Scale-Insensitive Detection Algorithm for FLIR Imagery

Sandro Der, Chris Dwan, Alex Chan, Heesung Kwon, and Nasser Nasrabadi

ARL-TN-175

February 2001

Approved for public release: distribution unlimited.
The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.
Scale-Insensitive Detection Algorithm for FLIR Imagery

Sandor Der, Chris Dwan, Alex Chan, Heesung Kwon, and Nasser Nasrabadi
Sensors and Electron Devices Directorate

Approved for public release; distribution unlimited.
Abstract

This report describes an algorithm for detecting military vehicles in FLIR imagery that will be used as a prescreener to eliminate large areas of the image from further analysis. The output is a list of likely target locations with confidence numbers to be sent to a more complex clutter-rejection algorithm for analysis. The algorithm uses simple features and is intended to be applicable to a wide variety of target-sensor geometries, sensor configurations, and applications.
Contents

1 Introduction 1

2 Data 2

3 Features 3
   3.1 Maximum Grey Level, Feature 0 3
   3.2 Contrastbox, Feature 1 3
   3.3 Average Gradient Strength, Feature 2 4
   3.4 Local Variation, Feature 3 4
   3.5 How Features Were Selected 4

4 Combining Features 6

5 Experimental Results 7

6 Conclusions and Future Work 13

References 14

Distribution 15

Report Documentation Page 17

Figures

1 ROC curve on Hunter Liggett April 1992 imagery 8
2 ROC curve on Yuma July 1992 imagery 8
3 ROC curve on Greyling August 1992 imagery 9
4 Easy image from Hunter Liggett April 1992 dataset 9
5 Results on image in figure 4 10
6 Moderate image from Hunter Liggett April 1992 dataset 11
7 Results on image in figure 6 12
1. Introduction

We designed the algorithm described in this report to address the need for a detection algorithm that could serve as a prescreener/detector for a broad number of applications. While most automatic target detection/recognition (ATD/R) algorithms use much problem-specific knowledge to improve performance, the result is an algorithm that is tailored to specific target types and poses. The approximate range to target is often required, with varying amounts of tolerance. For example, in some scenarios, it is assumed that the range is known to within one meter from a laser range finder or a digital map. In other scenarios, only the range to the center of the field of view and the depression angle is known, so that a flat-earth approximation provides the best estimate. Many algorithms, both model-based and learning-based, required either accurate range information or compensate for inaccurate information by attempting to detect targets at a number of different ranges within the tolerance of the range. Because many such algorithms are quite sensitive to scale, even a modest range tolerance requires that the algorithm attempt to match at a large number of closely spaced scales, driving up both the computational complexity and the false alarm rate. Algorithms have often used view-based neural networks [1-3] or statistical methods [4].

The proximate motivation for developing the scale-insensitive algorithm was to provide a fast prescreener for a robotic application for which no range information was available. Instead, the algorithm attempted to find targets at all ranges between some reasonable minimum, determined from operational requirements, and the maximum effective range of the sensor.

Another motivation was to develop an algorithm that could be applied to a wide variety of image sets and sensor types without the severe degradation in performance that commonly occurs with learning algorithms, such as neural networks and principal component analysis-based methods, that have been trained on a limited variety of sensor types, terrain types, and environmental conditions. While we recognize that with a suitable training set, learning algorithms will often perform better than other methods, such a scenario typically requires a large and expensive training set, which is sometimes not feasible.
2. Data

The dataset used in training and testing this system was the April 1992 Comanche forward looking infrared (FLIR) collection at Ft. Hunter Liggett, CA. This dataset consists of 1225 images, each of which is 720 by 480 pixels. Each image has a field of view of approximately 1.75 degrees squared.

Each image contains one or two targets in a hilly wooded background. Ground truth was available that provided target centroid, range to target, target type, target aspect, range to center of field of view, and depression angle. The target centroid and range to target were used to score the algorithm, as described in the experimental results section, but none of the target-specific information was used in the testing process. The algorithm assumes that only the vertical and horizontal fields of view and the pixel geometry are known. The only range information used is the operational minimum range and the maximum effective range of the sensor.
3. Features

Each feature is calculated for every pixel in the image. As more complex features are added in the future, it might become beneficial to calculate some of the features only at those locations for which the other feature values are high. While each feature assumes knowledge of the range to determine approximate target size, these features are not highly range sensitive. The algorithm calculates each feature at coarsely sampled ranges between the minimum and maximum allowed range.

Each feature described below was chosen based on intuition, with the criteria that they be monotonic and computationally simple. The features are described in decreasing order of importance.

3.1 Maximum Grey Level, Feature 0

The maximum grey level is the highest grey level within a roughly target-sized rectangle centered on the pixel. We chose it because in many FLIR images of vehicles, a few pixels are significantly hotter than the rest of the target or the background. These pixels are usually on the engine, the exhaust manifold, or the exhaust pipe. The feature is calculated as

$$F_{i,j}^0 = \max_{(k,l) \in N_{in}(i,j)} f(k,l),$$

where $f(k,l)$ is the grey-level value of the pixel in the $k$th row and $l$th column; $N_{in}(i,j)$ is the neighborhood of the pixel $(i,j)$ defined as a rectangle whose width is the length of the longest vehicle in the target set and whose height is that of the tallest vehicle in the target set. For the applications we have considered, the width is 7 m and the height, 3 m.

3.2 Contrastbox, Feature 1

The contrastbox feature measures the average grey level over a target-sized region and compares it to the grey level of the local background. We chose this feature because many pixels that are not on the engine or on other particularly hot portions of the target are still somewhat warmer than the natural background. This feature has been used by a large number of authors and is calculated as

$$F_{i,j}^1 = \frac{1}{n_{in}} \sum_{(k,l) \in N_{in}(i,j)} f(k,l) - \frac{1}{n_{out}} \sum_{(k,l) \in N_{out}(i,j)} f(k,l),$$

where $n_{out}$ is the number of pixels in $N_{out}(i,j)$, $n_{in}$ is the number of pixels in $N_{in}(i,j)$, and $N_{in}(i,j)$ is the target-sized neighborhood defined above. The neighborhood $N_{out}(i,j)$ contains all of the pixels in a larger rectangle around $(i,j)$, except those pixels in $N_{in}(i,j)$.
3.3 Average Gradient Strength, Feature 2

We chose the gradient-strength feature because manmade objects tend to show sharper internal detail than natural objects do, even when the average intensity is similar. To prevent large regions of background that show higher than normal variation from showing a high value for this feature, we subtract the average gradient strength of the local background from the average gradient strength of the target-sized region. The feature is calculated as

\[ F_{i,j}^{2} = \frac{1}{n_{in}} \sum_{(k,l) \in N_{in}(i,j)} G_{in}(i,j) - \frac{1}{n_{out}} \sum_{(k,l) \in N_{out}(i,j)} G_{out}(i,j) , \]  

where

\[ G_{in}(i,j) = G_{in}^{h}(i,j) + G_{in}^{v}(i,j) , \]

\[ G_{in}^{h}(i,j) = \sum_{(i,j) \in N_{in}} |f(i,j) - f(i,j + 1)| , \]

\[ G_{in}^{v}(i,j) = \sum_{(i,j) \in N_{in}} |f(i,j) - f(i + 1,j)| . \]

3.4 Local Variation, Feature 3

The local variation feature is calculated as

\[ F_{i,j}^{3} = \frac{1}{n_{in}} \sum_{(k,l) \in N_{in}(i,j)} L_{in}(i,j) - \frac{1}{n_{out}} \sum_{(k,l) \in N_{out}(i,j)} L_{out}(i,j) , \]

where

\[ L_{in}(i,j) = \sum_{(k,l) \in N_{in}(i,j)} |f(k,l) - \mu_{in}(i,j)| \]

and

\[ \mu_{in}(i,j) = \frac{1}{n_{in}} \sum_{(k,l) \in N_{in}(i,j)} f(k,l) . \]

3.5 How Features Were Selected

A full description of the feature selection is outside the scope of this report. We programmed a large number of features and calculated the value of these features over a large number of randomly selected pixels in the images of the training set. We also calculated the feature values at the ground truth location of the targets. We computed histograms for each feature for...
both the target and background pixels and calculated a measure of separability. We also calculated the correlation of the features to avoid choosing several features that are similar. Some of the features were highly correlated, which was expected because one of the purposes of the training was to determine which of similar features provided the greatest separability. For example, a number of contrast features were used, which normalized the target and background values by local standard deviation of the background, or of the target, or neither. Similarly, a number of gradient-strength features were calculated. The feature-pruning process was ad hoc; thus it would be reasonable to expect that performance improvement could be obtained by the use of a more rigorous approach.
4. Combining Features

Each feature is normalized across the image so that the feature value at each pixel represents the number of standard deviations that the pixel stands apart from the values for the same feature across the image. Thus the feature image for the $m$th feature is normalized as

$$F_{i,j}^{m,N} = \frac{F_{i,j}^{m} - \mu_{m}}{\sigma_{m}},$$

(10)

where

$$\mu_{m} = \frac{1}{M} \sum_{all(k,l)} F_{k,l}^{m},$$

(11)

and

$$\sigma_{m} = \frac{1}{M} \sum_{all(k,l)} (F_{k,l}^{m} - \mu_{m})^{2}.$$

(12)

After normalization, the features, each of which is calculated for each pixel, are linearly combined into a confidence image,

$$G_{i,j} = \sum_{m=0}^{3} \omega_{m} F_{i,j}^{m,N},$$

(13)

where the feature weights $\omega_{m}$ are determined with the use of an algorithm not described here. The confidence value of each pixel is mapped by a scaling function $S : \mathbb{R} \rightarrow [0, 1]$, as

$$S(G_{i,j}) = 1 - e^{\alpha G_{i,j}},$$

(14)

where $\alpha$ is a constant.

This scaling does not change the relative value of the various pixels; it merely scales them to the interval $[0, 1]$ for convenience. Confidence numbers are often limited to this interval because they are estimates of the a posteriori probability. While this is not true for our algorithm, the use of this interval is convenient for evaluators.

To determine the detection locations from the scaled confidence image, we choose the pixel value with the maximum confidence value. Then a target-sized neighborhood around the image is set to zero so that the search for subsequent detections will not choose a pixel location corresponding to the same target. The process is repeated an integer number of times, where the integer is chosen a priori.
5. Experimental Results

The training results on the Hunter Liggett April 1992 ROI database are shown in the receiver operating characteristics curve in figure 1. Figure 2 shows test results on the February 1992 ROI database collected at Yuma Proving Ground (YPG), and figure 3 shows test results on the Greyling August 1992 ROI database. The Yuma test data are much more difficult because they were taken in the desert in July, so many locations in the image have a higher apparent temperature than that of the targets. The data from Greyling, Michigan are significantly easier because the temperatures are milder, and the data are comparable in difficulty to the training data. Note that no training data were used from anywhere but Hunter Liggett, so the results suggest that the algorithm is not sensitive to the training background. This is not surprising given the simplicity of the algorithm. However, learning algorithms are often sensitive to training background. Figures 4 and 5 show a sample image and the results of the algorithm on the image. The crosses in figure 5 denote the ground-truth targets, and the x’s denote the detections on the targets. Detections are designated hits if the detection center falls anywhere on the actual target; otherwise, they are designated as false alarms. The top three detections, ranked by confidence number, are designated on the image. The top two detections are hits, while the third falls near the target and is designated a false alarm. Figures 6 and 7 show another somewhat more difficult image and associated algorithm results. The top detection falls on a target in the bottom left of the image, while the second highest detection is a false alarm near the center of the image. Although the location looks like a possible target, it is merely a warm spot on the dirt road.

The algorithm, with relatively minor modifications, has been used by the Demo III unmanned ground vehicle (UGV) program to reduce the amount of imagery that must be transmitted via radio link to a human user. It will also be used by the Sensors for UGV program at the Night Vision and Electronic Sensors Directorate to prescreen uncooled FLIR imagery and to indicate potential targets that should be looked at more closely with an active laser sensor. This algorithm has been used as a synthetic image-validation tool by measuring the performance of the algorithm in comparison to real imagery.
Figure 1. ROC curve on Hunter Liggett April 1992 imagery. Horizontal axis gives average number of false alarms per frame. Vertical axis is target-detection probability.

Figure 2. ROC curve on Yuma July 1992 imagery.
Figure 3. ROC curve on Greyling August 1992 imagery.

Figure 4. Easy image from Hunter Liggett April 1992 dataset.
Figure 5. Results on image in figure 4.
Figure 6. Moderate image from Hunter Liggett April 1992 dataset.
Figure 7. Results on image in figure 6.
6. Conclusions and Future Work

Future work might include a more systematic evaluation of potential features and an improved classification scheme that allows useful features that appear rarely to be incorporated. In a small minority of FLIR images of targets, a windshield will reflect cold sky, causing a few pixels to be extremely dark. The current scheme is not set up to incorporate such features because the weighting would be quite low since the feature is seldom useful.
References


Palisades Inst for Rsrch Svc Inc
ATTN E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402
Director
US Army Rsrch Lab
ATTN AMSRL-RO-D JCI Chang
ATTN AMSRL-RO-EN W D Bach
PO Box 12211
Research Triangle Park NC 27709

US Army Rsrch Lab
ATTN AMSRL-CI-AI-R Mail & Records Mgmt
ATTN AMSRL-CI-AP Techl Pub (2 copies)
ATTN AMSRL-CI-LL Techl Lib (2 copies)
ATTN AMSRL-D D R Smith
ATTN AMSRL-DD J M Miller
ATTN AMSRL-SE-SE H Kwon
ATTN AMSRL-SE-SE L A Chan
ATTN AMSRL-SE-SE N Nasrabadi
ATTN AMSRL-SE-SE P Gillespie
ATTN AMSRL-SE-SE S Der (20 copies)
ATTN AMSRL-SE-SR G Stolovy
Adelphi MD 20783-1197
# Scale-Insensitive Detection Algorithm for FLIR Imagery

**Authors:** Sandor Der, Chris Dwan, Alex Chan, Heesung Kwon, and Nasser Nasrabadi

**Performing Organization:** U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

**Sponsoring/Monitoring Agency:** U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

**Abstract:** This report describes an algorithm for detecting military vehicles in FLIR imagery that will be used as a prescreener to eliminate large areas of the image from further analysis. The output is a list of likely target locations with confidence numbers to be sent to a more complex clutter-rejection algorithm for analysis. The algorithm uses simple features and is intended to be applicable to a wide variety of target-sensor geometries, sensor configurations, and applications.