Representations of Shape for Structural Inference in Infrared Scenes

Aaron D. Lanterman, Michael I. Miller, Donald L. Snyder

Center for Imaging Science (http://cis.wustl.edu)
Electronic Systems and Signals Research Laboratory, Campus Box 1127
Washington University, St. Louis, Missouri 63130-4899
adl@essrl.wustl.edu, http://essrl.wustl.edu/~adl

ABSTRACT

Infrared scenes are modeled as consisting of two kinds of targets: flexible 2-D models for simple shapes and rigid 3-D faceted models for detailed targets. The flexible models permit rapid saccadic detection of targets and accommodate "clutter" objects not present in the target library. The rigid model library contains specific vehicles or other objects we wish to discriminate.

A likelihood model based on sensor statistics is combined with a prior distribution on possible scenes to form a posterior distribution for Bayesian inference. Nuisance parameters associated with the radiant intensities of the background and object facets are adaptively estimated as the inference proceeds.

A general Metropolis-Hastings acceptance/rejection algorithm for sampling from the posterior distribution is proposed.

Keywords: pattern theory, shape models, Markov chain Monte Carlo, automatic target recognition, infrared, FLIR

1 INTRODUCTION

The tremendous variability apparent in forward-looking infrared (FLIR) imagery makes the design of automatic target recognition systems for such sensors a daunting task. Targets take on vastly different manifestations with changes in orientation and thermodynamic state. Even if the ATR system is equipped with a thorough knowledge of the targets of interest and the infinite number of ways they may appear, any interesting scene is likely to contain elements not present in the algorithm's repertoire. Such "clutter" may cause false alarms or otherwise confuse the recognition and location of other targets.

Grenander's pattern theory[1–3] provides a promising mathematical foundation for tackling such challenges.[4] Application of the theory proceeds by 1) forming representations of complex scenes, 2) characterizing the observation process that describes how those scenes are sensed, and 3) engineering inference algorithms to deduce the representation underlying the sensed data. This philosophy has yielded tremendous advances in medical imaging[5,6] and computational linguistics,[7] where vast databases of knowledge are available. While applying pattern theory to ATR is similar in principle, several difficulties conspire to complicate matters. In biomedical applications, the data are often collected under controlled laboratory conditions, and much *a priori* information about the subject of study is available; unfortunately, ATR systems frequently operate in widely varying environments beyond the operator's influence, and *a priori* knowledge of environment is often severely limited.

For the past several years, our group has explored pattern-theoretic algorithms, based on general Metropolis-Hastings jump-diffusion processes, for the understanding of infrared scenes.[8,9] In that work, targets were considered members of a set of vehicles characterized by 3-D CAD models. The algorithm estimated number of targets, as well as their types, positions, and orientations. Over the course of that work, two fundamental needs became apparent, which we begin to address here: 1) The algorithm sought to represent every object in the scene with a target from the library, whether it was truly a member of the library or not, and 2) the initial detection of objects required manipulation of targets at their highest level of detail, sampling from the full space of positions, orientations, and types. Interestingly, both weaknesses arise from the rigidity of the representation, which suggests that we enrich the expressive power of the parameter space.

In the introduction to a series of lectures given at Washington University in 1995, Grenander suggested that pattern theory was not so much about the "representation of objects" as it was about the "representation of *knowledge* about objects." Heeding this advice, the agenda introduced here seeks to build varying degrees of knowledge into the representation; inference algorithms which flow between layers of detail are then a natural consequence. In this expanded framework, detection is accomplished via simple shapes, which can be rapidly manipulated to determine the overall shape of a target before the more detailed representations, which often are the most computationally expensive to manipulate, come into play. This addresses the second difficulty. The first

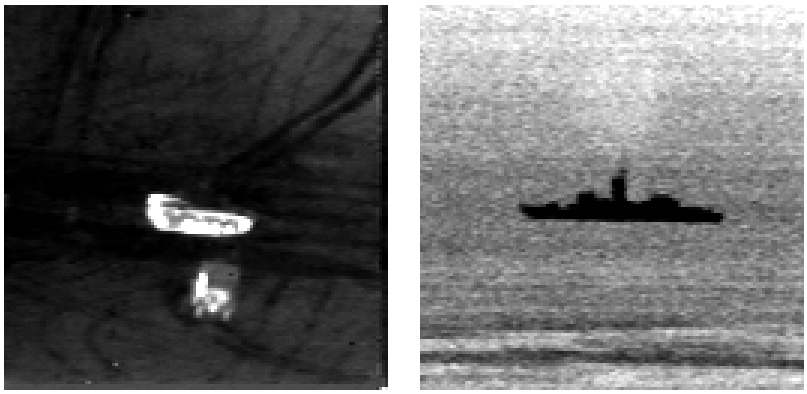

Figure 1: Example FLIR imagery. Left panel shows a side view of an M60 and a front fiew of a truck (courtesy Dr. Richard Sims, U.S. Army Missile Command.) Right panel shows a ship (courtesy Howard McCauley, Naval Air Warfare Center.) Notice in the left image, the targets have brighter intensity than the background, and conversely in the right image.

is dealt with by providing two classes of detailed targets: 3-D CAD representations for known target types, and highly flexible 2-D shapes to accommodate clutter.

2 REPRESENTATION

Let us begin with the shape models, both rigid and flexible. Denote an N-object configuration as[8]

$$X_N = \{(\mathcal{A}, \mathcal{X}_A) : A \in \mathcal{A}\}^N, \tag{1}$$

where $\mathcal{X}_A$ is the parameter space associated with target class A.

Since the number of objects is not known in advance, the complete object configuration space is

$$\mathcal{X} = \bigcup_{N=0}^{\infty} X_N. \tag{2}$$

2.1 Shapes

This study explores three kinds of shapes: rigid targets of interest, such as specific vehicles; simple 2-D shapes for rapid saccadic detection; and generic 2-D shapes for clutter modeling which we will call *blobs*. Denote $\mathcal{A} = \mathcal{A}_{rigid} \cup \mathcal{A}_{sac} \cup \mathcal{A}_{blob}$.

For rigid targets of class $A \in \mathcal{A}_{rigid} = \{M2, M60, T62, \ldots\}$, we have a parameter space $\mathcal{X}_A = \Re^2 \times [0, 2\pi)$, representing position and orientation on the 2-D ground plane.[8] Semi-rigid objects such as tanks with rotating turrets can be described with an additional small number of parameters. In this work we will simply focus on $\Re^2 \times [0, 2\pi)$.

Here, $\mathcal{A}_{sac}$ will consist of rectangles with vertices at the lattice points. Their parameter space consists of the x-coordinates of the left and right edges and the y-coordinates of the top and bottom edges. Other representations could be employed as well. Rectangles are easily manipulated, offering rapid saccadic detection. They will suffice for our current purposes; more general objects, such as parallelograms and arbitrary polygons, could be useful as well.

2.2 Blobs for clutter modeling

The pattern theoretic method often begins by defining *templates* which specify the overall structure of objects, and proceeds by formulating transformation groups and priors on those groups to accommodate typical variability. This tactic has worked well on subjects such as hands, leaves,[10,2,11] mitochondria,[3] amoebas,[12] and the rigid targets of interest in ATR.[13]

Clutter greatly complicates matters. The breadth of objects which blobs will need to accommodate precludes any obvious template; any priors placed on transformations of such a template would be necessarily uninformative. Thus, we are motivated to explore a somewhat more nonparametric approach, allowing the algorithm to mold blobs into whatever shapes they need to be.

Let $\mathcal{P} \subset \mathbf{Z}^2$ denote pixels on the FLIR detector. We define *blobs* via the following:

DEFINITION 2.1. *A* binary lattice pattern *on $\mathcal{P}$ is a function $f : \mathcal{P} \to \{0, 1\}$.*

DEFINITION 2.2. *The* primary compass directions *are*

$$PCD = \{(0,1), (0,-1), (1,0), (-1,0)\}. \tag{3}$$

DEFINITION 2.3. *A binary lattice pattern is a* blob *if for all $p_1, p_2 \in \mathcal{P}$ such that $f(p_1) = f(p_2)$, there exists a sequence of directions $(d_1, \ldots d_N) \in PCD^N$ such that*

$$f(p_1) = f(p_1 + \sum_{i=1}^{J} d_i) \text{ for all } J = 1, ..., N, \tag{4}$$

i.e., for any two foreground points, there is a path on the lattice which connects them while staying in the foreground, and for any two background points, there is a path on the lattice which connects them while staying in the background.

It will be convenient to define a predicate: $IsBlob[f] = 1$ if f is a blob, 0 otherwise. With this, define the following:

DEFINITION 2.4. *The set of blobs on $\mathcal{P}$ will be called $\mathcal{B}$ =* blobs.

$$\mathcal{B} = \{f : \mathcal{P} \to \{0, 1\} : IsBlob[f] = 1\}. \tag{5}$$

DEFINITION 2.5. *The* boundary operator $b : \mathcal{B} \to \mathcal{B}$ *which maps a blob f to its* boundary $b(f)$ *is given by*

$$b(f)(p) = \bigvee_{d \in PCD} f(p) \oplus f(p + d), \tag{6}$$

where $\oplus$ represents the exclusive-or operation.

The following similar definitions will be useful in later sections when we engineer Monte Carlo algorithms for sampling from B.

DEFINITION 2.6. *The* exterior boundary $eb(f)$ *of a blob f is given by*

$$eb(f)(p) = not[f(p)] \wedge b(f)(p), \tag{7}$$

i.e. all boundary points in the foreground.

DEFINITION 2.7. *The* interior boundary $ib(f)$ *of a blob f is given by*

$$ib(f)(p) = f(p) \wedge b(f)(p), \tag{8}$$

i.e., all boundary points in the background.

DEFINITION 2.8. *The* allowable additions $aa(f)$ *of a blob f is the subset of the exterior boundary such that adding a point anywhere along the $aa(f)$ yields another blob:*

$$aa(f)(p) = eb(f)(p) \wedge IsBlob[f \vee \delta(p)], \tag{9}$$

where $\delta(p)$ is the single-point blob with 1 at p and 0 elsewhere.

DEFINITION 2.9. *The* allowable additions $aa(f)$ *of a blob f is the subset of the interior boundary such that deleting a point anywhere along the $ad(f)$ yields another blob:*

$$ad(f)(p) = ib(f)(p) \wedge IsBlob[f - \delta(p)]. \tag{10}$$

Adding a point in the allowable additions will not form a "hole" in the blob, and deleting a point in the allowable deletions will not disconnect the blob into two separate blobs.

A simpler, computationally convenient characterization of $aa(f)$ and $ad(f)$ is helpful. At a given point p, imagine tracing along its eight neighbors; for that point to be in the allowable region, two and only two transitions, one from 0 to 1 and one from 1 to 0, should be observed as we move along the neighbors. Reusing notation somewhat and adding 1's (true) and 0's (zeros) as if they were ordinary integers, define

$$TwoChanges(f)(p) = \begin{cases} 1 & \text{if } [f(p+E) \oplus f(p+SE)] + [f(p+SE) \oplus f(p+S)] \\ & \quad + [f(p+S) \oplus f(p+SW)] + [f(p+SW) \oplus f(p+W)] + [f(p+W) \oplus f(p+NW)] \\ & \quad + [f(p+NW) \oplus f(p+N)] + [f(p+N) \oplus f(p+NE)] + [f(p+NE) \oplus f(p+E)] = 2 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where in (row,column) notation, $E = (0,1)$, $SE = (-1,1)$, $S = (-1,0)$, etc.

Then straightforward (albeit tedious) enumeration of cases shows that

$$aa(f) = eb(f) \wedge TwoChanges(f) \tag{12}$$

$$ad(f) = ib(f) \wedge TwoChanges(f) \tag{13}$$

3 OBSERVATION MODELING

Now that we have defined the parameter space, we explore the second part of the pattern theoretic methodology: the description of the observation process.

3.1 A Poisson likelihood model

Let x denote the parameters we wish to estimate, such as the number of objects, their types, positions, shapes, and so on. Treat these as random parameters in the Bayesian formulation.

Suppose that the image array data $d(\cdot)$ is Poisson distributed with mean $\lambda(\cdot)$ which is proportional to the radiant intensity of objects in the scene. For now, we will neglect the effects of camera point-spread. The loglikelihood of the data is[14]

$$L(d|\lambda) = -\sum_i \lambda(i) + \sum_i d(i) \ln \lambda(i). \tag{14}$$

Of course, λ itself is a complicated, highly nonlinear function of the configuration parameters. Suppose the desired parameters x specify a set of image regions R_j and that the intensity $\lambda(\cdot)$ is a constant λ_j across region R_j. A shape in class $\mathcal{A}_{flex}$ will be characterized by a single uniform intensity across its extent. A complex, 3-D CAD target in class $\mathcal{A}_{sac}$ or $\mathcal{A}_{blob}$ may consist of several regions, with each region consisting of several connected facets; for instance, the gun barrel and engine exhaust of a tank may be much hotter than the tank's body, depending on whether the gun was recently fired and whether the engine is running. We will assume that the radiant intensity is constant across each of these thermodynamic regions.

3.2 Nuisance parameters for backgrounds and blobs

The challenges unveiled in the introduction motivate the exploration of algorithms which require little prior knowledge of the intensities of the targets or the background. To best characterize that lack of knowledge, treat the intensities of the background and the targets in $\mathcal{A}_{sac}$ and $\mathcal{A}_{blob}$ as unknown, nonrandom parameters.

The λ_j's are nuisance parameters. If we treat them as nonrandom variables parameterizing the loglikelihood, we may replace the λ_j's in the loglikelihood with their maximum-likelihood estimates for a given realization of x. This was the technique employed to deal with the nuisance parameters in the HANDS study.[10] There the nuisance parameters were the intensities of the hand and of the background.

In this case, the ML estimate $\hat{\lambda}_j$ of the intensity λ_j of a region R_j containing N_j pixels is simply the average of the data values $d(\cdot)$ in that region

$$\hat{\lambda}_j = \frac{1}{N_j} \sum_{k \in R_j} d(k). \tag{15}$$

3.3 Nuisance parameters for detailed 3-D targets

A wide variety of IR simulation codes such as PRISM[15] have proven useful in simulating scenes for testing ATR algorithms, training human operators, and predicting the performance of both. Since these codes solve

heat transfer equations to produce their result, they require detailed models of target thermodynamics describing internal heat sources and transfer characteristics.

We will take an empirical statistical approach to modeling target thermodynamics and construct a prior distribution on the radiant intensities of target facets. By simulating a large number of radiance measurements, taken while varying environmental and internal heating parameters over reasonable ranges, we generate a population of radiance profiles to which we apply principal component analysis.[16] Assuming a Gaussian model, the first few eigenvectors provide a parisimonous representation of the covariance.

In the previous section, we substituted ML estimates since no convenient prior knowledge seems available; here, with a prior, standard Bayesian approaches apply. Ideally the nuisance parameters should be averaged out. Since the resulting integral is quite complicated and the integrand is likely to be rather peaked, we may employ the standard Laplace approximation (previously proposed in another context for Bayesian decision[17]), which is based upon a Taylor series expansion around the maximizing parameters. These calculations are currently being implemented. For the remainder of this paper, the radient intensities of the CAD models are assumed to be known.

3.4 Full loglikelihood

Substituting these estimates into the loglikelihood (14) yields

$$L(d|\hat{\lambda}) = -\sum_j N_j \hat{\lambda}_j + \sum_j \sum_{k \in R_j} d(k) \ln \hat{\lambda}_j. \tag{16}$$

4 INFERENCE

The posterior density described in the previous section, as is often the case in pattern theoretic applications, is analytically forboding; hence, we turn to *Markov chain Monte Carlo*[18–21] techniques for sampling from the posterior. MCMC techniques are enjoying increasing prominence in the statistical community; in particular, the recent volume edited by by Gilks, Richardson, and Spiegelhalter addresses these algorithms from a practical standpoint.[22]

The configuration space involves continuous parameters, such as the positions and orientations of targets of interest, and discrete parameters, such as the number of targets and their types; we also have the discrete parameters associated with blobs.

Our inference engines will be based on Metropolis-Hastings acceptance/rejection algorithms. Specification of a Metropolis-Hastings algorithm requires choosing a *proposal density* $r(x_{old}, x_{prop})$ (a density in x_{prop} parameterized by x_{old}). Let π denote the desired density. At each step, the algorithm draws a proposal x_{prop} from the proposal density. It accepts the proposal with *acceptance probability*

$$\alpha(x_{old}, x_{prop}) = \min\left\{ \frac{\pi(x_{prop}) r(x_{prop}, x_{old})}{\pi(x_{old}) r(x_{old}, x_{prop})}, 1 \right\}. \tag{17}$$

If the proposal is rejected, the algorithm remains at x_{old}. The generated chain $x(0), x(1), x(2), \ldots$ has the property that the density of $x(N)$ converges to the desired density π. There are many variations on this theme, including "deterministic scan" algorithms which cycle through different proposal densities for different variables or blocks of variables in a nonrandom fashion. Consult the references[18,19,22] for technical details about necessary conditions and modes of convergence. Previously reported jump-diffusion algorithms are easily assimilated into the general Metropolis-Hastings framework, as will be seen in Sec. 4.4.

At times, our inference engine must propose small changes to the poses and shapes of individual targets; at others, large changes, involving the addition or deletion of targets, or the changing of a target's representation, for instance from a saccadic shape to a specific 3-D target or a blob. In the following sections, we will restrict the discussion to individual aspects of this process, showing sampling algorithms operating on subsets of the parameter space for instructive purposes.

4.1 Detection

In analogy with the siccadic motion of human vision, we employ simple shapes to rapidly scan the data for interesting structures, as suggested by Grenander.[23] The detection scheme illustrated here uniformly covers the data with equally sized squares, and computes the loglikelihood associated with placing a square at each location. It does this at a variety of resolutions and chooses from all the candidate squares probabilistically. In addition to

being effective, the scheme yields a convenient recursive structure for speedy execution.

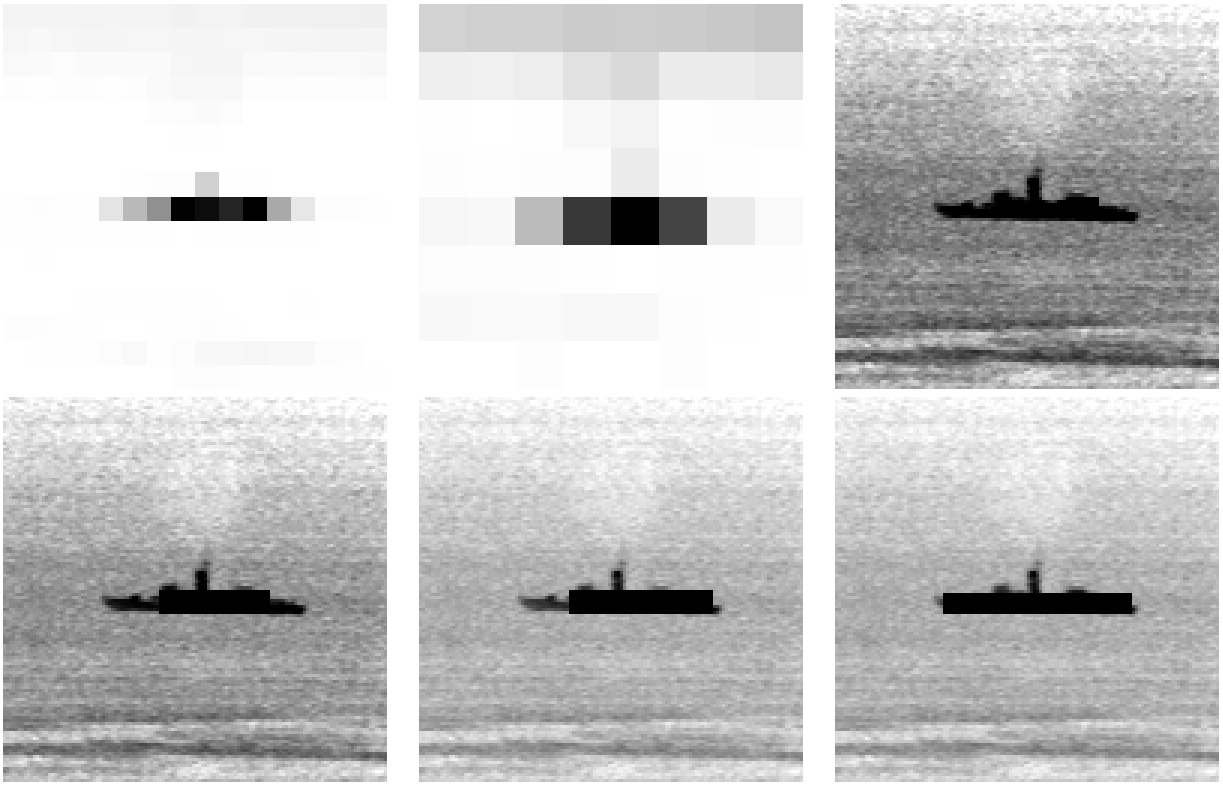

Figure 2: Finding the overall extent of a ship.

The first two panels of Fig. 2 show loglikelihoods for two different resolutions for the data in the right panel of 1. In this instance, the finer resolution, near the right side of image, yields the greatest gain in likelihood.

Notice that with the adaptive likelihood formulated in Sec. 3, the saccadic detector does not seek "bright" or "dark" areas per se; it seeks areas which are sufficiently different from the remainder of the image.

4.2 Saccadic shape manipulation

The saccadic shapes must be manipulated to determine the overall extent of the target. The remaining panels of Fig. 2 demonstrate a straightforward Metropolis-Hastings algorithm refining the rectangle to develop a rough sense of the shape of the ship.

4.3 Blob manipulation

An elegant procedure for sampling from the space of blobs requires more thought. The issues involved illustrate some questions that crop up frequently in designing Metropolis-Hastings samplers. Using the definitions from Sec. 2.2, we define the set of blobs which can be reached via the addition (birth) and deletion (death) of a point along the boundary:

$$\mathcal{T}_b^1(f) = \{f \vee \delta(p) : p \in \mathcal{P}, aa(f)(p) = 1\} \quad (18)$$

$$\mathcal{T}_d^1(f) = \{f - \delta(p) : p \in \mathcal{P}, ad(f)(p) = 1\} \quad (19)$$

Let $\mathcal{T}^1(f) = \mathcal{T}_b^1(f) \cup \mathcal{T}_d^1(f)$ be the subset of blobs which can be reached in one Metropolis move. The proposal should be drawn probabilistically from this set. Here are two possibilities (there are many others):

1) An easy proposer:

$$r(x_{old}, x_{prop}) = \frac{1}{|\mathcal{T}^1(x_{old})|} \text{for } x_{prop} \in \mathcal{T}^1(x_{old}), 0 \text{ otherwise} \tag{20}$$

$$\alpha(x_{old}, x_{prop}) = \min\left\{\frac{\pi(x_{prop})|\mathcal{T}^1(x_{old})|}{\pi(x_{old})|\mathcal{T}^1(x_{prop})|}, 1\right\} \tag{21}$$

2) A smart proposer:

$$r(x_{old}, x_{prop}) = \frac{\pi(x_{prop})}{\sum_{x' \in \mathcal{T}^1(x_{old})} \pi(x')} \text{for } x_{prop} \in \mathcal{T}^1(x_{old}), 0 \text{ otherwise} \tag{22}$$

$$\alpha(x_{old}, x_{prop}) = \min\left\{\frac{\sum_{x' \in \mathcal{T}^1(x_{old})} \pi(x')}{\sum_{x'' \in \mathcal{T}^1(x_{prop})} \pi(x'')}, 1\right\} \tag{23}$$

The first algorithm proposes uniformly. Each proposal involves little computation, and a simple likelihood comparison is all that is needed for the acceptance probability. The second examines the likelihood of each move, and selects based on that. Although each proposal (and acceptance calculation) involves substantially more computation than the uniform selection, each individual proposal is more likely, yielding more powerful moves and a higher acceptance rate. Our initial experiments explored the first algorithm. We found it to be excruciatingly slow, which motivated implementation of the second for the examples shown here.

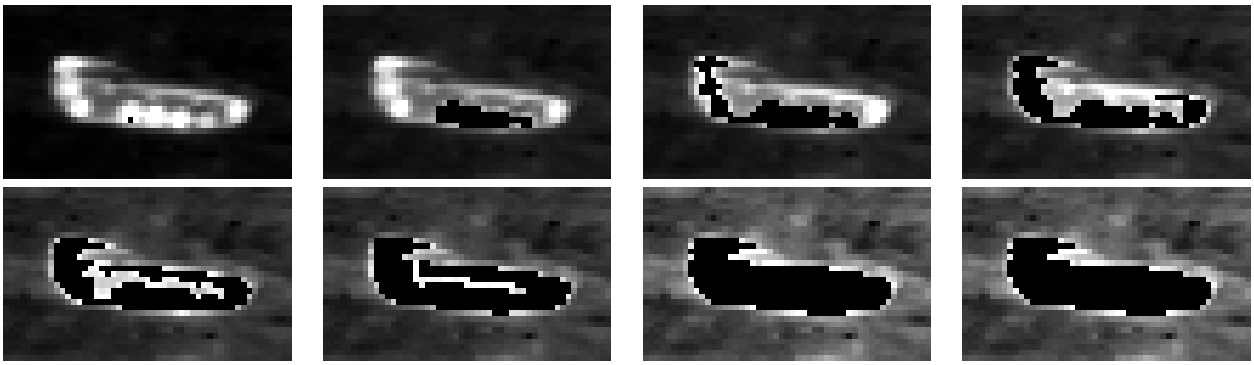

Figure 3: Generic shape inference on a tank, starting from a shape consisting of a single point in a hot region. The Metropolis-Hastings sampler adds pixels to fill out the tank body.

To illustrate the behavior of the sampler, examples using the M60 are shown in Figs. 3 and 4, beginning with a single point and a rectangle which is much larger than the target, respectively. In the first example, the point, which is on a bright section of the tank, snakes along the outside, filling in the bright regions first, and then moving in to fill the less-bright central area. In the second example, the block decays to isolate the tank. The lower left panel is particularly intriguing; notice there is a bright spot above the tank which the algorithm is loathe to disengage. After it pulls in the remainder of the background, it finally relents and withdraws the tentacle.

Of course, if these targets are present in the algorithm's 3-D target library, that would be a preferable representation, as discussed next.

4.4 3-D target manipulation

An example of a Langevin diffusion process, essentially a gradient ascent with an additional random term, refining the estimate of the orientation[8] of the M60, is shown in Fig. 5. The MICOM data were collected from a FLIR mounted aboard a helicopter. Altitude and range to target were collected along with the data, permitting rough reconstruction of the viewing geometry.

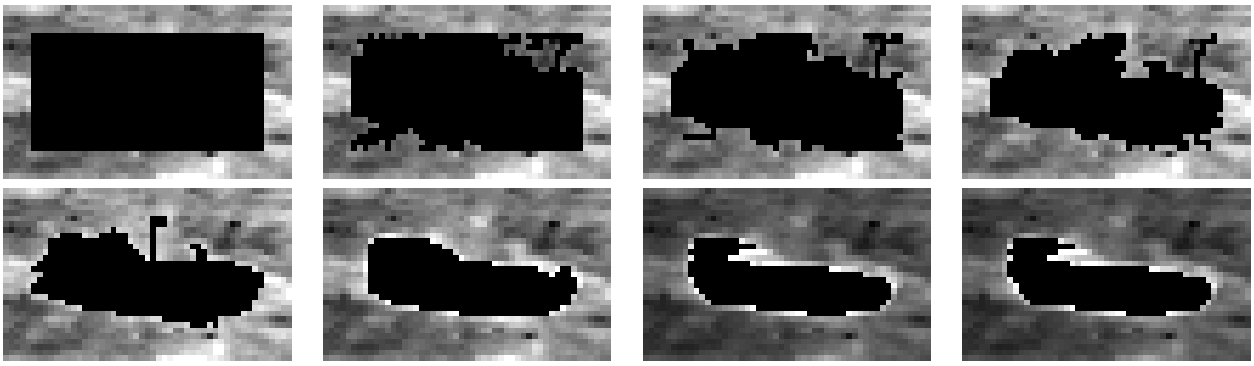

Figure 4: Generic shape inference on a tank, starting with a retangle larger than the tank. The Metropolis-Hastings sampler removes pixels until the tank body is left.

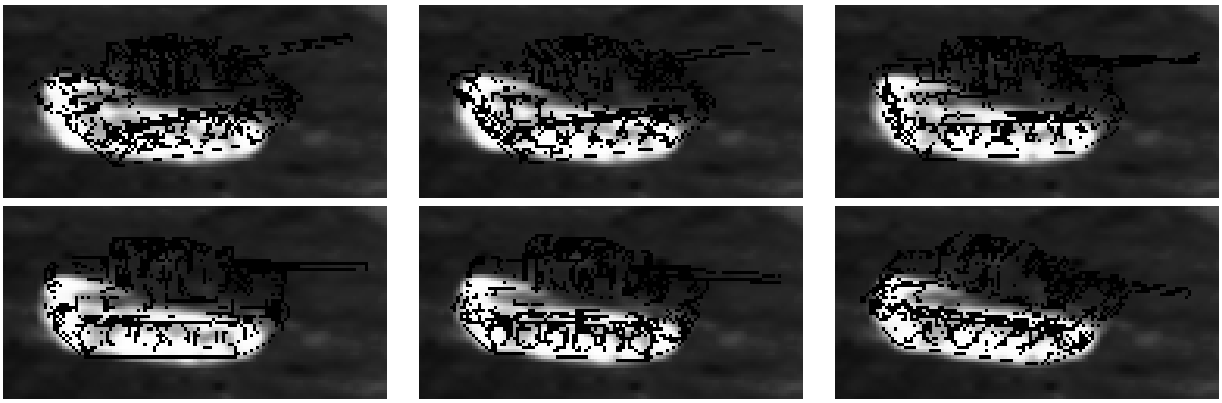

Figure 5: An example diffusion process refining the orientation of the M60 in the MICOM data.

Langevin diffusions must be discretized for computer implementation, and the discretized process only approximately maintains detailed balance. As suggested by J. Besag,[24] this can be readily remedied by performing an acceptance/rejection step after making the discretized Langevin diffusion; the resulting Metropolis-Hastings algorithm incorporates all the advantages of the Langevin diffusion approach, namely the analogy with the gradient search (natural for continuous variables) and completely parallel site updating. Thus diffusion techniques fit nicely within the Metropolis-Hastings umbrella.

If derivatives are difficult to compute, Metropolis-Hasting proposals can be made using a density centered around the old value, such as a Gaussian. The acceptance step ensures that we still achieve the target distribution. Most intriguingly, Gelfand and Mitter[25] have shown that certain continuous-time interpolations of some Metropolis sampling algorithms, such as the Gaussian proposals mentioned here, weakly converge to Langevin diffusions with the same stationary distribution. Thus, even if the gradient is not explicitly employed, it implicitly guides the inference in a limiting sense.

4.5 An example with multiple targets

We now present an example illustrating the detection and estimation of objects using saccadic shapes and blobs. For illustration, we will present the elements executed in a sequential sense; a complete algorithm would move between the various representations and include the 3-D models. Some further challenges in formulating such an algorithm become evident, as will be discussed in the final section.

The first two panels of Fig. 6 illustrate siccadic detection loglikelihoods. The next panel shows the detected square, which is deformed to match the overall tank shape in the fourth panel. The final two panels show the evolution of a blob starting from the rectangle.

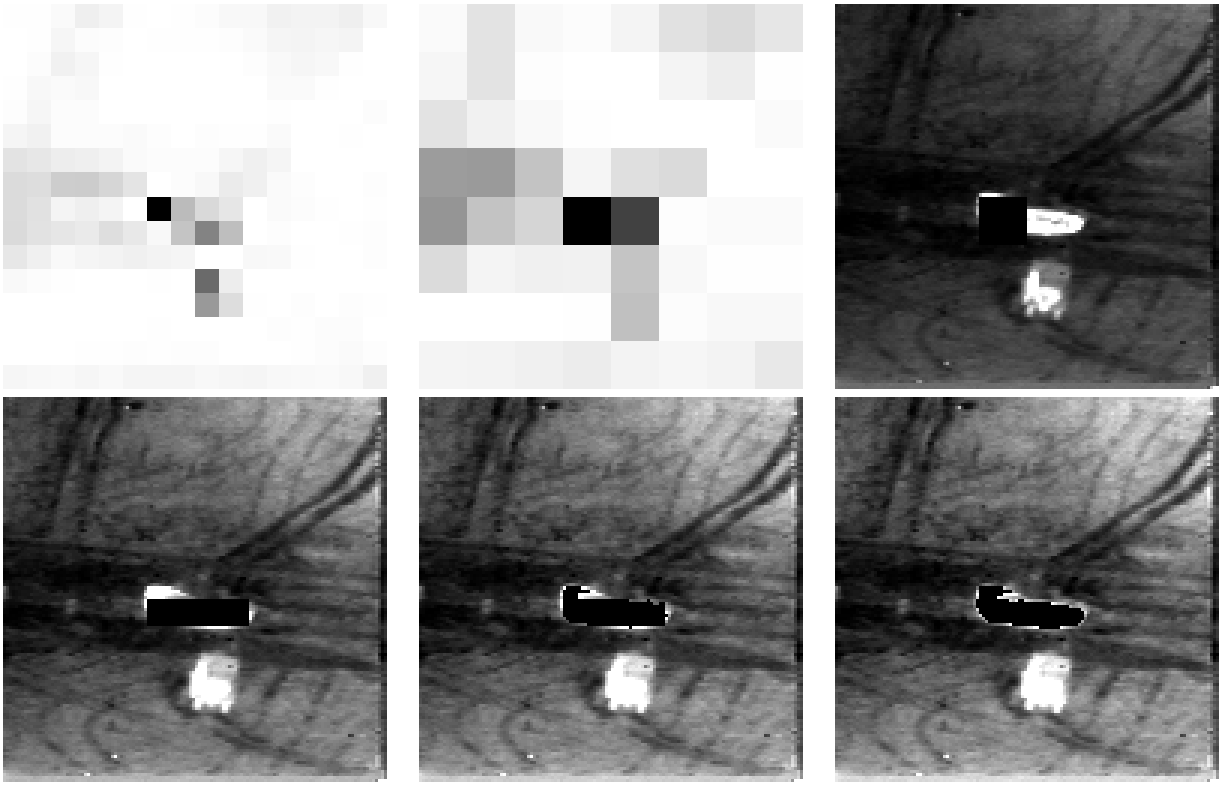

Figure 6: Complete detection, location, and generic shape estimation of a tank.

A analogous sequence of panels shown in Fig. 7 illustrate the detection and estimation of the truck. Since the tank has been found, it is removed from the likelihood calculation.

5 DIRECTIONS OF CURRENT AND FUTURE WORK

This paper has presented work in progress. The following paths demand exploration and are being travelled.

5.1 The reinterpretation of parameters

The above examples presented detection, location, and detailed estimation sequentially and separately. Our goal is an algorithm which performs these tasks simultaneously and in a unified manner. The task is complicated by the differences between the various representations. Ideally, we would like an algorithm which can propose changing a saccadic shape into a blob or a detailed target and vice-versa, and move between blob and 3-D target representations in the case of targets with an ambigious appearance. The algorithm must not only be able to make such moves, but for efficient inference, it must propose them in an intelligent way. The best way to approach this is not obvious at present. In the context of the Gibbs sampler, a cousin of the Metropolis-Hastings sampler, P. Green commented[26] that "The Gibbs sampler hardly even makes sense when x has a length that is not fixed, and *elements which need not have a fixed interpretation across all models.*" [Italics ours.] It is precisely this notion of "interpretation across models" which we must make precise.

5.2 The need for complexity constraints

If this algorithm is run without any explicit constraints on the number of targets permitted, the algorithm will pack the entire extent of the data with targets. This phenomenon was not observed in our previous work

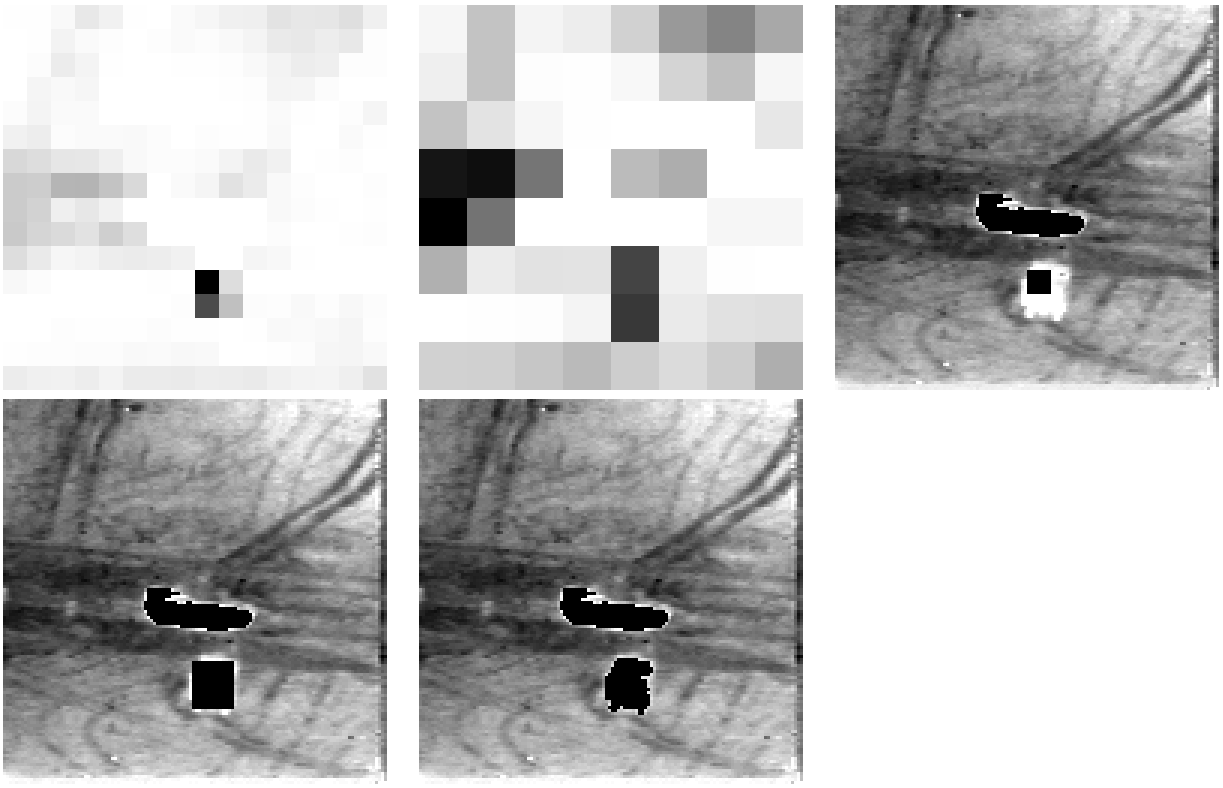

Figure 7: Detection, location, and generic shape estimation of the truck, following the estimation of the tank.

since the background and target intensities were fixed. The algorithm had nothing to gain by birthing a target over the background. Here targets have variable intensity, and the intensity, especially for the flexible models, is allowed to be the same or very close to that of the background. This is illustrated in Fig. 8, which exhibits the detection of the somewhat brighter sky which follows detection of the ship, and Fig. 9, which illustrates detection of the road to the left of the tank, which is slightly darker than the background, following estimation of the two vehicles.

At first, the algorithm will find targets with intensities that are quite different than the background, since these will yield the largest gains in likelihood. But even after it finds all the true targets (and perhaps a few other interesting features), the algorithm may continue to birth object after object, estimating intensities which are about the same as the main background, yielding only slight increases in loglikelihood, eventually filling in the background with extraneous objects, giving the overly detailed fit which "greedy" maximum-likelihood methods eagerly provide.

Such pathological behavior can be observed in many model order estimation problems, such as estimation of auditory-nerve discharge rates.[27] The solution is to place constraints on the model order, for which minimum description length[28–30] criterial are well suited.

6 ACKNOWLEDGEMENTS

This work was supported by ARO DAAH04-95-1-0494, ONR N00014-92-J1418, ONR/AASERT N00014-94-1-1135, and ARO/AASERT DAAH04-94-G-0209.

7 REFERENCES

[1] U. Grenander. *Elements of Pattern Theory.* Johns Hopkins Univ. Press, 1996.

Figure 8: Additional detection following the estimation of the shape of the ship. The algorithm finds the sky, which is of greater intensity than the rest of the background.

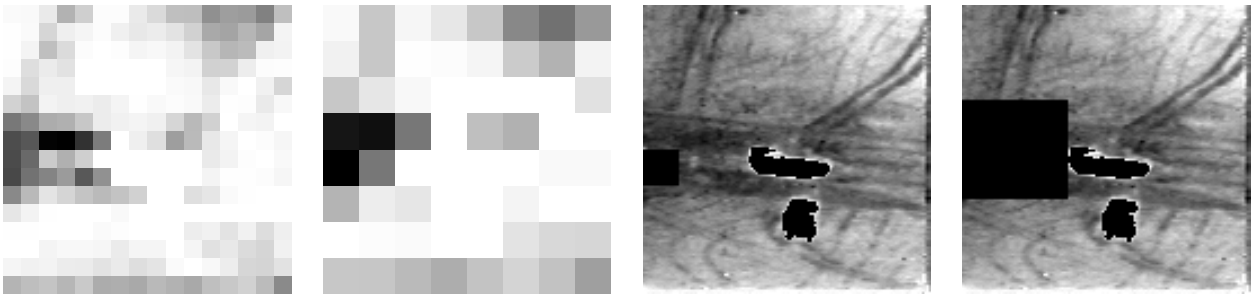

Figure 9: After finding the tank and the truck, the algorithm next finds the road, which is of less intensity than the rest of the background.

[2] U. Grenander. *General Pattern Theory.* Oxford Univ. Press, 1994.

[3] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society B*, 56(3):549–603, 1994.

[4] M.I. Miller, U. Grenander, J. A. O'Sullivan, and D.L. Snyder. Automatic target recognition organized via jump-diffusion algorithms. *IEEE Transactions on Image Processing*, 6(1):1–17, January 1997.

[5] M.I. Miller, G.E. Christensen, Y. Amit, and U. Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Science*, 90(24):11944–48, December 1993.

[6] G. E. Christensen, R. D. Rabbitt, and M.I. Miller. 3D brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.

[7] K. Mark, M.I. Miller, and U. Grenander. Constrained stochastic language models. In S.L. Levinson and L. Shepp, editors, *Image Models (and their Speech Model Cousins)*, pages 131–140. Springer-Verlag, 1996.

[8] A.D. Lanterman, M.I. Miller, and D.L. Snyder. General Metropolis-Hastings jump diffusions automatic target recognition in infrared scenes. *Optical Engineering*, 36(4), to appear April 1997.

[9] A.D. Lanterman. Jump-diffusion algorithms for the automated understanding of forward-looking infrared scenes. Master's thesis, Washington University, St. Louis, MO, May 1995.

[10] U. Grenander, Y. Chow, and D. Keenan. *HANDS: A Pattern Theoretic Study of Biological Shapes.* Springer-Verlag, New York, 1990.

[11] U. Grenander and D.M. Keenan. On the shape of plane images. *SIAM J. Appl. Math.*, 53(4):1072–1094, August 1993.

[12] M.I. Miller, S. Joshi, D. R. Maffitt, J. G. McNally, and U. Grenander. Membranes, mitochondria, and amoebae: Shape models. In K.V. Mardia, editor, *Advances in Applied Statistics: Statistics and Images*, volume 2. Carfax Publishing Co., Abingdon, Oxfordshire, UK, 1994.

[13] M.I. Miller, A. Srivastava, and U. Grenander. Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43(11):2678–2690, November 1995.

[14] D. L. Snyder and M. I. Miller. *Random Point Processes in Time and Space*. Springer-Verlag, 2nd edition, 1991.

[15] *Prism 3.2 User's Manual*. Keweenaw Research Center, Michigan Tech. Univ., Houghton, MI, 1995.

[16] M. Cooper, A.D. Lanterman, S. Joshi, and M.I. Miller. Representing the variation of thermodynamic state via principle components analysis. In *Proceedings of the Third Workshop on Conventional Weapon ATR*, November 1996.

[17] U. Grenander, M.I. Miller, and A. Srivastava. Hilbert-schmidt lower bounds for estimators on matrix lie groups. *Submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1997.

[18] J. Besag and P. J. Green. Spatial statistics and bayesian computation. *J. Royal Statistical Society B*, 55(1):25–38, 1993.

[19] A.F.M. Smith and G.O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *J. Royal Statistical Society B*, 55(1):3–37, 1993.

[20] W.R. Gilks, D.G. Clayton, D.J. Spiegelhalter, N.G. Best, A.J. McNeil, L.D. Sharples, and A.J. Kirby. Modelling complexity: Applications of gibbs sampling in medicine. *J. Royal Statistical Society B*, 55(1):39–52, 1993.

[21] Discussion on the meeting on the gibbs sampler and other markov chain monte carlo methods. *J. Royal Statistical Society B*, 55(1):53–101, 1993.

[22] D. Spiegelhalter W. Gilks, S. Richardson, editor. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[23] U. Grenander. A mathematical foundation of siccadic detection. Under development.

[24] J. Besag. Contribution to the discussion of paper by Grenander and Miller. *Journal of the Royal Statistical Society B*, 56(3):591–592, 1994.

[25] S.B. Gelfand and S.K. Mitter. Weak convergence of markov chain sampling methods and annealing algorithms to diffusions. *Journal of Optimization Theory and Applications*, 68(3):483–498, March 1991.

[26] P.J. Green. Reversible jump Markov chain Monte Carlo computation and bayesian model determination. *Biometrika*, 82(4):771–732, December 1995.

[27] K. E. Mark and M. I. Miller. Bayesian model selection and minimum description length estimation of auditory-nerve discharge rates. *Journal of the Acoustical Society of America*, 91, No. 2:989–1002, February 1992.

[28] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11:416–431, 1983.

[29] G. Schwartz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[30] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14, no.3:1080–1100, 1986.