Assessing Feature Relevance On-line Using Differential Discriminative Diagnosis

Mahesh Saptharishi

mahesh@andrew.cmu.edu

Thesis

For the Degree of Master of Science

Department of Electrical and Computer Engineering

Carnegie Mellon University

5000 Forbes Ave.,

Pittsburgh, PA 15213

Advisors:

Dr. John B. Hampshire II

Dr. Pradeep K. Khosla

TABLE OF CONTENTS

1.0 Introduction

Philosophers, inventors and great minds alike have described human creations as extensions of the human body. From early tools to today's computers, this description holds true. Machine learning attempts to create the ultimate extension: The extension of the human mind. This inspires allusions to certain human capabilities as subjective metrics for assessing and designing the capabilities of learning algorithms. Artificial neural networks (ANNs) have gained significant acceptance as powerful concept learners over the recent years. Partly because of suggestions of biological plausibility, neural networks have often been explored as potential solutions to those problems humans and other animals instinctively solve. Speculations about biological motivations aside, the statistical framework that ANNs provide for learning is truly invaluable. Traditionally, neural networks have been trained off-line with a fair amount of training data. Depending on the objective function used for training, a neural network estimates certain statistics from the training data. The training data consists of labelled feature vectors, where each feature vector could potentially consist of both relevant and irrelevant features. Good preprocessing hopes to eliminate the irrelevant features from the set of feature vectors. *Is preprocessing to assess and eliminate irrelevant features enough to ensure good performance of a neural network*?

Consider the task of tracking a person in a crowd. People can successfully track even in the presence of occlusion, awkward motion and many other subtleties in the environment. People not only learn with an explicit teacher, they also "learn from experience." By observing patterns of motion in the scene and *discriminating features* on the person being tracked, people effectively learn to temporally correspond the person. The number of people in the scene, their positions, shadows, etc., are constantly changing. Yet, the person is tracked with reasonable accuracy. If there is a set of features that is unique to that person, humans can instantly key on those features and use it to track. Consider the problem of maneuvering through a busy sidewalk. People can quickly identify the best path to their destination and constantly adapt this path to match the changing environment. Much of this is done even though there

are many changes occurring that are irrelevant to the task at hand. A neural network based algorithm can potentially be faced with many irrelevant features as its input. Although part of this problem can be alleviated with good preprocessing, a changing environment could also change the relevance of a feature.

Most commonly used learning algorithms, including neural networks, are inductive learning systems. Good generalization requires that the training data represent future inputs well. A key assumption in many strong convergence proofs for neural network training algorithms is that there is a very large, if not infinitely large, training set. In practice, the amount of labelled training data is usually limited. An acceptable size for the training set depends heavily on the learning task at hand. The previous paragraph introduced the problem of a changing environment and its relationship to the relevance of features. Proper sampling of the environment is of the essence in order to build an adequate training set. A bias towards certain environmental conditions in the training set also biases the knowledge learned by the neural network. Computational learning theory provides a theoretical view of the relationship between the size of the training set and the desired variance of the error rate. The discussion so far assumes an off-line learning algorithm. Although an on-line learning algorithm can potentially see more training data while it is executing, the data is unlabeled. In order for the algorithm to learn, there has to be a way to generate a teaching signal that can lead to convergence to the right concept. There has been an increasing interest within machine learning to develop methods that combine labeled and unlabeled data for training. The goal is to create a self-adaptive system by combining supervised and unsupervised learning algorithms. Given a robust learning algorithm that can learn on-line with unlabeled data in addition to learning off-line, some of the training set size problems can be alleviated. The previous statement assumes that the on-line, self-supervised learning algorithm is capable of assessing its performance and providing the appropriate teaching signal to better learn the intended concept.

The hypothesis addressed in this thesis states that by observing the decisions made by a robust classifier, the relevance of each feature, given as input to the classifier, to the decision can be assessed. Additionally, the relevance assessments can be used to improve the performance of the classifier on-line and enhance the training process by enabling the use of unlabeled examples. The novel algorithm suggested for estimating the relevance of each feature is referred to as *Differential Discriminative Diagnosis*. *Differential Discriminative Diagnosis* assesses the relevance of each feature to the task by diagnosing the discrimination power of each feature based on the difference in network outputs to different inputs.

1.1 Outline of the Thesis

Section 2.0 briefly reviews some notable methods concerning feature extraction and relevance assessments. Section 2.0 also reviews some popular on-line learning strategies. Section 3.0 describes the proposed algorithm, *differential discriminative diagnosis* abstractly. Section 4.0 describes *differential discriminative diagnosis* as applied to the task of temporally corresponding moving objects. Section 5.0 describes using the proposed algorithm for retrieving similar images from a database and using it to learn from unlabeled examples. Conclusions and future work constitute sections 6.0 and 7.0 respectively.

2.0 Tools for Extracting Features and On-line Learning

2.1 Features, Concepts, Knowledge and Relevance

Consider the example of tracking a person in the crowded scene again. The term "*discriminating features*" was used in section 1.0 without a precise definition, but was clear from the context. Perceptual features such as "long hair", "red shirt" or "tall" immediately come to mind in the context of the tracking example. From the point of view of a learning algorithm, we need to consider the notion of an *atomic feature*, a value that defines shape, color, depth etc. If the algorithm is fed a set of *atomic fea-*

tures, say a grayscale picture of a scene taken by a CCD camera, each pixel value is an *atomic feature*. A collection of these *atomic features* create recognizable patterns (or perceptual features) that we term as “person”, “hair”, “red”, “tall”, etc. This notion of an *atomic feature* is referred to simply as a feature throughout this thesis. The set of features provided to the learning algorithm is arranged in a vector and this vector is referred to as the “feature vector.” A feature vector may represent an image (a collection of pixels) of a person, symptoms of a medical ailment, etc. A concept is defined as the mapping of the feature vector to a label or classification. The concept can be described by the learning strategy and the associated parameters. This definition follows from the standard concept learning framework in machine learning. A set of learned concepts can be considered as the knowledge possessed by a learning algorithm.

The mapping from feature vector to a concept may rely on all or some of the features in the feature vector. In most real-world learning tasks, the problem of extracting just the features applicable to learning the concept is not always easy. The algorithm has to deal with a few or possibly many features that may not be applicable to learning the concept. If an *atomic feature* contributes to the concept, it is a relevant feature. Some features contribute more than others to the concept. The extent of a feature’s contribution is quantified by its *relevance*. If the *relevance* of each feature in the feature vector was known, the information can be used to improve the performance of the classifier and enhance the training process.

2.2 Tools for Assessing the Relevance of Features

A fair number of pattern recognition techniques exist to select feature subsets, to improve discrimination between classes, and also to reduce the dimensionality of the feature vectors. This section mentions a few of these techniques. The first notable technique is Fisher’s linear discriminant [1]. This technique seeks to reduce the dimensionality of the feature vector while maximizing the separation

between classes. Fisher's linear discriminant offers a good indication of the separability of two classes by means of the Fisher Ratio [1]. Other techniques with similar objectives include the Fukunaga-Koontz [2] and the Foley-Sammon [3] transforms. Both these techniques seek to improve discrimination while reducing the size of the feature vector by means of transformations. Techniques also exist to explicitly reduce dimensionality, but do not consider discrimination explicitly. Often a dimensionality reduction step is followed by a test step to check if the reduced feature set increases or decreases discrimination performance. Notable techniques that fit this description include the *branch and bound* method [4], *sequential forward selection* and *sequential backward elimination* [5]. Given that we wish to reduce the feature set by a certain number of features, the *branch and bound* method searches through possible feature subsets using trees to find one that is optimal. Optimality, in this technique, is with respect to the algorithm that is used to test the discrimination performance of the reduced feature set. The *sequential forward selection* and *sequential backward elimination* techniques are both faster than the *branch and bound* method, but are suboptimal. They provide a relatively fast feature selection method. Principal components analysis [6] offers a systematic method for reducing the feature set by projecting the original feature vector on a new set of basis functions. The weights associated with this projection can be used as a new set of features. Principal components analysis does not consider discrimination in its projection. Thus, it could potentially be quite suboptimal if discrimination is the final objective. Independent components analysis [7] improves on the basic principal components analysis technique. Most of the techniques considered so far are off-line techniques.

In addition to the described methods, various other feature subset selection techniques exist. These techniques fall into either the *filter* or the *wrapper* categories [8]. Two good examples of the filter model include RELIEF [9] and FOCUS [10]. RELIEF seeks to estimate the level of relevance, a continuous valued weight, of each feature. FOCUS performs an exhaustive search to find the minimum set of relevant features needed for the machine learning task. The *wrapper* model [8, 11, 12], searches the feature space to find one that increases the estimated accuracy of the learning algorithm. The *wrapper*

model is similar in spirit to the *sequential forward selection* and the *sequential backward elimination* methods. Koller and Sahami [13] suggest an information theoretic feature selection method that relies on the Kullback-Leibler information distance [14]. Scherf and Brauer [15] suggest another filter-based technique, EUBAFES for feature selection. EUBAFES estimates binary weights for each feature used by a radial basis function neural network. Cherkauer and Shavlik [16] introduce the notion of *transparency*, which is related to the minimum description length principle (MDL) [17]. The transparency measure is used to estimate the quality of input representations for neural networks.

Most of these techniques rely on finding a good set of features for a particular learning task using the training data. The implicit assumption is that the estimated relevant features stay relevant for all future inputs and the irrelevant features stay irrelevant. *If a set of good features are selected, but their relevance changes, can the assessed change in relevance be used to improve the learning algorithm online?* This thesis is primarily concerned with the task of pattern discrimination using artificial neural networks. *Differential Discriminative Diagnosis* offers a systematic way to assess the relevance of each feature in the feature vector to a discrimination task on-line. The assessed relevance can be used to improve the performance of the neural network classifier that performs the discrimination.

2.3 On-line Learning

Assessing the relevance of features on-line is itself a learning task. Regression with filters can be considered as one form of on-line learning. In addition, clustering with algorithms such as k-means, c-means and EM also have been extended to learn while the learning algorithm is actually executing [18, 19, 20, 21, 22]. Reinforcement learning [23, 24, 25] is a popular technique for learning from experience. This technique takes a state-action perspective to learn. Active learning [26] strategies query the user during the learning process. The success of these techniques motivates the need for more on-line learning strategies that easily extend current neural network algorithms to learn, improve or diagnose themselves on-line.

3.0 Differential Discriminative Diagnosis

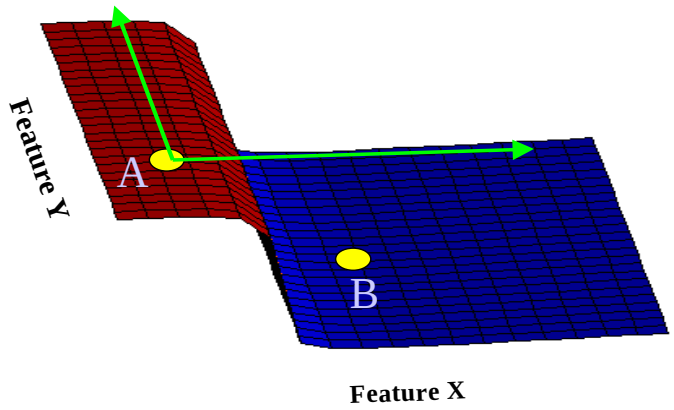
Consider the feature vector to decision mapping again. Specifically, consider the concepts learned by artificial neural networks to discriminate between classes. This section proposes a systematic method, referred to as *differential discriminative diagnosis,* for assessing the relevance of each feature in the feature vector to the discrimination concept learned. The task of assessing the relevance of features is posed as an online learning task. The estimated relevances are put to use in specific applications in sections 4 and 5.

3.1 The Optimization Perspective

Learning the parameters of an artificial neural network is an optimization problem. The parameters are optimized for a given training set with respect to the objective function or learning strategy. Estimating the influence of each parameter in the optimized network with respect to the objective function can be posed as a sensitivity analysis problem. Sensitivity analysis seeks to measure the effect of a perturbation of a particular parameter to the output or decision of the network. If the network is sensitive to a particular parameter, then that parameter contributes to the decision process and therefore is relevant. The Taylor series method for sensitivity analysis is one of the most popular techniques in optimization literature [27].

Figure 1 illustrates the notion of sensitivity and relevance. The surface plot shows the output space for a two-class, logistic linear classifier with a two dimensional input vector. As a feature vector moves along the X dimension, the classifier's decision changes from class 1 to class 2. Whereas, as the feature vector moves along the Y dimension, the classifier's decision stays constant. Feature X contributes to the classifier's decision, while feature Y does not influence the classifier. Thus, the feature X in each of the feature vectors A and B is relevant and feature Y is irrelevant.

FIGURE 1.



Feature vectors A and B consist of two features X and Y. A logistic linear classifier's decision boundary is shown as the surface plot. The classifier is sensitive to feature X.

3.1.1 Applications of Sensitivity Analysis to Neural Networks

In the context of neural networks, sensitivity analysis has been applied to prune unnecessary connections in the network. Specifically, Optimal Brain Damage [28] and Optimal Brain Surgeon [29] seek to estimate the saliency of each weight in the network to the learning task. Weights with relatively little salience are removed from the network and the network is then retrained with reduced connectivity. This strategy is quite successful in reducing the complexity of a neural network. Following Occam's Razor, if a less complex network learns the concept just as well as a more complex network, the network with lower complexity will probably generalize better [30].

A similar method of analysis can also be applied to assessing the relevance of features presented to a neural network. This technique is referred to as *discriminative diagnosis* [31] and was introduced in the context of fault detection and diagnosis. As with Optimal Brain Damage and Optimal Brain Surgeon, a Taylor series expansion of the classifier is used to determine the sensitivity of the feature to the decision. This notion can be formalized as in the original version of *discriminative diagnosis* as follows.

Let $C(\vec{X}_i)$ represent the output of the classifier $C(\ )$ to the input $\vec{X}_i$. The output produced by the function $C(\ )$ is a scalar value in accordance with the objective function. For example, in the case of the mean squared error objective function, $C(\vec{X}_i)$ is the mean squared error of the classification (When the algorithm is executing, the error is computed assuming that the classifier has made the right decision). Let us now consider another input, $\vec{X}_j$ where $i \neq j$ and $\vec{X}_j$ is of a different class than $\vec{X}_i$. The output of the classifier, $C(\vec{X}_j)$ to the new input $\vec{X}_j$ can be approximated using a second-order Taylor series expansion of the classifier around the input point $\vec{X}_i$. This approximation is shown in equation (1). $\boldsymbol{H}_{\vec{X}}$ here

$$C(\vec{X}_j) = C(\vec{X}_i + (\vec{X}_j - \vec{X}_i)) \cong C(\vec{X}_i) + (\vec{X}_j - \vec{X}_i)^T \nabla_{\vec{X}} C(\vec{X}_i) + \frac{1}{2}(\vec{X}_j - \vec{X}_i)^T \boldsymbol{H}_{\vec{X}} C(\vec{X}_i)(\vec{X}_j - \vec{X}_i) \tag{1}$$

represents the Hessian of $C(\ )$ with respect to the input $\vec{X}$. Given this form of the approximation, we can use it to approximate the difference in classifier outputs, $C(\vec{X}_i + (\vec{X}_j - \vec{X}_i)) - C(\vec{X}_i)$ in equation (2).

$$C(\vec{X}_i + (\vec{X}_j - \vec{X}_i)) - C(\vec{X}_i) \cong \underbrace{(\vec{X}_j - \vec{X}_i)^T \nabla_{\vec{X}} C(\vec{X}_i)}_{\text{first-order relevance}} + \underbrace{\frac{1}{2}(\vec{X}_j - \vec{X}_i)^T \boldsymbol{H}_{\vec{X}} C(\vec{X}_i)(\vec{X}_j - \vec{X}_i)}_{\text{second-order relevance}} \tag{2}$$

Equation (3) expands equation (2) where the relevance of each feature is the term within the outer sum-

$$= \sum_{k=1}^{m} \underbrace{(x_{j,k} - x_{i,k}) \left[\frac{\partial}{\partial x_{i,k}} C(\vec{X}_i) + \frac{1}{2} \sum_{l=1}^{m} \frac{\partial^2}{\partial x_{i,l}^2} C(\vec{X}_i)(x_{j,l} - x_{i,l}) \right]}_{\text{relevance of the feature } x_{i,k}} \tag{3}$$

mation. The index k and l iterate over each of the m features in the feature vectors $\vec{X}_j$ and $\vec{X}_i$. This technique was successfully used by Hampshire [31] in a fault detection and diagnosis task.

3.2 Estimating Feature Relevance With Memory

Discriminative diagnosis can be extended to make its relevance assessments with more than just two consecutive feature vectors. As the classifier sees and classifies more inputs, a variety of input vectors and classifications can be considered in estimating the relevance of a particular feature. Thus, we wish to consider feature vectors of different classes and also feature vectors of the same class. First, we define the vector $\hat{h}_{n,i,j} = \vec{X}_{n,j} - \vec{X}_{n-1,i}$ where n denotes a time instance, i and j represent distinct input vectors of the same or different classes. Additionally we define a diagonal matrix R that scales each parameter in the vector $\hat{h}_{n,i,j}$.We can now redefine equation (2) in equation (4). If R is the identity

$$C(\vec{X}_{n-1,i} + R\hat{h}_{n,i,j}) - C(\vec{X}_{n-1,i}) \cong \underbrace{(R\hat{h}_{n,i,j})^T \nabla_{\vec{X}} C(\vec{X}_{n-1,i})}_{\text{first-order relevance}} + \underbrace{\frac{1}{2}(R\hat{h}_{n,i,j})^T \boldsymbol{H}_{\vec{X}} C(\vec{X}_{n-1,i})(R\hat{h}_{n,i,j})}_{\text{second-order relevance}} \quad (4)$$

matrix, then equation (4) is identical to equation (2). If on the other hand, R has diagonal entries with values other than 1, then equation (4) is no longer the same as equation (2). We define $P_{n,i,j} = C(\vec{X}_{n-1,i} + R_i\hat{h}_{n,i,j}) - C(\vec{X}_{n-1,i})$, where R_i is not necessarily an identity matrix. Equation (5)

$$\frac{\partial P_{n,i,j}}{\partial r_{i,k,k}} = \underbrace{(x_{n,j,k} - x_{n-1,i,k})\left[\frac{\partial}{\partial x_{n,i,k}} C(\vec{X}_{n-1,i}) + \sum_{l=1}^{m} \frac{\partial^2}{\partial x_{n,i,l}^2} C(\vec{X}_{n-1,i}) r_{i,l,l}(x_{n,j,l} - x_{n-1,i,l})\right]}_{\text{relevance of the feature } x_{i,k}} \quad (5)$$

shows the derivative of $P_{n,i,j}$ with respect to each diagonal entry $r_{k,k}$ of the diagonal matrix R. Notice that equation (5) is almost the same as equation (4) (when R is the identity matrix) except for a constant multiplier of $\frac{1}{2}$ before the inner summation of the second-order terms.

3.2.1 The Relevance Differential

Discriminative diagnosis asks the question: "*How badly does the [feature vector $\vec{X}_j$] represent the [class of $\vec{X}_i$] compared to the [feature vector $\vec{X}_i$]*?" [31]. If $\vec{X}_j$ and $\vec{X}_i$ are of the same class, then we expect the two vectors to represent each other well. *Differential discriminative diagnosis* simultaneously considers a reference vector $\vec{X}_{n-1,\,i}$, a candidate vector $\vec{X}_{n,\,i}$ of the same class as the reference vector, and another candidate vector $\vec{X}_{n,\,j}$ of a different class. It is concerned with the difference between the *discriminative diagnosis* of each feature in the feature vectors. Thus, it asks the question: *Which features in $\vec{X}_{n,\,i}$, represent the reference vector's class well while not representing its class well in $\vec{X}_{n,\,j}$?* The answer to this question lies in maximizing the *relevance differential* $\Re_i$, defined in equation (6), with respect to the diagonal matrix R_i. The value of each diagonal entry in R_i reflects the relevance

$$\Re_i = \overbrace{\left|P_{n,\,i,\,j}\right|}^{\text{should have a large magnitude}} - \underbrace{\left|P_{n,\,i,\,i}\right|}_{\text{should be close to 0}},\ i \neq j \tag{6}$$

of each feature in the feature vectors $\vec{X}_{n,\,i}$ and $\vec{X}_{n,\,j}$. The effect induced on each diagonal term in R_i by maximizing equation (6) can be elaborated, based on the mechanics of the Taylor Series, as follows:

- Differences in $\hat{h}_{n,\,i,\,j}$ are "key" if they influence the decision of the classifier. Those key differences that are large in magnitude in $\hat{h}_{n,\,i,\,j}$, but are small in $\hat{h}_{n,\,i,\,i}$ are magnified. Key differences that are similar in magnitude in both $\hat{h}_{n,\,i,\,j}$ and $\hat{h}_{n,\,i,\,i}$ are suppressed.
- The predicted output of the classifier, $|P_{n,\,i,\,j}|$ to the input $\vec{X}_{n,\,j}$ with respect to $\vec{X}_{n-1,\,i}$ is made as inaccurate as possible. Thus, R_i magnifies key differences in the feature difference vector $\hat{h}_{n,\,i,\,j}$.
- The predicted output of the classifier, $|P_{n,\,i,\,i}|$ to the input $\vec{X}_{n,\,i}$ with respect to $\vec{X}_{n-1,\,i}$ is made as accurate as possible. Thus, R_i suppresses key differences in the feature difference vector $\hat{h}_{n,\,i,\,i}$.

- Certain differences in $\hat{h}_{n,i,j}$ could be key, but could also be unimportant in $\hat{h}_{n,i,i}$. In such a case R_i appropriately scales the differences.

The result of the maximization is that large magnitude values in R_i indicate relevant features and small magnitude values indicate irrelevant features. Note the difference between the relevance quantified in equation (3) and in equation (6).

3.2.2 Notes on Computations for Differential Discriminative Diagnosis

The maximization of equation (6) can be done in a number of ways. If the objective is to rank features based on relevance using many observations, then gradient ascent on R_i can be employed. Note that the Hessian of the classifier with respect to the input can be computed in linear time using the technique described in [32]. The analytic solution requires computing the inverse of the Hessian. A few fast iterative methods for computing the inverse of the Hessian are described in [32]. If the classifier is reasonably complex (i.e. a multi-layer network with many hidden units), and the Hessian is not ill-conditioned, an analytic solution can be found. If the classifier is not complex and the Hessian is ill-conditioned, a reasonable analytic solution can be found if it is assumed that the Hessian is diagonal.

3.3 Discussion

Differential discriminative diagnosis could potentially be classified as a *filter* approach since it assigns continuous valued weights to each feature in the feature vector. Unlike most of the techniques described in section 2.2, *differential discriminative diagnosis* estimates the relevance of features while the classifier is functioning. The relevance assessments explicitly consider discrimination as opposed to techniques such as principal components analysis. It builds on the existing accuracy of the classifier based on the observed data, thus, it can account for time-varying relevances. *Differential discriminative diagnosis* can also be used to select features off-line similar to the search techniques mentioned in section 2.2. Unlike the transform based feature set reduction techniques, *differential discriminative diagno-*

sis considers each feature independently. Note that this independence assumption is conditional on the classifier. Thus, if the classifier accounts for the dependencies between features, then significant dependencies are not ignored by the diagnosis process. Consequently, retinatopic feature vectors such as images, where there is sufficient similarity between them, work well with this technique. We focus on applications using imagery in the following sections. In section 4.0 we concentrate on an application where feature relevance is fundamentally time varying. We show that *differential discriminative diagnosis* significantly improves the performance of the classifier.

4.0 Agent-Based Moving Object Correspondence

In this section, we propose a novel method for temporally and spatially corresponding moving objects by automatically learning the relevance of an object's appearance features to the task of discrimination. Efficient correspondence is achieved by enforcing temporal consistency of the relevances for a particular object. Relevances are learned using *differential discriminative diagnosis*. An agent is assigned to each moving object in the scene. The agent possesses the basic capability to decide whether or not an object in the scene is the one it represents. Each agent customizes itself to the object by means of *differential discriminative diagnosis* as the object persists in the scene. We explain this correspondence scheme as applied to the task of corresponding moving people in a surveillance system.

4.1 The Problem

There has been an increased interest in distributed surveillance systems in recent years [33, 34, 35, 36, 37]. The objective is to provide critical information to the human user in real time. A surveillance network of reasonable size produces massive quantities of information. Much of this information is redundant and can inundate a human operator while distracting him or her from information of substance. A distributed surveillance system that can automatically eliminate the redundancy in the information conveyed to the user is invaluable. Changes in the scene, induced by the motions and actions of

people and vehicles, are usually the subject of interest in most urban surveillance scenarios. The ideal distributed surveillance system should be able to track all the motions and interactions of objects and raise appropriate flags when information of importance needs to be conveyed to the user. This paper addresses the problem of temporally corresponding moving objects to facilitate a good interpretation of the objects' actions.

The complexity of motions in the environment precludes the use of simple positional correspondence, i.e., correspondence based purely on the positions of moving objects. Positional correspondence also fails when moving objects are relatively large with respect to the field of view of the sensors. In such situations, other features of the moving objects, such as different appearance traits, need to be put to good use for robust correspondence. *How can we select appearance features so as to facilitate good correspondence?* The measure of goodness of the features we choose not only depends on the object in question, but also on other objects in the scene. A globally "good" set of features can be estimated *a priori*, but only a subset of these features might be relevant to the correspondence of a particular object. We pose the estimation of the relevance of globally good features for corresponding a particular object as an on-line learning task. *Differential discriminative diagnosis* provides a systematic method for estimating the relevance of features and checking the temporal consistency of these features for a particular object.

4.2 Related Work

Much work has been devoted to efficient object correspondence and tracking. Surveillance systems described in [33, 34, 35, 36] deal with the problem of detecting and tracking moving objects. The system described in [33] uses correlation with *dynamic templates* of the object as a method for temporally corresponding it. An IIR filter is used to adapt the *dynamic templates* over time. The system described in [34] uses linear prediction with Kalman filters of the position and size of the moving objects. The algorithm described by Cohen and Medioni in [35] combines the detection and tracking process. They use a graph representation to generate dynamic templates of each moving object. An

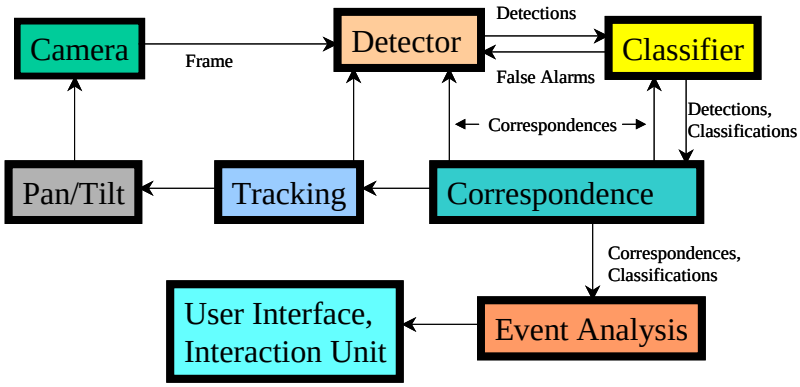
object's trajectory is determined by choosing an optimal path through the graph and enforcing a temporal coherence constraint. Haritaoglu, et al., [36] use correlation of the moving object's silhouette and template matching.

There has also been a good amount of work done in tracking specific objects. Notably, [38] describes tracking people and their actions. Gaussian models are used to represent 2-D regions or blobs. The model accounts for the position and color of the blobs. These blobs are used to track the position of the person in the scene. Wren and Pentland in [39] extend the notions described in [38] to a 3D context and model a person's physical actions explicitly. McKenna et al., [40] use a Gaussian mixture model of the color of an object to track it effectively. Black and Jepson in [41] describe an eigenspace method for tracking specific rigid objects. They use a multi-scale eigenspace approach to represent and match objects over time. They apply this technique to the task of tracking and recognizing the gestures of a moving hand. Rehg et al., [42] describe a method for tracking high-DOF articulated objects. They employ this method for tracking humans. They explicitly model the kinematics of articulated parts and use this model to perform correspondence. Other notable people tracking systems include *KidsRoom* [43] and *Cardboard People* [44]. These systems also seek to model the articulation of humans.

Our proposed method relies on knowing the class of the object (person, people or vehicle). Thus, there is domain knowledge incorporated in the correspondence process. The injected domain knowledge not only helps in making the correspondence process robust, but it also helps in making it computationally efficient. By accounting for different moving objects of interest, we come close to obtaining the versatility of class independent correspondence. In contrast to most of the methods mentioned in this section, our proposed technique poses moving object correspondence as a statistical pattern classification/discrimination problem. Rather than modeling motion, our algorithm finds stable discriminating features to correspond an object. We show that training an agent to correspond an object off-line and

giving it the capability to customize itself to the object on-line, leads to an efficient correspondence algorithm.

FIGURE 2.



CyberARIES high-level surveillance systems Architecture. Each block represents a concurrently running piece of software. Each arrow indicates the direction of information flow and information that is being transmitted

4.3 CyberARIES Surveillance Architecture

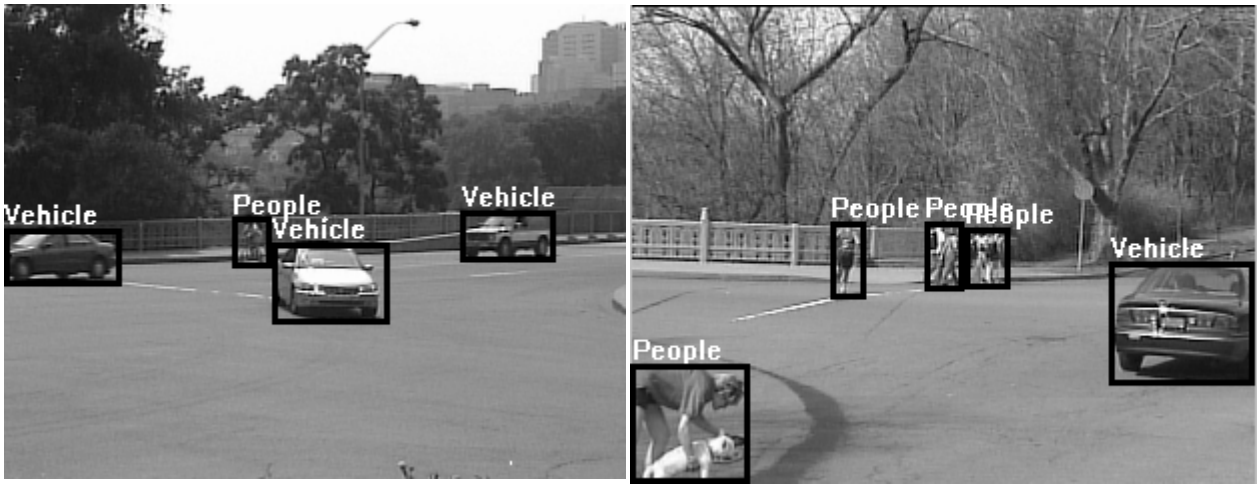
An agent-based architecture offers an efficient and convenient software infrastructure for a distributed surveillance system. Such an architecture facilitates the combined use of powerful tools from machine learning and computer vision. We have developed an agent based system called CyberARIES for Autonomous Reconnaissance and Intelligent Exploration. CyberARIES has been implemented to run on stationary and mobile surveillance platforms. The object correspondence scheme to be described in this paper uses the CyberARIES architecture as a fundamental implementation tool.

Figure 2 shows the connectivity of the surveillance architecture within CyberARIES with respect to the correspondence agents. The camera produces 320×240 8-bit grayscale images which are sent to the detector agent. The detector agent uses a bank of auto-regressive filters to model the background. The background model is then used to detect moving objects in the scene. Connected components analysis is used to segment the detected moving objects from the background. This detection and segmenta-

tion scheme has proven to be simple yet effective. The detection process is elaborated in the section 4.4. The detector agent then feeds a list of segmented objects to the classifier agent which in turn classifies the object as a “person”, “people” or a “vehicle.” The classifier also has the ability to reject detections of no interest to the surveillance task. Examples of uninteresting detections include moving foliage, false alarms caused by changing lighting conditions and high frequency motion of the camera on a vibrating platform. The correspondence agents work with the classified moving objects as their input.

In addition to the feedforward connectivity described so far, there also exist feedback connections from the classifier and the correspondence agents to the detector. The classifier feeds back the locations of the detected objects that were rejected. Information about the predicted future positions of the corresponded moving objects are also fed back to the detector. The information feedback is used to adapt the local sensitivity parameters of the detection filters. This simple feedback mechanism is extremely effective in improving the SNR of the detections. Figure 3 shows the system in action with the classifier designed to label each moving object as “people” (in this version of the system, class “people” includes class “person”) or a “vehicle.” The two figures also provide a clear idea of the typical operating environment for the system. Objects can either be a good distance away or very close to the camera.

FIGURE 3.



The system detects and classifies the moving people and vehicle in the scene.

4.4 Detecting and Segmenting a Moving Object

Our basic algorithm is similar to those described in [33, 34, 36]. The algorithm described in [33] uses an IIR filter to model the background. In [34], Grimson et al. use a Gaussian mixture to estimate the background of the scene. Both [33, 34] use color imagery in their background modeling process. Haritaoglu [36] describes a technique that estimates the maximum and minimum intensity differences for each pixel while there are no moving objects in the scene. This information is then used to detect moving objects. Like the system described in [36], we use grayscale imagery to estimate the background. A key difference in our work is the use of feedback from higher-level processes such as the classifier and the correspondence agents to adapt the detection process. This information feedback helps us simplify the detection algorithm while performing just as well as more complicated detection processes such as [34].

4.4.1 The AR Filter

We seek to model the background by using a set of AR filters to represent each pixel. Let $I_{n,x,y}$ represent a pixel at time n and at position (x, y) in a 320×240 8-bit grayscale image. Similarly, let

$B_{(n-1),x,y}$ represent the predicted background value for that pixel. A significant difference $D_{n,x,y} = I_{n,x,y} - B_{(n-1),x,y}$ between the image and the background values suggests the presence of a moving object. If $|D_{n,x,y}| - b_{n,x,y} > 0$, the pixel is classified as foreground. If $|D_{n,x,y}| - b_{n,x,y} \leq 0$, the pixel is classified as background, where $b_{n,x,y}$ represents the decision threshold for a particular pixel. At each step, we wish to gradually minimize the difference between the background model $B_{(n-1),x,y}$ and the image $I_{n,x,y}$ by using the update rule $B_{n,x,y} = B_{(n-1),x,y} + \eta D_{n,x,y}$, where η represents a small learning rate constant. The update rule can also be posed as an AR filter, as shown in equation (7).

$$B_{n,x,y} = (1-\eta)B_{(n-1),x,y} + \eta I_{n,x,y} \qquad (7)$$
$$0 \leq \eta \leq 1$$

Rather than making the threshold $b_{n,x,y}$ a constant, we adapt it just as we adapt the background, using an AR filter.

This simple AR filter-based background model is quite effective, but suffers when objects in the scene are moving too slowly. Often when the moving objects are slow, the background incorrectly acquires part of the object, resulting in false alarms. To alleviate this problem, we introduce a conditionally lagged background model. The conditionally lagged background model, $B^{cond}_{n,x,y}$ is set to the continuously updated model, $B_{n,x,y}$ if the pixel is classified as a background pixel. If the pixel is classified as a foreground pixel, we don't update the conditionally lagged background. If after some T time steps, the magnitude of the difference between the conditionally lagged background and the image, $D^{cond}_{n,x,y}$, is less than the magnitude of $D_{n,x,y}$, the value of $B_{n,x,y}$ is reset with the value of $B^{cond}_{n,x,y}$. On the other hand, if the magnitude of $D_{n,x,y}$ is less than the magnitude of $D^{cond}_{n,x,y}$, the value of $B^{cond}_{n,x,y}$ is set to $B_{n,x,y}$. The classification of a pixel as foreground or background now depends on $|D^{cond}_{n,x,y}| - b_{n,x,y}$. When T is chosen appropriately, this technique prevents the false alarms caused by the movement of slow objects.

Given a binary map of foreground detections, connected components analysis is used to segment the moving objects from the background. Prior to this step, morphological operations such as closing and erosion are performed to remove stray detections. Each positive binary value (i.e., a detection) is replaced with the actual grayscale value from the original image. Thus, the segmented image contains just the grayscale values of the moving object without the background.

4.4.2 The Feedback Mechanism

The feedback mechanism allows for adjusting the sensitivity parameters of the detections. The feedback contains information about the labels (as “people”,“person” or “vehicle”) and correspondences of each detected pixel. The information feedback is used to adapt a *Perceptron* to better classify a pixel as foreground or background. The classification of a pixel as foreground depends on the inequality $\left|D^{cond}_{n,x,y}\right| - b_{n,x,y} > 0$. We can redefine the classification rule as $\omega_0 + \omega_1\left|D^{cond}_{n,x,y}\right| - \omega_2 b_{n,x,y} > 0$, where each of the weights, ω_i, is adaptable. The weights are updated using the standard perceptron learning rule. If a pixel was classified as foreground and was part of a successfully labelled and corresponded object, the classification (as foreground) is considered correct. On the other hand, if the pixel was part of a rejected object, then the classification is considered incorrect. In practice we have found that adapting ω_0 alone (with $\omega_1 = \omega_2 = 1$) provides a considerable increase in detection performance.

For most surveillance applications, it is important to track foreground objects that become stationary after a while. The information feedback from the correspondence agent is used to determine which pixels represent a stationary object. A new AR filter is initialized to keep track of the changing intensity of the pixel. This new AR filter can be visualized as an additional background layer. As long as the difference between the value of the new AR filter and the original AR filter is significant (as determined by the *perceptron*), the layer is maintained. This ensures that the now stationary foreground object is not regressed into the original background of the scene.

4.5 The Basic Correspondence Agent

Consider the detected and classified people and vehicles in Figure 3. The correspondence agent is responsible for temporally corresponding each moving object. Under the CyberARIES framework, an agent is assigned to every moving object in the scene. This section describes the correspondence algorithm that each agent possesses before any on-line learning occurs.

4.5.1 Input Representation and Classification Problem

We pose the correspondence problem as a classification problem. Let $t_{n,s}$ denote an object t at time instance n belonging to a sequence s. The object t is represented by an intensity map with the background subtracted. Figure 4 shows an example of the intensity map contained in $t_{n,s}$. The appearance features that we are interested in are captured by $t_{n,s}$. The temporal correspondence problem can be defined as matching the object $t_{n,s}$ with a previously seen instance $t_{n-1,s}$. Let $X_{n,i,j}$ denote the magnitude of the difference of each pixel between two object instances as shown in equation (8). The

$$X_{n,i,j} = Cr(\left|Ce(Re(t_{n-1,i})) - Ce(Re(t_{n,j}))\right|) \tag{8}$$

subscripts i and j index potentially different sequences. The function $Re(\ )$ resizes the intensity map of the object to a standard size. The function $Ce(\ )$ centers the object in the image using its center of mass. The function $Cr(\ )$ crops the difference of the intensity maps to a prescribed size. Figure 5 shows the resized, centered and cropped versions of the images in Figure 4. Ideally, if $t_{n,i}$ and $t_{n-1,j}$ represented the same object at two different instances in time, then $X_{n,i,j}$ should contain mostly zeros with very few high magnitude values. Unfortunately, the articulation of a person's limbs induces large magnitude values, but the locations of these large magnitude values are more or less consistent. Noise and centering errors also cause large magnitude differences. The objective of the classifier is to decide whether or not a given $X_{n,i,j}$ represents the acceptable differences between two instances of the same

object or the differences between two instances of different objects. Thus, the classifier classifies each $X_{n,i,j}$ as a "match" or "no match."

FIGURE 4.

Two consecutive instances of a moving person from the same sequence. The object is represented by an intensity map of the person with the background subtracted away as shown above.

FIGURE 5.

The two images shown here are the resized, centered and cropped versions of the images above. Equation (8) is the absolute value of the difference between these two images.

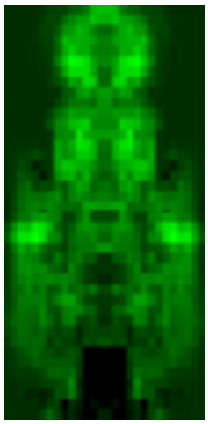
4.5.2 Designing and Learning a Classifier

An ideal classifier for this application should, with minimum functional complexity, approximate the Bayes-Optimal classifier well. The minimum complexity requirement is critical because of the computational constraints faced by a real-time surveillance system. It should approximate the Bayes-Optimal classifier well because we want the classifier's performance to be as good as possible given the input representation. Hampshire and Pearlmutter [45] prove the equivalence between Multi-Layer Perceptrons (MLPs) and Bayesian discriminant functions for two general classes of objective functions. The two classes can be categorized as error measures and classification figures of merit (CFM) [46]. Hampshire [45, 47] shows that *Differential Learning*, using the CFM objective function, generalizes better and requires less functional complexity than error measures such as Mean Squared Error. More-

over, *Differential Learning* focuses on maximizing the separation between classes rather than learning the *a posteriori* probabilities of the classes given a finite amount of training data [47].

For this particular application we chose a single output logistic linear neural network trained with *Differential Learning* as the classifier. A total of 249 sequences were available for training the classifier. A total of 120 sequences were used for independent testing. Each sequence contained an average of 15 instances of an object. Sequences were manually sorted from data collections in different environments. Different permutations of sequence pairs were constructed for training and testing. The classifier successfully matched instances of the same moving object with an accuracy of 87%. The 95% confidence interval is [84%, 90%]. Figure 6 shows the weights learned by the classifier. Notice the emphasis on the shoulder and head regions. Emphasis is also placed on the back and sides of the person. The weight layer was forced to be symmetric to account for any bias in the training data for a particular direction of movement for the moving objects.

FIGURE 6.



The weights learned by the classifier. Notice the emphasis on the head, shoulders and the sides. The weight layer was forced to be symmetric to account for an bias in the training data.

4.6 Differential Discriminative Diagnosis

The classifier's training process selects features on the person's body that help in the classification task given the training data. These features are "globally" relevant, i.e., the selected features help in discriminating a majority of the moving objects without being specific to a particular object. Different environmental conditions and different scenes may reduce or increase the relevance of certain features. More importantly, only a subset of the "globally" relevant features may be applicable to the correspondence of a moving object. In some cases, certain "globally" relevant features may actually hurt the correspondence process. Thus, identifying the feature subset that is relevant to the correspondence of a particular moving object could increase performance dramatically. An agent that represents a moving object customizes itself by estimating the relevance of each feature in the input vector $X_{n,i,j}$ based on the reaction of the classifier to the input and the other objects in the scene. This estimation process is accomplished by means of *differential discriminative diagnosis.*

4.6.1 Using Differential Discriminative Diagnosis

The features that contribute most to the discrimination task are those that consistently appear on the moving object and are different from those on other moving objects. Temporally consistent features have entries with values of zero or a low magnitude in the feature vector $X_{n,i,i}$. We are interested in the *discriminative* subset of these temporally consistent features that also have a consistently high magnitude in the feature vector $X_{n,i,j}$ where $i \neq j$. We would also like to apply our prior knowledge of feature relevance in the form of the optimized classifier. To this end, we use the *relevance differential* $\Re_{n,i,j}$ for the feature vectors $X_{n,i,i}$, $X_{n,i,j}$ and $X_{n-1,i,i}$ as shown in equation (6). The *relevance differential* is the difference between the magnitude of the approximated classifier output differences, $|P_{n,i,j}|$ defined in section 3.2. Recall that the index i denotes the sequence of the correct match. The *relevance differential* $\Re_{n,i,j}$, defined in equation (6) indicates the difference between the aggregated irrelevances of all the

features in the feature vectors $X_{n-1,i,i}$, $X_{n,i,i}$ and $X_{n,i,j}$. Given the feature vectors $X_{n-1,i,i}$, $X_{n,i,i}$ and $X_{n,i,j}$ where $i \neq j$, we find the features that contribute the most to the discrimination task by maximizing the *relevance differential* $\Re_{n,i,j}$ with respect to the matrix R_i. In other words, we wish to make the correct match, $X_{n-1,i,i}$ and $X_{n,i,i}$ as close to each other as possible with respect to the classifier, while making the incorrect match, $X_{n-1,i,i}$ and $X_{n,i,j}$ as far away from each other as possible. Features with a high magnitude entry in R_i are those that are both *temporally consistent* and are *discriminative*. Thus, the matrix R_i provides an indication of the relevance of each feature in $X_{n-1,i,i}$ to the correspondence task.

The maximization can be done either by gradient ascent or analytically. The analytical solution is possible because $\Re_{n,i,j}$ as a function of R_i is quadratic with only one local minimum or maximum. Unfortunately, the analytical solution involves computing the inverse of $H_X C(X_{n-1,i,i})$ and the gradient ascent process is too slow for our purpose. For a relatively small logistic linear classifier, the Hessian is ill-conditioned. Thus, even approximating the inverse leaves room for significant approximation errors. In order to make this computation feasible, we choose to assume that the Hessian is diagonal. This assumption doesn't hurt the computation significantly since we can account for the errors in the optimization process. Also, note that the Hessian can be computed off-line except for a multiplicative scalar that depends on the input feature vector $X_{n,i,j}$.

4.6.2 The Cumulative Relevance Differential

Given the diagonal assumption for the Hessian, we wish to maximize the *relevance differential* $\Re_{n,i,j}$ with respect to R_i for all the moving objects in the scene (indexed by j) and for all time (indexed by n). Thus, the expression to maximize for the i^{th} moving object is given by (9) We refer to equation

$$\underbrace{\sum_{n}\sum_{j \neq i} \Re_{n,i,j}} \tag{9}$$

(9) as the *cumulative relevance differential*, where n iterates over all time that the object was present in the scene, and j iterates through all the other moving objects in the scene at time instance n. We first consider maximizing $\Re_{n,i,j}$ for a particular n, i and j, which maximizes (6). Then we extend the derivation to maximize $\Re_{n,i,j}$ for a fixed n and i over all j. Finally, we derive a recursive equation to maximize over all n, which maximizes (9). Let ${}^{k}x$ denote the k[th] element of a vector x. Also, let M denote the number of elements in the vector x. Similarly, let ${}^{k}R_i$ and ${}^{k}H_X C(\ )$ denote the element in cell (k,k) in each of the matrices. We seek to maximize equation (6) with respect to each element ${}^{k}R_i$ of the matrix R_i. Equation (10) represents the value of ${}^{k}R_i$ at the local extremum of equation (6). Additionally, we

$$S_{n,i,j} = -\mathrm{sgn}(P_{n,i,j}) \times \mathrm{sgn}(P_{n,i})$$

$${}^{k}R^{ext}_{n,i,j} = \frac{(-({}^{k}h_{n,i,j} + S_{n,i,j}\,{}^{k}h_{n,i})\nabla_X {}^{k}C(X_{n-1,i,i}))}{({}^{k}h^2_{n,i,j} + S_{n,i,j}\,{}^{k}h^2_{n,i})({}^{k}H_X C(X_{n-1,i,i}))} \tag{10}$$

define the vectors $A_{n,i,j}$ and $B_{n,i,j}$ of the same size as R_i for notational and computational convenience. Each element, ${}^{k}A_{n,i,j}$ and ${}^{k}B_{n,i,j}$, of the vectors $A_{n,i,j}$ and $B_{n,i,j}$ is defined in equations (11) and (12). We can now derive the local extremum of the *relevance differential*, considering all the moving

$${}^{k}A_{n,i,j} = ({}^{k}h_{n,i,j} + S_{n,i,j}\,{}^{k}h_{n,i})\nabla_X {}^{k}C(X_{n-1,i,i}) \tag{11}$$

$${}^{k}B_{n,i,j} = ({}^{k}h^2_{n,i,j} + S_{n,i,j}\,{}^{k}h^2_{n,i})({}^{k}H_X C(X_{n-1,i,i})) \tag{12}$$

objects (indexed by j) in the scene at time instance n, in equation (13). Finally, equation (13) forms the

$${}^{k}R^{ext}_{n,i} = \frac{\left(-\sum_j {}^{k}A_{n,i,j}\right)}{\left(\sum_j {}^{k}B_{n,i,j}\right)} = -\frac{{}^{k}A_{i,n}}{{}^{k}B_{i,n}} \tag{13}$$

basis for deriving a recursive relationship to find the extremum of equation (9). Equation (14) shows the

$$ {}^{k}R_i^{ext} = -\frac{({}^{k}A_{i,n} + {}^{k}A_{i,n-1})}{({}^{k}B_{i,n} + {}^{k}B_{i,n-1})} \tag{14} $$

$$ 0 \le {}^{k}X_{n-1,i,i} + ({}^{k}R_i^{max})({}^{k}h_{n,i,j}) \le 255 \tag{15} $$

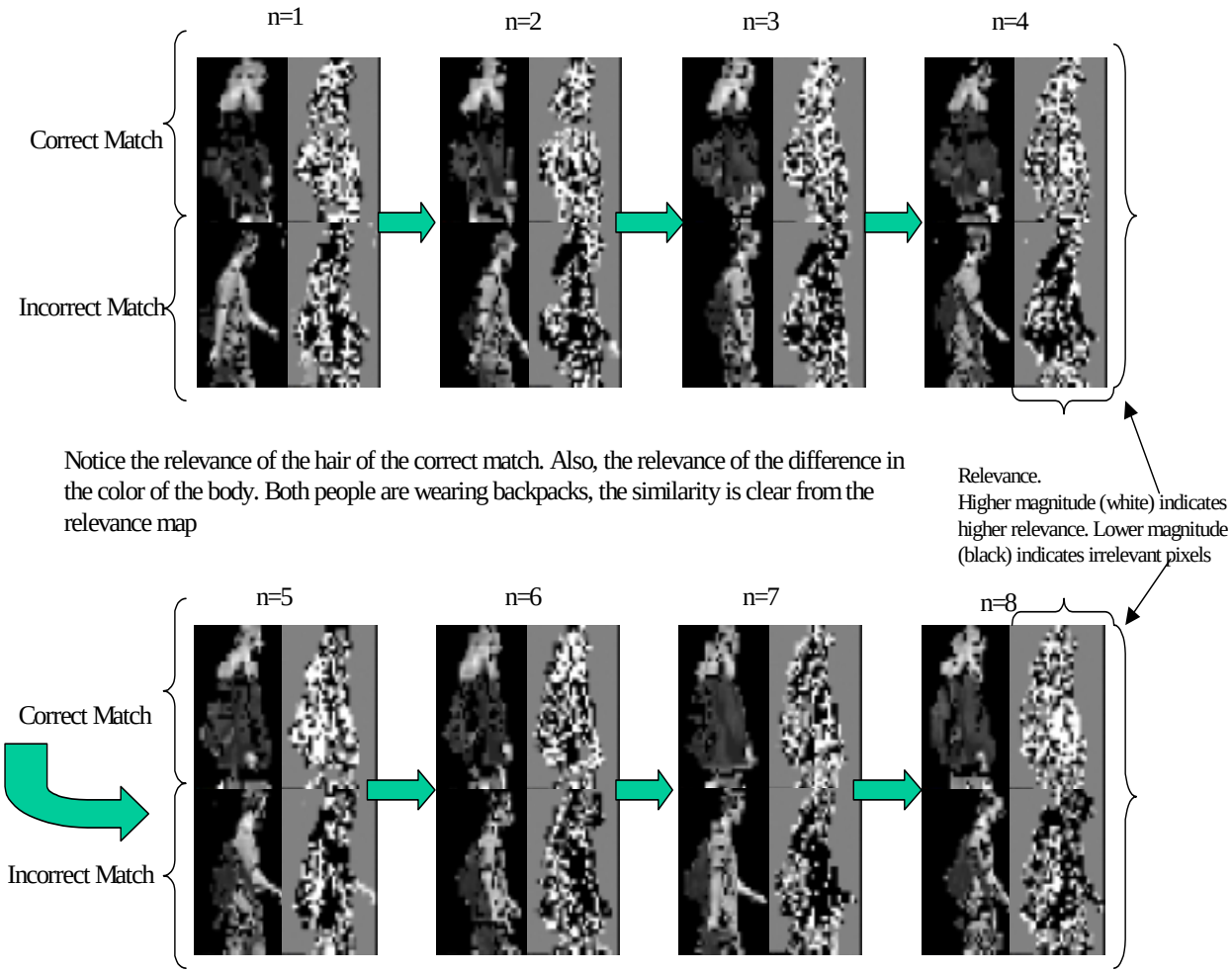
extremum of the *cumulative relevance differential*. The sign of ${}^{k}B_{i,n}$ in equation (14) determines if the extremum at ${}^{k}R_i^{ext}$ is a minimum or a maximum. Based on this fact and the boundary conditions on ${}^{k}R_i^{max}$ given by equation (15), we can find the R_i^{max} that maximizes the *cumulative relevance differential* (9).

Each agent adapts its relevance matrix R_i as it sees more instances of the object it represents. The agent can then decide whether or not an object is the one it represents based on the number of relevant pixels (pixels that are temporally consistent and discriminative) on the object. Instead of storing past instances of objects needed to adapt R, the agent stores the sufficient statistics $A_{n,i,j}$ and $B_{n,i,j}$. As a target moves from the field of view of one sensor to another, the agent follows the target.

4.7 Results

The performance of the correspondence agents was tested on the same 120 independent test sequences used to evaluate the basic correspondence paradigm described in section 4. The agents achieved an accuracy of 96%. The 95% confidence interval is [94.3%, 97.7%]. The customizing step shows statistically significant improvements over the 87% accuracy obtained using just the classifier. Figure 7 illustrates the agents powered by *differential discriminative diagnosis*. It shows the relevance of the pixels on the two persons as the agent performs correspondence.

FIGURE 7.



The relevant differences ($R_i h_{n,i,j}$) are shown here for a person being corresponded and another person in the scene. $R_i h_{n,i,j}$ has been thresholded to show the contrast between relevant and irrelevant pixels. The correct match has clearly fewer irrelevant pixels than the incorrect match.

4.8 Conclusions

The proposed correspondence algorithm has been shown to perform well in corresponding people. The algorithm can easily be extended to track vehicles. The algorithm has a few clear failure modes. A temporally abrupt and geometrically drastic change in viewing angle causes the algorithm to fail. The algorithm also fails when tracking a person who bends or twists such that a good number of

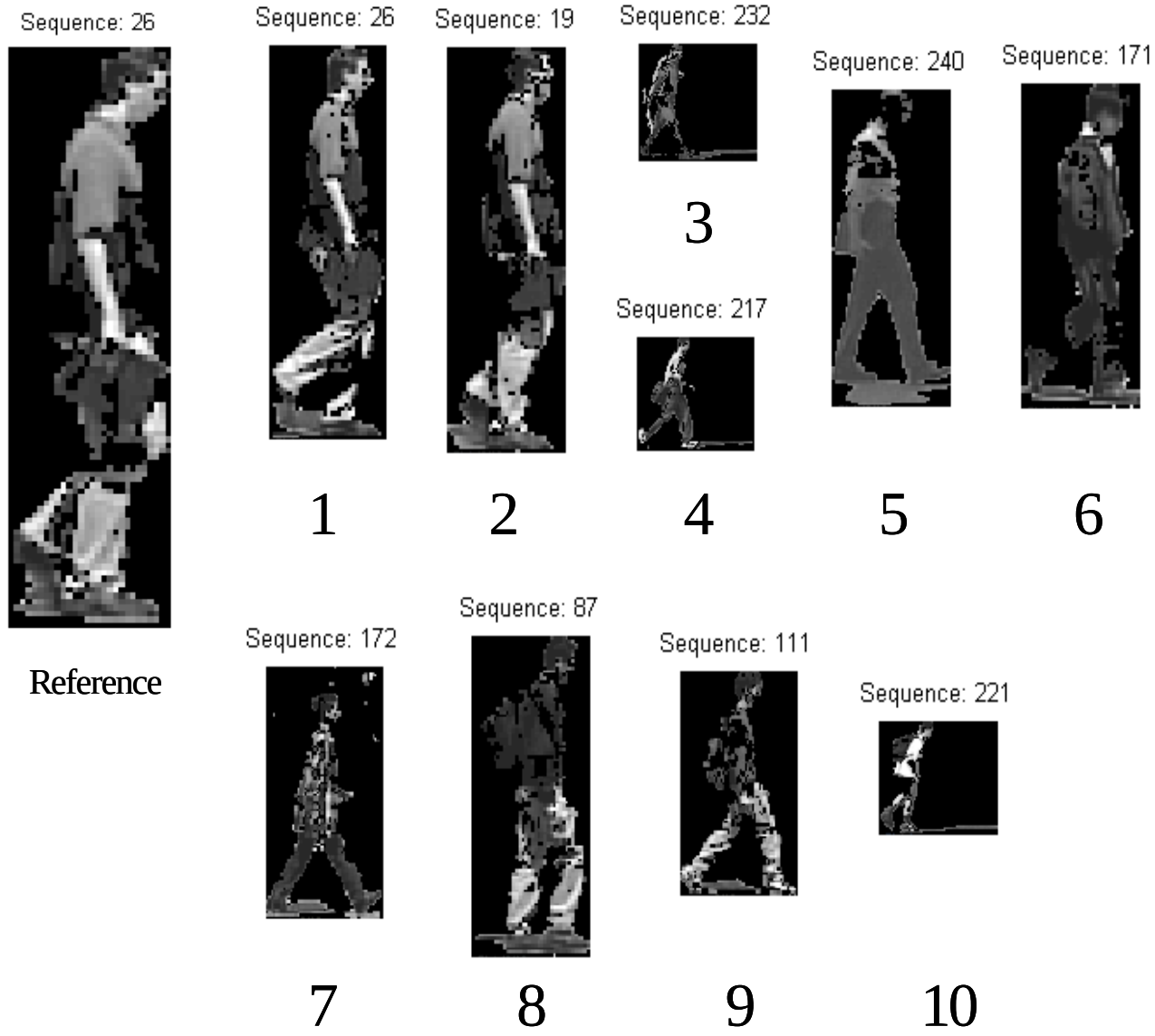
features that were previously visible are no longer in plain view. Occlusion is another cause for failure. We use this correspondence algorithm in conjunction with basic positional correspondence by means of linear prediction of the object's position. This alleviates the effects of some of the failure modes of appearance-based correspondence. We are currently experimenting with situations where more than one sensor is looking at the same target. We hope to extend this algorithm to be able to correspond targets not only within a sensor's field of view, but also among sensors. This raises interesting questions of choosing viewpoint-independent features or multiple sets of viewpoint-dependent features for efficient correspondence.

5.0 Retrieving Similar Images: Extending Correspondence

The moving object correspondence algorithm described in the previous section can easily be extended to a content based image retrieval context. In addition to corresponding consecutively occurring instances of a moving object, a surveillance system should also be able to correspond instances of a moving object seen by multiple sensors and at different time windows. Specifically, consider the scenario where a moving object passes through a sensor's field of view in the morning and then reappears later that evening. Interpretation of the moving object's actions may involve recognizing that it was the same object at two different windows in time. Similarly it might also be necessary not only to associate the same objects, but also similar objects.

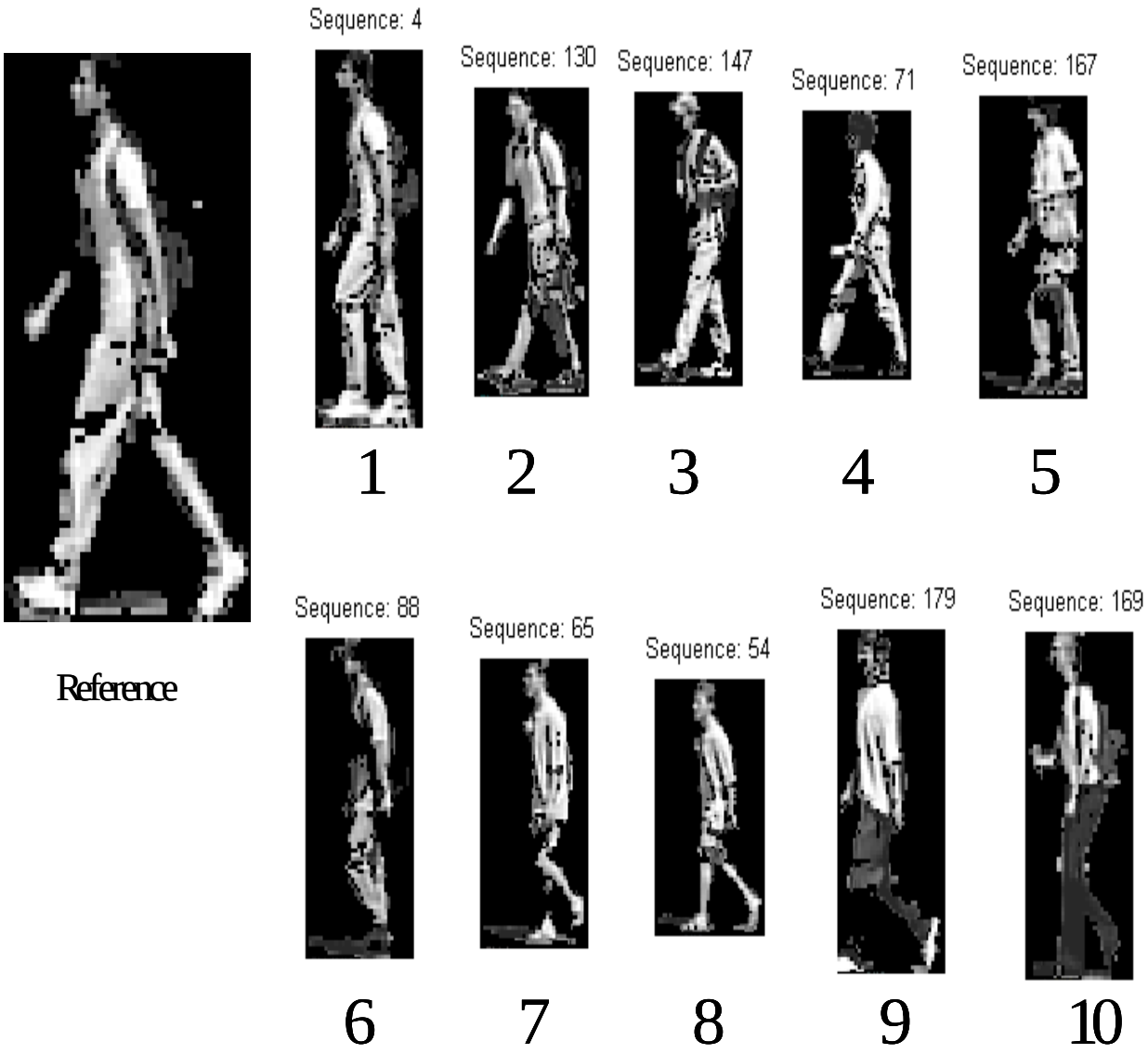
The correspondence scheme described in section 4.0 can easily be extended to the retrieval problem. When we save an image of a moving object in a database, we also save the associated R_i matrix. For every candidate object in the database, given an image of a reference object, we determine how many relevant pixels are on the candidate object. Pixels on a target are relevant if for each large magnitude value in R_i, the corresponding entry in $X_{n,i,j}$ is small. Each object can then be ranked based on the number of relevant pixels. The following figures illustrate the results of this retrieval scheme.

FIGURE 8.



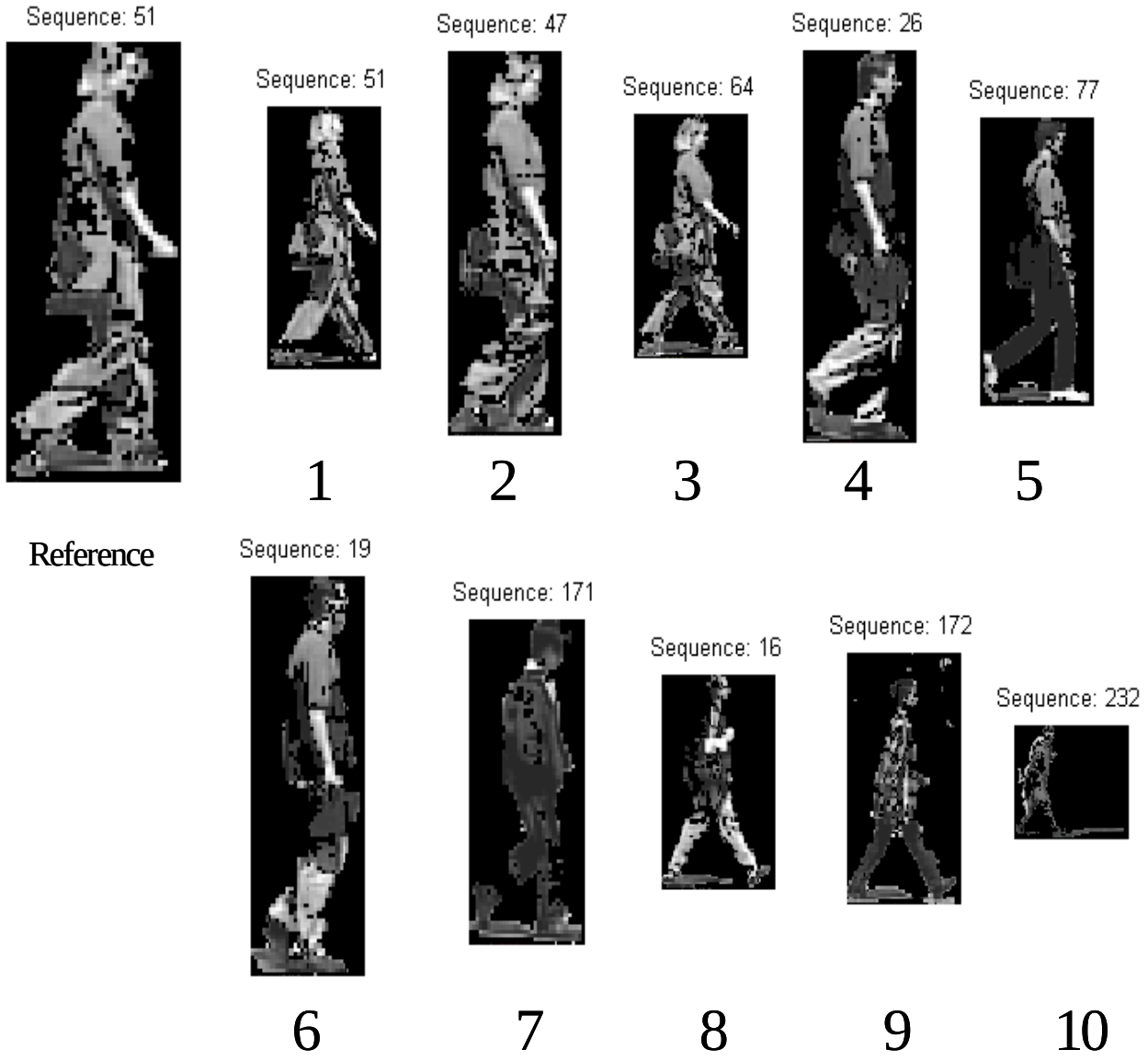
The top 10 matches out of a total 249 people are shown above. Note that the best match belongs to the same sequence as the reference person. Sequence 19 (the 2nd best match) is the same person at an earlier time window. Except for the reference, all other images are shown at their original sizes.

FIGURE 9.



The top 10 best matches (out of a possible 249 images) for another reference person are shown above. Again, notice that the best match belongs to the same sequence as the reference. The rest of the matches are not the same person, but are very similar in appearance.

FIGURE 10.



Notice in this figure that the algorithm picked two other sequences of the same person (matches 2 and 3). Matches 8 and 9 are perceptually not the same as the reference and it is not clear why these two sequences are listed in the top ten best matches.

5.1 Conclusions

We did not explore many retrieval methods. Potentially, well studied methods from the content-based image retrieval arena could perform better than the method used here. But, we have found that

this method works well with little additional computation and suits the purpose of the surveillance task. Moreover it validates the proposed correspondence technique as being robust. This retrieval mechanism can also be used to learn from unlabeled data. If feature vectors within a class share some similarities and share some differences with feature vectors of other classes, then *differential discriminative diagnosis* will rank those features as relevant. Unlabeled data can now be "labeled" based on the presence (or absence) of relevant features. This training process can be repeated iteratively till the label assigned to each of the unlabeled feature vectors are consistent.

6.0 Conclusions

This thesis makes the following novel contributions:

- A novel way for ranking the relevance of features to a discrimination task on-line.
- A novel way to account for time-varying relevances. An extreme case of ever changing relevances was presented in the correspondence application and was dealt with successfully using *differential discriminative diagnosis*.

Differential discriminative diagnosis is a simple and effective technique for estimating feature relevances. It can be used as an on-line learning algorithm to improve systems trained off-line. Unlike techniques such a principal components analysis, etc., it accounts for the discrimination power of a feature. At the same time, the relevance estimation process does not consider combinations of features as many of the transform based techniques suggested in section 2.2. *Differential discriminative diagnosis* assumes that given the classifier, each of the features in the feature vector are independent. This assumption may not be true in general. This technique is known to work well with images where a fair amount of perceptual similarity exists between feature vectors of the same class. *Differential discriminative diagnosis* was also used successfully for a path-planning problem. In this problem, a classifier was used to monitor a planned path as a mobile robot proceeded to its destination. The classifier raised

a flag when the path was determined to be bad because of a change in the environment (for e.g. moving people or vehicles). *Differential discriminative diagnosis* was used to determine the location(s) in the path that caused it to turn from good to bad. This problem and the proposed solution are described in detail in [48].

7.0 Future Work

An intuitive justification and empirical proof of the capabilities of *differential discriminative diagnosis* were presented in this thesis. Future work includes an in-depth analysis of the statistics learned by this algorithm. The relationship between the classifier and the algorithm also needs to be formalized. As a by-product of the analysis, failure modes of the algorithm also need to be outlined. We are currently experimenting with using this algorithm to learn with unlabeled data and recognizing previously unseen classes.

8.0 References

[1] R. A. Fisher. The use of multiple measurements in taxonomic problems. In Annals of Eugenics 7, pp. 179-188. Reprinted in Contributions to Mathematical Statistics, John Wiley: New York, 1950.

[2] K. Fukunaga. Intrinsic dimensionality extraction. In P. R. Krishnaiah and L. N. Kanal (Eds.), Classification, Pattern Recognition and Reduction of Dimensionality, Volume 2 of Handbook of statistics, pp. 347-360, 1982.

[3] D. H. Foley, J. W. Sammon Jr., An optimal set of discriminant vectors. In IEEE Transactions on Computers, vol. C-24 (3), pp. 280-288, 1975.

[4] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. In IEEE Transactions on Computers 26 (9), pp 917-922, 1977.

[5] P. A. Devijver and J. Kittler. Pattern Recognition: A Statistical Approach. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[6] I. T. Jollife. Principal Components Analysis. New York: Springer Verlag, 1986.

[7] G. Deco and D. Obradovic. An information-theoretic approach to neural computing. Springer-verlag, 1996.

[8] G. John, R. Kohavi and K. Pfleger. Irrelevant features and the subset selection problem. In Proceedings of Machine Learning, pp 121-129, 1994.

[9] K. Kira and L. A. Rendell. The feature selections problem: Traditional methods and a new algorithm. In proceedings of AAAI, pp 129-134, 1992.

[10] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In Proceedings of AAAI, pp 547-552, 1991.

[11] R. Caruana and D. Frietag. Greedy attribute selection. In Proceedings of Machine Learning, 1994.

[12] P. Langley and S. Sage. Induction of selective bayesian classifiers. In Procedings of UAI, pp 399-406, 1994.

[13] D. Koller and M. Sahami. Toward optimal feature selection. In Proceedings of Machine Learning, pp 284-292, 1996.

[14] S. Kullback and R. A. Leibler. On information and sufficiency. In Annals of Mathematical Statistics 22, pp 76-86, 1951.

[15] M. Scherf and W. Brauer. Improving RBF networks by the feature selection approach EUBAFES. In Proceedings of ICANN, pp. 391-396, 1997.

[16] K. J. Cherkauer and J. W. Shavlik. In Proceedings of Advances in Neural Information Processing Systems, pp. 45-51, 1996.

[17] J. Rissanen. Stochastic Complexity in Statistical Inquiry. World Scientific, 1989

[18] M. T. Fardancsh and O. K. Ersoy. Classification accuracy improvement of neural network classifiers by using unlabeled data. In IEEE Transactions on Geoscience and Remote Sensing, vol. 36 (3), pp. 1020-1025, 1998.

[19] G. Towell. Using unlabeled data for supervised learning. In Proceedings of the Advances of Neural Information Processing Systems, pp 647-653, 1995.

[20] C. X. Ling and H. Wang. Learning classifications from multiple sources of unsupervised data. In Proceedings of the Advances on Artificial Intelligence, pp. 284-295, 1996.

[21] D. Hamad, C. Firmin and J. Postaire. Unsupervised pattern classification by neural networks. In Mathematics and Computers in Simulation, vol. 41 (1-2), pp. 109-116, 1996.

[22] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of Computational Learning Theory, pp 92-100, 1998.

[23] R. E. Bellman Dynamic Programming. Princeton, NJ: Princeton University Press, 1957.

[24] D. Blackwell. Discounted dynamic programming. In Annals of Mathematical Statistics, vol. 36, pp 226-235, 1965.

[25] L. P. Kaebling, M. L. Littman and A. W. Moore. Reinforcement learning: A survey. In the Journal of AI Research, vol. 4, pp. 237-285, 1996.

[26] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In Proceedings of the Advances in Neural Information Processing Systems 7, pp. 231-238, 1995.

[27] D. A. Pierre. Optimization Theory With Applications. Dover Publications, New York, 1986.

[28] Y. LeCun, J. Denker and S. Solla. Optimal Brain Damage. In Advances in Neural Information Processing Systems, vol. 2, pp. 598-605, 1990.

[29] B. Hassibi and D.G. Stork. Second-order derivatives for network pruning: Optimal brain surgeon. In Advances in Neural Information Processing Systems, vol. 5, pp. 164-171, 1993.

[30] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth. Occam's razor. In Information Processing Letters, vol. 24, pp. 377-380, 1987.

[31] J.B. Hampshire II and D.A. Watola. Diagnosing and Correcting System Anomalies with a Robust Classifier. In IEEE Proceedings of the 1996 International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3507-3509, May, 1996.

[32] B. A. Pearlmutter. Fast Exact Multiplication by the Hessian. In Neural Computation, vol. 6 (1), pp. 147-160, 1994.

[33] A.J. Lipton, H. Fujiyoshi and R.S. Patil. Moving target classification and tracking from real time video. In IEEE Workshop on Applications of Computer Vision, pp. 8-14, 1998.

[34] W.E.L. Grimson, L. Lee, R. Romano, and C. Stauffer. Using adaptive tracking to classify and monitor activities in a site. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 22-31, 1998.

[35] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 319-325, 1999.

[36] I. Haritaoglu, D. Harwood and L. Davis. W^4: Who? When? Where? What? A real time system for detecting and tracking people. In IEEE International Conference on Automatic Face and Gesture Recognition, pp. 222-227, 1998.

[37] C. Diehl, M. Saptharishi, J. Hampshire, and P. Khosla. Collaborative surveillance using both fixed and mobile unattended ground sensor platforms. In SPIE Proceedings on Unattended Ground Sensor Technologies and Applications, vol. 3713, pp. 178-185, 1999.

[38] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland. Pfinder: real-time tracking of the human body. In IEEE International Conference on Automatic Face and Gesture Recognition, pp. 51-56, 1996.

[39] C. Wren and A. Pentland. Dynamic models of human motion. In IEEE International Conference on Automatic Face and Gesture Recognition, pp. 22-27, 1998.

[40] S.J. McKenna, Y. Raja and S. Gong. Tracking color objects using adaptive mixture models. In Image and Vision Computing 17, pp. 225-231, 1999.

[41] M.J. Black, A.D. Jepson. EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. In International Journal of Computer Vision, vol.26, no 1, pp. 63-84, 1998.

[42] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In Fifth International Conference on Computer Vision, pp. 612-617, 1995.

[43] A. Bobick, J. Davis, S. Intille, F. Baird, L.Campbell, Y. Irinov, C. Pinhanez and A. Wilson. Kidsroom: Action recognition in an interactive story environment. In M.I.T. TR No: 398, 1996.

[44] S. Ju, M.J. Black, Y. Yacoob. Cardboard People: A parameterized model of articulated image motion. In IEEE International Conference on Automatic Face and Gesture Recognition, pp. 38-44, 1996.

[45] J.B. Hampshire II and B.A. Pearlmutter. Equivalence Proofs for Multi-Layer Perceptron Classifiers and the Bayesian Discriminant Function. In Proceedings of the 1990 Connectionist Models Summer School, pp. 159-172, 1991.

[46] J.B. Hampshire II and A.Waibel. A Novel Objective Function for Improved Phoneme Recognition using Time-Delay Neural Networks. In IEEE Transactions on Neural Networks, vol. 1, no. 2, pp. 216-228, 1990.

[47] J.B. Hampshire II. A Differential Theory of Learning for Efficient Statistical Pattern Recognition. Ph.D.Thesis, Carnegie Mellon University, 1993.

[48] C. S. Oliver, M. Saptharishi, J. M. Dolan, A. Trebi-Ollennu and P. K. Khosla. Multi-robot path planning by predicting structure in a dynamic environment. Submitted to IFAC 2000.