Physical Computing for Self-adaptation of Spatial-temporal Networked Systems in Dynamic Environments

Shashi Phoha
Pennsylvania State University
Professor of Electrical and Computer Engineering
Director, Information Science and Technology Division
Applied Research Laboratory
204 Applied Science Building
University Park, PA. 16802, USA
Email: sxp26@psu.edu

Abstract

Physical computing involves the autonomous prognosis and adaptation of the physical state of a system in a dynamic environment through distributed sensing, computation, and fusion-driven self-organization. It incorporates distributed computation for both local self-awareness and situational awareness of the spatial-temporal processes through in-situ learning to discern the current state of the system, collaborative situation assessment, prediction and adaptive control. Significant impediments to physical computing in complex dynamic environments are due to high dimensionality, nonlinear interaction of parameters, lack of adequate environmental models, non-stationarity and complex interactive dynamics. This paper addresses these issues by developing constructive methods derived from statistical mechanics to formulate a higher level alphabet and minimal ontology that captures the essential physics of the distributed operational context as observed in asynchronous streams of sensor data. Formal semantics are then used to develop distributed computational algorithms to *discover* atomic causal patterns, represented as finite state automata, that capture the generating dynamics of the system. Under Markovian assumptions, the algorithm achieves maximal compression while preserving statistical predictability of system states. Hence, the causal patterns can be discovered in-situ in the observed data streams, and are not derived from a pre-selected model. Since the causal patterns preserve the statistical characteristics of quasi-stationary processes, they can be used for situation awareness and adaptive control.

Keywords: Statistical mechanics, machine learning, adaptive control, model discovery, spatial-temporal dynamics, information fusion, undersea sensor network, physical computing, mine countermeasures.

1 Introduction

Harnessing the collaborative power of networks of electronic devices with embedded sensing, computation and actuation capabilities is a new frontier in technology that holds the promise of unprecedented levels of dependable autonomy and control in the execution of complex dynamic missions. As a network of physically embodied sensing agents that interact with the sensed phenomena, the cognitive abilities of the nodes are constrained by the laws of physics. As a team of goal directed agents, they must jointly evolve in real world environments to optimize operational effectiveness and adapt to physical change. This paper deals with the computational intractability, physical limitations, and uncertain nonlinear dynamics that today inhibit in-situ adaptation of autonomous missions such as multi-asset undersea mine-hunting or anti-submarine warfare.

The concept of an artificial language is introduced that expresses the interactive dynamics of complex physical processes that characterize the causal behaviors of the networked multi-agent system with a view to distributed computation, communication, and in-situ control of the system in its operational environment. In the absence of a known model of the generating dynamics, the current state of the system is divined from the asynchronous multi-sensor data streams. Spatial-temporal change in the system is statistically captured as a change in the probability distribution of transitions of system states. Nonlinear filtering algorithms for the discovery of causal patterns of state transitions in the evolving system are presented. The discovery of these patterns is important for understanding the physical processes that generate them. Two steps are involved: symbolization of the evolving streams of sensor data by maximal entropy partitioning and dynamic filtering of the symbol stream to generate probabilistic finite state automata. Symbolization defines an alphabet that represents the multivariate system as a univariate symbol sequence. Causal patterns in this symbol sequence are discovered via nonlinear dynamic filtering algorithms and state-machine construction algorithms that represent system state transitions as probabilistic finite state automata. By thus fusing atomic symbols of causal knowledge embedded in the sensory observations, this paper formulates an interactive computational model of the evolutionary physics captured via the distributed sensor suite for adaptive computation. The causal patterns represent words in a formal machine language. The formal language thus generated, semantically represents the discovered model of the evolving system and can be used for machine learning, semantic reasoning for collaborative pattern discovery, and distributed control [1]. Under regularity conditions, it preserves the full predictive power of the original data streams and is computationally faster than classical methods (Bayesian, Hidden Markov Machines, Particle Filtering, etc.) [2]. Therefore, it enables in-situ discovery and prediction of spatial-temporal patterns in multi-sensor data streams. Hence, the dynamic control of the system can be adapted in-situ to its unmodeled physical evolution.

This paper formulates a rigorous scientific basis for constructing an alphabet and a set of atomic patterns (words) from this alphabet that preserve the statistical properties of the spatial-temporal multivariate data streams generated by the complex dynamic system in its operational environment. The formal language thus created captures the generating dynamics of the system and can be used for predictions of future behavior of the system. Section 2 of the paper summarizes the relevant background in Statistical Mechanics. Section 3 contains algorithms for semantic modeling to discover the generating formal language from asynchronous multivariate streams of sensor data. Section 4 delineates properties of the machine language and compares it to classical methods. Section 5 presents architectures and algorithms for distributed machine learning using this formal machine language. Section 6 illustrates the power of physical computing methods under resource constraints by analyzing side-scan sonar images taken by unmanned undersea vehicles for collaborative detection of undersea mines. Section 7 presents concepts of higher levels of machine semantics for semantic fusion and contextual decision-making. Section 8 introduces the concept of Artificial Languages and their broader impact on deep machine learning and distributed computation.

2 BACKGROUND

Statistical Mechanics provides an elaborate explanation of the emergence of macroscopic behavior in dynamical systems, where complexity accrues from the interactions of their microscopic constituents. The macroscopic and microscopic behavior is linked by statistical patterns that represent the probability distribution of the microstates. Concepts of statistical physics, originally developed to study collective properties of physical systems such as solids, liquids, and gases, have been extensively applied to investigate a diverse range of systems including chemical and biological structures such as colloids, emulsions, liquid crystals, complex fluids, polymers, bio-polymers and cellular, economical and sociological, ecological and a variety of network systems. The fundamental principles of equilibrium in statistical physics have been used to investigate stationary and quasi-stationary behavior of complex systems. This discipline is known as *thermodynamic formalism* of complex systems in the applied physics literature [3]. Such applications have been successful because it is possible to identify a hierarchically structured model based on the order parameter and complexity associated with each level of the hierarchy.

3 SEMANTIC MODELING AND PATTERN DISCOVERY

Sensors require physical interaction with the sensed phenomena and are not individually very reliable. Sensor signal deteriorates as the sensing distance increases. But in the vicinity of a stimulus, the sensor data is highly correlated, both spatially and temporally. Hence collaborative inference is needed for reliable identification and registration of spatio-temporal phenomena in complex dynamic missions. However, the nonlinear and non-stationary interactive dynamics of the observed parameters, noise and measurement inaccuracies, and uncontrollable exogenous environmental disturbances, make it impossible to apply fundamental laws of physics to achieve required modeling accuracy and precision. This section provides an alternative method of observation-based estimation of the generating dynamics of a complex system. This method, called *semantic modeling* consists of two steps: symbolization and nonlinear filtering. These two steps, described below, result in compressing the information contained in the ensemble of asynchronous data streams emanating from multiple sensors into identifiable macroscopic patterns of the system that represent the evolving characteristics of the underlying process dynamics of the system.

3.1 Symbolization

Symbolization is the process of discretizing the *n-dimensional* space of the time-series data generated by n individual sensors. This is often achieved by maximal entropy partitioning of the phase space and assigning a distinct symbol to each partition [1]. The set of symbols thus generated is called an alphabet for the dynamic system. As the system evolves in time, its trajectory is assumed to be circumscribed in a compact region of the n-dimensional phase space. Under specific stimuli, the trajectory travels through and touches various cells in the partition, generating the corresponding symbol from the alphabet. The generated multivariate asynchronous streams of sensor data can thus be encoded as a univariate sequence of symbols as shown in Fig. 1.

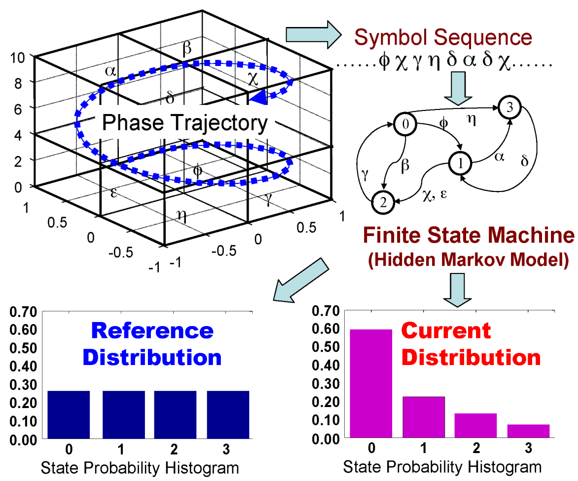

Figure 1: Semantic Representation of Sensor Data

The size δ of the cells of a partition and the cardinality of the alphabet are critical to the accuracy and complexity of semantic modeling and subsequent pattern discovery. The objective is to simultaneously capture the high fidelity dynamics of the system by choosing a small enough δ and keep the alphabet size small for controlling computational complexity. Maximal entropy partitioning has the benefit that partitions are coarser in regions of low data variability and finer in regions of high data variability. A phase space partition is called a *generating partition* of the phase space if every symbolic orbit uniquely identifies a system trajectory.

As an alternative to phase space partitioning, the time series data set $Y = \{\ldots, y_k, y_{k+1}, \ldots, y_0\}$ of sensor observations can be directly used for symbolic dynamic encoding. In this regard, several techniques have been reported in the literature that suggest appropriate transformation of the signal before employing the partitioning method for symbol generation to retrieve the relevant information from time series data. One such technique is based on *wavelet transform* of time series data, called Wavelet-Space Partitioning (WSP) [1] that is particularly effective with noisy data from high-dimensional dynamical systems. Another reported technique is Analytic-Signal-Space Partitioning (ASSP) [4] based on the Hilbert transform of time series data. The Hilbert transform has the added advantage that it yields an unambiguous definition of the symbol sequence in both space and time simultaneously and is robust to noise with relatively low computational cost.

3.2 Nonlinear Filtering and State Machine Construction Algorithms

The objective of nonlinear filtering of a symbol sequence is to encode small order representations of the semantic structure of recurrence in the sensed data. The mathematical framework in which the nonlinear filtering techniques are laid out is statistical inferencing based on the construction of Probabilistic Finite State Automata (PFSA) from a symbolized sequence of sensor data. In their survey paper, Angluin and Smith [5] define inductive inference as the process of hypothesizing a general rule from examples. The process of nonlinear filtering yields such a rule in the form of a PFSA which captures the internal structure of a black box that emits the symbolized sequence constructed from the sensor data.

Several algorithms for construction of PFSAs from a symbol sequence have been proposed in the literature. Initially, ϵ machine construction was proposed by Shalizi and Crutchfield [6]. This construction requires partitions of fixed length histories of a quasi-stationary process by defining equivalence classes of system trajectories that have a common morph,i.e. probability distribution of the next symbol. These equivalence classes define the causal states of the system. The evolution of the next symbol of the alphabet causes a transition from the current causal state of the system to another with probability $p(i, j)$. Together, the set S of causal states and the transition matrix P are called the ϵ machine generated by the ensemble of data. The Causal State Splitting Reconstruction (CSSR) algorithm described in [6] formulates a PFSA that represents the ϵ-machine from a given ensemble of data. However, this method is not suitable for in-situ adaptation because it requires the recomputation of the causal states after every evolution of a symbol. To make ϵ-machines useful for time critical physical computing, the D-Markov method of PFSA construction, detailed in [1], sustains the same finite state set of causal states for successive shifts of windows of a fixed length D representing the symbol subsequences generated during successive time epochs.

A finite state machine is constructed, where the states of the machine are defined corresponding to a given alphabet set Σ and window length D. The alphabet size $|\Sigma|$ is the total number of partition segments while the window length D is the length of consecutive symbol words, which are chosen as all possible words of length D from the symbol sequence. Each state belongs to an equivalence class of symbol words of length D or more, which is characterized by a word of length D at the leading edge. Therefore, the number n of such equivalence classes (i.e., states) is less than or equal to the total permutations of the alphabet symbols within words of length D. That is, $n \leq |\Sigma|^D$; some of the states may be forbidden with zero probability of occurrence. For example, if $\Sigma = \{0, 1\}$, i.e., $|\Sigma|$ = 2 and if D = 2, then the number of states is $n \leq |\Sigma|^D$ = 4; and the possible states are 00, 01, 10 and 11, as shown in Fig. 2

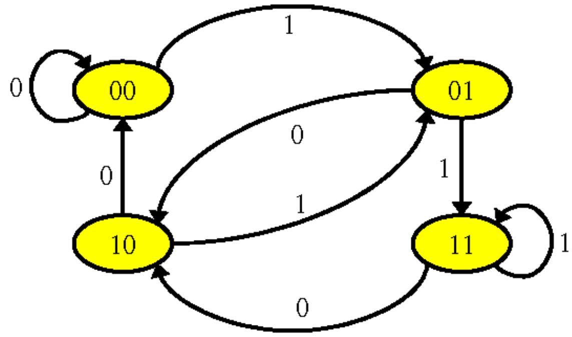

Figure 2: Example of finite state machine with $|D|$ = 2 and $\Sigma = \{0, 1\}$

The choice of $|\Sigma|$ and D depends on specific experiments, noise level and also the available computation power. A large alphabet may be noise-sensitive and a small alphabet could miss the details of signal dynamics. Similarly, while a larger value of D is more sensitive to signal distortion, it would create a much larger number of states requiring more computation power.

Using the symbol sequence generated from the time series data, the state machine is constructed on the principle of sliding block codes [7]. The window of length D on the symbol sequence $\ldots \sigma_{i_1}, \sigma_{i_2} \ldots \sigma_{i_k} \ldots$ is shifted to the right by one symbol, such that it retains the last (D-1) symbols of the previous state and appends it with the new symbol σ_{i_ℓ} at the end. The symbolic permutation in the current window gives rise to a new state. The machine constructed in this fashion is called the D-Markov machine [1], because of its Markov properties. A symbolic stationary process is called D-Markov if the probability of the next symbol depends only on the previous D symbols, i.e., $P\left(\sigma_{i_0} | \sigma_{i_{-1}} \ldots \sigma_{i_{-D}} \sigma_{i_{-D-1}} \ldots\right) = P\left(\sigma_{i_0} | \sigma_{i_{-1}} \ldots \sigma_{i_{-D}}\right)$.

The states of the machine are causal states and if maximal entropy partitioning is used to generate the symbol sequence then the probability for the system to be in any one of these states is uniform by construction. Hence, the reference distribution is uniform.

3.3 Pattern Discovery Algorithms

The PFSA generated using the D-Markov machine can be used in-situ to detect behavioral changes at a slower time scale.

The finite state machine constructed above has D-Markov properties because the probability of occurrence of symbol σ_{i_ℓ} on a particular state depends only on the configuration of that state, i.e., the previous D symbols. Once the alphabet size $|\Sigma|$ and word length D are determined at the nominal condition (i.e., time epoch t_0), they are kept constant for all slow time epochs $\{t_1, t_2, \ldots t_k \ldots\}$. That is, the partitioning and the state machine structure generated at the nominal condition serve as the reference frame for data analysis at subsequent slow time epochs.

The states of the machine are marked with the corresponding symbolic word permutation and the edges joining the states indicate the occurrence of a symbol σ_{i_ℓ} . The occurrence of a symbol at a state may keep the machine in the same state or move it to a new state. On a given symbol sequence $\ldots \sigma_{i_1}, \sigma_{i_2} \ldots \sigma_{i_\ell} \ldots$ generated from the time series data collected at a slow time epoch, a window of length D is moved by keeping a count of occurrences of word sequences $\sigma_{i_1} \ldots \sigma_{i_D} \sigma_{i_{D+1}}$ and $\sigma_{i_1} \ldots \sigma_{i_D}$ which are respectively denoted by $N(\sigma_{i_1} \ldots \sigma_{i_D} \sigma_{i_{D+1}})$ and $N(\sigma_{i_1} \ldots \sigma_{i_D})$. Note that if $N(\sigma_{i_1} \ldots \sigma_{i_D})$ = 0, then the state $q \equiv \sigma_{i_1} \ldots \sigma_{i_D} \in Q$ has zero probability of occurrence. For $N(\sigma_{i_1} \ldots \sigma_{i_D}) \neq 0$, the transitions probabilities are then obtained by these frequency counts as follows:

$$\pi_j k \equiv P(q_k | q_j) = \frac{P(q_k, q_j)}{P(q_j)} = \frac{P(\sigma_{i_1} \ldots \sigma_{i_D} \sigma)}{P(\sigma_{i_1} \ldots \sigma_{i_D})}$$

$$\Rightarrow \pi_j k \approx \frac{N(\sigma_{i_1} \ldots \sigma_{i_D} \sigma)}{N(\sigma_{i_1} \ldots \sigma_{i_D})}$$

where the corresponding states are denoted by $q_j \equiv \sigma_{i_1}, \sigma_{i_2} \ldots \sigma_{i_D}$ and $q_k \equiv \sigma_{i_2} \ldots \sigma_{i_D} \sigma$. The state transition matrix, $\Pi = [\pi]_{jk}$, satisfies the properties of a stochastic matrix, i.e., $\Sigma_k \pi_{jk} = 1 \forall j$.

As the system transitions from one state to another, the relative frequency of its visiting a particular causal state is updated resulting in a new frequency distribution F. If F is sufficiently distinguishable from the reference distribution, then the dynamic system has deviated from its original pattern and a new pattern is said to have been identified. Other more effective measures of semantic pattern deviation or semantic distance have been developed recently and are given in [8] and [9].

4 Artificial Language Properties

The above methods of formulating a mechanism that is most likely to generate a given sequence of sensor data are particularly effective for providing a succinct statistical characterization of the evolutionary physics inherent in the data for time critical situation awareness, fusion and prediction. It differs from classical methods by not requiring an initial guess at a prior distribution like the Bayesian approach which then proceeds to update the prior by maximizing the log likelihood ratio. In this approach, the model is discovered in the structure of the data stream and physical association of the identified states and graph connectivity with the causal semantics is a consequence of inherent recurrence properties of the data [10]. A generating partition provides maximal compression that preserves the predictability of the system state. Other methods like Hidden Markov Models (HMM), Neural Networks and Particle Filtering methods use a fixed topology and require more training than updating relative frequencies of state transitions in PFSAs. Experimental results in [11] show that learning a PFSA can be 10-100 times faster than that in HMM.

Another major difficulty in physical computing is the lack of availability of environmental models which influence system performance. By modeling the physical system as a multi-agent system, the environment is modeled as an agent that interacts with other agents. The exact dependencies of multiple agents are not explicitly modeled but are reflected in the ensemble of data generated in the operational environment. In the Bayesian approach, in contrast, adequate knowledge of the environment and its influence must be captured into a priori knowledge for making accurate predictions. In Bayesian networks, state machines are hand coded and a priori probability estimates are necessary. In-situ updates based on expectation maximization algorithms are not computationally feasible as opposed to semantic compression in the above mentioned physical computing algorithms. Hence, the formal machine language is superior for in-situ inductive inference-learning of a generating rule (represented as PFSA) from the observed data. It is also more noise tolerant due to inherent coarse graining and because the construction of the PFSA ignores sensing imperfections that may cause non-recurrent observed behavior. Further results comparing performance of above machine learning methods are given in [2].

5 Distributed Machine Learning

During the training phase of an experiment, a library may be developed off-line consisting of all significant causal patterns x_i of interest observed in data collected by a single sensor. As a distributed set of homogeneous sensors with limited on-board computation and communication capability collect data in a sensor field, each individual sensor can process its own data and formulate its own D-Markov machine for the observed semantic pattern. If this semantic pattern matches an element x_i from the on-board library then the sensor has identified a pattern of interest. The state machine structure generated off-line serves as the reference frame for data analysis and stays fixed from sensor to sensor and for subsequent slow time epochs. Since individual sensors achieve only limited confidence levels for classifying patterns of interest in noisy data, it may be necessary for them to collaborate with other sensors to achieve a high level of confidence. In this case the sensor needs to communicate only a code for the library pattern that it is observing, and form a cluster with nearest neighbors observing the same pattern within a small semantic distance. Communications are modeled as receive and send message passing events. Single-hop and multi-hop dynamic clustering algorithms based on semantic information are presented in [12] and [13]. Since, the representation of the emerging pattern differs only in relative frequency of transitions from the reference state space, data from multiple sensors can be semantically fused in-situ to classify observed patterns with high confidence as shown in the following section.

6 Undersea Mine-hunting Application

Recent developments in portable Unmanned undersea Vehicles (UUVs) with advanced on-board sensing devices like side scan sonar, and limited computational capability, are expected to transform undersea mine-countermeasure operations through adaptive collaboration in searching for and classifying mine-like-objects. The sonar sensors generate a monochromatic mapping of sea-bed bottom objects and vehicle induced image artifacts. Traditional approaches based on assigning a threshold of brightness to mapped features do not work well in distinguishing mine-like objects in a textured background. Bottom features themselves generate many false alarms. Calder et. al [14] have proposed improvements to traditional Bayesian detection methods using statistical and geometric properties of objects to reduce false alarms.

This section exemplifies the use of semantic analysis for pattern discovery in sonar images for detection of undersea mine-like objects. Data for training and validating the algorithms was obtained from the Naval Surface Warfare Center. Ground truth information is available for a set of images acquired by a UUV about the location of mines. The ensemble of data sets is partitioned into a training set of 100 images and a test set of 100 images for further analysis. Parameters required for the application of symbolic dynamics, such as alphabet size are chosen based on the ground-truth statistics. A geometric model, similar to [14] has been used for feature extraction to detect and classify mines in a sonar trace. This model is used in the training set to obtain the distributions of the various regions of a mine. A sequence of tests is determined to characterize a mine according to the identified distributions.

More details for applying the semantic analysis methods to two-dimensional images are given in [3] and [17]. Two important parameters need to be determined for successful partitioning of data. The first parameter is the alphabet size $|\Sigma|$ and the second is the vector of the partition segment boundaries. In mine detection, the essential robust features that need to be preserved are the bright reflections from the front of an object protruding above the sea bed and the long shadow that follows the object. Apart from this information, clutter is an important feature that has been used in the pattern analysis presented in this paper. It is observed that only three symbols are sufficient to characterize the features of interest. Fig. 3 shows histograms of density functions for mine, shadow, and clutter regions. The histograms depict the number of pixels vs pixel intensity for each region. These histograms are generated from the a priori known ground-truth information about the exact location of mines from the training data set of 100 sonar images. The histograms for the three regions are clearly separated. Therefore, a distinct single symbol in the alphabet Σ is assigned to each of the three features corresponding to mine, shadow and the clutter. The information gained by increasing the alphabet size is found to be too little to offset the additional computation. A point to note from the clutter histogram in Fig. 3 is that a small neighborhood around the mine region causes more bright spots from the mine to be relegated to the clutter region.

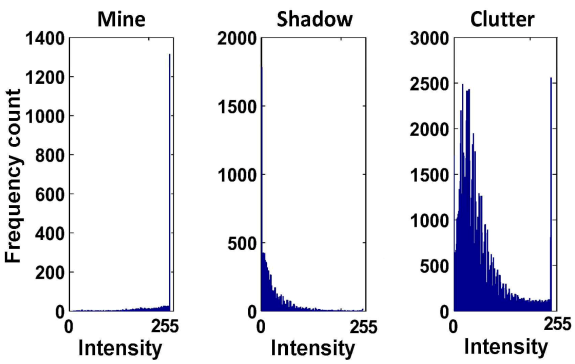

Figure 3: Histogram of sonar wave reflections for mines, shadows and clutter. The distribution of pixel intensities in the regions of mine-clutter and shadow-clutter showed similar trends, therefore, they have been combined together in a single histogram of clutter region.

The next important consideration is selection of segment boundaries of the partition. Traditional partitioning techniques (e.g., uniform partitioning and maximum entropy partitioning [15]) may not adequately capture the details of mine patterns; and conventional data-driven partitioning methods lead to a large alphabet size. This paper makes use of the statistical model information from the three histograms in Fig. 3. Partitioning is constructed by assigning symbol a to high intensity pixels ranging from 180 to 255 on the gray scale; similarly, symbol b is assigned to medium intensity pixels ranging from 56 to 179; and symbol c is assigned to low intensity pixels ranging from 0 to 55. As seen from Fig. 3, approximately 97% of the pixels in the mine histogram correspond to the symbol a (i.e., bright pixels); similarly, approximately 89% of the pixels in the shadow histogram correspond to the symbol c (i.e., dark pixels). A majority of the remaining (i.e., moderately dark) pixels corresponding to the symbol b belong to the clutter region. Thus, the entire image is symbolized and represented by a two dimensional array of symbols belonging to the set $\Sigma = \{a, b, c\}$. This partitioning scheme enables robust detection of mines with a high probability of detection and a very low probability of false alarms as discussed in the results section. Further, this symbolization greatly reduces the amount of memory required for any processing. The next subsection explains the method of feature extraction using the geometric model for mine detection.

A finite state Markov machine is now constructed, where the set of machine states is isomorphic to the symbol alphabet Σ [1]. As there are three symbols in Σ, the dimension of the state space is also 3. Symbol a corresponds to a bright pixel state in the sonar image, while symbol c corresponds to a dark pixel state that may be a part of a shadow. Symbol b denotes a mid-level pixel state.

A region β in the image space represents one of the three regions in the geometric model, i.e, the mine region, the shadow region, and the clutter region. The Markov assumption allows construction of the state probability vector $\mathbf{p}$ that is chosen to be the feature vector for a given bounded region β. The elements of $\mathbf{p} \triangleq [p_a \; p_b \; p_c]^T$ are calculated by frequency counting as:

$$p_i = Prob(\sigma_i \in \Sigma|\beta) \approx \frac{N(\sigma_i)}{\sum_{j\in\{a,b,c\}} N(\sigma_)}, i = a, b, c$$

where $N(\bullet)$ is the count of $\bullet$ in β. The construction of the feature vector p follows the sliding block code [16], where sliding of the geometric model along the sonar image is depicted in Fig. 4. For every pixel location (i, j), the geometric model is constructed around that pixel, such that (i, j) lies at the center point of the mine region. In this way, the feature vector is generated for each region of the geometric model. Therefore, for any pixel location (i, j) on the sonar image, the following four feature vectors (see Fig. 5) are generated.

1. $P^M(i,j) = \left[p_a^M(i,j)\, p_b^M(i,j)\, p_c^M(i,j)\right]^T$ is constructed from the **mine** region.
2. $P^S(i,j) = \left[p_a^S(i,j)\, p_b^S(i,j)\, p_c^S(i,j)\right]^T$ is constructed from the **shadow** region.
3. $P^{MC}(i,j) = \left[p_a^{MC}(i,j)\, p_b^{MC}(i,j)\, p_c^{MC}(i,j)\right]^T$ is constructed from the **clutter** region around the **mine**.
4. $P^{SC}(i,j) = \left[p_a^{SC}(i,j)\, p_b^{SC}(i,j)\, p_c^{SC}(i,j)\right]^T$ is constructed from the **clutter** region around the **shadow**.

A classifier was then constructed for identification of mines and non-mine-like objects based on mine classification methods used in [17].

A sliding window method is used to implement the geometric model and the classifier described above. For each pixel in the image, a model is constructed, as shown in Fig. 3. Assuming that the pixel under consideration is at the center of the mine region, the four feature vectors P^M, P^S, P^{MC} and P^{SC} are generated.

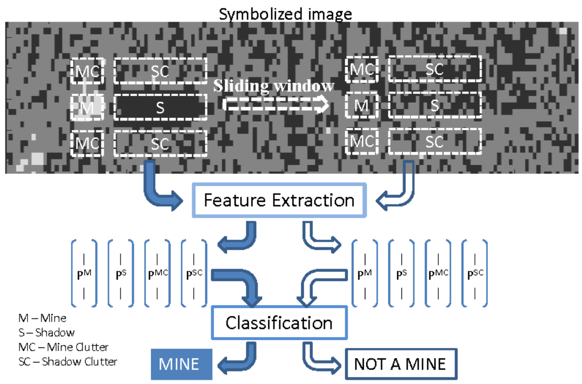

Figure 4: Symbolic dynamics-based mine detection

Then, the classification scheme is applied and a binary decision is made to determine whether the pixel location belongs to a potential mine. A binary number of 1 or 0 is assigned to each pixel of the image based on the classification as a mine or nonmine- like object, respectively.

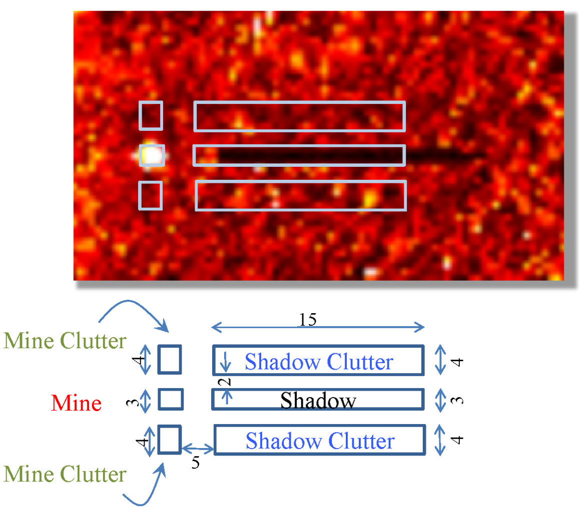

Figure 5: Geometric model for mine detection

To construct the receiver operating characteristics (*ROC*), a test data set consisting of 100 images is considered. These images consist of various textured backgrounds, with different types of sea-bed objects and vehicle-induced image artifacts. The images are in the range of 0 to 255 on the gray scale. Each of the four thresholds (i.e., λ_1, λ_2, λ_3, and λ_4) are varied from 0 to 1 in steps of 0.1 and the pattern analysis algorithm is executed over the entire set of test data. The number of false alarms, and the number of correct detections are counted in each threshold parameter combination. The *ROC* plot is constructed by joining the outermost points on the plot of probability of detection (P_D) versus false alarm rate (*FAR*) per image as shown in Fig. 6. For subsequent analysis, the chosen thresholds (λ_1 = 0.8, λ_2 = 0.7, λ_3 = 0.4 and λ_4 = 0.4) gives a probability of detection of 92% and 1.5 false alarms per image (or $900m^2$).

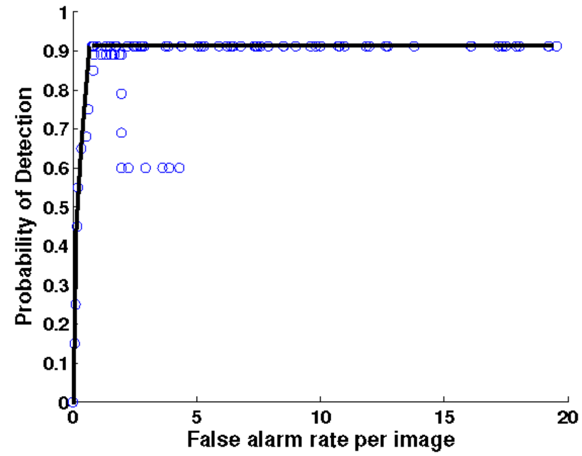

Figure 6: Receiver Operating Characteristics for Symbolic Dynamics based detection

This section presents the results generated upon execution of the pattern analysis algorithm on one hundred images from the test data set that is different from the training data set used to generate the partition.

Four representative images are shown in Fig. 7, each showing different levels of background noise, and sea-bed clutter. A representative set of threshold values is chosen to yield a high detection probability with an acceptable false alarm rate. An appropriate operating point is chosen on the *ROC* curve based on the premise that a missed detection cost is much higher than a false alarm. Tests show that the algorithm is capable of detecting mines in a high concentration of sea-bed clutter, including mines buried under vehicular artifacts. The algorithm has been executed on the entire set of test data with the same values of representative thresholds. With alternative choices of operating points on the *ROC* curves, the mine detection probability can be increased at the expense of increased false alarm rate as a trade-off between Type I and Type II errors.

The major advantages of the proposed pattern analysis algorithm for underwater mine detection are delineated below.

1. The algorithm is computationally efficient in terms of execution time and memory requirements as a consequence of a small alphabet and a small number of states in the Markov machine of the algorithm. As such the entire algorithm can be programmed and powered on a small microprocessor on-board a *UUV*.

2. In contrast to traditional Bayesian methods such as the likelihood-ratio-test, the symbolic pattern anal-

ysis does not require *a priori* knowledge of probability distribution for characterizing mines and non-mine-like objects. Specifically, the proposed algorithm is robust even if the unknown distributions are multi-modal.

7 Higher Level Machine Semantics and Information Context

The above method of physical computing has an important characteristic that makes it suitable for higher levels of situation awareness and higher levels of formal machine languages for command and control of distributed complex dynamic missions. Sensor data streams entail complex and dynamic dependencies and statistical patterns due to operational or environmental stimuli which can be exploited at higher levels of decision-making. Multiple decision-making contexts with differing and overlapping information semantics, spatial scales and temporal assumptions, need to capture the relevant dependencies embedded in the data streams. The physical computing framework defined in this paper can be naturally extended to multiple levels of hierarchy in both space and time for machine learning and decision-making. The multi-scale dynamics at the fast time scale and local interactions can be encoded through symbolization and causal pattern discovery methods described above, yielding a library of PFSAs. These PFSAs themselves form a higher level alphabet for encoding the evolutionary interactive dynamics of multiple sensor nodes in a region at the next level of fusion, decision and control, as shown in Fig. 8. Of course, the semantics at each level of the hierarchy represent abstract events that capture the decision-making structure at higher levels. Thus, each element of the higher level defines a context in which the lower level must operate.

8 Artificial Languages and Machine Collaboration

The process of learning the evolving situational contexts from spatio-temporal data streams is that of generating higher level formal language semantics that capture the structural dynamics and statistical predictability embedded in the original data. Conceptually, this process is the inverse process of generating a high level instruction for a computer to execute it at the machine level. Hence, physical computing reverses the role of computation in relating function to machine execution. Whereas a compiler requires the specification of semantics (executable instruction) to manipulate data, physical computing infers causal semantics from multi-scale data observable in physical systems. The operational contexts thus generated represent distributed machine cognition of higher level function

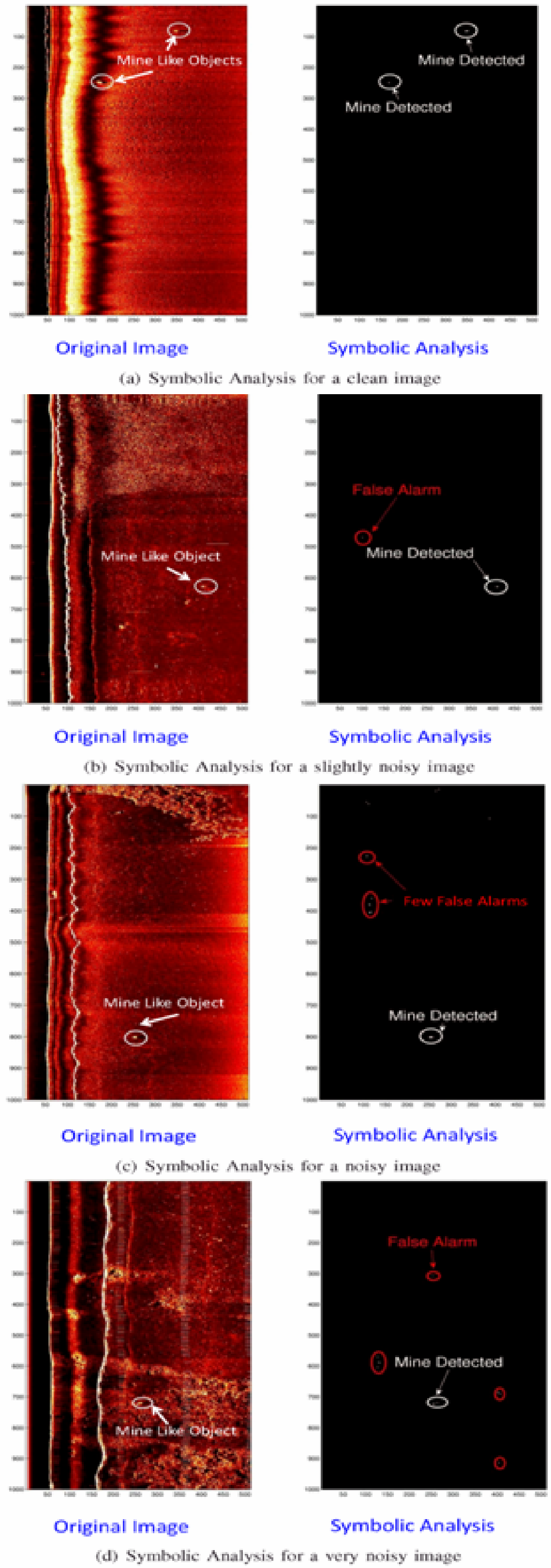

Figure 7: Representative images from test data set

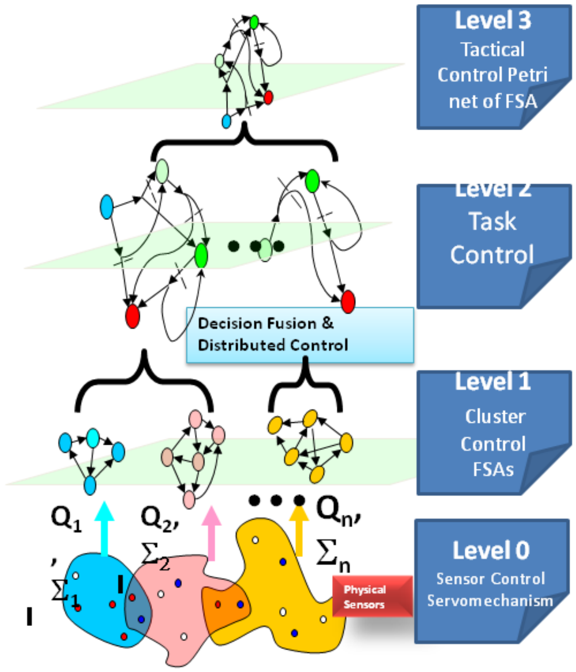

Figure 8: Multi-Layered Spatio-temporal Semantics

that generates the operational dynamics. If these higher level semantics are projected onto events of human cognition during a training phase, then distributed control algorithms can be devised for in-situ automated control of complex missions embedded in the evolving physics of a physical system. One such language is C3L- Control, Communications and Computation Language for collaboration of multiple devices. The syntax and semantics of this language are presented in [18] and updated in [19] a compiler development for executing mission scripts in this language is presented in [20].

Acknowledgement

This material is based upon work supported by the U.S. Army Research Laboratory and the U.S. Army Research Office (ARO) under Grant No. W911NF-07-1-0376 and by the Office of Naval Research under grant N00014-07-1-0288. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] A. Ray, “Symbolic dynamic analysis of complex systems for anomaly detection,” *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.

[2] C. Rao, A. Ray, S. Sarkar, and M. Yasar, “Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns,” *Signal, Image, and Video Processing*, vol. 3, no. 2, pp. 101–114, 2009, DOI 10.1007/s11760-008-0061-8.

[3] S. Gupta and A. Ray, “Statistical mechanics of complex systems for pattern identification,” *Journal of Statistical Physics*, vol. 134, no. 2, pp. 337–364, 2009, DOI: 10.1007/s10955-009-9679-3.

[4] A. Subbu, A. Srivastav, A. Ray, and E. Keller, “Symbolic dynamic filtering for image analysis: Theory and experimental validation,” *Signal, Image, and Video Processing (2009)*, in press, DOI: 10.1007/s11760-009-0122-7.

[5] D. Angluin and C. Smith, “Inductive inference: Theory and methods,” *ACM Computing Surveys (CSUR)*, vol. 15, no. 3, pp. 237 – 269, September 1983.

[6] C. R. Shalizi, K. L. Shalizi, and J. Crutchfield, “An algorithm for pattern discovery in time series,” Santa Fe Institute, Tech. Rep., 2002.

[7] D. Lind and M. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge: Cambridge University Press, 1995.

[8] A. Ray, V. Phoha, and S. Phoha, *Quantitative measure for discrete event supervisory control*. New York: Springer, 2005.

[9] I. Chattopadhyay and A. Ray, “Renormalized measure of regular languages,” *Int. J. Control*, vol. 79, no. 9, pp. 1107–1117, 2006.

[10] J. P. Crutchfield and K. Young, “Inferring statistical complexity,” *Physical Review Letters*, vol. 53, 1989.

[11] D. Ron, Y. Singer, and N. Tishby, “On the learnability and usage of acyclic probabilistic finite state automata,” *Journal of Computer and System Sciences*, vol. 56, no. 2, pp. 133–152, 1998.

[12] S. Garruzzo and D. Rosaci, “Agent clustering based on semantic negotiation,” *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 3, no. Issue 2, pp. 7.1–7.39, May 2008.

[13] S. Phoha, T. LaPorta, and C. Griffin, *Sensor Network Operations*. IEEE Press, Wiley Interscience, 2006, pp. 51–68.

[14] B. Calder, L. Linnett, and D. Carmichael, “Bayesian approach to object detection in sidescan sonar,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 145, no. 3, pp. 221–228, 1998.

[15] V. Rajagopalan and A.Ray, “Symbolic time series analysis via wavelet- based partitioning,” *Signal Processing 86*, vol. 11, pp. 3309–3320, 2006.

[16] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, p. 1729, 2002.

[17] C. Rao, K. Mukherjee, S. Gupta, A. Ray, and S. Phoha, "Underwater mine detection using symbolic pattern analysis of sidescan sonar images," in *2009 American Control Conference*, Hyatt Regency Riverfront, St. Louis, MO, June 10-12, 2009.

[18] S. Phoha and M. Schmiedekamp, "C3L: An integrated computation, communication and control language for dynamically controlled heterogeneous devices," in *Proceedings of Intelligent Systems and Control*, Cambridge, MA, November 19-21, 2007.

[19] J. Sustersic, "The control, communication and computation language (c3l): Completing the design cycle in complex distributed system development," Ph.D. dissertation, The Pennsylvania State University, University Park, PA 16803, May 2009.

[20] J. Sustersic and S. Phoha, "A C3L compiler for generating parallelized plans for collaborative multi-asset undersea missions," in *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, 2009.