OSU Department of Computer and Information Science Technical Report OSU-CISRC-11/02-TR26

Constructing Diagrams to Support Situation Understanding and Planning: Part 1: Diagramming Group Motions[1]

Bonny Banerjee
B. Chandrasekaran
John R. Josephson
Laboratory for Artificial Intelligence Research
Department of Computer & Information Science
The Ohio State University
Columbus, OH 43210

Robert Winkler
U. S. Army Research Lab
2800 Powder Mill Rd
Adelphi, MD 20783

{banerjee,Chandra,jj}@cis.ohio-state.edu
winkler@arl.army.mil

November, 2002

Abstract

Visual representations consisting of diagrammatic elements are ubiquitous in human problem solving. Diagrammatic Reasoning is a relatively new and challenging area of research in Artificial Intelligence and Human-Computer Interaction. The research described in this report is part of a larger project whose goal is to investigate a general Diagrammatic Reasoning architecture for problem solving. In some diagrammatic reasoning situations, such as military planning and weather prediction, it is necessary to abstract a mass of details into diagrammatic abstractions that are meaningful with respect to the problem solving goal. The research reported

[1] Research supported through participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009. Portions of this report were part of Banerjee's Master's Thesis submitted to the Department of Electrical Engineering, The Ohio State University, in August 2002.

here focuses on this problem as it arises in a military domain. Commanders represent and monitor their situation understanding and plans by drawing lines, arrows, regions and other diagrammatic objects on maps that contain terrain and other mission-relevant information. Some of the diagrammatic objects are lines of motion, while other objects are regions that abstract information about occupancy, control, and so on, while yet others are point objects that abstract only the location of some entity. This report describes the issues involved in building a diagram extraction system for this domain. We describe an architecture for the generation of diagrams that abstract significant groups and represent their motions from information about the locations and movements of a large number of Blue and Red military entities engaged in action. We present experimental results applied to data from military exercises. We also discuss techniques needed to generate other types of diagrammatic objects, and outline our research objectives for the near future.

Keywords: Diagram, Diagrammatic Reasoning, Situation Understanding, Clustering, Self-Organizing Neural Network.

1. INTRODUCTION

1.1 Motivation: The Bigger Picture of the Problem

The problem of interest in this project is to infer the intents and goals of a group of entities from their coordinated behavior guided by domain knowledge. An example of such a problem is the inference of the intents, spatial tactics, maneuvers, etc. of an army from the coordinated actions of a large number of its military units in the pursuit of a goal(s). The goal of the project at the Laboratory for Artificial Intelligence Research (LAIR) is to understand the cognitive representations and processes involved in reasoning about intents and plans of a coordinated collection of individuals, such as military units and agents, from visual representations of their locations and movements.

Given the locations and movements of a large number of individuals acting in a coordinated fashion in the pursuit of a goal(s), the problem is to infer from this information and domain knowledge the goal(s) that the group(s) as a whole might be pursuing. We wish to solve a version of this problem as it arises in the military domain. Our goal is to build an automated system that will take as input information about the location and movements over time of *military units* (consisting of individuals, vehicles, etc.) in the battlefield, relevant knowledge about the terrain and spatial information of the sort that is given by a map of the area, and produce as output the best hypothesis about the military maneuver that is being executed by them. The approach used by the researchers at LAIR decomposes the problem, as posed in the military domain, into the following parts (vide Fig 1):

1. **Extraction of a diagrammatic representation of the movements of groups at different levels of aggregation using perceptual organization:** First, we aggregate the individuals into meaningful groups at various levels of aggregation, and follow the

motions of these groups. The result of this analysis is a *diagram* that may be overlaid on the map. The diagram consists of lines of motion of various groups, along with labels that point to information about the groups, information such as Friend or Foe, type of unit, etc.

2. **Exploiting diagrammatic reasoning to abduce the type of maneuvers being attempted and eventually the plans and goals:** The second stage of the solution calls for matching the diagram with pre-stored templates of various types of maneuvers. Neither the diagram nor the templates are simple visual representations that are directly matched, but are complex knowledge structures that combine visual and symbolic elements to permit very flexible matching. Depending on the complexity of the situation, complex problem solving will be needed to identify the maneuver that is being carried out.

In what follows, we will focus only on the first step of the approach; identifying meaningful groups and calculating their motions to extract diagrams (vide Fig 2). Getting this algorithm to perform satisfactorily on complex real-world data may turn out to be challenging enough. The second step – the representations of maneuver templates and matching a given diagram to the maneuver templates to identify the best maneuver hypothesis – is left for the future.

1.2 What is a Diagram?

Diagrams are used widely as representations in many problem solving situations. While it is difficult to give a complete characterization of the necessary and sufficient conditions for a representation to be a diagram, within the scope of the present work, we will define diagram as a spatial representation consisting of objects (points, lines, regions, etc.), the objects being intended to represent some entities in the domain being represented (vide Fig 3). Thus, a Diagrammatic Representation (DR) is a form of Visual Representation (VR) consisting of abstracted diagrammatic objects that are relevant to problem solving [BC, 1997; BC, 2002; BC et al, 1993; BC & NN, 1993; NN & BC, 1991].

1.3 Organization of the Report

The report is organized as follows. In the next section, the problem of generating diagrams of group motions using perceptual organization at different levels of aggregation is discussed in considerable detail. A two-step approach has been proposed for solving the problem and the assumptions (constraints and requirements) specific to the military domain are outlined. Section 3 gives a brief overview of the perceptual grouping principles as relevant to the military domain, and the merits and demerits of different clustering algorithms are discussed. Special emphasis is laid on the two most widely used clustering algorithms, namely, the *k*-means clustering algorithm and the self-organizing feature map. Section 4 is dedicated to the clustering algorithm called the Information Extracting Self-Organizing Neural Network (IESONN) that has been used to group the individual elements based on certain domain specific perceptual properties. The philosophy behind the emergence of such an algorithm is included in Appendix B. The essence of information extraction is illustrated with the help of some simple synthesized datasets.

Appendix C contains a comparison between IESONN and closely related widely used traditional techniques. Section 5 presents the results obtained on the ARL datasets and on some synthesized datasets using the two-step approach discussed in Section 2. The results include determination of the groupings at the best level(s) of organization and the use of abductive inference to achieve consistency over a period of time at each level. It also illustrates which properties (velocity, identity, and proximity) might be considered with how much weightage for different datasets. Results are illustrated using both static (single frame) and dynamic (over a number of frames) formats. The concluding Section discusses the major contributions of the work presented in this report to the field of Artificial Intelligence. The Section ends with a note on some of the avenues of future research using diagrammatic representations.

2 THE PROBLEM

2.1 An Overview of the Problem

In the application that drives our research, we start with data about the locations and movements over a number of sampling instants of the individuals and vehicles of blue and red sides participating in an exercise at the National Technical Center. We also have terrain information. For the first set of experiments, we are interested in making hypotheses about the maneuvers that are being undertaken. This task is an intermediate stage in making hypotheses about the plans of the sides.

There are several different aspects of the situation and types of information that need to be diagrammed. First is the diagram of the terrain. Diagramming the terrain is similar to constructing a map, emphasizing abstractions relevant to military reasoning. This would consist of regions marked off as off limits for various reasons: mountains, not supportive of certain types of vehicles, rivers, etc. The diagram would also mark possible avenues of approach, and friendly and enemy regions and points of interest, such as cities, forts, etc. These are relatively static entities, and such a diagram can be constructed in advance. A terrain diagram corresponding to the terrain in Fig 4a is given in Fig 4b.

A diagram of the action needs to be overlaid on the diagram of the terrain. To diagram the action, it is useful to distinguish between different kinds of activities that take place before, during and after the battle, and the kind of motions that they involve. There is “movement to contact”, where a group is moving towards an enemy unit or some objective. There are defensive movements, where groups move to position themselves according to a defensive plan to be followed when contact with the enemy occurs. Then there are motions after contact begins, where the motions are determined by the interactions between the individuals involved in combat. Finally, there are movements associated with post-contact activities, such as retreat, and so on. While all of these involve group motions, and all motions are significant for some class of inferences, their characteristics are rather different, and they are informative for different inferential goals.

To describe the battle at the level of the plans and goals of the two sides, motions corresponding to movement to contact and motions of the defending side are important, but the latter are important only to the degree that they tell what the final defensive positions are. From a diagrammatic point of view, motions to contact are best described as lines of motion of significant groups, whereas defensive positions are best described as regions that block or threaten avenues of approach, and lines that describe defensive perimeters. While the motions during battle may be useful to describe its details, with respect to goals and outcomes, they can be replaced with simpler lines corresponding to any net motion. Attached to the various diagrammatic objects (such as lines and regions) will be symbolic abstractions of various kinds – such as identity, size, lethality, etc. – as needed for the inferential goals. As mentioned, movements of groups to contact can be best represented by line objects. We will shortly describe our current work in automatically constructing such diagrams of motion. Defensive positions can be represented by region objects standing for the spatial extent of the groups. As discussed earlier, blob abstraction algorithms described in [PE et al, 1998] can be useful for this purpose.

Fig 4c shows the overlay of Red defensive positions (unhatched region objects) on the terrain map. Because of the knowledge of approximate Red positions, the navigable routes in Fig 4b now become potential avenues of approach to objective for Blue forces. In the current report, the entire focus will be on constructing a diagram of motions of groups, which requires organizing the numerous individual agents on both sides into meaningful groups at different levels of aggregation and representing their motions as lines of motion.

Given as input a set of identities and locations of a large number of individual entities over a sequence of time, the problem is to extract a consistent account of the motion of groups of the entities across the entire length of time at multiple levels of organization. At any instant of time, the entities are required to be aggregated into one or more hierarchies of groups, any two such hierarchical structures being competitors of each other with respect to consistency. However, there might be time instants where only one hierarchical structure exists and hence the competition does not arise at all. Each level in the hierarchical structure at any instant of time corresponds to a particular level of organization at that instant.

Once the plausible hierarchical structures are determined for each instant of time, the next step is to determine the best level(s) of organization for which the diagram might be extracted. In the present problem, each grouping hypothesis consists of the hierarchy of the number of groups along with their respective constituents. For any time instant, we need to determine two issues:

1. The level(s) of the hierarchical structure at the present time instant which is most consistent with the grouping hypothesis at the best level(s) of organization at the previous time instants.
2. The best number of groups along with their constituents that is most compatible with the best number of groups along with their constituents at the last time instant at the chosen level of organization. This determination of the compatibility across time instants basically solves the *hypothesis matching* problem which allows us to get rid of the alternate hypotheses and only one of them emerges as the winner.

2.2 Nature of Input Data

The input to the system is the GPS data which consists of the identities and the track history of each of the military units (soldiers, tanks, etc.) obtained from ARL (vide Fig 5). The set of coordinates of each military unit taking part in the maneuvers at a given instant of time are extracted. Velocity information about each military unit is calculated from the coordinates at two consecutive instants of time. Due to the noisy nature of the input data, it is yet to be determined whether the velocity information should be used or not in this domain and if used, with what weightage. The work presented in the report is dedicated to exploiting the identity and motion information obtained from such data sets for aggregating the individual military units into meaningful groups using perceptual organization and hence extracting diagrams of the movements of those groups at the best level(s) of organization.

2.3 Top-Level Computational Strategy

The top-level computational strategy followed in this report for extracting a diagrammatic representation of the movements of groups at the different levels of organization given the set of coordinates and identities of each unit sampled at a sequence of time instants is as follows (vide Fig 2):

1. For each time instant, we generate a set of good hypotheses about meaningful groups at different levels of organization, based on proximity, similarities of identities and velocities of the units.
2. From the grouping at each level, we extract a consistent account of groups and hence draw lines describing the motions of the centroid of each group in order to obtain the desired diagram.

2.4 Assumptions: Domain Specific Requirements and Constraints

We do not intend to solve the problem of grouping in full generality as that is not required and desired for the present work. In order to group the individual military units into meaningful groups, we will exploit only those properties of perceptual organization that are relevant to the military domain. As for example, we will not consider similarity of rotational motion as a criterion for grouping different agents in the same group because such motion is very uncommon in the military domain.

In order to extract diagrams of the movements of groups at different levels of aggregation, we will resort to certain domain specific constraints and requirements, as follows:

1. Grouping is not only to be achieved at every instant of time in a discrete sense, but the grouping should also be consistent over a considerable period of time, both prior and after the time instant under consideration. This is necessary for the reduction of ambiguities as we look for consistency over a period of time.
2. In order to achieve the above requirement, at any given time instant, we will have to come up with not one best grouping hypothesis but a few very plausible ones and hence look for the most consistent one(s) across time instants. Also, additional higher level knowledge may be used to reinforce more consistency as maneuvers are being recognized, but we will not consider that issue in the current report.

2.5 Why is the Problem Complicated?

The problem at hand is complicated due to many reasons, some of which are discussed in the following.

2.5.1 Size

At a given time instant, there are many military units (of the order of 10^3). It takes a lot of computational power to perform clustering with so many data points at each time instant over a considerable length of time. For the present work, the accuracy of clustering is quite important because of the need for consistency across time instants.

2.5.2 Noise

The input data sets are noisy because of the use of very primitive tracking techniques which are not used any more. These tracking systems fail to receive GPS signals from military units under cloud cover or due to any other such interference. As a result, in the ARL data sets, we often find units cropping up now and then after being absent for considerable lengths of time. It becomes very difficult to assess the velocity information of such units and hence, often leads to spurious groupings if velocity information is used at all.

The ARL data sets contain unwanted information because of the presence of military units that do not participate in any maneuvers. Since every unit has a GPS associated with it, the non-participating units are also tracked the same way as the participating units leading to unnecessary information as far as recognition of maneuvers is concerned.

There are often few scattered units around each group and the parameters of the grouping algorithm have to be set judiciously in order to include those scattered units into the major group or keep them as separate distinct groups based on levels of organization. In order to determine whether to include these scattered units in the major group or not, one requires knowledge of their activities.

After a military unit has perished, its GPS still remains active sending out false positive signals, thus adding spurious information to the data. In order to get rid of this information, we track only the moving units in the process of diagrammatization.

Due to some reason, certain flying objects catch attention of the tracking system. As a result, we sometimes find existence of units in the ARL data sets which travel very fast from one position to the next compared to the other units. In our analysis, we get rid of these flying objects by discarding all units that travel faster than 45 miles per hour, as that is the maximum speed a military unit is known to travel on ground.

2.5.3 Alternative Hypotheses

Given a set of identities and locations of military units at any time instant, many alternative groupings are possible. Hence, it becomes necessary to assign a level of confidence to each such possible grouping and then choose the one best suitable for a given level of organization. Thus a robust yet generic measure of confidence to be assigned to each grouping is necessary.

2.5.4 Need for Consistency

At any time instant, only those groupings should be considered which are consistent over a length of time both before and after the time instant under consideration. It is important to impose such consistency in order to extract diagrams to infer about the intents/maneuvers that are being or will be carried on by the grouped units. Lack of consistency will only help to follow irrelevant groups' motions that will eventually complicate the inference procedure even more with spurious lines of motion.

2.5.5 Need for Aggregation at Multiple Levels of Organization

Since multiple levels of organization are being considered, all the above difficulties hold true for each of those levels. Organizations at lower levels are necessary in order to look into the phenomenon going on in more details. However, sometimes lower levels of organization provide more noise and irrelevant details that complicates the inference process. Hence, organizations at higher levels are also necessary.

2.6 Ways of Handling the Complications

2.6.1 Abductive Reasoning

At a particular instant of time, among the many possible alternative grouping hypotheses, we choose the best alternative by means of an inference procedure called Abductive Inference [JJ & SJ, 1994]. The best hypothesis is the one which best explains the ongoing phenomenon of diagrammatization with regards to certain factors like consistency. We group the individual military units at different levels of organization based on perceptual properties. At any given level of organization, we abductively choose that grouping which is the most consistent with the previous time instants. At any time instant and at any level of organization, such an inference is going to override the result due to the highest confidence assigned while grouping if that confidence fails to provide consistency across a length of time (vide Fig 14a). Also, choosing a particular grouping at a given level of organization helps to choose a particular grouping hypothesis and discard the others which eventually narrows down the search for grouping alternatives in the lower levels of organization for that time instant.

2.6.1.1 Assigning Plausibility for each time instant

For the present purpose, a grouping might be defined by the number of groups along with their constituent elements. Given the most plausible grouping at time instant t_{i-1}, we try to determine the most plausible grouping at the next time instant t_i that will provide the best consistent account of the motions of groups in the long run over the entire length of time. When the individual entities are grouped at a particular instant of time, each grouping is assigned a confidence or *plausibility*[2] based on the principle of *Ockham's razor* taking into account its improvement with respect to the previous grouping and the amount of system complexity incurred for achieving that improvement. This assignment of confidence is purely independent of any other information across time instants. Mathematically, such a measure of confidence or plausibility when the data set is partitioned into *k* clusters may be given by

$$\eta_k = \frac{\left(\phi_1 \big/ \phi_k \right)^a}{k^b} \qquad k = 1,2,...,n,\ 0 \le a,b < \infty; a,b \in \Re \qquad (2.1)$$

where ϕ_i is a measure of some property of the data set when it is partitioned into *i* clusters, *a* and *b* are parameters (real numbers) that allow the system or user to determine how much the system complexity should be emphasized for a particular application. The numerator in (2.1) serves to provide a measure of improvement that the system has achieved by partitioning the given data set into *k* clusters with respect to a single cluster while the denominator takes into account the system complexity incurred. When *b*=0, the system complexity is not taken into account. In most cases, we consider ϕ_k as the variance of the data set. Then the expression for η_k which suffices to account for the *Ockham's razor* is given by

$$\eta_k = \frac{\left(\sigma_1^2 \big/ \sigma_k^2 \right)^a}{k^b} \qquad k = 1,2,...,n,\ 0 \le a,b < \infty; a,b \in \Re \qquad (2.2)$$

where σ_i^2 is the total variance when the data set is partitioned into *i* clusters. Another measure for ϕ_k is discussed in Section 4.

2.6.1.2 Ensuring Consistency across time instants

We need to extract a consistent account of the motions of the groups over the entire period of time exploiting the grouping hypotheses obtained for each time instant. For each time instant, we choose the grouping with the highest plausibility (i.e. with the highest value of $\eta_i, i = 1,2,...,n$) assuming that corresponds to the best level of organization. Once this is done for the entire time period, we look back in search of inconsistencies across time instants. Generally two kinds of inconsistency are observed:

1. **Major Inconsistency**: This case happens when for a considerable length of time we

[2] For a scientific treatise of *plausibility* as used in the present context, the reader is referred to Appendix B of [JJ & SJ, 1994].

cannot find a consistent account of groupings. A major reason for the occurrence of such cases is the inability of η_i to provide a measure of plausibility for a single grouping. Hence, in order to overcome such major inconsistency, we check whether a single grouping explains the ongoing phenomenon satisfactorily or not. If it does, the existent groupings for the time instants under consideration are replaced by the single groupings at the respective time instants. However, if it does not, a closer look into the situation is taken to determine where the problem actually lies.

2. **Minor Inconsistency**: This case happens comparatively more frequently when there is a short burst of inconsistency, typically for less than five consecutive time instants. Reasons for such cases are manifold, the most important of them being noisy data. In order to overcome such adverse situations, at each time instant, we look for the grouping with the next highest plausibility and determine whether that would explain the ongoing phenomenon satisfactorily or not with respect to the previous and future time instants. If it does, the existent groupings for the time instants under consideration are replaced by the groupings with lesser plausibilities at the respective time instants. However, if it does not, a closer look into the situation is taken to determine where the problem actually lies.

This procedure generally suffices to provide a consistent account of the phenomenon going on. Results provided in Section 5 using both synthesized and real world data sets will illustrate the efficiency of this proposed approach.

2.6.2 Using Velocity Information

Velocity of each military unit is calculated from the input data (identity and location) given at two consecutive instants of time. Many alternate groupings are possible with the same input data at any given time instant. Additional information in the form of velocity is expected to create more robust groups and thus filter out some of the ambiguities and discard the less confident alternatives. However, due to the noisy nature of the input data, velocity information receives a much lower weight compared to the identity and proximity information (vide Fig 13b).

3 PERCEPTUAL GROUPING

3.1 Perceptual Grouping in Humans

According to the Gestalt psychologists [Wertheimer, Kohler, Koffka], the fundamental principles of perceptual organization are a set of generic criteria which underlie the natural mechanisms for partitioning the visual field. Some of these laws of organization, as relevant to the military domain, are as follows:

- ***The Law of Similarity:*** Similar elements of a stimulus tend to be part of a single unit.
- ***The Law of Proximity:*** Stimulus elements which are closer tend to be perceived as one entity.
- ***The Law of Common Fate:*** If a group of elements are moving with a common uniform

velocity through a field of similar elements, the moving elements are perceived as a part of a coherent group.

- ***The Law of Simplicity:*** In the stimulus where more than one figure can be perceived, the ambiguity is resolved in favor of the simplest alternative.

3.2 Different Approaches to Clustering

For the present problem, we will view perceptual grouping in the framework of clustering. We need to develop a suitable clustering algorithm in order to aggregate the individual military units into meaningful groups based on certain perceptual properties. Cluster analysis has been studied for a long time by numerous researchers working in varied fields. Recently, Breiman [LB, 2001] gave a distinction between two popular kinds of approaches to clustering, namely, the statistical approach and the machine learning approach.

3.2.1 The Data Modeling Culture: Statistical Approach

The analysis of this culture starts with assuming a stochastic data model for the clustering function. The values of the parameters for this function are estimated from the data and the model is then used for classification. Examples of such approaches are Discriminant Analysis, Logistic regression, Cox model, etc. The advantages of this approach are that the approaches are mathematically rigorous and are simple to understand. However, conclusions drawn from such modeled functions may be wrong if the model is a poor emulation of the training data. The goodness-of-fit tests have little power in higher dimensions and will not reject unless the lack of fit is extreme. This also leads to the problem of multiplicity of data models.

3.2.2 The Algorithmic Modeling Culture: Machine Learning Approach

The analysis of this culture considers the clustering function complex and unknown. Their approach is to find an algorithm to come up with the right classification given any arbitrary set of data. Examples of such approaches are Artificial Neural Networks (ANNs), Decision Trees, Support Vector Machines (SVMs), etc. These approaches look at a problem from a higher level compared to the other culture and deals with the same problems much more “intelligently”. No assumptions of data models are used since that would limit the scope of the solution. Many classes of algorithms are adaptive, biologically plausible and are very open to the real world problems. These approaches are always mathematically rigorous but might not be easily understandable.

In Appendix A, we visit some widely used clustering algorithms namely the k-means clustering algorithm and the self-organizing map (SOM) along with their pros and cons as well as the basic high level codes for implementing them.

4 Clustering Beyond the Imitation of Density

Though the self-organizing map and its variants have been used for feature extraction in numerous applications, yet almost all of them tend to extract redundant features due to introduction and updating of weights based on density and not on any other information regarding the data set under consideration. In the current section, a generalization over Kohonen's self-organizing feature map has been proposed. The processors of this proposed network, on convergence, tend to represent the information topology with respect to one or more desired properties of a given multidimensional data set in the framework of clustering. The proposed algorithm is called the Information Extracting Self-Organizing Neural Network (IESONN); the reader is referred to [BB, 2002] for details regarding this algorithm. Results obtained by deploying the proposed algorithm for extracting information from a wide range of multidimensional data sets, relevant to the military domain, for the generation of diagrams are presented in Section 5.

4.1 The IESONN Algorithm

Given a set of *N* data points $\{x_1(t), x_2(t), ..., x_N(t)\}$ and a set of variable (say, *k*) weights $\{w_1(t), w_2(t), ..., w_k(t)\}$ in a *d*-dimensional space (R^d) where *t* is the time coordinate, the IESONN is an algorithm for tuning the *k* weights to different domains of the data points such that w_i tend to be located in R^d in such a way that they approximate the function $\frac{1}{\phi(x)}$ of the data points in the sense of some minimal residual error; $\phi(x)$ may be given by

$$\phi(x) = \frac{\tau f(x)}{p(x)} \tag{4.1}$$

where τ is a factor to be determined. The functions *f(x)* and *p(x)* might be defined to reflect one or more desired properties of the data set under consideration. In this report, *p(x)* is defined as the probability density function (pdf) of the input data points while *f(x)* is defined by

$$f(x) = \frac{\varepsilon_c}{d} \tag{4.2}$$

where ε_c is the principal eigen value of the correlation matrix of the c^{th} cluster. When $\tau = \frac{1}{f(x)}$, the pdf *p(x)* is approximated by w_i as in SOM. When $\tau = p(x)$, the w_i are placed in such a way that the non-linearities in the data set are approximated with lesser emphasis on the pdf. One kind of optimal placement of w_i minimizes Φ, the expected reconstruction error, given by

$$\Phi = \int \left\| x - w_c \right\|^d \frac{1}{\phi(x)} dx \qquad (4.3)$$

where *dx* is the volume differential in R^d and the index $c = c(x)$ of the *winner* is a function of the input vector *x,* given by

$$\left\| x - w_c(t) \right\| = \min_i \left\{ \left\| x - w_i(t) \right\| \right\} \qquad (4.4)$$

The IESONN defines a clustering of the *N* data points into *k* partitions in an unsupervised and competitive manner such that Φ is minimized.

4.1.1 Initialization of the network

The IESONN is initialized with a very small number of non-interconnected processors, the weight corresponding to each of which assumes random initial values. Each feature vector, presented to the IESONN, is associated with an input vector from the *i*-dimensional input space, and an output vector from the *o*-dimensional output space (*i*+*o*=*d*). The weight vectors of the processors, having exactly the same input/output dimensions as the features, are updated iteratively on the basis of the feature space S, $S = \left(x_1, x_2, \ldots x_N \right)$ being the set of feature vectors initially.

4.1.2 Updating of weights

In IESONN, initially the topology is completely data driven. At time instant *t*, x_j is presented to the net. The net grows in size by means of a certain processor evolution mechanism, given by [3]

$$w_p\left(t+1\right) = w_p\left(t\right) + \alpha\left(t\right)[x_j - w_p\left(t\right)], \qquad 0< \alpha(t) <1 \qquad (4.5)$$

where $\alpha(t)$ is the gain term which decreases with *t* and $w_p(t)$ is the weight vector for the p^{th} processor at time *t*. All the weights compete and two winners $w_k(t)$ and $w_l(t)$ are selected according to (4.4) and (4.6) respectively.

$$\left\| x_j - w_l\left(t\right) \right\| = \min_{(i \neq k)} \left\{ \left\| x_j - w_i(t) \right\| \right\} \qquad (4.6)$$

x_j modifies $w_k(t)$ according to (5). $w_l(t)$ is also modified according to (5) if and only if it lies within a specified boundary, the radius R of which is given by

$$R = \frac{R_{initial}}{n_{weights}} \tag{4.7}$$

where $R_{initial}$ is the radius of the boundary at the initialization of the process while $n_{weights}$ is the number of processors at the instant under consideration. $R_{initial}$ is typically assigned a value to contain all the feature vectors in the entire data set. At any instant, R is adaptively chosen large enough not to hinder the influence of the nearby feature vectors on the processor. As the processor moves, it carries its boundary with itself, thus refraining from making the neighborhood topology of the net rigid.

In this process the modification of the weights is continued, the weights tend to approximate some desired property of the feature set in an orderly fashion. One presentation each of all the feature vectors makes one *sweep*. Several sweeps make one *phase*. One phase is completed when the weight vectors of the current set of processors converge, that is, when

$$\left\| w_i(t) - w_i(t') \right\| < \delta, \forall i \tag{4.8}$$

where t and t' are the time indices at the end of two consecutive sweeps and δ is a predetermined small positive quantity that decreases with t exponentially.

4.1.3 Introduction of a new processor

After the completion of a particular phase, a new phase starts with the introduction of a new processor. In order to choose the partition that deserves the new processor, the correlation matrix for each partition of the given data set is computed. Hence, the eigen value of each correlation matrix is computed. Each of the partitions is first normalized to have zero mean and unity variance to ensure that the eigen values are sensitive only to the pattern of the partitions and not to their spatial position. The new processor is introduced in that cluster which has the minimum value of $\phi(x)$ (vide equ. 4.1) among all the clusters, according to

$$w_{new} = \frac{Q_{\max} + Q_{\min}}{2} \tag{4.9}$$

where Q_{max} and Q_{min} are the two extreme points of the selected cluster. Lesser the value of $\phi(x)$, less correlated and more dense are the elements of that cluster.

The feature vectors are presented again to this new set of processors until they converge. This process continues until the desired number of processors is reached. For a discussion on the philosophy behind the IESONN along with the high level code for implementation, refer to Appendix B.

4.2 Results

The results obtained by using the proposed algorithm (IESONN) have been shown in Fig 7 and Fig 8. Comparison of these results with the traditional SOM and its variants suggests that introduction of weight vectors on the basis of information content not only helps in extracting more meaningful features but also helps to get rid of unnecessary information and thus helps assign more meaningful connectivity wherever relevant. Extraction of features based on information content becomes particularly useful in exploration of huge data-structures. Sometimes, it becomes necessary to infer from the extracted features of a huge data-structure in real-life problems [BB, 2003; BB et al, 2000] whereby the representation of the features based on uniqueness becomes all the more important. In all the illustrated figures, the circles denote the weight vectors on convergence while the crosses denote the feature vectors specifying the data set.

4.2.1 The Essence of Information Extraction

In Fig 7, the data set has been intentionally generated to have four identical points at (10.0,10.0). Since those four points are identical, three of them are redundant, as they don't contain any more information than one of them. It is noteworthy that unlike IESONN, the SOM fails to recognize this redundancy and hence, the representation of the information content in the data set is not achieved.

In Fig 8, a data set has been synthesized to contain features along a straight line and a "Λ" shaped structure. It can be seen that the SOM converges at spurious states while the IESONN successfully extracts features based on the information content using the same number of weight vectors. The introduction of the weights is noteworthy as for the SOM (and its variants) weights tend to be introduced based on density while for the IESONN weights are introduced based on information content or distinctiveness.

4.3 Conclusion

The illustrations demonstrate the ability of the proposed algorithm (IESONN) to extract "information" from a given data set. In the proposed algorithm, the nearest weight vector relative to a feature is left to wander freely in the state space while the neighborhoods of the other weights have been adaptively shrunk to reduce the influence of far off feature vectors. Thus the placement due to the introduction of the weights based on the information content or uniqueness of the clusters is kept significantly intact. As a result on convergence, the final positions of the weights tend to cluster the data set under consideration on the basis of uniqueness rather than based on feature density.

One of the issues to be considered is the sufficient number of processors required to successfully extract all the information contained in the given data set. This leads to the notion of *Ockham's razor* which says "simple but not simpler", i.e. the number of processors required to extract the necessary information from a given data set should not be multiplied beyond necessity. However, necessity might be defined in different ways for the same data set for different purposes and even, may not be known apriori. The number of processors required will depend on the specific purpose or the problem for which relevant information is being extracted. Fig 9 shows how to determine the sufficient number of processors required to extract information for the well known X-NOR classification problem. Since, two weight vectors are insufficient to classify the feature vectors correctly, it is a non-linear classification problem. However, three weight vectors can correctly classify the feature vectors just as well as four weight vectors, which is being shown in the plot of RMS error vs. number of processors. Thus, three processors are both necessary and sufficient to extract information for the X-NOR classification problem. In general, for the IESONN, a good measure of Ockham's razor may be given by

$$\eta_k = \frac{\left(\phi_1 / \phi_k \right)^a}{k^b} \qquad k = 1, 2, ..., n,\ 0 \le a, b < \infty; a, b \in \Re \qquad (4.10)$$

where the symbols denote the same as in (2.1).

The proposed algorithm (IESONN) has been experimented to successfully work with a lot of different practical problem areas like statistical pattern recognition (speech recognition, character recognition), navigational planning (obstacle avoidance and path-planning) of mobile robots [BB et al, 2000], adaptive design of various telecommunication devices [BB, 2003], image compression, structure exploration, multidimensional optimization and classification problems. In the next section, we will see how the proposed IESONN algorithm consistently handles data sets that occur in the military domain.

5 Results

This Section presents the results obtained by deploying the computational strategy (vide Fig 2) described in Section 2 for generating diagrammatic representations of the coordinated actions of a large number of military units in the pursuit of a goal(s) from their identities and locations.

5.1 Clustering using IESONN

The IESONN has been described as a clustering algorithm in the last chapter. Given a set of N data points in a d-dimensional space (R^d) and an integer k, the IESONN is an algorithm for partitioning the N data points into k disjoint subsets $S = (S_1, S_2, ..., S_k)$ containing $(N_1, N_2, ..., N_k)$ data points respectively so as to minimize an expected reconstruction error Φ, given by (4.3), such that $(x_i(t), w_j(t) \in R^d, S_p \bigcap_{\substack{1 \le p,q \le k \\ p \ne q}} S_q = \varnothing, \sum_{j=1}^{k} N_j = N)$, $x_i(t)$ and $w_j(t)$ being the i^{th} data point and centroid of the j^{th} cluster $\left(S_j\right)$ respectively, both at time instant t.

The IESONN is a generalization over one of the most widely used clustering algorithms called the SOM, discussed in Section 3.3.2. Due to the various drawbacks of SOM when applied to clustering, it had to be modified several times for different applications. However, almost all of these variants of SOM tend to represent the probability density function of the data set under consideration at convergence which is not always desired. The IESONN allows the system or user to tune its parameters according to the needs without the necessity to modify the algorithm. As discussed in Section 4, the results obtained from SOM can also be obtained from the IESONN i.e. the IESONN can also be used to imitate the probability density function of a data set if required. But at the same time, it can also be used to imitate some other desired property like the non-linearities in a data set with lesser emphasis on imitating the probability density function, if required, and hence effectively represent the unique or distinctive features of the data set under consideration. In the present application, we have equally emphasized the imitation of the probability density function and minimization of the non-linearities by setting $\tau = 1$, vide (4.1).

Fig 10a, b and c shows the proficiency of IESONN in generating alternative grouping hypotheses at multiple levels of organization using a typical synthesized data set. Fig 10c shows two mutually incompatible grouping hypotheses generated by the IESONN. In the process of diagrammatization, such incompatibilities are sorted out by taking into consideration the plausible grouping hypotheses at the previous and future time instants such that consistency is maintained in the long run. For a comparison between IESONN and other traditional approaches with regards to the present application, refer to Appendix C.

5.2 Results from the Military Domain

In this Section we will illustrate the diagrams that have been obtained by deploying the algorithm described in Section 2 and shown in Fig 2. We will follow a particular labeling procedure in all the illustrated diagrammatic representations in accordance with the US Army field manuals (FM 101-5). The paths followed by the centroid of the grouped military units will be marked with lines with arrow heads depicting the directions of motions. The centroid of the grouped entities at every instant of time will be denoted by dots for enemy and friendly units. It is noteworthy that a diagram, in the present application, is a mapping of the temporal information into some kind of annotated spatial information.

Fig 12 illustrates the generation of a diagrammatic representation from the synthesized identities and locations of the military units over 21 consecutive time instants. The data is virtually devoid of all noise and conservation of entities is maintained throughout. This data set is instrumental in showing how well the proposed architecture should work in ideal conditions or when the military uses tracking instruments with a very high precision. It is noteworthy that the proposed IESONN algorithm groups the individual military units at each time instant on the basis of proximity, similarities in identity and velocity. The accuracy of the motions in the resultant diagram especially during intersections shows the robustness of the proposed algorithm when accurate velocity information is obtainable.

Fig 13a and b shows the generation of a diagram from a real life ARL data set “n941a111”. This particular data set is extremely noisy as it has been generated by deploying primitive tracking systems which are not used anymore. The data dates back to an operation performed on 10th and 11th October, 1993 (dates are fictitious) over a period of fifteen hours with 1,827 military units (including friendly and enemy) taking part and is a part of a larger maneuver. The data for the enemy side is a bit noisier compared to the friendly side. Fig 13a illustrates the resultant diagram by first not using the identity information of the individual units and then by using the identity information. The diagram obtained by using the identity information is cleaner than the other one because identity information helps to get rid of many unnecessary lines of motion. If at least eighty percent of the members in a group at the present sampling instant does not exist in the predecessor of the group at the last time sampling instant, then the present group is not considered a descendant of its predecessor. This eighty percent rule is empirically followed throughout our analysis but it might be changed if necessity arises.

Unlike Fig 13a, Fig 13b illustrates the resultant diagram when similarity in velocity is considered with equal emphasis as proximity while grouping the individual military units into meaningful groups. Fig 13a illustrates the merging and splitting of groups which Fig 13b cannot. This is because, the velocity information is incomplete and unreliable due to reasons discussed in Section 2.5.2.

Fig 14a illustrates the use of abductive inference techniques to assure consistency over a length of time using the same ARL data set as in Fig 13. The confidence or plausibility associated with each grouping hypothesis at each sampling instant helps to choose the best grouping hypothesis at the best level of organization. After this procedure is completed, it is often seen that there occurs bursts of major and minor inconsistencies as discussed in Section 2.6.1.2. In order to ensure consistency, first we determine the inconsistent periods. Such an inconsistent period is illustrated in Fig 13a. Then the grouping hypotheses at each sampling instant in that period are revisited and each grouping hypothesis is compared with the chosen hypotheses at its neighboring sampling instants. The hypothesis at the present instant that best matches the hypotheses at the neighboring instants is chosen to override the result formerly obtained at the present instant. This procedure is illustrated from Fig 14b through g. The three-group hypothesis at each of the six consecutive sampling instants emerges to be the best hypothesis overriding the one-group hypothesis initially chosen at time instants t=1490 minutes and t=1510 minutes, thus ensuring consistency across time instants.

Fig 15 shows the generation of a diagram from another real life ARL data set “n941a113”. This data set is extremely noisy, being generated by the same primitive tracking systems as the previous one. This data dates back to an operation performed on 12th and 13th October, 1993 (dates are fictitious) over a period of twenty-eight hours with 1,830 military units (including friendly and enemy) taking part and is a part of a larger maneuver.

Fig 16 shows the generation of a diagram from yet another real life ARL data set “n941b116”. This data set has also been generated by deploying the same primitive tracking systems. The data dates back to an operation performed on 16th October, 1993 (dates are fictitious) over a period of nine hours with 1,834 military units (including friendly and enemy) taking part and is a part of a larger maneuver.

Fig 17 shows the comparison between the diagrams extracted deploying the proposed architecture and that drawn by LTC Gumbert on his visit to LAIR using the same data set. The three frames are shown in the illustration at regular intervals of time. It is noteworthy that the lines of motion in our extracted diagram very similarly follow the same path as the lines of motion drawn by an expert in the field of military maneuvers using his background knowledge after knowing the actual facts that happened on 16th October, 1993 between 1.00am and 10.05am, the duration of the maneuver. This maneuver has been identified as a *frontal attack* by the colonel which is clearly evident from our extracted diagram. This comparison provides a measure of the efficiency of the proposed architecture for the purpose of diagrammatization.

It might be noted that all motions in the extracted diagrams are significant for something, but only some motions are significant for understanding attacks, retreats, etc, where there is a large group-coordinated motion in one direction. Broadly, there are three types of motions which are not significant for this goal, and that might be abstracted out.

1. Zigzags and snakiness of movements caused by local terrain, but not significant regarding the broad direction.
2. Movements that are not related to coordinated directional goals, such as when a lot of units move around to prepare the defenses on the field.
3. Movements once action starts and when individuals' and subunits' motions are determined by local battle activity. Substantial zigzagging might be seen.

The series of diagrams in Fig 18 shows the resultant diagrams after the motions that are insignificant for the recognition of maneuvers have been abstracted away at multiple levels of abstraction. However, we have not considered any domain knowledge or any terrain information for smoothing the motions of the groups. Any turn that is very sharp (less than a prespecified angle) has been replaced iteratively by the resultant motion.

6 CONCLUSIONS

6.1 Main Contributions

Given the locations and movements of a large number of military units acting in a coordinated fashion in the pursuit of a goal(s), the problem was to infer from this information and domain knowledge the goal(s) that the group(s) as a whole might be pursuing by extracting meaningful diagrams of the motions of the group(s) overlaid on a terrain map. A diagram-extraction architecture was proposed, as shown in Fig 2, that provided a novel framework for obtaining a diagrammatic representation from a given visual representation motivated by the problem of recognition of intents/maneuvers in the military domain. For the purpose of grouping the individual military units into meaningful groups at multiple levels of aggregation, an adaptive unsupervised clustering algorithm was deployed which is a generalization of the SOM algorithm.

In order to effectively determine the optimum number of clusters present in a data set with no apriori knowledge of the data set, a simple yet robust measure was proposed which works satisfactorily in all the situations encountered in the military domain. The proposed measure also provides a confidence corresponding to the presence of a single cluster in the data set and hence, it is possible to find whether a single cluster is better than the other alternatives or not. This is by itself an open research problem among the various scientific communities [RT, 2000].

The proposed architecture automatically chooses the best level(s) of organization when generating the diagram of a given data set based on those plausibilities across time instants such that consistency is preserved in the long run. The illustrated results clearly manifest the success of the proposed architecture. Since the system abductively determines all by itself whether it should modify and/or rectify the results it has already obtained before producing the final results, it can be considered as an example of a system that improves its performance by its own evaluation.

6.2 Future Research

The work reported in this report solves a subproblem in the overall problem of constructing a diagram that helps effectively communicating the essence of the various activities that are taking place on the battlefield. We have explored the subproblem of drawing lines of motion corresponding to significant groups of military units. Units can be in motion for various purposes. All motion is significant in the sense that they all correspond to some intentions on the part of the agents who are in motion. Our goal is to construct diagrams that enable a problem solver to infer or express battle plans, and monitor them as battle proceeds. For this purpose, representing certain types of motions is important, while other types of motions will only complicate the diagram and distract the user from his problem solving goals.

Broadly speaking, before an engagement, defensive units might be moving about to position themselves in appropriate defensive positions. In this case, the motions themselves are not the important abstractions to be captured by the diagram. Rather, the locations and distributions of the final defensive positions need to be represented in the diagram. While the defenders are positioning themselves, the attacking side is moving to contact. Identifying significant groups and their motions and capturing the motions as lines of motions in the diagram is important. The work discussed in the report is intended to solve this problem. For this purpose, certain local motions – such as moving sideways to avoid an obstruction -- and zigzags – such as in following a winding road – are not significant, and might be smoothed out. We have discussed these issues in the body of the report.

When contact between the sides occurs, generally there is a lot of motion of individual units and small groups. But these motions are in response to very local battle goals, such as firing, or reacting to firing, and so on. If we need a diagram to capture the details of battle, diagramming these motions will be useful, but they are not significant bearers of information about over all battle plans. So, these details may be suppressed, if we can reliably identify that these motions correspond to post-contact battle details. If at the end of contact, one or the other side has been pushed back, then perhaps a line indicating the change in location would be useful to have in the diagram. At the end of contact, the two sides may move once again. For example, one or the other might retreat, move forward, or give chase. These motions again can be captured by identifying meaningful groups, their identities and their lines of motion. The techniques in this report will again be useful for capturing these motions. A useful diagram may consist of multiple segments, each diagramming a different stage of the battle.

Future research may be categorized into two types. The first type deals with improvements to the current algorithm for grouping and drawing coherent lines of motion. The following issues arise in this type. The first has to do with the role of velocity. Because of relatively poor quality of the current data set, especially the fact that information about unit locations are often missing for many instants of time and thus velocity cannot be reliably calculated, the relative importance of velocity information in comparison with proximity information has been hard to evaluate. Perhaps with better data, we will be able to answer this question more satisfactorily. Currently, the best we can say is that for the examples considered, satisfactory grouping could be achieved largely just with proximity as the basis for grouping. The second issue here is the clustering algorithm itself. We think that IESONN-based clustering has done pretty well, though perhaps many other clustering algorithms might have done more or less equally well. It is unclear if the specific properties of IESONN, e.g., it has more parameters for the designer to exploit for different types of generalization, are really essential for this domain. Conversely, perhaps other clustering algorithms, such as k-means, might actually be better suited. Our tentative conclusion is that the reason why our works well is not significantly due to this or other clustering algorithm used, but the use of appropriate criteria such as proximity, identity and to a lesser extent velocity, and also the various techniques for coherence based on an abductive inference perspective. However, we will remain open to improvements in the clustering algorithm.

The third issue is a more powerful abductive algorithm that makes use of more domain knowledge and a flexible problem solving architecture to make better decisions about coherence. In the long run, a certain amount of top down flow of information from higher levels of inference, such as maneuvers and plans, might be useful for producing better grouping hypotheses.

The second type of future research is extending the diagram construction goal from just lines of motion for groups to the series of diagrams for all the stages. For example, as discussed before, the diagram should include the defensive positions reached at the end of the precontact motions by the defending side. Similarly, we need to be able to decide when contact has occurred, and either diagrams the details of battle motions separately, or summarize it as net motions of sides at the end of contact. This will be followed up a post-end-of-contact tracking of motions.

Our immediate research goal is to add a knowledge-based problem solving architecture that makes use of and controls the parameters of the bottom-up group-hypothesizing and motion drawing modules so that the diagram is more accurate in displaying what is happening, and to show in the diagram not simply lines of motion, but also defensive regions and post-contact motions.

APPENDIX A

Clustering Algorithms revisited

In this section we are going to review some unsupervised clustering algorithms that have been widely used for numerous applications and discuss their pros and cons in view of the present problem. The goal of cluster analysis is to find disjoint subsets called clusters, such that at least one of the following criteria is satisfied [PH & BJ, 1997]:

1. **Homogeneity**: Entities within the same cluster should resemble one another.
2. **Separation**: Entities in different clusters should differ from one another.

1. The k-means Clustering Algorithm

Given a set of *N* data points in a *d*-dimensional space (R^d) and an integer *k*, the k-means clustering is an algorithm for partitioning the *N* data points into *k* disjoint subsets $(S_1, S_2, ..., S_k)$ containing $(N_1, N_2, ..., N_k)$ data points respectively, so as to minimize the sum-of-squares criterion given by

$$J = \sum_{j=1}^{k} \sum_{n=1}^{N_k} \left\| x_n - \mu_j \right\|^2 \qquad (3.1)$$

where x_n is a vector representing the n^{th} data point and μ_j is the centroid of the data points in S_j, $(x_n, \mu_j \in R^d, S_p \bigcap\limits_{\substack{1 \le p,q \le k \\ p \ne q}} S_q = \varnothing, \sum_{j=1}^{k} N_j = N)$.

The algorithm consists of a simple re-estimation procedure as follows. First, the data points are assigned at random to the *k* sets. Then the centroid is computed for each set. These two steps are alternated until a stopping criterion is met, i.e., when there is no further change in the assignment of the data points.

The Basic Algorithm:

```
Step 1: Initialization
    Initialize $N$, $k$, p according to the choice of the user or problem requirement.
    Initialize the centroids $\mu_1, \mu_2, ..., \mu_k$ randomly.
Step 2: Iterate
    While more than p centroids change, do
        For i from 1 to $N$
            Calculate the distance of the $k$ centroids from the i^th data point, $x_i$
```

```
                Find the centroid $\mu_t$ that is nearest to $x_i$
                Assign $x_i$ to the cluster $S_t$
            End.
            Calculate the new centroids of the clusters $(S_1, S_2, ..., S_k)$
        End.
Step 3: Result
        Output the $k$ centroids as the center of the $k$ clusters.
```

Drawbacks of the k-means Clustering Algorithm

The k-means clustering algorithm is often presented as a method which optimizes the center positions. However, it is important to note that the method is not a true global optimization algorithm [LB & YB, 1995]. In general, the algorithm does not achieve a global minimum of *J* (vide (3.1)) over the assignments. In fact, since the algorithm uses discrete assignments rather than a set of continuous parameters, the "minimum" it reaches cannot even be properly called a local minimum. Thus, it is a poor local method which ends up in the first stable configuration encountered which might have very serious consequences.

It is clear from the approach used in the k-means method that the solution supplied strongly depends on the initial positions of the centers of the would-be clusters. This can often result in very poor outcomes.

Advantages of the k-means Clustering Algorithm

Despite these limitations, the algorithm is used fairly frequently as a result of its ease of implementation. Generally, the various approaches to *k*-means clustering have time complexity O(R*kN*) where *k* is the number of desired clusters, *R* is the number of iterations needed for convergence, and *N* is the number of points needing to be placed into clusters [TK et al, 1999].

Modifications of the k-means Clustering Algorithm

The basic k-means clustering algorithm has been modified numerous times to fit into different perspectives, among which the fuzzy k-means [JD & AM, 1988] and the sequential k-means [JM, 1967; BM, 1996] clustering algorithms deserve special mention.

2. The Self-Organizing Feature Map (SOM)

Given a set of *N* input data points $\{x_1(t), x_2(t), ..., x_N(t)\}$ and a set of variable (say, *k*) reference vectors (or *codebook* vectors or *weights*) $\{m_1(t), m_2(t), ..., m_k(t)\}$ in a *d*-dimensional space (R^d) where *t* is the time coordinate, the SOM is an algorithm for tuning the *k* reference vectors to different domains of the input data points such that the node corresponding to m_i tends to be located in the input space R^d in such a way that they approximate the probability density function *p(x)* of the of the input data points in the sense of some minimal residual error [TK, 1990]. One kind of optimal placement of m_i minimizes *E,* the expected r^{th} power of the reconstruction error, given by

$$E = \int \|x - m_c\|^r \, p(x)dx \tag{3.2}$$

where *dx* is the volume differential in R^d and the index $c = c(x)$ of the *winner* is a function of the input vector *x,* given by

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \tag{3.3}$$

Equation (3.2) defines a placement of the codebook vectors (or nodes) into the signal (or input) space such that their point density function is an approximation to $\left[p(x)\right]^{\frac{d}{d+r}}$, where *d* is the dimensionality of *x* and m_i [TK, 1990]. In most practical applications $d \gg r$, and then the optimal vector quantization can be shown to approximate *p(x).* Usually, *r=2.*Thus, the SOM inherently defines a clustering of the *N* input data points into *k* clusters $(S_1, S_2, ..., S_k)$ such that $(x_j, m_i \in R^d, S_p \bigcap\limits_{\substack{1\le p,q\le k \\ p\neq q}} S_q = \varnothing, \sum\limits_{j=1}^{k} N_j = N)$ in an unsupervised and competitive manner.

The Basic Algorithm:

Step 1: Initialization

Initialize *N*, *k* according to the choice of the user or problem requirement.

Initialize weights from *N* inputs to the *M* output nodes (vide Fig 6) randomly. Set the initial radius of the neighborhood large enough to contain all the nodes.

Step 2: Present New Input

Step 3: Compute Distance to All Nodes

Compute distances d_j between the input and each output node *j* at time *t.*

Step 4: Select Output Node with Minimum Distance

Select node j^* as that output node with minimum d_j.

Step 5: Update Weights to Node j^* and Neighbors

Weights are updated for node j^* and all nodes in the neighborhood *Nc(t)* according to the following equation:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)\left[x(t) - m_i(t)\right], i \in N_c(t) \\ m_i(t), i \notin N_c(t) \end{cases} \tag{3.4}$$

where $\alpha(t)$ is a suitable monotonically decreasing sequence of scalar valued gain coefficients, $0 < \alpha(t) < 1$.

Step 6: Repeat by Going to Step 2.

Drawbacks of the SOM Algorithm

It was found that the neighborhood topology in SOM (vide Fig 6) is fixed which doesn't work well in some situations [JK et al, 1990; AD & SP, 1998; DC & SP, 1994]. This may be attributed to the fact that during the weight updating process, input vectors from the surrounding parts of the non-zero distribution may affect the weight vectors lying in the zero density areas. As the neighborhoods are shrunk the fluctuation vanishes making some processors remain outlier due to the residual effect. Moreover, due to the rigid topology of the net, the topology of the input pattern cannot be completely adapted.

Advantages of the SOM Algorithm

The main advantages of the SOM model, as compared to other clustering techniques, are its natural robustness and its very good illustrative power [AU & CV, 1994; AU, 1996]. Since it is an unsupervised algorithm, the SOM can be used for many real life data sets. The method is scalable, flexible, and reasonably fast. Additionally, the clusters are sorted according to the two dimensional regular discrete topology of the map. Thus, neighboring clusters are quite similar, while more distant clusters become increasingly diverse [TK, 1995]. Since the algorithm is adaptive, the result does not depend on the initialization of the weights unlike the *k*-means clustering algorithm.

Unlike the Carpenter-Grossberg classifier [GC & SG, 1986], the SOM can perform relatively well as a classifier in noise because the number of classes is fixed, weights adapt slowly, and adaptation stops after training. It produces impressive results when the desired number of clusters is prespecified and the amount of training data is large relative to the number of clusters desired [RL, 1987]. Using some benchmark data sets, SOM has been shown to work better than many classical approaches including the *k*-means clustering algorithm [AU & CV, 1994; AU, 1996].

Modifications of the SOM Algorithm

Many variants of the original algorithm were reported which included dynamic weighting of the input signals at each input of each cell, which improves the ordering when very different signals are used, and definition of neighborhoods in the learning algorithm by the minimal spanning tree, which provides a far better and faster approximation of prominently structured density functions [JK et al, 1990]. The Topology Adaptive Self-Organizing Neural Network proposed by Dutta et al [AD et al, 1997; AD & SP, 1998] helps to get rid of the rigid topology of Kohonen’s network.

APPENDIX B

Philosophy behind the IESONN

When implemented, the SOM and most of its variants tend to represent features of the multidimensional data set on the basis of density, which is not always desired. In order to represent the information content or distinctiveness in the data set, the clusters have to be formed in such a way that each cluster represents an unique set of information with respect to any desired property and not necessarily the density, and the number of clusters should be sufficient to hold the entire information contained in the given data set. When weights are introduced at the end of each phase, they should be introduced by comparing the information content (or uniqueness) of all the clusters and not solely on the basis of density. Further, after the weights are introduced in a region rich in non-linearities, utmost care should be taken to see that the weight vector does not migrate to regions of lower information content. Weights will, in general, have a natural tendency to migrate to regions irrespective of the information content because they are selectively updated based on the Euclidean distance, as a result of which the weights tend to settle for equilibrium based on density at convergence. In order to overcome this adverse situation, the neighborhood of the weight vectors have to be dynamically determined according to (4.7) such that they are influenced by feature vectors lying within that neighborhood only.

In order to compute the information content or a measure of uniqueness, the correlation matrix for each fragment of the given data set is computed. The correlation matrix provides a measure of correlation among different dimensions of all the elements in a cluster. It might be inferred that in general, better the correlation is, lesser is the information content in the cluster. This logic, though might not seem to be so obvious in higher dimensions at the first instant, is pretty obvious to perceive in two dimensions. It is due to the same logic that two points are enough to represent a straight line while two points are not enough to represent a parabola in any given dimension. This indicates that a parabola contains more information (or more non-linearity) than a straight line.

The principal eigen value of each correlation matrix is computed. As the variables become more correlated, the magnitude of the principal eigen value increases but the sum of the eigen values remains constant. Hence, the proposed algorithm (IESONN) tries to find the minimum of the principal (maximum) eigen values among all the clusters because lesser the principal eigen value is, less correlated the elements of that cluster are. At the same time the algorithm looks for the regions with maximum density. The new processor is introduced in that cluster which minimizes $\phi(x)$ according to (4.10). The degree of freedom defined by the parameter τ in (4.1) allows the user to adjust emphasis on minimizing non-linearities versus imitating the probability density function of the data set under consideration according to the requirements of the problem.

It is noteworthy that the feature vectors will attract the processors in each phase and hence their introduction in a particular cluster does not make considerable difference if the boundary of attraction is not intelligently determined at the beginning of each phase. As a result, for each weight vector, an adaptive nature of the boundary has been resorted to. The boundary is wide open when the algorithm starts with its initial list of processors, and any feature vector is free to attract any weight vector depending on the Euclidean distance. However, after a certain number of processors have been introduced, a restriction to the boundary is adaptively imposed according to (4.7) such that the newly introduced processor remains within that boundary at convergence. It might be noted that each time the radius of this boundary is chosen large enough not to hinder the significant influence of the feature vectors on the processor. Also, as the processor moves, it carries its boundary with itself, thus refraining from making the neighborhood topology of the net rigid. Thus the effect of introduction of the weights based on single value decomposition is kept intact.

The Algorithm:

```
Step 1: Initialization
        Initialize weights to random values, and the gain term $\alpha(t) \in (0,1)$.
        Set the initial radius $R_{initial}$ of the neighborhood.
        Obtain feature vectors in a random sequence.
        Initialize sweep and phase to zero.
        Set the desired number of processors at final convergence.
Step 2: Sweep
        For all feature vectors, update the weight vectors as obtained from (4.4) and (4.6), according to (4.5).
        Increment number of sweeps by 1.
Step 3: Check for convergence
        If (4.8) is not satisfied, go to Sep 2.
Step 4: Phase
        Assign connectivity to the processors if (4.9) is satisfied.
        If the desired number of processors hasn't yet been reached, find the single value decomposition of each cluster and insert a new processor according to (4.10), otherwise go to Step 5.
        Set sweep equal to zero, increment phase by 1.
        Randomize the feature vectors.
        Shrink the neighborhood of each processor according to (4.7), if required.
        Go to Step 2.
Step 5: Stop
```

APPENDIX C

Comparison between IESONN and the Traditional Approaches

The IESONN has been chosen as the grouping algorithm for having certain advantages over the other widely used grouping techniques as far as the present application is concerned. In this section, we will discuss the advantages of using IESONN for the present problem as the clustering algorithm with respect to some other prevalent clustering algorithms.

1. IESONN versus Decision Trees

Decision Tree is a form of inductive learning [SR & PN, 1995]. Logically, the decision tree can be expressed as a conjunction of individual implications corresponding to the paths through the tree ending in *Yes* nodes. Decision tree language is a propositional language, not as expressive as the predicate logic language used by the neural networks. Unlike the IESONN, the decision trees are limited to the representation of binary decision sets and have trouble representing relations between two or more objects. Like the IESONN, decision trees are simple and easy to implement.

In the context of clustering, the decision trees can be modified to perform cluster analysis whereby they take a top-down approach splitting the data points into two or more classes based on a single attribute at a time. However, in general, the cluster analysis approach is multivariate by definition, whereas the decision tree is univariate at each split. In other words, clusters are formed by cluster analysis in terms of *associations* between *all the active variables*, not by splitting at each node on a single variable as in a decision tree. It is little wonder that data miners resort to "boosting", "bagging" and cross-validation of different samples to try and find a "consensus" decision tree. If the initial split at the first node cuts through a natural cluster there is no hope of recovering the shape of the split cluster by a "top down" approach. The adaptive algorithms like the SOM, IESONN, etc. are naturally tolerant to noise and a faulty initial decision is not liable to produce erroneous results finally. Because of such reasons, the IESONN was preferred to the decision trees as a clustering algorithm in the present application.

2. IESONN versus Support Vector Machines (SVMs)

In SVM clustering [AB et al, 2001], data points are mapped from the data space to a high dimensional feature space, often using a Gaussian kernel. In the feature space, a hunt goes on for the smallest sphere such that it encloses the image of the data. This sphere is mapped back to the data space, where it forms a set of contours or cluster boundaries which enclose the data points. Just as in IESONN, it is always possible for a SVM to find a kernel rich enough to separate any data points. The number of support vectors for a real world problem can be very large resulting in high computational costs for on-line calculations. One major disadvantage of SVM is its necessity to solve a large scale convex quadratic programming problem. To overcome this disadvantage, the Least Square SVM deploys a linear Karush-Kuhn-Tucker system instead of quadratic programming but sparseness is lost as a result [JS et al, 2002]. Unlike the IESONN, SVMs scale rather poorly with the data size due to the quadratic optimization algorithm and the kernel transformation. Correct choice of kernel parameters is crucial for obtaining good results with SVMs, which practically means that an extensive search must be conducted on the parameter space before results can be trusted, whereas parameters in IESONN are very few and are strictly bounded. SVMs exhibit excellent generalization properties in many experiments, but suffer from the steep growth of number of support vectors with increasing size of the training set unlike in IESONN. In terms of generalization, the IESONN lacks behind SVM, though it possesses one of the best generalization capabilities among the unsupervised clustering algorithms. Since SVM is a supervised classification algorithm, it cannot be used for unsupervised clustering in the present application.

3. IESONN versus k-means Clustering Algorithm

The *k*-means clustering algorithm, discussed in section 3.3.1, is perhaps the most widely used clustering algorithm for real world applications mainly because of its simplicity and its computational efficiency in handling large multidimensional data sets. However, due to some serious drawbacks, the *k*-means clustering algorithm has not been used for clustering in the present application.

Good results from the *k*-means clustering algorithm require compact convex clusters of similar sizes [AD et al, 1999]. This is because the algorithm inherently tries to imitate the probability density function of the given data set. Another consequence of such *blind* imitation is that the algorithm is very sensitive to the outliers as they tend to bias the probability density function. But the most serious drawback of this algorithm is that the final results depend very heavily upon the initial positions of the centers of the would-be clusters [JP & PL, 1999]. Thus, a misplaced initialization will almost inevitably invite far reaching consequences. Also, the number of clusters that are present in the data set has to be provided as an input to the algorithm, which is not always possible.

The IESONN is carefully developed not to *blindly* imitate the probability density function of the data set under consideration. As a result, it is devoid of many of the drawbacks of *k*-means clustering algorithm like necessity of compact equal sized clusters and too much sensitivity to outliers. Also, being an adaptive algorithm, the IESONN gets rid of any serious consequences due to misplaced initializations. The proposed measure η_i in (2.1) and (2.2) allows the algorithm to select the best number of clusters present in the data set after the data set has been clustered.

4. IESONN versus Self-Organizing Feature Map (SOM)

The SOM, discussed in section 3.3.2, is perhaps the most widely used unsupervised clustering algorithm after the *k*-means clustering algorithm because it overcomes most of the drawbacks of *k*-means but inherits some others. Being an adaptive algorithm like the IESONN, the SOM gets rid of any serious consequences due to misplaced initializations. Like the IESONN, the SOM possesses the ability to learn complicated class boundaries in case of supervised applications.

Both of these approaches facilitate fast performance, natural robustness towards noise and ability to handle large number of fuzzy, overlapping, continuous attributes.

Like in *k*-means algorithm, the SOM also requires the number of clusters at convergence to be stated apriori. Since the SOM deliberately tries to imitate the probability density function of the data set under consideration, it also ends up clustering the data set in equal sized compact convex clusters and its performance is considerably influenced by the residual effect. Training time can be slow based on the nature of application and it always carries the risk of overfitting or underfitting the data set if the architecture is poorly chosen. The parameters of the IESONN are carefully chosen to overcome these drawbacks for the present application.

The IESONN offers more degrees of freedom to group the entities on the basis of uniqueness rather than on the basis of density. As a result much smaller groups can be identified in the presence of larger groups, vide Fig 11 for an example. Moreover, it inherits all the advantages of SOM and can be adjusted to obtain results as in SOM if required. Thus, for the problem of diagrammatization, the IESONN has been chosen as the suitable clustering algorithm after comparison with the widely used prevalent algorithms.

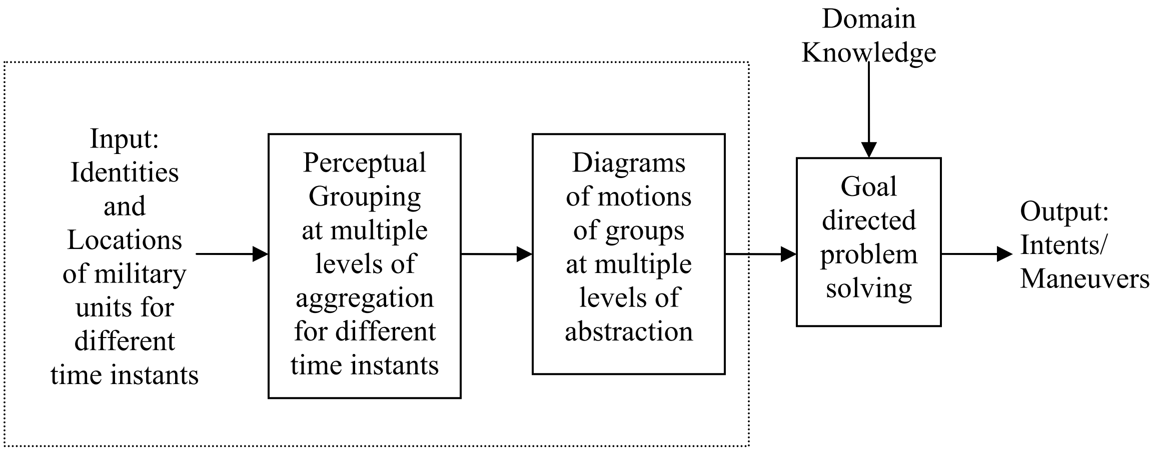

Domain Knowledge
Input: Identities and Locations of military units for different time instants
Perceptual Grouping at multiple levels of aggregation for different time instants
Diagrams of motions of groups at multiple levels of abstraction
Goal directed problem solving
Output: Intents/ Maneuvers

Fig 1: This block diagram shows the overall picture of the approach taken by the researchers at LAIR in order to infer about the intents, spatial tactics, maneuvers, etc. of an army from the coordinated actions of a large number of its military units in the pursuit of a goal(s). The part of the block diagram enclosed in the dotted block shows the first part of the approach i.e. extraction of a diagrammatic representation of the movements of groups at different levels of aggregation using perceptual organization.

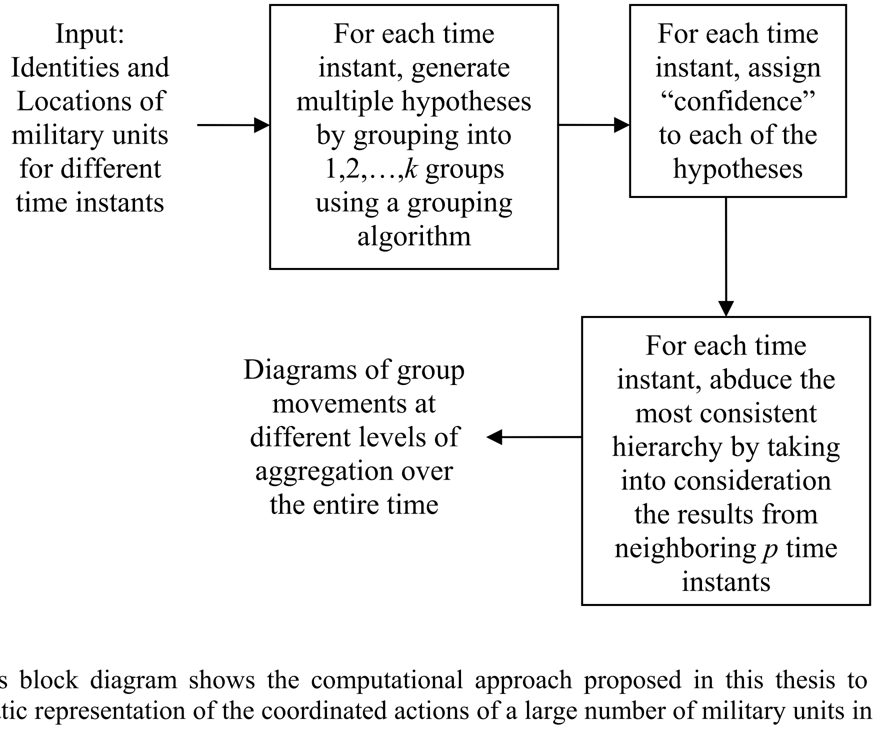

Input: Identities and Locations of military units for different time instants
For each time instant, generate multiple hypotheses by grouping into 1,2,…,k groups using a grouping algorithm
For each time instant, assign "confidence" to each of the hypotheses
For each time instant, abduce the most consistent hierarchy by taking into consideration the results from neighboring p time instants
Diagrams of group movements at different levels of aggregation over the entire time

Fig 2: This block diagram shows the computational approach proposed in this thesis to generate a diagrammatic representation of the coordinated actions of a large number of military units in the pursuit of a goal(s) from their identities and locations.

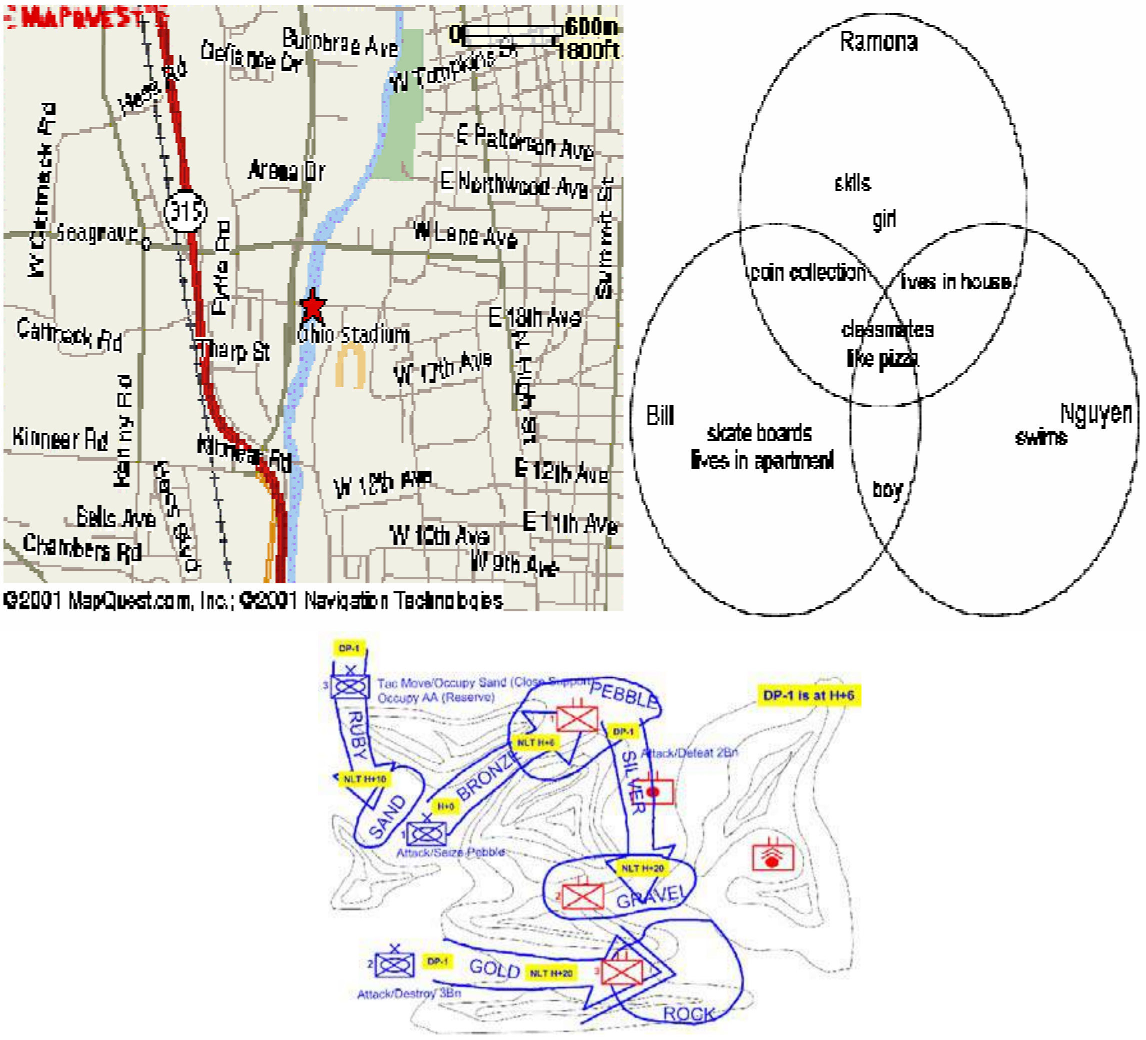
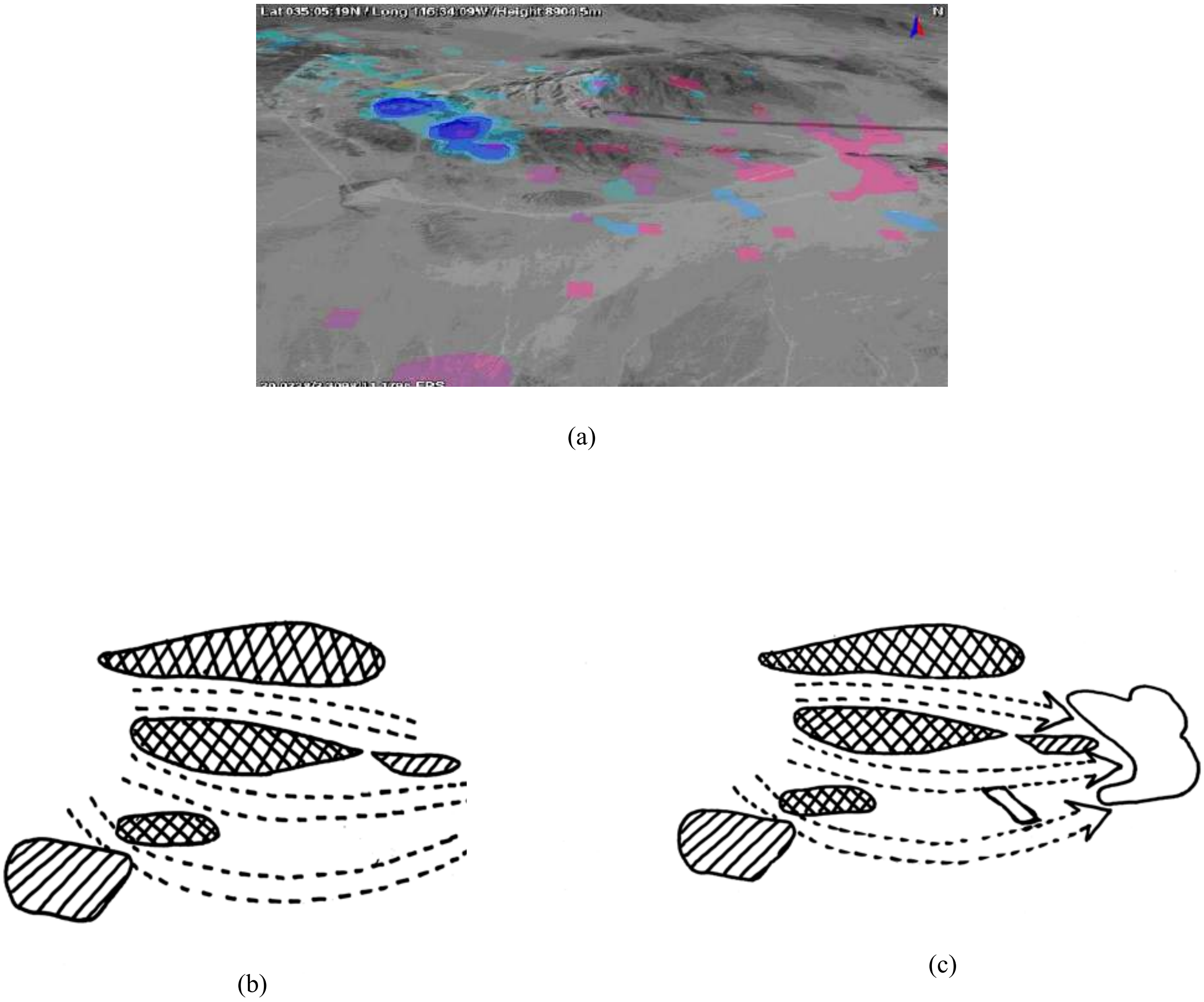


Fig 3: This figure shows some typical examples of diagrammatic representations like maps, Venn diagrams and a COA diagram using Army standard symbology. They are a kind of spatial representation consisting of abstracted diagrammatic objects (points, lines, regions, labels, etc.) for representing some entities in the domain under consideration. The map is taken from “www.mapquest.com”.



(a)

(b)

(c)

Fig 4: Figure (a) shows the *blobs* [JW, RW, PE] representing groups in an exercise at the National Technical Center. Figure (b) shows a diagram of the terrain obtained from figure (a). Double-hashing indicate “no-go” regions, and single-hashed ones are “slow-go”. The dotted lines indicate navigable paths. The terrain diagram might also include other elements such as military installations, rivers, etc. Figure (c) shows the terrain diagram with an overlay of Red defensive positions (unhatched regions), and avenues of approach (dotted arrows) for Blue towards the Red objective on the right, also obtained from figure (a).

```
Blue Player 532 named HHC/1-5                          is a Battalion
Red Player 533 named 203mm SP HOW OA13                 is a Battalion
Blue Player 534 named TH66                             is a M2_IFV
        Track History:
                (35.3707,-116.4353) @ 10-Oct-1993 08:50:00
Red Player 535 named 613                               is a BRDM
        Track History:
                (35.1311,-116.6279) @ 10-Oct-1993 08:50:00
Blue Player 536 named EH3                              is a MANPACK
        Track History:
                (35.1573,-116.6481) @ 10-Oct-1993 08:50:00
                (35.3182,-116.7364) @ 10-Oct-1993 16:20:00
                (35.3183,-116.7370) @ 10-Oct-1993 16:30:00
                (35.3186,-116.7298) @ 10-Oct-1993 16:50:00
                (35.3183,-116.7370) @ 10-Oct-1993 17:00:00
                (35.3224,-116.7169) @ 11-Oct-1993 06:45:00
                (35.3224,-116.7168) @ 11-Oct-1993 07:10:00
                (35.3224,-116.7168) @ 11-Oct-1993 07:15:00
                (35.3224,-116.7169) @ 11-Oct-1993 07:20:00
                (35.3223,-116.7169) @ 11-Oct-1993 07:25:00
                (35.3219,-116.7175) @ 11-Oct-1993 07:30:00
                (35.3215,-116.7177) @ 11-Oct-1993 07:35:00
                (35.3197,-116.7180) @ 11-Oct-1993 07:50:00
                (35.3240,-116.7189) @ 11-Oct-1993 07:55:00
                (35.3204,-116.7184) @ 11-Oct-1993 08:00:00
                (35.3215,-116.7177) @ 11-Oct-1993 08:05:00
                (35.3215,-116.7177) @ 11-Oct-1993 09:05:00
                (35.3214,-116.7174) @ 11-Oct-1993 09:30:00
                (35.3224,-116.7166) @ 11-Oct-1993 09:35:00
                (35.3224,-116.7166) @ 11-Oct-1993 09:40:00
                (35.3222,-116.7168) @ 11-Oct-1993 09:50:00
                (35.3214,-116.7177) @ 11-Oct-1993 09:55:00
                (35.3223,-116.7166) @ 11-Oct-1993 10:05:00
                (35.3206,-116.7063) @ 11-Oct-1993 10:10:00
                (35.3110,-116.6730) @ 11-Oct-1993 10:20:00
                (35.3077,-116.6521) @ 11-Oct-1993 10:25:00
                (35.3069,-116.6507) @ 11-Oct-1993 10:35:00
                (35.2936,-116.6405) @ 11-Oct-1993 10:40:00
Red Player 537 named A23                               is a T72_TANK
        Track History:
                (35.1329,-116.6326) @ 10-Oct-1993 08:50:00
                (35.1324,-116.6286) @ 10-Oct-1993 10:00:00
                (35.1324,-116.6285) @ 10-Oct-1993 10:20:00
                (35.1323,-116.6282) @ 10-Oct-1993 10:40:00
                (35.1328,-116.6293) @ 10-Oct-1993 10:50:00
                (35.1324,-116.6285) @ 10-Oct-1993 11:20:00
                (35.1325,-116.6292) @ 10-Oct-1993 11:30:00
                (35.1325,-116.6297) @ 10-Oct-1993 11:50:00
                (35.1324,-116.6286) @ 10-Oct-1993 12:00:00
                (35.1325,-116.6296) @ 10-Oct-1993 12:20:00
                (35.1328,-116.6286) @ 10-Oct-1993 12:30:00
```

Fig 5: An excerpt from the data provided by the Army Research Labs for the current project. This particular data has been obtained deploying 1,827 military units on the 10th and 11th of October, 1993 (dates are only for research purpose). It is particularly noisy because of the use of primitive tracking techniques.

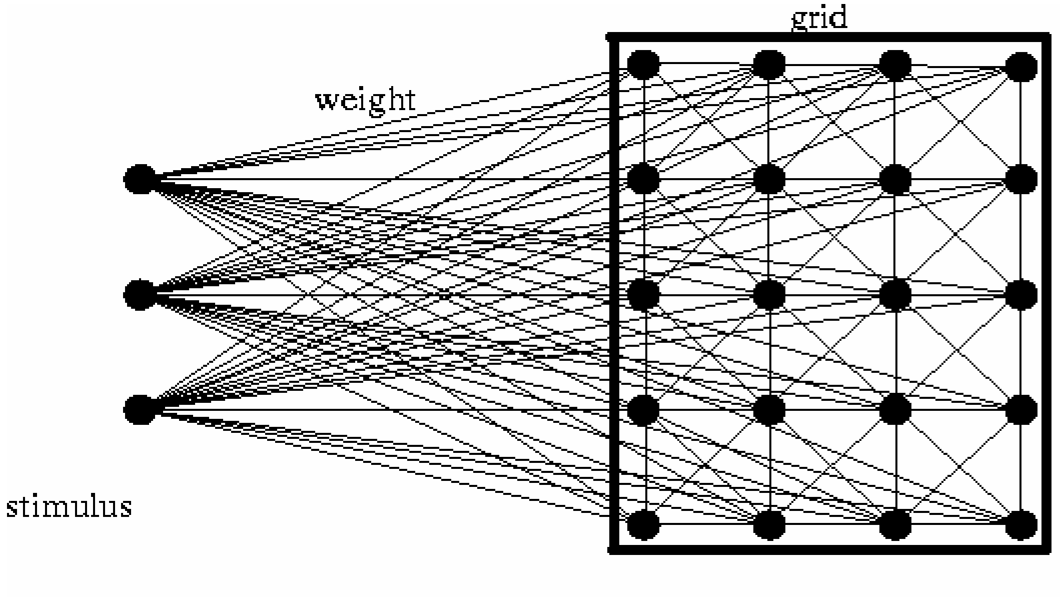
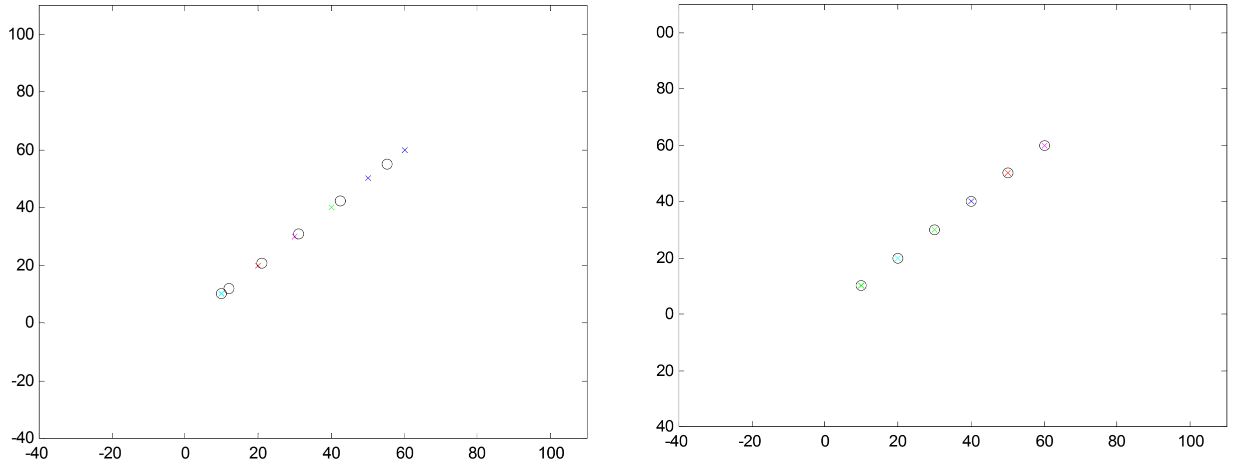


Fig 6: This is a schematic diagram of Kohonen's self-organizing feature map (SOM) network. Each unit of the two-dimensional grid is linked to the input vector (stimulus) by means of *d* synapses of weight m_i. Thus each unit is associated with a vector of dimension *d* which contains the weights, m_i, $i=1,2,\ldots,k$.



10.0000	10.0000
10.0000	10.0000
10.0000	10.0000
10.0000	10.0000
20.0000	20.0000
30.0000	30.0000
40.0000	40.0000
50.0000	50.0000
60.0000	60.0000

. (a)

10.0000	10.0000
11.9985	11.9985
20.8520	20.8520
30.9148	30.9148
42.4670	42.4670
55.1770	55.1770

(b)

10.0000	10.0000
20.0044	20.0044
29.9951	29.9951
39.9997	39.9997
50.0000	50.0000
59.9997	59.9997

(c)

Fig 7: The pair of columns in (a), (b) and (c) shows the coordinates of points in a synthesized dataset, the coordinates of the six weight vectors on convergence using the conventional SOM algorithm and the coordinates of the six weight vectors on convergence using the proposed algorithm (IESONN) respectively. The same is depicted in the figures presented above, where the synthesized data points are represented by "x" while the coordinates of the weights at convergence by "o", the left and the right figures are obtained by deploying the conventional SOM and the IESONN respectively.

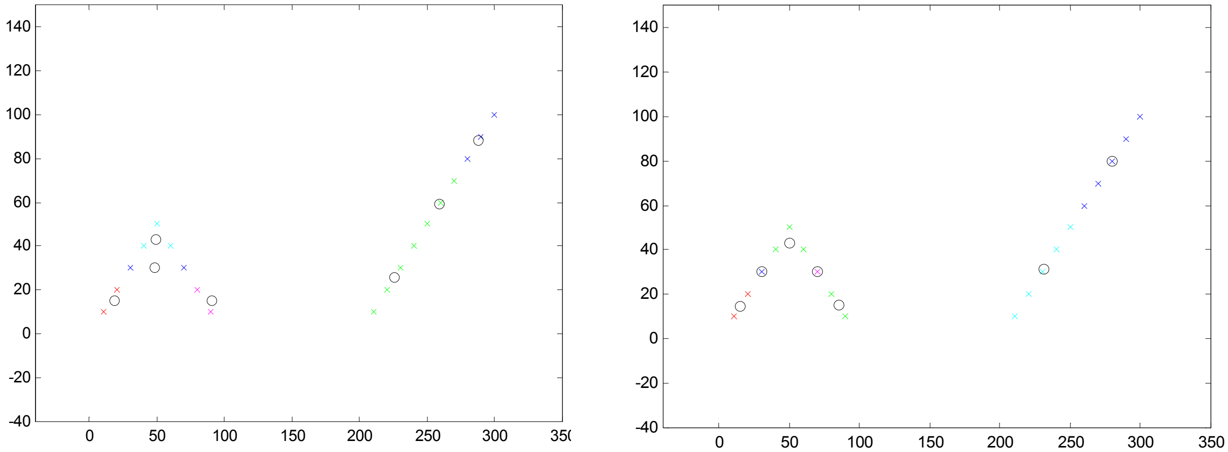


Fig 8: The left and the right figures show the position of the weight vectors on convergence using the traditional SOM and the proposed algorithm (IESONN) respectively. The SOM converges at spurious states while the IESONN successfully extracts features based on the information content or uniqueness using the same number of weight vectors.

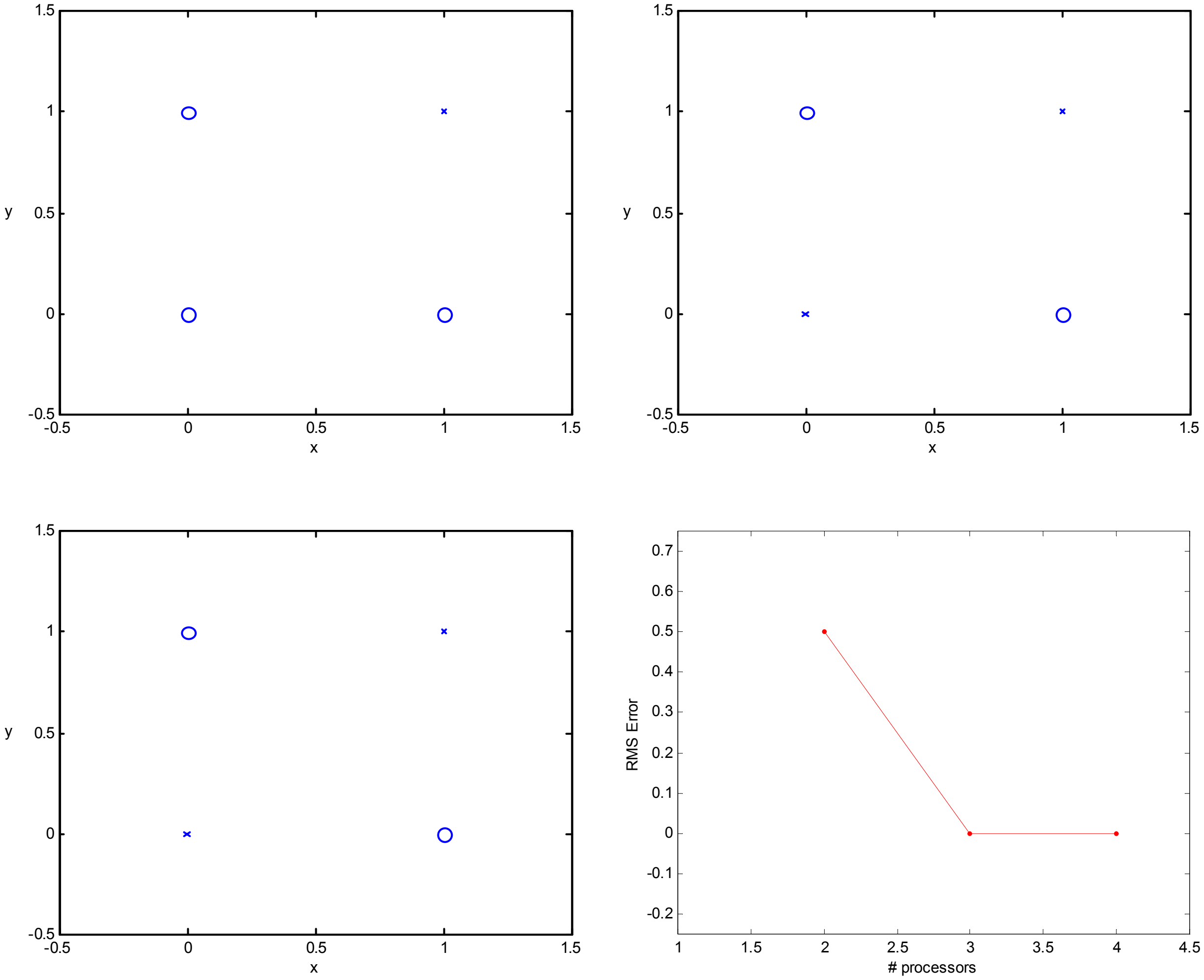


Fig 9: The above figure shows information extraction using the proposed algorithm (IESONN) for the well-known X-NOR classification problem. The 'x' and 'o' denote the feature vectors corresponding to outputs equal to 1 and 0 respectively. The top left, top right and bottom left are the classifications obtained when the IESONN converges with 2, 3 and 4 processors respectively. The bottom right shows the RMS error due to each of the above classifications. This example shows the classification capabilities of IESONN when the desired outputs are provided.

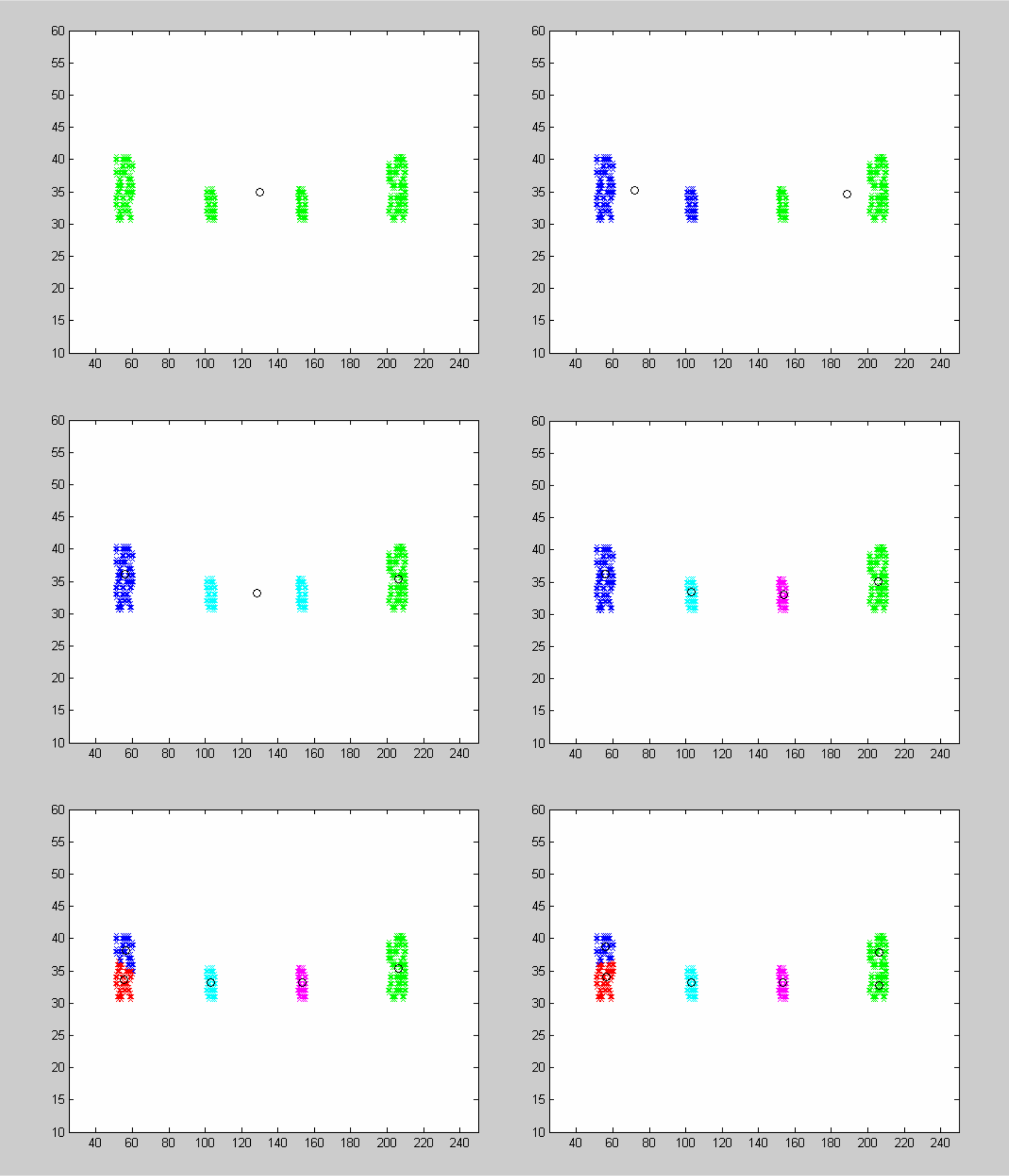


Fig 10a: The above figures illustrate the capability of IESONN to break down a given data set into multiple levels of organization. The synthesized data set used in this example has been shown to be broken down into 1, 2,…,6 groups and each time the representative centroids of the groups are shown by circles.

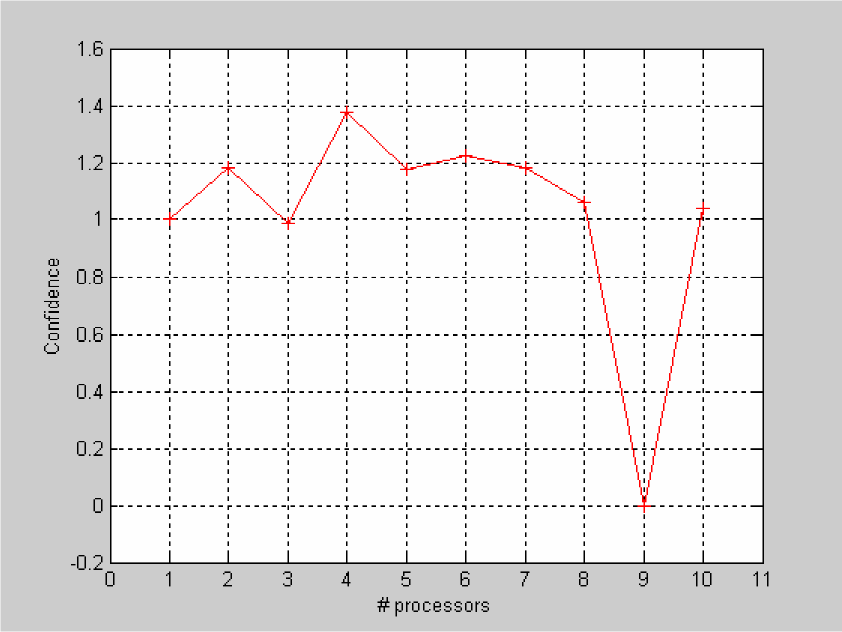


Fig 10b: This figure uses the synthesized data set (used for Fig 10a) to show the efficiency of the metric proposed in equations (2.1) and (2.2) respectively to determine which grouping hypotheses are better than the others. However, the conclusion drawn from this measure of confidence may be overridden by the abductive inference drawn from the neighboring frames in order to ensure consistency.

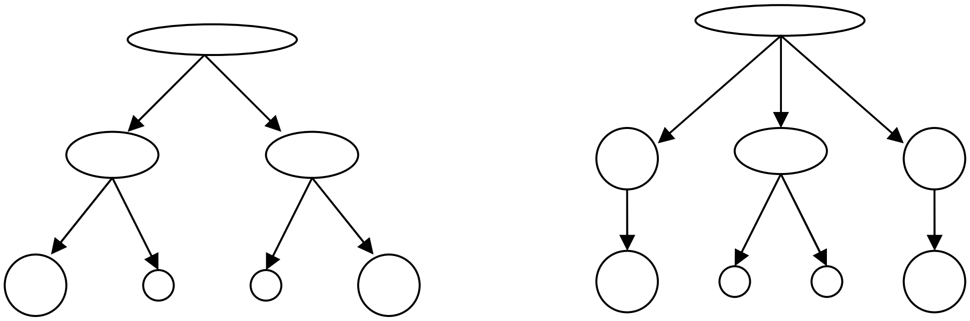


Fig 10c: This figure illustrates the two competitive hypotheses generated by the IESONN from the synthesized data set illustrated in Fig 10a. One of them will be chosen based on the best hypotheses at the neighboring time instants. However, not considering any information across time instants, the hypothesis on the left enjoys higher confidence than on the right, as obtained from Fig 10b.

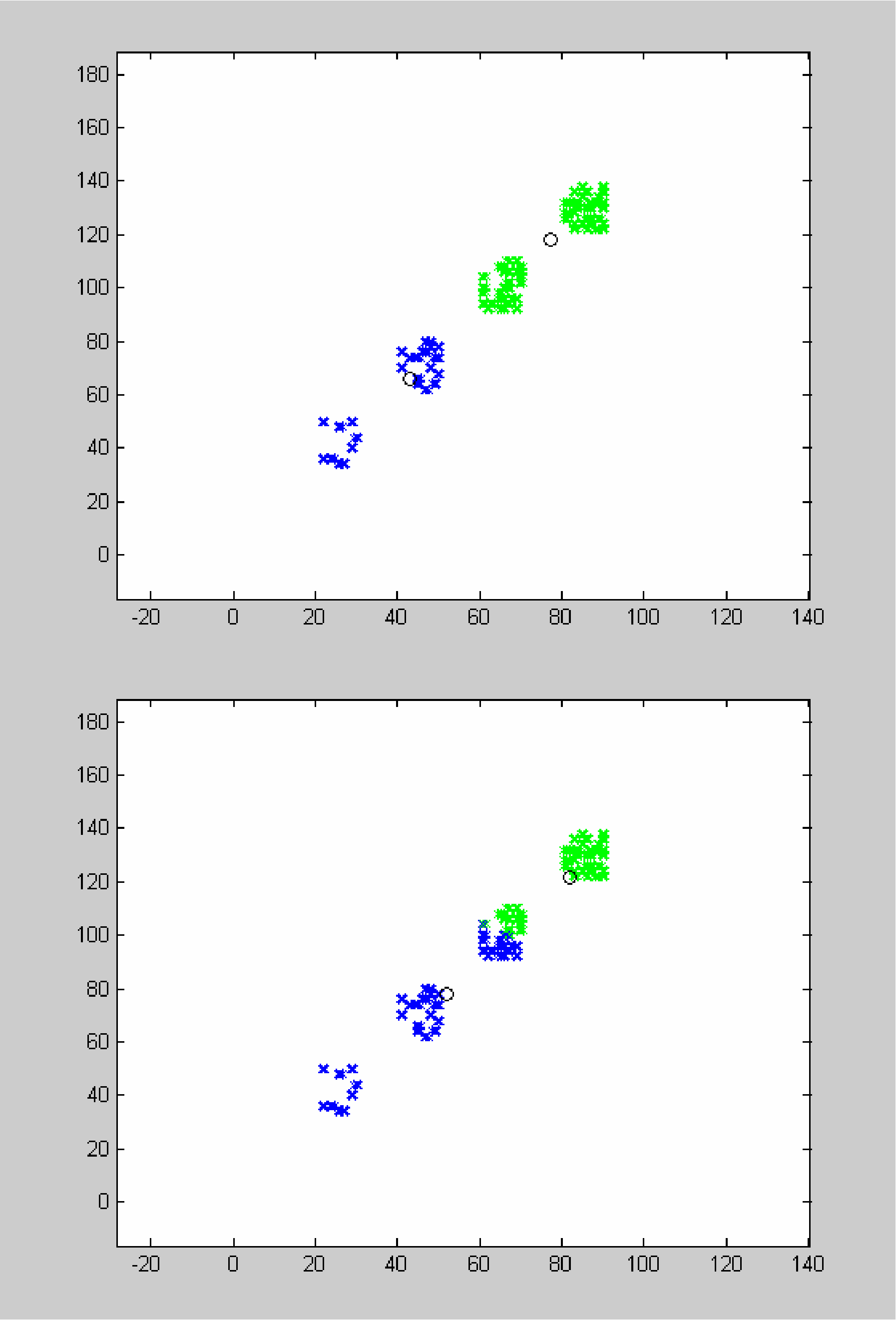


Fig 11: These figures illustrate a typical example of clustering using the proposed and closely related widely used conventional algorithms. The figure on top shows a synthesized data set being clustered into two groups on the basis of proximity by IESONN. The figure below shows the two clusters as obtained by deploying SOM and the classical *k*-means clustering algorithms on the same data set using the same proximity information. This example manifests the difference in performance of SOM and other such algorithms who try to imitate the probability density function of the data set under consideration with respect to IESONN. The IESONN performs differently due to its ability to cluster on the basis of uniqueness with controlled emphasis on density.

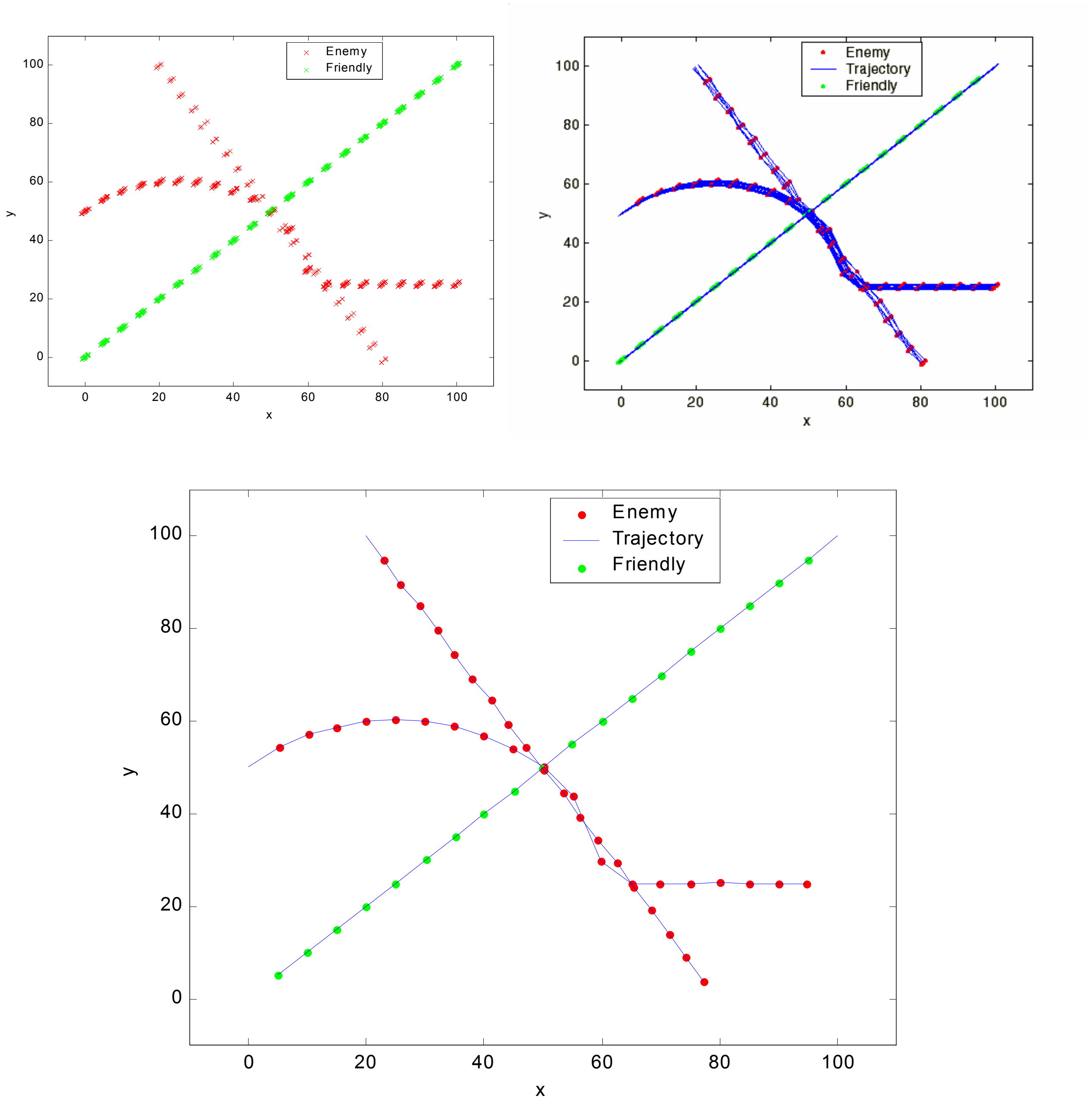


Fig 12: These figures illustrate the extraction of a diagrammatic representation from the identities, locations and velocities of individual entities. This synthesized data set is devoid of noise and we have emphasized similarity in velocity as much as proximity in grouping the individual units into meaningful groups. The figure on the top left shows the locations of the friendly and enemy units over a sequence of 21 frames. The one on the top right shows the trajectories of the individual units over the same sequence of 21 frames. The one at the bottom shows the diagram obtained by following the motions of the centers of the grouped units. The distinct lines of motion in the resultant diagram especially during intersections manifests the robustness of the proposed algorithm and the benefits of using the velocity information.

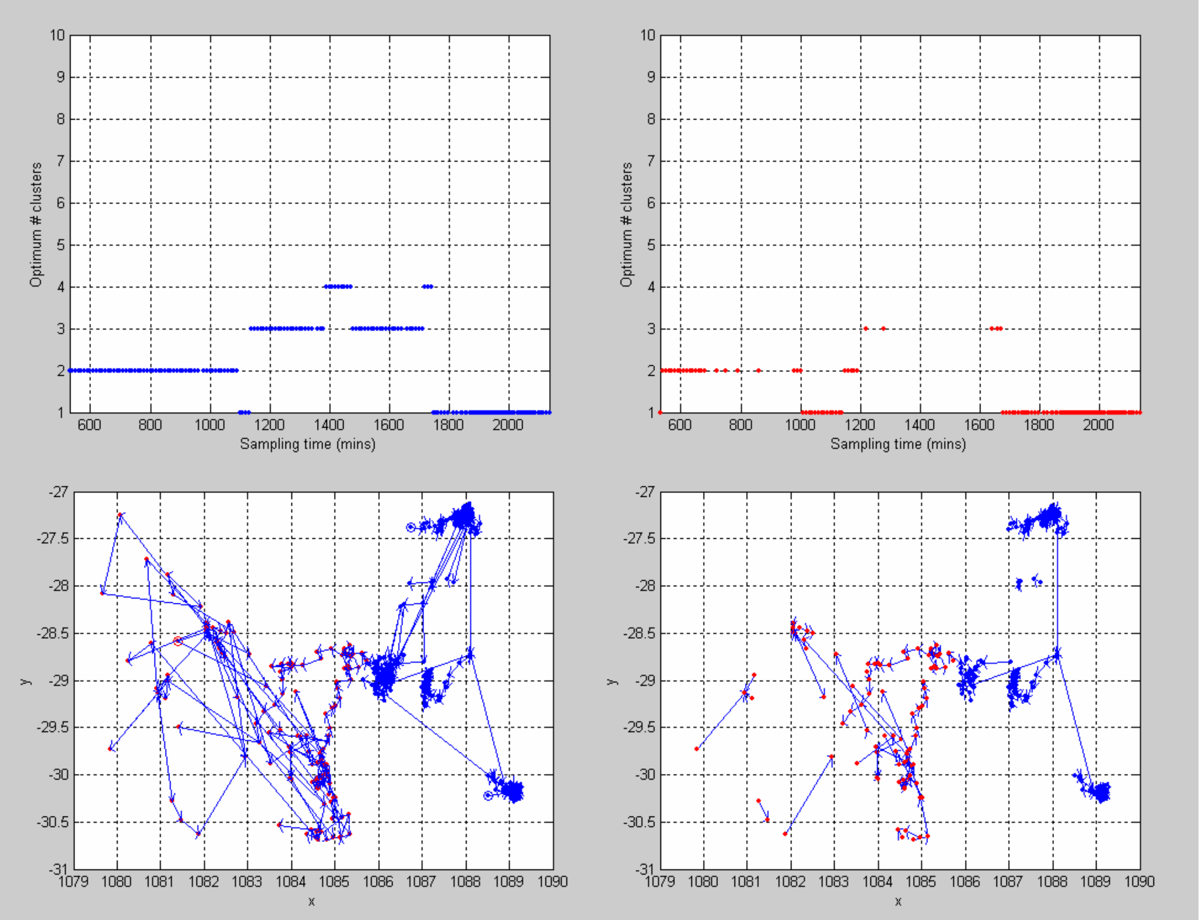


Fig 13a: The above figures illustrate the result of using the proposed algorithm for extracting diagrammatic representations from the identities and locations of a large number of individual military units at the best level of abstraction. The figure on the top left shows the consistency of determining the number of clusters that our algorithm produces for the friendly units only, over a duration of fifteen hours. The figure on the top right shows the consistency of determining the number of clusters for the enemy units only, over the same duration. The consistency is lesser in case of the enemy units because of more unreliability in the data. The figure on the bottom left is the diagram obtained by following the motions of the centers of the groups of military units for fifteen hours of operation without using their identity information while the one on bottom right shows the same but using the identity information of the individual military units.

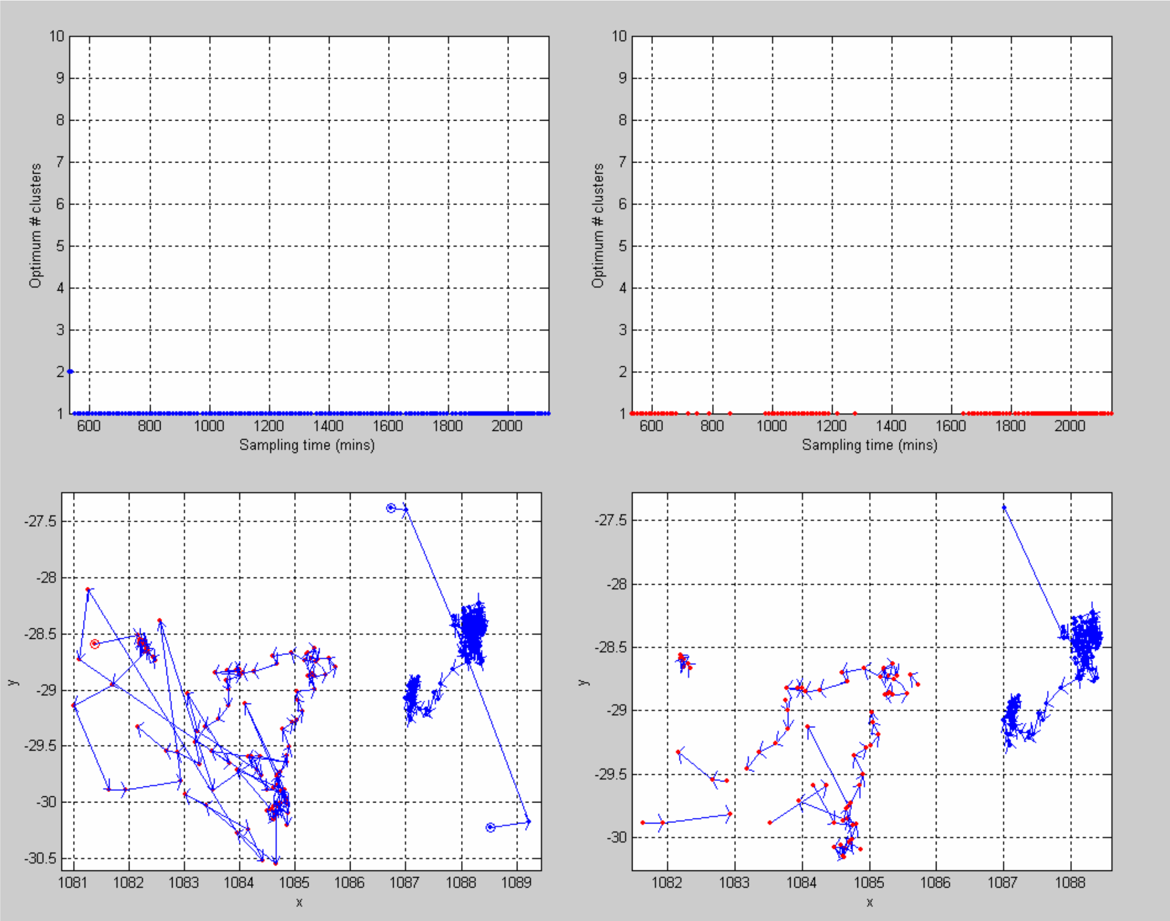


Fig 13b: The above figures illustrate the effect of using similarity in velocity as a criterion for grouping the individual military units into significant groups. The figure on the top left shows the consistency of determining the number of clusters that our algorithm produces for the friendly units only, over a duration of fifteen hours. The figure on the top right shows the consistency of determining the number of clusters for the enemy units only, over the same duration. The figure on the bottom left is the diagram obtained by following the motions of the centers of groups of military units for fifteen hours of operation without using their identity information while the one on bottom right shows the same but using the identity information of the individual military units. These results have been obtained from the same ARL data set as in Fig 13a but by equally emphasizing similarity of velocity and proximity. Note that unlike in Fig 13a, the optimum grouping hypothesis versus sampling time plot in the top left figure consistently yields a one-group hypothesis except for the first one or two sampling instants. This is because of the incomplete nature of the velocity information which is mainly due to the inability of the tracking system to track the GPS signals from the military units at each sampling instant. Also, note that unlike in Fig 13a, the splitting and merging of the groups cannot be identified anymore.

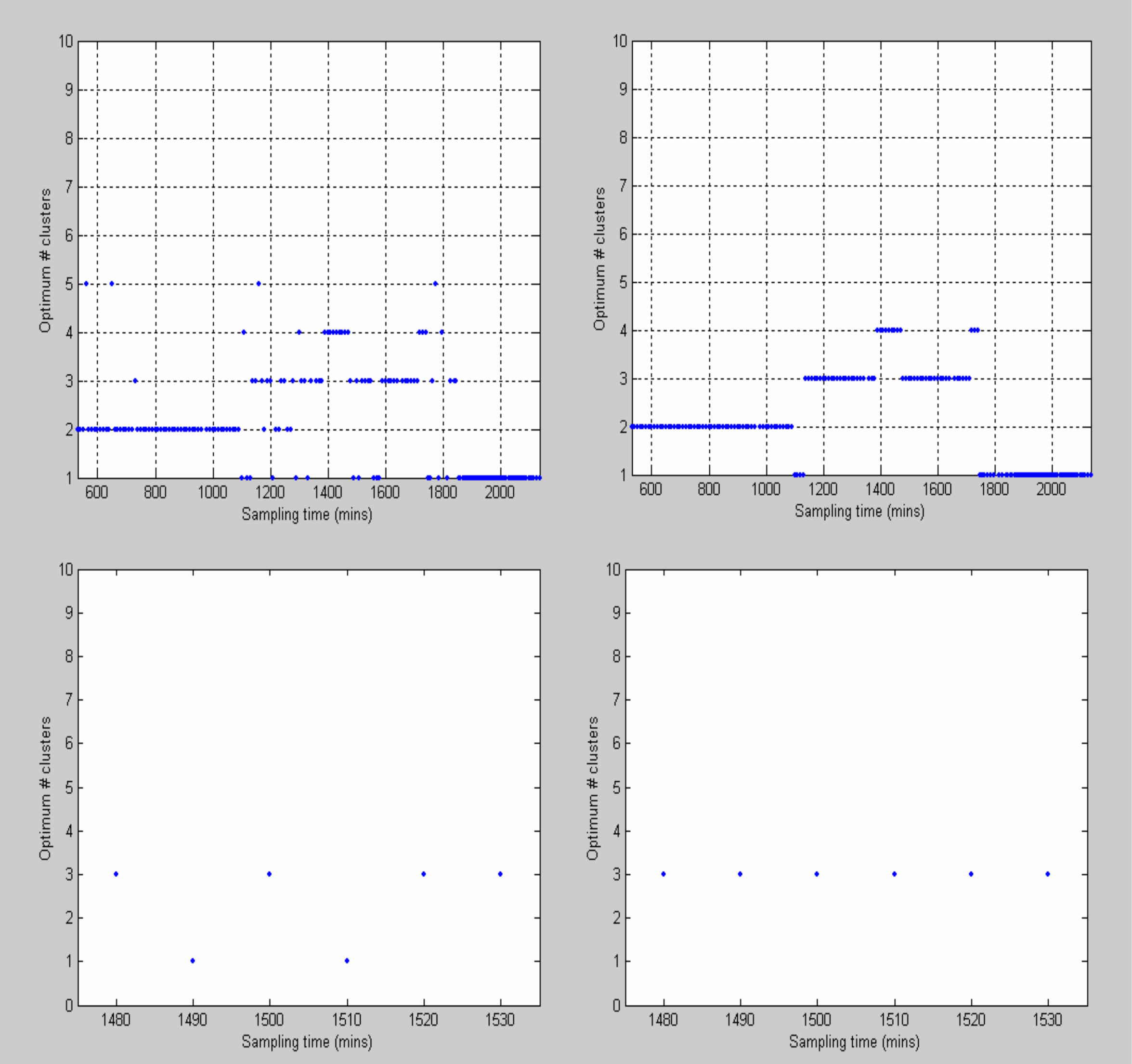


Fig 14a: These figures illustrate the optimum number of clusters obtained by deploying the proposed grouping algorithm on the ARL data “n941a111”, the same used in Fig 13a,b. The results shown above are obtained from the friendly units only. The figure on the top left shows the optimum groupings versus sampling time before applying abductive inference techniques to ensure consistency while the one on the top right shows the same after using abductive inference. The figures on the bottom left and right show the same but only for a duration of 50 minutes of sampling time.

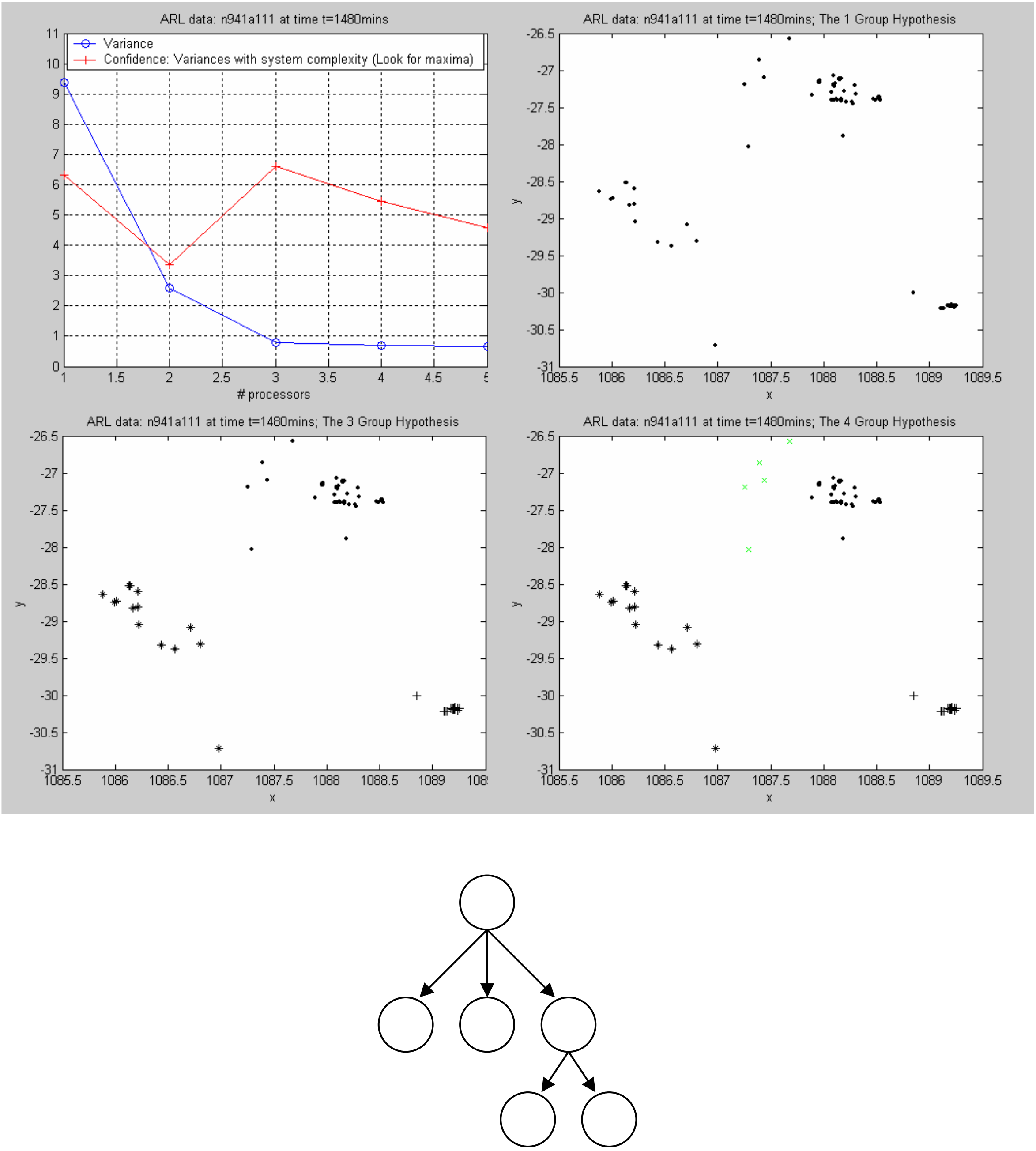


Fig 14b: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right, bottom left and bottom right show the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1480 mins using only the friendly units of the ARL data set “n941a111”. Note that the three-group hypothesis i.e. the hypothesis at the second level of organization enjoys highest confidence.

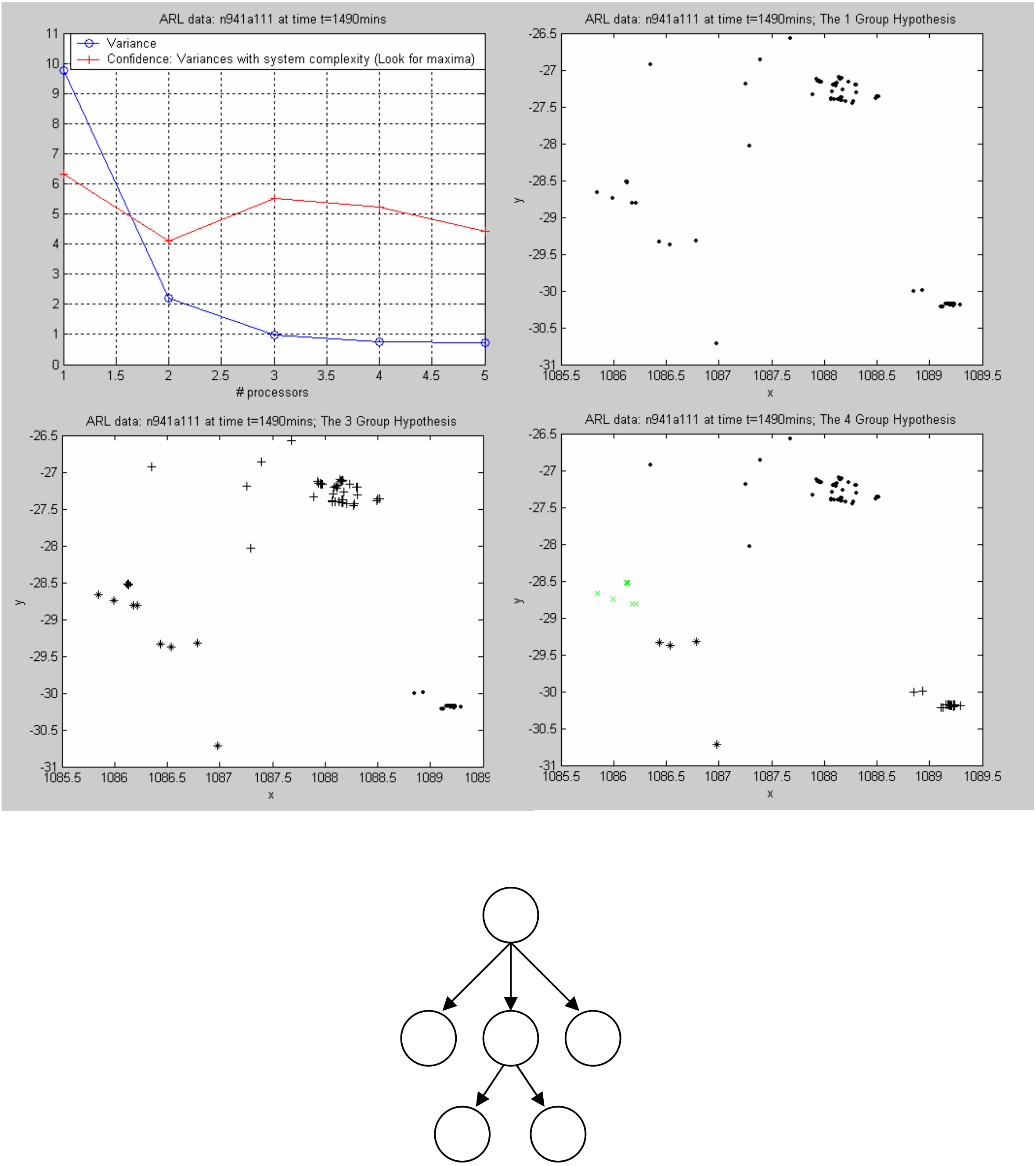


Fig 14c: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right, bottom left and bottom right show the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1490 mins using only the friendly units of the ARL data set “n941a111”. Note that the one-group hypothesis i.e. the hypothesis at the first level of organization enjoys highest confidence.

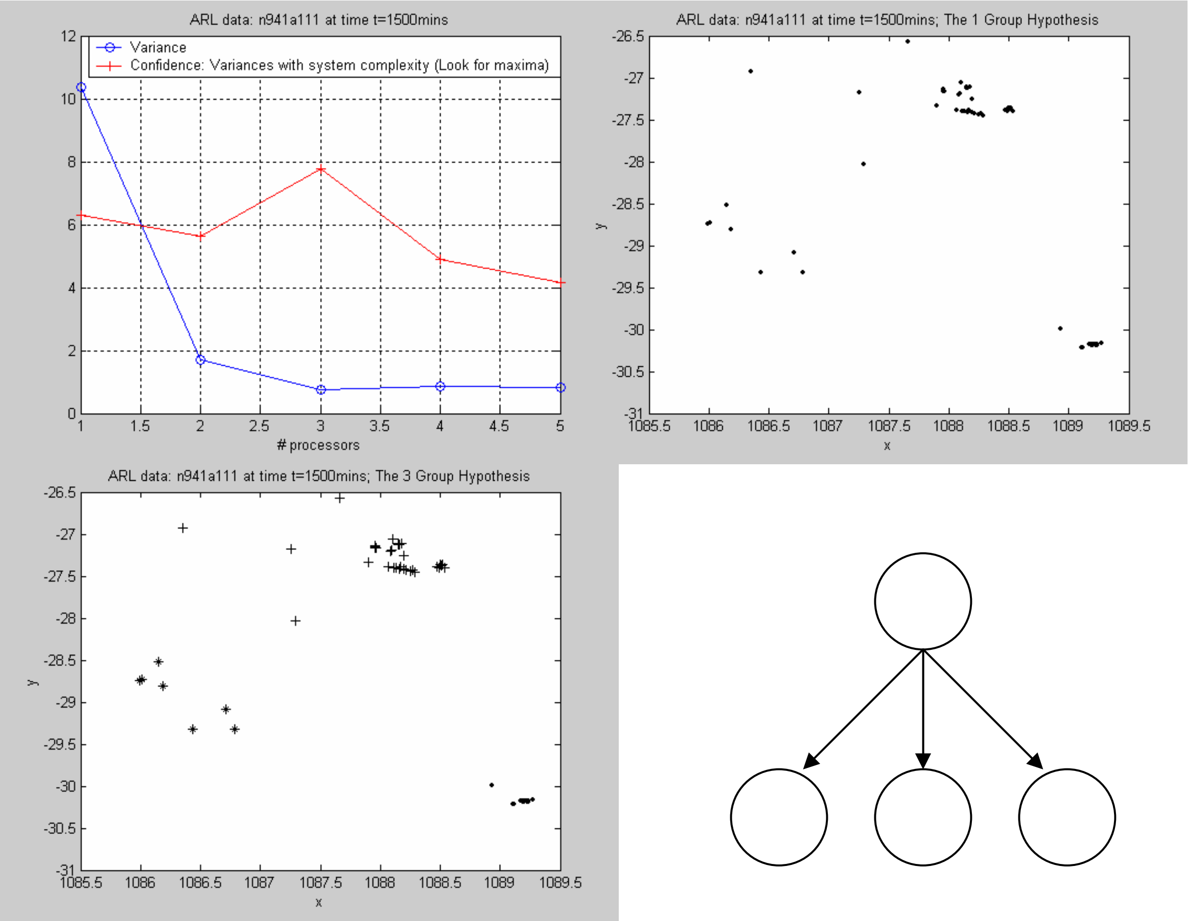


Fig 14d: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right and bottom left illustrate the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom right was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1500 mins using only the friendly units of the ARL data set “n941a111”. Note that the three-group hypothesis i.e. the hypothesis at the second level of organization enjoys highest confidence.

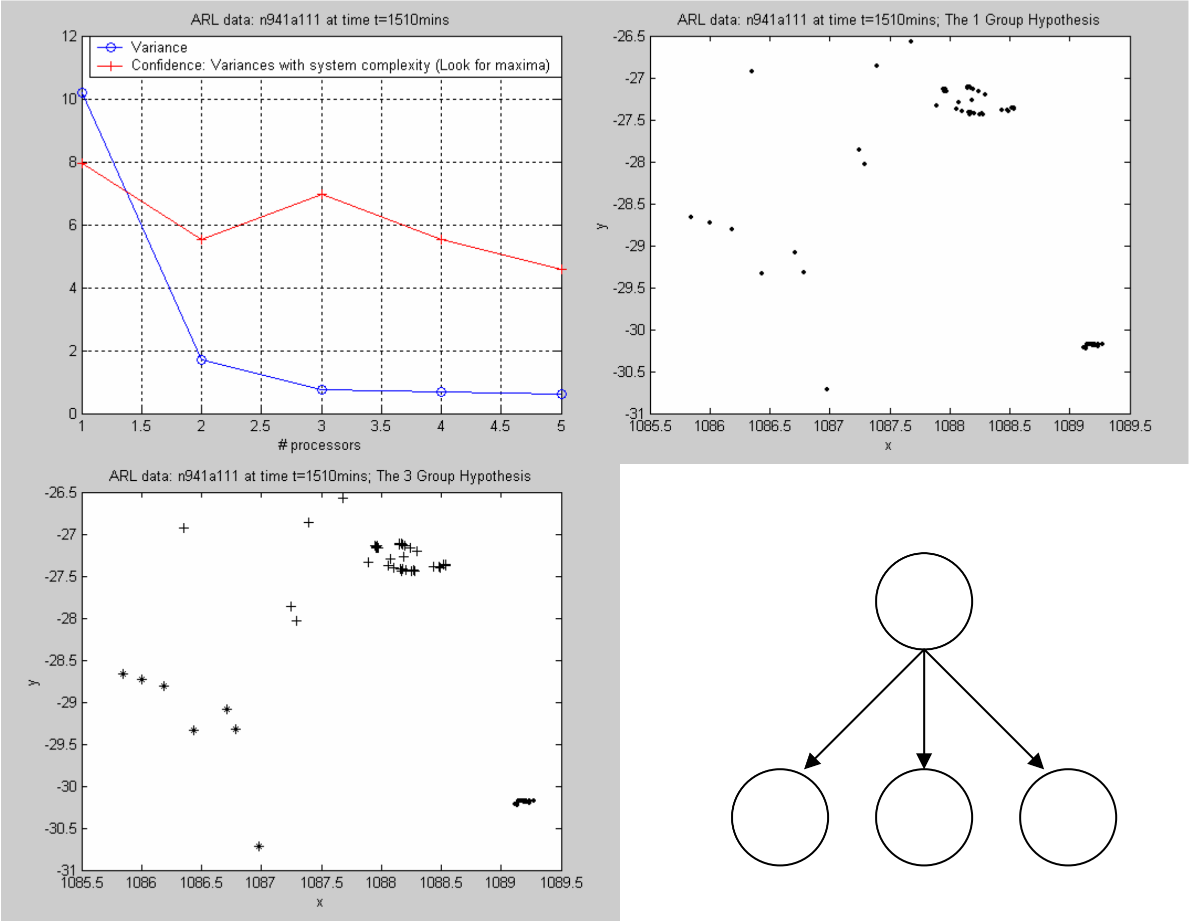


Fig 14e: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right and bottom left show the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom right was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1510 mins using only the friendly units of the ARL data set “n941a111”. Note that the one-group hypothesis i.e. the hypothesis at the first level of organization enjoys highest confidence.

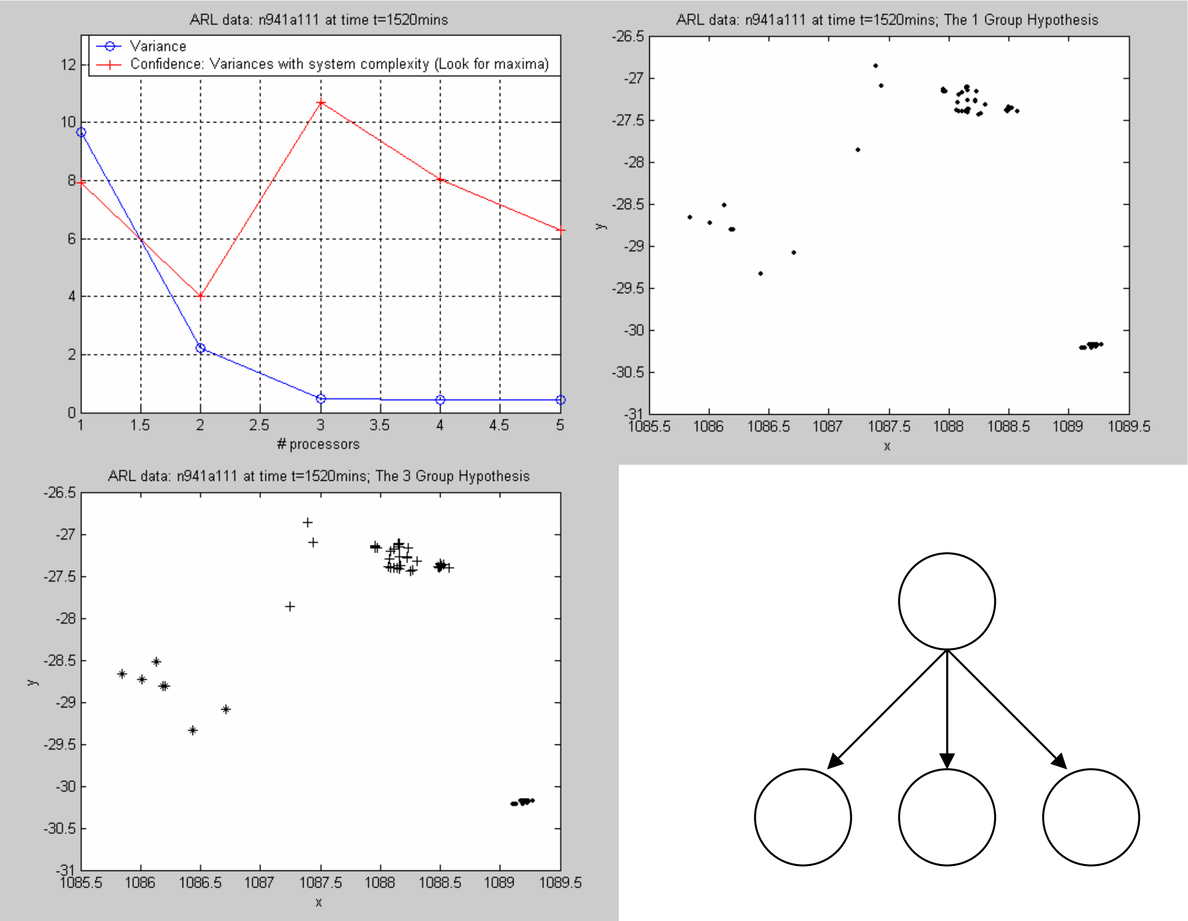


Fig 14f: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right and bottom left show the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom right was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1520 mins using only the friendly units of the ARL data set "n941a111". Note that the three-group hypothesis i.e. the hypothesis at the second level of organization enjoys highest confidence.

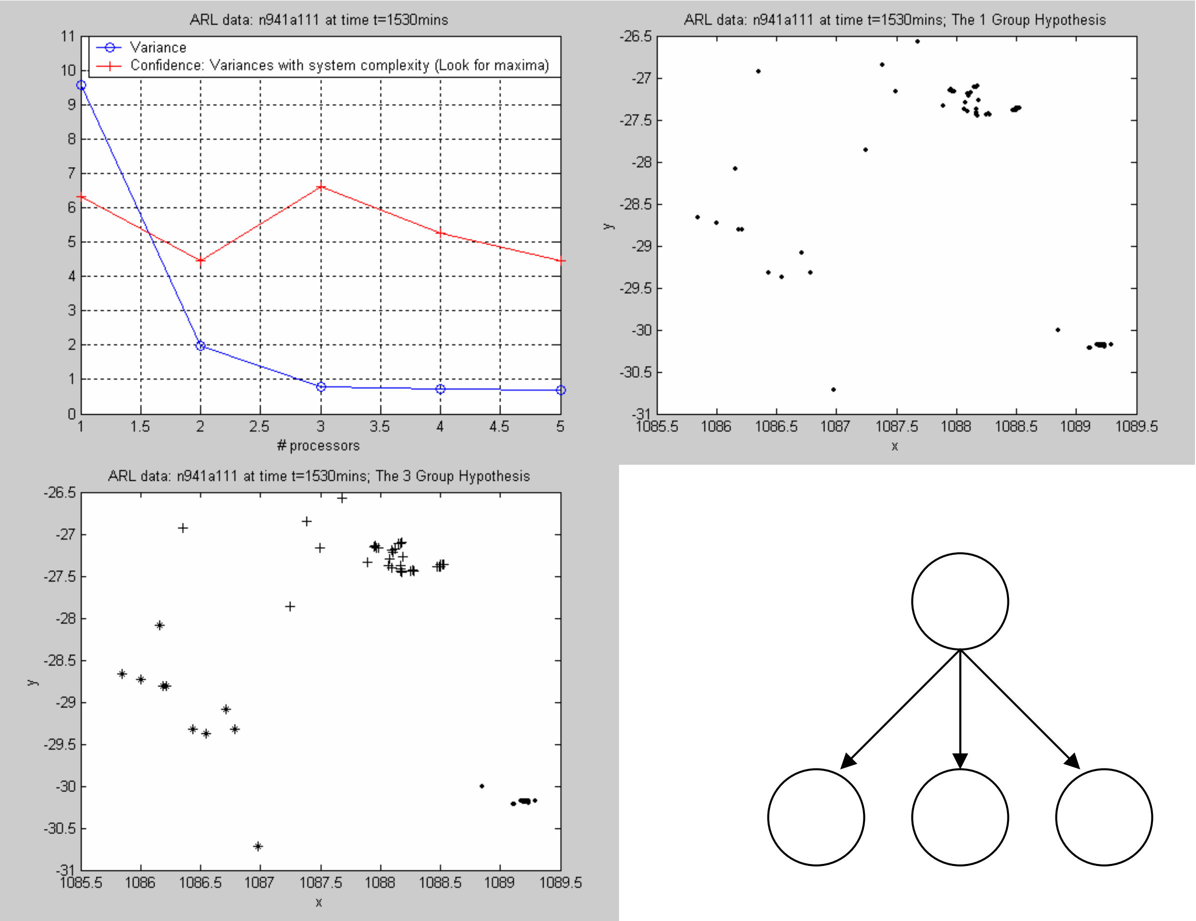


Fig 14g: The above figures illustrate the procedure of the formation of grouping hypotheses at multiple levels of organization. The figure on the top left shows the plot of confidence versus number of processors or clusters. The ones on the top right and bottom left show the memberships of the grouping hypotheses at each level of organization. The tree-like structure at the bottom right was adjudged the best resultant hypothesis taking all the levels of organization into consideration for time instant t=1530 mins using only the friendly units of the ARL data set “n941a111”. Note that the three-group hypothesis i.e. the hypothesis at the second level of organization enjoys highest confidence.

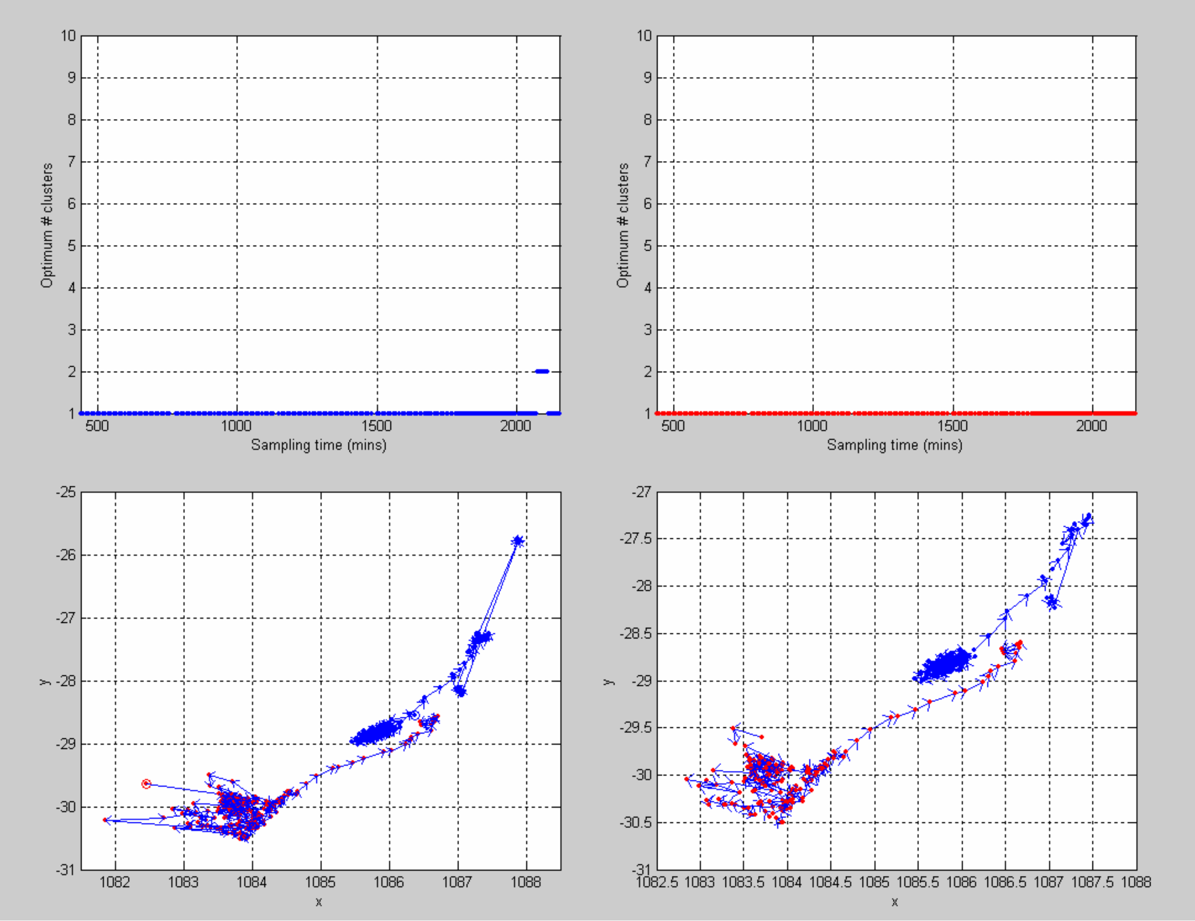


Fig 15: The above figures illustrate the result of using the proposed algorithm for extracting diagrammatic representations from the identities and locations of a large number of individual military units at the best level of abstraction. The figure on the top left shows the consistency of determining the number of clusters that our algorithm produces for the friendly units only, over a duration of twenty eight hours. The figure on the top right shows the consistency of determining the number of clusters for the enemy units only, over the same duration. The figure on the bottom left is the diagram obtained by following the motions of the centers of groups of military units for twenty eight hours of operation without using their identity information while the one on bottom right shows the same but using the identity information of the individual military units.

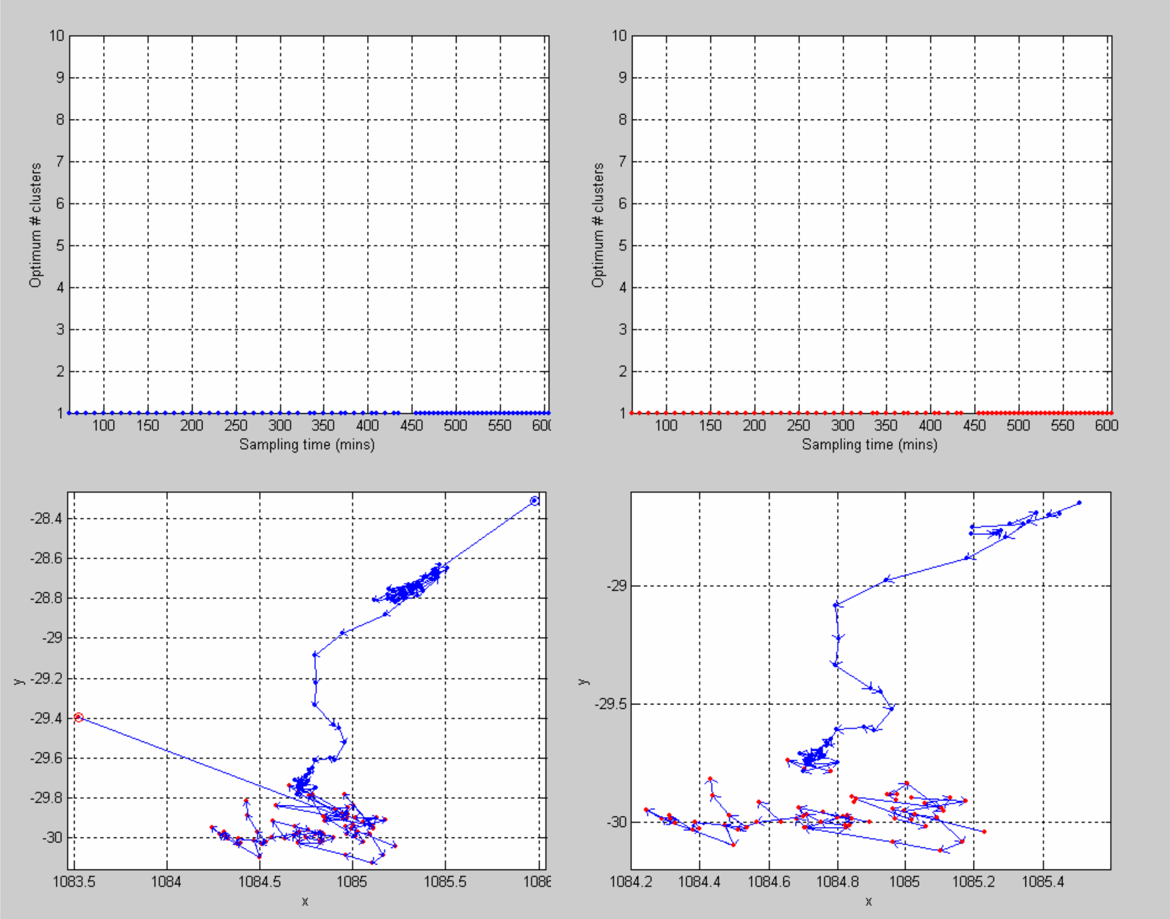
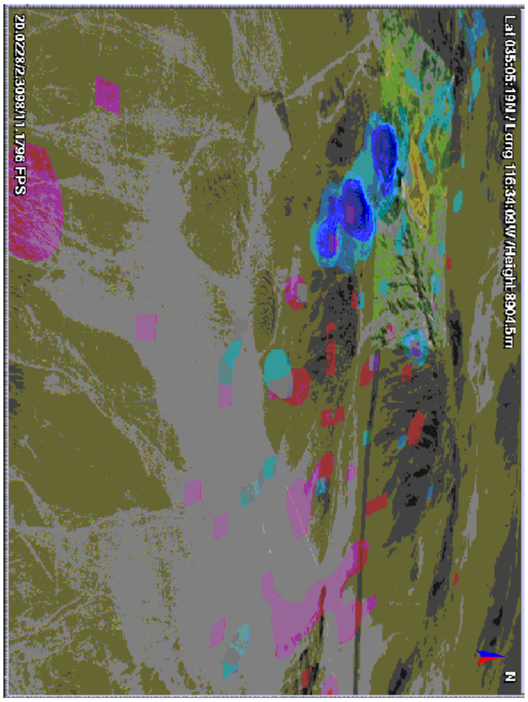
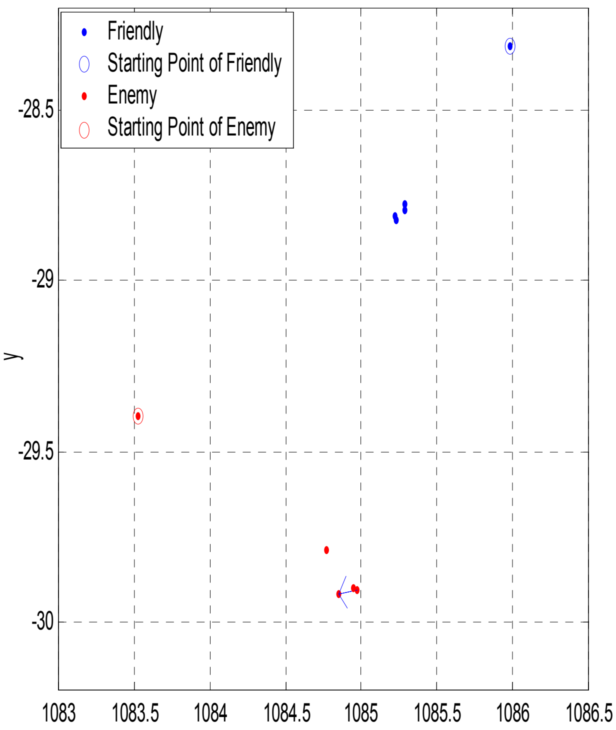
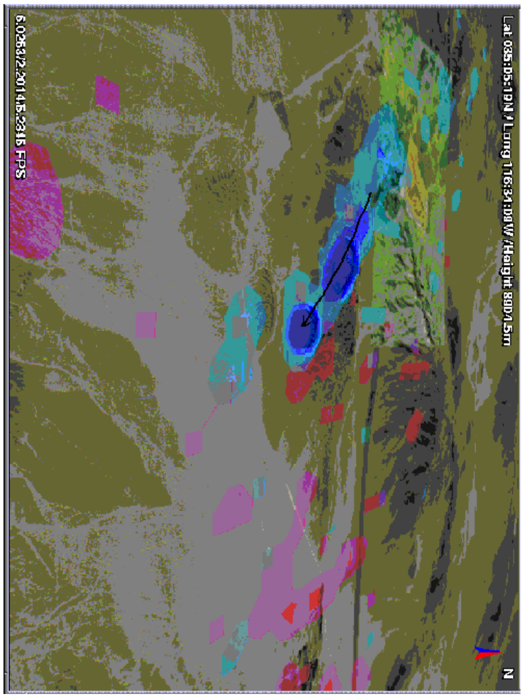
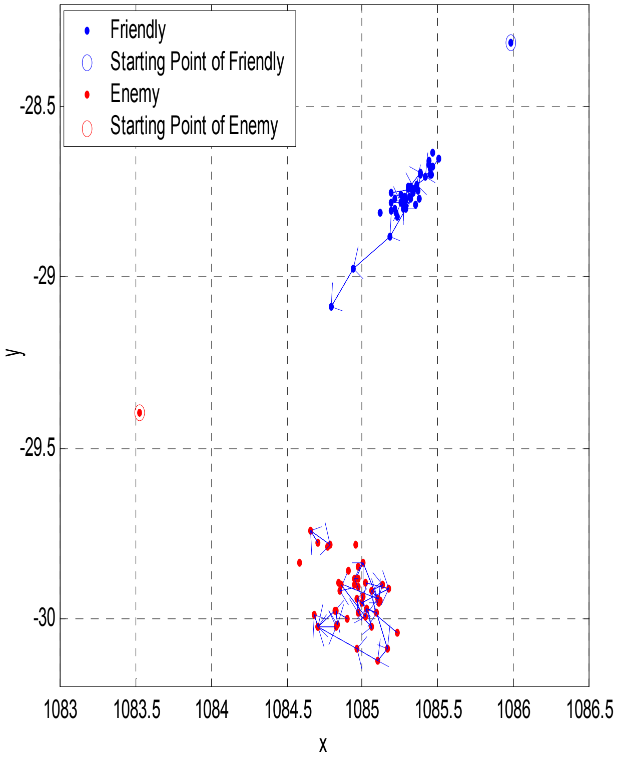


Fig 16: The above figures illustrate the result of using the proposed algorithm for extracting diagrammatic representations from the identities and locations of a large number of individual military units at the best level of abstraction. The figure on the top left shows the consistency of determining the number of clusters that our algorithm produces for the friendly units only, over a duration of nine hours. The figure on the top right shows the consistency of determining the number of clusters for the enemy units only, over the same duration. The figure on the bottom left is the diagram obtained by following the motions of the center of the groups of military units for nine hours of operation without using their identity information while the one on bottom right shows the same but using the identity information of the individual military units.


Lat 035:05:19N / Long 116:34:09W /Height 8904.5m
20.0228/2.3098/11.1796 FPS
N


Friendly
Starting Point of Friendly
Enemy
Starting Point of Enemy
-28.5
-29
-29.5
-30
y
1083
1083.5
1084
1084.5
1085
1085.5
1086
1086.5


Lat 035:05:19N / Long 116:34:09W /Height 8904.5m
6.0263/2.2014/5.2316 FPS
N


Friendly
Starting Point of Friendly
Enemy
Starting Point of Enemy
-28.5
-29
-29.5
-30
y
1083
1083.5
1084
1084.5
1085
1085.5
1086
1086.5
x

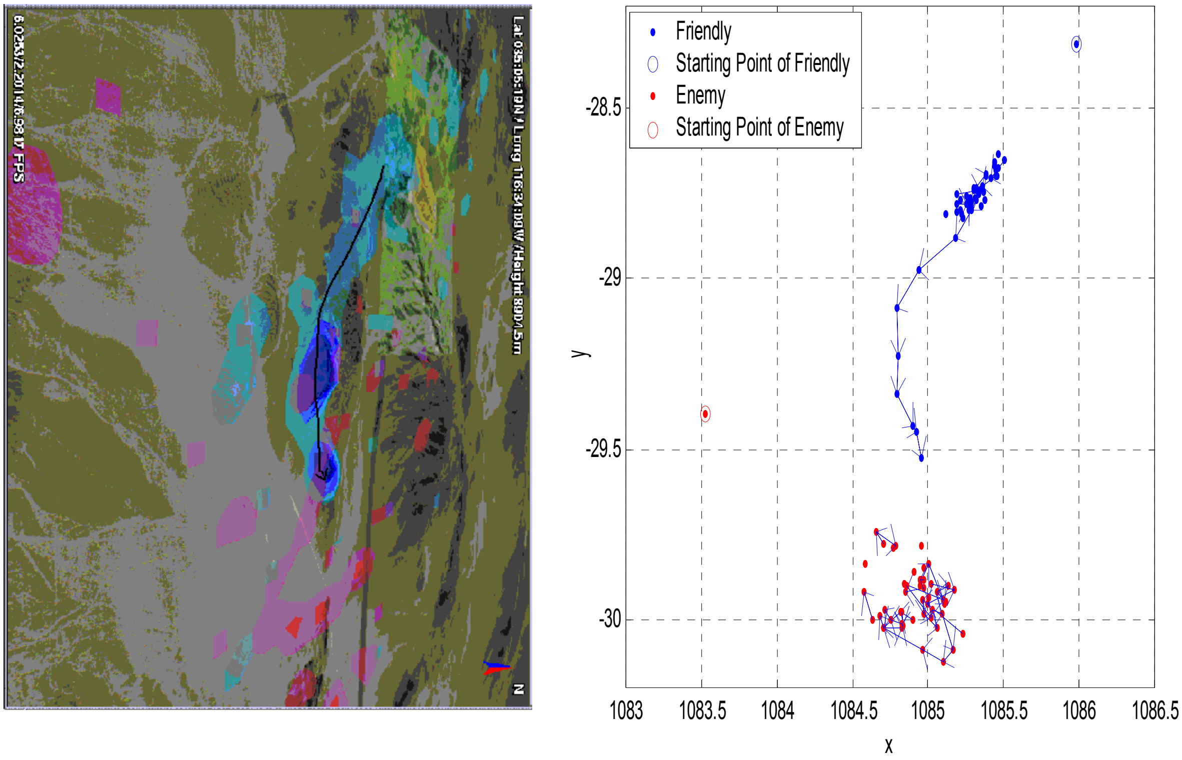


Fig 17: This figure compares between the diagrams extracted by deploying the proposed architecture and that drawn by LTC Gumbert on his visit to LAIR using the same data set as in Fig 16. The figures on the left are the ones drawn by the colonel laid on a terrain map. The patches are the enemy and friendly blobs; blobs being a kind of abstraction to facilitate visualization [JW, RW, PE]. The intensities of the blobs depict the density of military units. The figures on the right are the diagrams of motions of significant groups obtained from our diagrammatization architecture. It is noteworthy that the lines of motion in the diagram extracted by our system follow the same path as the lines of motion drawn by an expert in the field of military maneuvers using his background knowledge after knowing the actual facts that happened in the battlefield during the maneuver. This maneuver has been identified as a *frontal attack* by the colonel which is clearly evident from our extracted diagram.

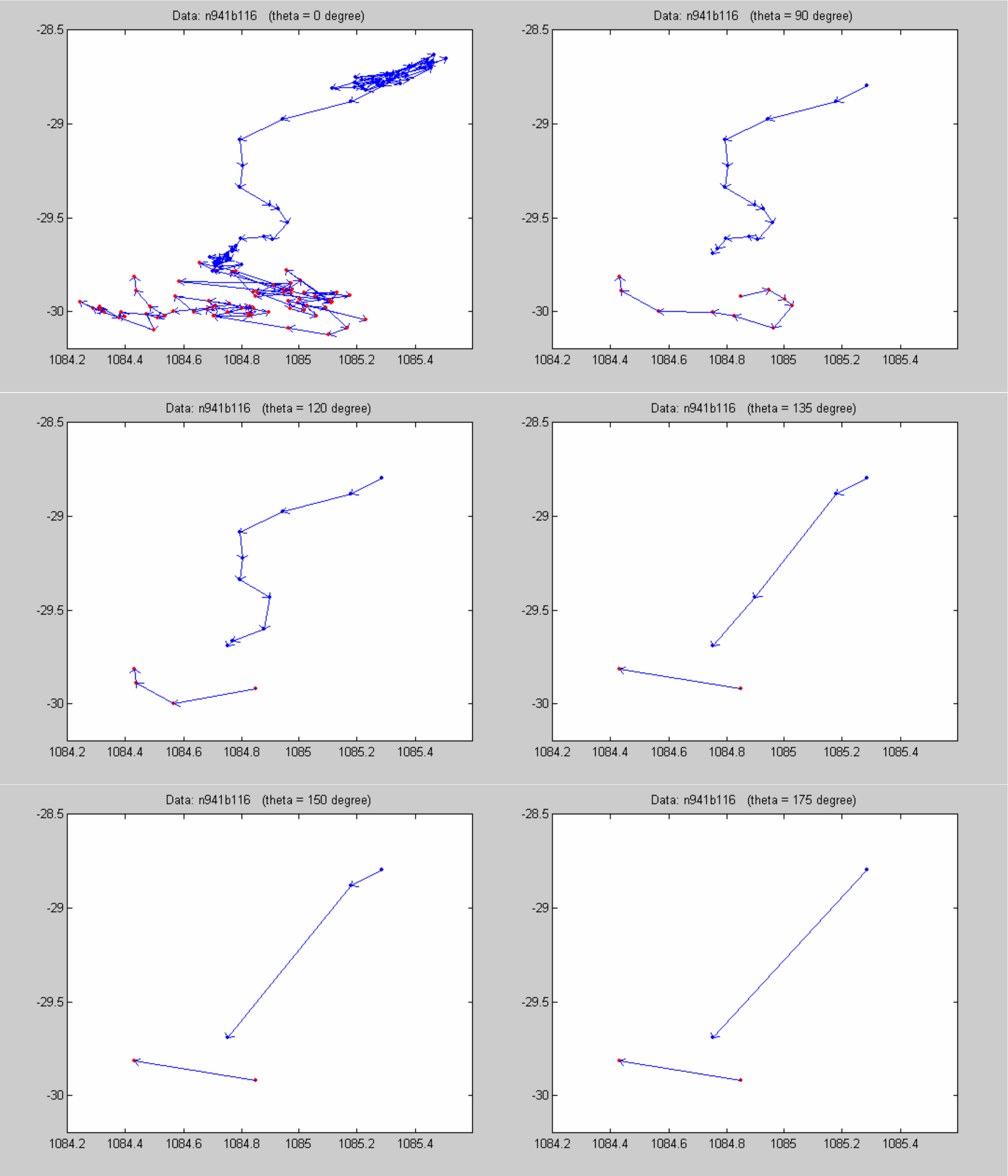


Fig 18: These figures show the output at different levels of abstraction of the diagram extraction system after pruning away the lines of motion that do not contribute to the recognition of maneuvers. The top left figure shows the crude output while the others depict the motions of the groups after abstracting away the unnecessary zigzag motions, not taking any domain knowledge and terrain information into consideration.

REFERENCES

[AB et al, 2001] A. Ben-Hur, D. Horn, H. T. Siegelmann and V. Vapnik, “Support Vector Clustering”, *Journal of Machine Learning Research*, Vol. 2, pp. 25-137, 2001.

[AC, 1998] A. Chmielecki, “What is information?”, *The Paideia Archive, Philosophy and Cognitive Science, 20th World Congress of Philosophy*, Boston, MA, August, 1998.

[AD et al, 1997] A. Datta, T. Pal and S. K. Parui, “A modified self-organizing neural net for shape extraction”, *Neurocomputing*, Vol.14, pp. 3-14, 1997.

[AD & SP, 1998] A. Datta and S.K. Parui, “Shape extraction: a comparative study between neural network based and conventional techniques”, *Neural Computing and Applications*, Vol.7, pp.343-355, 1998.

[AD et al, 1999] A. Demiriz, K. P. Bennett and M. J. Embrechts, “Semi-Supervised Clustering Using Genetic Algorithms”, In C. H. Dagli et al. (Eds.), *Intelligent Engineering Systems Through Artificial Neural Networks,* Vol. 9, pp. 809-814, New York, NY: ASME Press, 1999.

[AU & CV, 1994] A. Ultsch and C. Vetter, “Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark”, University of Marburg, FG Neuroinformatik & Kuenstliche Intelligenz, *Research Report 0994*, 1994.

[AU, 1996] A. Ultsch, “Self Organizing Neural Networks perform different from statistical k-means clustering”, In M. van der Meer, R. Schmidt, G. Wolf (Eds.): *BMBF Statusseminar Künstliche Intelligenz, Neuroinformatik und Intelligente Systeme*, pp. 433-443, München, April 1996.

[BB et al, 2000] B. Banerjee, A. Konar and S. Mukhopadhyay, “A Neuro-GA approach for the navigational planning of a mobile robot”, *Proc. Int. Conference on Computers, Communications and Devices (ICCCD)*, India, Dec. 2000.

[BB, 2002] B. Banerjee, “An Information Extracting Self-Organizing Neural Network”, *Proc. IEEE Annual Convention and Exhibition*, India, Dec. 2002.

[BB thesis, 2002] B. Banerjee, “Generating Diagrams of Group Motions from Locations and Movements of Individuals”, *Master’s Thesis* in Electrical Engineering, The Ohio State University, Columbus, August 2002.

[BB, 2003] B. Banerjee, “A self-organizing auto-associative network for the generalized physical design of microstrip patches”, *IEEE Trans. Antennas Propagat.*, April, 2003 [in press].

[BC et al, 1993] B. Chandrasekaran, N. H. Narayanan, and Yumi Iwasaki, "Reasoning with diagrammatic representations", *AI Magazine*, 14:2, pp. 49-56, Summer 1993.

[BC & NN, 1993] B. Chandrasekaran and N. H. Narayanan, "Perceptual representation and reasoning", in *Intelligent Systems: Concepts and Applications*, editor: Leon S. Sterling, Plenum Press, New York and London, pp. 67-76, 1993.

[BC, 1997] B. Chandrasekaran, “Diagrammatic Representation and Reasoning: Some Distinctions”, invited paper presented at the *AAAI Fall 97 Symposium Series*, Diagrammatic Reasoning, Boston, MA.

[BC & JJ, 2001] B. Chandrasekaran and J. R. Josephson, "Integrating Diagrammatic and Linguistic Representations in Problem-Solving", *Research Task proposal, Army Research Labs Collaborative Technology Alliance on Advanced Decision Architectures*, Oct 2001.

[BC, 2002] B. Chandrasekaran, "What does it mean for a computer to do diagrammatic reasoning?", keynote talk given at the *Diagrammatic Reasoning 2002 Conference*, Callaway Gardens, GA, April 18, 2002.

[BF, 1991] B. Fritzke, “Let it grow - self-organizing feature maps with problem dependent cell structure”, *In Kohonen T et al. (eds.), Artificial Neural Networks*, Vol.1, North-Holland, 1991.

[BM, 1996] B. Mirkin, *Mathematical Classification and Clustering*, Kluwer Academic Publishing, Dordrecht, The Netherlands, 1996.

[CS, 1948] C. E. Shannon, “A mathematical theory of communication”, *Bell System Technical Journal,* Vol.27, pp. 379-423 and 623-656, July and October, 1948.

[DC & SP, 1994] D. Choi and S. Park, “Self-creating and organizing neural networks”, *IEEE Neural Networks*, Vol.5, pp.561-575, 1994.

[FM 101-5] FM 101-5, *Operational Terms and Graphics*, Chapter 3, Graphic Control Maneuvers.

[GC & SG, 1986] G. A. Carpenter and S. Grossberg, “Neural Dynamics of Category Learning and Recognition: Attention, Memory Consolidation, and Amnesia”, *in J. Davis, R. Newburgh and E. Wegman (Eds.) Brain Structure, Learning and Memory*, AAAS Symposium Series, 1986.

[JD & AM, 1988] J. J. DeGruijter and A. B. McBratney, “A modified fuzzy k means for predictive classification”, In H. H. Bock (ed) *Classification and Related Methods of Data Analysis*, pp. 97-104, Elsevier Science, Amsterdam, 1988.

[JJ & SJ, 1994] J. R. Josephson and S. G. Josephson, *Abductive Inference: Computation, Philosophy, Technology*, New York, Cambridge University Press, 1994.

[JK et al, 1990] J. A. Kangas, T. Kohonen and J. Laaksonen, “Variants of self-organizing maps”, *IEEE Trans. Neural Networks*, Vol.1, pp. 93-99, 1990.

[JM, 1967] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations”, In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability* (ed. L. M. MeCam and J. Neyman), Vol. 1, pp. 281-297, University of California Press, Los Angeles, CA, 1967.

[JP & PL, 1999] J. A. L. J. M. Pena and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern Recognition Letters*, Vol. 20, pp. 1027-1040, 1999.

[JS et al, 2002] J. A. K. Suykens, J. De Brabanter, L. Lukas and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation", *Neurocomputing*, 2002 [in press].

[JW, RW, PE] J.D.Walrath, R.P.Winkler and P.J.Emmerman, “Visualization technique for improved situation awareness”, U.S. A.R.L.

[LB & YB, 1995] L. Bottou and Y. Bengio, “Convergence properties of the k-means algorithms”, In G. Tesauro and D. Touretzky (Eds.), *Advances in Neural Information Processing Systems,* Vol. 7, pp. 585-592, The MIT Press, Cambridge, MA, 1995.

[LB, 2001] L. Breiman, “Statistical Modeling: The Two Cultures”, *Statistical Science*, Vol.16, No.3, pp. 199-215, August 2001.

[MS & AM, 1993] M. Sabourin and A. Mitiche, “Modeling and classification of shape using a Kohonen associative memory with selective multiresolution”, *Neural Networks*, Vol.6, pp. 275-183, 1993.

[NN & BC, 1991] N. H. Narayanan and B. Chandrasekaran, "Reasoning Visually About Spatial Interactions", *Proc. of the 12th Int. Joint Conference on Artificial Intelligence*, Sydney, Australia, August 1991, pp. 360-365.

[PE et al, 1998] P. J. Emmerman, J. D. Walrath, and R. P. Winkler, “Making Complex, Multidimensional Battlefield Information Intuitive,” 21st Army Science Conference, Norfolk, VA, 1998.

[PH & BJ, 1997] P. Hansen and B. Jaumard, “Cluster analysis and mathematical programming”, *Mathematical Programming*, Vol. 79, pp. 191-215, 1997.

[RL, 1987] R. P. Lippmann, "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, April, pp. 4-22, 1987.

[RT, 2000] R. Tibshirani, G. Walther and T. Hastie, “Estimating the number of clusters in a dataset via the Gap statistic”, Technical Report, March 2000 [Published in JRSSB2000].

[SR & PN, 1995] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Inc., 1995.

[TK, 1982] T. Kohonen, “Self-Organized Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, Vol. 43, pp. 59-69, 1982.

[TK, 1990] T. Kohonen, “The Self-Organizing Map”, *Proc. IEEE*, Vol. 78, No. 9, pp. 1464-1480, 1990.

[TK, 1995] T. Kohonen, *Self-organizing maps*, New York: Springer, 1995.

[TK et al, 1999] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “The Analysis of a Simple k-Means Clustering Algorithm”, *Technical Report: LAMP-TR-038/CAR-TRA-937/CS-TR-4098/CCR-9712379*, University of Maryland, College Park, 1999.