A Rapidly Deployable Virtual Presence Extended Defense System

Mark W. Koch
Sandia National Laboratories†
PO Box 5800, MS 1163
Albuquerque, NM 87185-1163
mwkoch@sandia.gov

Casey Giron
Sandia National Laboratories
PO Box 5800, MS 1163
casgiro@sandia.gov

Hung D. Nguyen
Sandia National Laboratories
PO Box 5800, MS 1162
hdnguye@sandia.gov

Abstract

We have developed algorithms for a virtual presence and extended defense (VPED) system that automatically learns the detection map of a deployed sensor field without a-priori knowledge of the local terrain. The VPED system is a network of sensor pods, with each pod containing acoustic and seismic sensors. Each pod has a limited detection range, but a network of pods can form a virtual perimeter. The site's geography and soil conditions can affect the detection performance of the pods. Thus a network in the field may not have the same performance as a network designed in the lab. To solve this problem we automatically estimate a network's detection performance as it is being constructed. We demonstrate results using simulated and real data.

1. Introduction and Problem

We are developing a smart unattended sensor system for use as a virtual perimeter and extended defense. Each sensor pod has the ability to detect and classify moving targets, but at a limited range. By using a network of pods we can form a virtual perimeter with each pod responsible for a certain section of the perimeter. By fusing together individual detections we can improve the detection range of the network over a single pod's detection rage.

The site's geography and soil conditions affect the detection performance of the sensor pods. Thus a network designed in the lab may not have the same detection performance as a network in the field. Our solution is unique in that we estimate the network's detection characteristics as the sensor network is being laid out. This allows us to adapt to the local terrain by identifying vulnerabilities or gaps in the detection map. The system can suggestion insertion of new sensor pods or moving previously placed pods to fill these gaps. We believe our approach is general enough to apply to other types of sensor systems.

Historically, determining the probability of detection (PD) of an intrusion detection sensor has been done manually [3]. For example, one could approach the sensor from different directions and record the range of the initial detection. If each direction experiment is repeated 30 times, then the average range for a 90% PD with 95% confidence interval can be identified using a binomial distribution [2].

2. Overall Approach

Figure 1 shows a conceptual diagram of our overall approach. We assume a user draws a perimeter on a digital map around the area to be defended. The system then goes and determines the locations of suggested placements, indicated by the white filled circles.

Next a mobile deployment unit (MDU) wearing a differential GPS unit starts placing sensor pods as close as possible to the suggested placements as indicated by a laptop computer. The actual placements of the pods maybe at locations different from the suggested locations, since natural vegetation or other obstacles make the placement at these locations impossible. As the MDU places the pods, he or she makes detections that the system can use to update its probability of detection map and to determine the vulnerabilities of the network. This map represents the probability of detection for a constant false alarm rate or detection threshold. For example, the shaded gray area in Figure 1 shows a hypothetical detection region for the sensor network. Gaps in the map indicate vulnerabilities and are indicated by the striped regions. The MDU needs to fix these gaps by placing new sensors or moving previously placed sensors.

We can also plot detection probabilities as contours. This plot would be similar to an elevation contour map, but instead the contour lines would represent probabilities of detections instead of elevations. The required detection probability for the system is called the *critical PD*. Any gaps in the critical PD contour indicate vulnerabilities in the virtual perimeter.

† Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

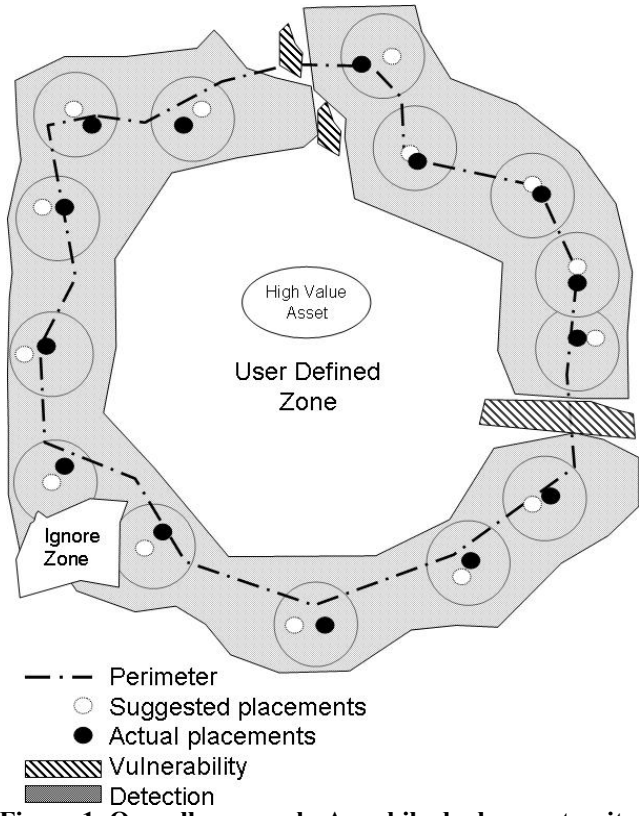

Figure 1. Overall approach. A mobile deployment unit places sensor pods close to suggested locations along a desired perimeter. System automatically estimates the probability of detection region and vulnerabilities.

3. Footstep detection

While our approach should work with any detection algorithm, we use the kurtosis footstep detection [5] algorithm to demonstrate the approach. Figure 2 shows a state diagram of the footstep algorithm. In the first stage of the algorithm an estimate kurtosis or "spikes in the data" is performed on the incoming seismic data channels in the time-domain. For signals that represent a human footstep, this fourth order statistical moment produces high values. If the kurtosis estimate exceeds a normalized threshold, the algorithm would then look for additional occurrences over time and verify the persistence of the feature. This persistence check removes transients from consideration that could potentially set the detector off and cause false alarms. This check also allows the detector to use subsequent information over time to integrate its belief of a footstep decision.

The persistence criterion rejects many of the signals that could potentially trigger a false alarm in the system. However, this first stage of the algorithm does poorly at rejecting rain and animals. These false alarms signatures have kurtosis similar to human footsteps and are persistent. Therefore a second stage algorithm uses spectral gating to estimate the gait or walk frequency of the target to further refine its decision. This gait frequency estimate is used to then classify whether or not the signal within the view of the sensor is a human footstep or another target class. This spectral feature estimates the periodicity of the spikes in the data.

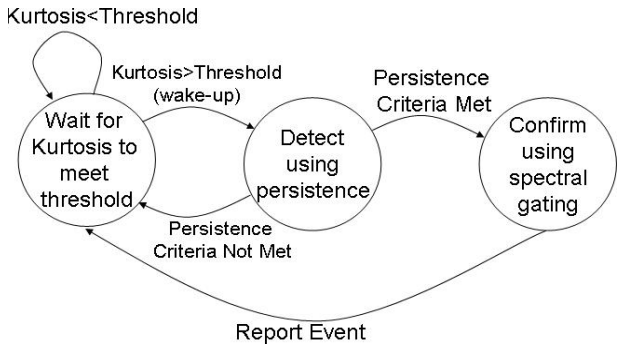

Figure 2. Footstep detection algorithm.

4. Sensor probability of detection function

For each sensor we estimate a sensor probability of detection function (SPDF). This function describes the probability of detection as a function of position relative to the sensor. Currently, we assume homogeneous geography and soil conditions which give an isotropic sensor response. Thus the PD varies only with range to the sensor. In future work, we plan to extend our approach to produce a nonisotropic SPDF. Unless explicitly noted, any mention to an SPDF in the rest of this paper will refer to the isotropic version. Figure 3 shows an example of an SPDF. The figure shows that as the range from the sensor increases the PD of the footstep detection algorithm decreases.

Figure 4 shows a block diagram illustrating the approach for estimating an SPDF. The MDU GPS produces location information (X,Y) in lat/long along with a timestamp T_m. The sensor detections $D=\{d_1 \ldots d_n\}$ are recorded as 0/1 along with the time of detection T_d. The first step in the process is to interpolate the MDU detections to match the time of the sensor detections. After aligning the MDU position and sensor pod detections in time we can estimate the range of the MDU to the pod. Thus the SPDF estimator uses a sequence of ranges $R=\{r_1 \ldots r_n\}$ of the MDU from the pod and a list of 0/1 detections $D=\{d_1 \ldots d_n\}$ from the

footstep algorithm. The result is the SPDF $\theta(r)$, where r represents the ranges of interest.

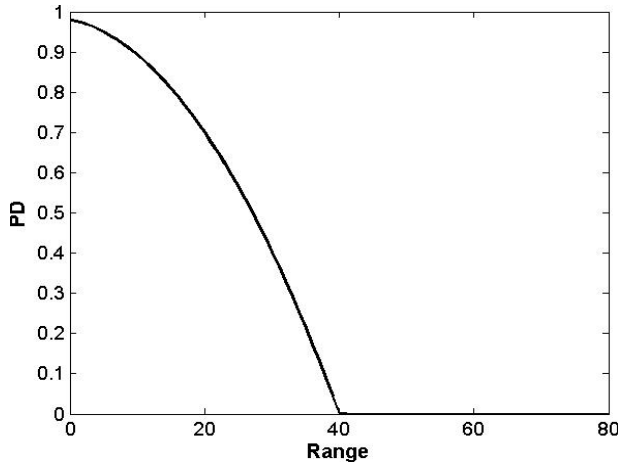

Figure 3. Example of a hypothetical SPDF.

4.1. Maximum likelihood SPDF estimation

From the number of detections $s(r)$ at range r and the number of nondetections $u(r)$ we can compute the probability of detection $\theta(r)$ as follows:

$$\theta(r) = \frac{s(r)}{s(r)+u(r)}. \quad \textbf{(1)}$$

Here the numerator represents the number of detection successes and the denominator represents the number of trials.

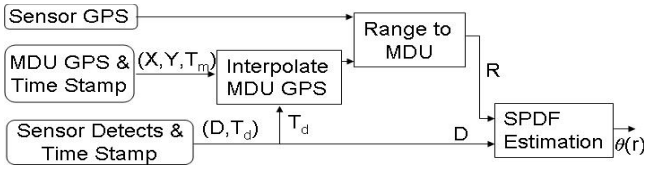

Figure 4. SPDF Approach.

One method for estimating $s(r)$ and $u(r)$ is to use a histogram approach. Thus we can partition the range of interest into m bins and count the number of detections in each bin to get $\hat{s}(r)$:

$$\hat{s}(r) = \sum_{i=1}^{n} I(c(r) - r_i; h) d_i, \quad \textbf{(2)}$$

where $I(z;h)$ is an indicator function on the interval $[-h,h]$ and $c(r)$ is a function that gives the bin center closest to r. Similarly, we can get $\hat{u}(r)$:

$$\hat{u}(r) = \sum_{i=1}^{n} I(c(r) - r_i; h)(1 - d_i). \quad \textbf{(3)}$$

Using

$$\hat{\theta}(r) = \frac{\hat{s}(r)}{\hat{s}(r)+\hat{u}(r)} \quad \textbf{(4)}$$

gives a maximum likelihood estimate of the probability of detection for range r.

4.2. Parzen kernel SPDF estimation

To get a smooth estimate we use the Parzen kernel approach [6] [7]. Here we use a kernel function rather than a box as the basic building blocks and center these smooth functions over every observation. The estimates for $\hat{s}(r)$ and $\hat{u}(r)$ become

$$\hat{s}(r) = \sum_{i=1}^{n} w(r - r_i; h_s(n)) d_i, \quad \textbf{(5)}$$

and

$$\hat{u}(r) = \sum_{i=1}^{n} w(r - r_i; h_u(n))(1 - d_i), \quad \textbf{(6)}$$

where $w(z;h)$ is a kernel function with an integral of one and bandwidth h. For the kernel estimate to converge to the actual function, we require

$$\lim_{n\to\infty} h(n) = 0 \text{ and } \lim_{n\to\infty} nh(n) = \infty \quad \textbf{(7)}$$

A common choice for the kernel function is a Gaussian

$$w(z;h) = \frac{1}{h\sqrt{2\pi}} e^{-z^2/2h^2}, \quad \textbf{(8)}$$

and for the bandwidth

$$h(n) = bn^{-\gamma}, 0 < \gamma < 1. \quad \textbf{(9)}$$

It has been shown that $\gamma = 1/5$ produces the best approximation in the sense of mean integrate squared error (MISE) [1]. Selection of the parameter b is more difficult, since we need to know the form of the function we are approximating. A conservative approach assumes the function is Gaussian and minimizing the MISE gives $b = \sigma(4/3)^{1/5}$ where σ represents the standard deviation of the range data [1]. A robust estimate for σ that accommodates outliers is given by the following median absolute deviation estimator:

$$\tilde{\sigma} = median\{[R - \tilde{\mu}_R]\}/0.6745, \quad \textbf{(10)}$$

where $\tilde{\mu}_R$ represents the median of the samples in R. Note that we use a different b for h_s and h_u.

4.2.1 Parzen kernel results

Figure 5 shows a simulated MDU walking in a spiral around a sensor pod. The detections and nondetections are generated using the SPDF shown in Figure 3 and are represented as stars and circles, respectively. Figure 6

shows the results of using Parzen kernels to estimate the SPDF. The dotted line in each graph represents the true SPDF. The legend indicates the total number of detections and nondetections or number of trials along the spiral path that were used to estimate an SPDF. As the number of trials increase the Parzen estimator converges closer and closer to the true SPDF. Note the sharp discontinuity at the max range of 40 is difficult for the Parzen estimator to approximate.

4.3. Bayesian SPDF estimation

To incorporate prior knowledge about the SPDF we use a Bayesian estimation approach. To simplify notation we will implicitly assume that θ, s, and u are a function of r. Recall that s represents the number of "successes" and the number of "tries" is represented by $s+u$. The conditional probability function for s given the probability of detection θ is a binomial distribution:

$$f(s \mid \theta) = \binom{s+u}{s} \theta^s (1-\theta)^u . \quad \textbf{(11)}$$

Here we hold θ fixed and examine the probability distribution of s over its possible values.

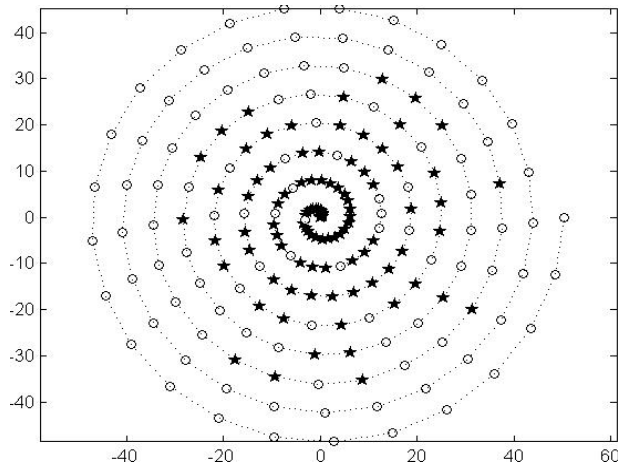

Figure 5. Simulated MDU walking in a spiral. Stars indicate detections and circles represent nondetections.

To use Bayes' theorem, we need a prior distribution $g(\theta)$ that gives our belief about the possible values of the parameter θ before gathering any data. The posterior distribution is proportional to the prior times the likelihood:

$$g(\theta \mid d) \propto g(\theta) f(s \mid \theta) . \quad \textbf{(12)}$$

The conjugate prior for the binomial is the $beta(a,b)$ density function:

$$g(\theta) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1} (1-\theta)^{b-1} . \quad \textbf{(13)}$$

where a and b are the shape parameters. The resulting posterior for our problem is:

$$g(\theta \mid d) = c\,\theta^{s+a-1} (1-\theta)^{u+b-1} . \quad \textbf{(14)}$$

where

$$\begin{aligned} c &= \int_0^1 g(\theta(r)) f(s \mid \theta) d\theta \\ &= \frac{\Gamma(s+u+a+b)}{\Gamma(s+a)\Gamma(u+b)} \end{aligned} . \quad \textbf{(15)}$$

The posterior is a beta distribution with shape parameters

$$a' = s+a \text{ and } b' = u+b . \quad \textbf{(16)}$$

The Bayes' solution is the mean of the conditional distribution of posterior for θ:

$$\tilde{\theta} = \frac{a+s}{a+b+s+u} . \quad \textbf{(17)}$$

The Bayes' solution can be rewritten as:

$$\tilde{\theta} = \left(\frac{s+u}{a+b+s+u} \right) \frac{s}{s+u} + \left(\frac{a+b}{a+b+s+u} \right) \frac{a}{a+b} . \quad \textbf{(18)}$$

Using (1) and making the dependence on r explicit:

$$\tilde{\theta}(r) = (1-\alpha(r))\hat{\theta}(r) + \alpha(r)\bar{\theta}(r) , \quad \textbf{(19)}$$

where

$$\alpha(r) = \frac{a(r)+b(r)}{a(r)+b(r)+s(r)+u(r)} , \quad \textbf{(20)}$$

and

$$\bar{\theta}(r) = \frac{a(r)}{a(r)+b(r)} . \quad \textbf{(21)}$$

For range r, $\bar{\theta}(r)$ is the mean of prior and the parameter $\alpha(r)$ represents a weighting factor with a value between 0 and 1. Thus the Bayes' estimate $\tilde{\theta}(r)$ is a weighted average of the maximum likelihood $\hat{\theta}(r)$ (estimated using the Parzen kernel approach) and the mean of the prior distribution $\bar{\theta}$. If $a=b=0$ then all the weight is placed on the maximum likelihood estimate $\hat{\theta}(r)$. If $s(r)+u(r)=0$ then there are no tries at range r and all the weight is placed on the mean of the prior distribution $\bar{\theta}(r)$. The sum $M = a(r)+b(r)$ represents the weight of the prior evidence and is a user supplied parameter.

4.3.1 SPDF Bayesian prior

The prior SPDF determines the mean of the prior. It represents our belief of the SPDF before collecting any data. Figure 7 shows different examples of possible prior SPDF's. The three things they have in common are:

1) A maximum PD at a range of zero,
2) The PD decreases as the range increases, and

3) The maximum range of the sensor pod.

From the prior SPDF $\bar{\theta}(r)$ and the weight of the prior M we can determine the shape parameters $a(r)$ and $b(r)$. From $\bar{\theta}(r) = a(r)/[a(r)+b(r)]$ and $M = a(r)+b(r)$ we get:

$$a(r) = M\bar{\theta}(r) \text{ and } b(r) = M - a(r) . \quad \textbf{(22)}$$

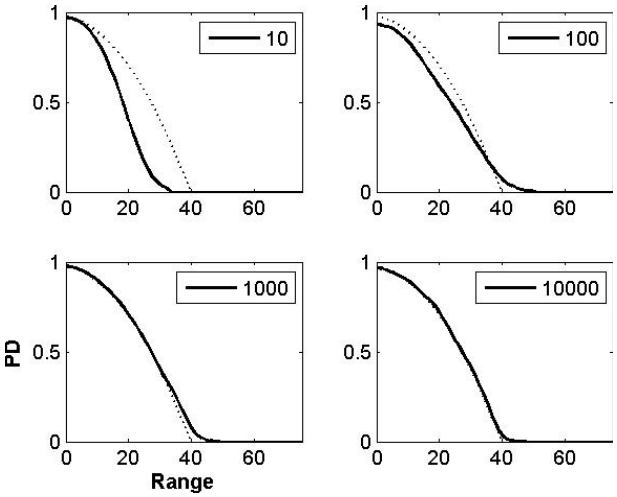

Figure 6. Parzen SPDF Estimation using simulated

4.3.2 SPDF Bayesian confidence interval

The posterior distribution $g(\theta \mid d)$ is $beta(a', b')$ where the shape parameters are given by (16). An equal tail area 95% confidence interval $[\theta_l, \theta_u]$ for θ can be found by selecting θ_l at the 2.5th percentile and selecting θ_u at the 97.5th percentile. The difference $\Delta\theta = \theta_u - \theta_l$ represents the uncertainty of the estimate. By collecting enough samples to make $\Delta\theta$ small allows us to control the credibility of the estimate.

4.3.3 SPDF Bayesian Results

Figure 8 shows results using a Bayes' SPDF estimator and a linear prior SPDF for *M=2*. The legend shows the total number of trials that were used to estimate the SPDF. For a small number of trials most of the weight is on the prior linear SPDF. As the number of trials increase the collected data takes over and the SPDF starts converging to the true SPDF.

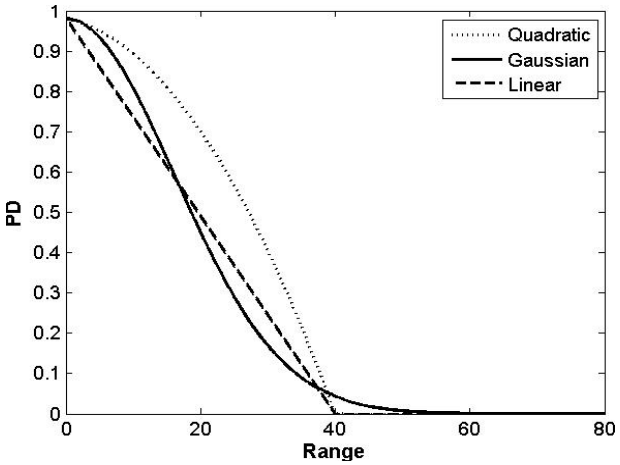

Figure 7. Example prior SPDF's.

4.4. Recursive SPDF estimation

We would like to estimate a detection map as we get positions and detections from the MDU. Thus after receiving a buffer of observations, we want to throw the observations away and not have to store them in memory. Unfortunately, we need to know the number of detections in order to set the Parzen bandwidth parameter h. An alternative is to use an estimator where h changes as the number of samples increase. The following estimator has been shown to converge [8][9][10] for reasonable kernels and as long as the properties in (7) are met:

$$f_n^*(r) = \sum_{i=1}^{n} w(r - r_i; h(i)) . \quad \textbf{(23)}$$

This can be rewritten recursively as

$$f_n^*(r) = f_{n-1}^*(r) + w(r - r_n; h(n)) . \quad \textbf{(24)}$$

Thus the maximum likelihood estimate after n observations becomes:

$$\theta_n^*(r) = \frac{s_n^*(r)}{s_n^*(r) + u_n^*(r)} . \quad \textbf{(25)}$$

where

$$s_n^*(r) = s_{n-1}^*(r) + w(r - r_n; h_s(n)) d_n , \quad \textbf{(26)}$$

and

$$u_n^*(r) = u_{n-1}^*(r) + w(r - r_n; h_u(n))(1 - d_n) , \quad \textbf{(27)}$$

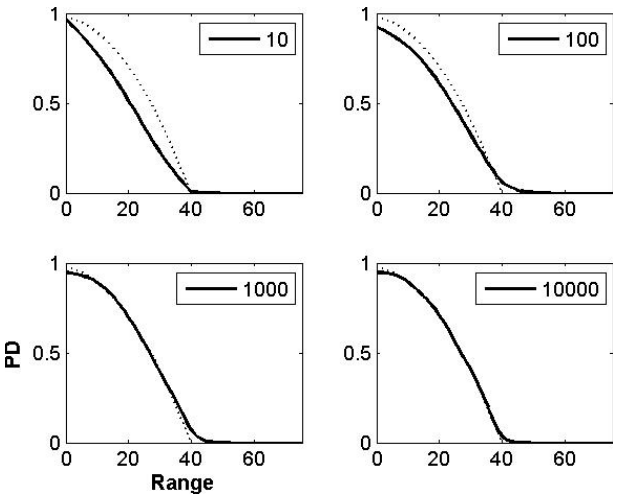

Figure 8. Bayesian results for *M=2* using simulated

Figure 9 shows the results of estimating an SPDF using the recursive Parzen approach. The legend indicates the total number of trials. Results are similar to the standard Parzens approach with a fixed bandwidth parameter.

5. Estimating a network of sensors PD map

The detection map is represented as a 2D image array with each pixel mapping to area on the ground and containing an estimate of the PD for the sensor network. We assume that the network reports a detection if any of the sensor pods have a detection. Let $\theta_p^*(r)$ represent the current SPDF for pod p at range r. Let (x_p, y_p) be the position of each pod and (x_i, y_i) be the center position of the i^{th} pixel of interest. We assume all positions are in some common world coordinate system. Let $\delta_p(i)$ represent the distance of i^{th} pixel (x_i, y_i) to all the pod positions (x_p, y_p). The probability that at least one pod detects a footstep is one minus the probability that they all don't detect a footstep. Thus the probability of detection at pixel (x_i, y_i) is:

$$\pi_i = 1 - \prod_{j=1}^{P} \left[1 - \theta_j^*(\delta_j(i))\right], \qquad \textbf{(28)}$$

where P denotes the total number of pods in operation. By computing π_i for every pixel in the detection map image we can get a detection map for the area of interest.

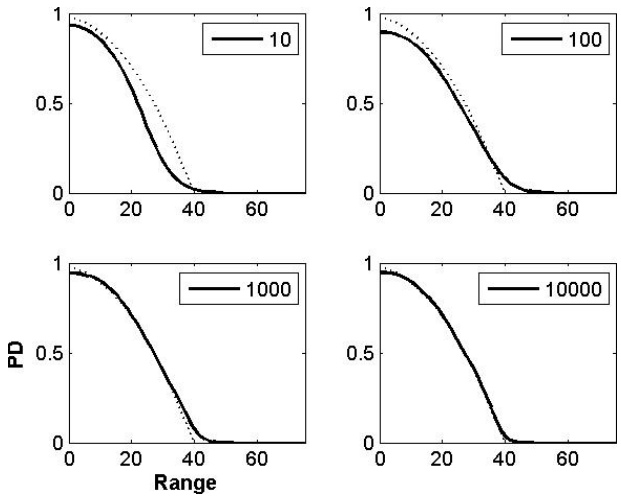

Figure 9. Recursive Parzen results for simulated data.

6. Processing the sensor data

To access the sensor data, the map client subscribes to specific data and message streams. Table 1 shows the four messages we are interested in. From these messages we can use the Bayes framework to update the SPDF and calculate a new estimated PD map.

Table 1. Sensor message types and their description.

Message Type	Description
1002	MDU or sensor pod initialized
1010	Footstep start detection (high state)
1011	Footstep stop detection (low state)
1018	MDU GPS reading

Figure 10 shows the processing of the sensor data. For a message type of 1002, the map client initializes a data structure for an MDU or sensor pod. The footstep detections are run length encoded, so only a footstep high state (detection) or low state (nondetection) is sent. Initially we assume no footstep detections. Anytime we get 1010 or 1011 we add the appropriate state and time stamp to the sensor data structure. A 1018 indicates a new MDU GPS location. Once we get two MDU locations with their respective time stamps, we can start interpolating to find MDU locations between the two time stamps. The 1018 message drives the update of the SPDF. This also triggers an update of the current footstep decision states and time stamps of all sensor data structures.

6.1. Results using actual sensor data

Figures 11 and 12 show the results of testing our algorithms using an existing network that currently exists

in the field. The network has eight sensor pods describing a rectangular area on level terrain. Four sensor pods are in the corners of the rectangle and the remaining four pods are in the middle of the sides of the rectangle. An MDU walked around visiting each pod as if he were setting up the network. Figure 12 shows the SPDF's that were estimated for each pod. Using the location of the pods and the generated SPDF's we produced the contour map in Figure 11. The filled circles show the location of the sensor pods, and the star shows the location of the high value assest. The line with the dot markers shows the path of the MDU. The contour lines represent the PD of the footstep detector for the entire network. The contour lines represent the mean of Bayes' estimate of PD. From the detection map we can see two gaps for a PD of 0.95, but the 0.90 PD line forms a virtual perimeter.

7. Conclusion and Future Work

For a seismic footstep detection algorithm, we have developed algorithms to learn a sensor's probability of detection. The algorithms are based on a Parzen kernel estimator to produce smooth estimates. With enough samples the Parzen estimator converges to the true SPDF function. We put the Parzen estimator in a Bayesian framework that allows us to incorporate prior information of the SPDF function and estimate confidence intervals. A recursive Parzen estimator allows us to build an SPDF as footstep algorithm results are received. Thus we don't have to buffer all the detection results to produce an SPDF. By combining the SPDF's from multiple sensors in a network we can produce a detection map. We have collected data using an existing network and produced a detection map of the results.

For future work, we plan on gathering data in varying terrain, such as on hillsides and in ravines. We are also interested in extending this approach to other perimeter security systems such as ground based radar. Other extensions are to modify the current approach to handle nonisotropic SPDF's and multiple MDU's. For the nonisotropic condition we assume PD changes not with range, but also azimuth. We also need to develop algorithms to automatically find gaps in the detection map, and suggest new locations of sensors or adding sensors to fill the gaps. Other improvements would have the system automatically request the MDU to walk around in certain areas to reduce the uncertainty in the confidence interval. To validate our approach we plan on determining the 0.9 PD line using the manual approach discussed in Section 1.

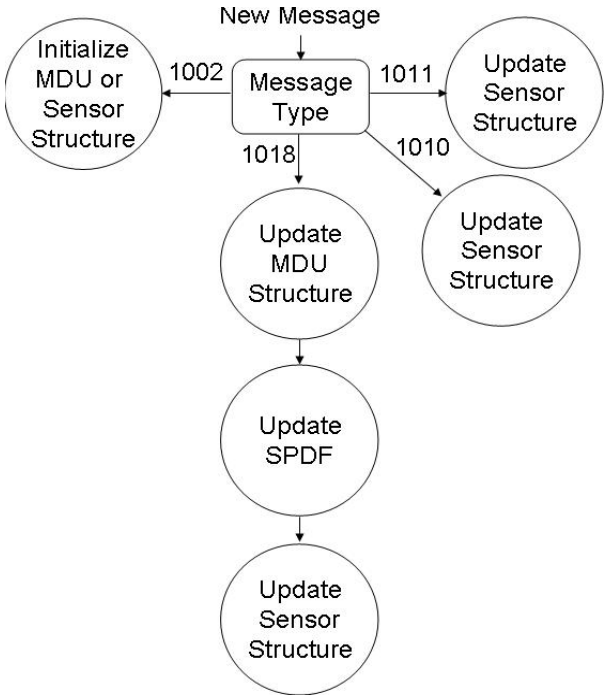

Figure 10. The processing of sensor data.

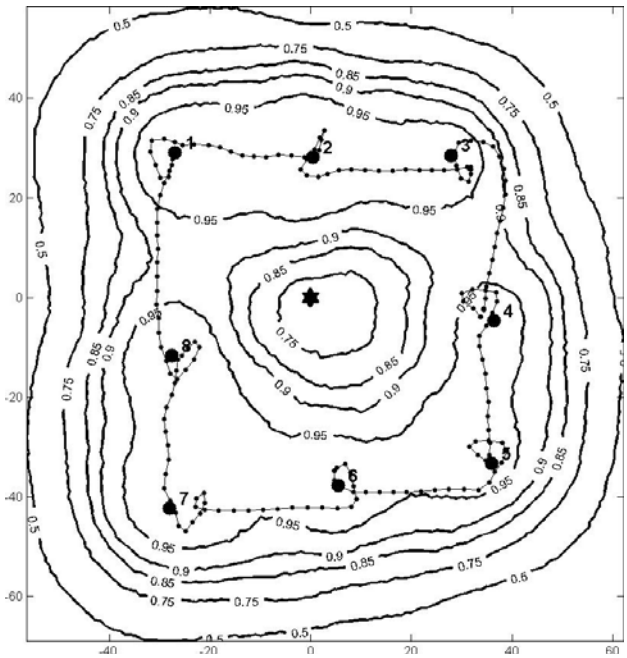

Figure 11. The estimate PD map for a deployed sensor

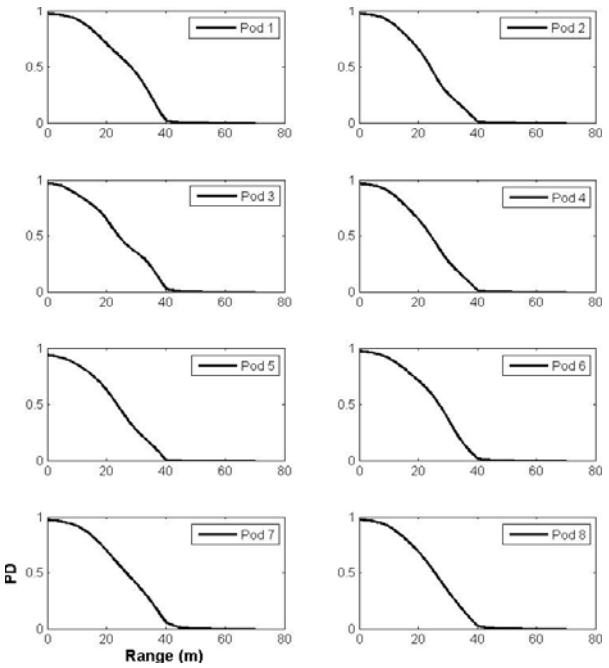

Figure 12. The SPDF's for a deployed sensor network.

References

[1] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*, Clarendon Press, 1997.

[2] J. R. Cooke, M. T. Lee, and J. P. Vanderdeck, "Binomial reliability table, (lower confidence limits for the binomial distribution)," National Technical Information Service, U. S. Department of Commerce, AD 444-344, 1964.

[3] M. L. Garcia, *The Design and Evaluation of Physical Protection Systems*, Butterworth-Heinemann, Boston, MA, 2001.

[4] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Macmillan, 1978.

[5] A. Pakhomov, A. Sicignano, M. Sandy, and T. Goldburt, "Seismic footstep signal characterization," Proceedings of SPIE, 5071: 297-304, 2003.

[6] E. Parzen, "On estimation of probability density function and mode," *Annals of Mathematical Statistics*, 33:1065-1076, 1962.

[7] M. Rosenblatt, "Remarks on some nonparameteric density function and mode," *Annals of Mathematical Statistics*, 27:832-837, 1956.

[8] E. J. Wegman and H. I. Davies, "Remarks on some recursive estimators of a probability density," The Annals of Statistics, 7(2):316-327, 1979.

[9] C. T. Wolverton and T. J. Wagner, "Asymptotically optimal discrminant functions for pattern classification," *IEEE Transactions on Information Theory*, IT-15:258-265, 1969.

[10] H. Yamato, "Sequential estimation of a continuous probability density function and mode," *Bulletin of Mathematical Statistics*, 14:1-12, 1971.