Relationships Between Computational System Performance and Recognition System Performance

Michael D. DeVore
Joseph A. O'Sullivan
Roger D. Chamberlain
Mark A. Franklin

Michael D. DeVore, Joseph A. O'Sullivan, Roger D. Chamberlain, and Mark A. Franklin, “Relationships Between Computational System Performance and Recognition System Performance." In *Proc. of SPIE 15th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls (Automatic Target Recognition XI)*, April 2001.

Department of Electrical Engineering
Washington University
St. Louis, Missouri

Relationships Between Computational System Performance and Recognition System Performance

Michael D. DeVore[a], Joseph A. O'Sullivan[b], Roger D. Chamberlain[c], Mark A. Franklin[d]

[a,b]Electronic Systems and Signals Research Laboratory
[c,d]Computer and Communications Research Center
Department of Electrical Engineering, Washington University, St. Louis, MO 63130

ABSTRACT

The implementation of computational systems to perform intensive operations often involves balancing the performance specification, system throughput, and available system resources. For problems of automatic target recognition (ATR), these three quantities of interest are the probability of classification error, the rate at which regions of interest are processed, and the computational power of the underlying hardware. An understanding of the inter-relationships between these factors can be an aid in making informed choices while exploring competing design possibilities. To model these relationships we have combined characterizations of ATR performance, which yield probability of classification error as a function of target model complexity, with analytical models of computational performance, which yield throughput as a function of target model complexity. Together, these constitute a parametric curve that is parameterized by target model complexity for any given recognition problem and hardware implementation. We demonstrate this approach on the problem of ATR from synthetic aperture radar imagery using a subset of the publicly released MSTAR dataset. We use this approach to characterize the achievable classification rate as a function of required throughput for various hardware configurations.

Keywords: synthetic aperture radar, automatic target recognition, MSTAR, performance-complexity

1. INTRODUCTION

The design of production level systems that implement computationally intensive algorithms often involves balancing result quality specifications, system throughput requirements, and available system resources. For automatic target recognition (ATR), these three factors correspond to the probability of classification error delivered by the system, the rate at which regions of interest are processed, and the computational power and I/O capabilities of the underlying hardware. An understanding of how these three factors are interrelated for a given problem will help designers explore implementation parameters and directly compare alternate implementation strategies.

The classification error rate delivered by an ATR system is directly related to the sophistication of the target models employed. More highly refined target models may be expected to yield lower error rates but generally have larger processing and data storage requirements. These larger requirements translate into slower operation and/or a need for greater computational capability. Knowledge of how the classification error rate depends on target model sophistication can be combined with analytic models for computation hardware and recognition algorithms to determine the relationship between error rate and system throughput.

We address the relationship between recognition accuracy and system throughput for ATR systems based on synthetic aperture radar (SAR) imagery. O'Sullivan, *et al.*[8] introduce a conditionally Gaussian model for the complex-valued pixels in SAR images which we exploit in the current study. O'Sullivan and DeVore[2,7] characterize several approaches to ATR from SAR imagery in terms of the best achievable performance as a function of model database complexity. The resulting performance-complexity curves provide a means of directly comparing the capabilities of alternate ATR algorithms without being overly sensitive to particular choices of implementation parameters. This paper builds upon that work by incorporating hierarchical target models as in DeVore, *et al.*[3] rather than multiple, fixed-complexity representations. Moreover, models for several computation architectures are developed and used to predict processing rates for target images as a function of target model complexity. In related work, Franklin, *et al.*[5]

E-mail: [a]mdd2@cis.wustl.edu, [b]jao@ee.wustl.edu, [c]roger@ccrc.wustl.edu, [d]jbf@ccrc.wustl.edu

have developed computationally based computer architecture performance models for ultrasonic image generation. These models are in the spirit of Hennessy and Patterson.[6]

Section 2 presents a conditionally Gaussian model for SAR imagery and addresses both parameter estimation from training data and recognition of unknown objects in images. Section 3 describes a hierarchical encoding of functions that parameterize the conditionally Gaussian model. A computational model which relates ATR system characterization to elapsed processing time for a hypothetical computer architecture is derived in Section 4. Section 5 contains the error rate and system throughput results of a four-class experiment conducted with publicly available data released through the Moving and Stationary Target Acquisition and Recognition (MSTAR) program conducted at Wright Laboratory under DARPA funding. Conclusions follow in Section 6.

2. SAR MODEL AND RECOGNITION ALGORITHMS

The conditionally Gaussian model[8] holds that the value r_i of a pixel in a SAR image of a target is a complex-Gaussian distributed random variable conditioned on the pixel location i, target class a, and target pose Θ. Further, r_i is well modeled as zero-mean and, given a and Θ, is independent of the values at other pixels. The probability density for an entire SAR image $\mathbf{r}$ is the product of the densities for each pixel and is given as

$$p(\mathbf{r}|a,\Theta,c^2) = \prod_{i=1}^{N} \frac{1}{\pi c^2 \sigma_i^2(a,\Theta)} e^{-|r_i|^2/c^2\sigma_i^2(a,\Theta)}, \tag{1}$$

where N is the number of pixels in the image, $\sigma_i^2(a,\Theta)$ is the variance of the value at pixel i when a target of class a is imaged at pose Θ under some nominal radar power, and c^2 is an unknown, deterministic scale factor accounting for radar power fluctuation. In the general case, the target pose, Θ, includes the target position and orientation relative to the radar platform as well as the states of all articulated components such as hatches, gun barrels, and turrets. For our purposes, we will consider only ground-based targets on level terrain imaged with a known depression angle and with unarticulated components. In this simplified case, Θ includes only the target position in range and cross-range directions and azimuthal rotation relative to the radar platform.

The probability density in (1) gives a model for a received SAR image in terms of a variance function which in general is unknown. We can predict this variance function from a model of the target itself based upon prior knowledge of its construction, for example a CAD model, or based upon prior observations in the form of training data. Given a set of power-normalized training data that consists of SAR images of the target from a wide variety of known poses, we can construct a target model that consists of estimates of the variance in the reflected radar signal, as a function of observation angle, for each patch on the surface of the target. This target model can be used to complete the SAR image model in (1) for any given pose, Θ, by applying the rotation and shift operations implied by Θ to the target model and projecting the result onto the radar slant plane corresponding to the given depression angle.[3]

Since we are considering a fixed depression angle and only azimuthal target rotation, we can restrict our attention to models of targets in the slant plane. Suppose we are given a set of M training images, $\{\mathbf{T}'_1, \mathbf{T}'_2, \ldots, \mathbf{T}'_M\}$, of target a with azimuth angles $\{\phi_1, \phi_2, \ldots, \phi_M\}$ and locations $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M\}$ relative to the radar platform and all collected with the same depression angle. We first register the images in both position and orientation, shifting each image $\mathbf{T}'_i$ by an amount $-\mathbf{s}_i$ and rotating the result by $-\phi_i$, to obtain a set of registered images $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_M\}$. The variance at each resolution cell, $\zeta_i^2(\phi)$, is approximated as piecewise constant in viewing angle over N_w intervals and its value over the kth interval, centered at $\phi_k = 2\pi k/N_w$, is estimated from a subset of registered training images collected with azimuth angles close to ϕ_k. That is,

$$\zeta_i^2(\phi_k) = \frac{1}{|\mathcal{I}_k|} \sum_{\mathbf{r}\in\mathcal{I}_k} |r_i|^2, \tag{2}$$

where $\mathcal{I}_k = \{\mathbf{T}_j \in \mathcal{T} : \phi_j \in [\phi_k - d/2, \phi_k + d/2]\}$ for some training interval width d. If $d > 2\pi/N_w$ then some training images may contribute to the estimate of $\zeta_i^2(\phi_k)$ for more than one interval k.

Given the estimated target model from (2), we can estimate the variance function in the SAR image model (1) for some pose Θ by selecting the closest corresponding interval center ϕ_k and applying the rotation and translation implied by Θ to the variance image $\boldsymbol{\zeta}^2(\phi_k)$. That is, $\boldsymbol{\sigma}^2(a,\Theta) = U(\Theta)\left[\boldsymbol{\zeta}^2(\phi_k)\right]$ where $U(\Theta)$ represents the transformation

operation implied by pose Θ. From this complete SAR image model we can perform joint target recognition and pose estimation given a SAR image $\mathbf{r}$ via the generalized likelihood ratio test (GLRT). We select the target and pose that maximizes the density (1) or, equivalently, its logarithm as in

$$\begin{bmatrix} \hat{a} \\ \hat{\Theta} \\ \hat{c}^2 \end{bmatrix} = \underset{[a,\Theta,c^2]^T}{\operatorname{argmax}} \ln p(\mathbf{r}|a,\Theta,c^2). \tag{3}$$

This can be expanded, removing terms independent of a, Θ, and c^2, as

$$\begin{bmatrix} \hat{a} \\ \hat{\Theta} \end{bmatrix} = \underset{[a,\Theta]^T}{\operatorname{argmax}} \left\{ -\sum_{i=1}^{N} \ln\left[\hat{c}^2(a,\Theta)\sigma_i^2(a,\Theta)\right] - \sum_{i=1}^{N} \frac{|r_i|^2}{\hat{c}^2(a,\Theta)\sigma_i^2(a,\Theta)} \right\}, \tag{4}$$

where

$$\hat{c}^2(a,\Theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{|r_i|^2}{\sigma_i^2(a,\Theta)}. \tag{5}$$

The target model estimated in (2) may be segmented to delineate target from clutter regions. Let the segmentation of target model a be denoted by the binary image $\mathbf{I}(a)$ which has the value 1 at every pixel determined to be on the target and 0 at all other pixels. Then the segmentation mask of the SAR image model $\boldsymbol{\sigma}^2(a,\Theta)$ is given by $U(\Theta)[\mathbf{I}(a)]$. Schmid and O'Sullivan[9] present an asymptotic analysis of the probability of error for a related two-class problem using simulated high range resolution radar data. They show that by retaining pixels in the estimated target models based upon a measure of information relative to a zero-mean Gaussian distribution with variance ξ^2 and computing a likelihood relative to that same distribution, they can achieve a better upper bound on the probability of error than without segmentation. Under this approach (3) becomes

$$\begin{bmatrix} \hat{a} \\ \hat{\Theta} \\ \hat{c}^2 \end{bmatrix} = \underset{[a,\Theta,c^2]^T}{\operatorname{argmax}} \sum_{i=1}^{N} \ln\left[\frac{p(r_i|c^2\sigma_i^2(a,\Theta))}{p(r_i|\xi^2)}\right] \left(U(\Theta)[\mathbf{I}(a)]\right)_i. \tag{6}$$

Expanding the probability densities and applying the logarithm yields the final expression,

$$\begin{aligned} \begin{bmatrix} \hat{a} \\ \hat{\Theta} \end{bmatrix} &= \underset{[a,\Theta]^T}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N(a,\Theta)} \ln\left[\frac{\xi^2}{\hat{c}^2(a,\Theta)\sigma_{(i)}^2(a,\Theta)}\right] + \sum_{i=1}^{N(a,\Theta)} |r_{(i)}|^2 \left(\frac{1}{\xi^2} - \frac{1}{\hat{c}^2(a,\Theta)\sigma_{(i)}^2(a,\Theta)}\right) \right\} \\ &= \underset{[a,\Theta]^T}{\operatorname{argmax}} \left\{ -N(a,\Theta)\ln\hat{c}^2(a,\Theta) - \left[\sum_{i=1}^{N(a,\Theta)} \ln\sigma_{(i)}^2(a,\Theta) - N(a,\Theta)\ln\xi^2 + N(a,\Theta)\right] + \frac{1}{\xi^2}\sum_{i=1}^{N(a,\Theta)} |r_{(i)}|^2 \right\} \end{aligned} \tag{7}$$

where $N(a,\Theta)$ is the number of nonzero elements in the segmentation mask, (i) is the ith nonzero pixel in the mask, and

$$\hat{c}^2(a,\Theta) = \frac{1}{N(a,\Theta)} \sum_{i=1}^{N(a,\Theta)} \frac{|r_{(i)}|^2}{\sigma_{(i)}^2(a,\Theta)}. \tag{8}$$

The first 3 terms in (7) follow by expanding the logarithm in the previous equation, and the last two follow by distributing the pixel magnitude and applying the definition of $\hat{c}^2$ in (8). The terms have been grouped according to data requirements. Note that only the first and last terms involve the observed SAR data and that the middle bracketed expression can be precomputed and stored as a single floating-point value. DeVore, *et al.*[4] demonstrate the effectiveness of this approach on actual SAR data with $\xi^2 = 0.0028$ and indicate that the results are relatively insensitive to the choice of ξ^2.

3. SUCCESSIVELY-REFINABLE PARAMETER ENCODING

The ATR algorithm specified in (7) and (8) is completely determined by the variance functions $\sigma_{(i)}^2(a,\Theta)$ and the locations of nonzero pixels in the segmentation mask. Direct implementation of this algorithm is difficult, however,

because $\Theta = [\theta, \mathbf{s}]$ is continuous in azimuth angle and in both range and cross-range position. If we approximate $\sigma^2_{(i)}(a, \Theta)$ as piecewise constant in these parameters, we can consider maximization over only a finite set. We begin by restricting consideration of target position to integer numbers of pixel widths. Further, we divide the set of azimuth angles into N_d nonoverlapping intervals of width d. The value of the variance function over the kth interval is then approximated as

$$\begin{aligned} \tilde{\sigma}^2_{d,(i)}(a, \theta_k) &= \frac{1}{d} \int_{2\pi k/N_d - d/2}^{2\pi k/N_d + d/2} \sigma^2_{(i)}(a, \theta) d\theta \\ &\approx \frac{\Delta_\theta}{d} \sum_{j=-d/2\Delta_\theta}^{d/2\Delta_\theta} \sigma^2_{(i)}(a, 2\pi k/N_d + j\Delta_\theta), \end{aligned} \tag{9}$$

where Δ_θ is a small angle chosen to evenly divide $d/2$. Because they are a finite set, these variances can be precomputed and stored in a database accessible to the ATR system. The accuracy of the ATR system will depend upon the interval width d. We could expect that finer grained approximations of the variance function, corresponding to small d, will yield improved performance but at the expense of more computation because of the larger search space, corresponding to large N_d.

We desire a system implementation that will proceed with computation until the time quota for classification has been consumed and then produce the best combination of a and Θ found up to that time. This is similar in spirit to progressively encoded JPEG images which yield successively more accurate representations as the number of transmitted bits grows. We choose a multi-resolution encoding of $\sigma^2_{(i)}(a, \theta)$ corresponding to a decreasing sequence of interval widths $d_1 = 2\pi, d_2 = \pi, \ldots, d_m = 2\pi/2^{m-1}$. From (9) we note that there is a direct relationship between the variance function approximated with intervals d_m and that approximated with d_{m+1}. Specifically,

$$\tilde{\sigma}^2_{d_m,(i)}(a, \theta_k) = \frac{1}{n_1 + n_2} \left[n_1 \tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k}) + n_2 \tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k+1}) \right], \tag{10}$$

where n_1 and n_2 are the number of terms in the approximation of $\tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k})$ and $\tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k+1})$, respectively. This relationship is likely to be useful in systems with low communication bandwidth between the database and processing unit. In such systems, given that $\tilde{\sigma}^2_{d_m,(i)}(a, \theta_k)$, $n_1 + n_2$, $\tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k})$, and n_1 have been communicated to the ATR processor, the processor can determine $\tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k+1})$ and n_2 so these do not need to be explicitly transmitted. In systems with high communication bandwidth, it may be preferable to simply transmit $\tilde{\sigma}^2_{d_{m+1},(i)}(a, \theta_{2k})$ even though enough information to calculate it has already been received at the processor.

We can exploit this hierarchical representation by performing the maximization indicated in (7) under the approximation for $\sigma^2_{(i)}$ given by d_1. If the allotted time has not yet expired, then the maximization can be performed again with the more refined approximation given by d_2, etc. By exploring the parameter space under approximation d_{m+1} according to an ordering of likelihood values established under approximation d_m, the most likely regions of the space are searched first. In this way the best results from the current approximation, d_{m+1} can be reported at any time, regardless of whether or not the entire space has been searched. In the limit, as the approximation intervals grow small and corresponding computation time grows large, the search results approach the true maximum of (7). DeVore, *et al.*[3] show in a four-class MSTAR experiment that the probability of classification error decreases much more quickly as a function of bits transmitted from the database under this hierarchical approach than under an exhaustive search through azimuth with a fixed approximation, d_m.

4. HARDWARE MODELS

In this section, we derive computational models for the algorithm given in (7) and (8) assuming that the hierarchical relationship of (10) is exploited whenever possible. We consider a network of processors, each with its own local memory sufficient to store all pixels of a SAR magnitude image and of two variance images $\boldsymbol{\sigma}^2(\theta, a)$. Further, SAR images to be classified are supplied from a single SAR imaging platform and the processors share access to a single database of model parameters $\tilde{\sigma}^2_{d,(i)}(a, \theta_k)$ from (9). Table 1 contains definitions of the parameters that characterize the ATR system. Our interest lies in the average throughput of the classification system and not on the latency in processing any one SAR image. To this end, we assume that incoming images arrive at a rate sufficient to allow all processors to stay fully utilized and suppose that a scheduling mechanism exists to coordinate processor activity.

P	the number of processors in the system
T_{cyc}	cycle time for the processor (sec)
T_{mem}	mean memory access time for a 64 bit floating-point value (sec)
S_{C}	size of a SAR image chip (bits)
BW	bandwidth of the interconnection network (bits/sec)
N_m	the average number of pixels in the segmented templates $\tilde{\boldsymbol{\sigma}}^2_{d_m}(a, \theta_k)$
N_{S}	the number of target positions considered
N_{T}	the number of targets considered
b_m	the average number of bits in the representation of the template $\tilde{\boldsymbol{\sigma}}^2_{d_m}(a, \theta_k)$
CPI	mean cycles per instruction executed by each processor, ignoring memory effects

Table 1. Parameters characterizing the ATR systems under consideration.

Because of the sequential nature of the operations, the time to process an image chip through approximation d_m includes the time to obtain the chip from the radar platform, the time to maximize over approximation d_1, and the time to maximize over approximations d_2 through d_m, in turn. If the processor interconnection network does not support generalized data broadcasting, the SAR image to be classified must be independently communicated to each processor in an operation that will take $\frac{S_{\text{C}}}{\text{BW}}\lceil \log_2(P+1) \rceil$ seconds.[1] This assumes a switched interconnection network and $P+1$ total nodes involved in the broadcast (P compute nodes and the source of the SAR images). For the approximation d_1, no processor will be able to make use of the hierarchical relationship in (10), so the time to maximize over this approximation, τ_{d_1}, will differ in form from that of the other approximations. For the approximations d_l, $l = 3, \ldots, m$, the hierarchical relationship can be exploited only half the time on average because, under the memory constraints previously mentioned, only one of the two variance images in a processor's memory can be refined. The other variance image has two refinements that must be communicated to a different processor. Since there are 2^{l-1} variance images of each target at level l, 2^{l-2} can be successively refined through one transmission followed by computation and 2^{l-2} must have both refinements transmitted. Under these assumptions, the time to process a SAR image chip through approximation d_m has the form

$$T_{chip} = \frac{S_{\text{C}}}{\text{BW}}\lceil \log_2(P+1) \rceil + N_{\text{T}}\tau_{d_1} + 2N_{\text{T}}\tau_{d_2} + \sum_{l=3}^{m} N_{\text{T}}\left[2^{l-2}\tau_{d_l} + 2^{l-2}\tau'_{d_l}\right], \tag{11}$$

where τ_{d_l} is the average time required to evaluate the expression in the argmax function of (7) when (10) can be directly exploited, and τ'_{d_l} is the evaluation time otherwise.

The processing times τ in (11) include the time to move the model parameters from the database to local memory, the time to move those parameters and the SAR image from local memory into the processor, and the time to perform the computations. We model these as interlaced but nonoverlapping stages of computation, so the total computation time is equal to their sum. If portions of each can overlap in a given implementation, then these quantities can be taken to represent effective times, equal to the total time multiplied by some nonoverlap percentage. For τ_{d_l}, which involve the communication of only a single variance image, the time to communicate the model parameters from the database takes b_l/BW seconds. For τ'_{d_l}, two communications are necessary totaling $2b_l/\text{BW}$ seconds on average.

The time to move all parameter and SAR image values from memory into the processor depends upon the ordering of computations. If the processors each have at least $2N_{\text{S}} + 2$ floating point registers, then the computations for all considered target locations can be interlaced using one register each to hold a SAR pixel value and a variance value and one register each to accumulate the quotients in (8) and the magnitudes in the last term of (7) for each target location. If the processors each have $4N_{\text{S}} + 3$ floating point registers, then computations for all considered target locations can be interlaced for two variances images. Assuming this is the case, then the computations for approximation d_1 require $2N_1$ floating-point read operations for the variance and corresponding SAR pixel, one read operation for the bracketed expression in (7), and one read operation to fetch ξ^2. The total is $2N_1 + 2$ reads and these can be distributed over P processors for a total memory read time of $\frac{2N_1+2}{P}T_{\text{mem}}$. Note that T_{mem} is a mean access time, incorporating the impact of a cache if present. For the approximations d_l, with $l \geq 2$, we can interlace the likelihood computations for two variance images and, if we are computing $\tilde{\boldsymbol{\sigma}}^2_{d_l}(a, \theta_{2k+1})$ from $\tilde{\boldsymbol{\sigma}}^2_{d_{l-1}}(a, \theta_k)$ and $\tilde{\boldsymbol{\sigma}}^2_{d_l}(a, \theta_{2k})$, we must read N_{l-1} floating-point values for each of the two variances and from the SAR image, along

Class	Vehicle	Serial No.	Training Set		Testing Set	
			Depression	Images	Depression	Images
BMP-2	#1	9563	17°	233	15°	195
	#2	9566		231		196
	#3	c21		233		196
BRDM-2	#1	E-71	17°	298	15°	263
BTR-70	#1	c71	17°	233	15°	196
T-72	#1	132	17°	232	15°	196
	#2	812		231		195
	#3	s7		228		191

Table 2. MSTAR images comprising the training and testing data. Given are the model and vehicle specification, depression angle from which the data were collected, and the number of images for each.

with two integers, n_1 and n_2 from (10), for every pixel, two floating-point values for the second bracketed expression in (7), and a single value of ξ^2. If two integers can be retrieved from memory in the time it takes to retrieve a single floating-point value, this will average $\frac{4N_{l-1}+3}{P}T_{\text{mem}}$ seconds. Finally, if we are not computing any of the variance values but simply retrieving them from memory, we need not retrieve the integers n_1 and n_2 for each pixel and the average time becomes $\frac{3N_l+3}{P}T_{\text{mem}}$ seconds.

Associating one instruction for each divide and accumulate operation in (8), one each for division and multiplication operations, and ten instructions for logarithm computation, then computing the first term of (7) will take $N_l + 12$ instructions on average. Assuming one instruction per addition, the last term of (7) can be computed in $N_l + 1$ instructions and there will be two additions required to combine these three quantities. The total is $2N_l + 15$ instructions on average to compute the expression in the argmax function for each of the N_{S} target positions. This work can be divided over P processors that average CPI clock cycles per instruction requiring $\frac{N_{\text{S}}(2N_l+15)}{P} \cdot \text{CPI} \cdot T_{\text{cyc}}$ seconds. When (10) is used to derive one of the variance images, four instructions are required for every pixel so a total of $4N_l$ instructions are required. These variances can be stored locally and do not need to be recomputed for each target location considered.

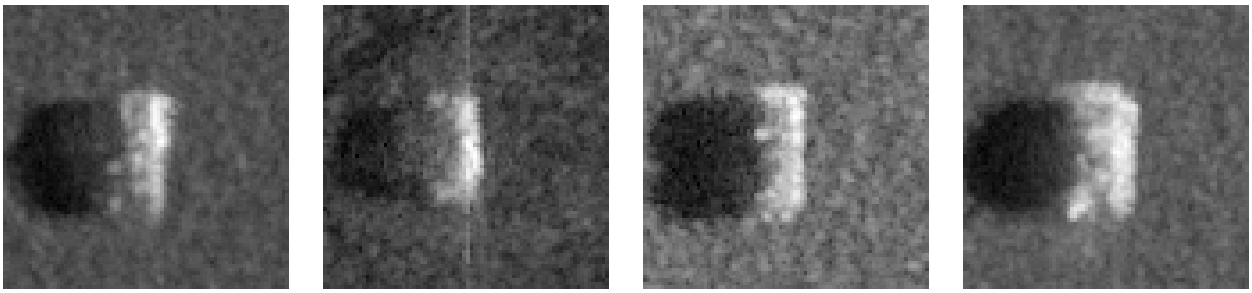
Letting the values τ_l in (11) equal the sum of the time to communicate parameters from the database, the time to load data from memory into the processor, and the time to perform the computation, the average time to classify an image chip becomes

$$\begin{aligned} T_{\text{chip}} \quad = \quad & \frac{S_{\text{C}}}{\text{BW}}\lceil \log_2(P+1) \rceil + N_{\text{T}} \left[\frac{b_1}{\text{BW}} + \frac{2N_1+2}{P} T_{\text{mem}} + \frac{N_{\text{S}}(2N_1+15)}{P} \cdot \text{CPI} \cdot T_{\text{cyc}} \right] + \\ & N_{\text{T}} \left[\frac{b_1}{\text{BW}} + \frac{4N_1+3}{P} T_{\text{mem}} + \frac{N_{\text{S}}(4N_2+30)+4N_2}{P} \cdot \text{CPI} \cdot T_{\text{cyc}} \right] + \\ & \sum_{l=3}^{m} \left\{ 2^{l-3} N_{\text{T}} \left[\frac{b_l}{\text{BW}} + \frac{4N_{l-1}+3}{P} T_{\text{mem}} + \frac{N_{\text{S}}(4N_l+30)+4N_l}{P} \cdot \text{CPI} \cdot T_{\text{cyc}} \right] + \right. \\ & \left. 2^{l-3} N_{\text{T}} \left[2\frac{b_l}{\text{BW}} + \frac{3N_l+3}{P} T_{\text{mem}} + \frac{N_{\text{S}}(4N_l+30)}{P} \cdot \text{CPI} \cdot T_{\text{cyc}} \right] \right\}. \end{aligned} \tag{12}$$

5. ACCURACY-THROUGHPUT RESULTS

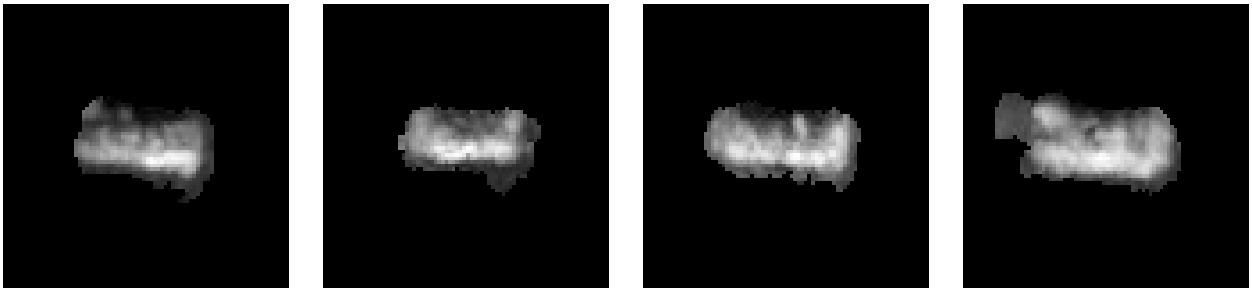
Experiments with four vehicle classes from the publicly available MSTAR dataset were conducted to assess the behavior of the ATR algorithm developed in Section 2 for a variety of hardware characteristics. Table 2 describes the collection of images used in the experiments. Images from each vehicle class were segregated into nonoverlapping training and testing sets according to the depression angle from which they were collected. Data for two of the classes, the BMP-2 and T-72, consist of images from several distinct vehicles.

Four target models were estimated from the training data according to Equation (2) with $N_w = 64$ and $d = 11.25°$. This combination yielded an average of from 7 to 9 samples for every estimated variance $\zeta_i^2(\phi_k)$ for models of the



BMP-2 BRDM-2 BTR-70 T-72

Figure 1. Estimated variance images from each of the four targets when illuminated from a SAR platform oriented 90° with respect to the main vehicle axis.



BMP-2 BRDM-2 BTR-70 T-72

Figure 2. Masked models of sensor output for each of the four targets oriented 90° relative to the SAR platform with variance functions approximated as piecewise constant over 32 intervals.

BRDM-2 and BTR-70 and roughly three times that amount for the BMP-2 and T-72 models. Images of the variance functions $\boldsymbol{\zeta}^2(a, \phi_k)$ for each of the four vehicle classes with $\phi_k = 90°$ are shown in Figure 1 on a logarithmic amplitude scale. Images of the masked sensor output models $\tilde{\boldsymbol{\sigma}}^2_{d_6}(a, \theta_k)$ for approximation $d_6 = \pi/16$ and $\phi_k = 90°$ are shown in Figure 2 on a logarithmic amplitude scale.

The percentage of erroneously classified test images as a function of required chip processing rate, $R_{\text{chip}} = 1/T_{\text{chip}}$ is shown for several combinations of hardware configurations in Figure 3. Each curve is plotted under the assumption of a fixed hardware configuration under varying throughput demands. For all experiments, a 1 GHz system clock is assumed and memory access time of 64 bits per clock cycle is supported via a prefetch capability (ie. T_{mem} is 1ns). An incoming SAR image chip is assumed to be an array of 128×128 floating-point values. Twenty-five discrete positions in the radar slant plane are considered for each image and the images are classified as one of four vehicle classes. Multi-issue processors capable of achieving 2 instructions per clock cycle are assumed. The values of N_m and b_m were determined from the estimated target models.

The left panel of Figure 3 shows results for 1, 2, and 4 processors assuming an interconnection network with 1 Gbps bandwidth. Point markers along the three curves show where each successive approximation d_m was begun. High chip rates correspond to coarse approximations of the variance functions from (9). With such approximations, the time to distribute incoming SAR images to all processors dominates the chip processing time and it becomes advantageous, in the sense that higher processing rates are attainable for fixed probabilities of error, to use fewer processors. With more refined approximations, the actual computation time becomes large enough to make the use of additional processors attractive. The right panel of the figure contains a detail view showing the region of the graph where this transition occurs. At fewer than 150 image chips per second, it is preferable to use a system of two processors and at fewer than around 60 image chips per second, it is preferable to use four.

The bottom panel shows the results assuming the interconnection network is replaced with a faster 10 Gbps

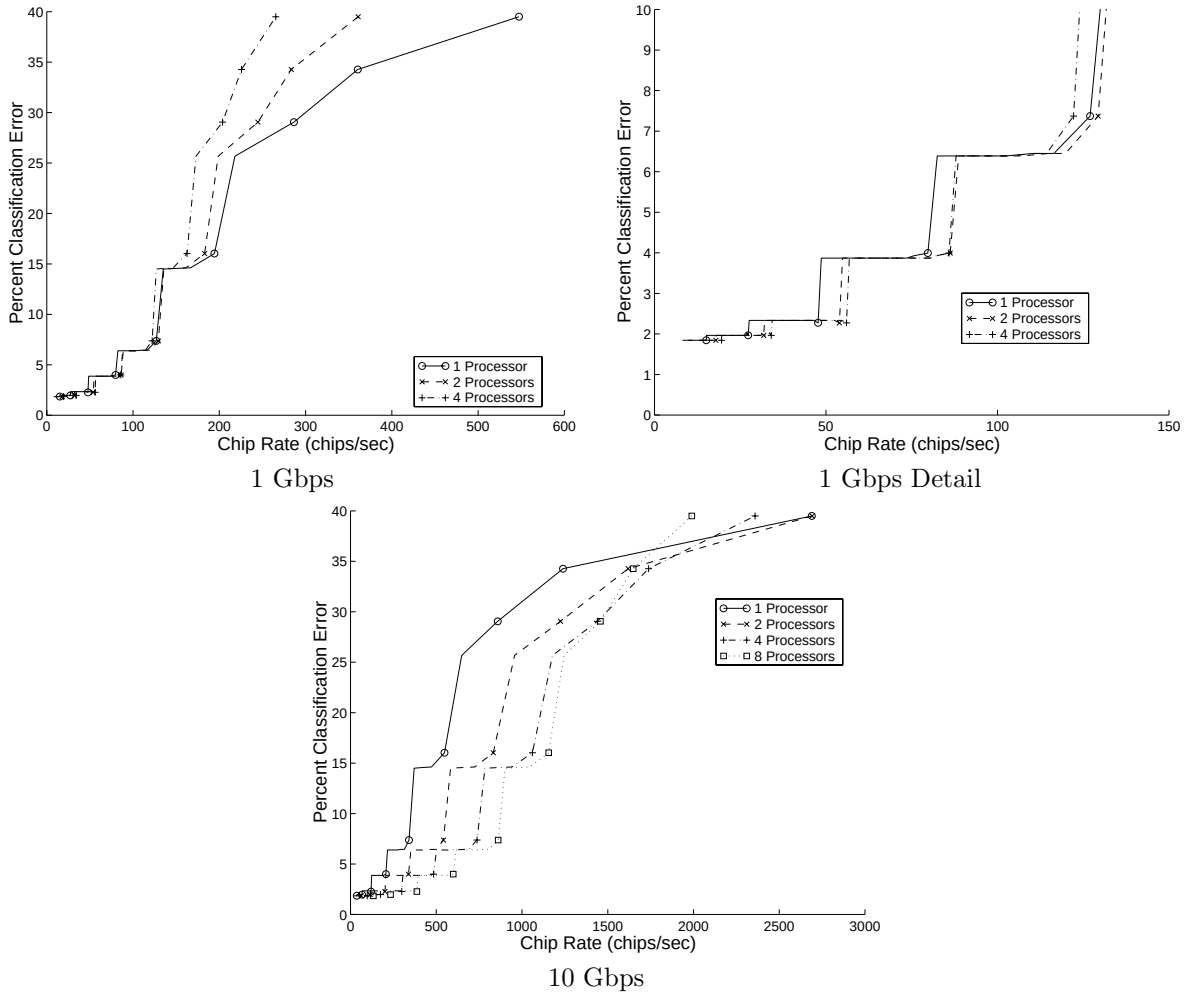


Figure 3. Percent classification errors as a function of chip throughput in a four-class experiment. Left panel shows results with 1 Gbps interconnection network. Right panel shows detail view of transition region with 1 Gbps interconnection. Bottom panel shows the much higher throughput attainable with a 10 Gbps interconnection.

communication path. The higher bandwidth makes eight processor configurations feasible. Noting the scale of the horizontal axis, the graph shows that much higher throughput is attainable with the faster interconnect. With such high bandwidth available, the time to distribute incoming SAR images to multiple processors dominates the chip processing time only under very coarse approximations. At chip processing rates below around 1400 per second, corresponding to approximations finer than d_1, it is preferable to employ eight processors.

6. CONCLUSIONS

We have illustrated a process of connecting a computational model with a model-based inference algorithm to relate the accuracy of a recognition system to required throughput in terms of ATR system parameters. The resulting accuracy-throughput curves can be used for exploring the design space. Parallelism was exploited in multi-processor systems by distributing an incoming SAR image to each processor and dynamically assigning regions of the search space to each based upon previous computations. Under this arrangement, the average time to classify an image consists of the time required to distribute the SAR image, the time to process initial approximation levels before processor local memory is exhausted, and the time to process remaining approximation levels in which portions of local processor memory must be overwritten by incoming model parameters.

The experimental results for the four-class recognition problem indicate that at high chip rates, corresponding to the use of only coarse models of sensor data, systems with fewer processors have an advantage because of reduced overhead involved with initial image distribution. At lower chip rates, corresponding to more sophisticated sensor data models, larger numbers of processors become attractive because of the increased computational requirements. Increasing the bandwidth of the interconnection network shifts the break-even point in the direction of higher chip processing rates.

ACKNOWLEDGMENTS

This work was supported in part by the US Army Research Office grant DAAH04-95-1-0494, the Office of Naval Research grant N00014-98-1-06-06, the Boeing Foundation, and by the DARPA VLSI Photonics Program under grant DAAL01-98-C-0074.

REFERENCES

1. D. E. Culler, J. P. Singh, and A. Gupta. *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann Publishers, San Francisco, 1999.
2. M. D. DeVore and J. A. O'Sullivan. A performance-complexity study of several approaches to automatic target recognition from synthetic aperture radar images. *IEEE Transactions on Aerospace and Electronic Systems*. Submitted for publication.
3. M. D. DeVore, J. A. O'Sullivan, S. Anand, and N. A. Schmid. Probabilistic approach to model extraction from training data. In E. G. Zelnio, editor, *Algorithms for Synthetic Aperture Radar Imagery VIII, Proc. of SPIE*, 2001.
4. M. D. DeVore, N. A. Schmid, and J. A. O'Sullivan. Analytical and experimental performance-complexity tradeoffs in ATR. In *Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems, and Computers*, Oct. 2000.
5. M. A. Franklin, A. Mahajan, and R. M. Arthur. Parallel implementations of 3D synthetic-focus ultrasonic image generation using MPI. In *Proceedings of the 12th IASTED International Conference on Parallel and Distributed Computing Systems*, Las Vegas, Nevada, Nov. 2000.
6. J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, San Francisco, 1996.
7. J. A. O'Sullivan and M. D. DeVore. Performance-complexity tradeoffs for several approaches to ATR from SAR images. In E. G. Zelnio, editor, *Algorithms for Synthetic Aperture Radar Imagery VII, Proc. of SPIE*, volume 4053, 2000.
8. J. A. O'Sullivan, M. D. DeVore, V. Kedia, and M. I. Miller. Automatic target recognition performance for SAR imagery using a conditionally Gaussian model. *IEEE Transactions on Aerospace and Electronic Systems*. Accepted for Publication.
9. N. A. Schmid and J. A. O'Sullivan. Thresholding method for reduction of dimensionality. *IEEE Transactions on Information Theory*. Accepted for Publication.