

# A BAYESIAN FRAMEWORK FOR STATISTICAL, MULTI-MODAL SENSOR FUSION

Michael J. Smith\*      Anuj Srivastava †

## Abstract

We propose a framework for obtaining statistical inferences from multi-modal and multi-sensor data. In particular, we consider a military battlefield scene and address problems that arise in tactical decision-making while using a wide variety of sensors (an infrared camera, an acoustic sensor array, a human scout, and a seismic sensor array). Outputs of these sensors vary widely, from 2D images and 1D signals to categorical reports. We propose novel statistical models for representing seismic sensor data and human scout reports while using standard models for images and acoustic data. Combining the joint likelihood function with a marked Poisson prior, we formulate a Bayesian framework and use a Metropolis-Hastings algorithm to generate inferences. We demonstrate this framework using experiments involving simulated data.

## 1 Introduction

Tactical decision makers in the military and in homeland security are increasingly dependent upon information collected by an ever-expanding array of electronic sensors. Commanders require systems that can either formulate decisions in an automated fashion or assist in decision making by processing the available sensor data. A specific problem is to detect, track, and recognize targets of interest in a battlefield situation using imaging and other sensing devices. The widespread use of sensors such as imaging devices has made them essential tools of non-invasive surveillance of battlefields and public areas such as airports and stadiums, as well as remote locations and areas of restricted access, where additional preventive measures are needed. Usage of multiple sensors observing a scene simultaneously has become a common situation. An important question for developing automated systems is: How to fuse information from these multiple sources to learn and understand the underlying scene? In this paper, we address this problem of sensor fusion using a statistical framework, by building probability models for sensor data and scene variables, and seeking high probability solutions.

What makes the problem of fusing sensor data a difficult one? An important issue is the widely different nature of outputs generated by different sensors. For instance, an IR camera generates a 2D image, a seismic sensor measures an electromagnetic wavefront, an acoustic sensor measures an audio signal, and a human scout reports categorical data. Traditional techniques of extracting features and merging feature vectors do not apply here directly. Past research in sensor fusion has generally focused on multiple sensors of similar type, e.g. multiple cameras or multiple signal receivers, and the solutions tend to exploit this similarity. The problem of sensor fusion from completely different sensors is much more difficult. An attractive solution is to take a statistical

---

\*Department of Mathematical Sciences, United States Military Academy, West Point, NY 10996

†Department of Statistics, Florida State University, Tallahassee, FL 32306

approach and to use joint probabilities instead of fusing data or features directly. That is, define a single inference space and use different sensor outputs to impose probabilities on this inference space. Despite differences in the nature of sensor outputs, the probabilities imposed can still be utilized individually or jointly to form scene estimates.

Some of the current ideas for fusing data from multiple sensors of similar type include the following. Viswanathan and Varshney [13] use likelihood ratio tests (LRTs) to combine the decisions of signal sensors operating in parallel; Costantini et al. [1] apply a least-squares approach to fuse synthetic aperture radar (SAR) images of different resolutions; Filippidis et al [2] study a similar problem using two SAR sensors. Rao et al [9] describe a decentralized Bayesian approach for identifying targets. Kam, Zhu, and Kalata [3] present a survey of techniques used in the problem of robot navigation including Kalman filtering, rule-based sensor fusion, fuzzy logic, and neural networks. However, rather limited attention has been focused on fusion of sensors with different modalities: Strobel et al. [12] describe the use of audio and video sensors for object localization using Kalman filtering; Ma et al. [5] use optical and radar sensor fusion for detecting lane and pavement boundaries. Some papers have focused on alternate frameworks for statistical sensor fusion: Mahler [6] develops the theory of finite-set statistics (FSST) as an extension of Bayesian methods for multiple-target tracking.

## 1.1 Bayesian Sensor Fusion

We take a fundamental approach to scene inference using a Bayesian formulation that is similar to the approach of Miller et al [7, 8]. Rather than extracting features, we choose to analyze the raw sensor data directly and jointly to estimate the locations and identities of target vehicles that are present. For this paper, we have avoided the difficulty of temporal registration of sensor outputs by assuming that all sensors are synchronized in time. However, our methodology obviates the need for spatial association — the fusion proceeds according to the conditional probabilities corresponding to each of the different data vectors.

We formulate the sensor fusion problem next. Consider a planar region of a battlefield containing an unknown number of targets of different types. Our goal is to use the sensor data to detect and recognize them. Let  $\mathcal{D} \subset \mathbb{R}^2$  be a region of interest in a battlefield, and let  $X$  denote an array of variables describing the target positions (in  $\mathcal{D}$ ) and types. In addition to target positions, there are a number of other variables, such as their pose, motion, load, etc, that can be of interest and, in general, one should estimate all of them. We simplify the problem by assuming these other variables to be known and fixed. In particular, we assume a fixed orientation for all target vehicles.

Table 1: Sensor Suite

<i>Label</i>	<i>Sensor</i>	<i>Nature of Operation</i>	<i>Detected Aspects</i>	<i>Output</i>
$s_1$	Infrared Camera	Low-Resolution Imager	Target Location & ID	2D Image Array ( $Y_1$ )
$s_2$	Acoustic Array	Audio Signal Receiver	Direction Only; No ID	1D Signal Vector ( $Y_2$ )
$s_3$	Scout	Human Vision	Rough Location; ID	Categorical Data ( $Y_3$ )
$s_4$	Seismic Array	Wave Receiver	Rough Location; Partial ID	Zone Detection ( $Y_4$ )

We cannot observe  $X$  directly; instead, we must rely on the data that the sensors generate. Sensors can typically detect only certain aspects of the scene; i.e., sensors are partial observers.

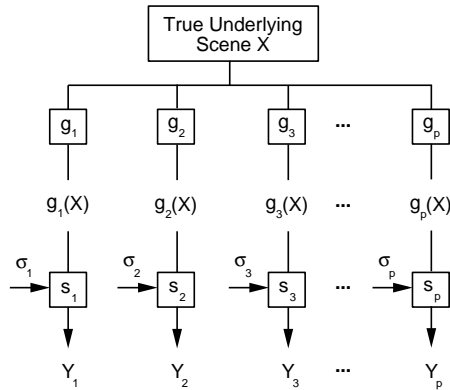


Figure 1: Sensor Data Derived from Projections of the Scene

Our goal is to use this partial and complementary information from different sensors to form a complete inference. As summarized in Table 1, an acoustic sensor array can detect the directions along which audio signals arrive from target vehicles, but it ascertains neither the targets’ radial distances along those directions nor the targets’ identities. A scout is trained to recognize target identities, but he has limited ability to report precise locations. Imaging sensors are also limited by their resolution, the possibility of target obscuration, and the presence of scene clutter. We assume the IR camera provides top views of the scenes using overhead shots. Despite their respective shortcomings, all of these sensors provide a means to discover the *number* of targets. In contrast, a seismic sensor is a “classifier” — it reports only target *type* (tracked vehicle, wheeled vehicle, dismounted personnel). We depend upon the complementary nature of the sensors and combine their data to conduct unified inference about the scene. Our choice of sensors is motivated by current practices and future plans of the military. In addition to the current routines of battlefield imaging using aerial (infrared) imaging and human scouting, the Army has interest in developing a variety of unmanned ground sensors (UGS) that include acoustic and seismic sensors. These UGS are advantageous over electronic/optical systems due to their low cost, low power requirement, and large detection/tracking range.

**Definition 1 Bayesian sensor fusion** is a methodology for scene inference that: (i) formulates a prior distribution for the scene, (ii) constructs probability models for multiple-sensor data conditioned on the scene, and (iii) conducts unified inference about the scene using the posterior distribution of the scene given the sensor data.

In Figure 1, we depict as projections  $g_1(X), \dots, g_p(X)$  the various aspects or attributes of the scene that our sensors  $s_1, \dots, s_p$  can detect. Each sensor is subject to observation errors  $\sigma_i$  in the generation of data vectors  $Y_1, \dots, Y_p$ . We assume that these errors are independent so that the  $Y_i$ s are conditionally independent given  $X$ . Let  $L_i(Y_i | X)$  denote the likelihood function for data vector  $Y_i$  conditioned on the scene  $X$  and let  $\nu_0(X)$  denote a prior distribution on the scene  $X$ . Applying Bayes’ rule and assuming conditional independence of the  $Y_i$ s given  $X$ , we obtain the posterior distribution of our interest:

$$\nu(X | Y_1, \dots, Y_p) \propto L_1(Y_1 | X) \cdots L_p(Y_p | X) \nu_0(X).$$

Our methodology leads us to generate estimates  $\hat{X}$  of the scene from the posterior distribution  $\nu(X | Y_1, \dots, Y_p)$ . Indeed, one may distinguish different Bayesian sensor fusion schemes according

to the sense in which their estimates are optimal. Several criteria such as MAP, posterior median, or MMSE, are commonly used. Techniques for producing optimal estimates according to any of these criteria are detailed in [11]. We employ Markov chain Monte Carlo methods to generate samples from the posterior distribution  $\nu(X | Y_1, \dots, Y_p)$ . Specifically, we implement a version of the Metropolis-Hastings algorithm in a MATLAB environment. We propose a prior distribution for the scene space and probability models for the four modes of sensor data mentioned above: infrared imagery ( $Y_1$ ), acoustic sensor data ( $Y_2$ ), a scout’s spot report ( $Y_3$ ), and seismic data ( $Y_4$ ). We apply our methodology to simulated battlefield scenes and obtain results that illustrate the inferential advantage to using all available sensor data.

Next, we outline major goals of this paper. (i) We propose statistical models for seismic sensor data and human scout reports, and derive their likelihood functions. (ii) Along with the established models for IR and acoustic sensors, we use these likelihood functions in formulating a fully Bayesian approach to battlefield inferences. And, (iii) we construct an MCMC solution to generating Bayesian inferences from the posterior distribution.

This paper is organized as follows. A representation of targets’ positions and identities, and statistical models for two sensors leading to a joint posterior distribution are presented in Section 2. A Metropolis-Hastings algorithm to sample from this posterior is described in Section 3. Some examples of scene inferences presented in Section 4. Finally, some simulation results are illustrated in Section 5.

## 2 Scene Representations and Sensor Models

This section presents statistical models and representations for the scene and the sensors. Because of its modular nature, our methodology can readily accommodate different or additional models that future research may suggest.

### 2.1 Scene Representation and Prior Model

Let  $X$  denote the positions and target identities of vehicles present in a region of the battlefield. We represent  $X$  as a point in the space  $\mathcal{X} = \bigcup_{n=0}^{\infty} (\mathcal{D} \times \mathcal{A})^n$ , where  $\mathcal{D} \subset \mathbb{R}^2$  is a battlefield region of interest,  $\mathcal{A} = \{\alpha_1, \dots, \alpha_M, \alpha_{\emptyset}\}$  is a set of  $M$  possible target types ( $\alpha_{\emptyset}$  means that no target is present), and  $n$  is the number of targets present. Since  $n$  is not known *a priori*, we allow for all possible values of  $n$  in the construction of  $\mathcal{X}$ . To support follow-on Markov chain development, we discretize the battlefield region  $\mathcal{D}$  along a rectangular grid: let  $\mathcal{D} = \{1, \dots, R\} \times \{1, \dots, C\}$  with  $R, C < \infty$ . This allows us to use  $(i, j)$  coordinates to denote target locations. We also impose the constraint  $n \leq RC$ . The motivation for an upper bound on the number of targets in a fixed region of the battlefield is clear: two targets cannot occupy the same physical space. We disallow the possibility that targets stack themselves vertically; the upper bound  $RC$  generously allows for target placements at each point in the discretized region. This modifies the state space to be both discrete and finite:  $\mathcal{X} = \bigcup_{n=0}^{RC} (\mathcal{D} \times \mathcal{A})^n$ . We express a typical state  $X \in \mathcal{X}$  as a matrix:

$$X = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \\ c_1 & c_2 & \cdots & c_n \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix}, \text{ where } (r_i, c_i)^T \text{ are coordinates of target locations. Each column of } X$$

represents a target described by its center-mass location (row and column) and its identity ( $\alpha$ ). Let  $\|X\| = n$  denote the number of columns in the state matrix  $X$  and let  $X_j$  denote the  $j^{\text{th}}$  column of  $X$  for  $j = 1, \dots, n$ . For  $n = 0$ , let  $X_{\emptyset}$  denote the empty state.

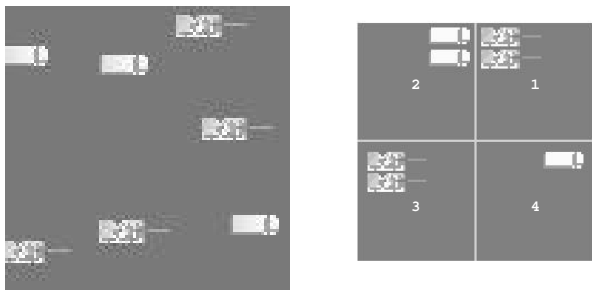


Figure 2: Left panel shows the top view of a simulated scene containing three trucks and four tanks. Right panel shows a visual rendering of scout’s spot report  $Y_3$  (right) with labeled quadrants.

We consider  $X$  to be a realization of a marked homogeneous Poisson spatial point process. In other words, we make the following collection of assumptions. Let  $N \sim \text{Poisson}(\lambda|\mathcal{D}|)$  for some  $\lambda > 0$  where  $|\cdot|$  denotes Lebesgue measure on  $\mathbb{R}^2$  and we assume that  $|\mathcal{D}| > 0$ . Conditioned on  $\{N = n\}$ , let the locations  $q_1, \dots, q_n$  of targets be distributed independently and uniformly in  $\mathcal{D}$ . Conditioned on the locations  $q_1, \dots, q_n$ , let the target identities be assigned independently: for each location, assign identity  $\alpha_j \in \mathcal{A}$  with probability  $\pi_j \geq 0$  for  $j = 1, \dots, M$  where  $\sum_{j=1}^M \pi_j = 1$ . These assumptions specify a prior probability measure  $\nu_0$  defined on a  $\sigma$ -field of subsets of  $\mathcal{X}$ .

## 2.2 Sensor Models

Here we detail the statistical models that we have adopted for the various sensors under consideration: infrared camera  $s_1$ , acoustic sensor array  $s_2$ , human scout  $s_3$ , and seismic sensor array  $s_4$ . For  $s_1$  and  $s_2$ , we use established models from the literature with incorporation details contained in [11]. However, this paper offers new models for  $s_3$  and  $s_4$  and provides detailed motivations for both.

### 2.2.1 Model for Scout’s Spot Report

Army units conduct routine tactical operations in accordance with standing operating procedures or SOPs. Among other provisions, SOPs prescribe reporting formats that scouts use to communicate their observations to higher headquarters. Here we assume that the “spot report” format calls for a partitioning of the observed area  $\mathcal{D}$  into four quadrants and that the report provides quadrant counts for each target type. See Figure 2 for an illustration. Let  $Y_3$  denote the scout’s spot report. We represent it as a vector of length  $4M$  where  $M$  is the number of target vehicle identities in  $\mathcal{A} \setminus \alpha_\emptyset = \{\alpha_1, \dots, \alpha_M\}$ . Each component of  $Y_3$  belongs to  $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$ . We propose a hierarchical model for the conditional distribution of  $Y_3$  given  $X$ .

To motivate the construction of the model, we may suppose that the scout sequentially answers questions that he poses to himself: *How many targets? Where are they? What are they?* He answers the first question by counting those vehicles that he can see. A reasonable model should therefore allow for a variety of cases: he sees all the vehicles that are present; he misses one or more; he “sees” one or more vehicles that are *not* present; he loses track of his count and begins repeating vehicles that he has already counted. But then, regardless of how the scout arrives at his collection of observed targets, he must decide — vehicle by vehicle — how to classify them according to quadrant and target type. Again, a reasonable model should allow for some ambiguity

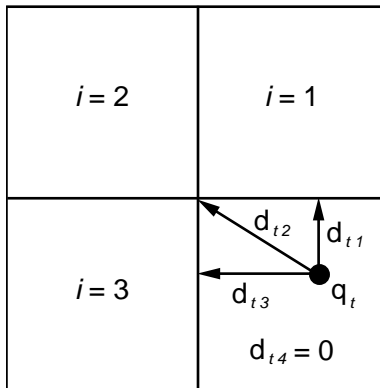


Figure 3: Distances to Nearest Quadrant Boundaries for a Fourth-Quadrant Target

in the quadrant classification of a target lying close to a quadrant boundary. The model detailed below exhibits one way to incorporate these observations about the nature of the scout’s report.

**Total Count:** Let  $N_S$  be the total number of target vehicles that the scout observes. We model  $N_S$  as a discrete random variable taking values in  $\mathbb{Z}_+$  with probability masses obtained by evaluating a Gaussian density function at these points and then normalizing. To specify the Gaussian density, we set the mean  $\mu$  equal to  $n$  (the actual number of targets in the scene) and we take the variance  $\sigma^2$  to be this function of the mean:

$$\sigma^2(n) = \begin{cases} \beta_0, & \text{if } n = 0; \\ n/\beta_1, & \text{if } n > 0, \end{cases}$$

where  $\beta_0$  and  $\beta_1$  are chosen to account for the scout’s level of training, his competence, the status of his equipment, weather conditions, and other sources of error. Note that the variance increases linearly with the true target count. Let  $\mathcal{G}(k) = \mathcal{G}(k | n, \beta_0, \beta_1)$  denote the probability mass that this discretized Gaussian distribution places on  $k$ . Then

$$\mathcal{G}(k) = \begin{cases} \frac{\exp\left(-\frac{1}{2\beta_0}(k-n)^2\right)}{\sum_{z \in \mathbb{Z}_+} \exp\left(-\frac{1}{2\beta_0}(z-n)^2\right)}, & n = 0; \\ \frac{\exp\left(-\frac{\beta_1}{2n}(k-n)^2\right)}{\sum_{z \in \mathbb{Z}_+} \exp\left(-\frac{\beta_1}{2n}(z-n)^2\right)}, & n > 0. \end{cases}$$

**Quadrant Target Counts:** Given  $\{N_S = n_0\}$ , we model the components of  $Y_3$  as sums of classification counts constrained so that  $\sum_{j=1}^{4M} (Y_3)_j = n_0$ . The counts tally the outcomes of “generalized Bernoulli” trials. That is, each observed target corresponds to a conditionally independent trial; each trial has  $4M$  possible outcomes corresponding to the scout’s possible quadrant & target-type classifications. The outcome of trial  $t$  is governed by parameters  $\{p_{tj}\}_{j=1}^{4M}$  that satisfy  $p_{tj} \geq 0$  and  $\sum_{j=1}^{4M} p_{tj} = 1$  for each  $t = 1, \dots, n_0$ . These generalized Bernoulli parameters, in turn, depend upon the scene  $X$ . (Note that we have a different collection of generalized Bernoulli parameters for each vehicle. If, instead, we had one fixed collection of parameters applicable for all  $n_0$  observed targets, then  $Y_3$  would follow a conditional multinomial distribution given  $X$ .)

Now we describe the choice of generalized Bernoulli parameters  $\{p_{tj}\}_{j=1}^{4M}$  for the  $t^{\text{th}}$  trial. Consider a target at location  $q_t = (r_t, c_t)^T$  and suppose that  $q_t$  lies within quadrant  $i_t$ . Our convention is that a target's location is specified by its center-of-mass. As illustrated in Figure 3, let  $d_{ti}$  denote the distance from  $q_t$  to the  $i^{\text{th}}$  quadrant — Euclidean distance to the nearest quadrant boundary — and set  $d_{ti} = 0$ . Then, for a fixed constant  $a > 0$ , set

$$\tilde{p}_{ti} = \frac{\exp(-d_{ti}/a)}{\sum_{j=1}^4 \exp(-d_{tj}/a)}, \quad i = 1, 2, 3, 4. \quad (1)$$

In words,  $\tilde{p}_{ti}$  is the probability that the scout reports quadrant  $i$  as the location for target  $t$ . Now we account for the scout's reported target type. Let  $I\{\alpha_t = j\}$  indicate that  $\alpha_j$  is the identity of target  $t$ ; let  $I\{\alpha_t \neq j\}$  indicate that  $\alpha_j$  is not the identity of target  $t$ . We use these indicators and a classification error parameter denoted  $\sigma_3$  to split each  $\tilde{p}_{ti}$ : for  $j = j_i = (i - 1)M + 1, \dots, iM$  with  $i = 1, 2, 3, 4$ , put

$$p_{tj} = (1 - \sigma_3) \tilde{p}_{ti} I\{\alpha_t = j_i\} + \frac{\sigma_3}{M - 1} \tilde{p}_{ti} I\{\alpha_t \neq j_i\}.$$

In words, the scout correctly reports the target type with high probability and he is equally likely to report any of the incorrect target types.

We apply the above formulation of generalized Bernoulli parameters  $\{p_{tj}\}_{j=1}^{4M}$  to each of the vehicles that the scout observes ( $t = 1, \dots, n_0$ ). If it happens that  $n_0 = n$ , where  $n$  is the correct number of vehicles, we assume that the scout observes each target exactly once and that he classifies them independently as above-described trials. In case  $n_0 < n$ , we assume that the scout observes and similarly classifies a proper subset of targets, where each of  $\binom{n}{n_0}$  subsets is equally likely. In case  $n < n_0 \leq 2n$ , we assume that the scout classifies all targets that are present and that he “double counts”  $n_0 - n$  targets, where each of  $\binom{n}{n_0 - n}$  collections of doubly-counted targets is equally likely. Let  $\lfloor \cdot \rfloor$  denote the greatest integer less than or equal to its argument. For  $n_0 > 2n$ , we assume that the scout repeatedly classifies each target  $k$  times, where  $k = \lfloor \frac{n_0}{n} \rfloor$ , and then augments this redundancy by including an equally-likely choice from among  $\binom{n}{r}$  subsets where  $r = n_0 \bmod k$ .

**Likelihood Function:** As suggested earlier, the scout's target-set selection can be modeled in many ways. For the scheme described above, conditioned on  $\{N_S = n_0\}$ , let  $T \in \mathcal{X}$  denote the array of targets that the scout observes. Let  $\mathcal{P}(T)$  denote the collection of column permutations of  $T$  and let  $T_o \in \mathcal{P}(T)$  denote an ordered  $n_0$ -tuple of targets (locations and identities). Then the description in this section leads to this likelihood function for  $Y_3$ :

$$L_3(Y_3 | X) = \frac{\mathcal{G}(n_0)}{n_0!} \sum_{T_o \in \mathcal{P}(T)} \prod_{t=(T_o)_1}^{(T_o)_{n_0}} p_{t1}^{(Y_3)_1} p_{t2}^{(Y_3)_2} \dots p_{t,4M}^{(Y_3)_{4M}}. \quad (2)$$

The permutations arise because the scout may perform his vehicle-by-vehicle classification according to any ordering; each makes an equally-weighted contribution to the likelihood.

### 2.2.2 Model for Seismic Data

Open-source documentation about seismic sensors is easy to find; see, for example, “Remote Battlefield Sensor System (REMBASS) and Improved Remote Battlefield Sensor System (IREMBASS)” at the location<sup>1</sup>. According to such sources, a seismic sensor detects and classifies (but does not

<sup>1</sup><http://www.fas.org/man/dod-101/sys/land/rembass.htm>

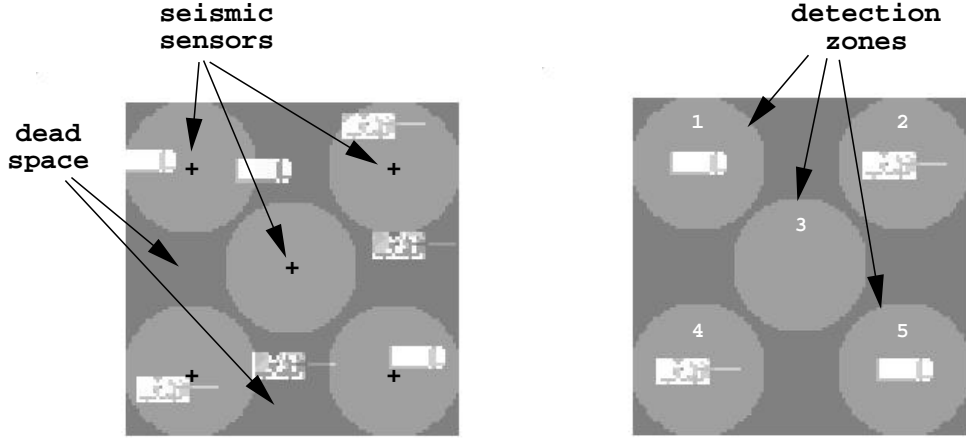


Figure 4: Same Simulated Scene with Overlay of Seismic Detection Zones (left) and Visual Rendering of Seismic Data  $Y_4$  (right) with  $k = 5$  labeled zones.

count) those targets whose ground vibrations emanate from within a circular detection zone of known radius. Depending upon the placement of the sensors, these detection zones may or may not overlap. Additionally, the battlefield region  $\mathcal{D}$  may contain “dead space” where target vehicles are not detectable by any of the seismic sensors.

Any statistical model that describes data collected by these sensors should reflect certain key aspects of the sensors’ behavior. First, whether or not the sensors detect the target vehicles depends upon the locations of the vehicles, the locations of the seismic sensors, and each sensor’s detection zone radius. As stated in Section 2.2.1, our convention is that a target’s location is specified by its center-of-mass. We assume that all seismic sensors have circular, non-intersecting detection zones with equal radii as depicted in Figure 4. Second, each sensor provides a single classification that summarizes the target-type presence in its zone. If at most one target type is present, there is no confusion. But a statistical model should contain some mechanism whereby the sensor reconciles the presence of more than one target type in its detection zone. The model that we propose offers one way to address these issues.

Assume that  $k$  seismic sensors having mutually disjoint detection zones generate a data vector  $Y_4$  with  $k$  components (one for each sensor). Let  $(Y_4)_j \in \mathcal{A}$  report the  $j^{\text{th}}$  sensor’s summary of target-type presence in its detection zone. Figure 4 provides an illustration for  $k = 5$ . Its left panel shows the overlay of detection zones on top of the same simulated scene displayed for the scout’s spot report. Note that two tanks and one truck lie in dead space — their center-mass locations are not within any of the detection zones. For this scene, the correct seismic data vector is  $Y_4 = [\alpha_2, \alpha_1, \alpha_\emptyset, \alpha_1, \alpha_2]'$  where  $\alpha_1 = \text{tank}$  and  $\alpha_2 = \text{truck}$ . A visual rendering of  $Y_4$  appears in the right panel where the labeling of the detection zones to match vector components is as follows: top-left is 1, top-right is 2, center is 3, bottom-left is 4, and bottom-right is 5.

Let  $\sigma_4$  denote a fixed error parameter and let  $n_{ij}$  denote the number of type- $\alpha_i$  targets ( $i = 1, \dots, M$ ) that are present in detection zone  $j = 1, \dots, k$ . Fix a detection zone  $j$  and let  $P(\cdot)$  denote a probability measure defined as follows on all subsets of  $\mathcal{A}$ .

- **Case 1** If zone  $j$  is devoid of target vehicles, we allow the sensor to report correctly with



high probability and we assume that the sensor is equally likely to report erroneously any of the target types:

$$P\{(Y_4)_j = y \mid n_{1j} = \dots = n_{Mj} = 0\} = \begin{cases} 1 - \sigma_4, & y = \alpha_\emptyset; \\ \frac{\sigma_4}{M}, & y \in \mathcal{A}; \\ 0, & \text{otherwise.} \end{cases}$$

- **Case 2** If zone  $j$  contains exactly one target type, we again allow the sensor to report correctly with high probability. However, in this case, we assume that the sensor is more likely to report an incorrect target type than to report the absence of targets; we assume that all wrong-type classifications are equally likely.

$$P\{(Y_4)_j = y \mid n_{ij} > 0 \text{ for } i = i_0 \text{ only}\} = \begin{cases} \frac{\sigma_4}{4}, & y = \alpha_\emptyset; \\ 1 - \sigma_4, & y = \alpha_{i_0}; \\ \frac{3\sigma_4}{4(M-1)}, & y \in \mathcal{A} \setminus \{\alpha_{i_0}\}; \\ 0, & \text{otherwise.} \end{cases}$$

- **Case 3** This is most interesting — the sensor must “decide” among competing target types. Denote by  $n_{\cdot j} = \sum_{i=1}^M n_{ij}$  the number of targets (of all types) present in detection zone  $j$ . Let  $I\{\alpha_t = i\}$  indicate whether  $\alpha_i$  is the identity of target  $t = 1, \dots, n_{\cdot j}$ . Let  $a > 0$  be a fixed constant and let  $d_t$  denote the distance from target  $t$  to the center of the detection zone. These distances are analogous to those depicted in Figure 3 and contribute to the classification probabilities in a manner similar to Equation 1.

$$P\{(Y_4)_j = y \mid 2 \leq |\{i : n_{ij} > 0\}|\} = \begin{cases} \sigma_4, & y = \alpha_\emptyset; \\ (1 - \sigma_4) \frac{\sum_{t=1}^{n_{\cdot j}} I\{\alpha_t = i\} e^{-d_t/a}}{\sum_{t=1}^{n_{\cdot j}} e^{-d_t/a}}, & y = \alpha_i \in \mathcal{A}; \\ 0, & \text{otherwise.} \end{cases}$$

We assume that the seismic sensors’ classifications are conditionally independent given the scene. The above enumeration of cases depending on  $X$  and the assumption of conditional independence lead to the following likelihood function for  $Y_4$ :

$$L_4(Y_4 | X) = \prod_{j=1}^k P\{(Y_4)_j = y | X\}. \quad (3)$$

### 2.3 The Posterior Distribution

The likelihood functions  $L_1(Y_1 | X)$  and  $L_2(Y_2 | X)$  for infrared images and acoustic data (respectively) are given in [11]. Combined with the likelihood functions derived in this paper, and along with the assumption of conditional independence of the data vectors, we may now express the posterior distribution:

$$\nu(X | Y_1, Y_2, Y_3, Y_4) \propto L_1(Y_1 | X) L_2(Y_2 | X) L_3(Y_3 | X) L_4(Y_4 | X) \nu_0(X). \quad (4)$$

Although we will sometimes use the shorthand  $\nu(\cdot) \equiv \nu(\cdot | Y_1, Y_2, Y_3, Y_4)$ , we will always mean that the likelihood functions  $L_i(Y_i | X)$  are defined (respectively) as in Equations 2 and 3 (and as in [11]) and that the prior distribution  $\nu_0$  is defined as in Section 2.1.

### 3 Metropolis-Hastings Algorithm

So far we have defined a posterior distribution  $\nu$  on the scene space  $\mathcal{X}$ , and our task now is to obtain samples from the posterior distribution  $\nu$  so that we may conduct scene inference. This section presents the algorithm we use to generate approximate samples from  $\nu$ .

#### 3.1 Transitions of the Markov Chain

We control the evolution of the Markov chain by restricting the one-step transitions to a class of “simple moves.” Although this slows down the convergence of the resulting Markov chain, we impose the restriction because analyzing the chain is easier in this setting [4, 7, 8].

Given the current state  $X^{(t)}$  at time  $t$ , we consider four fundamental types of transitions. To each type corresponds a collection of “neighboring” states (neighbors of  $X^{(t)}$ ) — the states that can be reached from  $X^{(t)}$  in one transition. We now introduce notation for these sets of neighbors.

1. The first simple move is **DEATH**. This means that we select and remove one of the current targets from the state matrix. Let  $\mathcal{N}_D(X^{(t)})$  denote the neighbors of state  $X^{(t)}$  under the **DEATH** transition. Define

$$\mathcal{N}_D(X^{(t)}) = \begin{cases} \{X_{-j}^{(t)} : j = 1, \dots, n\}, & \text{if } \|X^{(t)}\| \geq 1; \\ \{X^{(t)}\}, & \text{if } \|X^{(t)}\| = 0, \end{cases}$$

where  $X_{-j}^{(t)}$  denotes the matrix  $X^{(t)}$  after removing column  $j$ .

2. The second simple move is **CHANGE ID**. This means that we select a current target in the state matrix and change its identity  $\alpha$ . Let  $\mathcal{N}_C(X^{(t)})$  denote the neighbors of state  $X^{(t)}$  under the **CHANGE ID** transition. Define

$$\mathcal{N}_C(X^{(t)}) = \begin{cases} \{X_{\Delta j}^{(t)} : j = 1, \dots, n\}, & \text{if } \|X^{(t)}\| \geq 1; \\ \{X^{(t)}\}, & \text{if } \|X^{(t)}\| = 0, \end{cases}$$

where  $X_{\Delta j}^{(t)}$  denotes the matrix  $X^{(t)}$  after changing the identity component of column  $j$ .

3. The third simple move is **ADJUST**. This means that we select a current target in the state matrix and slightly perturb its location  $q$ . Let  $\mathcal{N}_A(X^{(t)})$  denote the neighbors of state  $X^{(t)}$  under the **ADJUST** transition. Define

$$\mathcal{N}_A(X^{(t)}) = \begin{cases} \{X_{\oplus j}^{(t)} : j = 1, \dots, n\}, & \text{if } \|X^{(t)}\| \geq 1; \\ \{X^{(t)}\}, & \text{if } \|X^{(t)}\| = 0, \end{cases}$$

where each  $X_{\oplus j}^{(t)}$  denotes as many as eight perturbations to the location components of  $X_j^{(t)}$ . For example, if a current target has location  $q = (i, j)$ , then we permit an adjustment to  $q' \in \{(i \pm 1, j), (i \pm 2, j), (i, j \pm 1), (i, j \pm 2)\} \cap \mathcal{D}$ . The symbol  $\oplus$  is suggestive of this perturbation pattern of rows and columns.

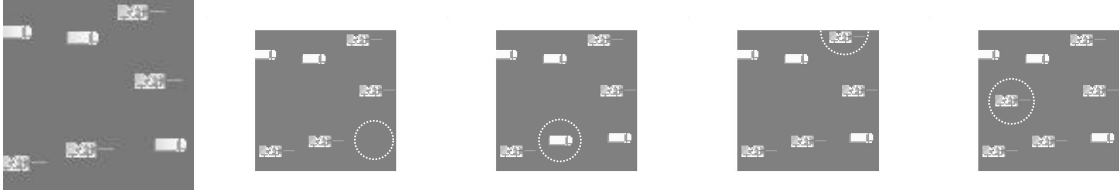


Figure 5: Simple Moves from current state (leftmost) to perform DEATH, CHANGE ID, ADJUST, and BIRTH (left-to-right).

4. The fourth simple move is **BIRTH**. This means that we augment the current state matrix by the addition of another target. Let  $T_{X^{(t)}} = \{X_j^{(t)} : j = 1, \dots, n\} \subset (\mathcal{D} \times \mathcal{A})$  denote the collection of targets represented in state matrix  $X^{(t)}$  and let  $\mathcal{N}_B(X^{(t)})$  denote the neighbors of state  $X^{(t)}$  under the **BIRTH** transition. Define

$$\mathcal{N}_B(X^{(t)}) = \{X_\tau^{(t)} : \tau \in (\mathcal{D} \times \mathcal{A}) \setminus T_{X^{(t)}}\},$$

where  $X_\tau^{(t)}$  is the augmentation of the matrix  $X^{(t)}$  by one additional column  $\tau$  corresponding to any “legal” target not already present:  $\|X_\tau^{(t)}\| = \|X^{(t)}\| + 1$ .

To help visualize the slight adjustments to a given state matrix  $X^{(t)}$  contained in the sets of neighbors  $\mathcal{N}_D(X^{(t)})$ ,  $\mathcal{N}_C(X^{(t)})$ ,  $\mathcal{N}_A(X^{(t)})$ ,  $\mathcal{N}_B(X^{(t)})$ , we present examples in Figure 5. The ADJUST example depicts a shift of the uppermost tank; the other examples are obvious. In Chapter 5, we present portions of Markov chain sample paths that exhibit incremental adjustments similar to Figure 5.

### 3.2 The Metropolis-Hastings Algorithm

We now state the basic algorithm that prescribes the evolution of our Markov chain; see, for example, Robert and Casella [10]. Fix a state space  $\mathcal{X}$  and let  $\nu$  (known as the *target* distribution) be a probability distribution on  $\mathcal{X}$ .

#### Algorithm 1 (Metropolis-Hastings)

Given the current state  $X^{(t)} \in \mathcal{X}$ ,

1. Generate  $Y_t \sim G(y|X^{(t)})$ . ( $G$  is called the *proposal distribution*.)

2. Set  $X^{(t+1)} = \begin{cases} Y_t & \text{w.p. } \gamma(X^{(t)}, Y_t); \\ X^{(t)} & \text{w.p. } 1 - \gamma(X^{(t)}, Y_t), \end{cases}$  where  $\gamma(x, y) = \min\left\{1, \frac{\nu(y)G(x|y)}{\nu(x)G(y|x)}\right\}$ .

For a large class of proposal distributions  $G$  and for  $X^{(1)} \sim F$  where  $F$  is an arbitrary probability distribution on  $\mathcal{X}$ , this algorithm is known to generate a Markov chain with unique stationary distribution  $\nu$ . For a detailed description of  $G$  based on the simple moves in Section 3.1 and for a discussion on asymptotic properties of this Markov chain, please refer to [11].

## 4 Conducting Scene Inference

Implementing the Metropolis-Hastings algorithm in MATLAB, we obtain an approximate sample from  $\nu(\cdot | Y_1, Y_2, Y_3, Y_4)$ . Specifically, we generate  $X^{(1)} \sim \nu_0$  (prior distribution) and then observe  $X^{(2)}, X^{(3)}, \dots$  according to the Metropolis-Hastings transition kernel. After stopping the chain, we discard the first  $B$  states (sometimes called a *burn-in* period to allow time for the Markov chain to approach its stationary distribution) and we retain, for purposes of inference,

$$\{X^{(B+1)}, X^{(B+2)}, \dots, X^{(B+R)}\}.$$

In this chapter, we describe methods for using our sample to answer a variety of questions. We denote the retained portion of the Markov chain by

$$\{X_j\}_{j=1}^R \quad \text{where we set} \quad X_1 = X^{(B+1)}, \dots, X_R = X^{(B+R)}. \quad (5)$$

Letting  $\nu$  denote the posterior distribution of the scene, we proceed under the assumption that  $\{X_j\} \sim \nu$ .

Having obtained a sample  $\{X_j\}$  from the posterior distribution  $\nu$ , we might wish to produce a *maximum a posteriori* estimate  $\hat{X}_{\text{MAP}}$  of the scene. Such an estimate is characterized by  $\hat{X}_{\text{MAP}} = \operatorname{argmax}_{X \in \mathcal{X}} \nu(X)$ , that is,  $\hat{X}_{\text{MAP}}$  is a *mode* of the posterior distribution. An obvious candidate to estimate  $\hat{X}_{\text{MAP}}$  is the sample mode: we can simply report the state matrix that appears most frequently among  $\{X_j\}$ . An alternative approach abandons the previously described sample and instead uses an adjustment to the Metropolis-Hastings algorithm given earlier. The technique is known as *simulated annealing* and it provides a means to obtain MAP estimates  $\hat{X}_{\text{MAP}}$ ; see, for example, Winkler [14].

## 5 Simulation Results

Now we present some experimental results demonstrating the proposed framework for Bayesian sensor fusion. In these experiments, we utilize sensor data simulated according to the models proposed.

We start with a simulated scene with corresponding sensor data in Figure 6 and construct a Markov chain to sample from the resulting posterior. Figure 7 shows periodic snapshots along a sample path of this Markov chain in  $\mathcal{X}$ . Before proceeding with scene inference, we make some qualitative observations about the performance of our algorithm. The top-left panel in Figure 7 depicts the initial state. Navigating through the panels in left-to-right, top-to-bottom fashion, we see the state of the chain at multiples of 100 steps. The bottom-right panel depicts the true scene. At a glance, we observe that this particular sample path evolves quite close to the true scene. Figure 8 illustrates how the *posterior energy* associated with a sample path regulates the evolution of the Metropolis-Hastings algorithm. It depicts  $H(X^{(t)}) \propto -\log \nu(X^{(t)})$  plotted against  $\frac{t}{25}$ . The non-increasing nature of the posterior energy indicates that the Metropolis-Hastings algorithm is indeed steering the sample path toward target configurations with more and more probability mass under the posterior distribution.

## 6 Summary

We have presented a statistical framework for merging information from multi-modal sensors in order to generate a unified inference. To setup a Bayesian problem, we have introduced statistical

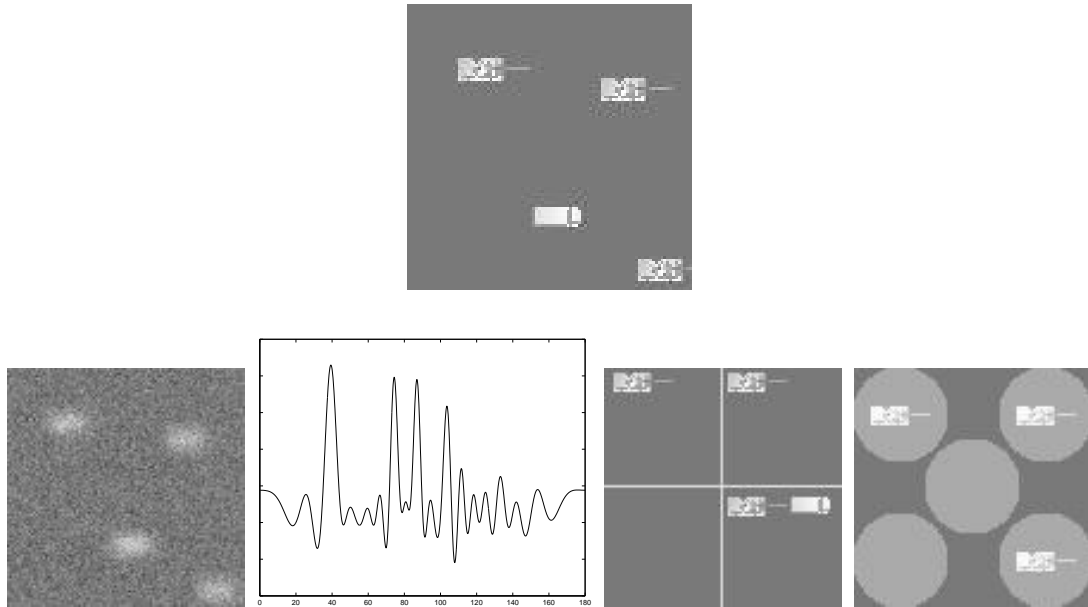


Figure 6: Simulated Scene 2 (top) and Corresponding Sensor Data (bottom) from (left to right) Infrared Camera, Acoustic Sensor Array, Scout, Seismic Sensor Array

models for two sensors - seismic sensor and human scout - and used established models for infrared camera and acoustic array. Assuming a homogeneous Poisson prior on the target placements in the scene, we formulate a posterior distribution on the configuration space, and utilize a Metropolis-Hastings algorithm to generate samples and inferences from it. Experimental results are presented for detecting and recognizing targets in a simulated battlefield scene.

## References

- [1] Mario Costantini, Alfonso Farina, and Francesco Zirilli. The fusion of different resolution SAR images. In *Proceedings of the IEEE, Vol 85, No 1*, pages 139–146. IEEE, January 1997.
- [2] A. Filippidis, L. C. Jain, and N. Martin. Fusion of intelligent agents for detection of aircrafts in SAR images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):378–384, 2000.
- [3] Moshe Kam, Xiaoxun Zhu, and Paul Kalata. Sensor fusion for mobile robot navigation. In *Proceedings of the IEEE, Vol 85, No 1*, pages 108–119. IEEE, January 1997.
- [4] A. Lanterman, M. Miller, and D. Snyder. General Metropolis-Hastings jump diffusions for automatic target recognition in infrared scenes. *Optical Engineering*, 36(4):1123–1137, 1997.
- [5] B. Ma, S. Lakshmanan, and A. O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transport Systems*, 1(3):135–147, 2000.
- [6] Ronald Mahler. *An Introduction to Multisource-Multitarget Statistics and its Applications*. Lockheed Martin, Eagan MN, 2000.

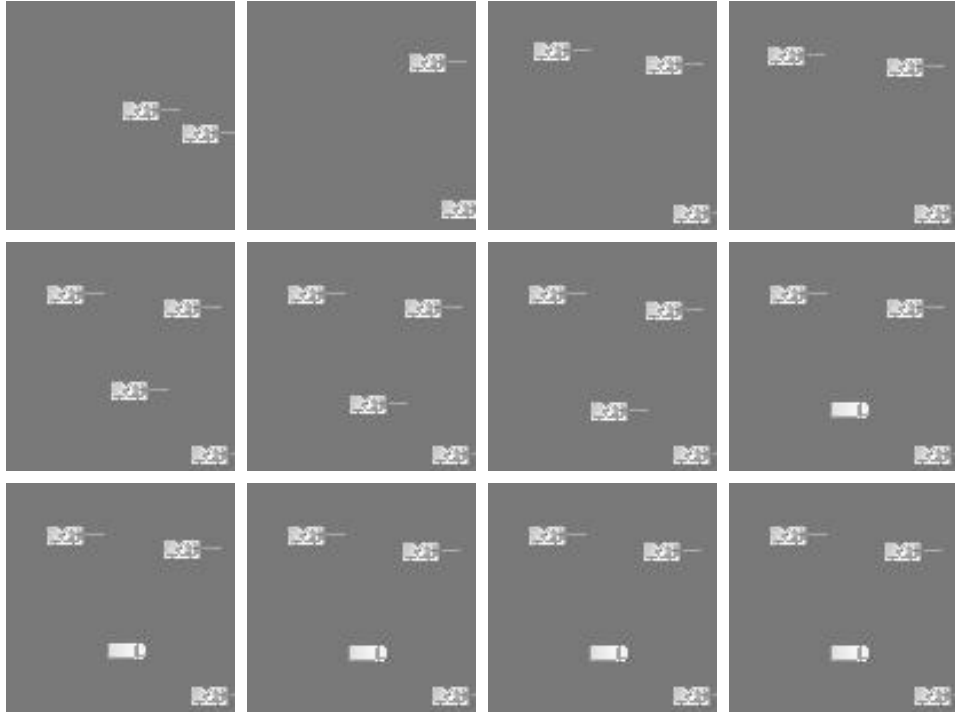


Figure 7: Evolution of Markov Chain for Simulated Scene 2: (left-to-right and top-to-bottom)  $X^{(1)}, X^{(100)}, X^{(200)}, \dots, X^{(1000)}, X_{\text{TRUE}}$

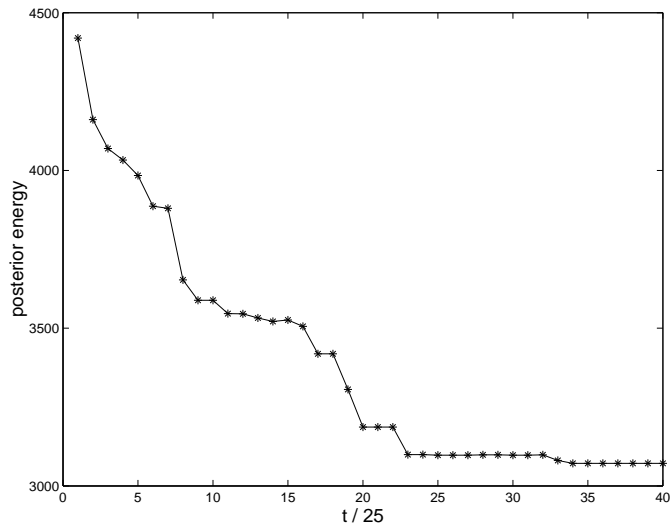


Figure 8: Posterior Energy: Evolution for the Figure 7 Sample Path

- [7] 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.
- [8] M. I. Miller, A. Srivastava, and U. Grenander. Conditional-expectation estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43(11):2678–2690, November 1995.
- [9] B. S. Rao and H. Durant-Whyte. A decentralized bayesian algorithm for indentification of tracked targets. *IEEE Transaction on Systems, Man and Cybernetics*, 23(6):1683–1698, 1993.
- [10] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.
- [11] M. J. Smith. Bayesian sensor fusion: A framework for using multi-modal sensors to estimate target locations & identities in a battlefield scene. *PhD Dissertation, Florida State University, Tallahassee, FL*, August 2003.
- [12] N. Strobel, S. Spors, and R. Rabenstein. Joint audio-video object localization and tracking. *IEEE Signal Processing Magazine*, 18(1), 2001.
- [13] Ramanarayanan Viswanathan and Pramod K. Varshney. Distributed detection with multiple sensors: Fundamentals. In *Proceedings of the IEEE, Vol 85, No 1*, pages 54–63. IEEE, January 1997.
- [14] Gerhard Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, New York, 1999.