Workshop on Detection and Classification of Difficult Targets, Redstone Arsenal, Alabama, 1998.

Clutter Modeling and Performance Analysis in Automatic Target Recognition [1]

Song Chun Zhu
Dept. of Computer Sci.
Stanford University
Stanford, CA 94305
szhu@cs.stanford.edu

Aaron Lanterman
Dept. of Elec. Eng.
Washington University
St.Louis, MO 63130
adl@cis.wustl.edu

Michael Miller
Dept. of Elec. Eng.
Washington University
St.Louis, MO 63130
mim@cis.wustl.edu

Abstract

The past decade has witnessed rapid development in accurate modeling of 3D targets and multiple sensor fusion in automatic target recognition (ATR), however, the scientific study for quantifying non-target objects in a cluttered scene has made very limited progress, due to its enormous difficulties. In this paper, we study two important themes in ATR: I) clutter modeling – how can we build generic and low-dimensional probabilistic models for cluttered scenes, and how can we automatically learn such models from observed images? II) performance analysis – how can we quantify the effects of clutter on the performance of the ATR algorithms, and how much do the learned clutter models improve ATR performance? We answer the above questions by combing two important trends which have emerged in the past few years. The first is the minimax entropy learning theory, proposed by Zhu, Wu and Mumford[12]. According to this theory, cluttered scenes are defined on random fields with features characterized by generic filters at various frequencies and orientations. Then a probability model is learned so that it reproduces the statistics of the most important features. The second is the theory for performance bounds (such as the Hilbert-Schmidt bound), proposed by Grenander, Miller and Srivastava[3]. This theory quantifies the accuracy of target estimation on Lie groups. We illustrate our theories by three groups of experiments: 1) clutter modeling, synthesis, removal; 2) effects of clutter on the performance bounds; 3) using a learned clutter model for ATR in contrast to an additive Gaussian noise model.

Key words: clutter modeling, ATR, Markov random field models, ATR performance analysis, minimax entropy principle, Hilbert-Schmidt bounds.

[1]The authors are members of the Center for Image Sciences supported by a grant DAAH-04-95-1-0494 from the ARO.

1 Introduction and motivations

In both military and civilian applications of computer vision, one often faces the task of identifying targets in a cluttered environment, for example, detecting tanks or trucks in a battle field, and looking for faces or chairs in a living room. In such tasks, one may have rigid 3D models or deformable templates for the targets which are projected on the image plane after rigid transformations, whereas it is practically infeasible to model accurately all objects in the environment. Such non-target objects, known as "clutter", cause false alarms in the background, or obscure the targets and thus deteriorate the ATR performance.

In a recent ATR survey paper by the Army laboratories [6], Ratches et. al. concluded that "the essence of understanding performance of ATR implementation is ... the mathematical descriptions of targets and clutter that enable understanding the fundamentals of the image content." In the past decades, the major progress in ATR performance have resulted from accurate target modeling and multiple sensor fusion. A major barrier for increasing ATR performance is clutter modeling as stated in [6]: "Probably the key problem associated with understanding and predicting ATR performance is the scientific understanding and quantification of clutter as it affects detection."

The early attempts in quantifying clutter have achieved very little success due to the insufficiency of these models [8, 9]. In figure 1, we display some typical cluttered FLIR (Forward Looking Infrared) scenes. It is clear that the background objects have rich structures which resemble the features of the targets, and which could not modeled by simple Gaussian or Poisson models.

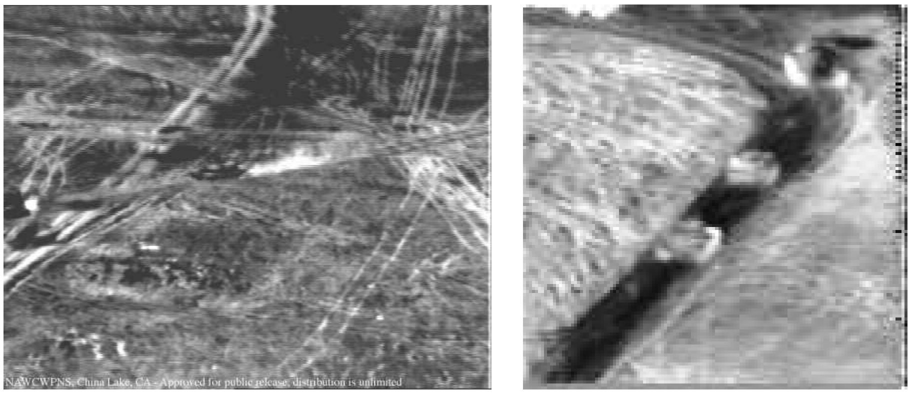


Figure 1 Cluttered scenes from FLIR sensors. a). FLIR scene from U.S. Naval Air Warfare Center, b). FLIR scene scene from U.S. Aviation and Missile Command.

In this paper, we address two important themes in ATR. I) clutter modeling – how can we build generic and low-dimensional probabilistic models for cluttered scenes, and how can we automatically learn such models from observed images? II) performance analysis – how can we quantify the effects of clutter on the performance of ATR algorithms, and how much do the learned clutter models improve ATR performance? We answer the above questions by combing two important trends that have emerged in the past few years. The first is the minimax entropy learning theory, proposed by Zhu, Wu and Mumford [12]. The second is the theory for performance bounds (such as the Hilbert-Schmidt bound), proposed by Grenander, Miller and Srivastava [3].

1. The theory for clutter modeling. We define the ensemble of images of cluttered scenes in a given application on the random fields, for example, the sensor images of a specific battle field. The structure and information content of the image ensemble is captured by an underlying probability distribution $f(\mathbf{I})$. Then given a set of sample images, our objective is to learn the clutter model $p(\mathbf{I})$ to approximate $f(\mathbf{I})$. The learning process is governed by the minimax entropy principle explained in the next section. The learned clutter models are Gibbs forms defined on the random fields, and the structure of the clutter is characterized by generic filters at various frequencies and orientations. The learned model $p(\mathbf{I})$ is constructed to reproduce the statistics extracted from the observed clutter images, while it has the maximum entropy. Then the most important clutter features are selected so that the Kullback-Leibler divergence $D(f||p)$ decreases as fast as possible. Thus $p(\mathbf{I})$ is a projection of $f(\mathbf{I})$ in a low dimensional space in analogy to Fourier transform theory. We will demonstrate experiments on the accuracy of the models by Markov chain Monte Carlo for both 1D and 2D signals.

2. Performance analysis. We analyze the effects of clutter modeling on ATR performance. This consists of two groups of experiments. One analyzes how the ATR performance measured by the Hilbert-Schmidt bound in estimating the target poses decreases with complex clutter. The other demonstrates that how a correct clutter model reduces false alarms and improves the accuracy of the target estimation.

The paper is organized as follows. In section (2), we discuss how to model clutter by the minimax entropy principle. Then we demonstrate experiments on clutter modeling and clutter removal in section (3). Then section (4) presents the basic theories of Bayes estimation and the Hilbert-Schmidt bounds, which is followed by experiments on how clutter effects performance in section (5). Section (6) studies how a learned clutter model improve ATR performance. We conclude the paper by a discussion in section (7).

2 Clutter modeling

In this section, we present the minimax entropy principle for learning models of clutter from a set of observed sample images, beginning with the basic issues in clutter modeling and the rationale for our statistical theory.

2.1 Theory of clutter modeling – an overview

In military applications, background objects are often generated by physical and chemical processes of stochastic nature, for example, desert, rock, snow, trees, river, and so on. It is reasonable for us to assume that the stochastic process has a stationary distribution, at least for a given season and geology location. Thus the imagery $\mathbf{I}$ captured by a sensor should follow an underlying distribution $f(\mathbf{I})$, which contains all the regularities and information about the clutter. Then the objective of modeling clutter is to estimate $f(\mathbf{I})$ by a model $p(\mathbf{I})$, and $p(\mathbf{I})$ should be learned from a set of independent samples $S^{\mathrm{obs}} = \{I_n^{\mathrm{obs}}, n = 1, 2, ..., M\}$ from $f(\mathbf{I})$.

Another way of viewing this modeling problem is to consider our computer as an encoding device, and the best model should produces code that has the minimal description length for the data ensemble[7]. Thus ideally the true model should account for both the physical and chemical stochastic processes, for example one could simulate the natural stochastic processes by computer graphics, which, however, is computationally infeasible. In practice it is desirable to pursue generic and low dimensional models $p(\mathbf{I})$, while $p(\mathbf{I})$ should be as close to $f(\mathbf{I})$ as possible. But do such generic, low dimensional models exist without simulating the physical and chemical processes? Our answer is yes.

Before we discuss our learning theory in detail, we demonstrate one simple example in one-dimension.

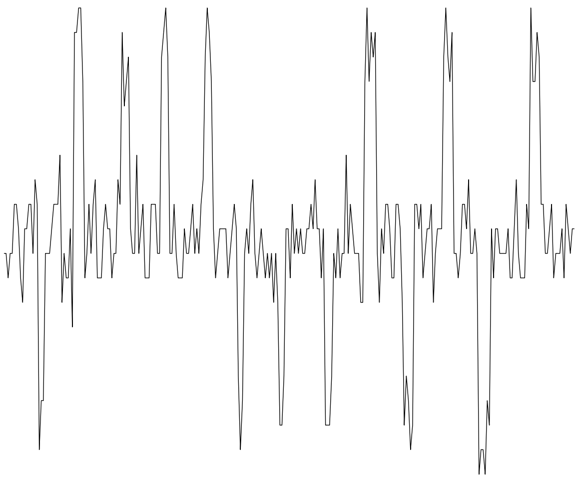


Figure 2 A fragment of one-dimensional stochastic signal generated by a program.

Figure 2 displays a fragment of one-dimensional stochastic signal, consists of a number of pulses, generated by a simple C-program. We simply manipulate four parameters–the height, width, interval, and sign – of the pulses at random, and add Gaussian noise drawn from $N(0, \sigma^2)$. The signal is of stochastic nature. Now our questions are: given a set of observed signals, according to what probability distribution are they generated? According to the theory of Komogorov complexity[1], The C code itself provides a lower bound for the Komogorov complexity.

We argue that one may not have to recover the original stochastic process; instead, we may characterize the structures in the signal by a generic set of feature extractors, for example, wavelet transforms. Figure 3 displays a set of wavelets of the following form

$$F(x;T) = ae^{-\frac{x^2}{2T^2}} cos(\frac{2\pi}{T}x),$$

where T is the period of the cosine wave.

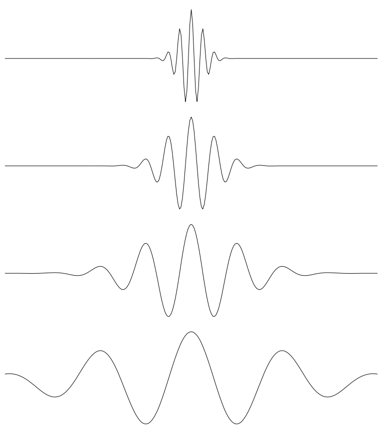


Figure 3 One-dimensional wavelet transforms at various frequencies (scales).

For computational ease, suppose that we choose a small number of wavelets $\{F^{(\alpha)}, \alpha = 1, 2, ...K\}$ ($K = 3$ in this example), and we convolve the wavelets with the observed signals in figure 2. Then we compute the histograms of the of the responses for the chosen wavelets respectively. We then construct a maximum entropy model $p(\mathbf{I})$,

$$p(\mathbf{I}) = \frac{1}{Z} \exp\{-\sum_{\alpha=1}^{K} \sum_{x} \lambda^{(\alpha)}(F^{(\alpha)} * \mathbf{I}(x))\}$$

where $\mathbf{I}$ is the one-dimensional signal, and $\lambda^{(\alpha)}(z)$ is the Lagrange multipliers (one dimensional function in this case), and it is learned so that $p(\mathbf{I})$ reproduces the observed histograms. Figure 4 plots the learned potential functions $\lambda^{(\alpha)}(z), \alpha = 1, 2, 3$.

The learning process is discussed in the next section. To verify the goodness of the learned $p(\mathbf{I})$, we simulated a Markov chain which has $p(\mathbf{I})$ as its stationary

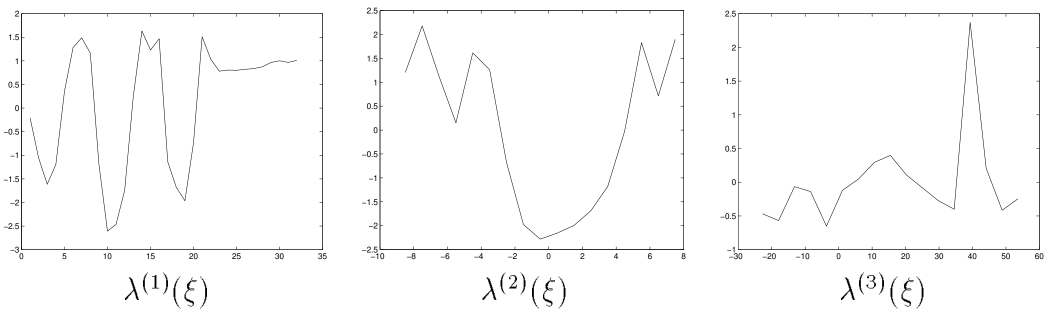


Figure 4 Learned potential functions $\lambda^{(\alpha)}(z)$, $\alpha = 1, 2, 3.$, and the horizontal axis is the wavelet responses $F^{(\alpha)} * \mathbf{I}$.

distribution. When it converges, it generates random (stochastic) samples from $p(\mathbf{I})$. Figure 2.1 displays a fragment of the stochastic signal.

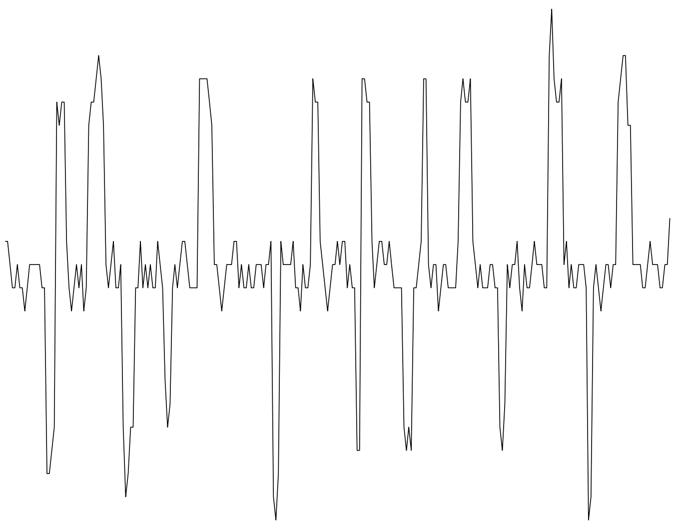


Figure 5 Fragment of the stochastic signal sampled from the model $p(\mathbf{I})$.

It is interesting to see that the stochastic signal generated from $p(\mathbf{I})$ resembles the one generated by the c-code. The similarity can be measured quantitatively through the statistics extracted from the two groups of signals by the wavelets. If any significant statistical divergence is found between the synthesized signals and the observed ones, we can always adapt $p(\mathbf{I})$ by including more wavelets.

The learned model $p(\mathbf{I})$ uses 66 parameters[2] for the three potential functions

[2]This number can be smaller if we try to fit the $\lambda^{(\alpha)}(z)$, $\alpha = 1, 2, 3$ by some parametric functions.

plotted in figure 4. It is less efficient than the original c-code, which uses 5 parameters only.[3] Nevertheless, $p(\mathbf{I})$ is a close approximation to $f(\mathbf{I})$, and most important, it has a generic and closed form.

In the next section, we discuss how this method is applied to modeling clutter images and study the details of the minimax entropy learning theory.

2.2 The minimax entropy learning theory

In the past few years, a new theoretical framework has emerged for image modeling and learning based on a minimax entropy principle proposed by Zhu, Wu and Mumford [12, 11]. This learning scheme has been applied to modeling a variety of textures and generic natural images with enormous successes. Figure 6 shows the diagram of the minimax entropy learning framework. It consists of two phases. The minimum entropy phase selects the most important features which best characterize the images or clutter signals. Then the maximum entropy phase construct the model $p(\mathbf{I})$ which reproduces the statistics extracted by the chosen features. We briefly describe the two learning phases below. Readers are referred to a previous paper for detailed account[12].

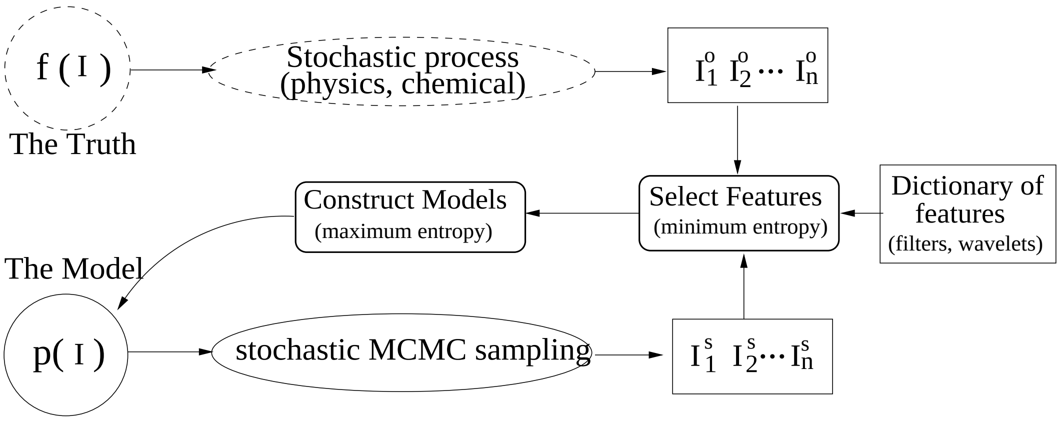


Figure 6 The diagram of clutter modeling by a minimax entropy principle.

The maximum entropy phase. Chosen a set of features $S = \{F^{(\alpha)}, \alpha = 1, 2, ..., K\}$, they are linear filters in our examples, in general this could be any non-linear functions of the clutter images. Suppose we observe M training images from $f(\mathbf{I})$ $S^{\mathrm{obs}} = \{\mathbf{I}_n^o, n = 1, 2, ..., M\}$. Then we compute statistics, for example, the histograms of the filter responses:

$$\mu_{\mathrm{obs}}^{(\alpha)}(z) = \frac{1}{M} \sum_{n=1}^{M} H^{(\alpha)}(z, \mathbf{I}_n^o)$$

[3]Both the c-code and the Monte Carlo method employ the same uniform random number generator.

$$= \frac{1}{M} \sum_{n=1}^{M} \sum_{(x,y)} \delta(z - F^{(\alpha)} * \mathbf{I}_n^o(x,y)), \quad \forall z \in R,$$

where $H^{(\alpha)}(z, \mathbf{I})$ is the histogram of the filter responses of $F^{(\alpha)}$ for image $\mathbf{I}$.

It is easy to see that $\mu_{\text{obs}}^{(\alpha)}(z)$ are estimates of the one-dimensional marginal distributions of $f(\mathbf{I})$. Then a model $p(\mathbf{I})$ should reproduce the observed statistics, i.e.

$$\int p(\mathbf{I}) \sum_{(x,y)} \delta(z - F^{(\alpha)} * \mathbf{I}(x,y)) d\mathbf{I} = \mu_{\text{obs}}^{(\alpha)}(z), \quad \alpha = 1, 2, ..., K, z \in R.$$

Let Ω_p be the set of all $p(\mathbf{I})$ that satisfies the above constraints, the maximum entropy principle simply chooses

$$p(\mathbf{I}) = \arg \max_{p \in \Omega_p} - \int p(\mathbf{I}) \log p(\mathbf{I}) d\mathbf{I}.$$

Solving the above constrained optimization problem by the Lagrange multipliers, we have $p(\mathbf{I})$ of the form,

$$p(\mathbf{I}; \Lambda, S) = \frac{1}{Z} \exp\{- \sum_{\alpha=1}^{K} \sum_{(x,y)} \lambda^{(\alpha)}(F^{(\alpha)} * \mathbf{I}(x,y))\},$$

where $Z = Z(\Lambda)$ is the partition function, $\Lambda = (\lambda^{(1)}(), \lambda^{(2)}(), ..., \lambda^{(K)}())$ are the one-dimensional potential functions, whose values are solved from the constraint equations. An equivalent form of the model is

$$p(\mathbf{I}; \Lambda, S) = \frac{1}{Z} \exp\{- \sum_{\alpha=1}^{K} \int \lambda^{(\alpha)}(z) H^{(\alpha)}(z, \mathbf{I}) dz\}.$$

See [12] for details.

The minimum entropy phase. Given a learned model $p(\mathbf{I}; \Lambda, S)$, we can generate a set of images $S^{\text{syn}} = \{\mathbf{I}_n^s, n = 1, 2, ...M'\}$ by Markov chain Monte Carlo[10] for model verification. This synthesis step may not be necessary if we can learn the model and compute its marginal distributions in closed form.

The quality of a model $p(\mathbf{I}; \Lambda, S)$ can be measured by the Kullback-Leibler divergence

$$D(f||p) = \int f(\mathbf{I}) \log \frac{f(\mathbf{I})}{p(\mathbf{I})} d\mathbf{I}$$

As $\int f(\mathbf{I}) H^{(\alpha)}(z, \mathbf{I}) d\mathbf{I} = \int p(\mathbf{I}) H^{(\alpha)}(z, \mathbf{I}) d\mathbf{I}$ as stated by the constraint equations in case we have enough observations, it is found that

$$D(f||p) = \int f(\mathbf{I}) \log f(\mathbf{I}) d\mathbf{I} - \int p(\mathbf{I}) \log p(\mathbf{I}) d\mathbf{I} = entropy(p) - entropy(f).$$

Thus we want to choose the best set of filters that minimizes $D(f||p)$ or minimizes $entropy(p)$.

$$S^* = \arg\min entropy(p(\mathbf{I}; \Lambda, S))$$

Zhu et.al. further developed the framework for selecting features in a stepwise manner. At each step, the filter which minimizes the entropy the most is selected by comparing the difference between $\mu_{\text{obs}}^{(\alpha)}(z)$ and $\mu_{\text{syn}}^{(\alpha)}(z)$. The latter is computed from the set of synthesize images S^{syn}.

Intuitively, the difference between $\mu_{\text{obs}}^{(\alpha)}(z)$ and $\mu_{\text{syn}}^{(\alpha)}(z)$ tells how much $p(\mathbf{I})$ is apart from $f(\mathbf{I})$ in the dimension spanned by the feature $F^{(\alpha)}$. The bigger the difference, the more important the feature $F^{(\alpha)}$ is.

To summarize the learning theory in figure 6, we choose a set of generic features to capture the structures of the observed clutter images, and use the Markov chain Monte Carlo process to verify the model. The learned model can be considered as the projection of the underlying model $f(\mathbf{I})$ into probability distributions of Gibbs forms in a low dimensional space. For a given model $p(\mathbf{I})$, there are many stochastic processes which have $p(\mathbf{I})$ as their stationary probability distributions. We employ the general method – Markov chain Monte Carlo – instead of simulating the physical and chemical process occurring in nature. The simulation of cluttered scenes via computer graphics shall be addressed in section (4).

3 Experiment I: clutter modeling and removal

In this section, we demonstrate some experiments on modeling clutter images, and we also show an experiment on clutter removal by the learned model.

For two dimensional images, the generic features are chosen from a set of Gabor filters, whose window functions are

$$Gabor(x, y : \omega, \theta) = Rot(\theta)[e^{-\frac{\omega^2}{8\pi}(4u^2+v^2)} \cdot e^{i\omega u}],$$

which are sine or cosine waves at frequency ω whose amplitudes are modulated by elongated Gaussian functions, then rotated by angle θ. For example, figure 3 shows two Gabor functions. The filter set contains filters at various frequencies and orientations.

The minimax entropy learning procedure is illustrated in figure 3. Figure 3.a is an observed texture/clutter image. We start with $p(\mathbf{I})$ being a uniform distribution, i.e. $K = 0$, then we choose one filter at each step so that the entropy of $p(\mathbf{I})$ decreases the most. Figure 3.b is a sampled image from $p(\mathbf{I})$ with $K = 6$ filters respectively. Like the 1D signal experiment discussed in section 2.1, there is almost no perceptual difference between the observed one and the synthesized one, thus $p(\mathbf{I})$ is a good enough approximation to $f(\mathbf{I})$.

In another example, the clutter pattern has long range structures such as the tree branches in figure 9, and the images are inhomogeneous. A model $p(\mathbf{I})$ is learned

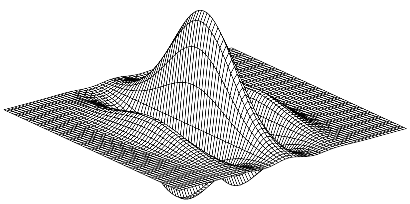


Figure 7 The window functions of two Gabor filters: the cosine component (left) and the sine component (right).

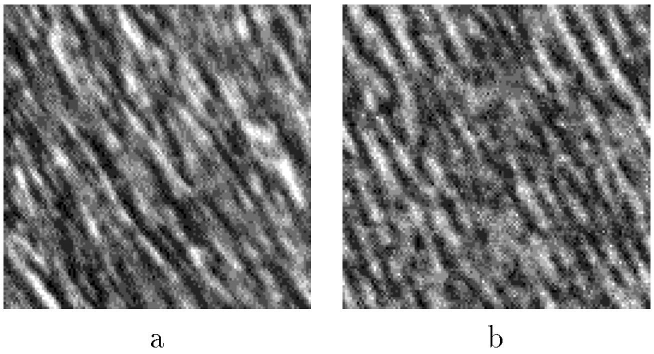


Figure 8 The modeling of a texture image . a). the ordinal observed texture/clutter image, b). the sampled image from the learned model $p(\mathbf{I}; \Lambda, S)$ with $K = 6$ filters.

from a small set of tree images, and a sample image from $p(\mathbf{I})$ is shown in figure 3. It could be more efficient if we design filters specifically tuned for the tree structures to capture the long branches and junctions of the tree branches.

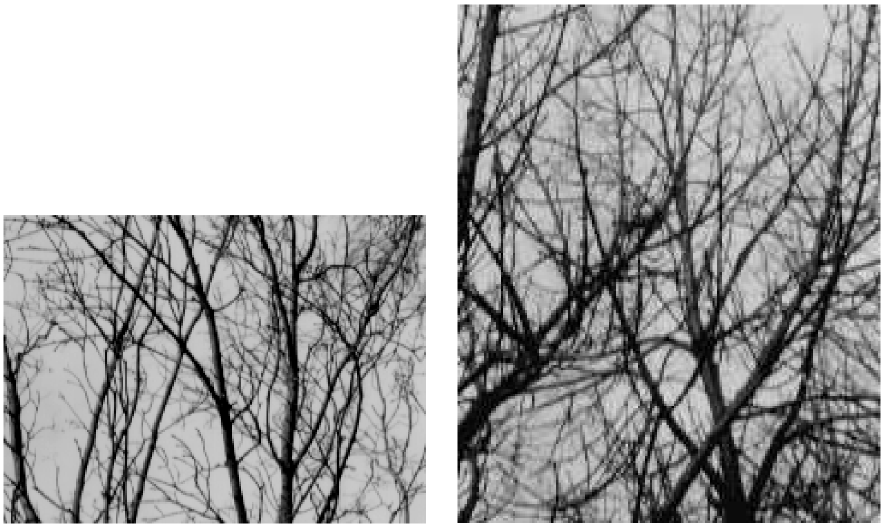


Figure 9 Two clutter images with inhomogeneous, long range structures.

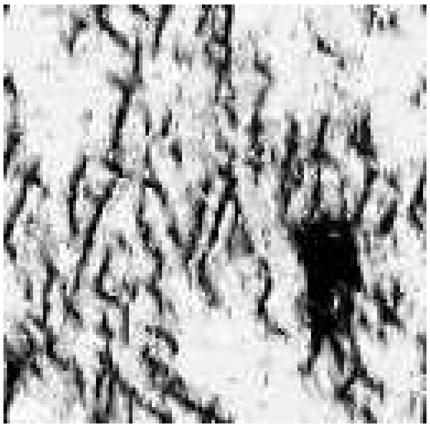


Figure 10 A sample image from the learned model of tree clutter.

To demonstrate how useful a good clutter model is, we demonstrate an experiment in figure 11. We learn two models, one $p_T(\mathbf{I}_T)$ for the tree clutter as above, and one $p_B(\mathbf{I}_B)$ for the city scenes (buildings). By assumption of an additive model

$$\mathbf{I}_{\text{obs}} = \mathbf{I}_T + \mathbf{I}_B$$

we can recover the building by computing the *maximum a posteriori* estimate in the Bayes framework,

$$\mathbf{I}_B^* = \arg\max_{\mathbf{I}_B} p(\mathbf{I}_{\rm obs}|\mathbf{I}_B)p_B(\mathbf{I}_B) = \arg\max_{\mathbf{I}_B} p_T(\mathbf{I}_{\rm obs} - \mathbf{I}_B)p_B(\mathbf{I}_B).$$

The additive model is a crude approximation to occlusion for computational simplicity. We shall consider occlusion explicitly in the next experiment.

Intuitively, the Gabor filters are efficient in detecting local features for trees and buildings, and thus the more accurate the model is, the better it can discriminate and remove the clutter.

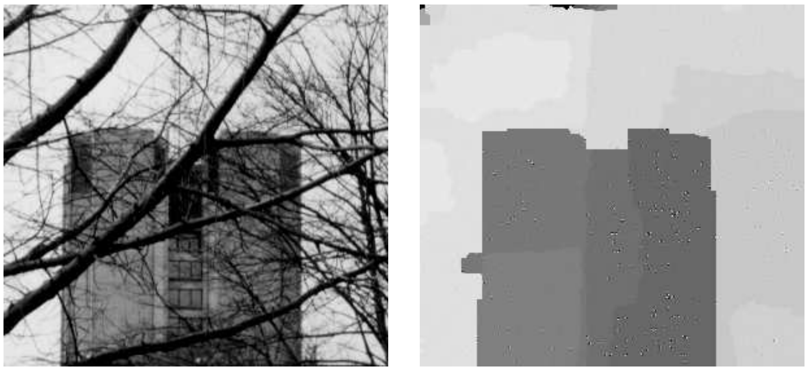


Figure 11 left: an input image with building cluttered with tree branches; right: a building image recovered from the clutter using the learned clutter model.

4 Performance analysis

In this section, we discuss theory of the performance bounds of target estimation and the effects of clutter on ATR.

With the rapid increase of computer power in the past decade, 3D model simulation and image rendering has been widely adopted in ATR and represents the major progress in improving ATR performance[6]. In the model-based method, a target, such as a truck, a tank, or a car, is constructed from a CAD representation, and this CAD model is specified by a vector of parameters $x \in \mathcal{X}$. For example, $\mathcal{X}$ includes the vertices and surface normals of a CAD model[3], and x may also include the orthogonal bases for characterizing the thermodynamic variabilities by a principal components analysis[2]. Then the 3D model is projected onto the image plane by weak-projective transformations, translation, rotation, and scaling, with hidden surfaces removed and surface texture preserved. The set of transformations $\mathcal{S} : R^n \to R^n$ are Lie groups. Complex models can be obtained by allowing target deformations.

As proposed by Grenander, Miller, Srivastava, and others[3, 5] in the pattern theoretic framework, the ideal images of a target form the orbit of the template under the group action,

$$\mathbf{I} = \{\mathbf{I}_{temp}(s \circ x), \forall s \in \mathcal{S}\}.$$

Thus the probability space for model inference is given by $\mathcal{S}$ for a given target, and the ideal image $\mathbf{I}_{temp}(s \circ x)$ is identifiable by its parametric representation s. For example s is a transformation matrix for the translation, rotation, and scaling of a target. A prior model $p(s)$ for s is often assigned in a given application.

A sensor image $\mathbf{I}_{\text{obs}}$ is generated from the ideal template image with clutter objects which may obscure part of the target. Thus the sensor image $\mathbf{I}_{\text{obs}}$ consists of two parts: pixels, labeled as ℓ_t, are generated from the target plus sensor noise, and other pixels, labeled as ℓ_c, are generated by the clutter process.

This image rendering process is expressed by the likelihood model $p^c(\mathbf{I}_{\text{obs}}|s)$ or $p^c(\mathbf{I}_{\text{obs}}|\mathbf{I}_{temp}(s \circ x))$. The sensor noise is often easy to model; for example, a Poisson process may be sufficient, while the clutter process could only be approximated. Thus, in ATR applications, we only have approximate likelihood models $p^a(\mathbf{I}_{\text{obs}}|s)$.

In the Bayes framework, a target is estimated by either the *maximum a posteriori (MAP)* estimate,

$$\hat{s}^* = \arg\max_{s \in \mathcal{S}} p^a(\mathbf{I}_{\text{obs}}|s)p(s)$$

or the *minimum mean squared error (MMSE)* estimate

$$\hat{s}^* = \arg\min_{s \in \mathcal{S}} \int d(s, s^{'})p^a(\mathbf{I}_{\text{obs}}|s^{'})p(s^{'})ds^{'}.$$

where $d(s, s^{'})$ is a distance measure, e.g. L_1 norm in the Euclidean space or the Hilbert-Schmidt distance on Lie groups [3].

The performance of the ATR algorithm is then measured by

$$B(s) = \int d(s, \hat{s}^*)p^c(\mathbf{I}_{\text{obs}}|s)d\mathbf{I}_{\text{obs}}, \quad s \in \mathcal{S},$$

i.e., given a true target s (location, orientation, scale), $B(s)$ is the average error of the estimate $\hat{s}^*$. In practice, $p^c(\mathbf{I}_{\text{obs}}|s)$ is simulated by computers graphics which often generate realistic battle scenes – the 3D models of clutter objects and the natural stochastic processes are simulated this time! Samples from $p^c(\mathbf{I}_{\text{obs}}|s)$ generated this way should be as close to reality as possible.

The performance bound will largely depend on the accuracy of the clutter model $p^a(\mathbf{I}_{\text{obs}}|s^{'})$, i.e. how close it models the stochastic process of clutter and how well it accounts for the obscuring effects. In general, the closer the model $p^a(\mathbf{I}_{\text{obs}}|s)$ is to $p^c(\mathbf{I}_{\text{obs}}|s)$, then the better the performance bound $B(s)$ is. However, how $B(s)$ depends on the distance between $p^a(\mathbf{I}_{\text{obs}}|s)$ and $p^c(\mathbf{I}_{\text{obs}}|s)$ has yet to be make explicit.

To summarize, we show the overall diagram for ATR performance analysis in figure 12.

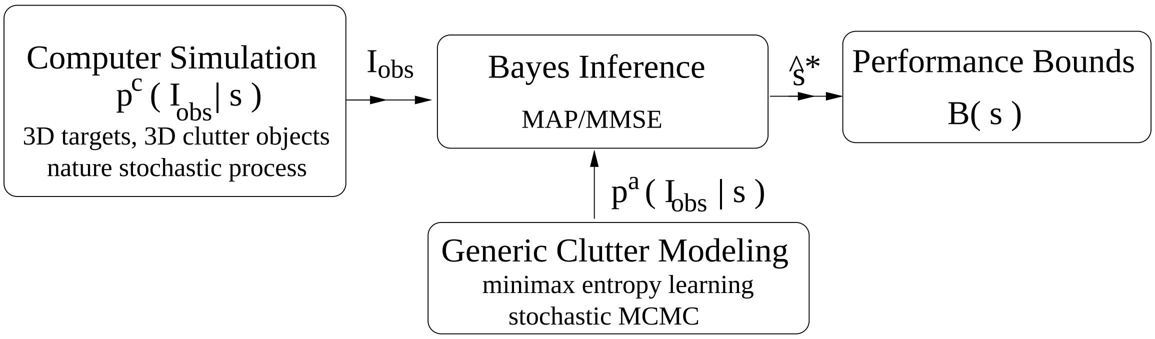


Figure 12 The overall diagram of ATR performance analysis

5 Experiment II: the clutter effects on ATR performance

In this section, we demonstrate the effects of clutter on ATR performance by two groups of experiments. In the experiments, the target is a tank modeled by a template, whose position (x, y) is assumed known. The goal is to estimate the orientation. The computer model $p^a(\mathbf{I}_{\rm obs}|s)$ for clutter is simply a Poisson distribution where the intensities of pixels are assumed to be independent. occlusion is not manipulated in $p^a(\mathbf{I}_{\rm obs}|s)$.

Block clutter. In our first group of experiments, small cluttering blocks were randomly placed over the scene. Two parameters were varied: the total number of cluttered pixels, and the size of the blocks. The leftmost column of figure 13 shows examples of block clutter with 540 total cluttered pixels; the second column column, 1080 cluttered pixels. From top to bottom, blocks of increasing size are employed. This permits exploration of the effects that the correlation of clutter has on performance.

Two variations of the experiment were explored: one in which the blocks have the approximate intensity of the tank (as shown in the left two columns of figure 13), and another in which they have the intensity of the background (as shown in the right two columns of figure 13). The former corresponds to "extending clutter" which appears to extend the shape of a target; the latter corresponds to "obscuring clutter" concealing parts of the target. The background intensity is assumed to be known.

The performance bound $B(s)$ for the two variations of the experiment are shown in figure 14. Two conclusions are drawn from figure 14: first, as the number of cluttered pixels increases, performance worsens; second, keeping the total number of cluttered pixels fixed, performance degrades as the block size increases. In essence, the low-correlated clutter has comparatively little effect on the model-based algorithm. The algorithm becomes more confused when the clutterers become sub-

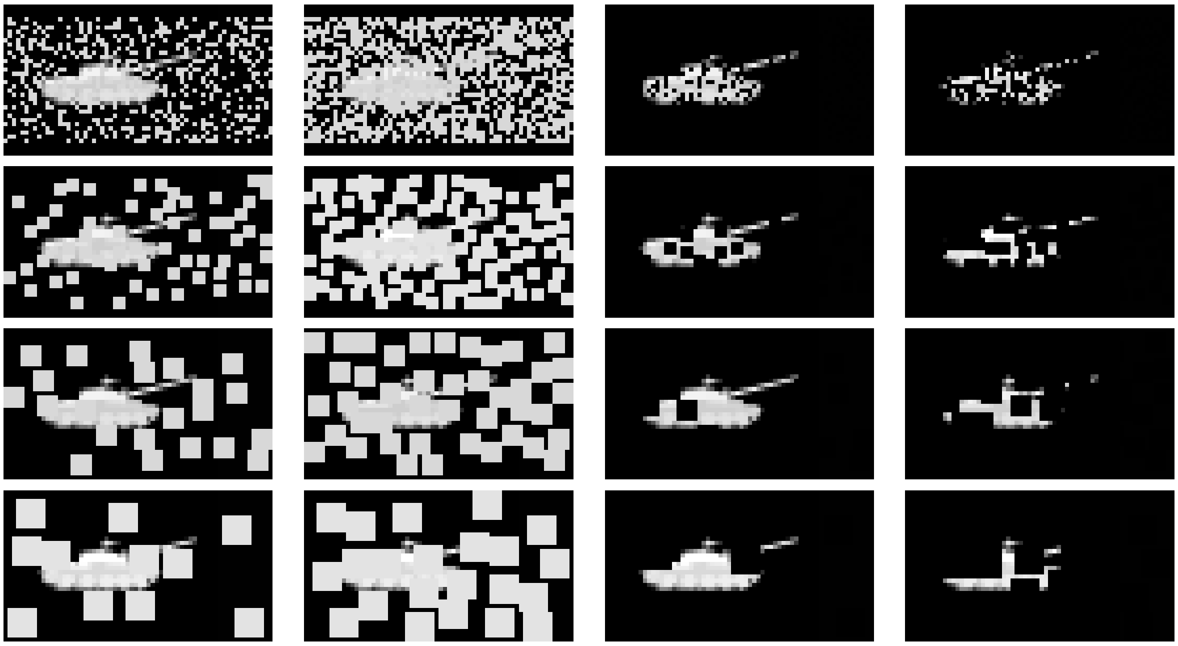


Figure 13 Examples of the block clutter. In the left two columns, the clutter has the intensity of the foreground; in the right two columns, the intensity of the background.

stantial enough in size to add significant features or to remove significant features from the target.

Tree clutter. In this experiment, 2-D silhouettes of trees were randomly superimposed into the scene. The trees silhouettes were formed by thresholding imagery provided by Theresa Kipp of the Army Research Laboratory. Unlike the proceeding experiment, they were allowed to overlap. Two parameters were varied: the shape of the tree, and the number of trees. The tree models used are shown in the upper row of figure 15. As in the block experiment, two variations were explored: one in which the trees had the intensity of the foreground, and another had the intensity of the background. Some sampled images from $p^c(\mathbf{I}_{\text{obs}}|s)$ are shown in the lower row of figure 15, which have the intensity of the foreground.

The results for the two variations of the experiment are shown in figure 16. Similar phenomena are observed as in the block experiments. Performance degrades as the number of trees increases. Notice that in both variations, Tree #2, which has a bushy, solid, concentrated appearance, induces the worst performance out of the three tree models.

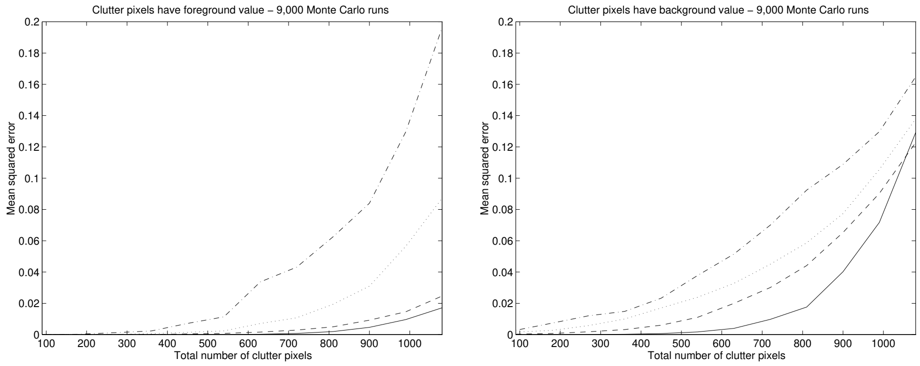


Figure 14 performance curves for block clutter having foreground (left) and background (right) appearances. From bottom to top, the lines correspond to blocks of size 1×1, 2×2, 3×3, and 4×4.

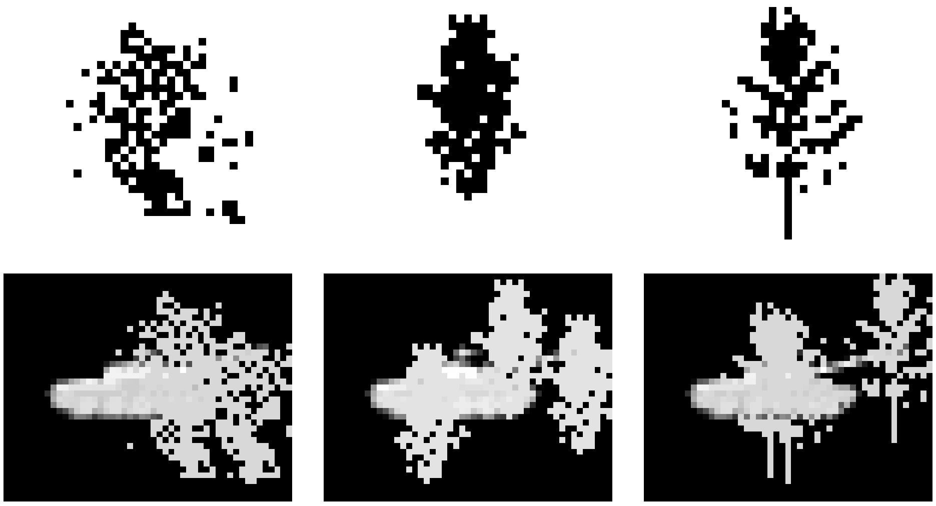


Figure 15 Left column shows tree models. Right two columns show examples of data from the tree clutter experiment.

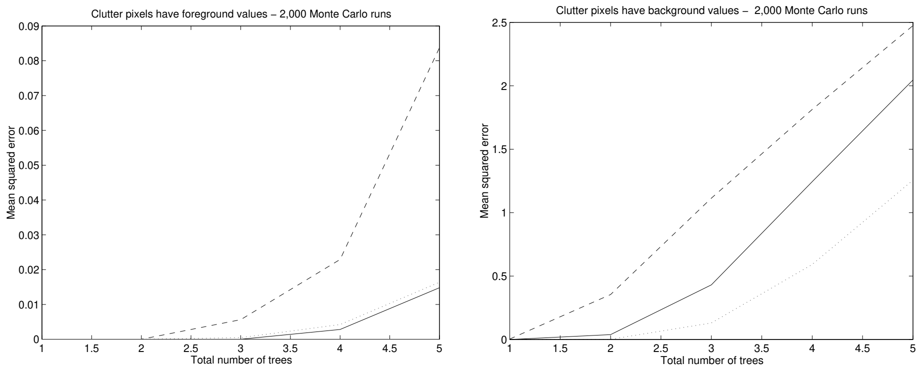


Figure 16 performance curves for tree clutter having foreground (left) and background (right) appearances. The solid, dashed, and dotted lines correspond to Trees 1, 2, and 3, respectively.

6 Experiments III: applying the learnt models

In this section, we experiment with the clutter model learned by the minimax entropy theory, and demonstrate how it improves the accuracy of detection. We take the one-dimensional signals studied in section 2.1 as an example.

Figure 17 shows how our one dimensional sensor signals are generated from $p^c(\mathbf{I}_{\text{obs}}|s)$: simply overlap a constant block of length L with a clutter signal at point X. The clutter signal is rendered by the c-code as discussed in section 2.1, and the impulses conceal part of the target at random place. A typical example of $\mathbf{I}_{\text{obs}}$ is shown in figure 18, with $X = 200$ and $L = 100$, and the signal length is 512 pixels.

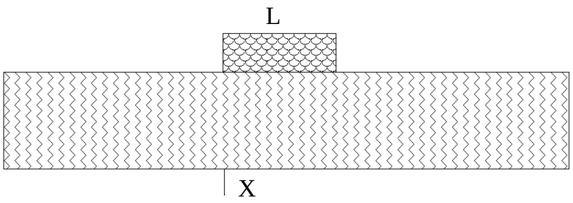


Figure 17 1D example: a target–rectangle of length L is hidden in a clutter signal.

Then we compute (X, L) by the maximum a posteriori estimate,

$$(X^*, L^*) = \arg\max_{X,L} p^a(X, L|\mathbf{I}_{\text{obs}}) = \arg\max_{X,L} p^a(\mathbf{I}_{obs}|X, L)p(X, L)$$

where the prior $p(X, L)$ is a uniform distribution over $[0, 512] \times [20, 128]$. The target is modeled as i.i.d Gaussian noise $N(\mu, \sigma^2)$ with the two nuisance parameters μ, σ.

We contrast two clutter models in this experiment. One is an i.i.d Gaussian model whose mean and variance are the sample mean and sample variance of the

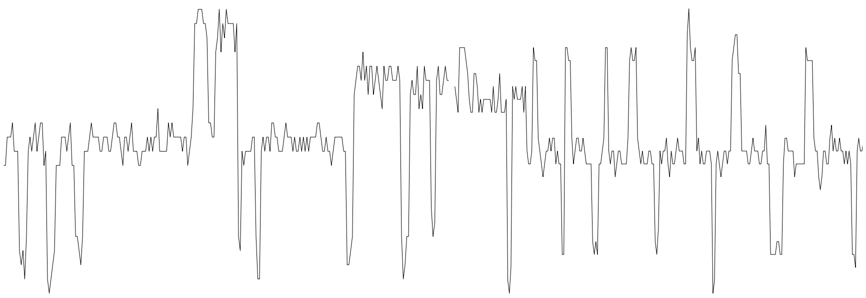


Figure 18 A observed signal with target of length $L = 100$ hidden at position $X = 200$.

true clutter rendered by the c-code, and other is the minimax entropy model learned in section 2.1. For computational ease, we simplify the learned model by a pseudo-likelihood approximation in the target detection. Readers are referred to [10] for theory of pseudo-likelihood distributions.

Figure 19 plots $-\log p^a(X, L|\mathbf{I}_{\text{obs}})$, where the dark points mean high probabilities, while bright points are for low probabilities. The upper one is computed using the Gaussian model for the clutter, and the dark points are false alarms, whereas the true target $(X = 200, L = 100)$ has a very low probability, compared with the false alarms. The projection of this function on the X axis, i.e. $-\log p^a(X|\mathbf{I}_{\text{obs}})$ is plotted in figure 20 (left).

The lower plot of figure 19 uses the learned clutter model with the pseudo-likelihood approximation. Its projection is also shown in figure 20 (right). Clearly, the true target is detected as the global maximum.

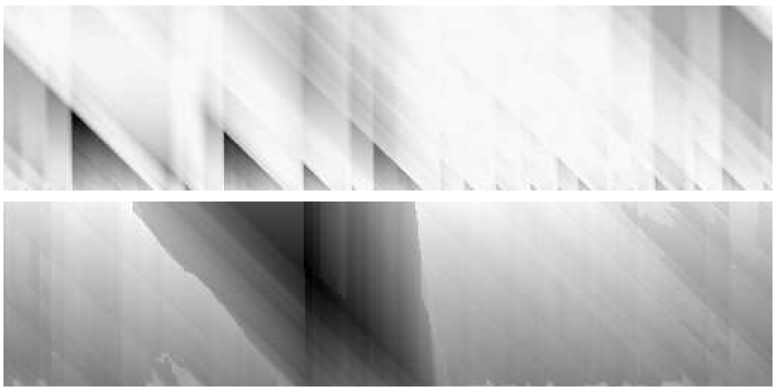
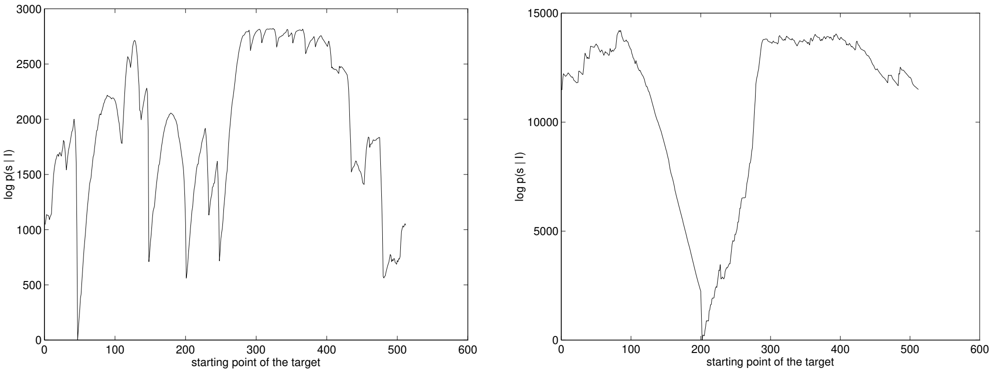


Figure 19 The horizontal axis is X, and the vertical axis is L. Upper: $-\log p(X, L|\mathbf{I}_{\text{obs}})$ Gaussian, lower: $-\log p(X, L|\mathbf{I}_{\text{obs}})$ pseudo-likelihood.



$-\log p(X|\mathbf{I}_{\text{obs}})$, Gaussian $\qquad$ $-\log p(X|\mathbf{I}_{\text{obs}})$, pseudo-likelihood

Figure 20 The projection of the plots into the X axis.

We will apply this scheme to 2D tank images in the future.

7 Discussion

In this paper, we studied the minimax entropy learning theory for clutter modeling, and discussed the general scheme for performance analysis. Three groups of experiments were demonstrated: 1) clutter modeling and learning, 2) effects of clutter on ATR performance, and 3) improvement of ATR performance with accurate clutter models.

There are other issues which are under investigation. Firstly, clutter should not be modeled independent of targets. For purpose of target detection, what important are the difference between targets and the clutter objects. For a lower cost clutter model and ATR system, the probability distribution for target and clutter should be learned together. Secondly, we want to explore occlusion issues explicitly in the the inference step. Thirdly, generic shape models of clutter objects should be formulated.

References

[1] T. Cover, and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[2] M. L. Cooper, A.D. Lanterman, S. Joshi, and M.I. Miller, "Representing the variation of thermodynamic state via principle component analysis", *Proc. Third Workshop on Conventional Weapon ATR*, November, 1996.

[3] U. Grenander, M.I. Miller, and A. Srivastava, "Hilbert-Schmidt lower bounds for estimators on matrix on Lie groups, preprint of Pattern Theory Group, Brown University, 1997.

[4] E.T. Jaynes, "Information theory and statistical mechanics", *Physical Review* 106, 620-630, 1957.

[5] M.I. Miller, U. Grenander, J.A. O'Sullivan, and D.L. Snyder, "Automatic target recognition organized via jump-diffusion algorithms", *IEEE Trans. Image Processing*, January,1997.

[6] J.A. Ratches, C.P. Walters, R.G. Buser and B.D. Guenther, "Aided and automatic target recognition based upon sensory inputs from image forming systems", *IEEE Trans. PAMI*, vol.19, No.9, Sept.1997.

[7] J. Rissanen. "Modelling by shortest data description" *Automatica*. 14:465-471. 1978.

[8] D.E. Schieder and M.R. Weathersby, "Detection performance in clutter with variable resolution", *IEEE Trans. AES*, vol. 19, no. 4., 1983.

[9] M.V. Shirvailkan and M.M. Trivedi, "Developing texture-based image clutter measures for object recognition", *Optical Eng.* vol. 31, no. 2, Dec. 1992.

[10] G. Winkler, *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods*, Springer-Verlag, 1995.

[11] S.C. Zhu and D.B. Mumford, "Prior Learning and Gibbs Reaction-Diffusion". IEEE Trans. on PAMI vol. 19, no. 11. Nov. 1997.

[12] S.C. Zhu, Y.N. Wu and D.B. Mumford, "Filters, Random fields, And Maximum Entropy (FRAME): towards a unified theory for texture modeling", *Int'l Journal of Computer Vision* 27(2) 1-20, March/April, 1998.