Detection of Buried Objects in FLIR Imaging Using mathematical morphology and SVM

Mihail Popescu, Alex Paino
Electrical and Computer Engineering Dept.
University of Missouri
Columbia, Missouri, USA
popescum@missouri.edu, atpgh6@mail.missouri.edu

Kevin Stone, James M. Keller
Electrical and Computer Engineering Dept.
University of Missouri,
Columbia, Missouri, USA
kes25c@mail.missouri.edu, kellerj@missouri.edu

***Abstract*—In this paper we describe a method for detecting buried objects of interest using a forward looking infrared camera (FLIR) installed on a moving vehicle. Infrared (IR) detection of buried targets is based on the thermal gradient between the object and the surrounding soil. The processing of FILR images consists in a spot-finding procedure that includes edge detection, opening and closing. Each spot is then described using texture features such as histogram of gradients (HOG) and local binary patterns (LBP) and assigned a target confidence using a support vector machine (SVM) classifier. Next, each spot together with its confidence is projected and summed in the UTM space. To validate our approach, we present results obtained on 6 one mile long runs recorded with a long wave IR (LWIR) camera installed on a moving vehicle.**

Keywords-forward looking infrared imaging; FLIR; IR; SVM; mathematical morphology; HOG; LBP

I. INTRODUCTION

Detection of buried targets continues to represent a challenging problem and multiple sensing modalities such as ground penetrating radar, metal detectors and infrared (IR) have been employed to address it. Among the variety sensors used to detect buried objects, IR is gaining in popularity due to the recent advances in un-cooled camera technology and multiband sensors.

Infrared cameras have been used in a variety of applications such as ground target recognition [1, 2], flying target tracking [3], remote sensing [4] and landmine detection [5-9]. IR object detection is based on the temperature difference between the target and the surrounding background. More specifically, the detection of buried objects is based on differences in soil texture, spectral composition and temperature between the area above the target and background [5].

In this paper we use a long wave IR (LWIR) camera installed in front of a vehicle (see Fig. 1) to detect objects buried in the road ahead. The vehicle drives on a road that has buried targets which exhibit certain IR signatures. Our goal is to cue the driver for possible buried objects of interest. While our ultimate goal is to cue the driver in real time, in this paper we present only a retrospective (offline) IR video processing methodology.

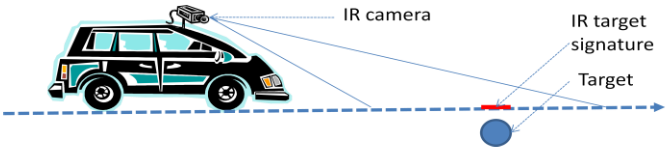


Figure 1. LWIR camera mounted in front of a moving vehicle to detect buried targets based on their IR signature

In previous papers [7-9] we presented several solutions to this problem. Our overall approach to FLIR detection has been to develop multiple target finder algorithms and then use classifier fusion in order to reduce the false alarm rate. The method presented in this paper extends the work presented in [9] and tries to address two challenges previously encountered which are the variability of the IR signature with the time of the day and the time fusion algorithm. The variability of the target signature is demonstrated in Fig. 2 and consists in the fact that both target and non-target signatures look differently depending on the time of the day and the outside temperature. For example, in the image from Fig. 2.a (taken at 11 am) the target is lighter than the background whereas in Fig 2.b (taken at 7 am) the opposite is true.

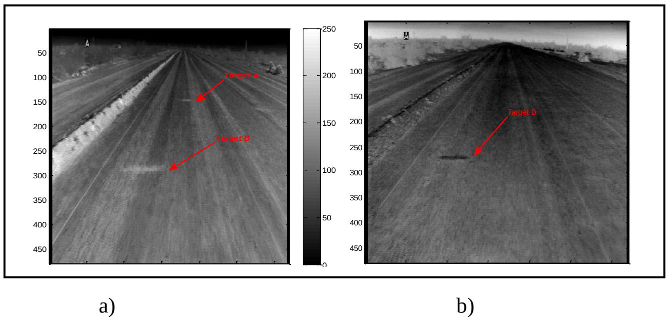


Figure 2. Typical vehicle mounted IR images taken at different time of the day: a) at 11 am, b) at 7 am

To address this challenge we propose in this paper a spot detection algorithm based on the Canny edge detector and mathematical morphology. Mathematical morphology was previously used in target recognition problems [10, 11].

The second challenge is represented by the necessity to correlate hits form consecutive frames. Here, a time fusion algorithm is required in order to recognize that hits from consecutive frames might represent the same object. In this paper we propose a temporal fusion methodology that projects the hits in the Universal Traverse Mercator (UTM) real road coordinates.

The rest of the paper is structured as follows: in section 2 we present the available dataset, in section 3 we describe the entire algorithm used for buried target detection, section 4 shows results obtained on the available datasets and in section 5 we provide conclusions and future work.

II. Available Dataset

The data used in this paper was collected on a country road about one mile long with a LWIR camera installed on a car (see Fig. 1). The road had 50 buried targets. We collected data from 6 runs on the one mile road. Typical frames from the collected movies are shown in Fig. 2. We can see two targets in Fig. 2.a and one target in Fig. 2.b. It is interesting to note that, since the two frames were taken at different time of the day, not only the target signatures differ between them but also the sky and the left roadside berm appearance.

Each run had a frame level ground truth, that is the (x,y) location in each frame of each of the 50 buried targets. If the detected spot (see next section) in a given frame had a ground truth located in its bounding box, it was declared a hit; otherwise it was declared a false alarm. The UTM location of the 50 buried targets was also known.

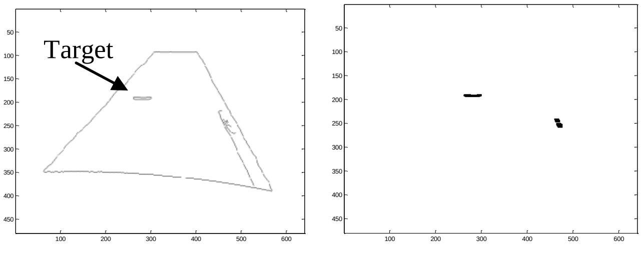
III. Buried Target Detection Algorithm

The buried target detection algorithm has four parts: spot detection, spot classification, temporal fusion and scoring.

A. *Spot detection algorithm*

The main steps of the spot finding algorithm are:

1. Find edges. Apply Canny edge detector to each IR frame. Sample output for this step is shown in Fig. 3.a where a target can be seen. The trapezoidal shape present in Fig. 3.a is an artifact of the fixed road map used for spot search. A dynamic road map based on road finding algorithm is presented in [12].



a. Canny edge detector b. Closing-Opening

Figure 3. Two steps of the spot finding algorithm

2. Close. Apply morphological closing with a 3×11 structuring element, SE.

Morphological binary image processing has two fundamental operations: erosion and dilation. If we denote by A the set of non-zero pixels of a binary image I, we can define the erosion of image I by structuring element (kernel) SE as the locus of the center of SE when this moves inside A. Essentially, as a result of erosion a binary shape A present in I becomes “skinnier”. Conversely, dilation of I by SE is defined as the locus of all point in SE when the center of SE moves inside A. Consequently, as a result of dilation a binary shape A becomes “fatter”. Using these two fundamental operations we can define other morphological operators such as closing, Closing(I, SE)=Erosion(Dilation, SE), SE), and opening, Opening(I, SE)=Dilation(Erosion, SE), SE). Note that while after erosion and dilation the size of the shape changes, after opening and closing the “desired” objects (targets in our case) remain unchanged in size. Intuitively, closing has the role of closing holes smaller than SE, while opening has the role of removing objects smaller than SE. In our case, in step 2 we close all contours that might belong to targets. Note in Fig. 3.a how two closed contours were found: one produced by a target and another one by a false alarm.

3. Open. Apply morphological opening with the same SE to remove noise smaller than SE. Sample output is shown in Fig. 3.b. Note how the trapezoidal artifact and all other open contours (lines) was removed by opening.

4. Connected components. Find the connected components in the image and compute various properties for each of them such as bounding box, centroid, orientation, etc. Of special importance is finding the bounding box that will next be used to compute the spot features. This is an important property of the proposed algorithm and distinguishes this paper from [9] where the window used for feature computation was fixed and determined by trial-and-error.

B. *Spot classification*

The spot classification procedure has the following steps:

1. Feature extraction. For each spot we computed local binary pattern (LBP) [13] and histogram of oriented gradients (HOG) features.

LBP is a texture descriptor defined on *P* neighbors situated at radius *R* from a center pixel, $C=(x_c, y_c)$. If the center pixel *C* has a gray level denoted by g_c, then its $\mathrm{LPB}_{P,R}$ value is:

$$\mathrm{LPB(C)} = \sum_{i=0}^{P-1} \mathrm{s(g_i - g_c)2^i},\ \mathrm{s(x{\geq}0)=1,\ s(x{<}0)=0}, \quad (5)$$

where g_i are the gray levels of its *P* neighbors. In other words, the *P* neighbors are assigned a value of 0 or 1, depending whether are smaller or greater than C, respectively. The resulting bit pattern is then mapped into a number. In our case, we use *R*=1 and *P*=8 which results in possible LPB values between 0 and 255. To represent a spot hit by LPB features we simply compute the histogram of all the LPB values obtained for the pixels in its bounding box related to its centroid, C. If we used all LPB values, the histogram (hence the feature space dimension) will have 256 bins. However, if we only use the uniform LPB patterns (patterns that have at most two 0-1 or 1-0 transitions) from the 256 possible ones, the feature space reduces to dimension 59 (number used in this paper).

HOG is a texture descriptor that computes the occurrences of gradient orientation in a given image region. The gradient of the region is computed by filtering the image with a [-1,0,1]

Work funded by ARO grant 57940-EV to support U. S. Army RDECOM CERDEC NVESD.

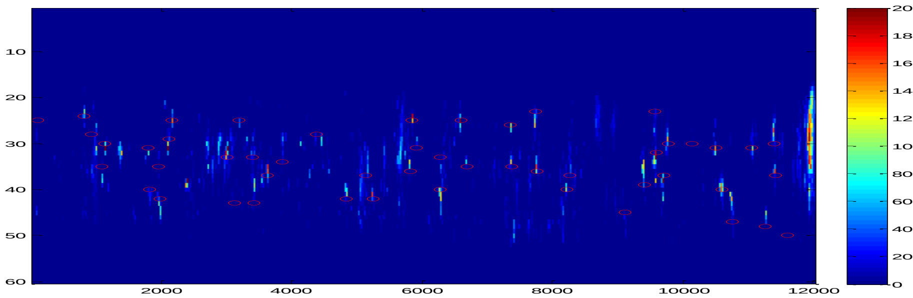


Figure 4. Typical UTM map output of the spot temporal fusion procedure (targets denoted by circles, bright spot denotes target confidence)

kernel to get the horizontal gradients and with a $[-1,0,1]^t$ for the vertical ones. Then the region is divided in $H \times V$ cells and a histogram with B bins is computed in each cell. The final HOG descriptor of the region is obtained by the concatenation of the histograms obtained in all cells. In our case we chose H=3, V=3 and B=9 resulting a HOG features vector of size 81 to describe each bounding box. The total number of features was 140.

2. Spot classification. We used a leave-one-out procedure to train a support vector machine (SVM) to compute the confidence that a spot i is a target in frame j, $c_{ij} \in [0,1]$ $i \in [1,N_j]$ and $j \in [1,M]$ where N_j is the number of spots found in frame j and M and the total number of frames (between 4000 and 6000). More specifically, we trained an SVM with the hits from 5 runs and then test the resulting model on the sixth run. We employed the MATLAB implementation of the LIBSVM library [15] in our experiments.

C. C. Temporal fusion of the image (frame) spots

After we compute the confidences c_{ij} that spots in frame j are targets, we project each spot together with its confidence in the UTM space.

The projection procedure is detailed in [7] and it is based on finding a mapping between the image space and UTM space. A typical output map of the projection procedure is shown in Fig. 4. The ground truth is marked with (red) circles whereas the bright areas denote aggregated spots. The grid used to produce the UTM image is 0.1m×0.1m, which means that the image shown in Fig. 4 corresponds to a surface 6 m wide and about 1,200 m long.

D. Scoring procedure

The scoring was performed on UTM maps similar to the one shown in Fig. 4. The main steps of the scoring procedures are:

1. Thresholding. Threshold the UTM map, O, with a value $T \in (\min(O), \max(O))$. For example, for the map shown in Fig. 4, typical thresholds could be $T \in (0, 200)$.

2. Closing. Each UTM spot might have multiple local maxima that become disconnected when thresholding is applied. To reconnect the parts of the same spot we apply a closing procedure similar to the one from section III.A.2.

3. Connected components. Apply a connected components procedure to the thresholded O, say NC UTM spots are detected.

4. Compute DR and FAR. Count hits, d, that are in a radius of HALO=0.5m from the UTM ground truth location, and compute the detection rate DR=d/50 and false alarm rate FAR=(NC-d)/Area(O).

5. Compute ROC. Compute a receiver operator characteristic (ROC) curve by choosing various thresholds T and repeating steps 2-4 given above.

IV. Results

Applying the algorithms described in the previous section we obtained the following results on the 6 runs mentioned in section II.

A. Spot extraction and temporal fusion results

In comparison to our previous work [9], here we changed target representation from corners to spots (section III.A) and the temporal fusion procedure based on the mean shift algorithm [7] by the one described in section III.C. In Fig. 5 we compare the results obtained with the new and the old [9] approach for two of the runs from our dataset.

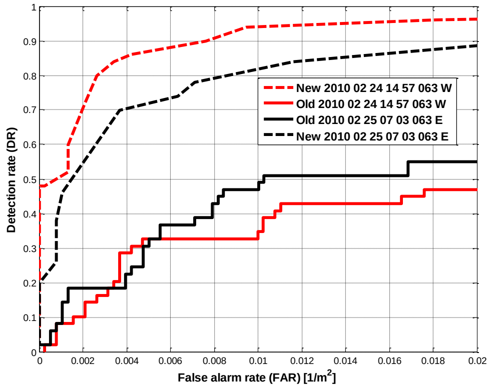


Figure 5. Comparison between spot based temporal fusion (New) and the mean shift approach [7] (Old) for two runs

From Fig. 5 we see that the new approach (dotted lines) produces greatly improved the results. There are multiple reasons for this drastic improvement. First, the corner representation of targets while suitable for image registration, introduce too many false alarms in target recognition problems. The reason is that many times, a target may have multiple corners selected as possible candidate hits, some of them being far from the ground truth location. Second, when corners are considered, it is hard to choose a window for feature extraction around a corner that accounts for image perspective. Third, the new temporal spot merging approach is more forgiving and less prone to false alarms like the one based on mean shift.

B. *The choice of SVM kernel*

Many times, when SVM are used, only the default (linear or polynomial) kernels are investigated. In Fig. 6 we show the results on two of the available runs for two choice of SVM kernel: linear and radial basis function (RBF). As we can see from Fig. 6, the change from linear (solid line) to RBF kernel (dotted) produced a significant performance improvement.

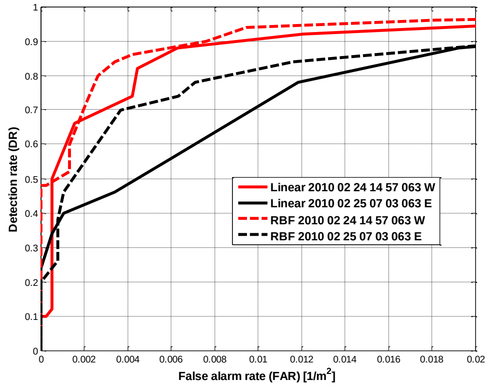


Figure 6. Comparison between a linear and a RBF kernel.

C. *Results for the entire dataset*

The results for the entire dataset are presented in Fig. 7. The results were obtained using a leave-one-run-out cross-validation approach (train on 5 runs and test on the 6th).

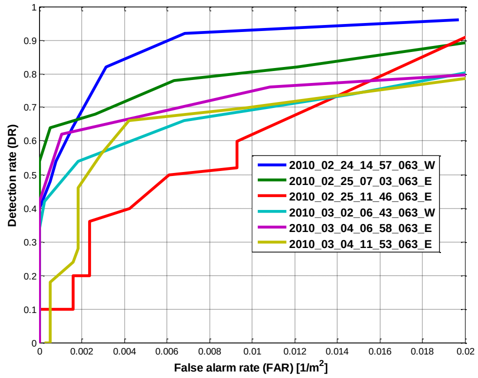


Figure 7. Results on the entire dataset

From Fig. 7 we see that we obtained at least 80% detection at a FAR of 0.02 1/m^2. We mention that a human observer that scored several of the runs without having any knowledge about the ground truth obtained around 80% detection and no false alarms hence there is still room for improvement in false alarm reduction. Even if the results obtained are reasonable, they are not uniform across the entire dataset (see the red line "2010_02_26_11_48_063_E") for reasons that we are not entirely sure. However, the good news is that neither the time of the day or the humidity of the run (last three runs were recorded after rain) seem to be the cause for the decrease in performance.

V. Conclusions

In this paper we presented a buried target recognition algorithm for FLIR imagery based on mathematical morphology and support vector machines. We found that the temporal fusion algorithm of the individual frame hits is very important for increasing the detection performance. Employing a radial basis function kernel may also lead to significant increase in classification performance if enough data is available. On a set of 6 one mile long runs we obtained a performance of about 80% detection at a FAR=0.02 1/m^2.

Acknowledgment

The authors would like to thank Dr. Don Reago, Mr. Pete Howard, Mr. Richard Weaver and Mr. Bob Luke at NVESD and Dr. Russell Harmon at ARO.

References

[1] Li B., Chellapa R., Zheng Q., Der S., Nasrabadi N., Chan L., Wang L., "Experimental Evaluation of FLIR ATR Approaches—A Comparative Study", *Computer Vision and Image Understanding* 84, 5–24 (2001).

[2] Stone, K., Keller, J. M., Popescu, M., Havens, T.C., Ho, K. C., "Forward Looking Anomaly Detection via Fusion of Infrared and Color Imagery", Proc. of SPIE 7664, 2010.

[3] Yu Y., Guo L., "Infrared Small Moving Target Detection Using Facet Model and Particle Filter", Proc of 2008 Congress on Image and Signal Processing, pp. 206-210.

[4] Schmuggea,T., French, A., Ritchie, JC, Rango A., Pelgrum H., "Temperature and emissivity separation from multispectral thermal infrared observations", Remote Sensing of Environment 79 (2002) 189–198.

[5] Winter,E. M., Fields,D. J., Carter, M. R., Benett, C. L., Lucey,P. G., Hohnson,J. R., Horton,K. A., and Bowman, A. P., "Assessment of Techniques for Airborne Infrared Land Mine Detection", Proc.of the Third International Airborne Remote Sensing Conference and Exhibition, Copenhagen, Environmental Research Institute of Michigan, Ann Arbor, Vol. II, 1997, pp. 44-51.

[6] Thanh N.T., Sahli, H., Hao D.N., "Infrared Thermography for Buried Landmine Detection: Inverse Problem Setting," *IEEE Transactions on Geoscience and Remote Sensing*, vol.46, no.12, pp.3987-4004, Dec. 2008.

[7] Stone K., Keller M., Popescu M., Spain C.J., "Buried explosive hazard detection using FLIR imagery", proceedings of SPIE Defense, Orlando, FL, April 24-29, 2011.

[8] Spain C.J., Popescu M., Keller J., Stone K., "Automatic Detection of targets in medium-wave Infrared Imagery using adaptive background mixture models", proceedings of SPIE Defense, Orlando, FL, April 24-29, 2011.

[9] Popescu M., Stone K., Keller J., "Detection of targets in forward-looking infrared imaging using a multiple instance learning framework", proceedings of SPIE Defense, Orlando, FL, April 24-29, 2011.

[10] Banerji, A., Goutsias, J., "A morphological approach to automatic mine detection problems", IEEE Trans Aero Elect Sys 34:1085–1096, 1998.

[11] Chang'an W., Shouda J., "Automatic Target Detection and tracking in FLIR Image sequences using Morphological connected operator", IIH-MSP, 2008, pp. 414-417.

[12] Lewis D., Keller J., "Automatic road finding in FLIR", CISDA 2012, Otawa, Canada, 2012.

[13] Heikkilä, M. and Pietikäinen, M. (2006), “A Texture-Based Method for Modeling the Background and Detecting Moving Objects”, IEEE Trans. Pattern Analysis and Machine Intelligence 28(4):657-662.

[14] Ludwig O., Delgado D., Goncalves V., and Nunes U., “Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,” In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009, St. Louis, 2009. V. 1. P. 432-437.

[15] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm