Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Multispectral detection and tracking of multiple moving targets in cluttered urban environments

Casey D. Demars
Michael C. Roggemann
Timothy C. Havens

SPIE.

Multispectral detection and tracking of multiple moving targets in cluttered urban environments

Casey D. Demars,[a,*] Michael C. Roggemann,[a] and Timothy C. Havens[a,b]
[a]Michigan Technological University, Department of Electrical and Computer Engineering, 1400 Townsend Drive, Houghton, Michigan 49931, United States
[b]Michigan Technological University, Department of Computer Science, 1400 Townsend Drive, Houghton, Michigan 49931, United States

Abstract. This paper presents an algorithm for target detection and tracking by fusion of multispectral imagery. In all spectral bands, we build a background model of the pixel intensities using a Gaussian mixture model, and pixels not belonging to the model are classified as foreground pixels. Foreground pixels from the spectral bands are weighted and summed into a single foreground map and filtered to give the fused foreground map. Foreground pixels are grouped into target candidates and associated with targets from a tracking database by matching features from the scale-invariant feature transform. The performance of our algorithm was evaluated with a synthetically generated data set of visible, near-infrared, midwave infrared, and long-wave infrared video sequences. With a fused combination of the spectral bands, the proposed algorithm lowers the false alarm rate while maintaining high detection rates. All 12 vehicles were tracked throughout the sequence, with one instance of a lost track that was later recovered. [DOI: 10.1117/1.OE.54.12.123106]

Keywords: image fusion; multitarget tracking; Gaussian mixture model; multispectral; scale-invariant feature transform features.

Paper 151023 received Jul. 27, 2015; accepted for publication Nov. 10, 2015; published online Dec. 14, 2015.

1 Introduction

Automatic detection and tracking of moving targets in full motion video from aerial imaging systems such as unmanned aerial vehicles (UAV) and satellites are of significant interest in the defense and security communities.[1–3] These aerial platforms can remain undetected from prospective targets and encompass a large surveillance area. Satellites have been a primary "spy" tool for decades and continue to provide for their respective nations, but their coverage is limited by orbital mechanics, and is hence not always sufficiently timely, nor can a satellite generally be launched on demand to address a short-term tactical matter in the field. Vast amounts of research have been invested in UAV surveillance, and UAVs have been a significant resource for intelligence gathering.[1,4,5] Large areas, such as open waters or borders, can be surveyed for intrusions, regions can be assessed for building of weapon facilities, or urban areas can be checked for potential threats.

The urban environment is of interest for this work. Urban environments provide significant challenges to the problem of automatically detecting and tracking moving vehicles. These areas generally contain complicated clutter and a collection of different targets, e.g., humans, buildings, roads, and vehicles. Each of these different entities also varies greatly in shape and size that challenge automatic target detection and recognition algorithms. Trees, buildings, tunnels, and other formations result in object occlusions that affect the appearance of the targets and sometimes completely block the targets from view for a few to several consecutive frames. Images in the visible spectrum (0.4 to 0.7 μm) provide reflected spectral information that creates contrast between targets, and between targets and the background. The visible spectrum requires good illumination during the daytime hours. Imaging in the long-wave infrared (LWIR) band (8 to 14 μm) is dependent on the temperature and thermal emissivity of the target, but is not dependent on solar illumination, and thus provides nighttime imaging capabilities. The effects of atmospheric aerosols also play a role in these imaging modalities. For example, Mie scattering significantly hinders the performance of visible imaging in the presence of fog aerosols.[6] The wavelengths of the midwave infrared (MWIR) and LWIR bands are longer than the visible wavelength, making them less susceptible to the attenuation due to Mie scattering, and thus provide some immunity to the effects of fog and other aerosols on image quality.[6]

The approach of multispectral detection and tracking fuses information obtained from images in different spectral bands to improve detection statistics. Various approaches have been taken in algorithm development for detecting and tracking using multispectral imagery where the fusion framework takes place in three stages of the processing: pixel level,[7–11] feature level,[12–14] and decision level.[15] Fusion at the pixel level creates a single image that is a composition of the pixels in the multispectral images. It is often used to create a single image that is interpreted by an operator.[16,17] The combination of pixels into a single image is difficult as there is not always a correlation of the pixel values from the different spectral images, and it has been found that a mild anticorrelation exists between the visible and LWIR bands.[18] Feature-level fusion combines the by-product of processing of individual spectral bands. These processing products include numerous classifications of features, such as foreground maps, histograms, edge contours, and texture

*Address all correspondence to: Casey D. Demars, E-mail: cddemars@mtu.edu

features. Processing of individual spectral bands allows feature extraction algorithms to be optimized for each band. In decision-level fusion, processing is performed on each independent spectral band where a decision is made, such as object size and location. These decisions are fused based on band-specific confidence levels to give an overall decision.

To exploit benefits of each spectral band, feature-level and decision-level fusion allow algorithm development tailored for their respective bands. Algorithms fusing background models from different image modalities to create a common foreground for target detection have been demonstrated.[12–15] Chen and Wolf[13] model the foreground in both visible and LWIR imagery with the mixture-of-Gaussians model, while using an adaptive learning rate that is based on the decision of each spectral band. They also fuse the two spectral bands for their appearance model to increase the performance of target association. Torresan et al.[15] perform the background subtraction on each individual spectral modality and merge the results by picking a master and slave foreground map based upon the confidence of each modality. By modeling each background pixel's intensity as a single Gaussian distribution, Davis and Sharma[12] extract regions-of-interest by the intersection of the visible and LWIR foreground maps. Salient contours from the regions-of-interest are then calculated from both visible and LWIR images and fused to create a single contour saliency map.

Table 1 Spectral band and their respective wavelengths.

Spectral band	Wavelengths (μm)
Visible	0.4 to 0.7
Near-infrared (NIR)	0.8 to 1.2
Midwave infrared (MWIR)	3 to 5
Long-wave infrared (LWIR)	8 to 14

The aforementioned works consider visible and LWIR bands; our algorithm additionally exploits near-infrared (NIR) and MWIR bands, and the combinations of spectral bands. Table 1 shows the spectral bands used and their associated wavelengths. The main contribution of this work is an algorithm to fuse multispectral data sets to reliably detect and track moving targets with high-probability and low-false alarm rate. We focus on detection and tracking of vehicles through an urban scene that includes partial occlusions and crowded traffic intersections. A block diagram of our proposed algorithm is shown in Fig. 1. To compensate for fluctuating pixel intensities in each spectral band, background models using a Gaussian mixture model (GMM) adapt to the evolving scenes and detect foreground pixels. Foreground pixels from different spectral bands are fused into a foreground region and filtered to obtain a single foreground map that represents pixel regions belonging to target candidates. Features based on the scale-invariant feature transform (SIFT) are extracted from these target regions and used for two purposes:[19] detecting targets missed by the segmentation detection, and associating targets from a tracking database constructed from prior frames. Lastly, locations for each target are estimated and the GMM mixture is updated.

To develop and evaluate the algorithm, we created a UAV imaging scenario that was synthetically generated from the digital imaging and remote sensing image generation (DIRSIG) toolset.[20] DIRSIG is a mature and widely used simulation package for 0.4-μm to 20-μm wavelengths. An urban scene with 12 vehicles was simulated at visible, NIR, MWIR, and LWIR wavelengths. A normal traffic scenario was simulated using the open-source tool simulation of urban mobility (SUMO) to provide realistic traffic maneuvers. Figure 2 shows a 2000×2000 pixel frame from each spectral band. By visual inspection, the appearance of target vehicles varies between the scenes, providing different intensity information.

The remainder of the paper is organized as follows: Sec. 2 presents the method for foreground extraction using the GMM and the region growing process to group disjoint

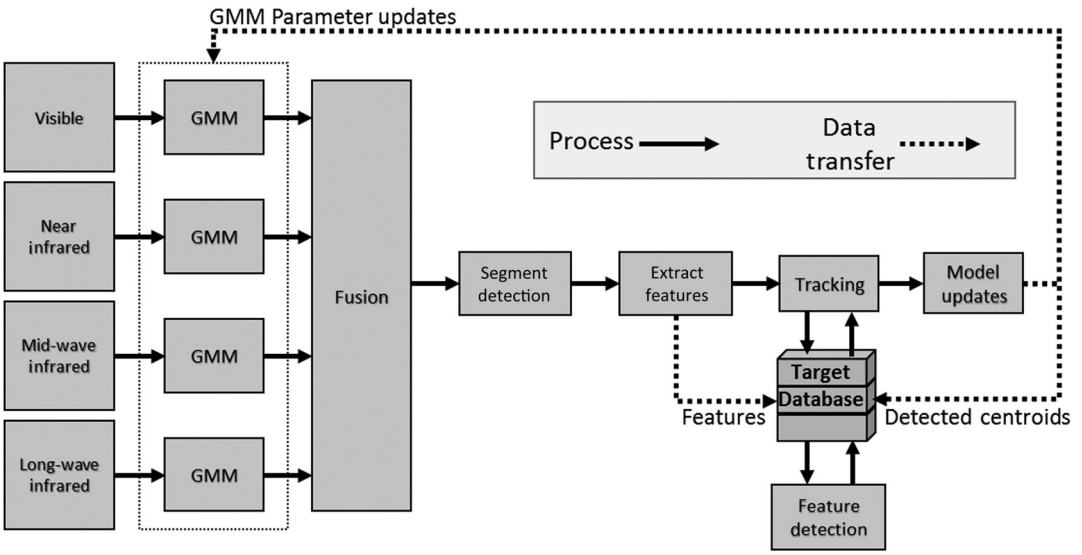

Fig. 1 Algorithm process flowchart.

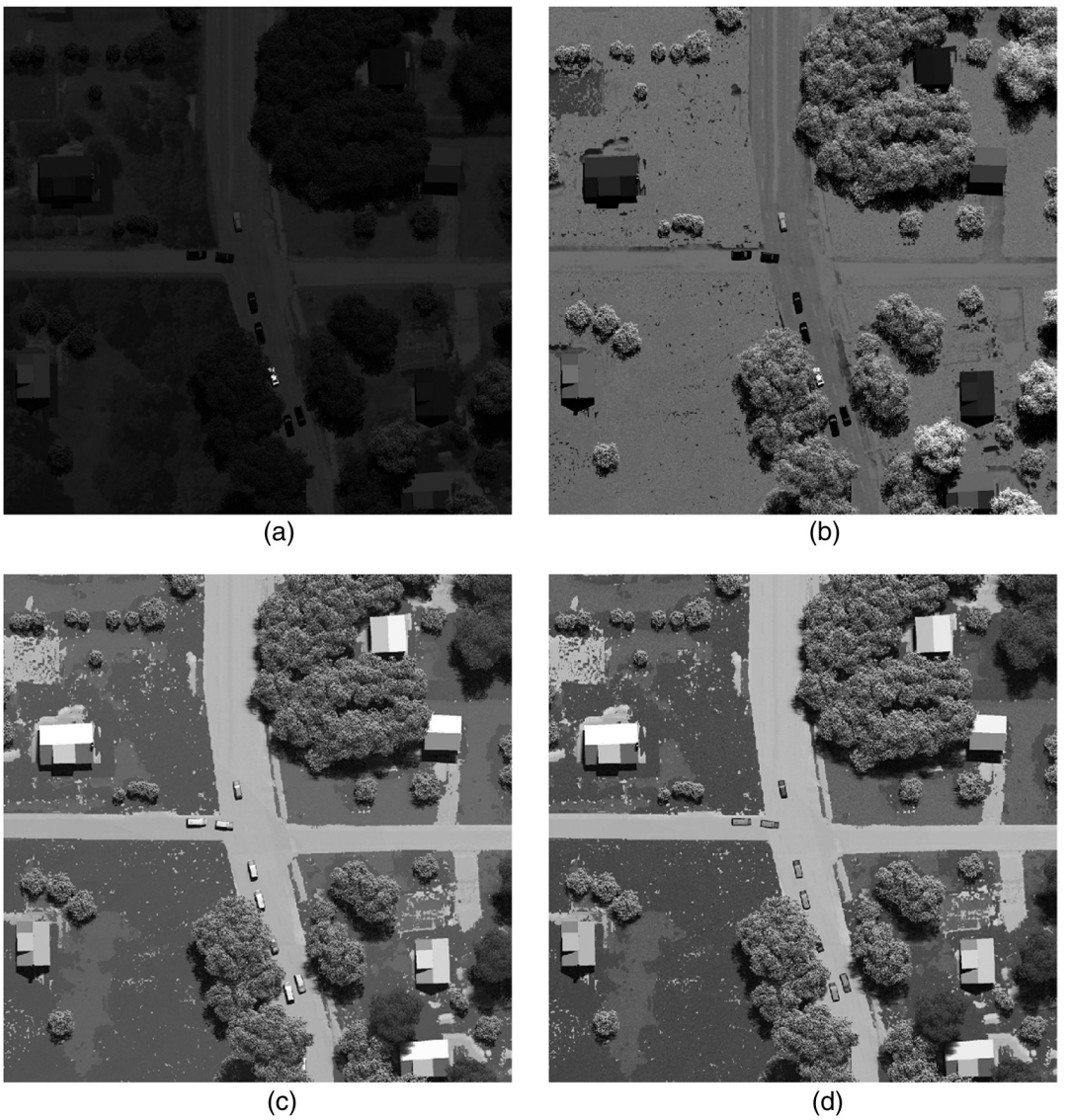

Fig. 2 Example multispectral frames from evaluated digital imaging and remote sensing image generation data set: (a) visible, (b) near-infrared (NIR), (c) midwave infrared (MWIR), and (d) long-wave infrared (LWIR).

pixels. We also discuss our method for fusing the spectral modalities in Sec. 2. Section 3 presents the association target candidates with track sequences. Experimental results on the performance of the algorithm are presented in Sec. 4. In Sec. 5, conclusions are presented.

2 Detection and Segmentation Algorithm

In this section, we describe the detection and segmentation algorithms; we then present the fusion process used to combine foreground maps to build pixel regions that represent target candidates. Pixel intensities fluctuate due to changes in illumination and movement from both background and target objects. This does not allow a single value to characterize the time history of the intensity of a single pixel for a given video sequence. To compensate for these changes, background modeling techniques are used to describe the probability distribution of the pixels' intensity by empirically deriving the parameters from the video sequence. The GMM has been successfully demonstrated to compensate for the fluctuations in pixel intensities.[21–23] In a scene where the sensor is fixed, keeping the viewpoint stationary, we use statistical information extracted from the time history of the intensity fluctuations to understand the probability distribution of intensity at each pixel, and use these distributions to make hypotheses about the label of each pixel. Each pixel in the scene is classified as a foreground or background pixel, and we update the parameters of the GMM during each frame. We now describe this process in detail.

We define $X(x, y; t)$ as the pixel intensity at location (x, y) and time t. The goal is to classify this pixel as a background or foreground pixel by fitting it to a distribution model. The distribution of the time history of the intensity, $P[X(x, y; t)]$, is modeled as a sum of weighted Gaussian distributions:

$$P[X(x,y;t)] = \sum_{j=1}^{K} w_{j,t}(x,y)\mathcal{N}[X(x,y;t), \mu_{j,t}(x,y), \Sigma_{j,t}(x,y)], \tag{1}$$

where K is the number of Gaussian distributions; $\mu_{j,t}(x,y)$ is the mean of the distributions; and the covariance matrix, which is assumed to be diagonal, is given by $\Sigma_{j,t}(x,y) = \sigma_{j,t}^2(x,y)I$, where I is the identity matrix. The weighting factor $w_{j,t}(x,y)$ represents the portion of which the j'th Gaussian that comprises the entire model, and is dependent on the number of occurrences for the particular distribution. This weighting has range $0 < w_{j,t} \leq 1$, and is normalized such that $\sum_{j=1}^{K} w_{j,t} = 1$. The Gaussian probability density function is

$$N[X(x,y;t), \mu_{j,t}(x,y), \Sigma_{j,t}(x,y)] = \frac{1}{(2\pi)^{n/2}|\Sigma_{j,t}(x,y)|^{1/2}} \exp\left\{-\frac{1}{2}[X(x,y;t) - \mu_{j,t}(x,y)]^{\mathrm{T}} \times \Sigma_{j,t}(x,y)^{-1}[X(x,y;t) - \mu_{j,t}(x,y)]\right\}. \quad (2)$$

From the K distributions, it must be determined that the number of distributions are classified as belonging to the background. We select the top B weighted distributions as the background, where

$$B = \mathrm{argmin}_b\left[\sum_{j=1}^{b} w_{j,t}(x,y) > \mathrm{Thr}\right]. \quad (3)$$

The threshold Thr is user defined with range (0,1) and is dependent on the scene.

Table 2 Gaussian mixture model parameters.

	VIS	NIR	MWIR	LWIR
Learning rate	0.0004	0.0005	0.0004	0.0003
Threshold	0.7	0.8	0.7	0.5

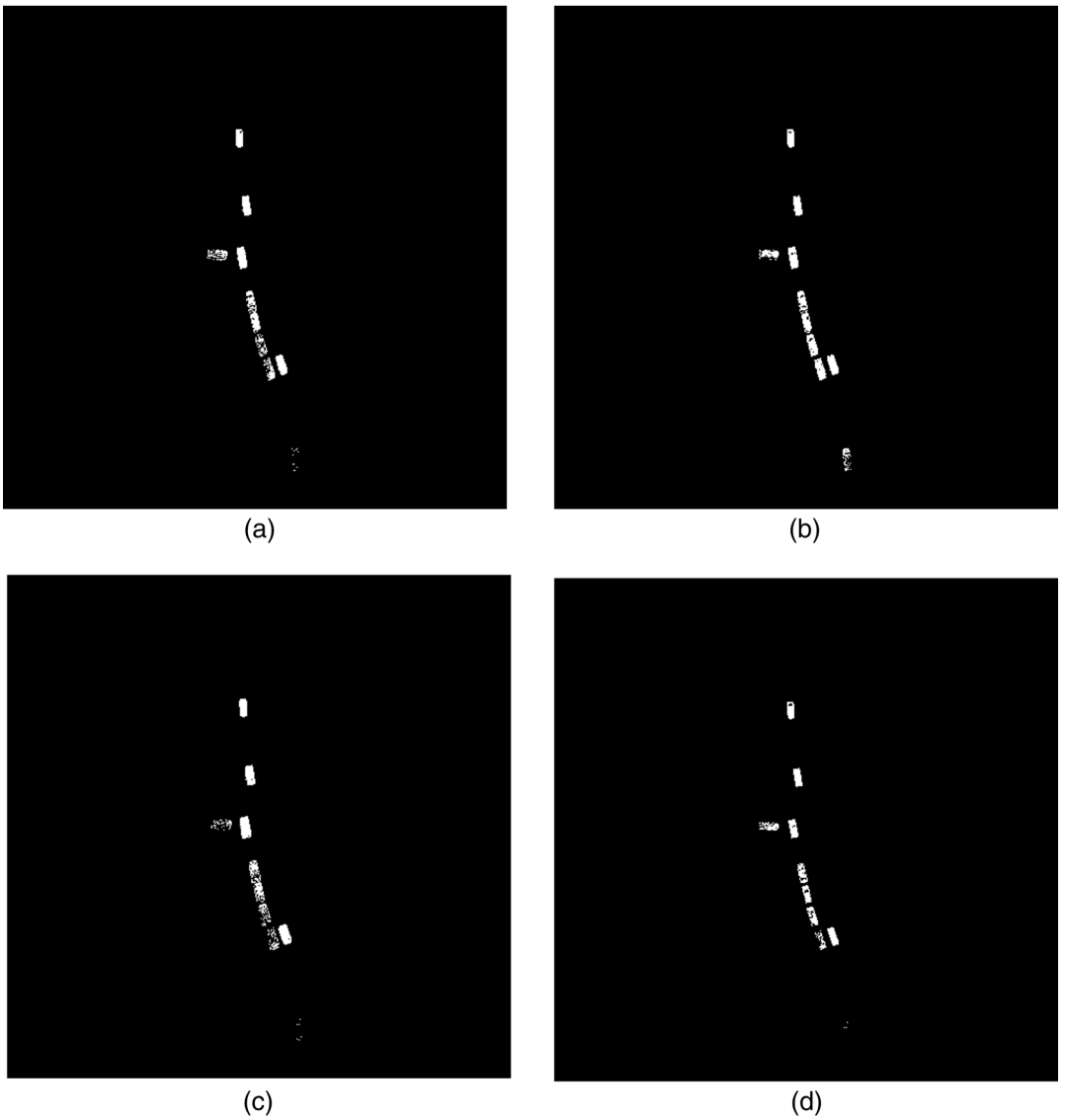

Fig. 3 Foreground images at frame 600: (a) VIS, (b) NIR, (c) MWIR, and (d) LWIR.

In a complex scene with multiple moving targets where pixel distributions vary among targets, and among targets and background, more Gaussian models will be present and thus require higher Thr. In the scenes tested with this algorithm, few objects were moving and typically only one Gaussian mode was needed to describe the background. By executing the GMM algorithm with a series of parameters on the test data set, the optimal Thr was empirically derived for each spectral band by comparing correctly detected pixels to falsely detected pixels. The resulting values of Thr are shown in Table 2. In the algorithm, LWIR had the lowest Thr at 0.5, which is attributed to the distributions of the target intensities being similar, along with being lower than the background surrounding the targets.

To evaluate whether the current pixel intensity $X(x,y;t)$ is a background or foreground pixel, we calculate the *a priori* probability of that pixel intensity belonging to each of the K distribution components. If the intensity value falls within 2.5 standard deviations of any background distribution, it is labeled background; otherwise, it is labeled as foreground. Following the classification of the pixel, the distribution parameters are updated as[23]

$$w_{j,t+1}(x,y) = w_{j,t}(x,y) + \alpha[1 - w_{j,t}(x,y)], \tag{4}$$

$$\mu_{j,t+1}(x,y) = \mu_{j,t}(x,y) + [\alpha/w_{j,t}(x,y)][X(x,y;t) - \mu_{j,t}(x,y)], \tag{5}$$

$$\sigma^2_{j,t+1}(x,y) = \sigma^2_{j,t} + [\alpha/w_{j,t}(x,y)]\{[X(x,y;t) - \mu_{j,t}(x,y)]^{\mathrm{T}} \times [X(x,y;t) - \mu_{j,t}(x,y)] - \sigma_{j,t}(x,y)^2\}, \tag{6}$$

where α is the learning rate. In a scene where objects typically move slowly, the update equations should also update at a slower rate and require a smaller α. After the experimentation, we found the optimal α for each data set as shown in Table 2. All spectral bands used a low α, with the lowest value in the LWIR band, which can be attributed to no shadows being present.

The GMM algorithm produces intermediate foreground maps in all spectral bands that do not represent the complete target region and do not necessarily correlate with one another. This is a consequence of discrepancies in the foreground modeling, and is caused by low SNR between the target and background. Examples of intermediate foreground maps at frame 600 are shown in Fig. 3. In the NIR band, the bottom target has a high number of foreground pixels in comparison with the other bands. In the MWIR band, the target on the left has a low number of foreground pixels whereas the other bands have a high number of pixels. The fusion of foreground maps from multispectral video creates combined foreground maps that accurately estimate the centroid of the target with a low-false alarm rate. This is distinct from previous efforts[12–17] in that we have considered additional spectral bands for foreground fusion, and use SIFT features for unique target identification and detection of missed targets.

We define a fused foreground map, $\mathrm{FG}_{\mathrm{FUS}}(x,y)$, as the sum of individual weighted foreground maps, where w represents the weighting and the subscript represents the respective band,

$$\mathrm{FG}_{\mathrm{FUS}}(x,y) = w_{\mathrm{VIS}}\mathrm{FG}_{\mathrm{VIS}}(x,y) + w_{\mathrm{NIR}}\mathrm{FG}_{\mathrm{NIR}}(x,y) + w_{\mathrm{MWIR}}\mathrm{FG}_{\mathrm{MWIR}}(x,y) + w_{\mathrm{LWIR}}\mathrm{FG}_{\mathrm{LWIR}}(x,y), \tag{7}$$

$\mathrm{FG}_{\mathrm{FUS}}(x,y)$ is spatially filtered with a 3×3 Gaussian filter with $\sigma = 0.5$. Thresholding of $\mathrm{FG}_{\mathrm{FUS}}(x,y)$ is performed to remove pixels that have a low-foreground probability of belonging to the foreground, $\mathrm{FG}_{\mathrm{FUS}}(x,y) < \mathrm{th}$. The spectral combinations and their respective thresholds are shown in Table 3. Thresholds were chosen by the lowest false alarm rate produced by the detection algorithm. False alarm rates for the series of tested thresholds are presented in the Sec. 4 in Table 4.

An example of a fused foreground is shown in Fig. 4(a). A smoothing of the combined foreground map is applied using a two-dimensional Gaussian filter and shown in Fig. 4(b). The filtering results in filling of gaps where pixels were missed from the foreground map without overdilating the region. The final foreground map is shown in Fig. 4(c). A zoomed area on a car region depicting the foreground fusion process is shown in Fig. 5. The effect of thresholding the fused and filtered foreground map is illustrated; the target shadow is removed from the foreground region.

The final step of creating the pixel regions that represent the detected candidates is an image closing, which consists of a dilation followed by an erosion. The structuring element of this procedure is a disk with a radius of four pixels. The

Table 3 Spectral combinations and their respective background threshold.

Combination	Background threshold
VIS	—
NIR	—
MWIR	—
LWIR	0.2
VIS–NIR	1.0
VIS–MWIR	—
VIS–LWIR	1.6
NIR–MWIR	0.4
NIR–LWIR	1.2
MWIR–LWIR	1.0
VIS–NIR–MWIR	1.2
VIS–NIR–LWIR	2.0
VIS–MWIR–LWIR	2.0
NIR–MWIR–LWIR	1.2
VIS–NIR–MWIR–LWIR (TOT)	2.0
VIS–NIR–MWIR–3*LWIR (TOT3)	3.0
VIS–3*LWIR	3.4

Table 4 False alarm rates produced for given thresholds.

Threshold	VIS	NIR	MWIR	LWIR	VIS NIR	VIS MWIR	VIS LWIR	NIR MWIR	NIR LWIR	MWIR LWIR	VIS NIR MWIR	VIS NIR LWIR	VIS MWIR LWIR	NIR MWIR LWIR	TOT	TOT3	VIS LWIR3
0.0	**1.14**	**1.07**	**1.15**	1.32	1.17	**1.14**	1.38	1.18	1.45	1.41	1.17	1.42	1.39	1.45	1.42	1.42	1.38
0.2	1.30	1.08	1.19	**1.30**	1.34	1.26	1.41	1.12	1.44	1.35	1.31	1.41	1.39	1.46	1.37	1.38	1.38
0.4	1.30	1.08	1.19	1.30	1.31	1.26	1.40	**1.03**	1.43	1.36	1.30	1.39	1.39	1.44	1.36	1.36	1.41
0.6	1.30	1.08	1.19	1.30	1.31	1.26	1.40	1.03	1.43	1.36	1.30	1.39	1.39	1.44	1.37	1.36	1.40
0.8	1.39	1.26	1.77	1.65	1.26	1.27	1.46	1.11	1.42	1.46	1.26	1.49	1.49	1.43	1.49	1.50	1.44
1.0	—	—	—	—	**1.08**	1.15	1.16	1.12	1.10	**0.96**	1.02	1.16	1.23	1.01	1.17	1.41	1.30
1.2	–	–	–	—	1.14	1.31	1.24	1.24	**1.08**	1.17	**0.99**	1.15	1.22	**0.99**	1.16	1.36	1.29
1.4	—	—	—	——	1.15	1.33	1.24	1.25	1.12	1.24	1.02	1.12	1.24	1.02	1.17	1.38	1.30
1.6	—	—	—	—	1.12	1.36	**1.17**	1.34	1.16	1.34	1.11	1.16	1.27	1.05	1.18	1.37	1.30
1.8	—	—	—	—	1.31	1.53	1.23	1.45	1.17	1.28	1.16	1.29	1.24	1.16	1.19	1.37	1.30
2.0	—	—	—	—	—	—	—	—	—	—	1.22	**1.08**	**1.10**	1.13	**1.04**	1.36	1.31
2.2	—	—	—	—	—	—	—	—	—	—	1.30	1.08	1.21	1.16	1.07	1.35	1.32
2.4	—	—	—	—	—	—	—	—	—	—	1.37	1.11	1.24	1.09	1.09	1.40	1.56
2.6	—	—	—	—	—	—	—	—	—	—	1.34	1.15	1.26	1.12	1.13	1.47	1.54
2.8	—	—	—	—	—	—	—	—	—	—	1.47	1.30	1.21	1.15	1.14	1.41	1.50
3.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.15	**1.16**	1.29
3.2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.17	1.20	1.18
3.4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.16	1.18	**1.14**
3.6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.24	1.21	1.14
3.8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.25	1.21	1.29
4.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.06	—
4.2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.08	—
4.4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.07	—
4.6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.06	—
4.8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.11	—
5.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1.15	—

Note: dashes (—) imply the threshold exceeded the maximum obtainable pixel value in the image.

dilation operation fills in voids between pixel segments and grows the size of the region. In the erosion operation, we attempt to remove any unnecessary region growth that is a by-product of the dilation. Pixel regions that do not exceed an area of 200 pixels are filtered to remove the objects that may not represent vehicle-sized objects.

3 Target Tracking

The association of targets involves relating a track sequence from prior frames with target candidates detected in the current frame. This task is trivial in the case where targets stay separated and no occlusions exist. However, in actual practice and in this data set, targets become merged or occluded, making distinguishing between targets difficult.

We have chosen to use SIFT features for identification due to their robustness with respect to changes in rotation and scale, and their invariance to change in camera viewpoints and illumination changes.[19] Due to our reliance on these features to uniquely identify targets, we require them to be robust in long-term tracking applications. A disadvantage of SIFT is the heavy computations required for the keypoints, where typical processing times are tenths of seconds to multiple seconds per frame in a normal CPU implementation.[24,25] Developments in graphics processing units

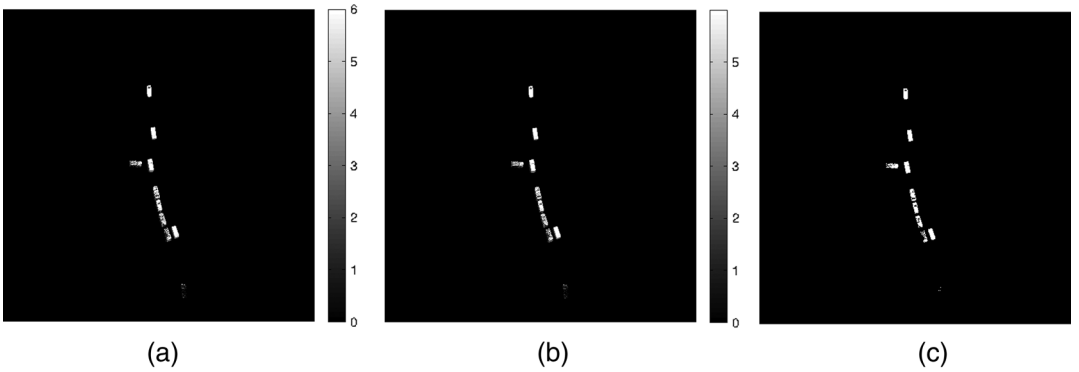

Fig. 4 Fused foreground maps: (a) the fused foreground maps, $FG_{FUS}(x, y)$; (b) the foreground map after applying a Gaussian filter; and (c) the foreground after a threshold has been applied to the filtered image.

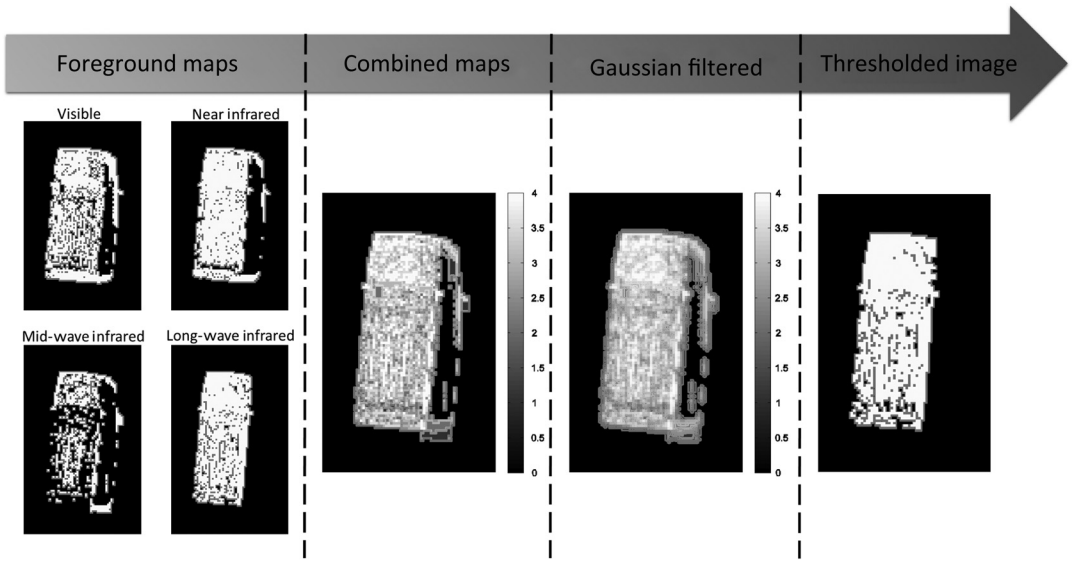

Fig. 5 Zoomed area highlighting the foreground fusion process.

(GPUs) and field programmable gate arrays (FPGAs) have created opportunities for real-time algorithms. SIFT implementations have been developed for both GPUs[25–27] and FPGAs,[24] where the results demonstrate real-time SIFT calculations.

SIFT features are composed of a keypoint that gives subpixel location and orientation of the feature, along with a descriptor that is calculated based on local pixel texture. In the SIFT algorithm, keypoints are first identified at multiple scales. A scale space of the image is created with varying amounts of blur applied to each image using the Gaussian kernel. The blurred image is defined as

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y), \tag{8}$$

where the Gaussian kernel $G(x, y, k\sigma)$ with variance $k\sigma$ is

$$G(x, y, k\sigma) = \frac{1}{2\pi k\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\,k\sigma^2}\right). \tag{9}$$

Within the scale space, difference of Gaussian (DoG) images are calculated by

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{10}$$

The local extrema in the DoG images at each scale are found by comparing the pixel value with its eight surrounding pixels and the nine neighboring pixels from each of the nearest blurred images. To create an invariance to scale, the extrema must exist on multiple scales. A filtering step of the detected extremas in the DoG images is implemented based on the intensity; an extrema with a low intensity is susceptible to changes in illumination and is therefore unstable and removed from the keypoints.

A reference orientation for subsequent processing is given to the keypoint to provide invariance to rotation. From the blurred image in which the extrema was located, the gradient magnitude $m(x, y, k\sigma)$ is calculated by

$$m(x, y, k\sigma) = \sqrt{[L(x+1, y, k\sigma) - L(x-1, y, k\sigma)]^2 + [L(x, y+1, k\sigma) - L(x, y-1, k\sigma)]^2} \tag{11}$$

and the orientation, $\theta(x, y, k\sigma)$, by

$$\theta(x, y, k\sigma) = \tan^{-1}\left[\frac{L(x, y+1, k\sigma) - L(x, y-1, k\sigma)}{L(x+1, y, k\sigma) - L(x-1, y, k\sigma)}\right]. \tag{12}$$

A histogram of 10 deg bins is created of the orientations, and the magnitudes added to the histograms are the gradient magnitudes that are Gaussian weighted with a variance of $1.5k\sigma$. The peaks of the histograms are detected, where the highest peak and any peaks above 80% of the highest peak are selected as orientations for the new keypoints. Peaks in the histogram represent dominant directions of the local gradients.

Unique identifications are generated for each keypoint, referred to as descriptors. A 16×16 region around the keypoint, with respect to the calculated orientation, is divided into 4×4 subregions. Gradient magnitudes and orientations are calculated for each pixel in these subregions, and histograms with 45 deg bins are calculated for each subregion. Through the use of a Gaussian weighting mask with $\sigma = 1/2$ of the descriptor window width (16 pixels for our case), points are inversely weighted proportional to their distance from the keypoint to decrease their contributions and reduce errors caused by window displacements.

To match features from a tracked objects database to features from the current scene, a matching score is calculated by the Euclidean distance between two descriptors. The matching score between a tracked object and a frame object is calculated by

$$\begin{aligned}\text{Score}(D^{\text{obj}}, D^{\text{frm}}) &= \sqrt{(d_1^{\text{obj}} - d_1^{\text{frm}})^2 + (d_2^{\text{obj}} + d_2^{\text{frm}})^2 + \cdots + (d_n^{\text{obj}} + d_n^{\text{frm}})^2} \\ &= \sqrt{\sum_{i=1}^{n} (d_i^{\text{obj}} - d_i^{\text{frm}})^2},\end{aligned} \tag{13}$$

where $D^{\text{obj}} = (d_1^{\text{obj}}, d_2^{\text{obj}}, \ldots, d_n^{\text{obj}})$ is the tracked object descriptor, $D^{\text{frm}} = (d_1^{\text{frm}}, d_2^{\text{frm}}, \ldots, d_n^{\text{frm}})$ is the frame object descriptor, and n is the length of the descriptor vector, which is 128 for our case.

The feature with the shortest Euclidean distance, i.e., the nearest neighbor, is selected as the matching feature. To remove matches that do not have a good match, a comparison is made between the nearest neighbor and the next nearest neighbor. If the ratio of the match scores between the nearest and the next nearest neighbor is >0.8, the match is rejected. Lowe[19] found that this method rejects 90% of all incorrect keypoints and only removes 5% of the correct matches.

Updating the track location is based on several factors. The search region for a matching target candidate is limited to the track's estimated bounding box, preventing erroneous associations with targets of similar appearance but at a distance away. In the event that multiple targets are located in the track bounding box, such as at a road intersection when cars become merged, SIFT features are used to select the correct target. If no target is found in the bounding box, SIFT features are matched in the bounding box region and provide a velocity measurement for a linear motion model. Tracks are propagated if no target is matched and no SIFT features are found, a typical occurrence when the target may be partially or fully occluded from view of the sensor. The propagation projects the location of the bounding box linearly into future frames based on the most recent position and velocity prior to the occlusion.

4 Experiment

The performance of this algorithm was evaluated with a synthetically generated data set using the DIRSIG toolset.[20] A standard midlatitude summer model was used for the atmospheric model MODTRAN.[28] The thermal signatures for 12 vehicles were simulated with the thermal prediction software MuSES.[29] Realistic traffic patterns were generated using the SUMO traffic simulator.[30] The video sequence consists of 600 frames of 2000×2000 pixels sampled at 20 frames/s. The ground sample distance is 0.0635 m and frames are coaligned where pixels correspond geometrically between frames and registered between the spectral bands. As this is a synthetically generated data set, the locations of pixels corresponding to each vehicle is known, providing ground truth centroids of vehicles in the scene.

4.1 *Detection*

We now present the performance for detecting moving targets using our fusion algorithm applied to the DIRSIG data set. For the evaluation of segmented detection rates, a successful detection is a group of pixels that has a centroid with a Euclidean distance within 0.95 m of the centroid of a ground truth object; otherwise, it is considered a false alarm. False alarms are reported on a per frame basis. Targets occluded by 20% or more are not factored into the target detection score.

False alarm rates for a series of examined thresholds are shown in Table 4, where the optimal thresholds are given in bold. The false alarm rate is presented by the number of false alarms per frame. The optimal thresholds for image filtering were selected by choosing the lowest false alarm rate for their respective spectral combination. LWIR resulted in the highest false alarm rate at 1.30. Three of the fusion combinations have false alarms less than 1: MWIR–LWIR, VIS–NIR–MWIR, and NIR–MWIR–LWIR. MWIR–LWIR presented the lowest false alarm rate of 0.96. In the single spectral bands, NIR had the lowest false alarm rate with 1.07.

Detection rates by segmentation for a series of examined thresholds are shown in Table 5, where the results for the optimal thresholds for each spectral combination are given in bold. LWIR achieved the highest detection rate of 0.94 and VIS had a detection rate of 0.93. VIS–MWIR and VIS–LWIR resulted in detection rates of 0.93, while the weighted VIS–NIR–MWIR–3*LWIR (TOT3) and VIS–LWIR3 had results of 0.91 and 0.92, respectively.

Total detection rates and false alarm rates are shown in graph form in Fig. 6. The total detection rates include detections by both segmented objects and features. In the single spectral bands, LWIR resulted in a detection rate of 0.94, but suffered from the highest false alarm rate of 1.30. The detection rate of VIS was slightly lower at 0.93, but lowered the false alarm rate to 1.14. VIS–MWIR had a detection rate of 0.94, while lowering the false alarm rate to 1.14. MWIR–LWIR produced the lowest false alarm rate at 0.96 with a

detection rate of 0.91. These presented fusion results demonstrate that fusing multiple spectral bands lowers false alarms while maintaining high detection rates.

The contribution to the overall detection by segmented objects and features is shown in Fig. 7. The black bar indicates the rate by segmented detection and the gray bar is the additional detection rate by using the SIFT features. Detection by segmentation is the primary detection mechanism and contributes to the bulk of the detection rate, whereas feature detection is secondary and has a smaller impact on the overall detection rate. Pixel texture varies in each spectral band, providing different spatial features that are independent of one another. Extracting features from different spectral bands provides additional features for tracking and identification. Single spectral bands did not have any detection by features, which we attributed to an insignificant number of features to match between the target database and the scene. NIR–MWIR had the highest contribution for detections by features at 0.040. We attributed this to the high number of false pixels that were detected by segmentation, incorporating features that belong to the background. VIS–MWIR–LWIR had the next highest contribution of

Table 5 Detection rates produced for given thresholds.

Threshold	VIS	NIR	MWIR	LWIR	VIS NIR	VIS MWIR	VIS LWIR	NIR MWIR	NIR LWIR	MWIR LWIR	VIS NIR MWIR	VIS NIR LWIR	VIS MWIR LWIR	NIR MWIR LWIR	TOT	TOT3	VIS LWIR3
0.0	**0.93**	**0.91**	**0.89**	0.96	0.93	**0.93**	0.93	0.91	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
0.2	0.92	0.87	0.85	**0.94**	0.92	0.92	0.93	0.89	0.91	0.92	0.92	0.93	0.93	0.91	0.93	0.94	0.93
0.4	0.91	0.87	0.85	0.94	0.92	0.92	0.93	**0.89**	0.91	0.92	0.92	0.93	0.93	0.92	0.93	0.93	0.93
0.6	0.91	0.87	0.85	0.94	0.92	0.92	0.93	0.89	0.91	0.92	0.92	0.93	0.93	0.92	0.93	0.94	0.93
0.8	0.88	0.84	0.80	0.89	0.90	0.89	0.90	0.86	0.89	0.89	0.90	0.90	0.90	0.90	0.91	0.92	0.92
1.0	—	—	—	—	**0.87**	0.87	0.91	0.85	0.89	**0.90**	0.88	0.90	0.89	0.89	0.90	0.92	0.94
1.2	—	—	—	—	0.85	0.84	0.91	0.84	**0.88**	0.86	**0.88**	0.90	0.89	**0.88**	0.90	0.92	0.94
1.4	—	—	—	—	0.85	0.84	0.91	0.84	0.88	0.86	0.87	0.89	0.89	0.88	0.89	0.91	0.94
1.6	—	—	—	—	0.86	0.84	**0.93**	0.83	0.87	0.85	0.86	0.89	0.88	0.87	0.89	0.91	0.94
1.8	—	—	—	—	0.83	0.77	0.91	0.78	0.86	0.78	0.85	0.88	0.90	0.86	0.88	0.91	0.94
2.0	—	—	—	—	—	—	—	—	—	—	0.85	**0.89**	**0.90**	0.87	**0.88**	0.92	0.93
2.2	—	—	—	—	—	—	—	—	—	—	0.84	0.88	0.86	0.85	0.87	0.91	0.93
2.4	—	—	—	—	—	—	—	—	—	—	0.84	0.88	0.86	0.86	0.87	0.90	0.90
2.6	—	—	—	—	—	—	—	—	—	—	0.82	0.87	0.84	0.84	0.87	0.89	0.90
2.8	—	—	—	—	—	—	—	—	—	—	0.76	0.83	0.75	0.76	0.87	0.90	0.89
3.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.86	**0.91**	0.90
3.2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.85	0.90	0.92
3.4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.84	0.90	**0.92**
3.6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.81	0.90	0.92
3.8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.73	0.90	0.87
4.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.89	—
4.2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.89	—
4.4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.89	—
4.6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.89	—
4.8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.88	—
5.0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.86	—

Note: dashes (—) imply the threshold exceeded the maximum obtainable pixel value in the image.

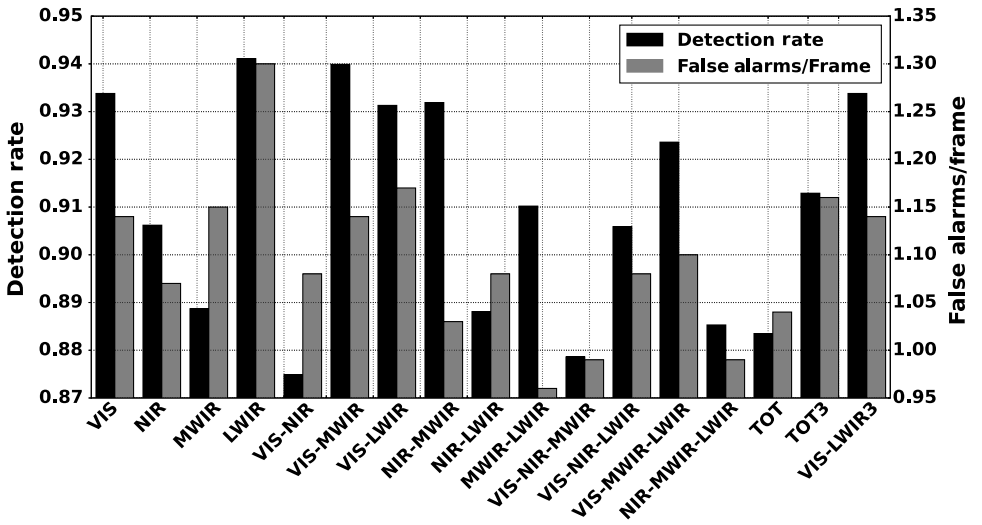

Fig. 6 Detection and false alarm rates.

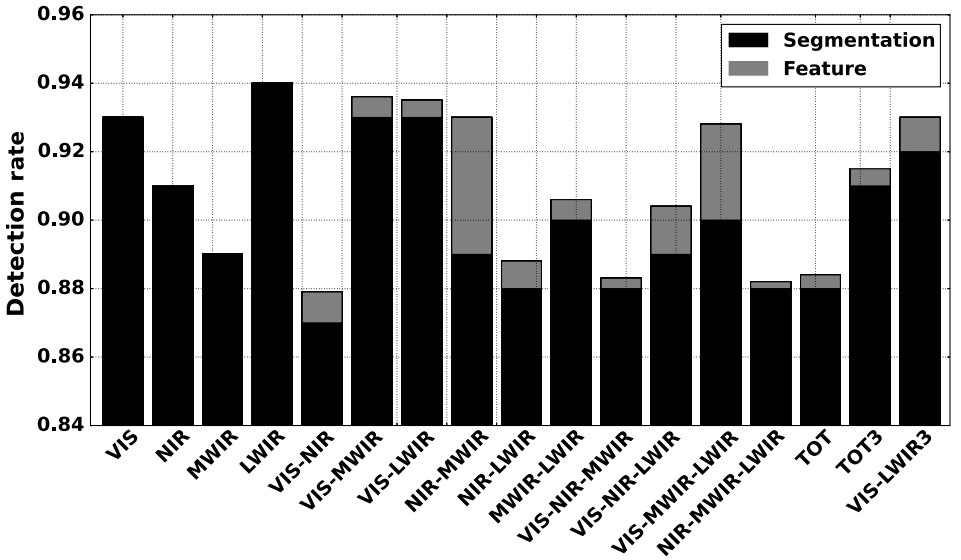

Fig. 7 Detection rate with segmented objects and features.

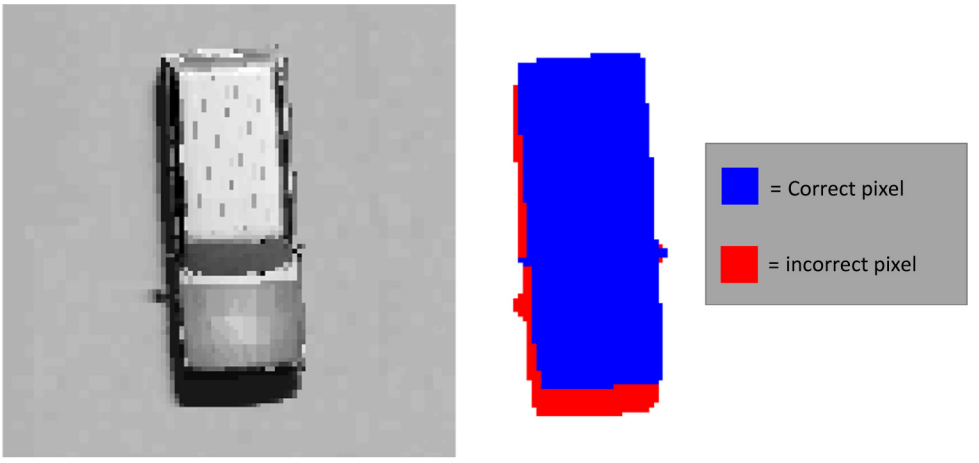

Fig. 8 Example showing how detected pixels are scored as true positives or false alarms: (a) an example vehicle in MWIR; (b) detected foreground pixels; blue indicates true positive and red indicates false alarm.

feature detections with 0.028. The overall detection contribution by features is not significant for this data set, but provides a means to track targets in difficult situations such as busy intersections or partial occlusions when they would otherwise be lost.

Algorithm performance for estimating the targets true centroid by correctly detecting pixels that belong to ground truth objects will now be discussed. A correctly detected pixel is defined as belonging to a ground truth object; otherwise, it is classified as a false pixel. A pixel scoring example is shown in Fig. 8. Figure 8(a) is a ground truth vehicle in MWIR and Fig. 8(b) is labeled as foreground pixels. Blue pixels represent true positives that belong to the ground truth object, and red pixels represent pixels that were falsely detected. For this example, false pixels are attributed to the vehicle shadow.

We define the pixel detection rate for the full video sequence as

$$\text{pixel detection rate} = \frac{\sum_{i=1}^{N} \text{detected ground truth pixels}}{\sum_{i=1}^{N} \text{total ground truth pixels}}, \tag{14}$$

where N is the number of frames. High pixel detection rates result in accurate estimates of target centroids, but falsely detected pixels can negatively affect the centroid calculation, resulting in less accurate results. The false pixel rate is measured per frame and presented as

$$\text{false pixel rate} = \frac{\sum_{i=1}^{N} \text{false pixels detected}}{P * N}, \tag{15}$$

where P is the number of nontarget pixels in the frame and N is the number of frames.

Pixel detection rates and false pixel rates for all spectral bands and fusion combinations are shown in Fig. 9. LWIR

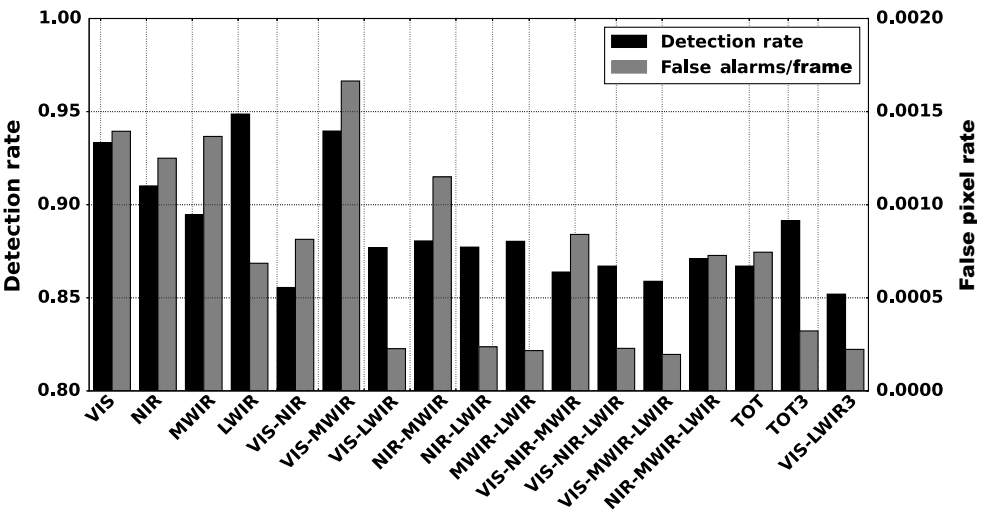

Fig. 9 Rates for pixel detections and false pixels.

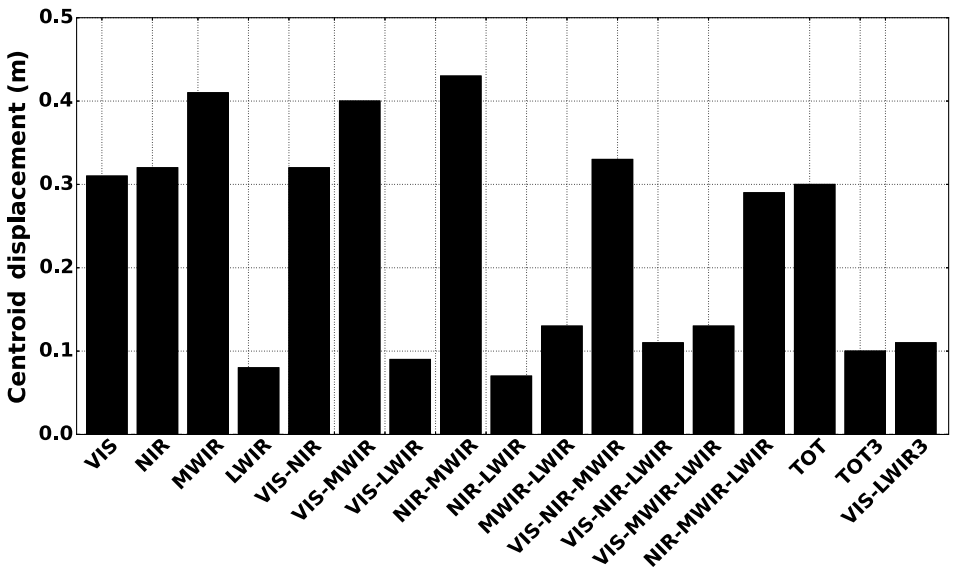

Fig. 10 Mean centroid displacement of detected targets.

produced the highest pixel detection rate of 0.95 and a false pixel rate of 0.0007. The fusion combination VIS–MWIR produced a high pixel detection rate of 0.94, but suffered the highest false pixel rate of 0.0017. TOT3 resulted in a detection rate of 0.89 and false pixel rate of 0.0003. Seven fusion combinations resulted in false pixel rates <0.0005.

Displacement error between detected objects and their respective ground truth centroids is a measure of how accurately an algorithm estimates the true centroid of the target. For target tracking, centroids are input to filters that predict future targets locations, i.e., Kalman filtering, which require accurate estimates. Displacements of detected targets over all frames were measured and the root-mean-square error (RMSE) was calculated as

$$\mathrm{RMSE} = \frac{\sqrt{\sum_{t=1}^{N}[(x-\hat{x})^2 + (y-\hat{y})^2]}}{N}, \tag{16}$$

where $(\hat{x}, \hat{y})$ are the coordinates of the measured centroid, (x, y) are the ground truth centroids, and N is the number of detections. The results are presented in Fig. 10. NIR–LWIR had an RMSE of 0.07 m, which was the lowest for all spectral combinations. LWIR had the next lowest RMSE at 0.08 m, whereas the other single spectral bands had errors >0.3 m. The fusion results presented highlight the centroid accuracy improvements made by fusing spectral bands as compared with using single bands.

4.2 *Tracking*

We now evaluate the performance of the fusion algorithm to associate targets between scenes and create a tracking profile using the foreground combination map (VIS–LWIR). This weighted combination was chosen due to the low centroid error, along with the high detection rate and low-false alarm rate. In this evaluation, the bottom 400 rows of pixels are not considered for the tracking results due to trees

Fig. 11 Single frames from the tracking sequence: (a) frame 150, (b) frame 300, (c) frame 450, and (d) frame 600. (Video 1, MPEG, 14 MB [URL: http://dx.doi.org/10.1117/1.OE.54.12.123106.1]).

Table 6 Results of vehicle tracking.

Total vehicles	Full tracks	Lost tracks	Switch tracks	False tracks
12	11	1	0	3

obstructing the view of the imaging sensors, preventing full vehicle segmentation. Sample images from the tracking sequence are shown in Fig. 11. The yellow box indicates the track algorithm has used a segmented object to update the track location. A teal box indicates that no segmented object matched and SIFT features were used to update the track location.

The motion of 12 vehicles was simulated for an urban traffic environment. Tracks were initiated on all 12 vehicles during the video. Of those 12 vehicle tracks, 11 were tracked though the entire video sequence with no errors. Due to being idle for extended periods of time, one track was lost, but a new track was initiated after it initiated movement. There were no instances where track identities were switched between vehicles and only one instance of a false track. Three false tracks were produced, where two false tracks are attributed to the idle vehicle. The tracking results are summarized in Table 6.

5 Conclusion

In this paper, we proposed an algorithm to fuse multispectral data sets to increase detection accuracy of a video tracker, while maintaining a high detection rate and low-false alarms per frame. Previous works consider visible and LWIR data sets;[12–17] we extend previous work to include NIR and MWIR. In these four spectral bands, we build a GMM to detect foreground pixels by modeling the time history of the pixels intensities. Foreground pixels from all spectral bands are weighted and fused into foreground maps, and formed into targets candidates. Target candidates are tracked through the frame sequence using SIFT features to track missed detections and uniquely identify targets. Our proposed algorithm was tested on synthetically generated data using the DIRSIG toolset of visible, NIR, MWIR, and LWIR imagery. Compared with the single spectral band base, the fused algorithm improves detection accuracy while improving detection rates and lowering false alarm rates. The detection results provided input to a video tracker that detected the 12 moving vehicles in the scene. Of those 12 targets, 11 were tracked with no failures, one vehicle showed track-loss, but this track was reinitiated, and three false tracks occurred.

Acknowledgments

This paper has been approved for public release by the Air Force, Case No. 88ABW-2015-0296, date: January 27, 2015. This work was sponsored by the Air Force Research Laboratory, Contract No. FA8750-12-2-0108. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Department of the Air Force, Air Force Research Laboratory, or the U.S. Government. We would like to thank Brian Wilson and Michigan Tech Research Institute for generating the DIRSIG data sets used in this work.

References

1. F. Heintz, P. Rudol, and P. Doherty, "From images to traffic behavior—a UAV tracking and monitoring application," in *10th Int. Conf. on Information Fusion*, Quebec, pp. 1–8, IEEE (2007).
2. L. Meng and J. P. Kerekes, "Object tracking using high resolution satellite imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(1), 146–152 (2012).
3. P. Liang et al., "Spatial context for moving vehicle detection in wide area motion imagery with multiple kernel learning," *Proc. SPIE* **8751**, 875105 (2013).
4. K. S. Kumar et al., "Visual and thermal image fusion for UAV based target tracking," *MATLAB - A Ubiquitous Tool for the Practical Engineer*, C. M. Ionescu, Ed., p. 307 (2011).
5. H. Choi and Y. Kim, "UAV guidance using a monocular-vision sensor for aerial target tracking," *Control Eng. Pract.* **22**, 10–19 (2014).
6. K. Beier and H. Gemperlein, "Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation," *Aerosp. Sci. Technol.* **8**(1), 63–71 (2004).
7. H. Li, B. Manjunath, and S. K. Mitra, "Multisensor image fusion using the wavelet transform," *Graph. Models Image Process.* **57**(3), 235–245 (1995).
8. A. Ardeshir Goshtasby and S. Nikolov, "Image fusion: advances in the state of the art," *Inf. Fusion* **8**(2), 114–118 (2007).
9. J. Nunez et al., "Multiresolution-based image fusion with additive wavelet decomposition," *IEEE Trans. Geosci. Remote Sens.* **37**(3), 1204–1211 (1999).
10. S. Li, B. Yang, and J. Hu, "Performance comparison of different multi-resolution transforms for image fusion," *Inf. Fusion* **12**(2), 74–84 (2011).
11. Q.-G. Miao et al., "A novel algorithm of image fusion using shearlets," *Opt. Commun.* **284**(6), 1540–1547 (2011).
12. J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Comput. Vision Image Understanding* **106**(2), 162–182 (2007).
13. C.-Y. Chen and W. Wolf, "Background modeling and object tracking using multi-spectral sensors," in *Proc. of the 4th ACM Int. Workshop on Video Surveillance and Sensor Networks*, pp. 27–34, ACM (2006).
14. A. Leykin and R. Hammoud, "Pedestrian tracking by fusion of thermal-visible surveillance videos," *Mach. Vision Appl.* **21**(4), 587–595 (2010).
15. H. Torresan et al., "Advanced surveillance systems: combining video and thermal imagery for pedestrian detection," *Proc. SPIE* **5405**, 506–515 (2004).
16. Z.-J. Feng et al., "Infrared target detection and location for visual surveillance using fusion scheme of visible and infrared images," *Math. Prob. Eng.* **2013**, 720979 (2013).
17. L. Xiaoqi et al., "The infrared and visible image fusion algorithm based on target separation and sparse representation," *Infrared Phys. Technol.* **67**, 397–407 (2014).
18. D. Scribner, P. Warren, and J. Schuler, "Extending color vision methods to bands beyond the visible," in *Proc. IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS'99)*, pp. 33–40, IEEE (1999).
19. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**(2), 91–110 (2004).
20. J. S. Sanders and S. D. Brown, "Utilization of DIRSIG in support of real-time infrared scene generation," *Proc. SPIE* **4029**, 278–285 (2000).
21. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, IEEE (1999).
22. P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, 1st ed., P. Remagnino et al., Eds., Vol. 1, pp. 135–144, Springer, New York City (2002).
23. Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. of the 17th Int. Conf. on Pattern Recognition (ICPR) 2004*, Vol. 2, pp. 28–31, IEEE (2004).
24. L. Chang et al., "FPGA-based detection of sift interest keypoints," *Mach. Vision Appl.* **24**(2), 371–392 (2013).
25. H. Fassold and J. Rosner, "A real-time GPU implementation of the sift algorithm for largescale video analysis tasks," *Proc. SPIE* **9400**, 940007 (2015).
26. C. Wu, "SiftGPU: a GPU implementation of scale invariant feature transform (SIFT)" (2007).
27. S. N. Sinha et al., "Feature tracking and matching in video using programmable graphics hardware," *Mach. Vision Appl.* **22**(1), 207–217 (2011).
28. A. Berk, L. S. Bernstein, and D. C. Robertson, "MODTRAN: a moderate resolution model for lowtran," Technical Report SSI-TR-154,

DTIC Document, Geophysics Laboratory, Air Force Systems Command, Massachusetts (1987).
29. K. Johnson et al., "Muses: a new heat and signature management design tool for virtual prototyping," in *Proc. of Ninth Annual Ground Target Modeling and Validation Conf.*, Houghton, Michigan (1998).
30. M. Behrisch et al., "Sumo-simulation of urban mobility-an overview," in *SIMUL 2011, The Third Int. Conf. on Advances in System Simulation*, pp. 55–60 (2011).

Casey D. Demars received his BS and MS degrees in electrical engineering from Michigan Technological University in 2010 and 2012, respectively, and currently he is pursuing his PhD in electrical engineering from Michigan Technological University. From 2010 to 2012, he was employed by Calumet Electronics Corporation as the technical lead in establishing prototype and testing capabilities for optical polymer waveguides. His PhD research focus is on detection and tracking by exploiting heterogeneous sensor data, particularly multispectral imagery. He is a student member of IEEE and SPIE.

Michael C. Roggemann is a professor of electrical engineering at Michigan Technological University. He joined Michigan Tech in 1997. Prior to joining Michigan Tech, he was a USAF officer who is now honorably retired. He took his BSEE degree in 1982 from Iowa State University, and his MSEE and PhD degrees from the Air Force Institute of Technology in 1983 and 1989, respectively. He is a fellow of the SPIE and OSA.

Timothy C. Havens received his MS degree in electrical engineering from Michigan Technological University, Houghton, in 2000 and the PhD degree in electrical and computer engineering from the University of Missouri, Columbia, in 2010. He was an associate technical staff with MIT Lincoln Laboratory from 2000-2006. He is currently the William and Gloria Jackson assistant professor with the Departments of Electrical and Computer Engineering and Computer Science at Michigan Technological University.