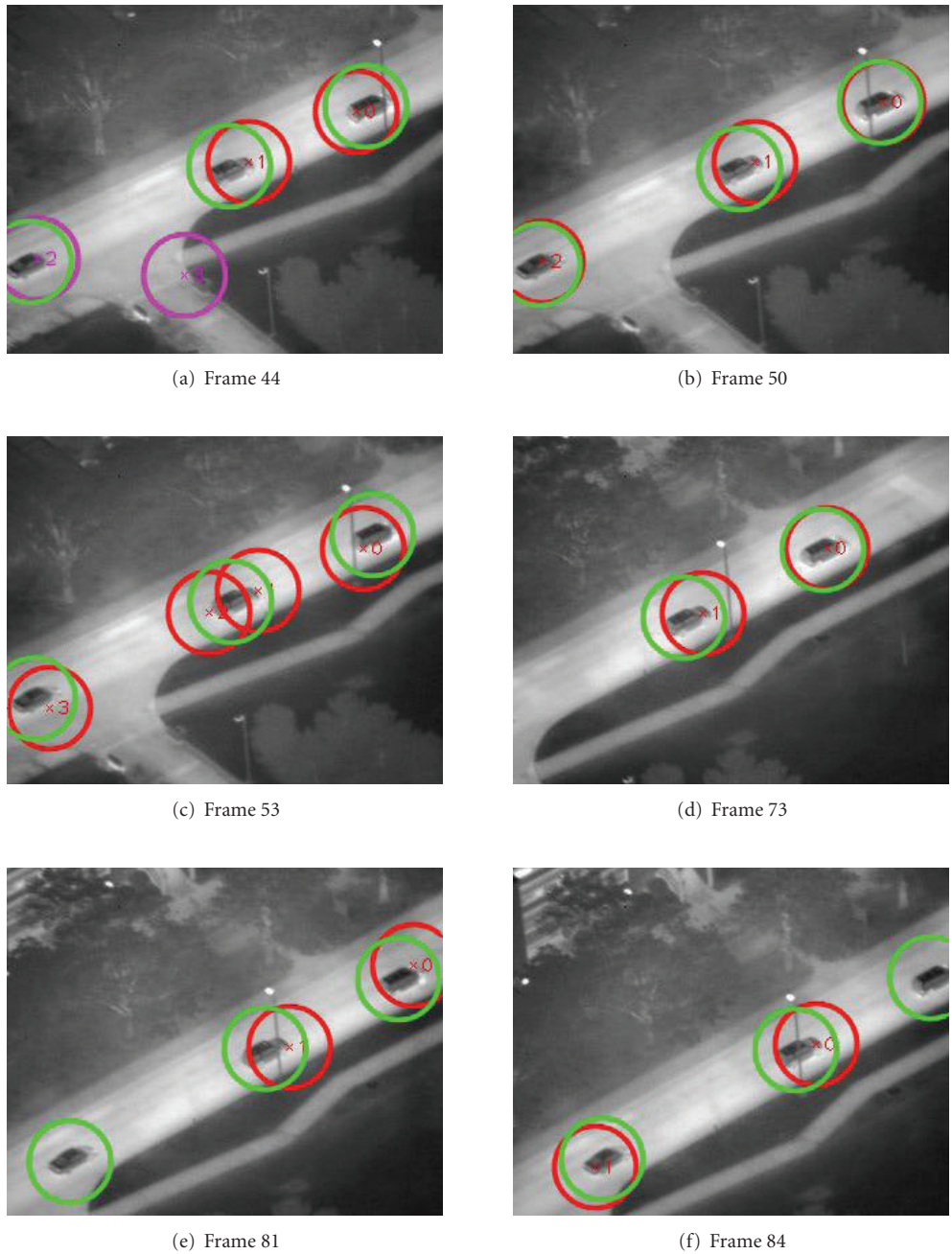
(c) Frame 37 (d) Frame 69

(e) Frame 78 (f) Frame 82

Figure 17: Target detection results in frame 29, 32, 37, 69, 78, and 82. Green circles mark the ground truth target positions, labeled manually. Red circles means targets detected based on outliers clustering and specular highlights. Purple circles mark the output of the dynamic Gaussian detector.

flows belonging to inliers to control the orientation of the Gabor filter. We call it *dynamic Gabor filter*. This is the first contribution of this paper. After the dynamic Gabor filtering, the image changes appear as high intensity in dynamic Gabor filter response. We call these high intensity *specular highlights*. In specular highlight detection, we use a simple but efficient detector to extract the specular highlight points. These specular highlight points are clustered to indentify the specular highlight center and its size. In the last step, it employs the outlier clustering and specular highlights to localize the targets. If a specular highlight lies in an outlier cluster, it is considered as a target. If a specular highlight does not lie in any outlier cluster, it employs the Gaussian detector to identify the target. The orientation of the specular highlight is used to control the orientation of Gaussian kernel. We call this detector *dynamic Gaussian detector*. This is the second contribution of this paper.

This algorithm was implemented in C++ and OpenCV. We tested the algorithm by using the airborne IR videos from AFRL VIVID datasets. The correct detection rate is



(a) Frame 44 (b) Frame 50

(c) Frame 53 (d) Frame 73

(e) Frame 81 (f) Frame 84

Figure 18: Target detection results in frame 44, 50, 53, 73, 81, and 84. Green circles mark the ground truth target positions, labeled manually. Red circles means targets detected based on outliers clustering and specular highlights. Purple circles mark the output of the dynamic Gaussian detector.

86.6%, and the hit rate for the correct detection is 78.6%. The processing rate is 581 ms/frame, that is, approximate 2 frames per second. This speed meets the requirement for many real-time target detection and tracking systems. As seen in Figures 17 and 18, in some cases the system fail to detect the targets or it mistakenly detects the image changes caused by the background significant features such as tree leaves or building corners. This can be improved by two efforts. The first one is to improve the inliers/outliers separation algorithm so that it maximally recognizes the feature points belonging to the background as the inliers. The second effort is to improve the dynamic Gaussian detector. Currently, the threshold for the dynamic Gaussian detector is set at a high value. This rejects some specular highlights to be recognized as targets. However, if this threshold is set at a low value, it will bring about false detection. And σ_x and σ_y in dynamic Gaussian detector are fixed. These can be dynamically changed according to the detection results of the dynamic Gabor filter. As shown in Section 3.1.1, six feature point detectors have been evaluated by employing

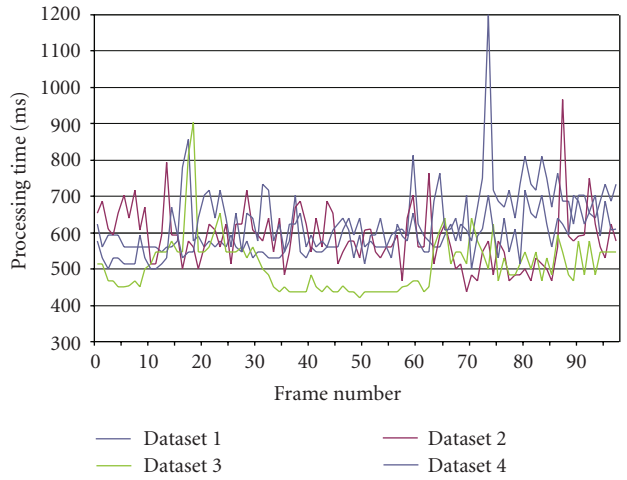


Figure 19: Processing time for multiple moving target detection.

the synthesized images and IR images. The Shi-Tomasi's method shows the best performance experimentally. The detailed performance analysis of these feature point detectors needs the theoretical investigation of these six detectors. The theoretical comparison of them will be detailed in our next paper. As shown in Section 3.1.3, this paper evaluated three transformation models between image frames. The experiment result shows the affine transformation model has best performance. This is because that, for the airborne-based IR image, the camera is far away from the object and the panning and tiling are not distinguished. The further theoretical study of these transformation models is our future work. Furthermore, since the target detection is a part of target tracking system, we will apply this algorithm to the target tracking system. This is also our future works.

Acknowledgments

This paper was partially supported by a Grant from AFRL under Minority Leaders Program, contract No. TENN 06-S567-07-C2. The authors would like to thank AFRL for providing the datasets used in this research. The authors also would like to thank anonymous reviewers for their careful review and valuable comments.

References

[1] A. Yilmaz, K. Shafique, and M. Shah, "Target tracking in airborne forward looking infrared imagery," *Image and Vision Computing*, vol. 21, no. 7, pp. 623–635, 2003.

[2] J. Y. Chen and I. S. Reed, "A detection algorithm for optical targets in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, no. 1, pp. 46–59, 1987.

[3] M. S. Longmire and E. H. Takken, "Lms and matched digital filters for optical clutter suppression," *Applied Optics*, vol. 27, no. 6, pp. 1141–1159, 1988.

[4] H. Shekarforoush and R. Chellappa, "A multi-fractal formalism for stabilization, object detection and tracking in FLIR sequences," in *Proceedings of the International Conference on Image Processing (ICIP '00)*, pp. 78–81, September 2000.

[5] D. Davies, P. Palmer, and Mirmehdi, "Detection and tracking of very small low contrast objects," in *Proceedings of the 9th British Machine Vision Conference*, pp. 599–608, September 1998.

[6] A. Strehl and J. K. Aggarwal, "Detecting moving objects in airborne forward looking infra-red sequences," *Machine Vision Applications Journal*, vol. 11, pp. 267–276, 2000.

[7] U. Braga-Neto, M. Choudhary, and J. Goutsias, "Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators," *Journal of Electronic Imaging*, vol. 13, no. 4, pp. 802–813, 2004.

[8] A. Morin, "Adaptive spatial filtering techniques for the detection of targets in infrared imaging seekers," in *Acquisition, Tracking, and Pointing XIV*, vol. 4025 of *Proceedings of SPIE*, pp. 182–193, 2000.

[9] A. P. Tzannes and D. H. Brooks, "Detection of point targets in image sequences by hypothesis testing: a temporal test first approach," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, pp. 3377–3380, March 1999.

[10] Z. Yin and R. Collins, "Moving object localization in thermal imagery by forward-backward MHI," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (OTCBVS '06)*, New York, NY, USA, June 2006.

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.

[12] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the 9th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, Springer, June 1994.

[13] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[16] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the 9th European Conference on Computer Vision (ECCV '06)*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 430–443, 2006.

[17] F. Mohanna and F. Mokhtarian, "Performance evaluation of corner detection algorithms under similarity and affine transforms," in *Proceedings of the British Machine Vision Conference*, pp. 353–362, 2001.

[18] B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, Mass, USA, 1986.

[19] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[20] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[21] C. L. Zitnick, N. Jojic, and S. B. Kang, "Consistent segmentation for optical flow estimation," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1308–1315, October 2005.

[22] J. Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm*, Intel Corporation, 2003.

[23] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. P. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, October 2007.

[24] C. Moler, "Least squares," in *Numerical Computing with MATLAB*, chapter 5, pp. 141–159, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2008.

[25] S. Araki, T. Matsuoka, N. Yokoya, and H. Takemura, "Real-time tracking of multiple moving object contours in a moving camera image sequence," *IEICE Transactions on Information and Systems*, vol. 83, no. 7, pp. 1583–1591, 2000.

[26] D. H. Rao and P. P. Panduranga, "A survey on image enhancement techniques: classical spatial filter, neural network, cellular neural network and fuzzy filter," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '06)*, pp. 2821–2826, December 2006.

[27] D. Gabor, "Theory of communication," *Journal of IEE*, vol. 93, no. 26, pp. 429–457, 1946.

[28] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, no. 10, pp. 847–856, 1980.

[29] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, "Section 4.4: *kn*—nearest-neighbor estimation," in *Pattern Classification*, Wiley InterScience, Malden, Mass, USA, 2004.

[31] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[32] J. Li, X. Gao, and L. Jiao, "A novel clustering method based on SVM," in *Advances in Neural Networks*, Lecture Notes in Computer Science, pp. 57–62, Springer, Berlin, Germany, 2005.

[33] C. Balletti and F. Guerra, "Image matching for historical maps comparison," *e-Perimetron*, vol. 4, no. 3, pp. 180–186, 2009.