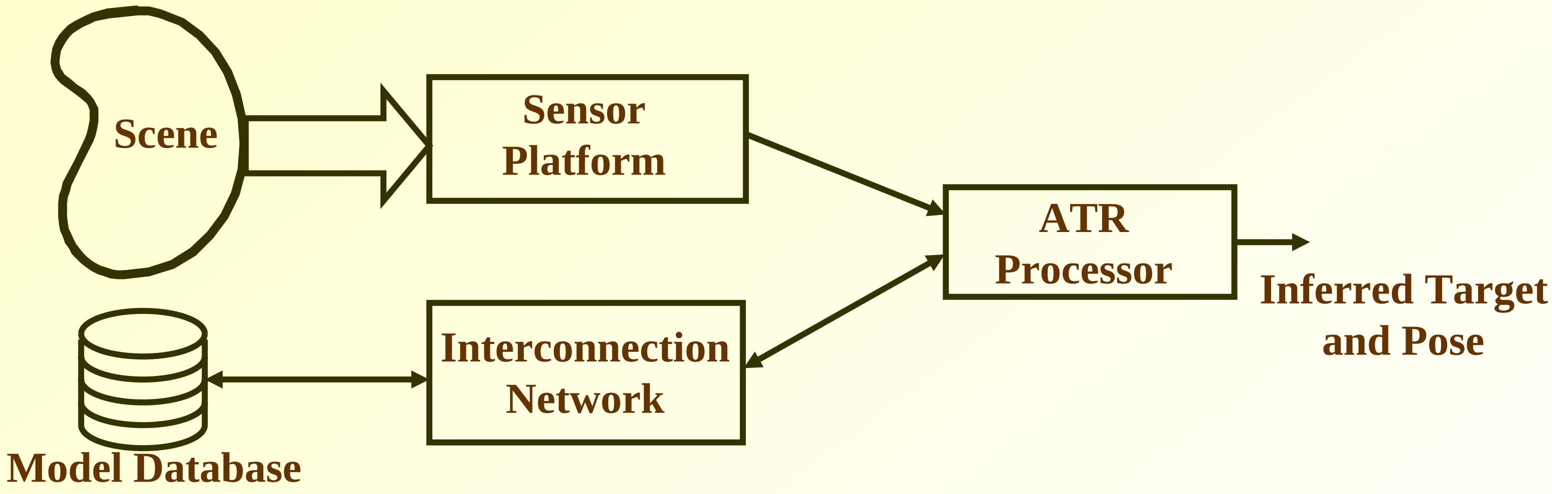
Networked ATR Systems Design Considerations: System Performance and Resource Constraints

Joseph A. O'Sullivan
Washington University in St. Louis
jao@ee.wustl.edu

Michael D. DeVore
University of Virginia
mdevore@virginia.edu

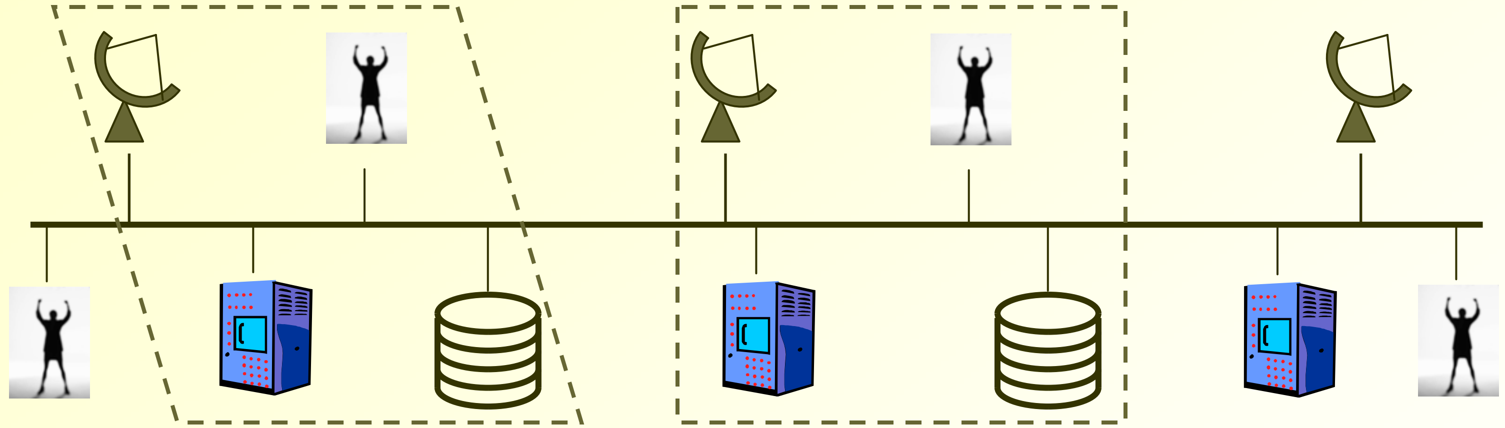
Alan J. Van Nevel
Naval Air Warfare Center
vannevelaj@navair.navy.mil

Single ATR System



- ATR processor solves a sequence of inference problems
  - May sport multiple CPUs
  - Problem computations may overlap
- Sensor platform collects scene info for processor
  - May incorporate multiple measurements
- Model database represents known objects
  - May support multiple resolutions
- Network connects sensor, processor, and database

Networked ATR System



Goal: Near optimal use of resources in a **dynamic** environment

- System operates within constraints
  - Accuracy, bandwidth, computational capability, throughput
- Demands and resources may change during an engagement
  - Damage, jamming, new capabilities, preemption, budgeting, etc.
- Successively refinable search algorithms to adjust operating point on the fly

Model-Based ATR

Maximum-likelihood from statistical models

$$\begin{bmatrix} \hat{a} \\ \hat{\theta} \end{bmatrix} = \underset{a,\theta}{\operatorname{argmax}}\, p(\boldsymbol{r} \mid a, \theta)$$

Where

$\boldsymbol{r}$ is an observation vector

a is a target class

θ is a target pose

For fixed a, function p constitutes a target model

- Generally estimated from training data
- Often of a complexity-restricted class

Likelihood Approximations

- Consider a sequence of approximations p_1, p_2, … with p_{n+1} a better approximation than p_n

$$\Pr[\text{error} \mid p_{n+1}] \leq \Pr[\text{error} \mid p_n]$$

- Let $C(p_n)$ be a measure of average resource consumption when approximation p_n is employed
- Since better approximations often involve higher complexity, we expect

$$C(p_{n+1}) \geq C(p_n)$$

- Static implementation by selecting n to satisfy constraints on $\Pr[\text{error} \mid p_n]$ and $C(p_n)$

Example: Approximating Likelihoods

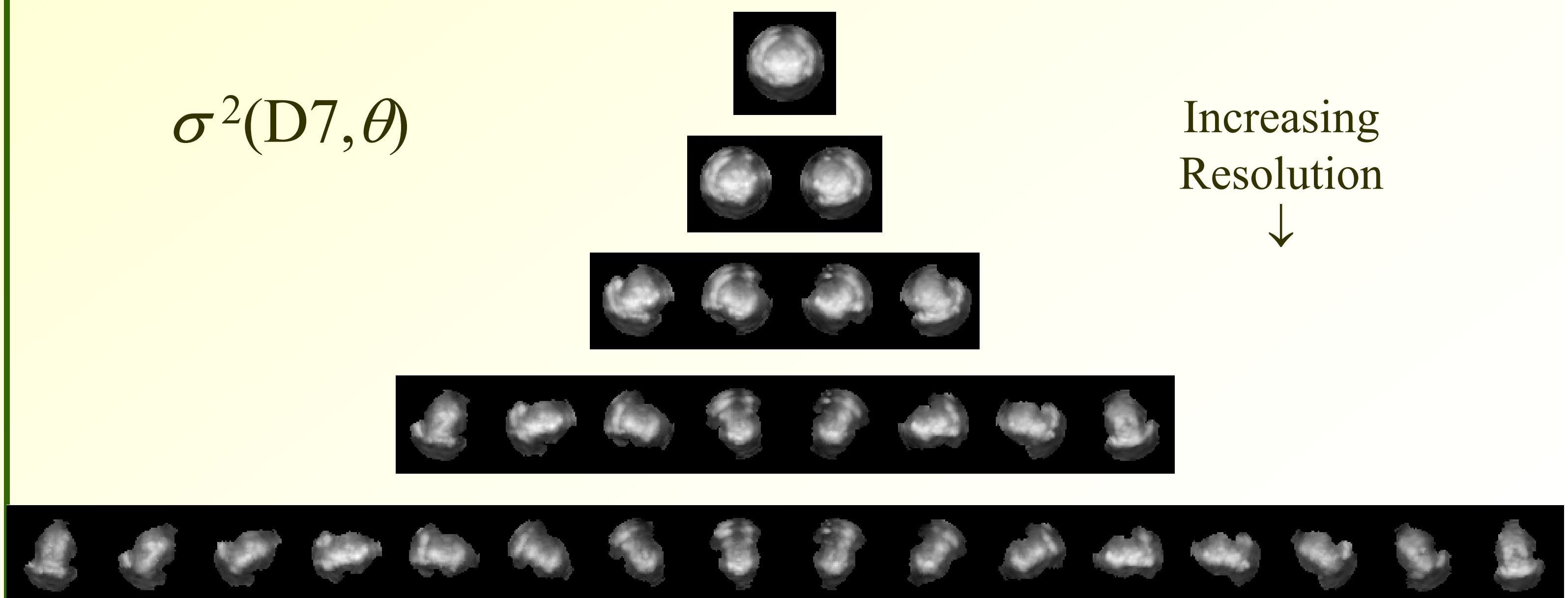
- Model the SAR image of a bulldozer as a function of azimuth

$$r_i \sim \mathrm{CN}(0,\ \sigma_i^2(a,\theta)\ )$$

- Likelihood function depends on parameter function σ_i^2
- Sequence of piecewise constant approximations

$\sigma^2(\mathrm{D7},\theta)$

Increasing
Resolution
↓



Increasing azimuth →

Dynamic Reconfigurability

- Seek algorithms that dynamically adjust to fit requirements
  - can't necessarily determine *n* ahead of time
- Let $\Delta C(p_{n+1})$ be the additional resources consumed using p_{n+1} assuming problem with p_n already solved
- Good designs characterized by

$$C(p_{n+1}) \approx C(p_n) + \Delta C(p_{n+1})$$

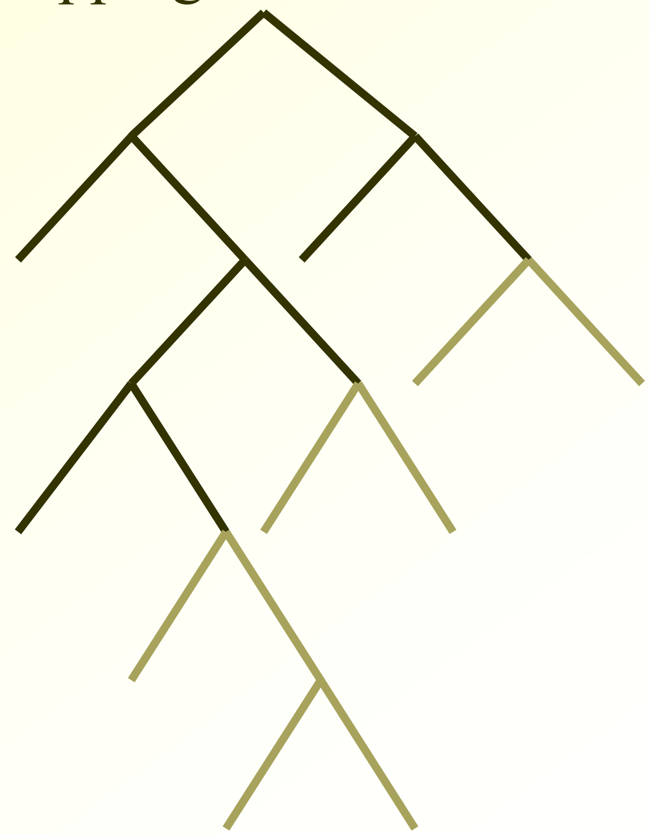
- Produce a sequence of answers (a_1, θ_1), (a_2, θ_2), … with increasing accuracy and resource consumption

$$C'(p_{\mathrm{n+1}}) = C(p_1) + \Delta C(p_2) + \ldots + \Delta C(p_{n+1})$$

  - stop when resource allocation exhausted

Example: Delta Cost Functions

- Let cost be average number of bits read from database
- 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:

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

Sequence Selection

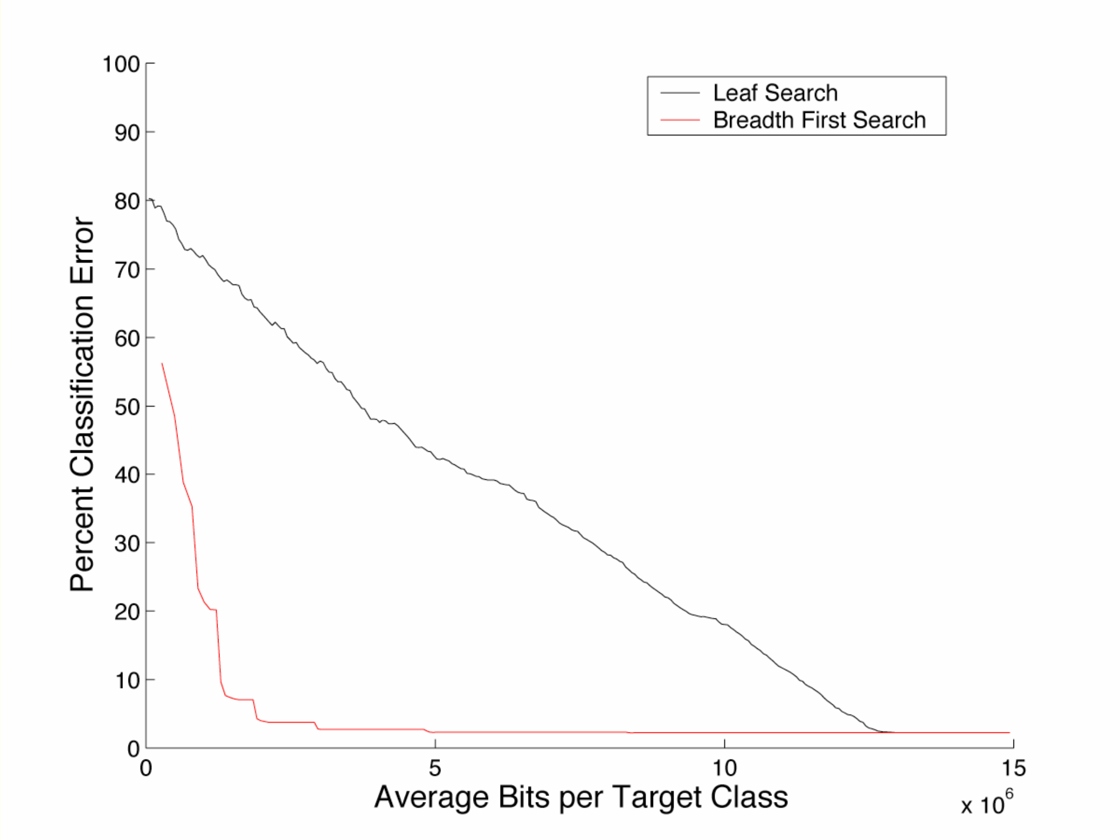
- Selection of sequence p_n drastically affects the parametric curve Pr[error | p_n] vs. $C'(p_n)$
- Good designs decrease error rapidly at start of sequence
  - useful results even if search is terminated early
  - can make use of additional resources if available
- Example: Error probability vs. database communication
  - Design #1: “Leaf Search”

    Refine sequential 1.4^o intervals
  - Design #2: “Breadth First”

    Divide the most likely interval

Example: Search Algorithm

Error rate vs. bits transmitted from database to processor



- Classification depends on extent of search
- Eventually, search covers all possibilities
- Breadth-first search quickly finds good solutions (a, θ)
- Small overhead present with ordered searches

Other Consumption Measures

- Network bandwidth is one of many types of resources
- Other average rates of resource consumption:
  - Elapsed time per classification
  - CPU cycles per classification
  - Database (magnetic) storage per model class
  - Power dissipation
- Changing resource consumption rates due to:
  - Variation in application requirements
  - Reallocation of resources to higher priority tasks
  - Damaged or offline computation elements
  - Disrupted communication paths
  - Power considerations

Example: Throughput

Time to process through approximation p_m includes time to:

- distribute SAR image to each CPU
- process each approximation until local memory is full
- process each remaining approximation

$$T_{\text{chip}} = \frac{S_c}{\text{BW}}\left\lceil \log_2(P+1)\right\rceil + \sum_{l=1}^{l_{\text{mem}}} 2^{l-1} N_{\text{T}} \tau_{d_l} + \sum_{l=l_{\text{mem}}+1}^{m} 2^{l-2} N_{\text{T}}\left(\tau_{d_l} + \tau'_{d_l}\right)$$

Where:

S_c = bits per SAR image

P = number of processors

BW = network bandwidth

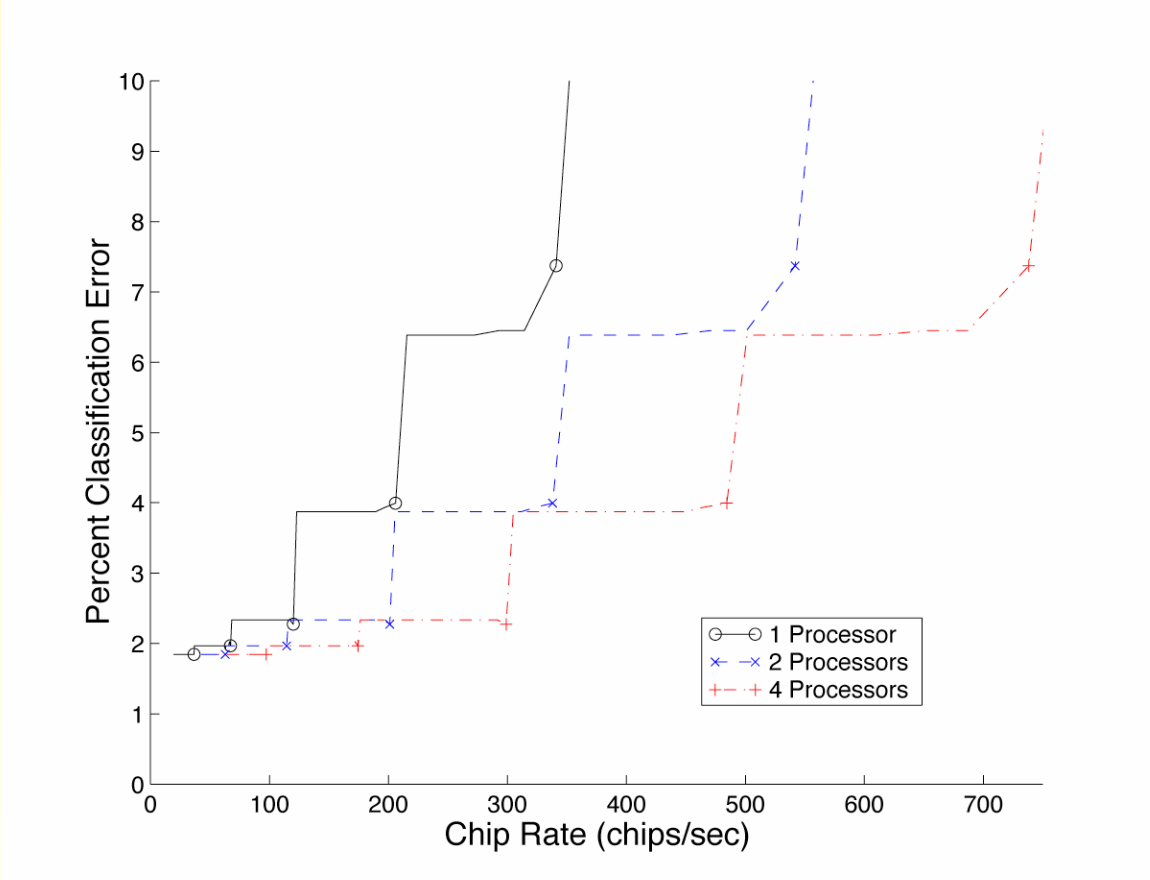
N_{T} = number of target classes

τ_d = average time per template at approximation p_d exploiting hierarchy.
Receive variance, compute variance, and compute likelihoods.

τ'_d = average time per template without exploiting hierarchy.
Receive variance and compute likelihood.

Example: Throughput

- High throughput corresponds to coarse approximations
- Markers denote doubling in # of representation intervals



With: Chip Rate = $1/T_{chip}$ | 1 GHz clock | 64 bit read per clock cycle
4 target classes | 0.5 CPI | 25 target locations
10 Gbps interconnection

Opportunity: Dynamic Bandwidth

- Co-design of search algorithms, object models, and data compression
- Search algorithms exploiting nested model families
  - Quickly locate good candidate hypotheses
- Object representations to support search algorithms
  - Likelihood sequences determined during search
  - Efficient manipulation by processor
  - Low ΔC in terms of bit rate
- Data compression optimized for recognition
  - Typical compression designed for low visual degradation
  - Model and sensor data compression for accurate recognition

Opportunities: Dynamic Environments

- Models for network resource consumption
  - Multiple inference objectives using shared structures
- Achievable accuracy surfaces
  - Vector-valued resource consumption measures
- Characterize robustness relative to varying resources
  - Basis for comparing alternate designs
- Feasible resource allocations given accuracy and resource constraints
  - Decision aid for dynamic reallocation

Opportunities: System Design

- System architecture
- Partitioning effort across distributed elements
- Modules which can operate in concert or isolation

Plan

work with china lake to identify scenarios of interest
sensor(s) & specs
operational scenario
- time requirements (min-max range)
- local vs. distributed processors
- number of targets
- performance goals

Use available data
develop simulations
- apply methodology to scenario

Extend theoretical/analytical results
Demonstrate utility of approach in a problem of interest to Navy