Automatic Target Recognition Performance
Subject to System Resource Constraints

Michael D. DeVore

Advisor: Joseph A. O’Sullivan

Electronic Systems and Signals Research Laboratory

Washington University in St. Louis
School of Engineering and Applied Science

Supported by: ARO Center for Imaging Science DAAH 04-95-1-04-94
ONR MURI N00014-98-1-06-06
Boeing-McDonnell Foundation
Motivating Question

How can we compare different approaches to recognition?

Criteria like probability of error are sensitive to model approximations. To directly compare recognition approaches we could compare:

1. A fixed parameterization
2. The best performance across all parameterizations
3. The best performance as a function of parameterization constraints

Accuracy as a function of consumed resources.

Consequent Question

How can we achieve the best tradeoff between accuracy and resource consumption?
The ATR Problem

\[ a = T_{72} \]

\[ r \]

\[ \theta = 45^\circ \]

Model Database

Publicly available SAR data from the MSTAR program
Model-Based Recognition

Training Data

Scene and Sensor Physics

Function Estimation

Processing

$L(r|a, \theta)$

Inference

Raw HRR Data

SAR Image

Log-likelihood function

$\hat{a}=T72$
Model-Based Recognition

Our ATR work has focused on:

• Appropriate models for image data
  - Various distribution families
  - Implication of alternate models on ATR performance
  - Empirical fit of distributions and range of model validity

• Approximation error and estimation error
  - Function estimation from finite data
  - Information content in observations
  - Target models from training observations

• Performance subject to resource constraints
  - Best performance under database complexity constraints
  - Successively-refinable decisions
  - Computational system requirements
  - System performance under time constraints
Stochastic Models for SAR

We have considered variants of several models for SAR imagery:
Conditionally Gaussian, log-normal magnitudes, quarter-power normal magnitudes, Rician magnitudes

Example: Model pixel $i$ as independent, zero mean, complex conditionally Gaussian

$$
p_{R|\Theta,A,C^2}(r|\theta,a,c^2) = \prod_i \frac{1}{\pi c^2 \sigma_i^2(\theta,a)} e^{-\frac{|r_i|^2}{c^2 \sigma_i^2(\theta,a)}}
$$

Where: $\sigma_i^2(\theta,a) = \text{variance function over pose and class}$
$c^2 = \text{constant over all pixels to account for power fluctuation}$

To classify an image of an unknown target, we seek the maximum over $a$, $\theta$, and $c^2$
Piecewise-Constant Approximation

Given $N$ registered training images $q_j$ of a target with pose $\phi_j$, estimate variances over $N_w$ windows of width $d$

$$\hat{\sigma}^2(\theta_k, a) = \frac{1}{n_k} \sum_{j: \phi_j \in w_k} |q_j|^2$$

where

$$w_k = \left[ \frac{2\pi k}{N_w} - \frac{d}{2}, \frac{2\pi k}{N_w} + \frac{d}{2} \right]$$
Accuracy and Model Sophistication

Recognition accuracy depends on model correctness and:

- Segmentation properties
- Approximation complexity

Coarse model of a T62 tank, 1 template with 16K floats

Fine model of a T72 tank (1/7 relative scale), 72 templates totaling 1.1M floats
Accuracy and Model Sophistication

Approximate the performance-complexity relationship by testing under 240 parameter combinations. Convex hull denotes the achievable region.
Resource Consumption

Previous examples demonstrate accuracy and storage consumption. Other resource requirements may be of interest:

Network - how many bits transmitted?
Processor - how many operations?
Memory - how much local storage?

Time - how long will it take?
Accuracy and Hardware Throughput

Processing throughput also depends on model complexity.

- ATR hinges on likelihood function evaluation
- Each implementation decision sets a maximum number of function evaluations per unit time
- Maximum number of function evaluations determines what complexity of model can be used
- Complexity of model determines ATR accuracy
- For a given hardware configuration, accuracy and throughput are parametrically related.
Accuracy and Hardware Throughput

Example: Maximize $\ln p(r|\theta,a)$ over $360^\circ$ azimuth and 10 targets

$$\arg\max_{\theta,a} \sum_i \ln \sigma_i^2(\theta,a) + \frac{|r_i|^2}{\sigma_i^2(\theta,a)}$$

Computational Model: Chip processing rate $R=1/T_{CHIP}$, where

$$T_{CHIP} = 3 \frac{LMN}{P} T_1 + \frac{LMN}{P} T_2 + T_3$$

- $T_{CHIP}$ sec/SAR Image
- $T_1$ sec/clock cycle
- $T_2$ sec/template memory read
- $T_3$ sec/SAR Image load
- $L$ templates/target
- $M$ targets
- $N$ pixels/template
- $P$ processors

Assumptions:
- Multi-issue CPU with 2 instructions/clock cycle
- 6 instructions per pixel
Accuracy and Hardware Throughput

$T_2 = T_1$ with prefetch

1 GHz clock

16 KB/SAR Image (4B floats)

$M = 10$ targets

Varying target model complexity

($L$ templates/target and $N$ pixels/template)

1 Gb/s Interconnection Network

10 Gb/s Interconnection Network
Methods of Improvement

We now have a tool to assess design improvements.

Specifically interested in improved accuracy-resource tradeoffs.

Improved performance through:

• Better models
  - models of target reflectivity, not sensor output
  - segment target from clutter

• Better search strategies
  - low-resolution searches to restrict large spaces

• Successively-refinable models
  - dynamically adjustable operating point
Better Models: Target Models

Register training images is orientation as well as position.

Variance estimate for an unregistered image $r$ with pose $\phi$ formed by appropriately transforming the estimate from the closest $w_k$.

Registered Variance Images

Transformed Estimates

$T_{0^\circ}$

$T_{90^\circ}$

$T_{180^\circ}$

$T_{270^\circ}$
Better Models: Segmentation

• Presume a null-hypothesis $H_0: r_i \sim \text{CN}(0, \xi^2)$ which models the scenario where no target is present in the image

• Quantify pixel information relative to the null-hypothesis

\[
D(p|q) = \int p(x) \log \left( \frac{p(x)}{q(x)} \right) dx
\]

For each target retain pixels $i$ for which:

\[
\frac{1}{N_w} \sum_k D\left( \hat{\sigma}_i^2 (\theta_k, a) \parallel \xi^2 \right) \geq \tau
\]

Segmentation of target models, not of images
Better Models: Segmentation

Yields an ordering of pixels by their empirical information relative to null-hypothesis.

For null-hypothesis $\xi^2=0.0028$ - approximate background variance - pixels on illuminated side of target are deemed most informative.
Modified Likelihood

Accommodate varying pixel selection by computing likelihood relative to the null-hypothesis

$$\hat{a}, \hat{\theta}, \hat{c}^2 = \arg\max_{a, \theta, c^2} \prod_{i \in S_a} \left\{ \frac{1}{\pi c^2 \sigma_i^2(\theta, a)} \exp\left( -\frac{|r_i|^2}{c^2 \sigma_i^2(\theta, a)} \right) \right\} \left/ \left( \frac{1}{\pi \xi^2} \exp\left( -\frac{|r|_i^2}{\xi^2} \right) \right) \right\}$$

Schmid & O’Sullivan “Thresholding Method for Reduction of Dimensionality” 18
Better Search Strategies

Exploit the steep side of the performance-complexity curve

- Don’t search a fixed approximation in arbitrary order
- Use a sequence of searches with increasing model sophistication
- Order the search at each level by the results of the previous level
Successively-Refinable Models

- Represent model parameter functions, i.e. $\sigma_i^2(\theta, a)$, as a collection of hierarchical approximations.
- Divide azimuth into $N_d$ non-overlapping intervals of width $d$.

$$\tilde{\sigma}_{d,i}^2(\theta_k, a) = \frac{1}{d} \int_{\frac{2\pi k}{N_d} - \frac{d}{2}}^{\frac{2\pi k}{N_d} + \frac{d}{2}} \sigma_i^2(\theta, a) d\theta$$

Approximations $d$ and $d/2$ are hierarchically related:

$$\tilde{\sigma}_{d,i}^2(\theta_k, a) = \frac{1}{2} \left[ \tilde{\sigma}_{\frac{d}{2},i}^2(\theta_{2k}, a) + \tilde{\sigma}_{\frac{d}{2},i}^2(\theta_{2k+1}, a) \right]$$
Successively-Refinable Models

Consider decreasing interval widths $d_1 = 2\pi$, $d_2 = \pi$, ..., $d_m = 2\pi/2^{m-1}$

Approximate variance images for target D7 from $d_1$ through $d_5$.

Search over $\theta_k$ in level $i$ ordered by the most likely pose at level $i-1$
Four-Class Experiment

Classification error rate as a function of the number of bits transmitted between the database and processor

- Classification depends on extent of search
- Eventually, search covers all possibilities
- Breadth-first search quickly finds good combinations of \((a, \theta)\)
- Small overhead present with ordered searches
Confuser Rejection

Relative entropy to reject target classes not in the database

- One-sample estimate of variance from observation: $|r_i|^2$
- Relative entropy quantifies difference from training estimate
- Reject observation if the empirical relative entropy is too large

$$\frac{1}{|S_\hat{a}|} \sum_{i \in S_\hat{a}} D\left(r_i^2 \| \hat{\sigma}_i^2 (\hat{\theta}, \hat{a}) \right) > \gamma$$

Four trained targets: BMP-2, BRDM-2, BTR-70, T-72
Four confuser targets: 2S1, BTR-60, D7, T-62
Conclusions

• Working toward an end-to-end model for the ATR problem
• Model all components from sensors to computational hardware
• Methodology for quantifying the effects of implementation decisions
• Support a search for the best performance under system constraint
• For detailed experimental results, see http://cis.wustl.edu/mstar

Other current efforts:

• Quantify model correctness independent of recognition accuracy
• Prediction of recognition accuracy from training data
• More sophisticated hardware models
Appendix Material
Log-Normal Model for SAR

Minimize distance between $r_{dB} = 20 \log |r|$ and dB templates

$$d^2(r_{dB}, \mu_{LN}) = \|r_{dB} - \mu_{LN}\|^2$$

Make decisions according to:

$$\begin{bmatrix} \hat{a} \\ \hat{\theta} \end{bmatrix} = \operatorname{argmax} d^2(r_{db}, \mu_{LN}(\theta,a))$$

Alternatively, use a form of normalization:

$$d^2\left(r_{dB} - r_{dB}, \mu_{LN}(\theta,a) - \mu_{LN}(\theta,a)\right)$$
Quarter-Power Normal Model for SAR

Minimize distance between \( r_{QP} = |r|^{1/2} \) and quarter power templates

\[
d^2(\mathbf{r}_{QP}, \mu_{QP}) = \| \mathbf{r}_{QP} - \mu_{QP} \|^2
\]

Make decisions according to:

\[
\begin{bmatrix}
\hat{a} \\
\hat{\theta}
\end{bmatrix} = \underset{[a, \theta]^T}{\text{argmax}} d^2(\mathbf{r}_{QP}, \mu_{QP}(\theta, a))
\]

Or, normalized by vector magnitude:

\[
d^2\left( \frac{\mathbf{r}_{QP}}{\| \mathbf{r}_{QP} \|}, \frac{\mu_{QP}(\theta_k, a)}{\| \mu_{QP}(\theta_k, a) \|} \right)
\]