Synthetic Discriminant Function
Performance Versus Filter Count

Michael J. Vrabel

ARL-TN-157

April 2000

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.
Synthetic Discriminant Function Performance Versus Filter Count

Michael J. Vrabel
Sensors and Electron Devices Directorate

Sponsored by
AMCOM*
Redstone Arsenal, AL 35898-5240

*Also sponsored by the U.S. Army Research Laboratory.
Abstract

The relationship between performance and filter count is examined for a synthetic discriminant function-based target-detection algorithm.
## Contents

1 Introduction 1  
2 Clustering 2  
3 Results 4  
4 Conclusion 8  
References 9  
Appendix. Attributes—Image Set and SDF Codes 11  
Distribution 15  
Report Documentation Page 17
Figures

1  Example scenes with targets indicated by crosshairs . . . . .  5
2  Target-detection rate versus false alarms . . . . . . . . . .  6
3  Threshold versus false-alarm and target-detection rates . . .  7
This report examines the relationship between the performance of the synthetic discriminant function (SDF) approach to automatic target recognition (ATR) and SDF filter count. The SDF approach to ATR is relatively mature, with much research published on various aspects of this concept appearing in the literature [1–7]. While a classification capability presently exists, I do not include it in this report. Consequently, this report is to be restricted to target detection only.
2. Clustering

To develop a set of SDF filters, one must have a reasonable method for organizing target images (the basis for the filters) based on some measure of target image similarity. For present purposes, all images are masked scene size (128 x 128 pixels) representations with varying-sized centered targets. One obvious and reasonable basis for clustering images is the Euclidean distance between images in gray-scale pixel space. The images used to construct the filters can be represented by a set of reference vectors, one for each filter, that minimizes the following cost function:

\[ \varepsilon = \sum_{i} \sum_{x \in \xi_i} \| x - c_i \|^2. \]  

(1)

where \( x \) is an image vector, \( c_i \) is a reference vector for cluster \( \xi_i \), and \( x \in \xi_i \) if

\[ \| x - c_i \| < \| x - c_j \|, j \neq i. \]  

(2)

An obvious approach to a solution is the well-known k-means algorithm [8]. Traditional k-means suffers from problems addressed by a number of authors. The algorithm to be used in this report is the latest and probably the best iteration on this concept [9]. It is referred to as "optimal adaptive k-means." In adaptive k-means, the k-means uses an iterative approach to finding a set of optimum reference vectors. An updated vector is computed as

\[ c_{k,T+1} = c_{k,T} + M_k \nu(x_T - c_{k,T}). \]  

(3)

where \( \nu \) is a constant governing the learning rate and \( M_k \) is defined as

\[ M_k = 1 \quad \text{if} \quad v_k(\| x - c_k \|^2) \leq v_i(\| x - c_i \|^2) \quad i \neq k; \]  

(4)

\[ M_k = 0 \quad \text{otherwise}. \]

To obtain the within-region variation \( v_k \), one calculates

\[ v_{k,T+1} = \alpha v_{k,T} + (1 - \alpha)(M_k \| x_T - c_{k,T} \|^2), \]  

(5)

with \( \alpha \) as a constant selected to be slightly less than 1.

One potential problem suffered by all k-means algorithms is the inability to define initial reference vectors (with its potential consequence on finding
a globally optimum solution). Rather than selecting initial reference vectors randomly, I used a recurrent neural network to perform this task. It has been shown that the Hopfield neural network [10] can be used to solve various problems in optimization [11–14]. Its properties are detailed elsewhere [10]. In the following, I briefly summarize its attributes. The asynchronously updated state of each neural network node is

\[ V_n = \Gamma \left( \sum_{m=1}^{N} W_{nm} V_m \right), \]  

(6)

where

- \( V_m \) is the present state of node \( m \),
- \( N \) is the node count,
- \( W_{nm} \) is the weight from node \( n \) to \( m \), and
- \( \Gamma \) is a hard thresholding function. If the following restrictions are imposed upon the weights,

\[ W_{nm} = W_{mn} \text{ and } W_{mm} = 0, \]

then an energy function can be associated with the network (assuming a zero externally applied threshold):

\[ E = -1/2 \sum_{n=1}^{N} \sum_{m=1}^{N} W_{nm} V_n V_m. \]  

(7)

An asynchronous network update guarantees an eventual settling into a final invariant state for all nodes—a state representing either a global or local energy minimum. Note from equations (6) and (7) that a two-population distribution will be selected that minimizes the absolute values of the weights within each population and maximizes them between populations. The resultant vectors have just the attributes required for initializing \( k \)-means. In this instance, the weights are equated with an average difference in gray-scale pixel values among the filter-generating images. The neural network divides the image population into two clusters. To increase the population cluster count, one determines the standard deviation of each cluster and subdivides the cluster with the largest deviation using, once more, the Hopfield neural network. In this way, the initial population can be divided into as many clusters as desired.
3. Results

The purpose of this work is to explore the relationships among four significant variables: (1) filter count, (2) false-alarm rate, (3) target-detection rate, and (4) SDF threshold. To test the performance of the SDF algorithm as a function of filter count, I modified the original code* to allow a filter count independent of the target count. I then ran the code with the chosen filter population and combined the results for each filter. The output from each filter was examined for peaks in the response as per standards built into the SDF code, and the magnitude and location of each peak were stored for a maximum of the 60 largest peaks. This meant that a maximum of 480 peak responses could be stored for the maximum tested filter count of eight for each analyzed scene. All peak responses for a given filter set were ordered by magnitude, and the resultant data set was scanned starting at the maximum response. A circle of one of three chosen radii (3, 5, and 10 pixels) was centered on the location of the given peak, and the peaks within the circle were combined two different ways. The first approach was to consider any peak falling within a given circle as coincident with the main peak of that circle with the consequence that all lesser peaks were eliminated from the peak list.

The second approach was to define a location for the within-circle response by averaging the weighted locations of all peaks within the circle. The weighting factor was just the location peak response. Again, the ancillary within-circle peaks were eliminated from the peak response set. After comparing both approaches, I found little difference in the final results. Thus, all subsequent results are to be given for the first (and simpler) approach. A target was considered detected if its location fell within a radius of the peak response location as defined previously. Similarly, when consolidating all background hits falling within the preceding defined circle into a consequent single hit, I assumed that all hits within the circle were detecting the same false background attribute.

Figure 1 contains four examples of the target-containing scenes used to examine filter performance. The images in these scenes were selected from the beginning, middle, and end of the test sequence and represent the whole data set. The crosshairs are centered on the targets. The images are in an as-given state, with no rescaling or image renormalization. A narrow band

*The original version of the SDF code was written by Lipchen (Alex) Chan of ARL.
of pixels appearing about the perimeter of some of the images has an intensity distribution unlike the rest of the scene. For this reason, I discarded all peak responses occurring within six pixels of the scene perimeter. All images are $128 \times 128$ pixels and the test sequence comprises 236 scenes, each containing two target images. Additional details about each of these two images and the images used to create the filter sets are given in the appendix.

False alarm versus target-detection rates are given in figure 2. The graph legend identifies the number of filters for a target-detection rate radius (as previously noted) of 3, 5, and 10 pixels. In general, the larger the filter count the poorer the performance. Figure 3 is the background and target performance as a function of threshold. As can be seen, the relatively modest target-detection rate improvement with filter count increase is more than offset by the corresponding increase in false alarms.

The SDF model has a number of desirable attributes:

- It readily adapts to a changing target environment.
- It appears to extract even low-quality images.
- Speed of execution appears to make the SDF model amenable to a real time implementation.
Figure 2. Target-detection rate versus false alarms. Legends show filter counts of 1 to 4, 6, and 8 and three circle radii (a) 3, (b) 5, and (c) 10 pixels.
Figure 3. Threshold versus false-alarm and target-detection rates. Legends show filter counts of 1, 2, 4, and 8 and circle radii of 3 and 5 pixels.
4. Conclusion

A clustering algorithm was developed to optimally organize a set of correlation filter images for an SDF-based target-detection algorithm into one through eight populations. The resultant set of one through eight filters was tested (target-detection versus false-alarm rate) with the SDF algorithm. It was demonstrated that for the test population of target-containing scenes, the optimum SDF filter count is one.
References


Appendix. Attributes—Image Set and SDF Codes

All filters were constructed from an image data set provided by the U.S. Army Aviation and Missile Command (AMCOM). This set contained 716 images and comprises the following images: 11815041.rl.apc through 11815226 rl.apc and 11816053%~ through 11816288%~, where %~ is one of the following: .m60, .tnk, _rl.m60, or _rl.tnk.

The SDF-based ATR algorithm was tested on the following set of 236 AMCOM-provided scenes: L1816S00053.rl.bin through L1816S00288_ri.bin. Each scene contained two target images: the M60 and tnk.

The following is a flowchart of the codes used for this study, along with the files generated by each code. This chart is followed by a description of the contents of each output file. All source code is available upon request.

- **filter_cluster.c**
  outputs: ↓
  1.: diagnostics
  2.: distance_matrix
  3.: hopfield_clusters
  4.: vector%d.dat
  5.: matrix
  6.: merged_file

- **make_list.c**
  outputs: ↓
  7.: detect_list%d → mv to: test.list

- **sdf.c → a.out -bd 1**
  (change value NOBJECT to filter count)
  outputs: ↓
  8.: test.dfil

- **make_frame_list.c**
  outputs: ↓
  9.: test.list
- **sdf.c** → a.out -td 1
  outputs:
  10.: detection.dat
  11.: images.dat
  12.: sdf_output.dat

- **sdf_imaging.c**
  outputs:
  13.: scene.dat → MATLAB™ imagery

- **sdf_evaluate.c**
  outputs:
  14.: results.%d.%d.dat

- **plot_maker.c** → MATLAB 2D curves

**Output file contents.**

- **diagnostics:** Contains a description of the code performance at every 50 iterations for each cluster. This includes (1) \((c_{k,T} - c_{k,i})/c_{k,i}\), where \(i\) refers to the initial value of the reference vector; (2) the filter image population count for each cluster; and (3) the within-region variation \(\nu_k\).

- **distance_matrix:** Contains the gray-scale distance matrix for the 716 filter images.

- **hopfield_clusters:** Assigns each of the 716 filter images to its appropriate cluster for filter counts of 1 through \(n\), with \(n\) nominally assigned the value 8. This is for the Hopfield clustering model.

- **vector%d.dat:** Contains the 128 × 128 final reference vectors for a cluster count defined by \(\%d\).

- **matrix:** Is averaged distances between the Hopfield cluster populations for cluster count of 1 through, nominally, 8.

- **merged_file:** For the 716 filter images, contains in column format the cluster membership for a 1 through 8 cluster population, along with the image ID.

- **detect_list:** Contains from merged_file the population for a selected cluster count along with the full paths to the individual images.

- **test.dfil:** Is the filter set generated by sdf.c to be used by the detection function of the aforementioned code.

- **test.list:** Contains table of image file names of scenes to be analyzed by the SDF code detector.
detection.dat: Contains the detector response map for a limited number of scenes along with the corresponding target locations.

images.dat: Contains a limited number of the unprocessed scenes used as input to the detector.

sdf_output.dat: Lists the locations of all targets on all scenes passed through the detector along with the top 60 peaks of the detector’s output for each filter.

scene.dat: Contains a square array of image pixels composed of either (multiple) unprocessed scenes or the output of the detector with all targets indicated with crosshairs and in a MATLAB compatible format.

results.%d.%d.dat: Is in a four-column format: first column are threshold values; second column, corresponding first target-detection rate; third column, the second target-detection rate, and fourth column, the per scene background hit count. The first %d is the filter count and the second %d is the aforementioned radius value.
## Distribution

<table>
<thead>
<tr>
<th>Organization</th>
<th>Address Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adminstr</td>
<td>Defns Teclh Info Ctr 8725 John J Kingman Rd Ste 0944 FT Belvoir VA 22060-6218</td>
</tr>
<tr>
<td>Ofc of the Secy of Defns</td>
<td>The Pentagon Washington DC 20301-3080</td>
</tr>
<tr>
<td>Oft of the Secy of Defns</td>
<td>Attn DTIC-OCP 8725 John J Kingman Rd Ste 0944 FT Belvoir VA 22060-6218</td>
</tr>
<tr>
<td>Oft of the Secy of Defns</td>
<td>Attn ODDRE (R&amp;AT) 8725 John J Kingman Rd Ste 0944 FT Belvoir VA 22060-6218</td>
</tr>
<tr>
<td>Oft of the Secy of Defns</td>
<td>Attn OUSD(A&amp;T)/ODDR&amp;E(R) 3080 Defense Pentagon Washington DC 20301-7100</td>
</tr>
<tr>
<td>AMCOM MRDEC</td>
<td>Attn AMSMI-RD W C McCorkle 3080 Defense Pentagon Washington DC 20301-7100</td>
</tr>
<tr>
<td>CECOM</td>
<td>Attn PM GPS COL S Young FT Monmouth NJ 07703</td>
</tr>
<tr>
<td>Dir for MANPRINT</td>
<td>Ofc of the Deputy Chief of Staff for Prsnnl Attn J Hiller The Pentagon Rm 2C733</td>
</tr>
<tr>
<td>Night Vsn &amp; Elect Sensors Dir</td>
<td>Washington DC 20301-0300</td>
</tr>
<tr>
<td>US Army ARDEC</td>
<td>Attn AMSTA-AR-TD M Fisette Bldg 1 Picatinny Arsenal NJ 07806-5000</td>
</tr>
<tr>
<td>US Army Info Sys Engrg Cmd</td>
<td>Attn ASQB-OTD F Jenia FT Huachuca AZ 85613-5300</td>
</tr>
<tr>
<td>US Army Natick RDEC</td>
<td>Acting Techl Dir Attn SSCNC-T P Brandler Natick MA 01760-5002</td>
</tr>
<tr>
<td>US Army Simulation, Train, &amp; Instrmtn Cmd</td>
<td>Attn J Stahl 12350 Research Parkway Orlando FL 32826-3726</td>
</tr>
<tr>
<td>US Army Tank-Automtv Cmd Rsrch, Dev, &amp; Engrg Ctr</td>
<td>Attn AMSTA-TR J Chapin Warren MI 48397-5000</td>
</tr>
<tr>
<td>US Military Academy</td>
<td>Mathematical Sci Ctr of Excellence Attn MDN-A LTC M D Phillips</td>
</tr>
<tr>
<td>Night Vsn &amp; Elect Sensors Dir</td>
<td>Dept of Mathematical Sci Thayer Hall West Point NY 10996-1786</td>
</tr>
<tr>
<td>Nav Surface Warfare Ctr</td>
<td>Attn Code R07 J Pennella 17320 Dahlgren Rd Bldg 1470 Rm 1101 Dahlgren VA 22448</td>
</tr>
<tr>
<td>DARPA</td>
<td>5100</td>
</tr>
<tr>
<td>Hicks &amp; Associates Inc</td>
<td>Attn G Singley III 1710 Goodrich Dr Ste 1300 McLean VA 22102</td>
</tr>
<tr>
<td>Palisades Inst for Rsrch Svc Inc</td>
<td>Attn E Carr 1745 Jefferson Davis Hwy Ste 500 Arlington VA 22202-3402</td>
</tr>
<tr>
<td>US Army Rsrch Ofc</td>
<td>Attn AMSRL-RO-D JCI Chang</td>
</tr>
<tr>
<td>US Army Train &amp; Doctrine Cmnd</td>
<td>Attn AMSRL-RO-EN W Bach</td>
</tr>
<tr>
<td>Research Triangle Park NC 27709</td>
<td>PO Box 12211</td>
</tr>
</tbody>
</table>

15
Distribution (cont’d)

US Army Soldier & Biol Chem Cmmnd
Dir of Rsrch & Techly Dirctry
Attn SMCCR-RS I G Resnick
Aberdeen Proving Ground MD 21010-5423

TECOM
Attn AMSTE-CL
Aberdeen Proving Ground MD 21005-5057

US Army Rsrch Lab
Attn AMSRL-DD J M Miller
Attn AMSRL-CI-AI-A Mail & Records Mgmt

US Army Rsrch Lab (cont’d)
Attn AMSRL-CI-AP Techl Pub (3 copies)
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-IS-EP P Gillespie
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SF-SA J Eicke
Attn AMSRL-SE-SE M Vrabel (5 copies)
Adelphi MD 20783-1197
The relationship between performance and filter count is examined for a synthetic discriminant function-based target-detection algorithm.