Remote Detection using Fused Data

Timothy Myles Payne, B.Sc. (Dest.), B.E. (Hons.)
The University of Adelaide
Department of Electrical and Electronic Engineering
Adelaide, 5005
Australia

August 31, 1994

Awarded 1995

[1]This work was supported by the Opto-Electronics Division, Surveillance Research Laboratory of the Defence Science Technology Organisation, Australia, the Department of Electrical and Electronic Engineering, the University of Adelaide, Australia, and the Cooperative Research Centre for Sensor Signal and Information Processing, the Levels, Pooraka, South Australia, 5095.

Declaration

This thesis has been submitted to the Faculty of Engineering at the University of Adelaide for examination in respect of the Degree of Doctor of Philosophy (by research).

It contains no material which has been accepted for the award of any other degree or diploma in any University, and to the best of the author's knowledge and belief contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

The author hereby consents to this thesis being made available for photocopying and for loans as the University deems fitting, should the thesis be accepted for the award of the Degree.

Signature_____

Date August 31, 1994

Abstract

The problem is detecting and tracking objects at large ranges, when no target features are visible, with imaging type sensors. A system is developed which estimates the optical flow of the scene in a parallel architecture similar to that of an artificial neural network. From the estimated motion of the scene, hypotheses are determined about the object type. These hypotheses are determined in a manner which leads to a bounded probability range driven by the data and which incorporates doubt and allows fusion of information from other sensors along similar lines to Dempster-Shafer reasoning. An uncertain probabilistic approach to decision making is employed so that sparse decision spaces and noisy data don't cause biased decisions due to generalisation. Incorporation of information from a different sensor type enables false objects caused by correlated noise sources such as multipath reflections to be removed.

The approach is designed to operate in a parallel architecture in high noise, high resolution situations where conventional approaches scale very poorly, consequently it is not envisaged that the approach presented in this thesis will compete with more conventional approaches which are applicable in different problem domains. The approach is proved and is simulated using both real and artificial information. The real test data comes from an IR array and a passive sonar sensor.

Contents

Symbol definitions

0.1 Notation

Attempts have been made to make the symbols and equations in this thesis as simple and yet general as possible. Many symbols refer to vectors or matrices yet are not labeled as such and may appear to be treated as scalars since it doesn't really matter what they are in terms of the equations.

Components of the vectors are selected by the application of a subscript. As an example consider x. Generally x is used to refer to the position of an object in space or in a structure. If the structure is one dimensional then x is a scalar, but it is also possible in higher dimensional problems that x has many components $x_0, x_1, \ldots, x_N$. Subscripts have two meanings, they can be used to reference components of a vector, such as x_0, they can also be used as an index to a particular vector. Where this is the case it should be obvious, and a special subscript such as α may be used to ensure the correct interpretation.

If x appears in an equation such as

$$z = ax + 3 + b \tag{0.1}$$

then evaluation is performed differently depending upon whether x, a and b are vectors. If x, a and b are scalars then calculation proceeds in a straight forward manner; however, if any of x, a or b is a vector – if more than one of them is a vector they must have the same size – calculation is performed on a point by point basis **not using matrix algebra**. As an example consider that a, b, x and z are all vectors of the same size then (0.1) can be rewritten as

$$\begin{aligned} z_0 &= a_0 x_0 + 3 + b_0, \\ z_1 &= a_1 x_1 + 3 + b_1, \\ &\vdots \\ z_N &= a_N x_N + 3 + b_N. \end{aligned}$$

In a similar way a function may appear without a parameter. The most common term left out in a function is t and this normally implies that all terms in the calculation have the same time indices; only when calculations require terms from different times does the parameter t need to appear.

If matrix or vector operations (eg. $\|x\|$) appear in expressions then those operations are performed using normal matrix algebra. The result of the operation is usually a scalar and is used in the rest of the expression in the same pointwise manner as described previously.

Matrix magnitudes unless otherwise specified are L2 norms.

0.2 Definitions

The following definitions appear roughly in the order of appearance of the symbols in the chapters.

U_A represents the highest probability that can be attached to a hypothesis A.

L_A represents the lowest probability that can be attached to a hypothesis A.

A_{L_A,U_A} represents the range of probabilities for a hypothesis, A. The first subscript is the lower probability, L_A, and the second subscript is the upper probability, U_A.

θ is the set of all hypotheses.

x is generally used to refer to a spatial coordinate and is generally of arbitrary dimension.

y typically represents the response of a sensor at a point. Array sensor responses consist of an array of ys and are usually represented by ϕ.

N total number of training points.

$target(x, y)$ is the hypothesis that there is a target at the augmented spatial position (x, y).

t_i is a label for the i^{th} training point location in x.

p is a label for a point located in x.

$n_i(p)$ represents the effective number of training points at the point p caused by the training point t_i.

$G_i(t_i, p)$ is the Green's function of the training point location ambiguity. If it is assumed that the ambiguity is due to a noise process such as sensor platform vibration then $G_i(t_i, p)$ is the probability function of the noise.

$\mathbb{X}$ is the set of all possible position indices, ie. $\mathbb{X} = [0, 1, 2, fred, anylabel]$ defines a space with points $p_0, p_1, p_2, p_{fred}, p_{anylabel}$. This enables equations to be formed that are independent of dimension and which are not necessarily uniform, eg. p_{fred} could represent the point $(0.2, 4.3)$ in a 2 dimensional Euclidean space or the point $(0, 0, 1)$ in a non-Euclidean 3 dimensional space.

n is the number of elements in $\mathbb{X}$.

t represents time and is always a scalar.

C_j is the set of all training points which satisfy hypothesis j only.

$N_j(p)$ is the total number of effective training points at the point, p, caused by the training points in the set C_j.

$N'_j(p)$ is the same as $N_j(p)$ except that additional training points have been added which are consistent with the original distribution of training points.

$f_j(p)$ is the relative frequency, which is an estimate of the probability that point p is a member of class j.

$f'_j(p)$ is the same as $f_j(p)$ except that additional training points have been added which are consistent with the original distribution of training points.

$\mathbb{Y}$ is the set of possible y values.

R is the number of elements in $\mathbb{Y}$ and corresponds to the resolution of the sensor.

ϕ represents the sensor response. It is generally a matrix indexed by x and t and is written as $\phi(x, t)$.

v usually represents velocity.

$\hat{\phi}$ is an estimate of ϕ formed from knowledge of ϕ at earlier times.

$\overline{\phi}$ is an estimate of the true noise free ϕ formed from $\hat{\phi}$ and noisy measurements of the true ϕ.

φ is similar to ϕ except that it has been augmented; see Section 2.2.3 and Section 4.5

$\mathbb{Z} = (\mathbb{X}, \mathbb{Y})$ is an augmented space which is indexed with an index from $\mathbb{X}$ and $\mathbb{Y}$

α is used either as a scalar parameter or as a label for a point in either the sensor state space or the augmented sensor state space. As a label to a point in the state spaces α is an element of the set of indices used in the space so that the dimension of the space is arbitrary. For example in a 7 dimension space α might equal $[0, 1, 1, 2, 1, 4, 2]$.

β same as α.

γ same as α.

$\mathbb{D}$ represents the set of coordinate axes so that in an n dimensional space $\mathbb{D}$ has n elements.

$|\mathbb{D}|$ is the number of elements in $\mathbb{D}$, see the definition of $\mathbb{D}$. In a 3 dimensional space $|\mathbb{D}| = 3$

i refers to a coordinate. For example in a 5 dimensional world with the coordinate axes $\mathbb{D} = \{d_1, d_2, \ldots, d_5\}$, i could represent any of the axes in $\mathbb{D}$.

j same as i.

k same as i.

T_t is a transformation at time t consisting of 2 vectors, $T_t^{\mu}(\alpha)$ and $T_t^{\sigma}(\alpha)$, in $\mathbb{D}$ attached to each point in the augmented state space.

$T_t^{\mu}(\alpha)$ is the mean direction vector at the point α at time t.

$T_t^{\sigma}(\alpha)$ is the variance vector at the point α at time t.

$?_i$ As a subscript to a fully labeled point (except possibly for t) or as the subscript of a label point such as α the subscript changes the term to the value of the i^{th} coordinate or component in $\mathbb{D}$ whether the label is in $\mathbb{D}$ or in another space which must be mapped into $\mathbb{D}$ first.

$\triangle_\alpha(t)$ is the prediction error at time t, and is the error between the predicted matrix, $\hat{\varphi}(t)$ and the actual matrix $\varphi(t)$.

n_o represents noise in the measurement process.

n_{T_t} represents noise introduced to the prediction process by objects manoeuvring in the scene.

$\overline{X}^e_{\varphi_\alpha(t)}$ is the expected value (1^{st} moment) of the error between the smoothed estimate of the matrix, $\overline{\varphi}_\alpha(t)$ and the actual matrix, $\varphi_\alpha(t)$.

$\overline{X}^{2e}_{\varphi_\alpha(t)\varphi_\alpha(t)}$ is the expected value (2^{nd} moment) of the square of the error between the smoothed estimate of the matrix, $\overline{\varphi}_\alpha(t)$ and the actual matrix, $\varphi_\alpha(t)$.

$\overline{X}_{n_o}$ expected value of n_o.

$\overline{X}_{n_{T_t}}$ expected value of n_{T_t}.

$\overline{X}^{equil}_{\varphi}$ is the value that a sequence of $\overline{X}^e_{\varphi_\alpha(t)}$ converge to.

$\overline{X}^{2equil}_{\varphi\varphi}$ is the value that a sequence of $\overline{X}^{2e}_{\varphi_\alpha(t)\varphi_\alpha(t)}$ converge to.

$p(x)$ is a probability function indexed by x.

Chapter 1

Introduction

1.1 General

In this document I am interested in tracking moving objects, which are devoid of features, within a noisy background. The purpose of this introduction is to introduce the reader to the particular problem. This is critical since engineering decisions were made in the course of development which can be justified only in the particular circumstances relevant to the problem.

To make intelligent decisions information is required. As we extend the area we wish to make decisions over the quality of the information may decrease and the quantity increase making the interpretation of the raw data, to obtain useful information, more difficult. In some applications we now extend our range of information acquisition using sensors capable of working day or night, with ranges and resolutions far exceeding that of the our natural biological sensors. The need to gather all this information for long periods, and the alien nature of the information to our brains, means that we can no longer analyze the results easily. The solution is to automate the analysis process and present only critical information or conclusions to a human operator, but even this amount of information can overload our meagre minds.

Complete automation, aside from the moral dilemmas associated with giving control to machines, is not possible because machines have difficulty making decisions in a vague uncertain world. Humans have always attempted to remove doubt by using multiple sensors to gather information; the farmer looking at the sky notes not just the color, structure and motion of the clouds, but also the fluctuations in the temperature and pressure. The Indian chief listens to the reports of all his scouts, and using the information he has learnt through

experience, he decides what should happen. Machines on the other hand deal with numbers or symbols. If the data is from several sources and generates different conclusions then the data must be wrong, yet a decision must be made, and the conflict between the sources resolved.

In our modern world one facet of this information gathering is the long range detection, tracking and classification of objects. This task, commonly known as surveillance, is the topic of this thesis, which attempts to develop an automatic approach to the combination of data from different sensors to form decisions which are meaningful and can be utilized by either humans or further automatic systems.

1.2 The Problem

The help in the surveillance task, data from two or more sensors is fused to improve the accuracy and completeness of the information obtained. The solution approach has civil and military applications, such as in the detection of missiles or the tracking of ships. The success of both these applications is limited using single sensor systems. Surface based radar systems for example suffer from multi-path limitations for low angle observations, and infra-red (IR) imaging systems require large amounts of processing to initiate and track objects moving in a dense clutter or noise environment, which occurs particularly at low angles over water due to specular reflections from waves.

In the approach presented, knowledge from disparate sources is combined, so that false objects, which are the result of noise processes that are correlated with the signal of interest, are removed. This correlated noise occurs in situations such as multipath reflections, and is removed using fusion since the correlated noise for each sensor is different, and agreement between the sensors occurs only at true objects. The assumption that the correlated noise is different at each sensor is valid because the sensors are located at different sites or are intrinsically different, for example a radar and an IR sensor view different portions of the electromagnetic spectrum, and so the correlated noise in each sensor is caused by different processes and will be different. Conventional approaches to the tracking and data fusion problem must be ruled out, since they do not provide conclusions with a measure of doubt which can facilitate fusion in situations

where the information to be combined is conflicting.[1] This measure of doubt is necessary if information from multiple sources is to be combined so that the information from a sensor, which may support or refute the information from other sensors, can be incorporated meaningfully.

The technique usually used to resolve conflicts between sensors is to assign an apriori weighting to sensor accuracies based on the confidence in decisions from a sensor. Unfortunately this apriori weighting generates the problem of determining the weights, which are often non-stationary and non-uniform over a sensor field. A new approach is proposed which attempts to avoid these problems by incorporating data-driven measures of doubt into all conclusions so that sensor conflicts can be resolved using weights which are a function of the data presented to each sensor.

1.3 Thesis Organization

This thesis is organized such that a person wishing to understand what I have done would merely commence reading at the start and progress sequentially. The review of current and past work has been separated from the main text and placed in the appendices so that the flow of ideas is not overwhelmed by review material. This was deemed necessary because of the number of fields covered, and because in many instances the review material is not a precursor to the original information in the chapters, but justifies the effort developing a new approach. Information about conventional approaches, which are invalid under the given assumptions are located in the appendices. The distinction between original and review work is easy, since if the material is in a chapter it's original; if it's in an appendix other than Appendix A, then it is either a review of others' work, or analysis of that work. Appendix A represents analysis and additional

[1] It is important to understand the difference between probability and doubt. A probability states the exact frequency of an event occurrence, while doubt has no such exact knowledge. If you tossed an unbiased coin 10 times you would expect the coin to land on each side and you can determine the probability of that happening since knowledge that the coin is unbiased gives you information obtained from an infinitely long trial. If you are given another coin, which may or may not be biased, then there is doubt, since no information exists about the behaviour of the new coin. Imposing a distribution on the expected probability of the new coin, based on a small test expresses a form of doubt, yet there is still doubt that the distribution is correct. If additional information arrives which conflicts with the results of the short test, then conclusions are impossible. There are however other representations of doubt other than statistical distributions which are able to deal with conflicting information and it is systems employing these forms of logic which are sought in this document.

explanations of original work in the chapters. Some people may find that reading the appendix before the bulk of the material will lead to a better understanding of some of the decisions; however whenever material in the appendix is relevant to material in the chapters, I have attempted to signify this situation with cross references.

The solution to the design is in a modular form, which leaves me, as an author, in a quandary whether to explain the overall structure which is meaningless without knowledge of the blocks, or explaining the nature of the blocks without knowledge of their purpose. There is no simple solution to this problem, and I ask that the reader forgive me, for the necessity to continually flip forwards and backwards through the text. In an attempt to minimize the cross referencing by the reader, I will commence, in Chapter 2, by deriving the structure of the solution in an intuitive manner, with vague descriptions which allude to results presented in the following chapters. The following chapters will give precise descriptions of the solution components, while the final chapter will cover some simulations on the entire network.

Chapter 3 and 4 contain the major original contributions of this thesis. The ideas in these chapters are novel and provide the driving force for the rest of the work.

Chapter 3 presents a method of classifying sensor outputs in a neural type architecture which is able to quantify the amount of information used in making the decision and consequently to assign a measure of doubt to the decision that was made. Chapter 4 describes a method of extracting information about the evolution of a sensor signal which leads to information about the objects being viewed. The information generated is probabilistic so that the next stage in the overall tracking network, the classifier presented in Chapter 3, has all the information it requires to make decisions which include a measure of the decision accuracy based on doubt in the sensor information. Proofs for the parallel algorithm are left until Appendix A. Chapter 5 presents the results of analysis of the overall system stability. This brings together many of the results in the earlier chapters to make statements about the overall performance. Chapter 6 contains the algorithm presented in the earlier chapters in one location with the general mathematical objects, replaced with specific data structures. This allows the reader to follow what is essentially an example system. The algorithm, as

applied to the example system, is used for the majority of the simulations in Chapter 7. Chapter 7 contains the results and analysis of simulations of the overall tracking and decision network.

Appendix A contains convergence proofs for the optical flow determination algorithm presented in Chapter 4 along with some explanation about what the convergence means in terms of sensor and scene characteristics. Appendix B contains a brief review of some of the other approaches which are typically used in the solution of tracking objects in noise. Appendix C lists several of the more common neural network architectures prevalent today. Although these networks are not directly utilised in the solution, the architecture and the motivation behind some of the networks was extended and modified to produce the solution presented. The Back Propagation neural net isexamined in more detail than some of the other approaches, and the effect of some of the assumptions in the approach and the consequent limitations of the method in the problem under consideration are considered. Appendix C also briefly considers some possible implementation technologies for neural networks. These are not explored fully and are presented just to show that implementation is indeed possible. Some assumptions have been made through the development of the algorithm, where these assumptions have been arbitrary the assumption has been made to make implementation easier. Appendix D contains a review of some typical approaches to data fusion, along with brief descriptions of some of the advantages and disadvantages of each. Appendix E summarises some of the issues and theory of Dempster-Shafer Reasoning, which is similar to the approach adopted through out most of the subsystem components that perform reasoning. The rest of the introduction contains definitions and explanations of the critical background areas for future chapters.

1.4 Data Fusion

The demand for information from electronic sensing systems often exceeds that obtainable from any single sensor. To obtain the required amount of information, data from several sensors must be combined. The combination is termed "data fusion" and can occur at a number of different hierarchical levels. Approaches are generally subdivided into three levels.(Labuz, 1988)

1. Alignment, Correlation and Assignment, Tracking, and Attribute fusion. This is the lowest level and generally involves the fusion of data or low level information to improve the resolution or accuracy of that data. An example of such a system would be the fusion of a TV image with an Infra-red image to obtain a better estimate of the scene. Each sensor detects features of the scene with different confidences. The high confidence detections from one sensor complement the weak detections from the other, so that the resultant image contains more accurate information about the object's features in the scene than could be achieved by either sensor alone.

2. Situation assessment and Threat assessment. This level of fusion makes decisions about incoming sensor information and evaluates the significance of the information. A fighter pilot seeing an approaching plane on his radar and knowing from intelligence information that there are enemy aircraft located in that area, will fuse the radar and intelligence information to estimate the threat the approaching aircraft represents.

3. Sensor Control. This is the highest level of data fusion and involves the control of the sensor information itself. For instance, in navigating a course there will be regions where the amount of information is small, and the task's success will be threatened. In the previous example, it is unacceptable to shoot a missile at every plane, because the pilot knows that there are enemy planes about, and that he is therefore at risk. What is required is more information. The pilot must seek more information about the plane's identity by directing his sensors at the potential threat or by seeking more information from other sensors.

There are many types of sensors used for the automatic detection and tracking of objects. The following list contains some examples of systems where the data could be combined from several sources to obtain improved surveillance.(Selzer & Gutfinger, 1988)

- IR (*Infra-red*)
- Conventional Radar
- Imaging Laser Radar

- Laser Radar (Vibrometry)
- Synthetic Aperture Radar
- Visual
- Acoustic sensors

In this thesis a general approach is introduced using an example of fusion of data from different types of visual or IR sensors and radar. The final solution is compatible with other sensors, and so fusion with other sensor combinations, using the same approach is possible. It is assumed that the radar information used in the fusion is simply bearing and asymuth information, which is the same information obtained by the sensors with which it is being combined. Additional information such as range and velocity are not incorporated as part of this work but can be included by simply increasing the dimension of the tracking problem associated with the radar sensor and deriving a new set of combination rules so that the fusion of different dimension trackers can be performed.

Conventional radar is commonly used in surveillance systems to detect and track objects automatically. IR imagers can also be used to perform tracking, although their normal role is in identification. The performance of trackers using these sensors degrades when the object becomes faint due to noise or distance, for example: radar resolution is often insufficient for identification, and IR systems require large amounts of processing to detect and track targets in noise (especially since they generally have a narrow field of view and may only view the object briefly.) To overcome the limited resolution in radar systems a great deal of work has been performed in the area of radar imaging, and in an attempt to obtain the required resolution, modern antenna arrays have become increasingly larger until their mechanical construction and placement on modern platforms is now difficult.

An alternative solution involves combining the information from both systems so that the limitations of each are overcome by the advantages of the other. In the simplest combination approach a conventional radar generates position and track information, while an imaging sensor operating in the IR spectrum attempts identification from the object features. This simple combination fails to fully utilize the available information. The imaging sensor can also provide

information about the position and track of the object while the radar sensor also provides information about some object features, such as the the propulsion method in the case of aircraft: jet or propeller. In an environment where the noise is correlated with the target, such as the case with multipath reflections in low elevation angle radar observations, the use of positional information from the imaging sensor would improve the performance of the radar sensor. A review of the field of data fusion can be found in White & Llinas (1990); Waltz (1981) and briefly in appendix D.[2]

As the range is extended the benefit of an imaging sensor is reduced since object features become unresolved. The object becomes no more than a noisy dot or pixel, yet the sensor still collects information, which in conjunction with the other sensor, gives information regarding the position of the object, which exceeds the performance of either of the individual sensors alone. The object can no longer be detected by its features contained in the IR sensor data and there is some doubt as to exactly where the object is. The IR information must now be processed in an approach similar to that used in Radar tracking. This is difficult, because when using Kalman or MHT techniques (see appendix B) on sensors which produce images, and particularly in noisy situations, the amount of processing required to solve the data association problem becomes prohibitive(Kuczewski, 1987). Once an object can be tracked, hypotheses regarding the nature and existence of the object can be developed, and so by improving the range at which tracking can be performed successfully the detection range of the overall sensor system is increased.

In an attempt to solve this problem an algorithm designed for a finely grained parallel architecture is employed. The resultant performance of the algorithm may not be as good as conventional routines; however the scaling problems of conventional routines (Kuczewski, 1987) make them totally inapplicable in the high noise environments for which the proposed routine is designed. Conventional approaches to the problem are not investigated in great depth, and performance comparisons are avoided because this algorithm is designed to operate in situations where conventional approaches fail to be practical.

[2]In this document the fusion of dissimilar data from two sensors is considered for tracking targets. The problems of command and control are not considered.

1.5 IR Systems

1.5.1 The Data

The IR systems considered provide image data at discrete points in space. This image data is provided to the rest of the system at discrete intervals in time so that the data is discrete in both the spatial and temporal dimensions. For long range surveillance systems the task is to track and consequently detect objects whose spatial extent is less than the point spread function of the imaging system, and whose features are therefore unresolved. This task, when performed in the presence of clutter and noise, can be tiring for human observers. When the problem of data fusion of Radar information is added the human observer may be overloaded and unable to cope with all the information. This overload may cause additional fatigue and lead to a suboptimal fusion of the data.

The data to be analyzed will also be subject to omissions or false information, either through limitations of the sensors or because of active interference. In this situation the manual task of analyzing the data becomes even more tiresome, and critical.

It is assumed that the coherence period of the background noise to a temporal model will be short.

1.5.2 Pre-Processing Techniques

Current techniques for filtering the data (without considering data fusion), to improve signal-to-noise levels, involve the use of linear optimal filtering using techniques such as the Wiener and matched filters (Tao, Hilmers, Evenor & Yehoshua, 1980; Otazo & Parenti, 1980; Takken, Friedman, Milton & Nitzberg, 1979)). One of the problems with these techniques is in determining the filter. In many cases the optimum filter will change over time, due to a change in scene or may even vary dramatically over a single scene (as would be the case if the horizon was in the scene.) For any real application the filter must adapt and this requires a significant amount of processing ability if the adaptation and filtering are to be executed in real time. The approach presented in this thesis avoids the use of these filtering techniques, since detection of objects is performed by recognising a difference between the dynamics of a pixel and the surrounding pixels. Thus knowledge of the noise is needed, and filtering would only make the

problem more difficult.

1.5.3 Tracking and Acquisition of Targets

The tracking of featureless targets in an IR image can be done a number of ways, but the most popular approach (excepting space based systems) appears to be *Probabilistic Data Association* (PDA), or in the case of multiple targets, *Joint Probabilistic Data Association* (JPDA). There have been a number of papers which have demonstrated the success of these techniques (Houles & Bar-Shalom, 1989; Reid, 1979; Fortmann, Bar-Shalom & Scheffe, 1983; Colgrove, Davis & Ayliffe, 1986; Singer, Sea & Housewright, 1974; Blackman, 1986). **All** of these implementations are incomplete in that they fail to consider all, or in some cases any, of the following possible scenarios(Reid, 1979; Kurien & Liggins, 1988)[3]:

- Multiple Targets.
- Missing Measurements.
- False Alarms.
- Track Initiation.
- Target Splitting.
- Target Termination.
- Unknown Target Dynamics.
- Unknown Statistics.
- Translation in Data between sensors. (ie. tracks from different sensors, of the same target, appear to be different.)

In fact JPDA, by itself, is not appropriate for the situation described in 1.2. See Appendix B.3 for more information.

To initiate a target track, information from a series of frames in time is required. This process is difficult to execute in real time, with existing computers,

[3]Reid (1979) provides a table listing which approaches fail to meet which criteria, this table has not been reproduced, since the deficiencies and features of the individual approaches is not important, only that a new approach can be justified by the deficiencies of the existing approaches.

yet it must be performed continuously if the system is to be useful. Since a time sequence of frames exists for target acquisition, processing would be less than optimal if this data was not used in maintaining and filtering a track— neither PDA, nor JPDA algorithms utilize the information from these multiple frames. Algorithms exist which make use of the data from several frames(Rauch, Futterman & Kemmer, 1981), but these routines make use of a high target speed with respect to a slow drift in the background, and are able to extract objects by comparing successive frames. This approach is valid in a downward-looking space platform; however in the problem case being considered it is inappropriate, especially for a bearing and azimuth only sensor, such as an IR array, where a target can approach the sensor head on, giving no apparent target motion relative to the background. Another approach is *Multi Hypothesis Tracking* (MHT) which stores information about the probability of all possible tracks (see Appendix B.4.2), so that the dynamics of previous frames are stored. Unfortunately this technique has a computational and memory load which grows rapidly with the number of possible object points, and is consequently invalid, without extensive modifications, given the problem assumptions. The velocity estimation solution established in Chapter 4 is a modified parallel version of MHT with a statistical distribution on the family of track estimates, which avoids the computational and memory load that can occur when using MHT.

What is required is a new algorithm, the requirements of which are listed in section 1.7

1.6 Radar Systems

There are many types of radar systems which process the data in different ways. For the purposes of this thesis, the type of radar system is assumed to provide azimuth and elevation information only. Other information which is available from most radars can be incorporated into the approach; however this makes the combination and feedback of global decisions more complex, and has not been performed. The bulk of the work is performed on data from an IR sensor, while the radar is included to provide a source of disparate information to demonstrate the utility of the fusion approach.

1.7 Tracking System Requirements

The objective is to create a system that is able to track the data and determine the existence of targets. It should be able to deal with all of the scenarios listed in 1.5.3.

The characteristics which should be possessed by the tracking and detection algorithm are:

- The algorithm should be able to determine a level of belief in its conclusions. ie. track initiation with a bounded uncertainty such as between 70% and 90% certainty.
- Decisions should allow for conflicting or supporting evidence, ie. Shafer-Dempster reasoning(Bogler, 1987).
- Decisions should not be forced when using inadequate information, as occurs in Bayesian techniques(Bogler, 1987).
- A degradation in the performance of one of the sensors (due to failure or counter measures) should be handled by altering the significance attached to the sensor.
- The significance attached to each sensor's conclusions should not be formed apriori, but should be a function of the observed data.

For the processor to perform all of these functions a brief history of the inputs is required. The greater the amount of input history stored, the greater the amount of data storage required, but from the evidence of previous initiation and track routines(Houles & Bar-Shalom, 1989; Reid, 1979; Fortmann, Bar-Shalom & Scheffe, 1983; Colgrove, Davis & Ayliffe, 1986; Singer, Sea & Housewright, 1974) this data is usually required for initiation anyway. It is possible to store the input history by keeping a copy of the sensors detections; however, this can require a huge amount of memory, particularly with large array sensors, which are typical of current IR arrays. Routines exist, such as multiple hypotheses tracking (MHT)[4], which rather than store the previous inputs, store some essential features along with the perceived dynamics of those features. These approaches

[4]Refer appendix B.4.2

require less memory for low noise, low object count scenarios, but actually require more otherwise.

1.8 Neural Networks

Neural techniques represent a method of programming parallel computers, as well as developing architectures for building and programming new computers which can even be based on analogue components. Many systems labeled as neural computers are not new, and similar solution approaches can be obtained by different methods.

Artificial Neural Networks (ANN)[5] provide a means of parallel processing, and are particularly suited to non-linear and statistical problems, which may require adaptation. Originally the intention was to use neural networks to implement some of the tracking and data fusion processes; however, there are some problems associated with conventional ANN, which has led me to develop a new method, which still exploits the fine grain computational architecture of neural networks but avoids some of the following problems:

- Can a neural network solve the problem? Many conventional networks can not be guaranteed to solve certain types of problem.

- Decision boundaries in neural networks, generally with a Sigmoid shape, are statistically accurate only when the noise processes have an approximate Gaussian distribution with constant variance. It is necessary to determine whether a mix of statistical decision boundaries can be implemented. Refer to appendix C.1.2

- Neural Networks have been shown to make Baye's optimal decisions (Wan, 1990; Ruck et al., 1990) under ideal conditions. A network needs to be found which can determine the validity of a decision along with a level of certainty.

- Due to the massive amount of input data to the network there is a large number of connections. It needs to be determined whether this large number can be implemented and trained or whether there is a relationship

[5]Some neural networks are reviewed in a cursory manner in appendix C.

between connections which will reduce the dimensionality of the training problem. The reduction in the size of the training problem will also assist in the adaptation speed of the network to changes in the environment during run time, since the dimension and hence size of the solution space that needs to be searched will be reduced. See Appendix C.8[6]

- One of the problems in implementing data fusion is determining which hypotheses should be evaluated. The number of possible hypotheses can be prohibitively large, making evaluation of them all difficult, or some of the hypotheses may be unknown at the time of design. Neural networks are capable of evaluating a finite set of known hypotheses. It is necessary to evaluate **all** hypotheses for each sensor so that disparate information from different sensors can be fused in a manner which is able to resolve conflicts between sensors. These conflicts can not be resolved when the only conclusion from each sensor processing system is the most likely hypothesis[7], rather than a measure of the merit of each hypothesis.(ie. the most likely hypothesis from all the data, may be different to the most likely hypothesis from an individual sensor.)

The solutions to the tasks considered use a parallel architecture with a connectivity which is consistent with ANN. The training is different from conventional approaches, which enables conclusions about the network performance to be made.

1.9 References

Blackman, S. S. (1986). *Multiple Target Tracking with Radar Applications.* Norwood, MA: Artech House.

Bogler, P. L. (1987). Shafer-Dempster reasoning with applications to multisensor target identification systems. *IEEE Transactions on Systems, Man & Cybernetics, SMC-17*(6), 968–977.

[6]The solution took advantage of a number of relationships which removed the need to train altogether.

[7]A situation which will arise with a nonlinear output stage on a neural network.

Colgrove, S. B., Davis, A. W., & Ayliffe, J. K. (1986). Track initiation and nearest neighbours incorporated into probabilistic data association. *Electrical and Electronics Engineering, Australia, 6*(3), 191–198.

Fortmann, T. E., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering, OE-8*(3), 173–184.

Houles, A. & Bar-Shalom, Y. (1989). Multisensor tracking of a manoeuvering target in clutter. *IEEE Transactions of Aerospace and Electronic Systems, AES-25*(2), 176–189.

Kuczewski, R. M. (1987). Neural network approaches to multi-target tracking. *Proceedings of the IEEE International Conference on Neural Networks, 1*, 347–361.

Kurien, T. & Liggins, M. E. (1988). Report-to-target assignment in multisensor multitarget tracking. In *Proceedings of the 27th, Conference on Decision and Control*, pages 2484–2488.

Labuz, J. (1988). Coordinated tracking of multiple point targets with multiple cameras. *Society of Photo-Optical Instrumentation Engineers, 1003*, 263–269.

Otazo, J. J. & Parenti, R. R. (1980). Digital filters for the detection of resolved and unresolved targets embedded in infrared scenes. *SPIE Proc. (Smart Sensors), 178.*

Rauch, H. E., Futterman, W. I., & Kemmer, D. B. (1981). Background suppression and tracking with a staring mosaic sensor. *Optical Engineering, 20*(1), 103–110.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control, AC-24*(6), 843–854.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M., & Suter, B. W. (1990). The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks, 1*(4), 296–298.

Selzer, F. & Gutfinger, D. (1988). Ladar and flir based sensor fusion for automatic target classification. *Society of Photo-Optical Instrumentation Engineers, 1003*, 236–246.

Singer, R. A., Sea, R. G., & Housewright, K. B. (1974). Derivation and evaluation of improved tracking filters for use in dense multitarget environments. *IEEE Transactions in Information Theory, IT-20*(4), 423–432.

Takken, E. H., Friedman, D., Milton, A. F., & Nitzberg, R. (1979). Least mean square spatial filter for infrared cenes. *Applied Optics, 18*(24).

Tao, T. F., Hilmers, D., Evenor, B., & Yehoshua, D. B. (1980). Focal plane processing techniques for background clutter suppression and target detection. *SPIE Proc. (Smart Sensors), 178.*

Waltz, E. L. (1981). Computational considerations for fusion in target identification systems. *Proc. NAECON*, 492–497.

Wan, E. A. (1990). Neural network classification: A bayesian interpretation. *IEEE Transactions on Neural Networks, 1*(4), 303–305.

White, F. E. & Llinas, D. J. (1990). Data fusion: The process of ccc i. *Defense Electronics*, 77–83.

Chapter 2

Fusion using Reasoned Feedback

2.1 Introduction

A consistent approach is presented for the fusion of multi-sensor information. The fusion process allows for different types of sensors which can be located at different sites and have no or only partial overlap in their coverage. The information from each sensor is processed locally to remove noise and evaluate hypotheses about objects in its field of view. These hypotheses' evaluations are transmitted to a central location where decisions from other sensors are fused using an approach similar to Shafer-Dempster reasoning (appendix E). The reasoned conclusion of the fusion is fed back to the local processing at each sensor to improve future hypotheses. Although the approach is applicable for almost any type of sensor system, for clarity the examples presented assume an imaging system working in the visual or IR spectrum.[1]

2.2 Network Structure

2.2.1 General Network Structures

There are two main architectures which can be classified into either central (figure 2.1) or local processing (figure 2.2) of sensor data(Maren, Pap & Harston, 1989; Clark & Yuille, 1990; Reibman & Nolte, 1987b; Reibman & Nolte, 1987a; Waltz & Llinas, 1990). In the central case all information is transmitted to a central site where it is combined. This case is optimum in that all information is available for the process of decision making; however the processing requires a large amount of centralized computation power and high bandwidth commu-

[1]This chapter consists primarily of information presented in Payne (1992)

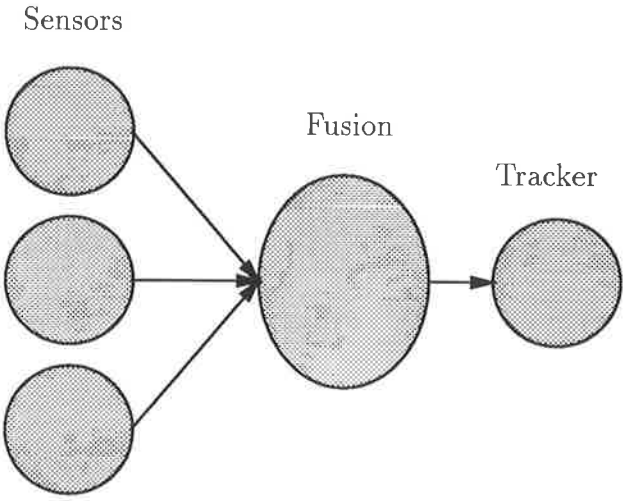

Figure 2.1: Centralized processing in a fusion network

nication channels. The local approach processes the sensor information at each sensor and sends the resultant information, tracks, or attributes, to the fusion center where information is combined. This approach requires less communication but suffers from the disadvantage that conflicts between sensors are difficult to resolve. In situations where classification of scene characteristics can be difficult and uncertain, faint or noisy objects, which in the optimal case could be detected by the central approach are missed by the local technique, since no single detector has sufficient information to make an informed decision.

The approach presented here is a combination of both the central and local processing architectures in figures 2.1 and 2.2. Information from the sensors is processed at the sensor site and a representation of doubt is included in the information transmitted to the central site. At the central site information from other sensors is incorporated in a reasoned manner to form a decision which makes the best use of the available data. The amount of information transmitted from each sensor will be less than in the central processing case since the data will have a lower entropy. In a situation where the sensors are distributed and possibly autonomous, the new approach is robust to system component or communication breakdown, since each of the autonomous units processes its own data so that the individual units are minimally affected by overall system failure.

2.2.2 Proposed Network Structure

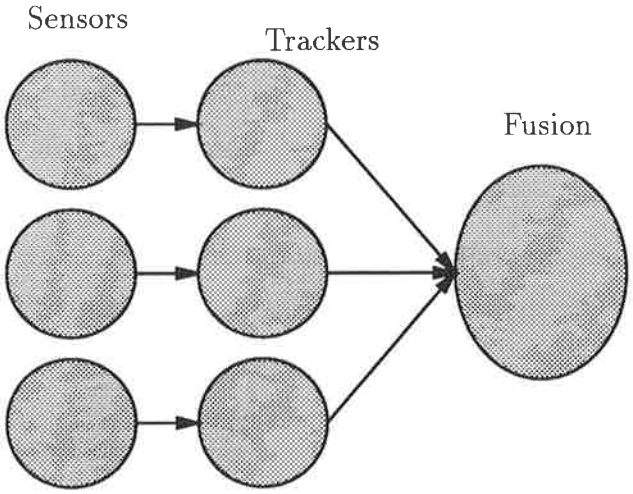

Figure 2.2: Localized processing in a fusion network

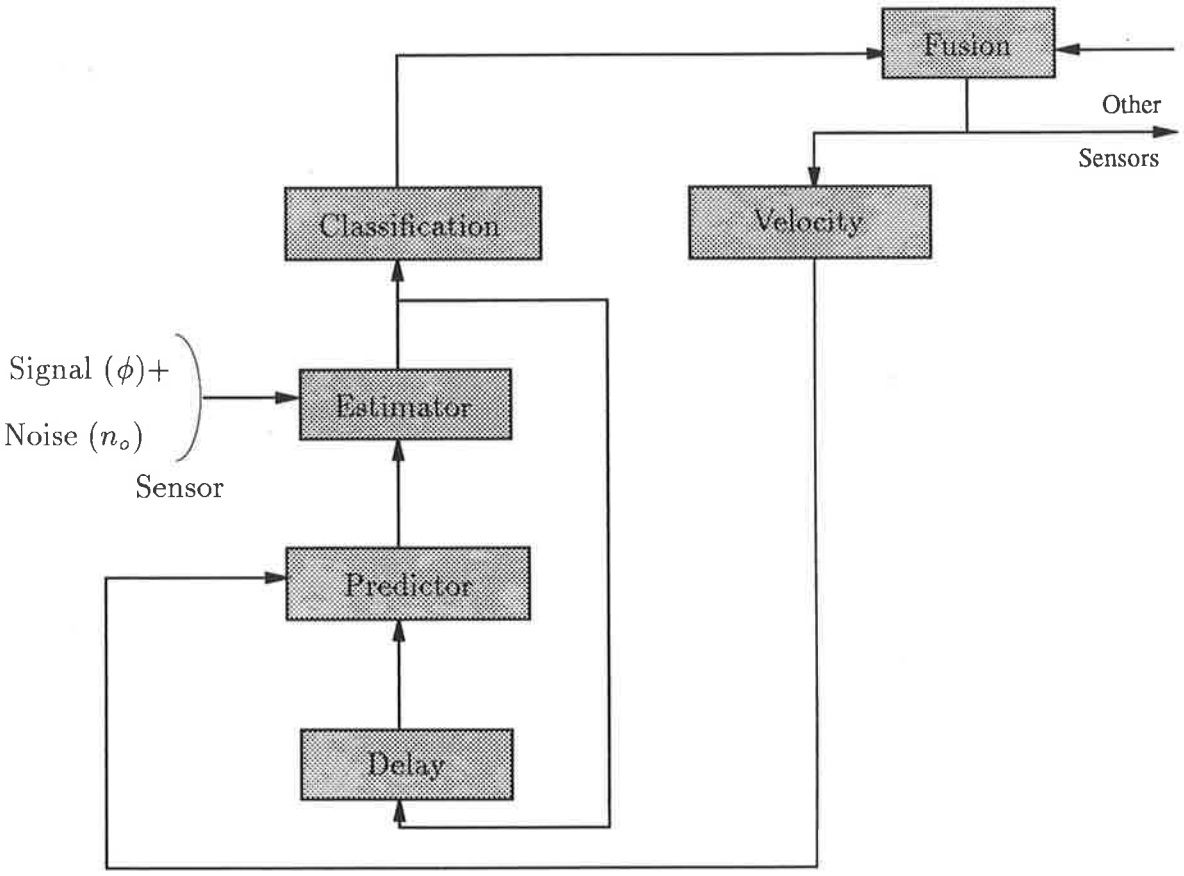

Figure 2.3: Structure of fusion network

The structure of the proposed architecture is shown in figure 2.3. The sensor produces a noisy output which consists of a portion representing the ideal world, which I shall call signal, ϕ, and a disturbance, which I shall call noise, n_o. The noisy output from the sensor is the input to the proposed network. The ideal world consists of a simplified version of reality upon which the models are based and the decisions made. The ideal world does not contain errors introduced by the medium between the viewed scene and the sensor, or by imperfections in the sensor system. The signal and noise can be represented as a matrix of data where each location in the matrix represents the intensity of a pixel at a particular azimuth and elevation. (for a non-pixelated sensor such as a radar the location in the matrix could correspond to range or velocity while the pixel intensity could correspond to the signal return strength. The dimension of the matrix could also be increased so that all of these parameters could be represented.)

In situations where the signal components are continuous they can often be improved by a process such as optimal filtering if the noise or signal adhere to some known model. For the proposed sensors optimal filtering is not possible since even the simple motion of an object moving between adjacent pixels will produce a near discontinuous change in the value of the signal components at the source and destination pixel locations (This is explained further in section 4.3). The signals are not really discontinuous since the sensors are assumed to have a finite integration time and scintillation by natural processes blurs points, such that motion produces a smooth change in the components of the signal. This motion although smooth, is rather abrupt, and leaves only a small potential for improvement by filtering when the relationships between components are not considered.

The relationships between components of the matrix are continually changing, since adjacent pixels only affect certain neighbours and this effect depends on the direction of the object motion at that location. Although there is a similarity in the proposed approach to that of Kalman filtering the extension of the Kalman filter equations to allow for relationships between components which vary continuously as a function of the **true** input would result in difficult and cumbersome equations, even if they were stable.

The proposed approach considers the interaction of all vector components in parallel when forming its predictions. The observed phenomenon (signal) must

obey some sort of temporal model, such that if the state of the phenomenon is known, prediction of future states to within a tolerable error is possible. Specifically, the case of tracking extremely faint objects is considered, where it is not possible to detect objects by their attributes. This makes the problem more difficult than when the objects features can be detected, since the uncertainty about an object's motion is increased due to the increased number of possible motions or associations. If the object was easily recognized then there would be no uncertainty in determining an objects perceived motion. If an object in noise has no features then it is impossible to locate and recognise the object in its new position and infer that it has moved from one point to another, hence determining its motion parameters, since the object could have moved to anywhere. Recognition of the object also allows a reduction in the dimension of the space in which tracking will occur since it is possible to assign a label to the object and attach it to a particular spatial location and track the evolution of the object's spatial location and motion parameters.

2.2.3 Local Processing

Each local unit modifies the sensor output by its expectation of the signal $\hat{\phi}$. This expectation is determined by a mapping function which attempts to predict what the next signal $\hat{\phi}(t+1)$ should be from the previous filtered sensor outputs $\overline{\phi}(t) \ldots \overline{\phi}(0)$.

$$\hat{\phi}(t+1) = f\left(\overline{\phi}(t), \overline{\phi}(t-1), \ldots, \overline{\phi}(0)\right). \tag{2.1}$$

The filtered sensor output $\overline{\phi}$ is a weighted average of the sensor output $\phi + n_o$ and the local prediction $\hat{\phi}$, so that

$$\overline{\phi} = \frac{a(\phi + n_o) + b\hat{\phi}}{a+b}. \tag{2.2}$$

The function f is actually implemented in a recursive manner so that f is a function of the current filtered sensor output, $\overline{\phi}(t)$, and a state vector representing the evolution of these filtered outputs, $v(t)$.

$$\hat{\phi}(t+1) = f\left(\overline{\phi}(t), v(t)\right). \tag{2.3}$$

$v(t)$ is calculated as the expected mapping from $\overline{\phi}(t-1)$ to $\overline{\phi}(t)$ given $v(t-1)$. The history in the function $v(t)$, and the values of a and b in (2.2), control the

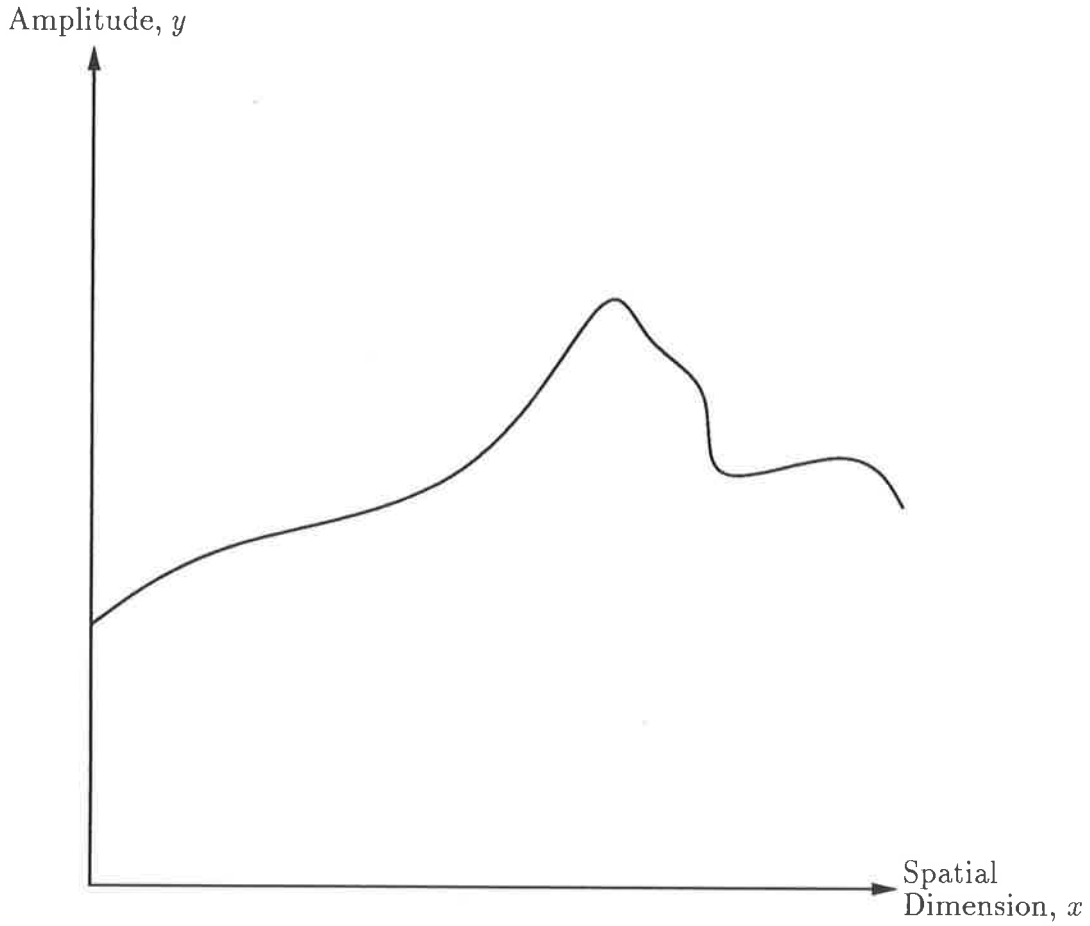

Figure 2.4: 1D signal

sensitivity and temporal bandwidth of f. $v(t)$ actually represents the velocity components of all the pixels. The reader is referred to Chapter 4 for more information on f. Since a probabilistic transformation is applied to the signal, the representation is augmented to represent all possible values with a probability assigned to each value. As an illustration of this augmentation consider a one dimensional signal $y(x)$, shown in Figure 2.4. The signal when augmented to two dimensions (Figure 2.5) is now represented by $p(x, y)$ which represents the probability of $y(x)$ for all combinations of x, y. The advantage of this representation is that it is now possible to track a number of possible signals (or objects) simultaneously.

Some of the objects (signals) to be tracked may result in very low probability tracks given the information from only one sensor. However when combined with the information from other sensors, which alone might also have generated a low

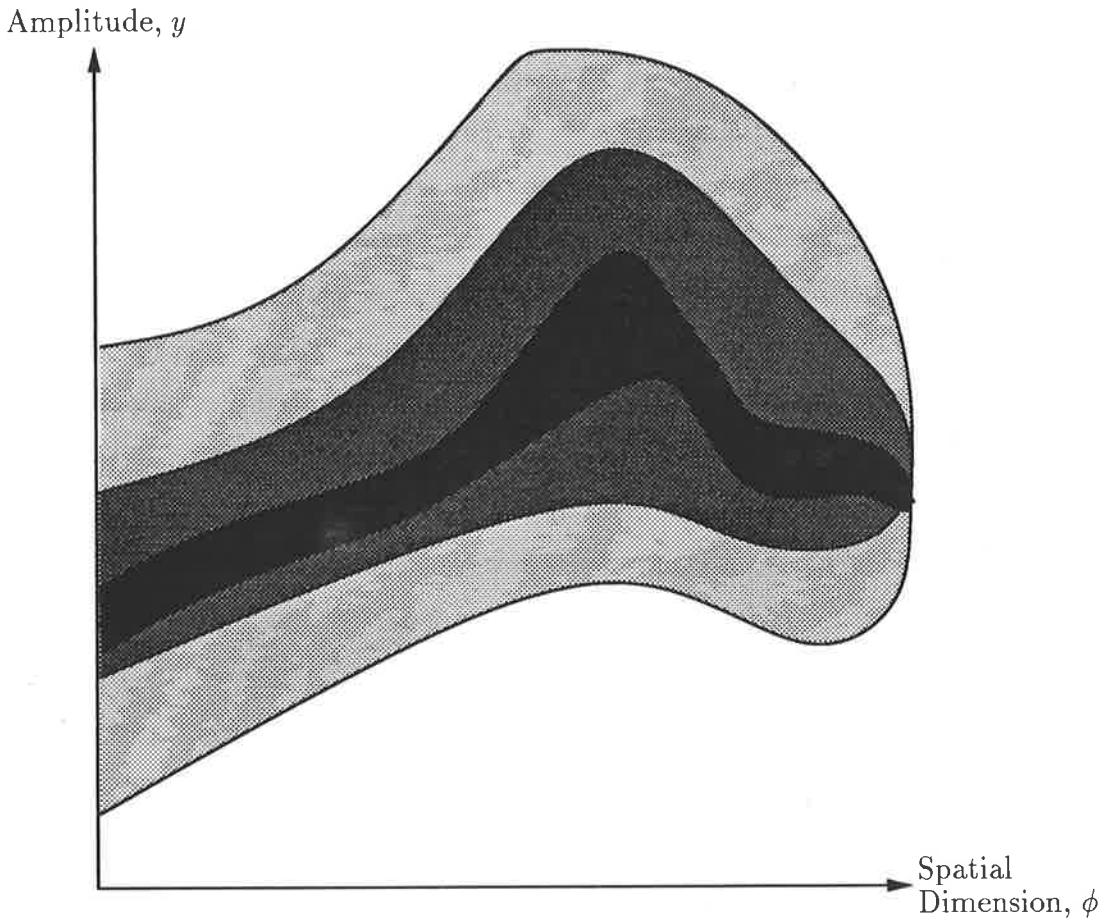

Figure 2.5: Contour plot of the augmented 2D equivalent to Figure 2.4. The contours represent the probability of the signal at the ϕ, y position. Darker shading represents higher probability.

probability track for the true object, the information contributes to finding the true track. By allowing low probability signals to be tracked and maintaining a measure of the track confidence, local processing followed by central fusion can be performed, while still obtaining some of the robustness obtainable when all the processing occurs at a central site. This is only possible if information about all the tracked objects is available at the fusion site and so local processors must evaluate all possibilities.

The mapping function f and the velocity $v(t)$ are also augmented so that f attempts to find maps between the augmented signals. It is necessary to understand that the mapping attempts to generate an estimate of the sensor output. Constraints on the estimation limit the prediction to that of the signal, since the coherence period of the background noise to any temporal model will be short, and the mapping function, f, is biased towards models which are consistent over longer periods of time[2]. The difficulty arises in trying to determine an appropriate model, or mapping function from the large number of possibilities. Analysis in Chapter 4 shows that for $a = 1 - \epsilon$ and $b = \epsilon$ in (2.2) the mapping will converge to the correct model, and that the moments of the probability error at any of the augmented positions after convergence are given by

$$\|\overline{X}_{\varphi}^{equil}\| = \frac{(1-\epsilon)\overline{X}_{n_o} + \epsilon\overline{X}_{n_{T_t}}}{1-\epsilon k} \tag{2.4}$$

and

$$\overline{X}_{\varphi\varphi}^{2equil} = \frac{(1-2\epsilon+\epsilon^2)\overline{X}^2_{n_o n_o} + \epsilon^2\overline{X}^2_{n_{T_t} n_{T_t}}}{1-\epsilon^2 k^2}, \tag{2.5}$$

where $\overline{X}_{n_o}$, $\overline{X}^2_{n_o n_o}$, $\overline{X}_{n_{T_t}}$ and $\overline{X}^2_{n_{T_t} n_{T_t}}$ are the first and second moments of the noise and target manoeuvres respectively and $k < 1$. The amount of noise removed and the sensitivity of the filter to manoeuvres are related such that optimization of one degrades the performance of the other, and since tracking of targets must be maintained, the amount of filtering available may be quite limited. Although convergence is guaranteed the algorithm requires time to converge and if the target of interest is manoeuvring, or the background varying, faster than the time taken for the algorithm to converge then the procedure will not work and

[2]Coherent noise is detected and tracked as if it were an object. This noise is removed at the time of fusion since the coherent noise will be different at each sensor and will result in conflicts between sensors when fusing which will invalidate the information.

another approach should be attempted; however experimental evidence indicates that the convergence is quick enough for the situations tested in Chapter 7.

Because of the high noise and convergence to the correct mapping function, which is only guaranteed in the limit as time goes to infinity, assistance is provided to the tracking modules by feeding back the global conclusions, (fused sensor estimates) to the sensors. This additional information is used, at the sensor, to bias the mapping function, thus reducing the number of possible models, and increasing the probability of finding the correct model. In particular the fusion of information from a central level reduces the noise associated with background information so that the component due to $\overline{X}_{n_o}$ is reduced. The consequence of the selective noise reduction is that the sensitivity to target manoeuvres can be set at a higher point, while still maintaining adequate filtering, since the trade off between signal noise and target manoeuvre sensitivity has now been broken. In addition, k may be reduced by the fusion of additional information further improving the filtering.

The noise filtered signal is now processed to extract features or tracks. These items are extracted in the block labeled as "Classification" in Figure 2.3. At each point in the sensor field, all possible hypotheses about the nature of the object at that point are evaluated and assigned a range of probabilities which are bounded above and below by an upper and lower probability, which are analogous to plausibility and support as defined in Shafer (1976) (see appendix E). The probability intervals are determined in a neural type of architecture which incorporates a representation of its knowledge obtained during training. The bounds on the implied classification frequency are obtained by searching valid alternative training spaces. The χ^2 test is used to determine whether hypothetical input spaces are consistent with the real training space. This input space search network is covered further in Chapter 3. It should be noted that the probability intervals obtained are not true plausibility and support in the sense of Shafer (1976) but approximations to it; as the χ^2 hypothesis test certainty approaches 1, the functions approach those of Shafer (1976). Typically a value of 0.95 is sufficient though, and this is used here.[3] The reason for deriving a

[3] 0.95 is arbitrary. Selection of a value too close to 1 will result in a meaningless result, since the Gaussian distribution has an unbounded domain and the hypothetical training set would consist of all sets so that the upper and lower probability would be 1 and 0 respectively, which reveals nothing.

range of probabilities is that the space where hypotheses are to be made has a large dimension, and any reasonable amount of training data must be sparse in this space. Given that the training information only covers a small portion of the decision space, new decisions, unless they occur at a previously trained point, must be inferred from existing trained points, yet although we know what happens at the training points we really have no knowledge about what happens in between. The process of generalising the results from training to cover untrained points involves modeling the decision space. The most common model is that the probability of a hypothesis being true or false varies smoothly between training points, yet this is a model which requires additional training information to establish, or validate. Without assigning a model apriori to the decision space, generalization between training points cannot be guaranteed, and since no knowledge of a model is assumed, determination of a hypothesis at all unknown points in the decision space would require the hypothesis decision network to possess an infinite amount of information. In this work the problem is circumvented by assuming that the model is imprecise and instead of generating a single probability that a hypotheses is true, an interval of probabilities is generated.[4]

Since an interval of probabilities is being used the usual combination rules for probabilistic decisions are inapplicable. The following definitions and terminology will be used to help explain the combination of the bounded hypothesis probabilities. Upper probability U_A represents the maximum possible probability that the hypothesis A may take had more training been possible. Similarly the lower probability L_A represents the minimum probability. The amount of doubt in the decision is represented by the difference in upper and lower probabilities, so that in the extreme cases where $U_A = 1$ and $L_A = 0$, there is no information about hypothesis A, and when $U_A = L_A$ there is no doubt in the probability that the hypothesis will be correct. Note that this does not mean that the hypothesis is known to be true or false, but that there is no generalisation present, and that the probability of the hypothesis being correct is known completely. The representation of doubt prevents uncertain decisions, which are reached by extrapolating from training information, from dominating the fusion process.

[4]This is similar to Fuzzy reasoning except that Fuzzy reasoning goes further and assumes a model of the doubt rather than just the bounds.

2.2.4 Central Processing

The upper and lower probability vectors (one upper and lower probability for every spatial position or pixel) from each sensor represent the view of that sensor in its own local coordinates. Other sensors may be located in a different spot or have different sized fields of view which are also represented in each sensor's own local coordinate frame. Before combination can take place all of the information from each sensor must undergo a transformation so that it is aligned on a global space common to all the sensors. This alignment is not a simple process, and I have ignored this problem so that the effort can be concentrated on the tracking and fusion aspects of the network. If a sensor's field of view does not cover the common global space then the regions that the sensor does not cover can be filled with the vacuous hypothesis ($U = 1, L = 0$), which allows the combination of the sensors to occur with out varying the combination procedure over the global space as the number of sensors covering a point in space varies (Figure 2.6). The operation of mapping the local information to the global coordinate frame along with padding using the vacuous hypothesis can be performed at either the local classification or the central fusion site; however it makes more sense to perform it at the central site, since the amount of information to transmit after padding with the vacuous hypothesis increases, especially if the sensor coverage is only a small portion of the global field.

The information from all sensors is combined in the Fusion layer by a process similar to Shafer-Dempster reasoning[5], using the following rules (Garvey et al., 1981) and terminology:

The statements to the left of the $\rightarrow$ allow the statement on the right to be inferred. The probability that the hypothesis is true is bounded by a lower and upper probability, these are written as a pair of subscripts to the hypothesis they represent. For example A_{L_A,U_A} says that hypothesis A has a probability of being true which is bounded below by L_A and above by U_A.

[5]The combination rules are also the same as the combination rules associated with Fuzzy logic, except that only the bounds of the function are combined since nothing is really known about the shape of the distribution between the bounds. As an extension to this work Fuzzy functions could be generated by varying the certainty used in the χ^2 test to generate a whole series of bounds thus synthesizing a surface.

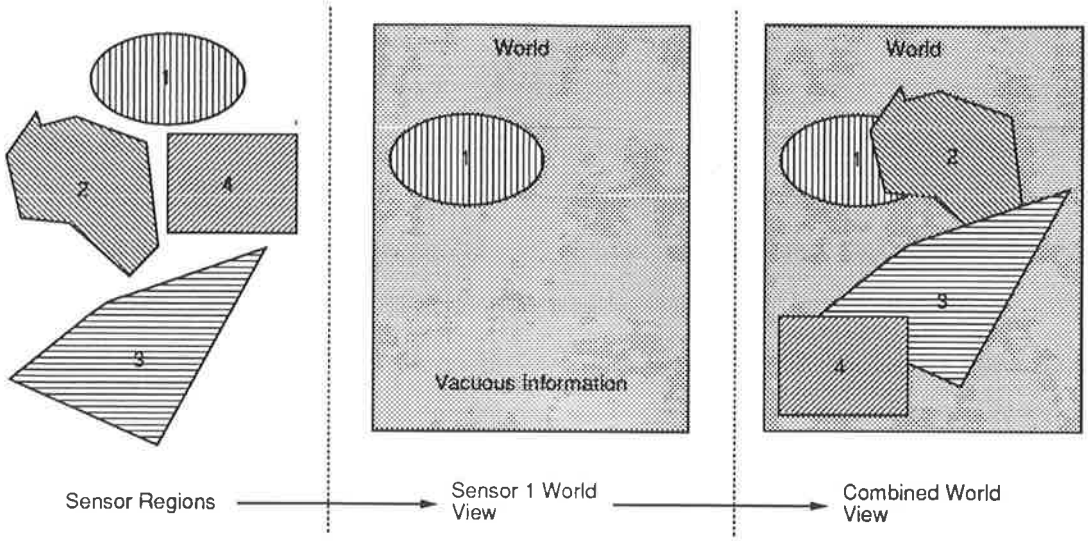

Figure 2.6: Generation of Mapped hypothesis. The figure shows how the regions covered by each sensor (sensor 1 in the example) are located in a world coordinate system and padded with vacuous information to fill the world coordinate space before combination with the aligned and padded information from the other sensors.

$$A \subset \theta \rightarrow A_{0,1} \tag{2.6}$$

$$A_{L_{1A},U_{1A}} \ , \ A_{L_{2A},U_{2A}} \rightarrow A_{L_A,U_A} \tag{2.7}$$

$$A_{L_A,U_A} \ , \ B_{L_B,U_B} \rightarrow (A \cup B)_{L_{A\cup B},U_{A\cup B}} \tag{2.8}$$

$$A_{L_A,U_A} \ , \ B_{L_B,U_B} \rightarrow (A \cap B)_{L_{A\cap B},U_{A\cap B}} \tag{2.9}$$

where:

$$L_A = \max(L_{1A}, L_{2A})$$

$$U_A = \min(U_{1A}, U_{2A})$$

$$L_{A\cup B} = \max(L_A, L_B)$$

$$U_{A\cup B} = \min(1, U_A + U_B)$$

$$L_{A\cap B} = \max(0, L_A + L_B - 1)$$

$$U_{A\cap B} = \min(U_A, U_B)$$

θ represents the set of possible hypotheses.

(2.6) is the vacuous hypothesis where no information is available.

(2.7) represents the combination of information from different sensors about the same hypothesis.

(2.8) and (2.9) represent the combination of different hypotheses from different sensors.

The conclusions reached by the fusion layer represent an improved estimate of the real world over the individual sensors. Correlated noise has been removed or at least discredited, and faint objects which are improbable to the individual sensors, are now more probable since they are supported by multiple sensors. This information can be fed back to the local sensors to aid in the estimation of an object's motion. One of the problems with the local estimation of the motion parameters was that objects could not be identified and the consequent association problem became extremely complicated. The global information simplifies this by identifying certain points as objects or background, this enables improved estimates of parameters to be determined, ie. background velocities should be consistent, and can be averaged together to obtain an improved estimate. The global fused information represents the probability that an object exists at a particular position and has a particular motion, unfortunately this hypothesis probability is difficult to translate back into a picture that should be seen by the sensor. To incorporate the information, the local mapping function, $v(t)$, at each sensor is biased by the global motion estimates. The incorporation of the information in this way has a number of benefits. The sensor is essentially told to look for an object moving in a certain manner, if the object does not exist in the sensor's scene then it will still not find it. If an object does exist but is too faint to be detected on its own, then the bias will provide assistance which increases the chance of the sensor finding the object and its motion.

2.3 Network implementation

The construction of a fusion network is explained to aid the reader's understanding of the approach. For simplicity a system consisting of only two sensors is discussed. These sensors are different, so that the detected target possesses different features in the sensors' returned signals. For the example, make sensor 1 a two dimensional IR imaging sensor, and sensor 2 a visual imaging sensor looking at the same scene. Assume that these systems have the same resolution and that

points in the sensor field can be indexed as (ϕ, y). These systems represent the information, energy in the IR and visible spectrums, in a two dimensional array which is augmented to three dimensions so that all the possible energies at a particular (x, y) are assigned a probability. This allows the individual sensors to track all possible signals, rather than just high target probability signals.

Target extraction units operate on the augmented signals to determine the existence of a target at each spatial location and energy in the IR or visible spectrum, so that a three dimensional set of hypotheses evaluations are formed representing the probability of a target at (x, y) having a particular energy in each of the spectrums. Where the sensor's spatial regions overlap the information from the sensors can be combined. **Since there may be no correlation between the energy in the IR or visual spectrums of a target**, information about target position must be reduced to the common two dimensional spatial frame before fusion can occur. This is performed by combining all the information about classification in each sensor at the same spatial position with (2.8), such that if $target(x, y)$ represents the set of hypotheses where a target is present at position (x, y), the hypothesis is evaluated for the union of all hypotheses $A \in target(x, y)$. The modifications to (2.8) for more than 2 hypotheses is straightforward so that

$$L(A_1 \cup A_2 \cup \cdots \cup A_n) = \max(L(A_1), L(A_2), \ldots, L(A_n)) \tag{2.10}$$

and

$$U(A_1 \cup A_2 \cup \cdots \cup A_n) = \min(1, U(A_1) + U(A_2) + \cdots + U(A_n)), \tag{2.11}$$

$$\text{for } \{A_1, A_2, \ldots, A_n\} = target(x, y). \tag{2.12}$$

These hypotheses can now be combined along the lines of (2.7) at each (x, y), to derive a global conclusion about target presence. This is the information presented to an operator or successive processing stages. There are three possibilities for the information, and the effect they have on future hypotheses.

- doubt $U > L$ In this case the sensors enhance each other's decisions enabling each to pick up a faint target it may otherwise have missed.

- certainty $U = L$. The probability of the target class is known.

- conflict $U < L$ In this case the sensors disagree and the conflicting hypotheses are penalized so that the sensors are forced to make their hypotheses from local considerations.

The global hypotheses at each spatial position is now used to bias the mapping function to look for objects with the dynamics believed to exist, given the results of the central fusion. If the object does not actually exist, then the sensor will still not find it. If the biasing of the mapping function at a sensor allows a consistency within the noise to be found, then the sensor will have found the object believed to be there and will reinforce the global conclusion.

2.3.1 Local and Global System Interaction

The structural advantages of this system lie in the robustness of the system organization. It is possible to break the system at almost any point above that of the local processing, and it will still perform, although less than optimally. This feature has also enabled analysis of the overall network characteristics, which are formed around the assumption that the local operations are stable and the feedback is only weakly coupled(Clark & Yuille, 1990). The actual analysis of the local structure and the particular nature of its operation are outlined in the following chapters. Operating independently the local unit forms a mapping function which is calculated as the expected mapping between filtered inputs. (see Chapter 4) This mapping is augmented by knowledge from the central site such that the local velocity state vector (Optic flow) $v_l(t)$, is modified to the weighted average of the expected value, given the local data and the probabilities of the various global target and background dynamics. If we define the local state estimate $v_l(t)$, the target state to be $v_t(t)$ and the background state to be $v_b(t)$ then the new estimate of the state $v(t)$ at each point in the augmented space is given by

$$v(t) = \frac{v_l(t) + L_{g,target}U_{l,target}v_t(t) + L_{g,background}U_{l,background}v_b(t)}{1 + L_{g,target}U_{l,target} + L_{g,background}U_{l,background}}, \quad (2.13)$$

where U_g, L_g, U_l, L_l represent the global and local upper and lower probabilities of target or background existence at a point (x, y). It should be remembered that the global hypotheses information applies to all signal levels at a particular spatial coordinate. In a system where the central fusion unit fails, there is still an advantage in feeding the sensor's own hypotheses back to bias the optic flow

determination, since use can be made of additional information regarding the nature of the points. As an example the points hypothesized to be background should have the same velocity components but due to prediction errors, atmospheric disturbances and other noise the predictions will vary over the scene. The classification of background points allows an estimate of the true background motion to be made from all the classified points and to correct the perceived motion of the background points that are fused back into the mapping unit.

Before the hypothesis determination network has been trained, (2.13) is invalid for the fusion of information from a central level back to the local level, since L_l and U_l are not defined, consequently the operation of the network must be divided into a training and an operational phase so that each local unit operates independently until its classification network is fully trained, and the feedback from the global unit can be applied.

2.4 Network operation

The benefits of this approach to the combination of sensor information can be demonstrated in a number of examples.

Example 1 *Consider the case where two sensors are looking in a common region of space. One of the sensors detects a target at a particular position with a higher probability than the other. Using (2.13) to combine the information the possibilities itemized in section 2.3 will occur. As a result of competitive interaction between the local sensor systems over time, a combined hypothesis will eventually be formed.*

Example 2 *Two sensors are looking in a common region of space where noise is correlated with the objects. Each sensor sees multiple objects because there is consistent motion of both the object and the noise; however due to different view points and/or regions of the electromagnetic spectrum that the sensors see, the noise correlated with the object may be different in each sensor. Using (2.13) to combine the information the possibilities itemized in section 2.3 will occur. As a result of competitive interaction between the systems over time, a combined hypothesis will eventually be formed which eliminates the correlated noise, since no agreement occurs between the sensors at these points.*

Example 3 *Consider the case where the coverage of two sensors is only partially overlapping such as sensor 1 and 2 in figure 2.6. In regions covered by only one sensor, hypotheses are unaffected by the other sensor's information, since it is vacuous and consequently has no affect. The overlapping regions are processed as in the previous examples. This consistent approach to the fusion of sensors with different views is important, especially in the case where the sensor systems could be moving relative to each other.*

2.5 Summary

A consistent approach has been presented which handles some of the more difficult tasks associated with data fusion, in a natural manner. The proposed approach bases the majority of computation at the local level, thus enabling the fusion of sensors over large areas by reducing the bandwidth of the communication required between the sensors and the fusion center, while still allowing global hypotheses to affect the conclusions at individual sensors. Simulations and analysis which demonstrate the validity of the proposed approach appear in chapter 7.

2.6 References

Clark, J. J. & Yuille, A. L. (1990). *Data Fusion for Sensor Information Processing Systems.* Kluwer Academic Publishers.

Garvey, T. D., Lowrance, J. D., & Fischler, M. A. (1981). An inference technique for integrating knowledge from disparate sources. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence, 1*, 319–325.

Maren, A. J., Pap, R. M., & Harston, C. T. (1989). A heirarchical data strusture representation for using multisensor information. *SPIE (Sensor Fusion II), 1100*, 162–178.

Payne, T. (1992). Central fusion of remote disparate processed sensor information using reasoned feedback to enhance sensor processing. In Green, D. & Bossomaier, T. (Eds.), *Complex Systems: From Biology to Computation*, pages 248–259. IOS Press.

Payne, T. (1994). Central fusion of remote disparate processed sensor information using reasoned feedback to enhance sensor processing. *Complexity International, 1.*

Reibman, A. R. & Nolte, L. W. (1987a). Design and performance of distributed detection networks. *IEEE Transactions of Aerospace and Electronic Systems, AES-23*(6), 789–797.

Reibman, A. R. & Nolte, L. W. (1987b). Optimal detection and performance of distributed sensor systems. *IEEE Transactions of Aerospace and Electronic Systems*, *AES-23*(1), 24–30.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton: Princeton University Press.

Waltz, E. & Llinas, J. (1990). *Multisensor Data Fusion.* Artech House.

Chapter 3

Signal Classification with Uncertainties

3.1 Introduction

This chapter presents an approach for classification of a signal in noise using a parallel network which has a connectivity similar to an artificial neural network.[1] The approach presented is similar to a probabilistic neural network, except that the outputs have been modified to give a representation of the level of doubt (or the amount of generalization) required to form the decision. This type of data representation is important when the classifier represents only a portion of the overall network. By determining the possible generalization error made during classification, successive decision makers can use this information to make decisions which are not biased by the original classification error, which was caused by improper generalization, or by a sensor failure at a lower level. The region of uncertainty is a function of the information, so the quality of decisions from individual decision makers will vary with the input.

3.2 Background

The determination of statistically optimal decisions using the information from multiple sources is a problem which is significant in a number of fields including communication, control, surveillance, and transportation, as well as in large tasks where partitioning of the problem into modules enables development to be managed easily.

[1] This chapter consists primarily of information presented in Payne (1993) and Payne (1992)

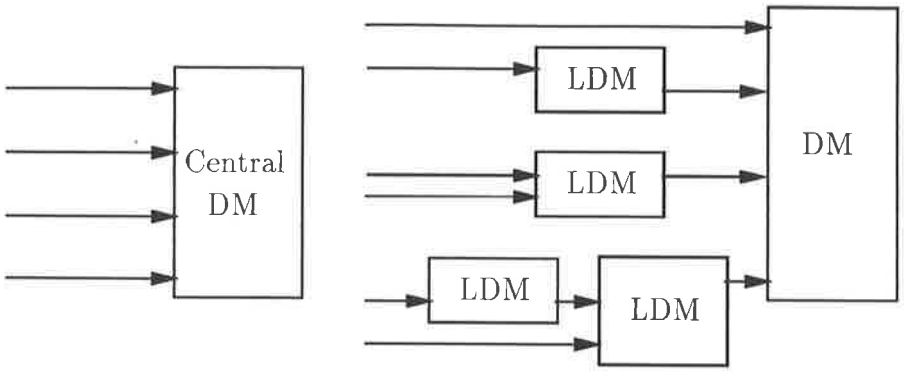

Figure 3.1: Examples of centralized decision makers (DM) and local decision makers (LDM)

There are a number of reasons why a modular solution is sought for the tracker (see Section 2.2.2) :

- The network could be distributed over a large area,, and partitioning of the problem into modules allows computational burdens to be distributed.
- The computational distribution improves the operational robustness.
- Both the modular design and the distributed computation make modification of the network by adding or removing additional sensors straightforward.

The nature of the network connectivity generally affects performance with a centralized network having the theoretical optimum decision performance, since it possesses the greatest amount of information when making a decision. Examples of a distributed decision network and a centralized system are depicted in figure 3.1. There are operational disadvantages with a central decision network, such as a lack of operational robustness, and a high communication bandwidth between the central site and the sensors. The lack of operational robustness is due to the optimal manner in which the information is processed. If part of the information becomes faulty due to sensor failure for instance then this situation must be identified and the system must be replaced or redesigned given the information that a component has failed. The identification of a sensor fault is difficult. The high communication bandwidth is caused by transferring all the raw data, rather than processed information to a central site. Much of this data

contains no valuable information while the distributed system filters out this garbage and only transmits valuable information to the central site. Distributed decision networks have been covered in a number of papers, Reibman & Nolte (1987b); Reibman & Nolte (1987a); Drakopoulos & Lee (1992). These papers either ignore the implications of the uncertainty in the sensed signal or assume that there is also knowledge regarding the uncertainty in the decisions made from a sensor. This uncertainty is generally fixed for particular decisions from each sensor. In some situations the information from a sensor makes it possible to assign a level of confidence to its decisions. In this situation a fixed region of uncertainty unnecessarily penalizes the decision making process. The information which controls the amount of uncertainty may come from additional information from the sensor, such as in example 4, where the size of the image controls the amount of doubt in determining what the image is.

Example 4 *Consider a television recognition system. Objects which are close will be presented in more detail, so that better decisions can be made about their identity. This closeness is easily determined by the decision making network, so that the uncertainty can now be a function of what the sensor sees.*

The level of uncertainty can also be made a function of the amount of generalization required for the sensor to come to a particular conclusion. The uncertainty caused by generalisation is the approach dealt with in this chapter, and is demonstrated in Example 5.

Example 5 *If the decision network is trained to recognize particular features, and then asked to classify something with similar but different features then the extrapolation of the decision to these similar features is known as generalization, and a measure of dissimilarity between the known and observed features can modify the level of doubt attributed to a decision.*

The question remains as to how the region of uncertainty about a decision should be determined. Conventional decision making techniques including maximum entropy (Zhuang, Zhao & Huang, 1991), fuzzy boundaries (Hunt, Qi & DeKruger, 1992), and smoothness constraints (Joerding & Meador, 1991) are based on assumptions about class boundaries, while the approach presented here enables decisions to be made from sparse training information, where it is assumed only that the statistics of the measurement noise processes are known. Consequently the knowledge required is of the measurement system, rather than the measured system. The approach is similar to that of probabilistic neural networks (Specht,

1990; Specht, 1991) except that conclusions are represented as upper and lower probabilities for each class. The procedure for combining the information is covered elsewhere in the literature (Tessem, 1992) and shall not be mentioned further. This chapter concentrates on the classification of the raw data and the representation of the doubt in the decision.

It should be noted that this approach does not generate a specific classification, it presents evaluations and determines their significance. The information could then be used with an objective function, which considers the implication of a classification error, to form decisions in a manner as presented in Drakopoulos & Lee (1992).

3.3 Theory

In the proposed approach all possible classifications are evaluated and assigned a range of probabilities which are bounded above and below by an upper and lower probability. The mechanism of how these probabilities are determined is presented in the next section of this paper. A range of probabilities is determined, since a single probability would require the classification network to possess an infinite amount of information. Although the input dimension of the network is not extremely large (3, 2 spatial and one signal dimension) other sensor systems can have very large dimensional inputs so that the training information in the input space will be sparse. The actual state of the input will generally not occur at a state where training data existed. The classifications generated are the result of extrapolating from known states, and consequently there is some uncertainty in the decisions made.

A noisy signal can be represented in the signal state space as an impulse function convolved with the noise density function.

Each of the training points represents a known classification at a known point. The effect of convolving each training point with the noise function is that points near the training point can represent a known classification, but with an effective number of training points less than one.

Let the training points occur in the state space at points $x = t_i$ where the index $i = 0, \ldots, N-1$, and N is the number of training points. At a point $x = p$ in the state space the number $n_i(p)$ of effective training points associated with a

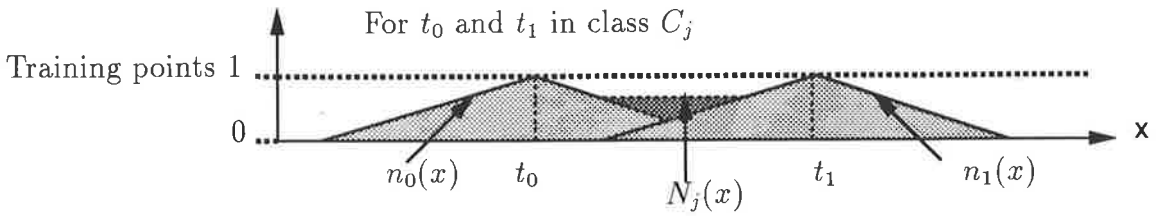

Figure 3.2: Effective number of training points

classification from the training point at $x = t_i$ is given by

$$n_i(p) = G_i(t_i, p), \tag{3.1}$$

where $G_i(t_i, p)$ represents the probability function of the noise around the point $x = t_i$ evaluated at the point $x = p$. This function has a number of possibilities, among them is the Gaussian which approximates the actual function of many natural phenomena. The Gaussian is used in simulations in a later section of the paper. The Gaussian is a consistent estimator (Parzen, 1962) which allows the class distribution to be approximated given that it is smooth and that the first partial derivatives are small (Specht, 1967).

For each class there is a subset C_j of training points t_i, where all the t_is which satisfy hypothesis j are members. Let θ be the set of all classes.

It is possible to estimate the probabilities of the various classes at a particular point, $p = x$, by evaluating their frequencies, $f_j(p)$, where

$$f_j(p) = \frac{N_j(p)}{\sum_{k\in\theta} N_k(p)} \tag{3.2}$$

$$\text{and } N_k(p) = \sum_{t_i \in C_k} n_i(p). \tag{3.3}$$

$N_j(p)$ represents the total effective number of training points of classification j at the point $p = x$. (Figure 3.2)

$f_j(p)$ represents the relative frequency, which is an estimate of the probability that point p is a member of class j.

It is possible that had more information been used during training, then the function $f_j(x)$ would have been different; however assuming that the data comes from the same distribution it is possible to determine bounds on the possible $f_j(x)$. Using the χ^2 distribution it is possible to compare the derived frequency function $f_j(x)$, with another set of frequencies $f'_j(x)$, to determine whether, with

the given information, they are not significantly different. Compute

$$\chi^2 = \sum_{j\in\theta} N_j^x \sum_{i\in\theta} \frac{(f_i(x) - f_i'(x))^2}{f_i(x)}. \tag{3.4}$$

If

$$\chi^2 < \chi_k^2(1-F) \tag{3.5}$$

is satisfied then the distributions are not significantly different.

k is the number of degrees of freedom, which in the case of no additional constraints, is given by $k = \|\theta\| - 1$, where $\|\theta\|$ is the number of classes.

F represents the significance level required. For example $F = 0.95$ gives a 95% confidence interval on $f_j(x)$. The 95% confidence interval is arbitrary. The confidence interval needs to be large enough so that it will almost always bound the correct solution, but not so large that the upper and lower probabilities of the class distribution give a possible generalization error which makes any conclusive decisions impossible. The upper and lower probabilities are not exact and are only estimates for which 95% of situations will be correct. Had a noise distribution been used which was zero outside a finite region of its domain, then it would be possible to use the 100% upper and lower bounds. Using a Gaussian which has a non zero probability throughout its domain means that the 100% upper and lower bounds will always generate $U_j = 1$ and $L_j = 0$, which is the vacuous hypothesis, and provides no information for successive stages.

The aim is to find the maximum and minimum value of $f_j'(p)$ for all j at $x = p$ which satisfy (3.5). These are calculated using (3.2) but substituting $N_l'(p)$ for $N_l(p)$ where

$$N_l'(x) = \begin{cases} N_l(x) + a_j(x), & \text{for } j = l \\ N_l(x), & \text{for } j \neq m. \end{cases} \tag{3.6}$$

A search is performed for the maximum value of a_j which still satisfies (3.5). The resultant $f_j'(x)$ is the maximum possible value of $f_j'(x)$ and is associated with the upper probability. The search over a_j is performed for all $j \in \theta$, so that the upper and lower probabilities are given by

$$U_j(x) = \max_{a_j, \chi^2 < \chi_k^2(1-F)} f_j'(x) \tag{3.7}$$

$$\text{and } L_j(x) = \min_{\forall m} \min_{a_m, \chi^2 < \chi_k^2(1-F)} f_j'(x). \tag{3.8}$$

The Upper probability, U_j represents the maximum possible probability that the classification j could take, if more training had occurred. Similarly the lower probability, L_j, represents the minimum probability. The possible generalization error in the classification is represented by the difference between the Upper and Lower probabilities, so that in the extreme cases where $U_j = 1$ and $L_j = 0$, there is no information about classification j, and when $U_j = L_j$ there is no generalization error. Note that when $U_j = L_j$ there could still be doubt about the true classification since what is known is the exact probability of each classification, not the exact classification. By representing the potential generalization error and transmitting it to successive levels in the decision making structure subsequent decision makers are prevented from attributing too much value to an uncertain decision. The combination and propagation of this information is covered further in Tessem (1992).

3.4 Implementation

Each decision maker has two distinct phases of operation: training and evaluation. In the evaluation phase the decision maker implements the functions outlined in Section 3.3. The training of the decision maker requires the storage of all the training points along with the class that they belong to. In some situations this may require an extremely large amount of storage, and may consequently require a large amount of computational time to follow the algorithm in Section 3.3. It is possible to implement this operation in parallel hardware which has a structure similar to that of a Kohonen neural network. (See Appendix C)

Figure 3.3 shows a possible parallel implementation of the decision network. Until now the points in the decision space have been indexed by x. In many of the figures x is a scalar since this makes diagrams easier to draw, but this is not necessarily true and x could be a vector. The components of the decision space vectors are shown in the figure using superscripts, while subscripts are used to enumerate the training points and the hypotheses. This violates previous terminology conventions and this terminology applies to this section only. The convention change is required because Figure 3.3 requires both labels for the vectors (subscripts) and indices to the vectors' components (superscripts, but in all other sections it is a subscript.) The input vector x is assumed to be constant,

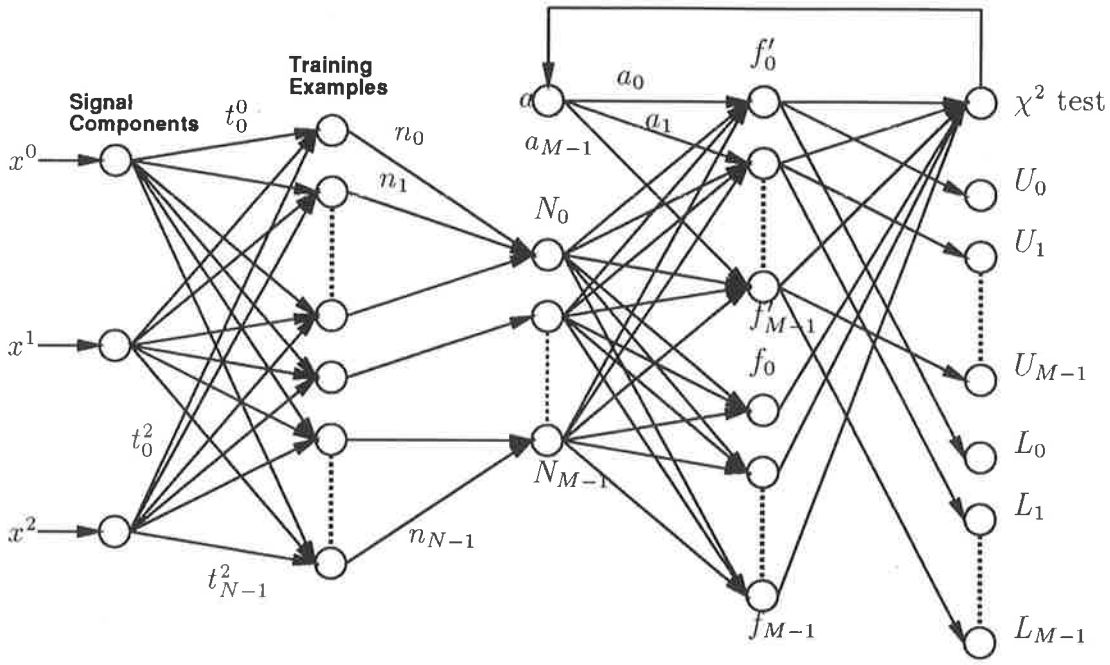

Figure 3.3: Possible neural implementation of decision maker

and hence reference to it is unnecessary. There are M possible classifications. The connections to each neuron (processing element) in the second layer (labeled as "Training examples" in Figure 3.3) represent a trained classification center, such that a signal component is modified on passing through these connections, to the difference between the signal value and the trained value, in the case of the first vector component, $t^0 - x^0$. The signal arriving at the neuron from the input is thus a measure of the distance between the training and input vectors in the decision space. The neuron combines the resultant components through the noise probability distribution function such that its result is n_i, (3.1). Connections to the next layer connect only neurons trained with the same classification. The n_i are summed at the neuron to N_j, (3.3). f_j are computed by the next layer using (3.2) while f_j' are computed using the modification made to (3.2) outlined in (3.6). The neuron in the figure labeled as "a" provides a vector output with a different component, a_j, being transmitted to each f_j' neuron. The value of this output vector controls the search outlined in (3.7) and (3.8) with the χ^2 test, (3.5), being performed by another neuron and the result fed back to the "a" neuron to constrain the search to the valid region where (3.5) holds. The U_j and L_j neurons in the final layer find the maximum and minimum value respectively

of their input for all "a" which satisfy the χ^2 test in (3.5).

The limitation of the neural network type implementation of this classification algorithm is generally not speed but the number of parallel processors. If the number of training points N exceeds the total number of neurons in the training layer then training information will be lost. The amount of information loss can be minimized in circumstances where the training points for each class are distributed in a smooth and continuous way. In this case all of the information can be represented by modifying the trained neurons and the noise distribution functions, so that the modified centers and noise distribution functions approximate all the training data. The amount of information lost will be a function of the number of training points which are modified and the complexity of the real training point distribution. If all of the training data is known a priori then there are a number of optimization techniques for determining the optimal set of modified training points(Hartman, Keeler & Kowalski, 1990; Park & Sandberg, 1991). The situation of online learning is a more difficult one, since the network stores all the previous training classifications in the neuron connections of the training layer. To add another training point the connections must be modified, so that the N modified neurons in the training layer now approximate the response, as well as possible, that a network with $N + 1$ neurons in the training layer would have produced, without losing any of the information from previously learnt training points. Iterative approaches to the solution of this problem are possible but this requires many more training points or representation of previous training points, which may not be possible.

In the simulations which follow in Section 3.5 each training point is represented by a single neuron. In Chapter 7 simulations use either one neuron for each training point or use an offline training procedure.

3.5 Simulation

To demonstrate the versatility of the decision making process a number of simulations were performed, using the approach presented in this chapter, on trivial decision tasks, rather than the complete tracking network where it is difficult to separate the characteristics of the individual components. The problem is to classify a new point into the appropriate class without knowing the correct

class boundary. The only information is some sample (training) points where the correct classes are known. The problem is compounded by the fact that all the information about each point's location is affected by noise, the characteristics of which are known. The simulations presented here pertain to a one dimensional input field, which has been normalized to the $[0, 1]$ domain. There are only two possible classes, $\theta = \{0, 1\}$ with the class boundary located at $x = 0.5$. The noise in the point's position is considered to be zero mean Gaussian, with a variance of 0.05. To study the classification performance of the algorithm presented in Section 3.3 points distributed uniformly in $[0, 1]$ were classified. The results of the classifications are plotted against the points value in the following figures. Figure 3.4 and 3.5 show the upper, nominal, lower probabilities and the doubt (or generalisation) obtained after 50 training points are distributed uniformly between 0 and 1.

There are several interesting points in Figures 3.4 and 3.5 which should be noted. The nominal probability curve divides the regions as would be expected. The separation of the upper and lower probability curves, which represents the amount of possible generalization error, varies despite the fact that the training information is distributed evenly. This is caused by two factors: firstly the possible generalization error is larger at the edges, since points near the boundary have their receptive region reduced by the intervention of the boundary, thus reducing the number of training points within the noise function about points near the boundary of the domain (Figure 3.6); secondly there is an increased generalization error about the point where the class frequency equals 50%; this is caused by the variance of the relative frequency. If the true probability were p then the variance of the relative frequency is given by $\frac{p(1-p)}{n}$ where n is the number of samples. The variance of the relative frequency as a function of the true relative frequency is symmetric and attains a maximum at $p = 0.5$, and so an estimate of the relative frequency is less reliable the closer the true value is to $\frac{1}{2}$. The generalization error is consequently a function of both the relative frequency variance and the number of samples, if training points were not distributed uniformly, then the resultant doubt curve would have a different shape.

The effect on classification of changing the training set to 5 training points can be seen by comparing class 0 probabilities in Figures 3.4 and 3.7. With fewer

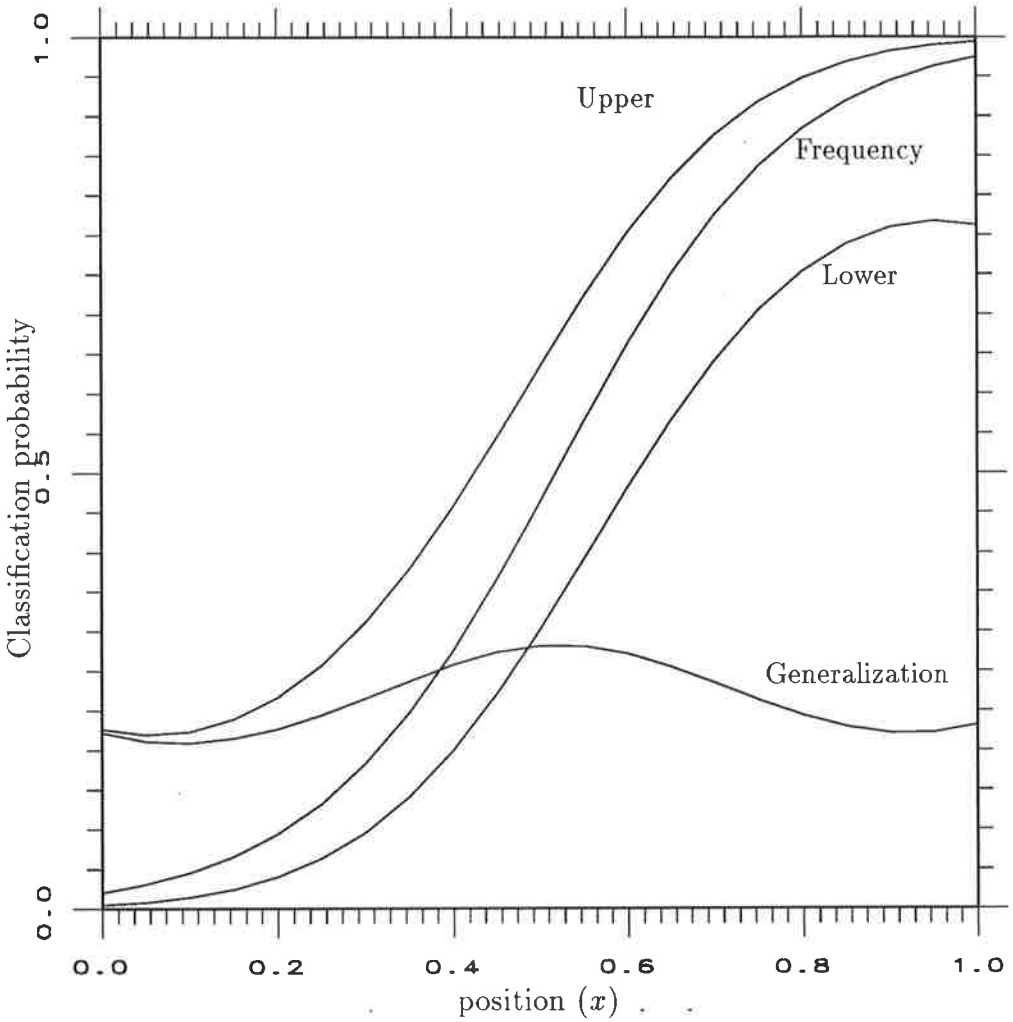

Figure 3.4: Classification 0 for a class boundary at $x = 0.5$ with 50 training points distributed uniformly over x. The curves are the upper, implied, lower and generalization error probabilities.

training points the region of doubt has increased. In Figure 3.8 the training information was not uniform, but Gaussian distributed about $\frac{1}{4}$ for class 1 and $\frac{3}{4}$ for class 0. Since most of the data is clustered about these centers, the doubt is lower here.

The nature of the classifications in Figures 3.4 to 3.8 is extremely simple, yet the approach is also applicable for more complicated regions. Consider the situation where points are classified as type 1 if the n^{th} digit after the decimal point of its coordinate is greater than the first digit after the decimal point of its coordinate. (e.g. for $n = 7$, at $x = 0.3xxxxx4x$ the correct classification is

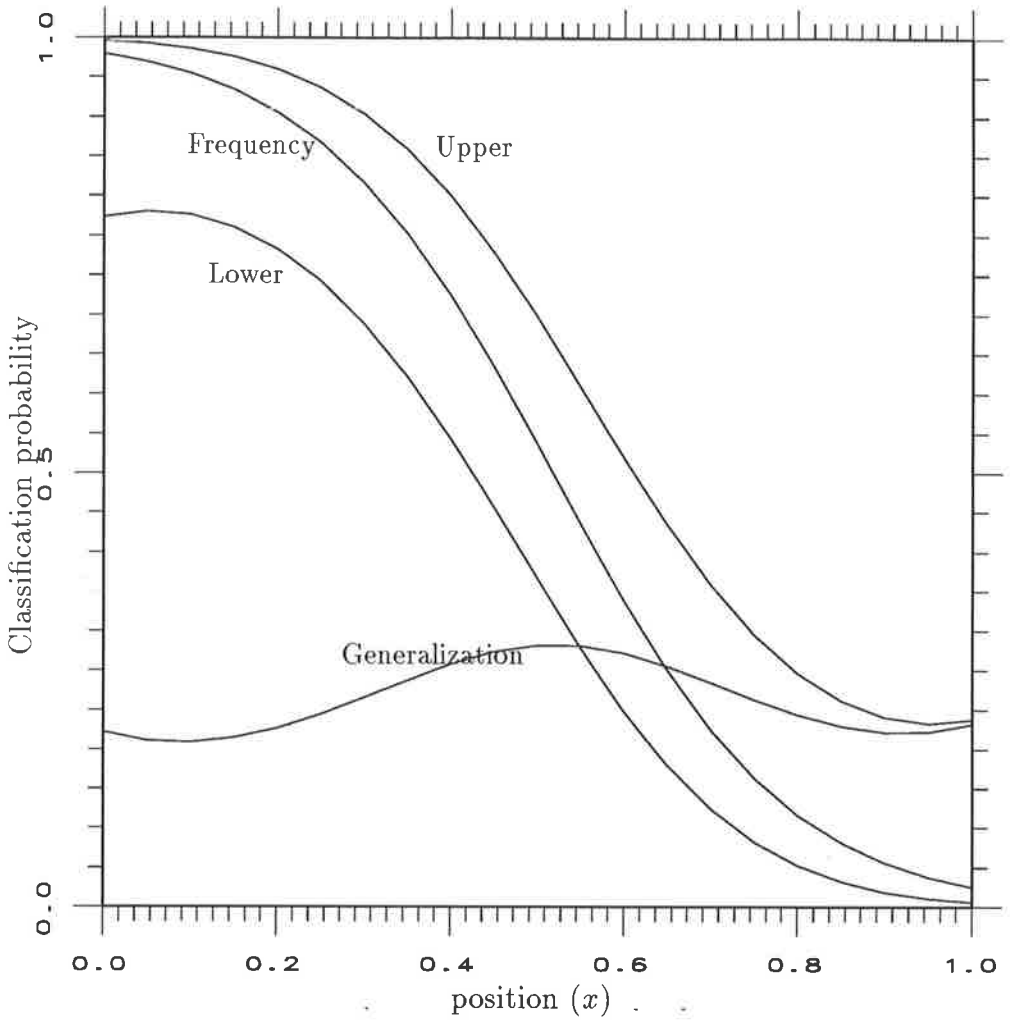

Figure 3.5: Classification 1 for a class boundary at $x = 0.5$ with 50 training points distributed uniformly over x. The curves are the upper, implied, lower and generalization error probabilities.

type 1, at $x = 0.4xxxxx3x$ the class is type zero) The result is a complicated clustering of regions of each hypothesis such that the density of class 1, in a region which is much larger than the scale at which the classes are changing, will be equal to the position along the number line. For n large it is impossible to model the precise boundaries between the classes, since the cell size of the classes is small compared with the variance of the noise. The best that is possible for a classification where the sensor sensitivity is insufficient to make a decision is to assign probabilities to the various alternatives. The result of the simulation is shown in Figure 3.9, where it can be seen that the region of true probabilities

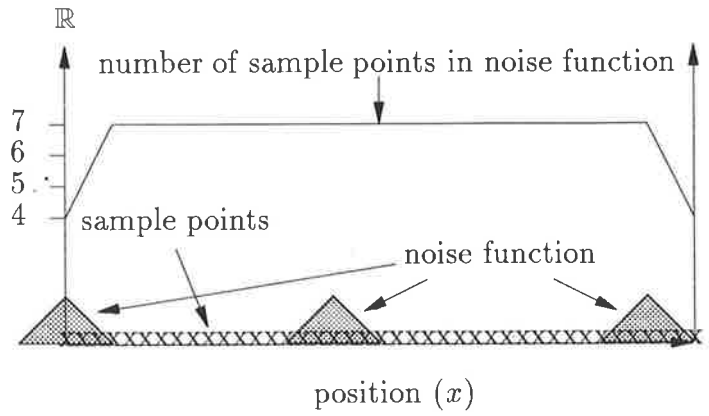

Figure 3.6: The number of sample points covered by the noise function centered at each point is reduced at the edges. This has the effect of increasing the generalization at the edges since less information is available.

(with a significance level of 95%) includes the true density, represented by the diagonal line.

Another example of a difficult classification task due to the fine resolution of the class boundary structure occurs when the class is distributed fractally, but without a smooth class density structure as occurred in the previous example. An example of such a classification task is obtained using a Cantor dust to define the class structure. A Cantor dust is constructed by setting the middle third of a graph of height 1 to 0. The remaining portions of the graph which are 1 are then likewise divided by setting the middle third to zero, and so on (figure 3.10. The result is a fractal which is mainly zero but has spikes at certain intervals. In this situation classification (Figure 3.11 is poor because the class densities do not vary smoothly, and the network's resolution is inadequate to model the boundaries.

3.6 Summary

This classification network performs classification of its inputs, and includes a representation of doubt with each conclusion. The merit of this approach is that no assumptions are made about the class boundaries to enable generalization to occur between known classes. The network can be used in situations where the noise model is known, but the nature of the class boundaries isn't, without fear of affecting consequential decisions which utilize the results of a classification made

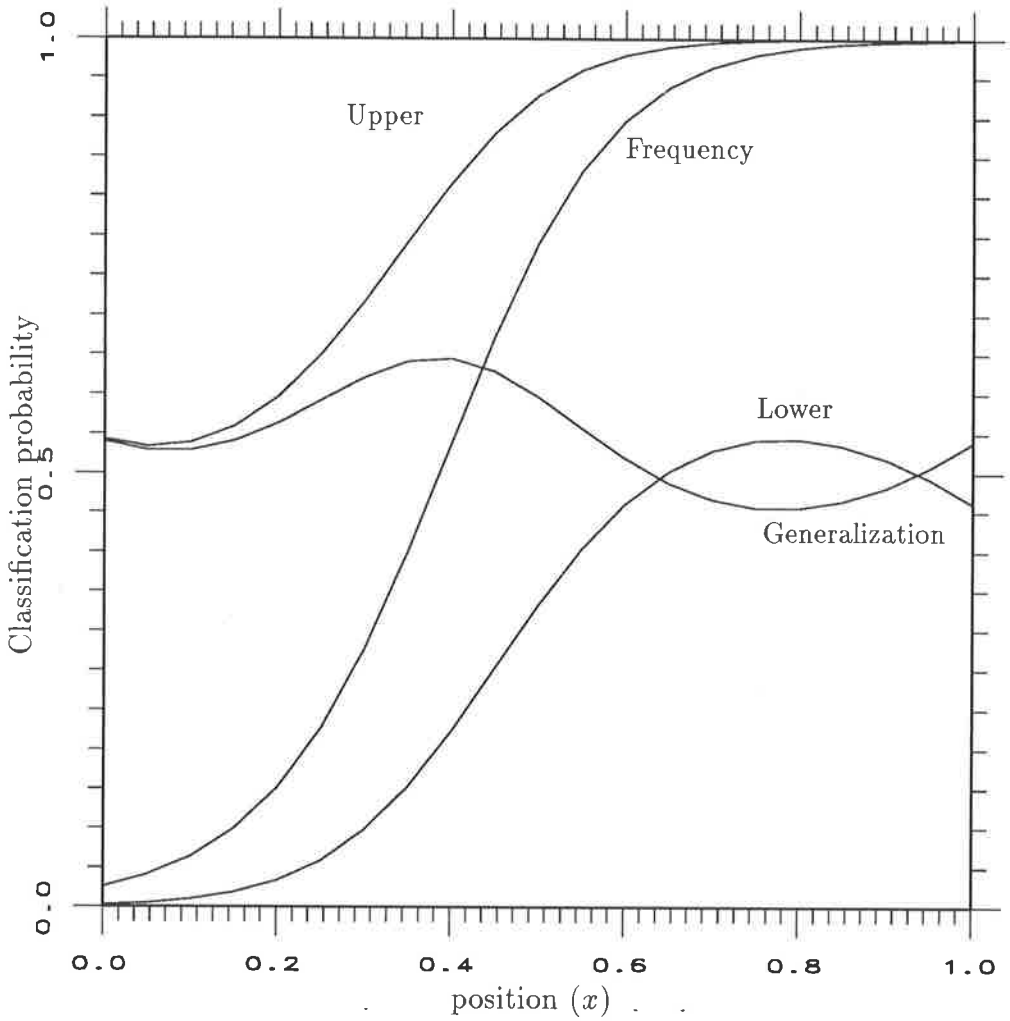

Figure 3.7: Classification 0 for a class boundary at $x = 0.5$ with 5 training points distributed uniformly over x. The curves are the upper, implied, lower and generalization error probabilities.

with a large, possibly incorrect, generalisation. This is particularly important in distributed decision networks and makes this classification scheme useful in such a situation where another network which does not incorporate a measure of the classification doubt may produce unreasonable results.

3.7 References

Drakopoulos, E. & Lee, C.-C. (1992). Decision rules for distributed decision networks with uncertainties. *IEEE Transactions on Automatic Control*, *37*(1),

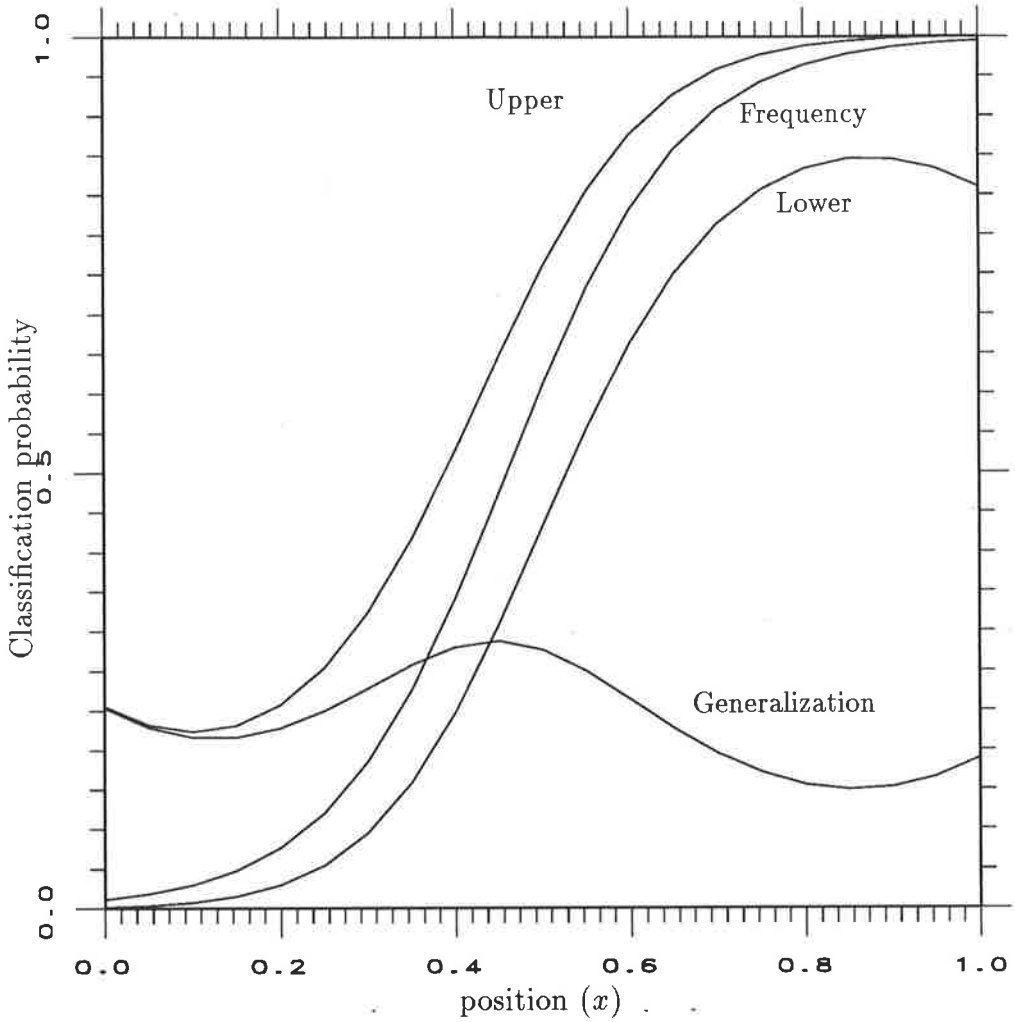

Figure 3.8: Classification 0 for a class boundary at $x = 0.5$ with 50 training points distributed equally about the class centers with a Gaussian density. The curves are the upper, implied, lower and generalization error probabilities.

5–14.

Hartman, E., Keeler, J. D., & Kowalski, J. M. (1990). Layered neural networks with gaussian hidden units as universal approximations. *Neural Computation*, *2*(2), 210–215.

Hunt, B. R., Qi, Y. Y., & DeKruger, D. (1992). A generalization method for back-propagation using fuzzy sets. *Proceedings of the Third Australian Conference on Neural Networks*, 12–16.

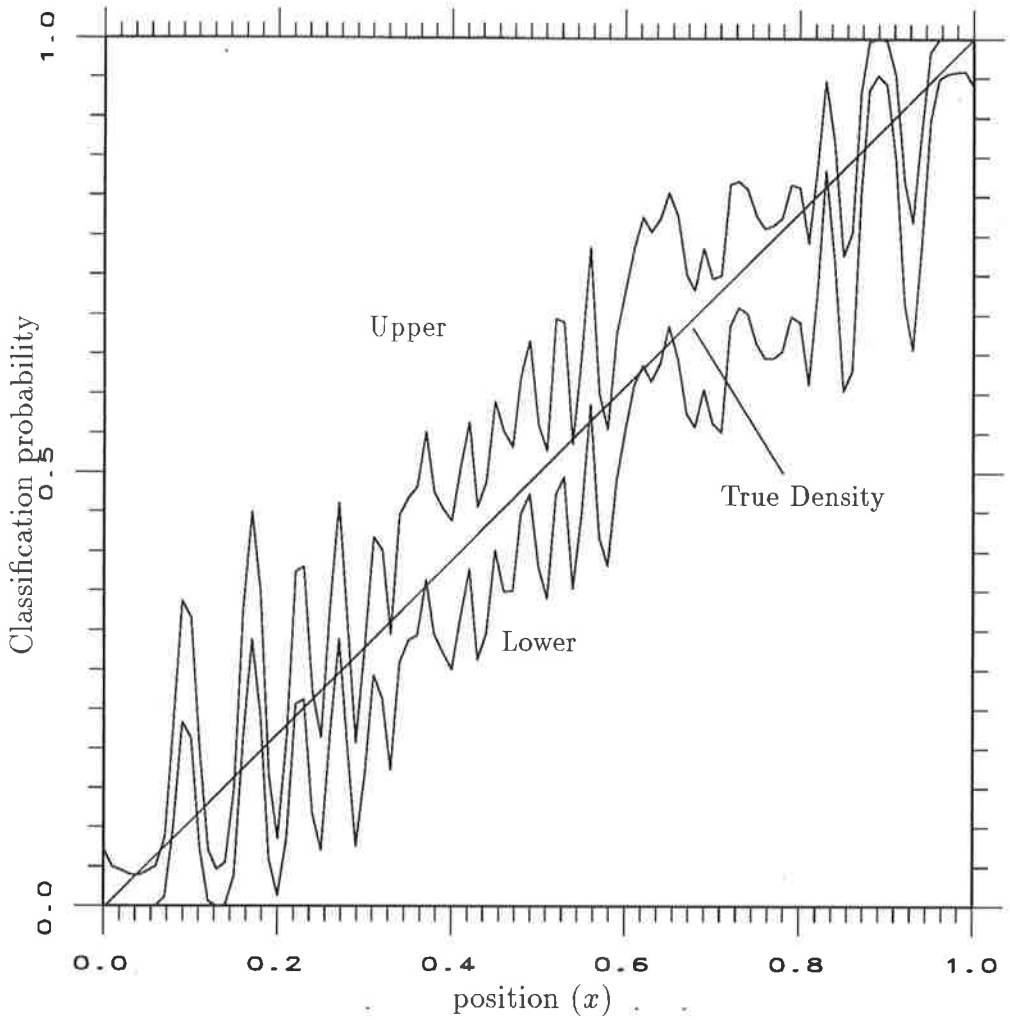

Figure 3.9: Classification 1 for the number line with 100 training points distributed uniformly over x. The curves show the upper and lower probabilities which bound the true class probability.

Joerding, W. H. & Meador, J. L. (1991). Encoding a priori information in feedforward networks. *Neural Networks*, *4*, 847–856.

Park, J. & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, *3*(2), 246–257.

Parzen, E. (1962). On estimation of a probability density and mode. *Annals of Mathematical Statistics*, *33*, 1065–1076.

Payne, T. (1992). Central fusion of remote disparate processed sensor information using reasoned feedback to enhance sensor processing. In Green, D.

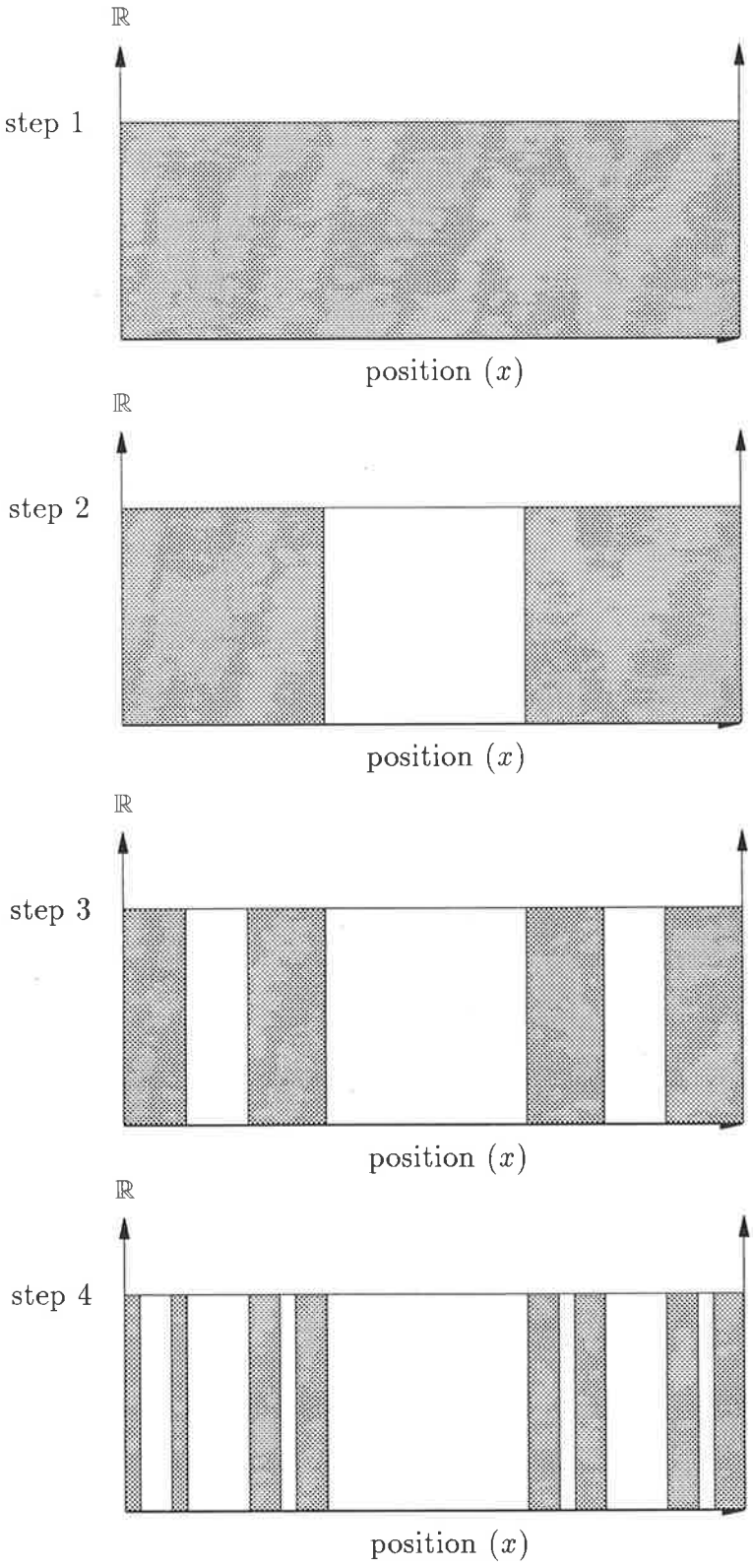

Figure 3.10: Stages in the development of a Cantor dust

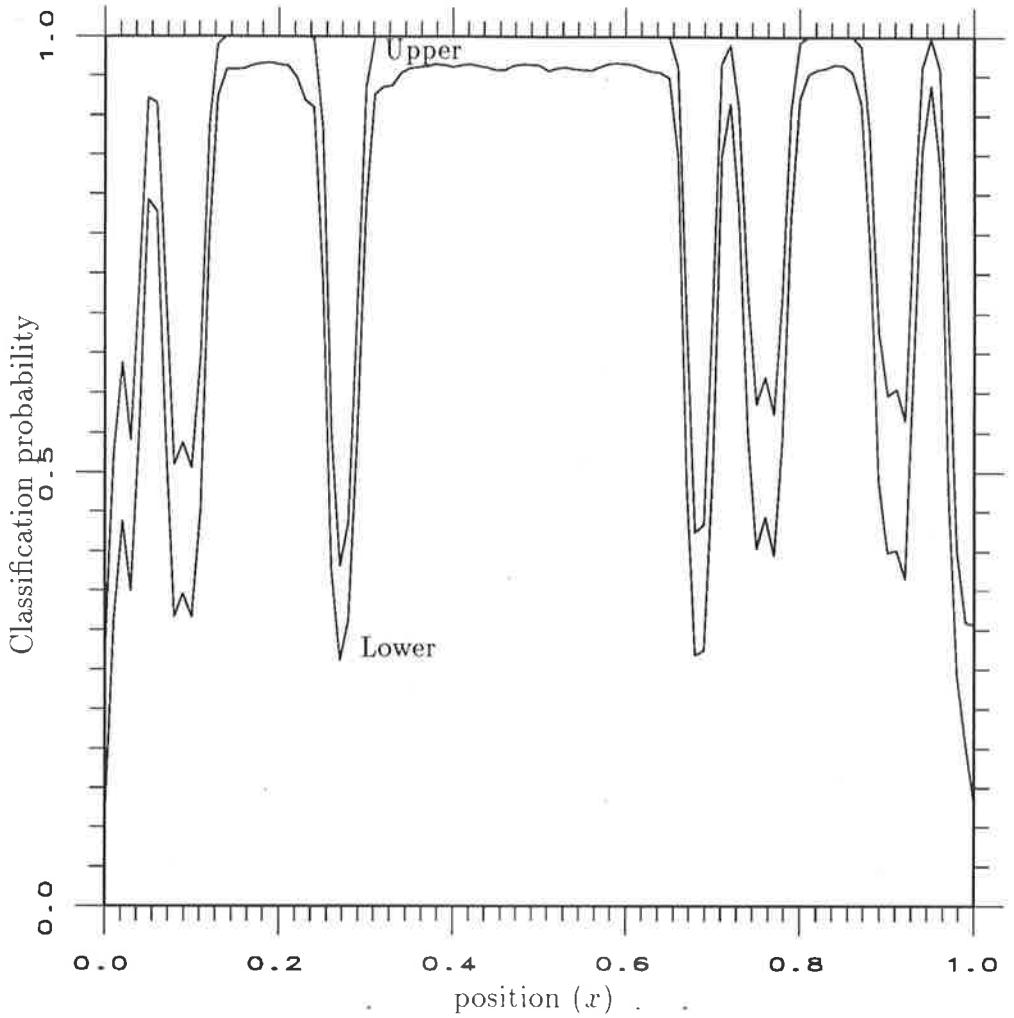

Figure 3.11: Classification 1 for the Cantor dust with 100 training points distributed uniformly over x. The curves show the upper and lower probabilities of class 1.

& Bossomaier, T. (Eds.), *Complex Systems: From Biology to Computation*, pages 248–259. IOS Press.

Payne, T. (1993). Signal classification for distributed decision networks with uncertainties and unmodeled class distributions. In *Proceedings of the International Conference on Acoustics, Speech, & Signal Processing.*

Reibman, A. R. & Nolte, L. W. (1987a). Design and performance of distributed detection networks. *IEEE Transactions of Aerospace and Electronic Systems, AES-23*(6), 789–797.

Reibman, A. R. & Nolte, L. W. (1987b). Optimal detection and performance of distributed sensor systems. *IEEE Transactions of Aerospace and Electronic Systems*, *AES-23*(1), 24–30.

Specht, D. F. (1967). Generation of polynomial discriminant functions for pattern recognition. *IEEE Transactions of Electronic Computers*, *EC-16*, 308–319.

Specht, D. F. (1990). Probabilistic neural networks. *IEEE Transactions on Neural Networks*, *3*, 109–118.

Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, *2*(6), 568–576.

Tessem, B. (1992). Interval probability propagation. *International Journal of Approximate Reasoning*, *7*(3), 95–120.

Zhuang, X., Zhao, Y., & Huang, T. S. (1991). A neural net algorithm for multidimensional maximum entropy spectrum estimation. *Neural Networks*, *4*, 619–626.

Chapter 4

Optical Flow Determination

4.1 Introduction

A parallel algorithm is presented for the extraction of the optical flow in an image which contains a noisy background.[1] An optical flow represents the motion of pixels in a scene so that every pixel in an image has an associated velocity in the image plane. The approach generalizes to higher dimensional spaces so that it can also be used to model the evolution of a noisy n dimensional signal. It is assumed that valid signal points adhere to a restricted domain of temporal models, while noise points don't. By analyzing the flow[2] it is possible to evaluate hypotheses about target existence (Chapter 3). These hypotheses are fused with other sensors' hypotheses (Chapter 2) and the global result is fed back to the predictor unit (Section 4.5) to improve the determination of optical flow(Figure 2.3).

4.2 Background

A lot of work has been performed in extracting optical flow from images which have structure. Konrad & Dubois (1992) impose smoothness constraints on the motion field of regions which are bounded by motion discontinuities, so that

[1] This chapter consists primarily of information presented in Payne (1993b)

[2] Including information about the sensor scene would improve the hypothesis determination considerably; however, this type of information is variable with operating conditions and sensor type. Including information about the scene at the time of classification would have raised a doubt over the features being used to classify the object. One of the intentions of this work is to extract objects buried in noise, which means that they have no sensed features which are visible, thus the only feature of an object which can be used for classification when the object is buried in noise is the velocity characteristics which are derived from a sequence of images. Including only the velocity characteristics of the scene forces the fusion to be driven by the velocity characteristics only.

optimization for multiple objects in the scene can be facilitated — the assumption is made that motion boundaries will coincide with an intensity edge. This is quite reasonable since "in general, a 3-D scene giving rise to a motion discontinuity will also contribute to an intensity edge."(Konrad & Dubois, 1992). Various other approaches are used to handle the discontinuities at motion boundaries (Horn & Schunck, 1981; Nagel & Enkelmann, 1986; Hildreth, 1984), for instance Hildreth (1984) performs smoothing only along intensity contours. Unfortunately these approaches rely on global optimization of the motion field and are consequently useless when dealing with objects which are so distant that they appear as a point, particularly when large amplitude noise is also present. The present work uses a different method when finding the optical flow in images by ignoring the structure in the image, and looking for temporal consistency in the flow field. As a result it is possible to handle situations where the background is extremely noisy.

The approach is designed to be implemented in a massively parallel architecture, similar in structure to a neural network. This parallel architecture has many simple processors arranged in layers which communicate between each other to resolve the more complicated global functions.

One of the limitations of conventional trackers which has motivated this work in an attempt to avoid the problem, is the increase in computational load which occurs as the number of points required to be associated increases. The problem of data association arises in tracking problems when attempting to determine which points are related to other points at a different time. To solve the problem it is necessary to recognize and label all the points. (See Appendix B) Points within the signal classes discussed here cannot be identified, and so every point must be associated with every other point. Since no filtering or thresholding occurs, the large number of points prevents a realistic solution to the problem using a standard approach to data association and tracking such as Joint Probabilistic Data Association (JPDA)(Fortmann, Bar-Shalom & Scheffe, 1983; Bar-Shalom & Tse, 1975; Colgrove, Davis & Ayliffe, 1986). Tracking is performed by modeling the evolution of the entire sensor field rather than individual elements.

4.3 Definitions

For simplicity we will restrict the tracking (modeling) problem to 4 dimensions; two spatial dimensions, x, one temporal dimension, t and a signal level y at each spatial and temporal location. This 4 dimensional system corresponds to the output for a two dimensional sensor, such as produced by a television camera. The output from the sensor is a matrix, ϕ, of signal levels at discrete points in space and time given by

$$\phi(x,t) \in \mathbb{Y} \quad \text{where } x \in \mathbb{X}. \tag{4.1}$$

$\mathbb{X}$ represents the valid set of n matrix indices (the matrix has n elements.) $\mathbb{Y}$ is bounded and will be considered to be quantised to R levels, since even if the sensor generates a continuous output, noise limits the effective resolution.

The values in a matrix can be viewed as coordinates in a geometrical space (state space) with the number of dimensions of this space equal to the number of elements in the matrix. Each value in the matrix represents the position on an axis, so that a matrix is now represented by a point in this high dimensional space rather than as a matrix or image in a lower dimensional space which is much harder to analyse. The state space representation provides an easy mechanism for viewing the evolution of a matrix by studying the path of the point over time. As an example consider a vector v with 2 elements, v_1 and v_2.

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{4.2}$$

If the vector is modified over time and the position of the vector is plotted in a geometrical space such as a simple cartesian space then a trajectory of the vector's motion in the cartesian space is formed. As an example consider the situation where

$$v(t) = \begin{bmatrix} v_1(t-1) + v_2(t-1) + 1 \\ \frac{v_2(t-1)}{2} \end{bmatrix} \tag{4.3}$$

$$\text{and } v(0) = \begin{bmatrix} 0 \\ 5 \end{bmatrix}. \tag{4.4}$$

This situation is depicted in figure 4.1. Unfortunately as a visualisation tool this approach is only worthwhile for vectors with 3 components[3] It should be noted

[3]This is not quite true since often a subset of the vector is all that is of interest, or a slice can be taken through a higher dimensional space to get a cross section of what is happening. The fact remains that 3 dimensions is the maximum number of components that can be viewed in a single plot.

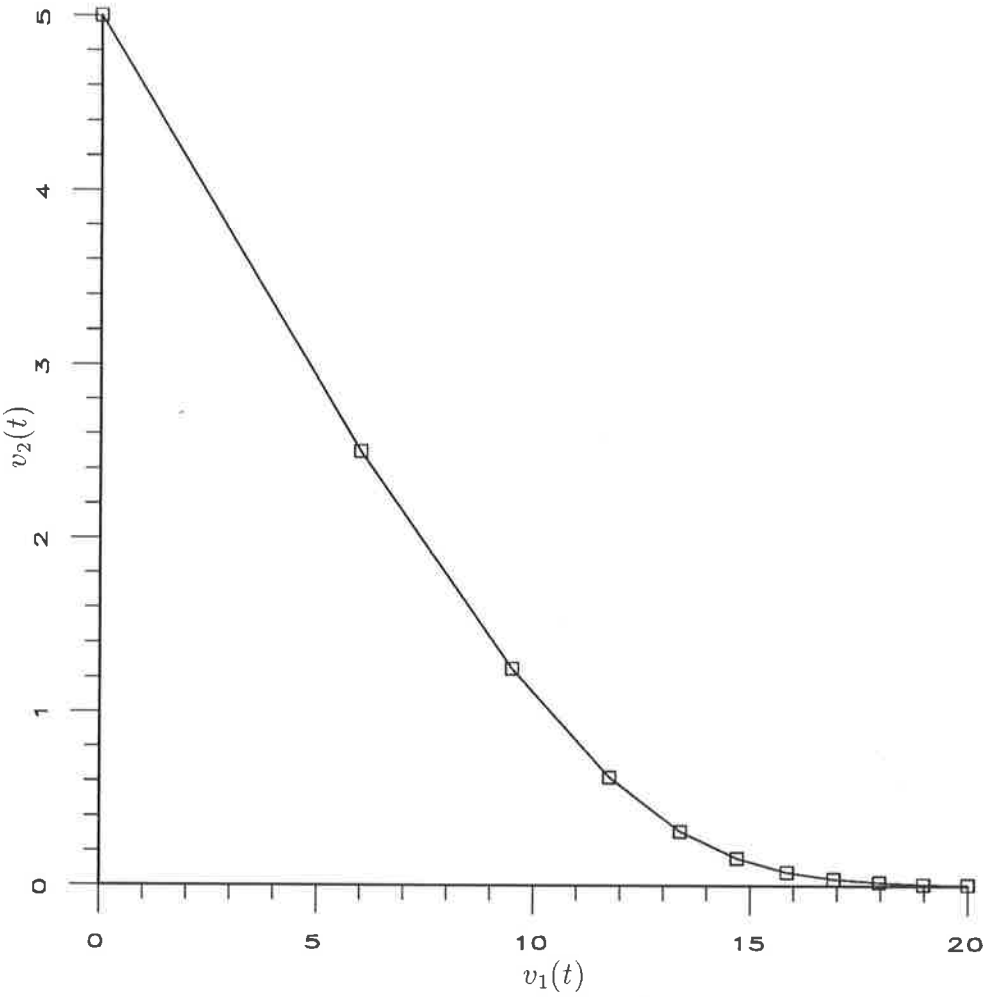

Figure 4.1: Trajectory of a vector's components through time t.

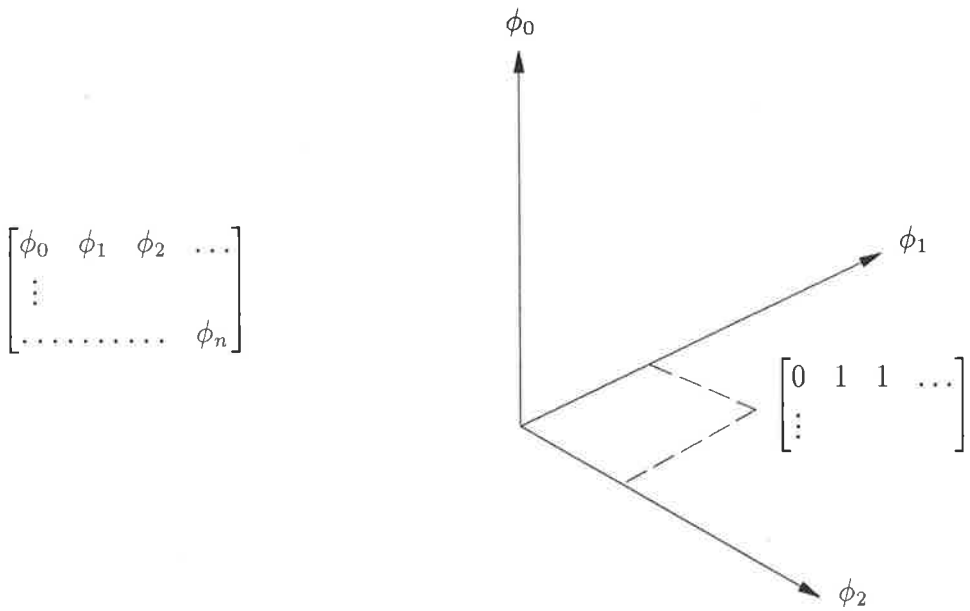

Figure 4.2: Image matrix with a subset of components shown displaying the manner in which components of the matrix are treated as coordinate axes in a geometrical space.

that the space need not be cartesian but that there is generally a transformation between valid spaces so that although the trajectory may appear different when plotted in a different geometrical space, the relationship between points in the trajectory is constant.

The state (or position in the geometrical space) of a matrix, ϕ, is represented in an n dimensional space where each pixel (represented by a location in the matrix) represents an orthogonal coordinate axis with the coordinate given by the pixel intensity (or value in the matrix). Figure 4.2 illustrates this point by displaying a subset of the coordinates of the matrix. In the figure only 3 of the n dimensions are shown.

4.4 Theory

If the evolution of the system represented by a point in the state space can be predicted consistently so that the position of the point corresponding to the sensor measurements is within a closed region about the predicted point, where the size of this error is small and a function of the sensor noise levels, then the underlying model of the signal has been discovered. This model contains

information about the flow of objects within the sensor's view.

The difficulty that arises in determining transformations between successive matrices arises out of the way simple motion in real space becomes complicated in the representation of the matrix state. For instance a simple motion of an object from one pixel to another is equivalent to a 90° rotation in the state space, and a successive move to the next pixel results in another rotation, but in a different hyperplane. (Figure 4.3)

With multiple objects and noise, the complexity of the transformations required to map a matrix to the successive matrix becomes overly complicated. The situation will be simplified slightly by exploiting typical sensor characteristics. Realistic objects will appear to be blurred to adjacent pixels in the sensor image by natural processes, such as scintillation or vibration of the sensor platform. Since the data acquisition occurs over a finite period of time in which the sensed scene is subjected to scintillation, vibration and scattering, the sensed scene is actually the convolution of the instantaneous noisy scene and the distribution of the scintillation, vibration and scattering. This convolution effectively blurs objects to adjacent pixels. See Figure 4.4 and 4.5. This causes the error in the estimated state matrix to be smooth with respect to the pixel displacement error. ie. the error in the estimated target position, in the embedding real space (number of pixels between the true location and the predicted location), is a smooth function of the error between the matrices representing the true (ϕ) and predicted states ($\hat{\phi}$) at the same time.

Example 6 *Without the blurring the error between a predicted state of the matrix,*

$$\hat{\phi} = \begin{bmatrix} 0 & 1 & 0 & \dots \\ \vdots & & & \\ \dots\dots & & & 0 \end{bmatrix}, \tag{4.5}$$

where all the values not shown are considered to be 0. *The true* ϕ,

$$\phi = \begin{bmatrix} 1 & 0 & 0 & \dots \\ \vdots & & & \\ \dots\dots & & & 0 \end{bmatrix}, \tag{4.6}$$

obtained by subtracting the estimated matrix from the true matrix results in an

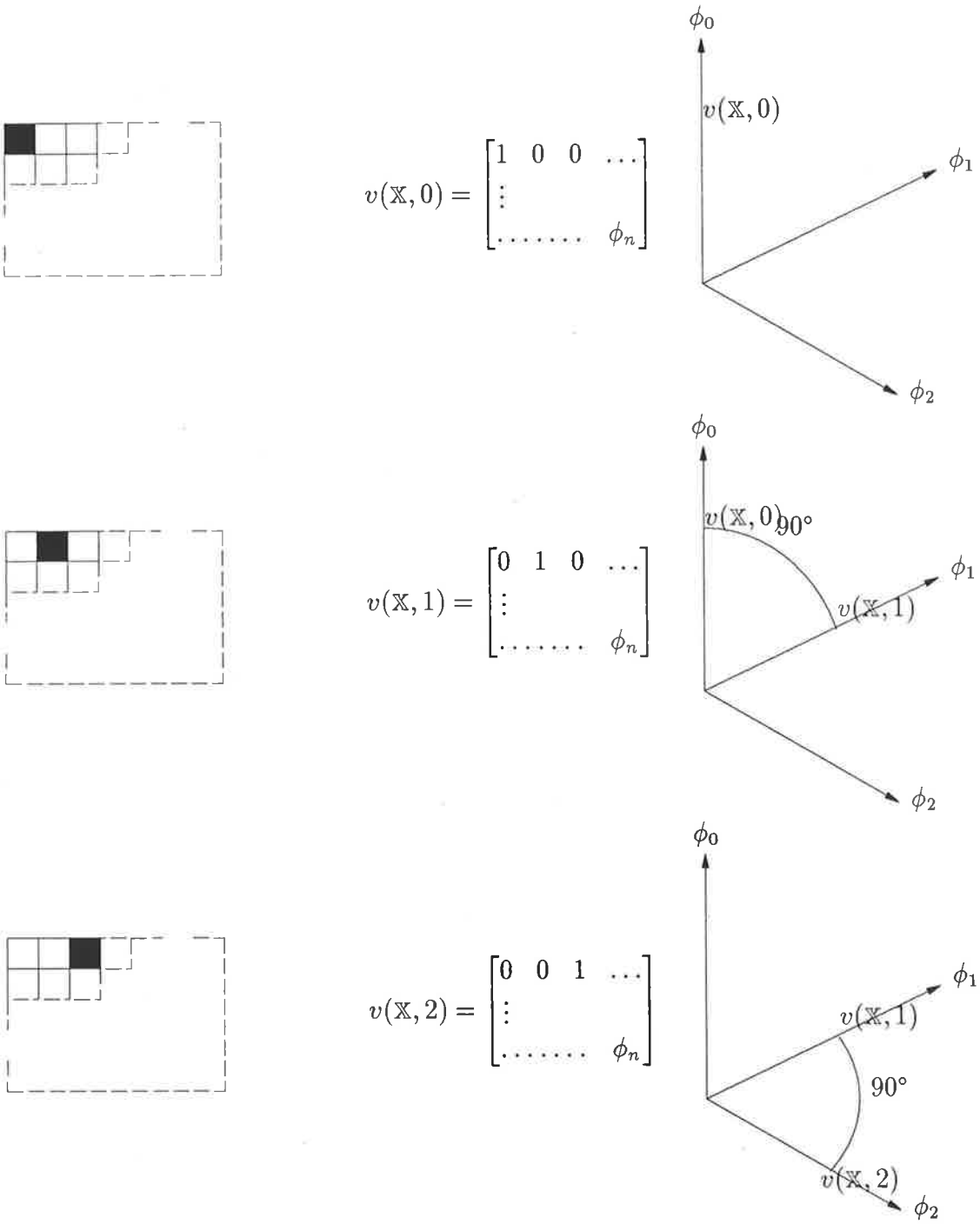

Figure 4.3: Simple motion in the image results in complicated motion in the matrix state space. In this example a constant motion between adjacent pixels results in a 90° rotation in the ϕ_{01} plane followed by a 90° rotation in the ϕ_{12} plane.

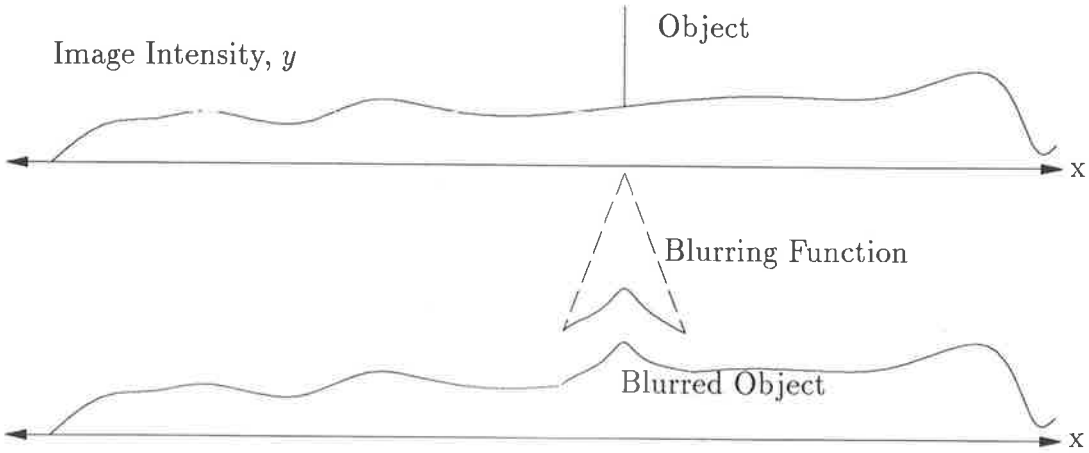

Figure 4.4: Effect of blurring on image smoothness.

error of

$$\phi^e = \hat{\phi} - \varphi \tag{4.7}$$

$$= \begin{bmatrix} 1 & -1 & 0 & \dots \\ \vdots & & & \\ \dots\dots\dots & & & 0 \end{bmatrix}. \tag{4.8}$$

However this has the same L2-norm as the error obtained if a different estimate of the matrix

$$\hat{\phi} = \begin{bmatrix} 0 & 0 & 1 & \dots \\ \vdots & & & \\ \dots\dots & & & 0 \end{bmatrix}, \tag{4.9}$$

had been used. Despite resulting in a matrix error that has the same magnitude as the error obtained using 4.5 , the predicted location of the object in the image (x domain) has an error, ϕ^e, twice that of the original estimate.

Example 7 *To illustrate the significance of the blurred input if the above errors are recalculated after blurring using an estimate of*

$$\hat{\phi} = \begin{bmatrix} 0.6 & 1 & 0.6 & \dots \\ \vdots & & & \\ \dots\dots\dots & & & 0 \end{bmatrix} \tag{4.10}$$

and the true ϕ,

$$\phi = \begin{bmatrix} 1 & 0.6 & 0.1 & \dots \\ \vdots & & & \\ \dots\dots\dots & & & 0 \end{bmatrix}, \tag{4.11}$$

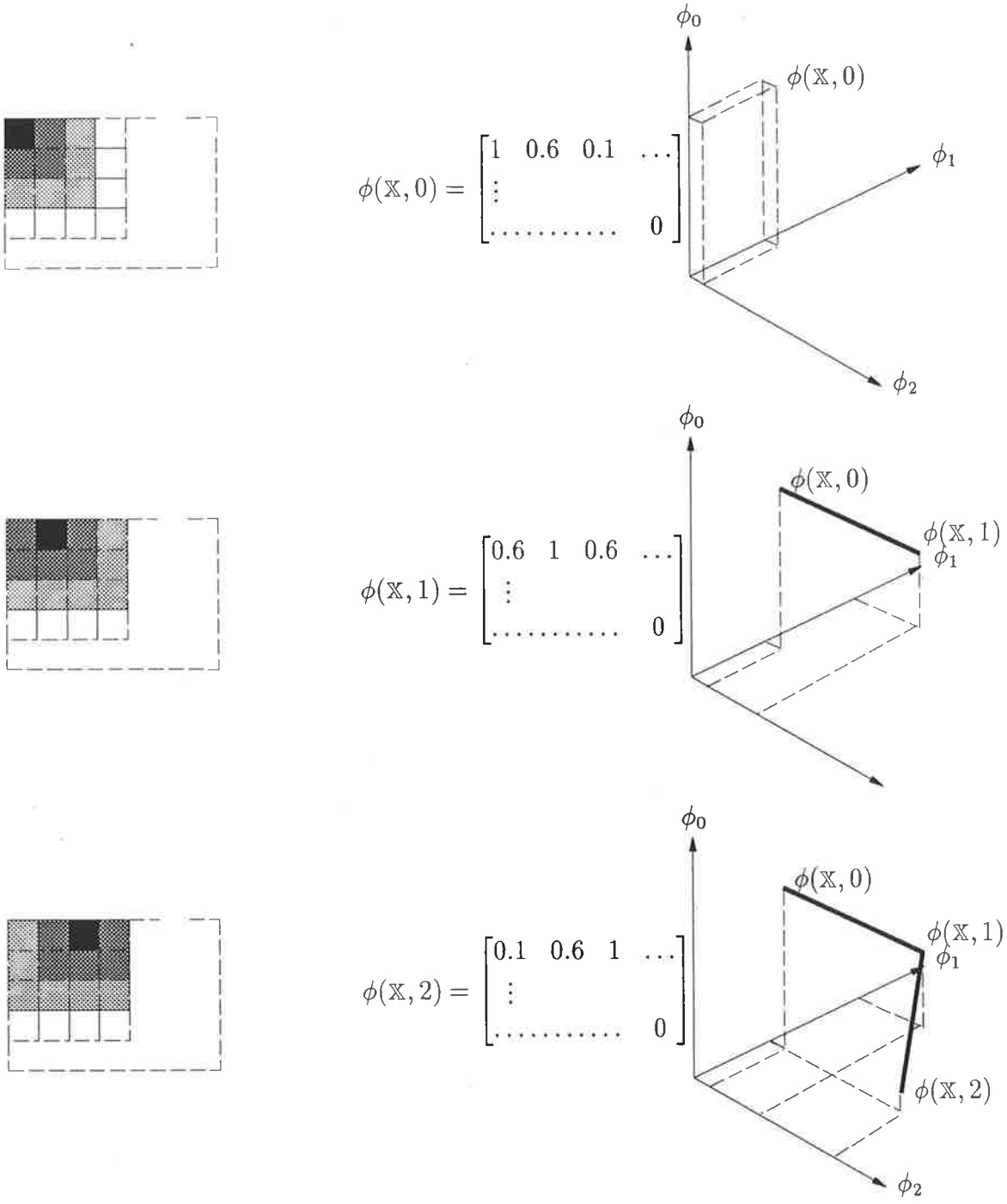

Figure 4.5: Simple motion in the image results in complicated motion in the matrix state space. In this example blurring has occurred between pixels so that the complicated motion in the matrix state space has been smoothed. Consider for example the ϕ_0 coordinate where successive values are now 1,0.6,0.1 while in figure 4.3 the sequence was 1,0,0 for the same image translations.

then the error is

$$\phi^e = \begin{bmatrix} 0.4 & -0.4 & -0.5 & \dots \\ \vdots & & & \\ \dots\dots\dots\dots & & & 0 \end{bmatrix} \tag{4.12}$$

which has a smaller error magnitude than is obtained for the estimate

$$\hat{\phi} = \begin{bmatrix} 0.1 & 0.6 & 1 & \dots \\ \vdots & & & \\ \dots\dots\dots & & & 0 \end{bmatrix} \tag{4.13}$$

which has an error of

$$\phi^e = \begin{bmatrix} 0.9 & 0 & -0.9 & \dots \\ \vdots & & & \\ \dots\dots\dots & & & 0 \end{bmatrix}. \tag{4.14}$$

Errors in predictions are therefore measurable in the ϕ space after blurring and allow the following analysis.

The evolution of the matrix is modeled by attempting to predict future matrices. This is done by finding a transformation operator, T_t, which adapts so that an estimate $\hat{\phi}(t+1)$, of the matrix $\phi(t+1)$ given by

$$\hat{\phi}(t+1) = T_t(\phi(t)), \tag{4.15}$$

is such that the magnitude of the matrix prediction error $\|\phi^e(t+1)\|$, given by

$$\|\phi^e(t+1)\| = \|\phi(t+1) - \hat{\phi}(t+1)\| < \epsilon,$$

is small. The transformation T_t is dependent on time and evolves so that

$$T_{t+1} = f\left(T_t, \dots, T_0, \overline{\phi}(t+1), \dots, \overline{\phi}(0)\right),$$

where

$$\overline{\phi(t)} = E\left\{\phi(t)|\hat{\phi}(t), \phi(t) + n_o\right\}.$$

n_o represents noise in the sensing process introduced either by the sensor or by the medium being sensed. The time dependence of the transformation has some important implications for implementation. The original intention was to use a form of feed forward neural network, such as the perceptron or neocognitron (see Rumelhart et al. (1986); Fukushima (1987); Nguyen & Widrow (1990); Touretzky & Pomerleau (1989) and Appendix C); however these networks are impractical in this application for a number of reasons depending upon the manner in which the network history is stored:

- A conventional back propagation network encodes the information about $T_t, \ldots, T_0$ and $\overline{\phi}(t), \ldots, \overline{\phi}(0)$ in the neuron connections, so that the history of the inputs and the required transformations make up the training history of the network; however the adaptation of the neuron connections is slow with respect to the speed at which the input changes, so that they are effectively time independent. If the operator, T_t is independent of time such that $T_t = T$ then it must also be independent of the spatial dimensions because it is possible for an object of interest to be moving in any direction at any point in space. This total invariance results in an inadequate mapping since the best transformation possible is one that assumes an object can move in any direction with equal probability and hence a blurred input is the best estimate of a future signal.

- It is possible to give the network all the prior information at its inputs rather than encoding it in the neuron connections, or even a certain amount of the signal history, i.e. $\overline{\phi_{t+1}}, \ldots, \overline{\phi_{t-5}}$ and this approach will work, with longer sequences of input history producing better estimates of the scene motion; however this is also impractical since it will make the already large network several orders of magnitude bigger.

If velocity vector information is available then the transformation has some temporal information and meaningful predictions can be made. Velocity information is not available from the sensors and must be deduced. Alternatives such as recurrent back propagation (Pearlmutter, 1990) are capable of determining the required velocity vector information and processing it as required. The Recurrent Back Propagation algorithm replaces the input history of the network with the vital information extracted from the input sequence, so that instead of a set of inputs $\overline{\phi_{t+1}}, \ldots, \overline{\phi_{t-5}}$, which require a large amount of memory, the dynamics of the objects within the input are presented which describe the evolution of the input sequence. In the case presented, the dynamic history of the objects in the scene is actually the transformation T_t so that the transformation update process can be imagined more easily when the transformation update rule is written in the form of (4.16). However, there is no guarantee that a recurrent neural network's training will ever converge for the mapping determination problem,

since there are many false minima in the neuron weight space.[4] For ease of implementation T_t is actually implemented as

$$T_{t+1} = f\left(T_t, \overline{\phi}(t+1)\right), \tag{4.16}$$

since T_t encodes information about $\overline{\phi}(t), \ldots, \overline{\phi}(0)$ and $T_{t-1}, \ldots, T_0$.

4.5 Transformation Determination

There are a large number of transformations, T_t, between a matrix $\phi(t)$ and $\phi(t+1)$. There are also a large number of transformations, T_{t+1}, between $\phi(t+1)$ and $\phi(t+2)$; however, only a limited subset of the transformations T_t and T_{t+1} are related by an acceptable set of rules. As more matrices are determined the set of transformations which are consistent across the entire sequence is reduced until there is only one valid sequence. At this point tracking has been established. Consider as an example of consistency the three sequential signals in Figure 4.6, numerous mappings exist between any pair of signals, although very few are consistent across all three. These consistent mappings are what is sought.

Since the knowledge of the signal is uncertain it makes sense to classify pixel values with a level of confidence. To avoid placing a statistical model on the signal levels in each pixel, a multinomial distribution is induced by augmenting the sensor space, so that for each pixel, every possible level has a value attached. This value represents the probability that the sensor would have seen that level in a noise free ideal world. The augmented state space has nR dimensions, since the sensor has n pixels each of which can be resolved into R levels.

Thus the representation of the matrix is modified from (4.1) to

$$\varphi(z,t) \in [0,1] \text{ where } z \in \mathbb{Z}. \tag{4.17}$$

$\mathbb{Z}$ represents the set of indices for the spatial and amplitude coordinates, $\mathbb{Z} = (\mathbb{X}, \mathbb{Y})$. This multinomial representation has a number of advantages because it is now possible to handle objects and backgrounds which have an unknown pixel probability distribution and structure. It also allows objects in the field of view

[4]The solution architecture is consistent with that of a recurrent neural network; however the structure of the network and the meaning of the "hidden" and "recurrent nodes" has been carefully established to incorporate all the prior knowledge regarding the problem. In effect, a network to solve the transformation determination problem has been created explicitly rather than by training.

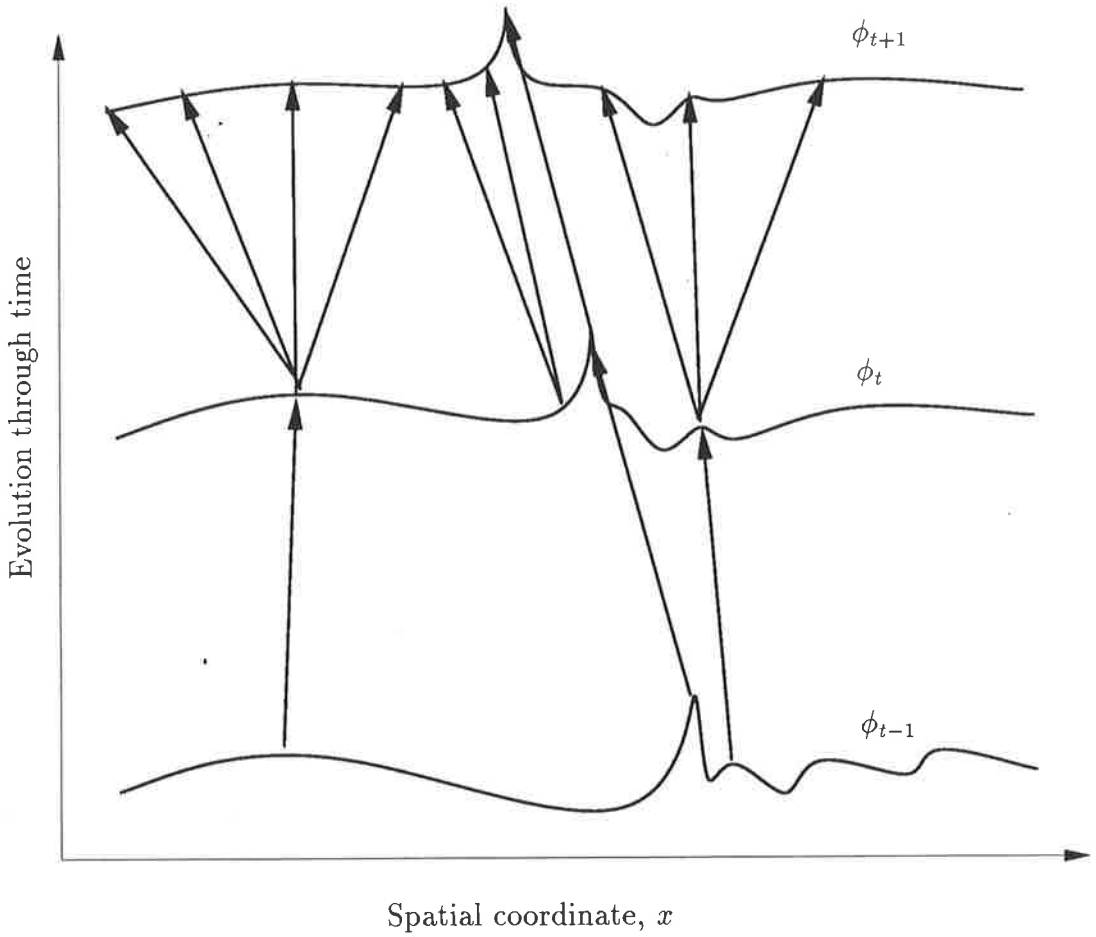

Figure 4.6: Acceptable relations between mappings.

to be obscured, since the tracking network will continue to update the object's track (providing the obscuring object is a different color or intensity, which is a reasonable assumption since their ranges are different), instead of forgetting about all non visible targets. The obscured object's confidence will be gradually depleted since uncertainty in the object's motion will increase, thus distributing the probability of the object's location over a successively wider region until the object can be assumed to no longer exist. Now since the vector, $\varphi(z)$, actually represents the returned signal, the sum of the probabilities of signal strength must sum to 1 for each spatial position. In the following, variables α, β, and γ are labels of points in either the state space or the augmented state space, while variables i, j and k refer to coordinates in the real world (see Section 0.2). For generality in the following it is assumed that the real world has coordinates which

are members of $\mathbb{D}$. This means that the following equations are equally valid in different systems that have a different number of dimensions. In the example pursued through out this document the number of dimensions, $|\mathbb{D}| = 3$; however other values are possible with other sensors. For a particular spatial coordinate, $x \in \mathbb{X}$,

$$\sum_{\alpha \in (x,\mathbb{Y})} \varphi_\alpha = 1. \tag{4.18}$$

The above expression is summed over the signal level components at a spatial point, x. If the sensor has a resolution of R levels then R terms must be summed. Over the entire input space

$$\sum_{\alpha \in \mathbb{Z}} \varphi_\alpha = n. \tag{4.19}$$

Input information to the network is actually embedded in noise, so that the input to the network is actually $\phi(t) + n_o$. This signal can be represented in the probabilistic representation of (4.17) by considering the sensor system characteristics, which will tend to blur the input, distributing the probabilities. The predicted matrix will also be affected by the noise on the inputs, and errors in the expected transformation. As a result of the smooth error surface the predicted state will be confined to a closed region of the surface about the true state.

Since each point in the space $\mathbb{Z}$ can move independently, there is a transformation operator T attached to each point. The nature of this operator is uncertain and so a family of operators are considered with a probability assigned to each member. This operator could be represented as in the Markov case; however, to reduce computational requirements the operator is modeled as a statistical family. The transformation consists of a mapping with parameters which are distributed in a Gaussian manner, and need only be represented by two vectors attached to each point in the augmented space, an expected transformation, T^μ, and a variance about that expectation, T^σ. This means that there is a matrix of $2nR$ vectors each with the same number of dimensions as the real embedding space and signal amplitude(in the example , $|\mathbb{D}| = 3$.) Future matrices are predicted by summing the probability transmitted to them from the previous field via a transformation T_t shown in (4.15). The operator T_t over the entire matrix can be expanded to show the effect of the mapping at a single point and how the

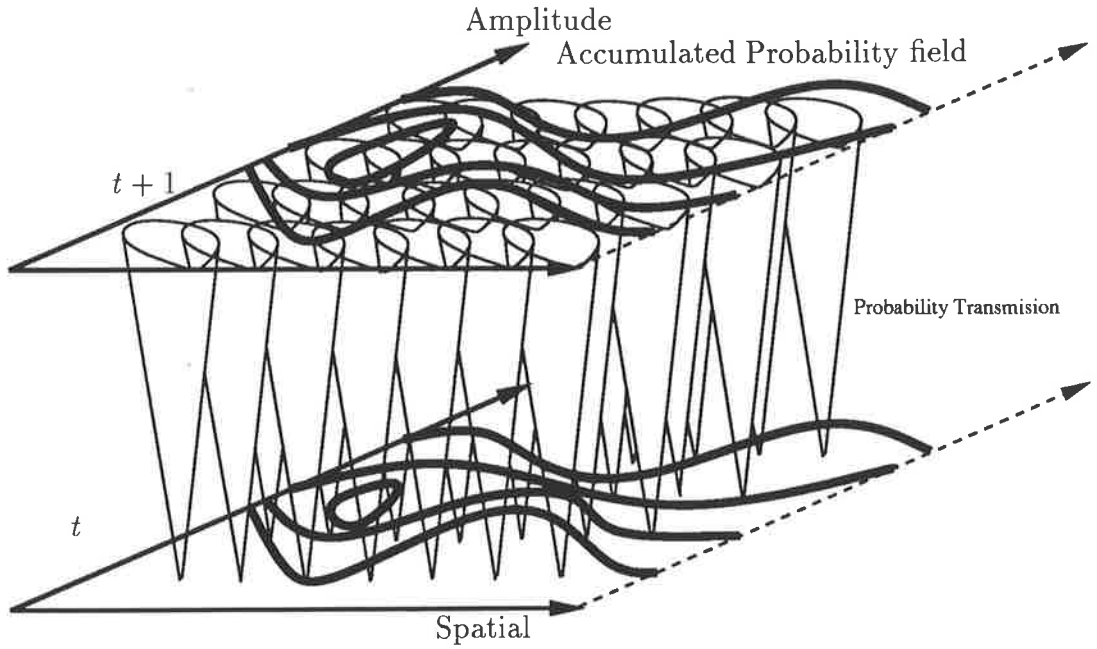

Figure 4.7: Probability transmission from the augmented space at time t to $t+1$. The probability at time t and $t+1$ is shown by contours, while the mapping functions are shown as cones. The probability at time t is distributed by the mapping functions to time $t+1$ where the probability is accumulated to form an estimate of the next input, the probability density of which is shown by the contours at $t+1$.

individual mapping parameters affect the result.

$$\hat{\varphi}_\beta(t+1) = \sum_{\alpha\in\mathbb{Z}} \overline{\varphi}_\alpha(t) T_t(\alpha,\beta) \tag{4.20}$$

$$= \sum_{\alpha\in\mathbb{Z}} \overline{\varphi}_\alpha(t) \left(\prod_{i\in\mathbb{D}} \frac{e^{-\left(\beta_i - T_i^\mu(\alpha)\right)^2 / 2\left(T_i^\sigma(\alpha)\right)^2}}{\sqrt{2\pi} T_i^\sigma(\alpha)} \right) \tag{4.21}$$

β_i represents the value of the ith coordinate in the real embedding space of the point referenced by β. (See Section 0.2)

(4.20) represents the probability transmitted to $x^\beta(t+1)$ from $x_\alpha(t)$ through a collection of transmission functions with Gaussian spatial extents. This is illustrated in figure 4.7.

If it is assumed that the transformation is very close to the correct transformation for the motion of the objects and background (this is shown in Chapter A), then the resulting estimates will have a smaller error than the noisy inputs. Since the expected value will have a smaller error it makes sense to determine the next

estimate from this estimate. Unfortunately this approach will propagate small errors in the transformation so that successive estimates possess a bigger and bigger error. If the transformation operator is not known precisely then the noisy input may be closer to the true input than the estimate. One approach to estimating the true input $\varphi(t+1)$, is to combine the estimated signal with the measured signal. This means that there is a trade off between the amount of improvement available by correctly estimating the transformation, and the sensitivity to changes in the viewed object's motion. The new matrix, $\hat{\varphi}(t+1)$ is combined with the latest noisy measured matrix to produce an estimate of the matrix, $\overline{\varphi}(t+1)$ without noise,

$$\overline{\varphi}(t+1) = E\left\{\varphi(t+1)|\hat{\varphi}(t+1), \varphi(t+1)+n_o\right\} \tag{4.22}$$

$$= (1-\epsilon)(\varphi(t+1)+n_o) + \epsilon\hat{\varphi}(t+1). \tag{4.23}$$

The estimate of the matrix $\overline{\varphi}(t+1)$ is determined by tracing a trajectory along the geodesic between the augmented equivalent point, $\phi(t+1)+n_o$ and $\hat{\varphi}(t+1)$ by a ratio $\epsilon \in [0,1]$. The value of this parameter will control the dynamics of the tracking process, such as convergence time, noise rejection and will also control the sensitivity to target manoeuvres.

The transformation is updated in a probabilistic manner so that the transformation represents the expected mapping given the current smoothed estimate, the previous smoothed estimate and the transformation between them, which contains information about estimated signals in the past, since it was formed from the expected mapping of earlier signals. The transformation update equations are given by

$$T^{\mu}_{t+1}(\alpha)_i = \sum_{\beta\in\mathbb{Z}} \frac{|2\alpha_i - \beta_i| p(\alpha,\beta)}{\sum_{\gamma\in\mathbb{Z}} p(\alpha,\gamma)} \text{ and} \tag{4.24}$$

$$T^{\sigma}_{t+1}(\alpha)_i = \sum_{\beta\in\mathbb{Z}} \left(\frac{|2\alpha_i - \beta_i|^2 p(\alpha,\beta)}{\sum_{\gamma\in\mathbb{Z}} p(\alpha,\gamma)} \right) - \left(T^{\mu}_{t+1}(\alpha)\right)^2 \text{ where} \tag{4.25}$$

$$p(\alpha,\beta) = \overline{\varphi}_{\beta}(t)\overline{\varphi}_{\alpha}(t+1) \prod_{i\in\mathbb{D}} \frac{1}{\sqrt{2\pi}T^{\sigma}_i(\beta)} e^{-\left(\alpha_i - T^{\mu}_i(\beta)\right)^2 / 2\left(T^{\sigma}_i(\beta)\right)^2} \tag{4.26}$$

(4.26) is the product of the probabilities at a point β in φ at the current time, t, the point α at the previous time frame, $t-1$ and of the probability that a transformation between the points α at time $t-1$ and β at time t exists. Since the probability of the mapping between α at $t-1$ and β at time t was derived

in an identical manner at the previous time frame, (4.26) actually represents an iterative determination of the product of all the points along a straight path, of which the mapping between α and β is a portion at the last time instant. Some weighting, which is caused by the normalisation of probabilities in (4.24) and (4.25), is applied which increases the significance of the more recent probabilities. For example if the probabilities of a points existence in the paths, which are consistent in Figure 4.6, are multiplied together in a weighted manner, along with other points which are also consistent with those paths from previous frames then a probability will be determined which represents the probability that the consistent path exists. (4.24) and (4.25) are simply the expected values of the transformation given that (4.26) represents the probability that a particular path exists.

(4.20) to (4.26) represent two coupled evolutionary systems that are shown to converge in Appendix A.

4.6 Information Fusion

To fuse information together from multiple sources it is necessary to represent the information from each sensor in a common format. In a situation where the sensors view different parts of the electromagnetic spectrum, and the objects of interest are featureless, direct comparison of the sensor data is meaningless. Comparisons must be made at a higher inference level, ie. target, background. An inference of this type requires a classification of object type, and so to keep with the methodology of this work and avoid the errors caused by incorrect generalisation, a decision is made about the degree of similarity between points and trained targets or background objects while still maintaining a representation of doubt. The similarity decision is made using the calculated flow field only, since the sensed image field, ϕ or φ, has no information regarding the nature of the point. Since the data is vague and the training incomplete, the result is an upper and lower probability of target existence. (Chapter 2) This representation allows combination of data from multiple sensors with both conflict and agreement. The resultant global conclusions from a range of sensors are fed back to modify (4.24) and (4.25) so that the flow field update is biased with the global evaluation of an objects dynamics (Chapter 5), as well as knowledge that the object is background

or target.

4.7 Summary

The material in this chapter presents a procedure for determining the optical flow of a sequence of scenes. Once the optical flow has been determined it contains information about the motion of objects within the scene. Proof of the procedure outlined in this chapter is contained in Appendix A.

4.8 References

Bar-Shalom, Y. & Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, *11*, 451–460.

Colgrove, S. B., Davis, A. W., & Ayliffe, J. K. (1986). Track initiation and nearest neighbours incorporated into probabilistic data association. *Electrical and Electronics Engineering, Australia*, *6*(3), 191–198.

Fortmann, T. E., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, *OE-8*(3), 173–184.

Fukushima, K. (1987). Neural network model for selective attention in isual pattern ecognition and associative recall. *Applied Optics*, *26*, 4985–4992.

Hildreth, E. (1984). Computations underlying the measurement of visual motion. *Artificial Intelligence*, *23*, 309–354.

Horn, B. P. & Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, *17*, 185–203.

Konrad, J. & Dubois, E. (1992). Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(9), 910–927.

Nagel, H. H. & Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*, 565–593.

Nguyen, D. H. & Widrow, B. (1990). Neural networks for self-learning control systems. *IEEE Control Systems Magazine, 10*, 18–23.

Payne, T. (1993a). Signal classification for distributed decision networks with uncertainties and unmodeled class distributions. In *Proceedings of the International Conference on Acoustics, Speech, & Signal Processing.*

Payne, T. (1993b). Target extraction and optical flow determination from fused sensor information. In *SPIE Conference on Signal Processing, Sensor Fusion and Target Recognition*, volume 1955, pages 183–194, Orlando Florida.

Pearlmutter, B. A. (1990). Dynamic recurrent neural networks. Technical Report CMU-CS-90-196, Scool of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press.

Touretzky, D. S. & Pomerleau, D. A. (1989). Whats hidden in the hidden layers. *Byte*, 227–233.

Chapter 5

Combined System Dynamics

Analysis of the network components in Chapters 3 and 4 has shown that each is stable or performs satisfactorily in isolation; however, when feedback from the global fused results is fed back to the local processing modules (Section 2.2.2) the nature of the overall network described in Chapter 2 changes. Of particular concern is the optical flow determination module in Chapter 4. It would also be nice to know what improvements are possible as a result of multi sensor fusion.

Under certain conditions outlined in Section A the optical flow determination converges to equations A.12 and A.17. This is the result for an individual sensor working without feedback. What happens when numerous sensors are fused together? This chapter deals with this question by making assumptions which enable the conclusions to be found more easily. The result of making these assumptions is that the final result represents the qualitative nature of the combined system performance.

5.1 Assumptions

Equations A.12 and A.17 represent the statistics of the pixel errors at equilibrium. The size of this pixel error will be a monotonic function of the optical flow error assuming that convergence has indeed occurred and is maintained despite object manoeuvres. The optical flow field generated by the approach in Chapter 4 is classified via the approach in Chapter 3 so that an upper and lower probability for each object type at each location is generated. The simplification that the classification probability (The probability that the true classification is believed) made by the classification module (Chapter 3) is a monotonic function of the velocity errors determined from the optic flow (Chapter 4) is true in a

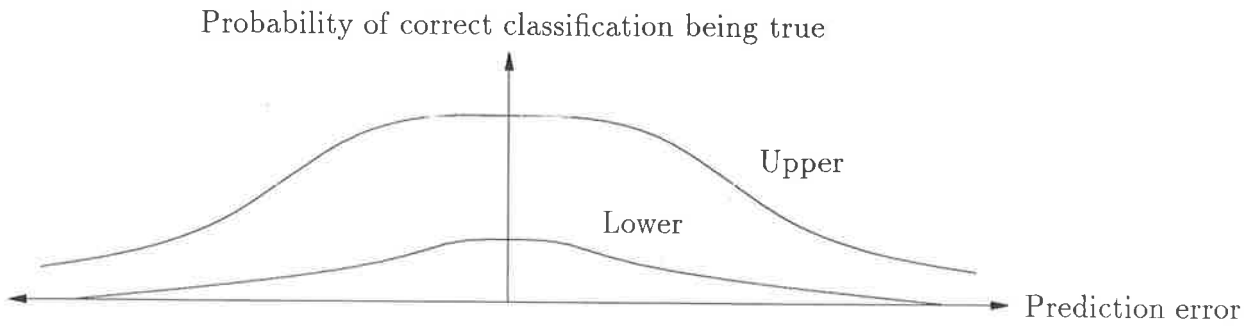

Figure 5.1: Assumed relationship between velocity prediction errors and the probability of the true classification

region about the true classification and fails only away from the classification where probabilities are small and have little effect on the combined classifications. This means that as the error in the optical flow increases, the probability associated with the correct classification is reduced monotonically (Figure 5.1). This assumption would appear to be reasonable since training instances about the classification boundaries are distributed in what is assumed to be a Gaussian distribution (see figure 7.7 to 7.11.) Classification probability refers to the lower probability; the upper probability is a function of the lower probability and the amount of effective training at each point (Section 3.3). The density of training points is certainly not uniform, but for the purpose of this analysis it is assumed that the upper probability also has a Gaussian like shape, with the same mean but possibly with a different variance, and a vertical offset, so that the upper probability of classification will have the same type of relationship with the velocity error as the lower probability.

5.2 Fusion performance

Using the fusion rules listed in Section 2.2.4 in particular (2.7) it can be seen that the global upper probability is a minimum of the local upper probabilities and the global lower probability is the maximum of the local lower probabilities. Since the local probabilities, L_l and U_l, are a function of the velocity prediction errors, the distribution of which is known, the global classification probability distributions, $p(L_g)$ and $p(U_g)$, can be determined by integrating the $p(L_l)$ or $p(U_l)$, where the maximum of the lower probability functions $L_{l_1}, \ldots, L_{l_n}$ or the minimum of the upper probability functions equals the global lower probability

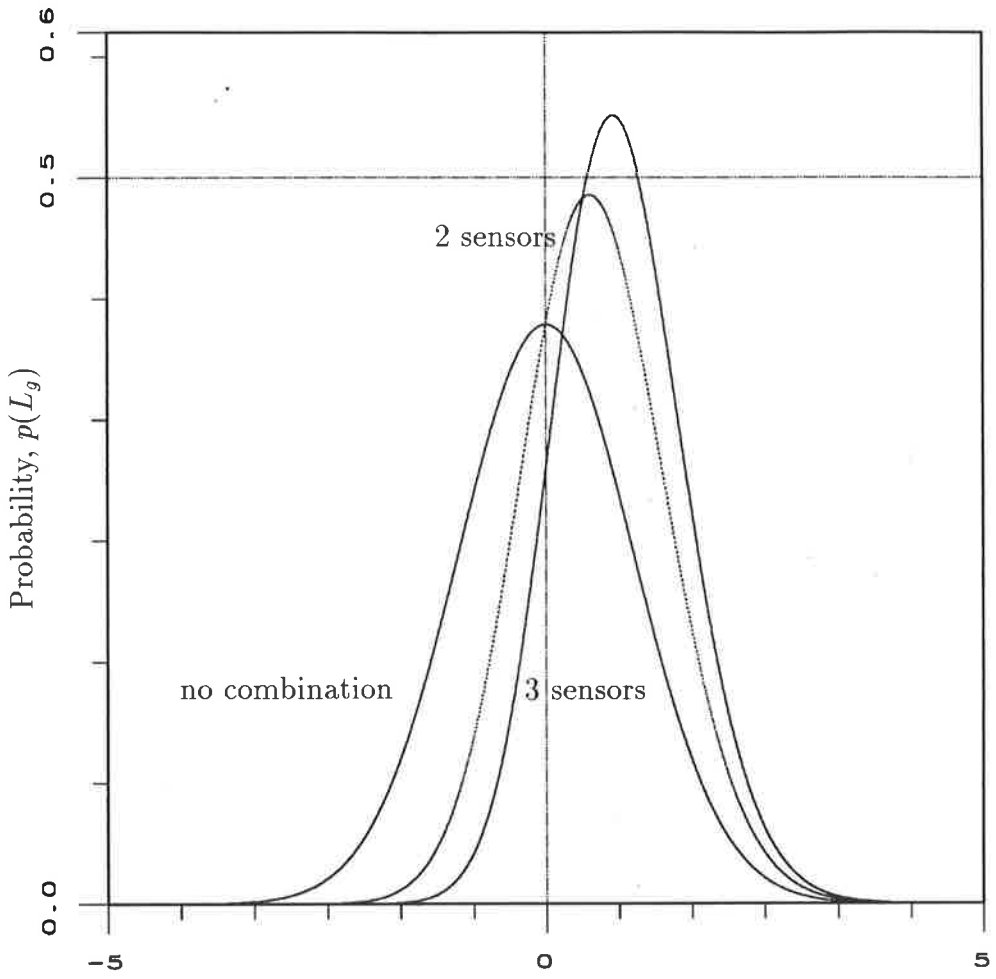

Figure 5.2: The distribution of the fused global lower probability, $p(L_g)$, when no combination occurs, two sensors are combined and three sensors are combined respectively. All sensors have an identical $p(L_l)$. The horizontal axis represents the value of the error in the lower probability, L_g, after fusing relative to the correct position for the information for an individual sensor, $\overline{L_l}$, in terms of the standard deviation of each sensor's distribution. As the number of sensors fused increases the center of the distribution of probability L_g shifts right and reduces in variance.

L_g or global upper probability U_g.

As an example take the situation where each sensor provides the same amount of information. Assuming that the relationship between the local classification probabilities and the velocity prediction error probability functions for all the sensors to be fused are identical and give a local classification probability for

each sensor which is a Gaussian, $p(L_l) = G(x)$, then the probability function of the global lower probability function, $p(L_g)$, as more and more sensors are combined can be calculated analytically and an example is shown in Figure 5.2. For n sensors

$$p(L_g) = \int \cdots \int_{max(L_{l_1},\ldots,L_{l_n})=L_g} G(L_{l_1})G(L_{l_2}) \ldots G(L_{l_n}) \mathrm{d}L_{l_n} \ldots \mathrm{d}L_{l_1} \quad (5.1)$$

where L_{l_i} refers to the lower probability attached to the classification from sensor i. The assumption that all sensors have the same classification distribution and that the distribution is gaussian is not necessary and the calculation can be performed with any combination of relationships. The relationships depicted in Figure 5.2 were chosen to show the effect of combining sensors which provide the same amount of information. For sensors which contain different amounts of information the most informed sensor will tend to dominate irrespective of the number of sensors combined.

The distribution of the upper classification probability is similar given all the assumptions, except that it is mirrored about the zero vertical axis, since the upper classification probabilities are formed from the minimum of all the sensor estimates rather than the maximum as was the case when generating Figure 5.2.

The result of these distributions is that as the number of sensors is increased the degree of doubt has been reduced since the bounded range where the true probability exists is now smaller. The question must now be raised as to whether this new smaller region of doubt does include the true value. The answer of course is that there is a small probability that this will not be the case, the actual bounds are only confidence limits, which in the simulations were set at 95%. By combining the information from several sensors additional information can be included and it would be expected that if the information from different sensors was independent (different parts of the electromagnetic spectrum and different view points) and the conclusions agreed, then there should be a reduction in the doubt.

5.3 Feedback

Along with the reduction of doubt in the classification there is a reduction in doubt in the predicted motion of the object. The global prediction of motion cor-

responds to a weighted average of the sensor decisions. The weights are derived from the lower probabilities associated with each of the motion estimates.

This global estimate of motion parameters is used in (2.13) to bias the estimate of motion locally towards the global values. This will consequently reduce the size of the prediction error, which will in turn reduce the size of the classification error and so on. The amount of improvement is limited by a number of factors.

- Even perfect estimates of the scene motion do not result in precise classifications, since doubt regarding the classification always exists unless the amount of training information available is unbounded.
- The error in prediction resulting from target manoeuvres can not be avoided.

Given sensors which provide similar classification accuracies (similar $p(L_l)$ distributions as in Figure 5.2), the improvements possible are small (unless the variance of $p(L_l)$ is large) and probably not worth the effort. The main advantages of the work become evident when the information and the $p(L_l)$ from the sensors are different which will generally occur when the sensors are remote from each other or view a significantly different portion of the electromagnetic spectrum. In this case the method provides a mechanism for selecting the most appropriate classification from the collection and biasing the sensor prediction units in an attempt to find the best overall result. The results on conflicting evidence can be seen in simulations in Chapter 7.

Chapter 6

Algorithm Implementation

This Chapter contains an example implementation of the algorithm presented in earlier chapters. The algorithm is identical to that used to generate the majority of simulations presented in Chapter 7 (The sonar simulations differ only in the geometry of x).

6.1 The Signal

The signal, ϕ, is a 2 dimensional matrix containing 32 rows and 32 columns. Each of the values in the matrix corresponds to the intensity of a pixel in a 32x32 image ($n = 32 \times 32 = 1024$ pixels). The image (matrix) is indexed by x and t, (4.1), where the range of values of x, $\mathbb{X}$, are the coordinate ranging from $(1,1)$ to $(32,32)$. For notational ease the components of the vector x will be represented by x_1 and x_2. The sensor which produces the image produces a discrete output with 32 possible levels at each pixel (i.e. $R = 32$), thus the range of the values in ϕ is $\mathbb{Y}$, which consists of the integers $1, \ldots, 32$. Unfortunately this image is noisy and to avoid making assumptions about the distribution of the noise an augmented signal also needs to be defined (Section 2.2.3). The augmented signal at each instant in time, φ, (4.17), is a three dimensional matrix which is $32 \times 32 \times 32$ since there are 32 different discrete levels possible at each pixel. Each of the values in φ represents the probability that the image will have a pixel at that location with that intensity.

An image formed by the sensor has an unknown noise distribution so the noisy image, $\phi + n_o$, is augmented to the matrix, $\varphi + n_o$, by assigning all the probability to the point in φ which corresponds to the intensity of the pixel in ϕ, e.g. if the pixel at location (4,7) in ϕ has an intensity of 21, then the value at

(4,7,21) in φ is 1 and all the other values at (4,7,y), where y is the intensity, are 0.

6.2 Initialisation

The initialisation of the transformation vectors, T^{μ} and T^{σ} will vary from situation to situation; however, in the situations simulated it is assumed that the platform is stationary and since the bulk of the scene will be background the expected motion between frames is 0. i.e. $T^{\mu} = 0$. Since this value of 0, although generally true, is not true at all locations in the scene, the variances, T^{σ}, are made large to minimise the bias introduced by the initialisation assumptions.

The smoothed estimate of the image, $\overline{\varphi}$, is the same as the noisy input, φ for the first pass, rather than (4.23) since $\hat{\varphi}$ does not exist and φ represents all the knowledge available.

6.3 Transformation

The expected value of φ at the next time frame, $\hat{\varphi}(t+1)$, is calculated using (4.20) and (4.21) which for the example above can be written as

$$\hat{\varphi}_{x_1',x_2',y'}(t+1) = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \overline{\varphi}_{x_1,x_2,y}(t) \frac{e^{-\left(x_1'-T_1^{\mu}(x_1,x_2,y)\right)^2/2\left(T_1^{\sigma}(x_1,x_2,y)\right)^2}}{\sqrt{2\pi}T_1^{\sigma}(x_1,x_2,y)} \times \frac{e^{-\left(x_2'-T_2^{\mu}(x_1,x_2,y)\right)^2/2\left(T_2^{\sigma}(x_1,x_2,y)\right)^2}}{\sqrt{2\pi}T_2^{\sigma}(x_1,x_2,y)} \times \frac{e^{-\left(y'-T_3^{\mu}(x_1,x_2,y)\right)^2/2\left(T_3^{\sigma}(x_1,x_2,y)\right)^2}}{\sqrt{2\pi}T_3^{\sigma}(x_1,x_2,y)}$$

The dash on (x_1', x_2', y') does not mean anything except that we are talking about a different point to (x_1, x_2, y). The smoothed estimate is calculated using (4.23) which for the example above can be rewritten as

$$\overline{\varphi}_{x_1,x_2,y}(t+1) = (1-\epsilon)(\varphi_{x_1,x_2,y}(t+1) + n_o) + \epsilon\hat{\varphi}_{x_1,x_2,y}(t+1). \qquad (6.1)$$

The transformation can now be updated using (4.24) to (4.26) which can be rewritten to

$$T^{\mu}_{t+1}(x'_1,x'_2,y')_1 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \frac{|2x'_1 - x_1|p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")}, \tag{6.2}$$

$$T^{\sigma}_{t+1}(x'_1,x'_2,y')_1 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \left(\frac{|2x'_1 - x_1|^2 p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")} \right) - \left(T^{\mu}_{t+1}(x'_1,x'_2,y')\right)^2, \tag{6.3}$$

$$T^{\mu}_{t+1}(x'_1,x'_2,y')_2 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \frac{|2x'_2 - x_2|p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")}, \tag{6.4}$$

$$T^{\sigma}_{t+1}(x'_1,x'_2,y')_2 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \left(\frac{|2x'_2 - x_2|^2 p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")} \right) - \left(T^{\mu}_{t+1}(x'_1,x'_2,y')\right)^2, \tag{6.5}$$

$$T^{\mu}_{t+1}(x'_1,x'_2,y')_3 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \frac{|2y' - y|p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")} \text{ and} \tag{6.6}$$

$$T^{\sigma}_{t+1}(x'_1,x'_2,y')_3 = \sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} \left(\frac{|y' - y|^2 p(x'_1,x'_2,y',x_1,x_2,y)}{\sum_{\substack{x_1"\in\{1,\ldots,32\}\\ x_2"\in\{1,\ldots,32\}\\ y"\in\{1,\ldots,32\}}} p(x'_1,x'_2,y',x_1",x_2",y")} \right) - \left(T^{\mu}_{t+1}(x'_1,x'_2,y')\right)^2 \tag{6.7}$$

where

$$\begin{aligned} p(x'_1,x'_2,y',x_1,x_2,y) &= \overline{\varphi}_{x_1,x_2,y}(t)\overline{\varphi}_{x'_1,x'_2,y'}(t+1) \\ &\times \frac{1}{\sqrt{2\pi}T^{\sigma}_1(x_1,x_2,y)} e^{-\left(x'_1 - T^{\mu}_1(x_1,x_2,y)\right)^2 / 2\left(T^{\sigma}_1(x_1,x_2,y)\right)^2} \\ &\times \frac{1}{\sqrt{2\pi}T^{\sigma}_2(x_1,x_2,y)} e^{-\left(x'_2 - T^{\mu}_2(x_1,x_2,y)\right)^2 / 2\left(T^{\sigma}_2(x_1,x_2,y)\right)^2} \\ &\times \frac{1}{\sqrt{2\pi}T^{\sigma}_3(x_1,x_2,y)} e^{-\left(y' - T^{\mu}_3(x_1,x_2,y)\right)^2 / 2\left(T^{\sigma}_3(x_1,x_2,y)\right)^2} \end{aligned} \tag{6.8}$$

6.4 Hypothesis Determination

For each point $\varphi_{x_1,x_2,y}$, two parameters are determined, a divergence, $div_{x_1,x_2,y}$, and a velocity contrast, $vc_{x_1,x_2,y}$

$$\begin{aligned}(div_{x_1,x_2,y})^2 = &\left(\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_1^{\mu}(x_1',x_2',y')\,(x_1'-x_1)\right)^2 \\ &+\left(\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_2^{\mu}(x_1',x_2',y')\,(x_2'-x_2)\right)^2 \\ &+\left(\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_3^{\mu}(x_1',x_2',y')\,(y'-y)\right)^2 \end{aligned} \tag{6.9}$$

$$\begin{aligned}(vc_{x_1,x_2,y})^2 = &\left(\frac{1}{9}\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_1^{\mu}(x_1',x_2',y') - 2T_1^{\mu}(x_1,x_2,y)\right)^2 \\ &+\left(\frac{1}{9}\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_2^{\mu}(x_1',x_2',y') - 2T_2^{\mu}(x_1,x_2,y)\right)^2 \\ &+\left(\frac{1}{9}\sum_{\substack{x_1'\in\{x_1-1,x_1,x_1+1\}\\ x_2'\in\{x_2-1,x_2,x_2+1\}\\ y'\in\{y-1,y,y+1\}}} T_3^{\mu}(x_1',x_2',y') - 2T_3^{\mu}(x_1,x_2,y)\right)^2 \end{aligned} \tag{6.10}$$

6.4.1 Training

A number of alternatives are possible for the training of the hypothesis determination module. The training of this network will only be covered briefly, since the training schemes are not novel, and the emphasis is placed on the structure

and operation of the network which is original. For completeness however, the training must be discussed, but for the sake of simplicity, and because some of the simulations were performed using the off-line approach, the off-line approach will only be presented.

The operations performed in Section 6.3 can occur without the rest of the network. These operations should be allowed to continue until the velocity estimates have stabilised. Once stable the network should be allowed to continue; however, in addition to the previous operations, *div* and *vc* parameters should be derived for a selection of points where the true classification is known. This will lead to a plot such as Figure 7.7, which shows the distribution of the classes in terms of their *div* and *vc* coordinates. Ideally each one of these points should be represented in the network, but to speed up simulations the clusters of points are synthesised by a function which models their density and numbers. In the simulations performed in this thesis, each class was represented by a gaussian. These gaussians were determined by examining the plots and calculating the center (at point t) and radius of each cluster. The i^{th} gaussian, t_i, is represented by a mean value, which is a vector consisting of two components, $t_{div}^{\mu_i}$ and $t_{vc}^{\mu_i}$, and a variance which also consists of two components, $t_{div}^{\sigma_i}$ and $t_{vc}^{\sigma_i}$

6.4.2 Probability Determination

The training information is now represented by N (Section 3.3) 2 dimensional gaussians. For each point p in φ, i.e. $p = (x_1, x_2, y)$, the *div* and *vc* parameters are determined using the expressions in Section 6.3. For each of the $i = 1, \ldots, N$ gaussians, (3.1)

$$\begin{aligned} n_i(p) = & \frac{1}{\sqrt{2\pi} t_{div}^{\sigma_i}} e^{-\left(div_p - t_{div}^{\mu_i}\right)^2 / 2\left(t_{div}^{\sigma_i}\right)^2} \\ & \times \frac{1}{\sqrt{2\pi} t_{vc}^{\sigma_i}} e^{-\left(vc_p - t_{vc}^{\mu_i}\right)^2 / 2\left(t_{vc}^{\sigma_i}\right)^2}, \end{aligned} \tag{6.11}$$

is determined. The value of $n_i(p)$, is simply the value of gaussian i at the point p. If all the n_i generated from gaussians representing data from the same class are summed, then $N_k(p)$ is produced, (3.3),

$$N_k(p) = \sum_{t_i \in C_k} n_i(p). \tag{6.12}$$

The symbol k is the class name and C_k is the set of all the gaussians used to model the class k. (3.2),

$$f_j(p) = \frac{N_j(p)}{\sum_{k\in\theta} N_k(p)}, \tag{6.13}$$

uses these $N_k(p)$ to form a nominal frequency, $f_j(p)$, for the different classes. The set of all possible hypothesis is represented by θ. The $N_k(p)$ are perturbed, (3.6), by adding an amount a_j to each hypothesis $N_j(p)$ to form $N'_j(p)$ and then checking for consistency using (3.4) and (3.5).

$$\chi^2 = \sum_{j\in\theta} N_j^x \sum_{i\in\theta} \frac{(f_i(x) - f'_i(x))^2}{f_i(x)}. \tag{6.14}$$

f'_i is formed in the same way as f_i except that the $N'_j(p)$ are used instead of the $N_j(p)$. If

$$\chi^2 < \chi^2_k(1 - F) \tag{6.15}$$

is satisfied then the perturbed $N'_j(p)$ are consistent with the original training information. $\chi^2_k(1 - F)$ is obtained from tables depending on the number of hypotheses and the degree of certainty required (see the comments in Section 3.3). The $N_k(p)$ are perturbed to find all the possible consistent $N_k(p)$, and from those consistent $N_k(p)$, the highest and lowest frequencies, (3.7) and (3.8), are found for each class hypothesis. In reality it is not necessary to evaluate every possible perturbation. In the simulations it was assumed that the training information given could not change and that perturbations could only be made by obtaining more information. Given these assumptions the determination of the upper probability for a class j is simple, since it is necessary to perturb the value of N_j only. The value of a_j is increased until the value of N'_j leads to values of f which no longer satisfy the χ^2 test. This one dimensional search is performed for every class so that every class has an upper probability. In the process of increasing N_j to N'_j the value of f'_j is also increased from f_j; however, even though the N_i for $i \neq j$ are not altered the f_i are minimised when the f_j are maximised. The lower probability is the lowest f'_i, obtained by maximising the other f'_js one at a time.

6.5 Fusion

The above procedures are performed for every sensor (Figure 2.3). These results are then fused. The fusion rules, that apply to the upper and lower probabilities

determined in Section 6.4.2, for each hypothesis, vary depending upon the nature of the decision. For the example, with two electro-optic sensors which view the scene from the same location but in a different portion of the spectrum, objects should have the same x_1 and x_2 position in φ, but very little can be said about the relationship of an objects y position from the two sensors. Consequently it only makes sense to fuse hypotheses regarding the existence of an object at a particular x_1 and x_2 position. The probabilities determined in Section 6.4.2 can be modified to hypotheses about existence at x_1 and x_2 positions rather than at x_1, x_2 and y positions by forming the union of hypotheses at the same x_1 and x_2 positions. Let A_1 be the hypothesis that there is a target at a position in φ at (4,7,1) and A_2 be the hypothesis that there is a target at the position (4,7,2) then the hypothesis that there is a target at a position in ϕ at (4,7) can be determined by finding the union of all the hypothesis in φ with the same x_1 and x_2 positions. i.e. $A_1 \cup A_2 \cup \ldots A_R$. If $L(A_1)$ is used to represent the the lower probability of hypothesis A_1, and $U(A_1)$ the upper probability then the union of the hypothesis A_1 to A_R can be represented by, (2.10),

$$L(A) = L\left(A_1 \cup A_2 \cup \cdots \cup A_R\right) = \max\left(L(A_1), L(A_2), \ldots, L(A_R)\right) \quad (6.16)$$

$$U(A) = U\left(A_1 \cup A_2 \cup \cdots \cup A_R\right) = \min\left(1, U(A_1) + U(A_2) + \cdots + U(A_R)\right) \quad (6.17)$$

with in each sensor. After this is performed in each sensor, the sensors will have compatible hypotheses, so it remains only to combine the information using (2.9), where hypothesis A represents the hypothesis that sensor 1 sees a target at (4,7), and hypothesis B represents the hypothesis that sensor 2 sees a target at the same position.

$$L(A \cap B) = \max(0, L(A) + L(B) - 1) \quad (6.18)$$

$$U(A \cap B) = \min(U(A), U(B)) \quad (6.19)$$

Global estimates of target and background velocities (x_1 and x_2 components only) need to be determined to allow the fused estimates to affect the local trackers. The background velocity is estimated from the weighted average of all velocities, weighted by the amount of support for that point being background. ie. for the

background velocity in the x_1 direction

$$v_{background_1} = \frac{\sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} T^{\mu}(x_1,x_2,y)_1 L(x_1,x_2,y \text{ is a background point})}{\sum_{\substack{x_1\in\{1,\ldots,32\}\\ x_2\in\{1,\ldots,32\}\\ y\in\{1,\ldots,32\}}} L(x_1,x_2,y \text{ is a background point})}. \tag{6.20}$$

The target velocity is simply the velocity of the point in φ from all sensors at the same x_1 and x_2 positions which has the highest lower probability that it is a target. The values of T^{μ} in each sensor are modified by (2.13). For the component in the x_1 direction this is expressed as

$$T^{\mu}(x_1,x_2,y)_1 = \frac{T^{\mu}(x_1,x_2,y)_1 + L(gt)U(lt)v_{target}(x_1,x_2) + L(gb)U(lb)v_{background_1}}{1 + L(gt)U(lt) + L(gb)U(lb)}, \tag{6.21}$$

where gt and gb are the global target and background hypothesis at the point (x_1, x_2). lt and lb are the local target and background hypothesis at the point (x_1, x_2, y). Global hypothesis refer to the hypotheses formed by combining the hypotheses from multiple sensors, while local hypotheses are the results of classification of the signal at a sensor.

This completes the operation of the fusion network. The processor will repeat the algorithm for each new image arriving at the sensor.

Chapter 7

Simulation of Tracking Network

7.1 The fake world

There are a number of reasons why only limited real sensor data has been used in this evaluation. The first is a political justification since the inclusion of some of the available real data would have imposed constraints on the publishability of the work, and the second is due to practical limitations in processing. The artificial world created for the simulation is simpler to process than the output expected from most modern sensors since the resolution is small, which keeps computation time low. The artificial data is also controllable, and so training information is easily obtainable. Obtaining training information is a big problem when processing real data as will be explained later. The artificial data although simpler than the real world can make the problem more difficult to solve because the lower resolution used for simulations causes less distinction between an object and the background; however it is for these critical areas, where objects are close to the noise floor, that the approach is designed, and where conventional approaches fail. Where possible, noise has been added to the artificial data to make the problem as realistic as possible.

The simulated world as seen through an imaging sensor is discretised to a spatial grid with a resolution of 32 pixels in both the x and y directions. The value at each pixel is discretised to 32 different possible levels. The world model is continuous, although boring, since the background is uniformly blank. There are four objects which move around on this world, some of them maintain a constant motion while others are continuously manoeuvring with some even changing their color. Onto this simulated world look two sensors, which have different characteristics. The background and the targets obtained from each sensor have

different return strengths. Two of the objects appear at a different place in each sensor to provide a discrepancy between the sensor scenes to test the exclusion of false objects by the fusion process. The other two objects appear at the same location in each sensor, although at different return strengths, and simulate real objects which should be detected.

7.2 Comparison with other approaches

Direct comparison with alternative approaches is difficult and unjustified for a number of reasons. The main alternative approaches to this technique rely on tracking through processes such as Kalman filtering and Data Association (Appendix B). These approaches rely on preprocessing, ad hoc decisions and parameter selections, all of which have been deliberately avoided or at least minimised in this work. (The only parameter required in this work is ϵ, see (4.23)). The conventional approaches are also serial in nature and consequently scale extremely poorly when objects are concealed within noise, whereas the approach presented in this work is designed to be implemented in a parallel architecture, and in a custom parallel architecture performance would be extremely quick.

Comparison of this work, which is in its early stages of development, with conventional trackers, which are a mature technology, is unfair, and there are many improvements which could be applied to this approach to improve performance. Improvements could be made by using a better model of target motion (instead of looking for all consistent motions, the model would be biased to look for certain types of motion). The approach presented requires almost no tuning for specific applications so can be used for different tracking problems with very little modification.

The final difficulty in comparing the approach in this work to conventional approaches is the manner in which the approach presents conclusions. One of the difficulties in using a representation of probability that incorporates doubt is that it is no longer possible to make a simple (although potentially faulty) decision based purely on a probability threshold since a range of probabilities now exist. Before a decision can be made, cost functions need to be known about the cost of making correct and incorrect decisions so that bounded ranges of probability that stradle the usual probability threshold can be processed meaningfully. This

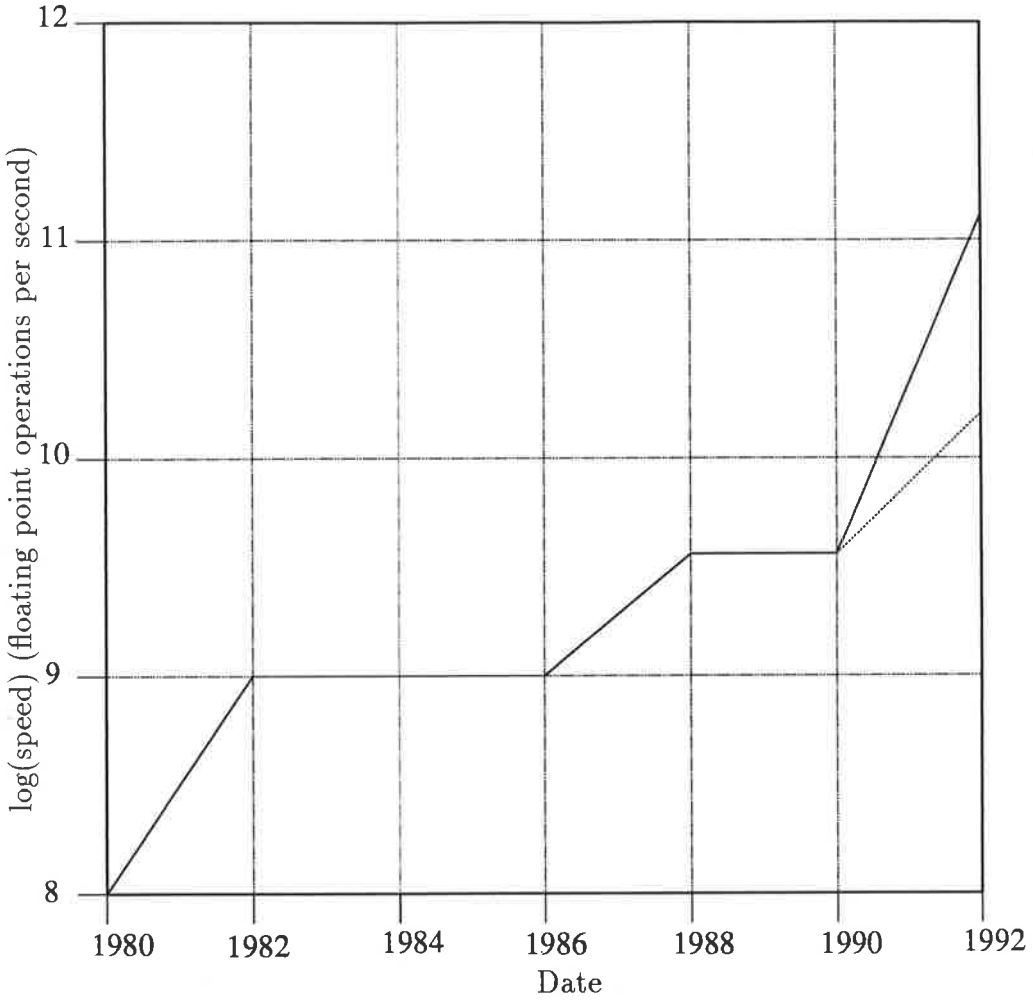

Figure 7.1: Speed of Super Computers(Catlett, 1992). For 1992, 2 figures are given 131 billion floating point operations per second for a 1024 node CM5 and 16 billion for a 16 processor Cray Y-MP C90.

type of problem is dealt with in Drakopoulos & Lee (1992).

7.3 Computational considerations

The procedure carried out in this simulation requires on the order of 20,000 operations per augmented pixel. In the simulations which follow a serial computer would require 6.5×10^8 computations per input picture.

Given the speed of modern super computers, (Refer to Figure 7.1.) with the current top of the line super computers performing 131 billion operations

per second, it would be possible to process 2015 frames per second; however modern sensors are easily reaching resolutions of 512x512 with 8 bit resolution giving 512 levels. This would result in a computational demand of approximately $2.7x10^{12}$ operations per sensor scan and with a frame rate of 20 frames per second a machine is required which is 400 times faster than existing machines. If you can assume (and it is a bad assumption since it ignores the possibility of new architectures) that the speed of computational increase is consistent then by extrapolating from the graph in Figure 7.1, machines capable of this power should exist within the next 15 years.

The assumptions upon which the feasibility has been done are wrong. The computational power of the super computers listed is for floating point operations, while the algorithmic precision requirements are satisfied using integers. The performance of these machines is generally better on integers than on floating point mathematics although the majority of development is dedicated to increasing the floating point speed, often involving special modules for floating point operations. This means that there is no correlation between the predicted increase in floating point computational speed and integer computational speed. Another point which should be noted is that in a multi sensor environment one of these super computers would be required for each sensor, and for the robustness of the entire operation all computation of local information should occur at the censor site, especially if the sensor network is destributed over a large area. For the algorithm to be feasible the computer must not only be able to perform the required calculations but it should be small and light with low power requirements.

There is an alternative computer architecture based on coherent light. The simple nature of the processes in the algorithm, make implementation in an optical computer possible.

7.4 Trained decision points

Since the space which is being classified has been augmented (Chapter 2), three classifications are possible: object, background, and nul. The meaning of the first two classifications is obvious, the nul hypothesis is used to train points which are part of the augmented space and do not represent any real features. The

network is trained to recognise the objects accurately, and the selection of all the training points is focused on this objective. It is assumed during the training process that the nature of the points is known, which is not a problem in the simulated environment, but would have been a major complication in a real task. After each time step in the training process, training points are selected at points where the objects exist. A similar number of training points are selected for the background and the nul hypothesis; however, there are many more of these points and so a selection process is implemented where points are chosen giving priority to those with similarity to object points. This has the effect in the decision space of concentrating the trained points around the boundary of the object points.

The classification of points has been carried out using only the velocity information from a small region about a point. To simplify and speed simulations the information from each point in the augmented space has been reduced to two parameters which represent roughly the divergence of the surrounding points and the velocity contrast of a point from the average of the surrounding points. The selection of these two parameters for use rather than many other possibilities is not rigorous, and there are sure to be many other systems; however the operation of the network appears to be robust to these variations. During development coding errors caused classification to be performed using other parameters, and although the decision space was altered the overall network response appeared to be consistent.

7.5 Tracking Results

Four simulations were run, on various types of data both real and artificial, on two different machines. Each simulates both the full system (Closed loop) and the system where the fused results are not fed back to the sensors (Open loop) so that each sensor effectively operates alone. The open and closed loop simulations are performed with identical object motions and noise statistics and come from a common initial state with an identical classification network. The initial state simulations were carried out in the open loop configuration to allow convergence to occur. The first two simulations were performed on a CM2 super computer and the last two were performed on a DEC5000 after characteristics of the training space were determined allowing the evaluation module of the

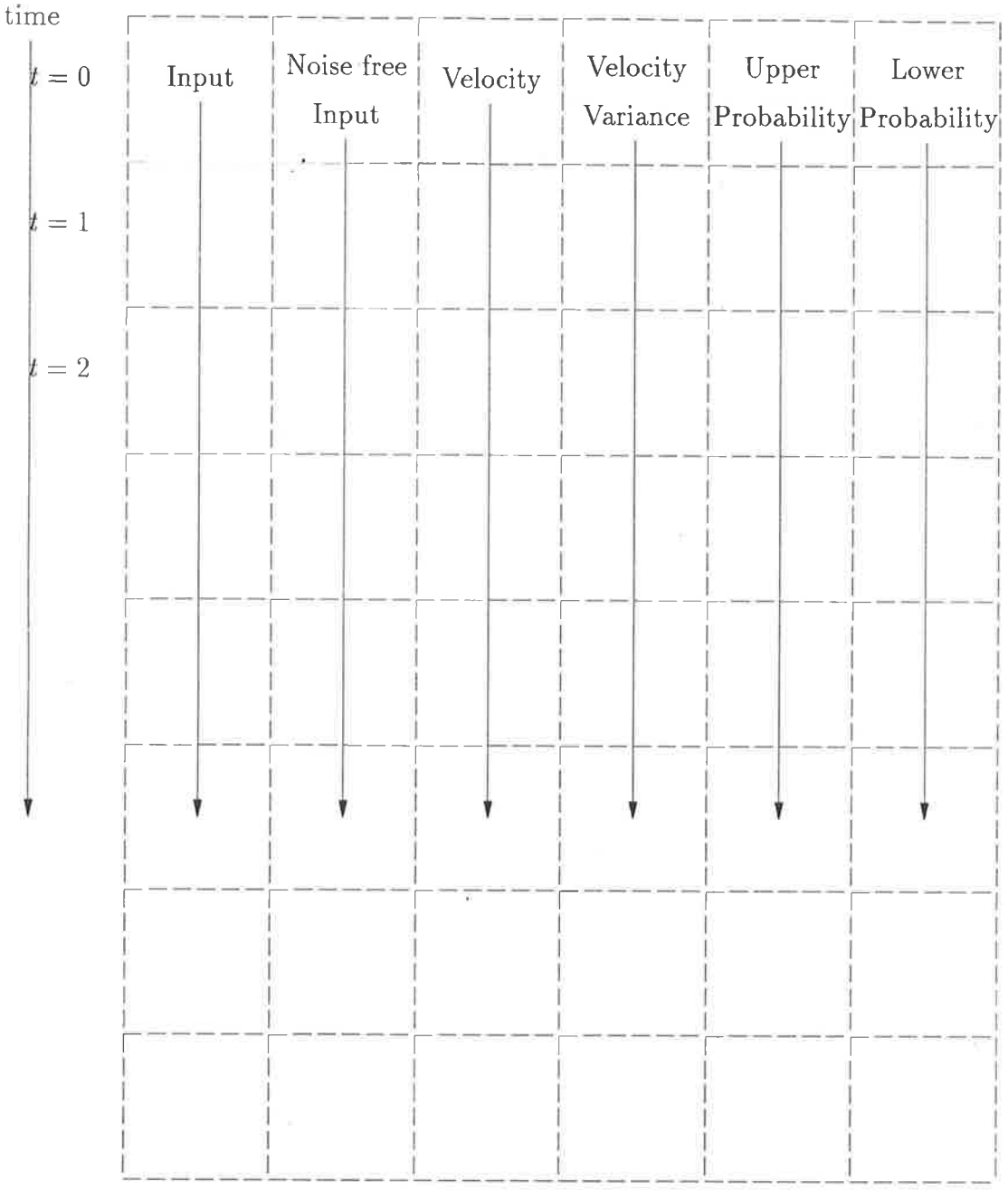

Figure 7.2: Information format in simulation figures 7.12, 7.13,7.14, 7.15, 7.24, 7.25,7.26 and 7.27

network (Chapter 3) to be simplified and the computational load reduced.

The simulation output which is represented by a series of figures show six columns (or strips) of information (refer Figure 7.2, 7.3, 7.4, 7.5 and 7.2) from the two sensors and the global level. These strips represent various information about the status of the trackers and their decisions. The information contained in each strip needs some further explanation since some of the information is only

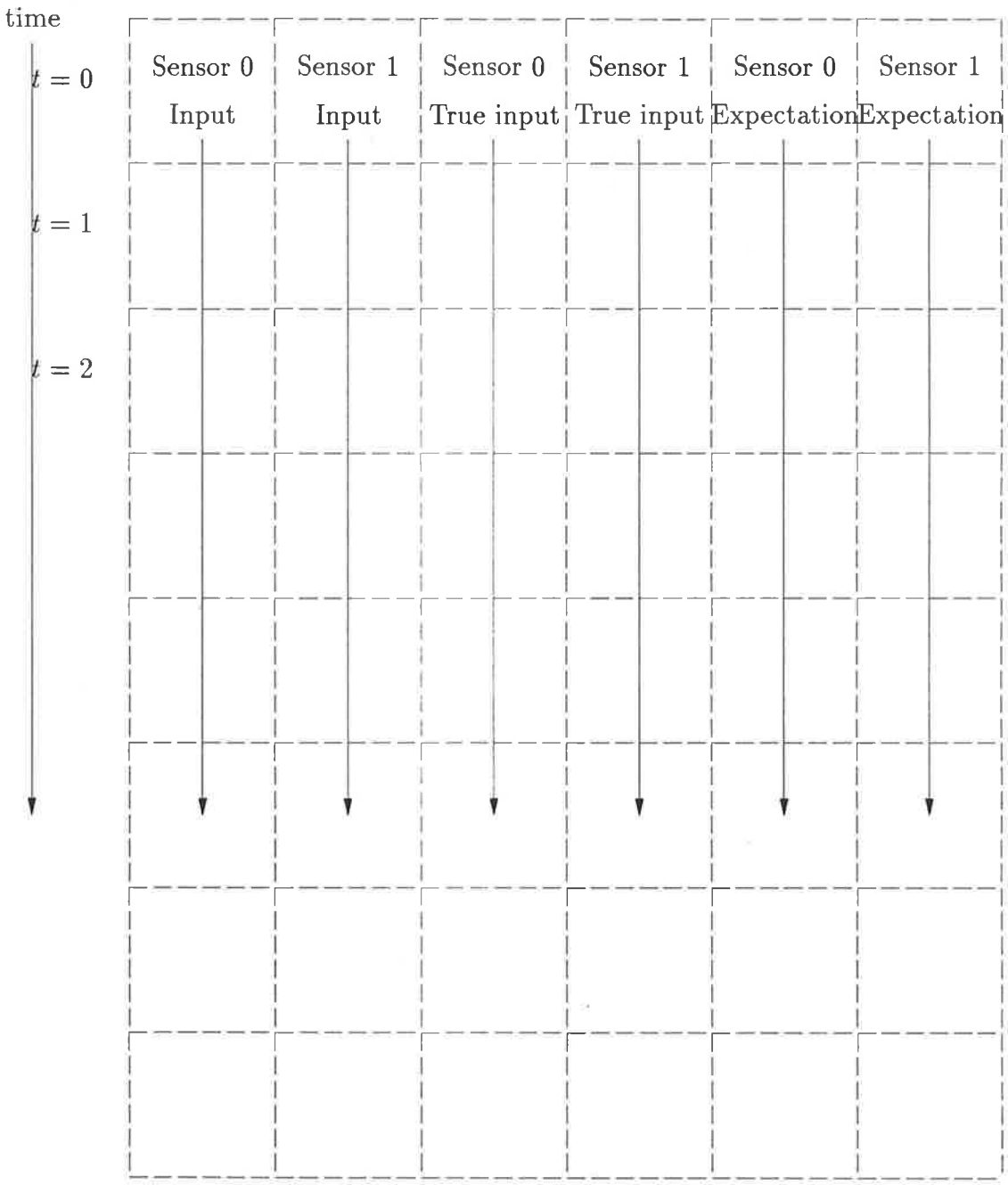

Figure 7.3: Information format in simulation figures 7.16, 7.29, 7.28 and 7.17

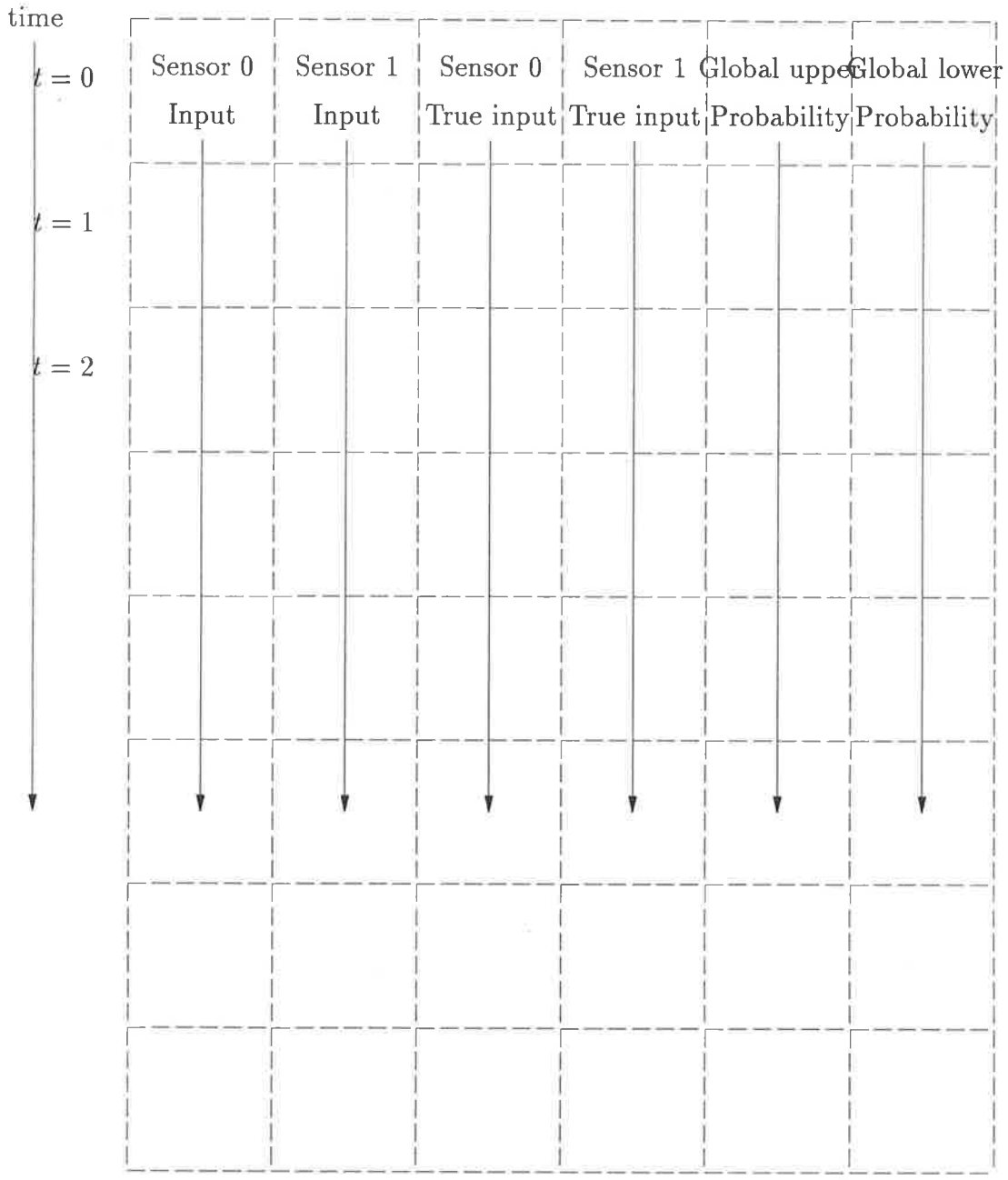

Figure 7.4: Information format in simulation figures 7.18, 7.19, 7.30 and 7.31

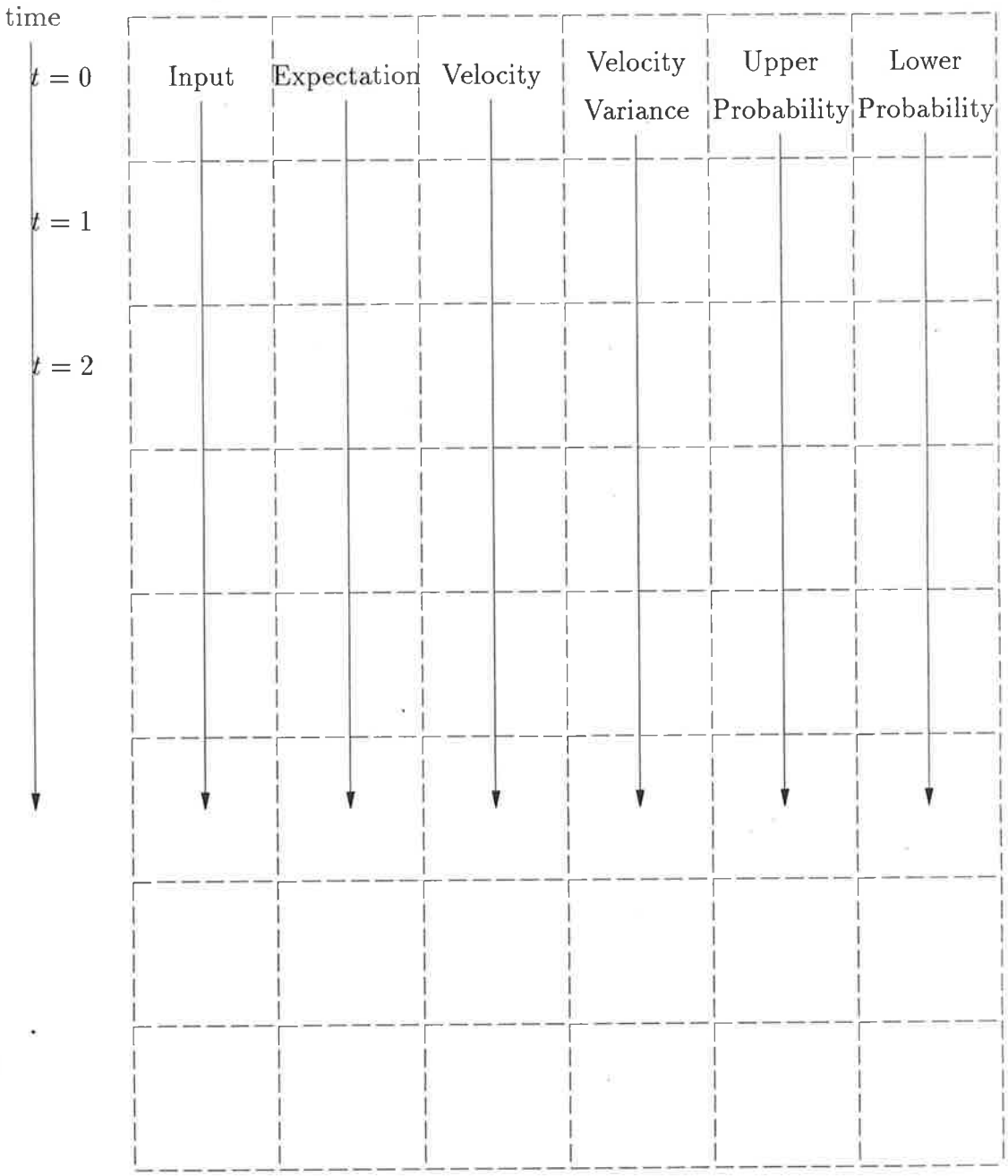

Figure 7.5: Information format in simulation figures 7.32 and 7.33

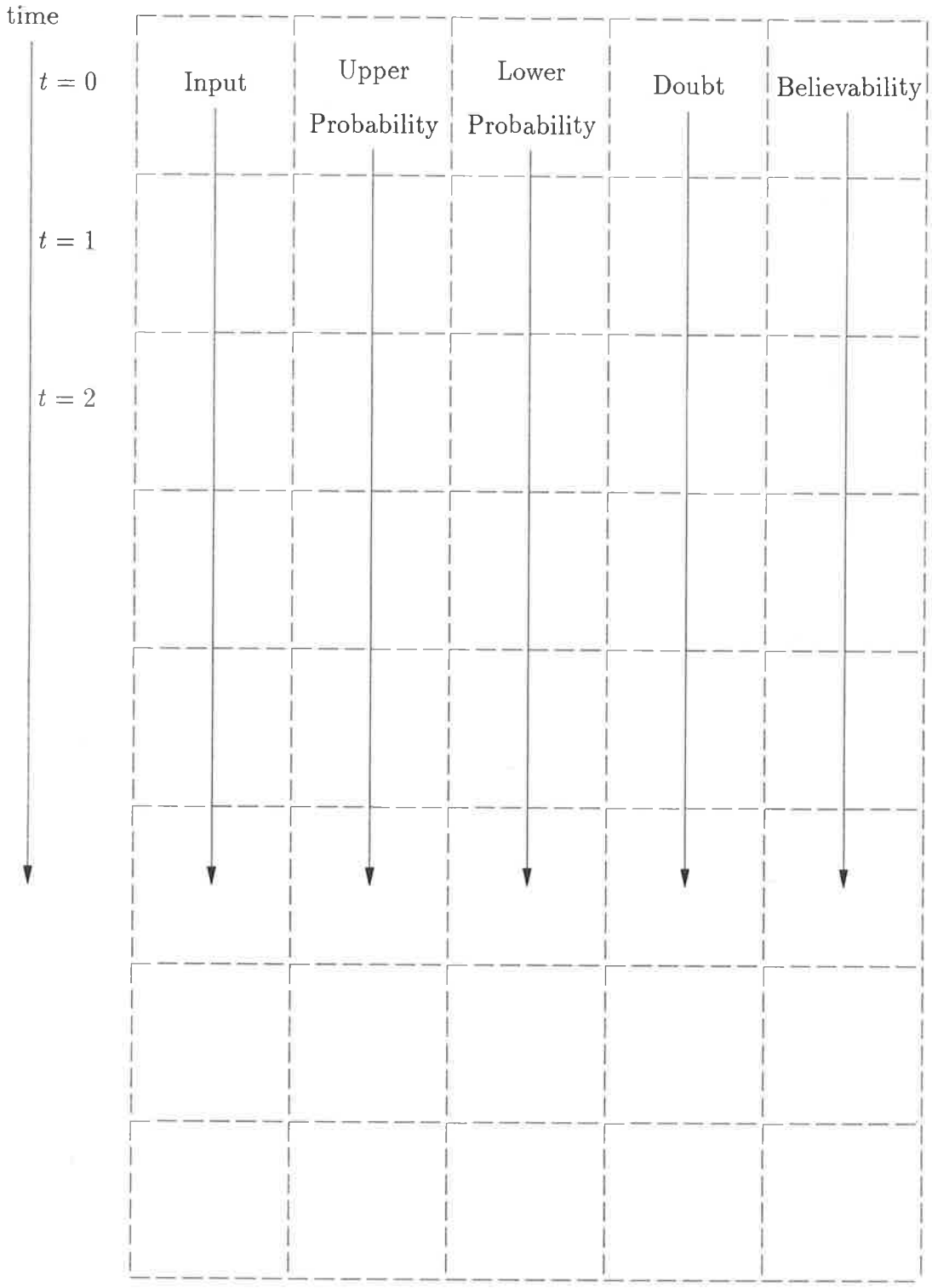

Figure 7.6: Information format in simulation figures 7.34 and 7.34

a subset of the augmented information. The strips labeled as input and true input are complete since the information is two dimensional and so is pictured completely. The velocity, velocity variance and the expectation (The expected value of the input) are all augmented spaces so that they are actually three dimensional and only a slice of the information can be shown. The image of the expectation shows the pixel at each spatial location with the highest probability. The intensity of the pixels in the velocity and velocity variance strips corresponds to the magnitude of each of these parameters at the point with the highest probability in the expectation image. The pixel intensity in the upper and lower probability images depict the maximum value of upper or lower probability at a spatial location. Each row in the diagram represents the state of the network and its parameters at the same time, so that successive rows represent the state of the network at successive iterations.

The velocity, velocity variance, and upper and lower probability images are all normalised between their maximum and minimum values to show the maximum amount of contrast. This is particularly important when comparing the difference between the open and closed loop upper and lower probabilities, which appear similar, but in some cases the maximum value attained differs by a factor greater than 2.

7.5.1 CM2 simulations

If the return from the sensor has intensities ranging from 0 to 1, then the background of sensor 1 has an intensity of 0.5 while the intensity of sensor two's background is 0.7. A zero mean gaussian noise with a variance of 0.003 for sensor 1 and 0.002 for sensor 2 is added to the background and targets to produce a noisy input. The targets are represented in the simulation as points of varying intensity, which change intensity as they move through the scene. The intensity of the points varies from zero to one so that the signal-to-noise ratio varies for each target from scene to scene.

After convergence of the program a sequence of 25 time steps were used to train the classification network as described in Chapter 3. From this point the simulation of the open and closed loop sequence diverge and the sequences shown occur 25 time steps after this separation.

Figure 7.7 shows the distribution of training points in the decision space.

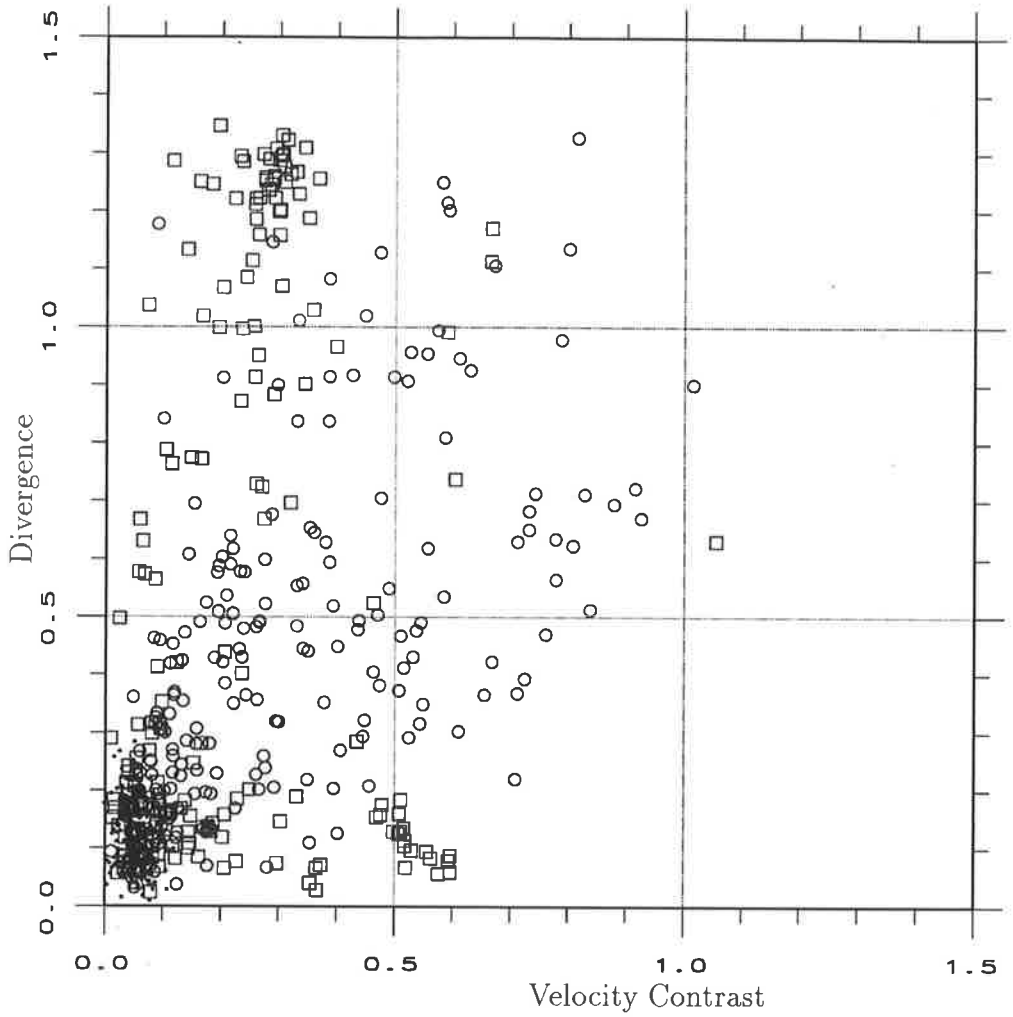

Figure 7.7: The full decision space, object training points are represented by circles, null points by squares, and background points by dots.

Rather than use all the information surrounding each point in the image in the classification module a trade off for computation speed has been formed where only two parameters are processed by the classification network for each image point. These parameters are the 'divergence' of the optical flow about each image point, and the difference between the velocity at a point, and the average of its neighbors, which shall be called the 'velocity contrast'. The manner in which the null training points surround the object points should be noted. The resolution of this plot is insufficient near the origin at this scale so it has been expanded in Figure 7.8. To further clarify the distribution of the points figures 7.9, 7.10 and

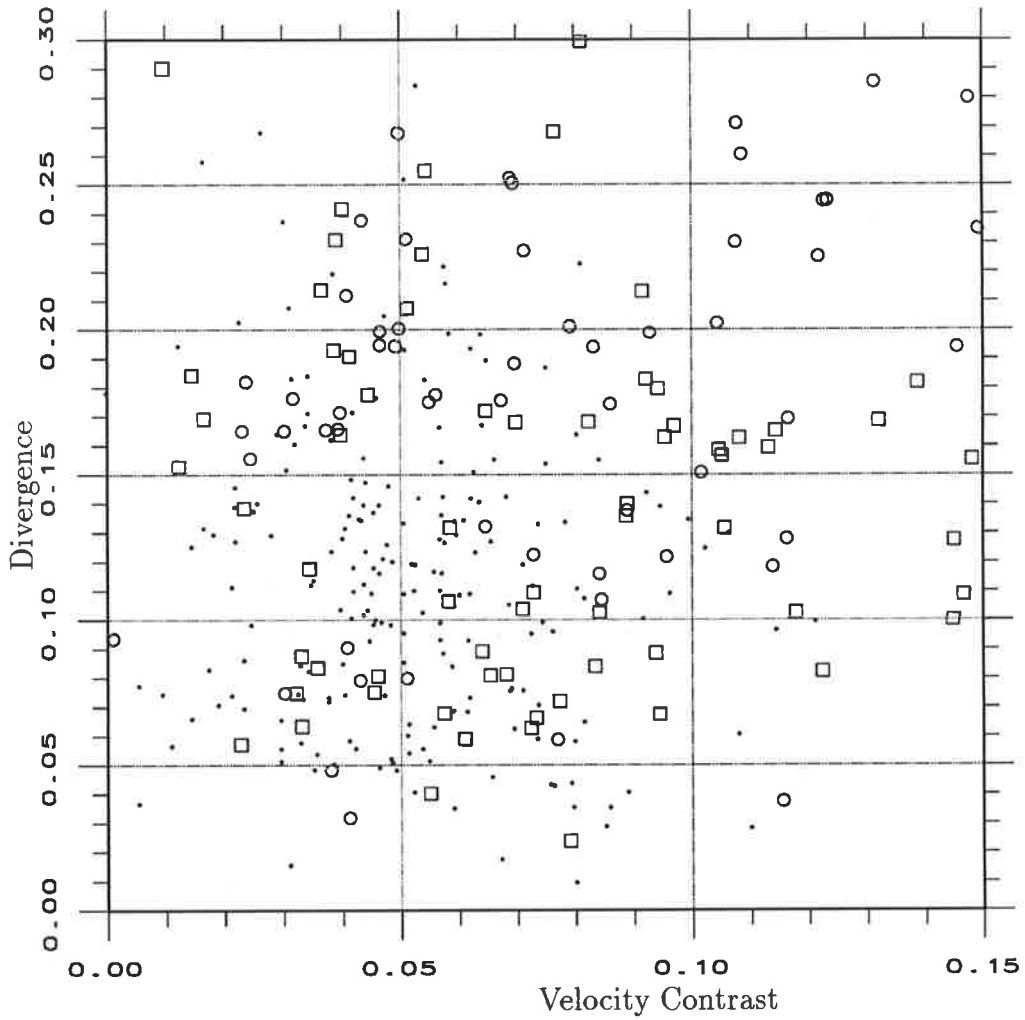

Figure 7.8: Enlarged portion of decision space, showing the region near the origin of Figure 7.7

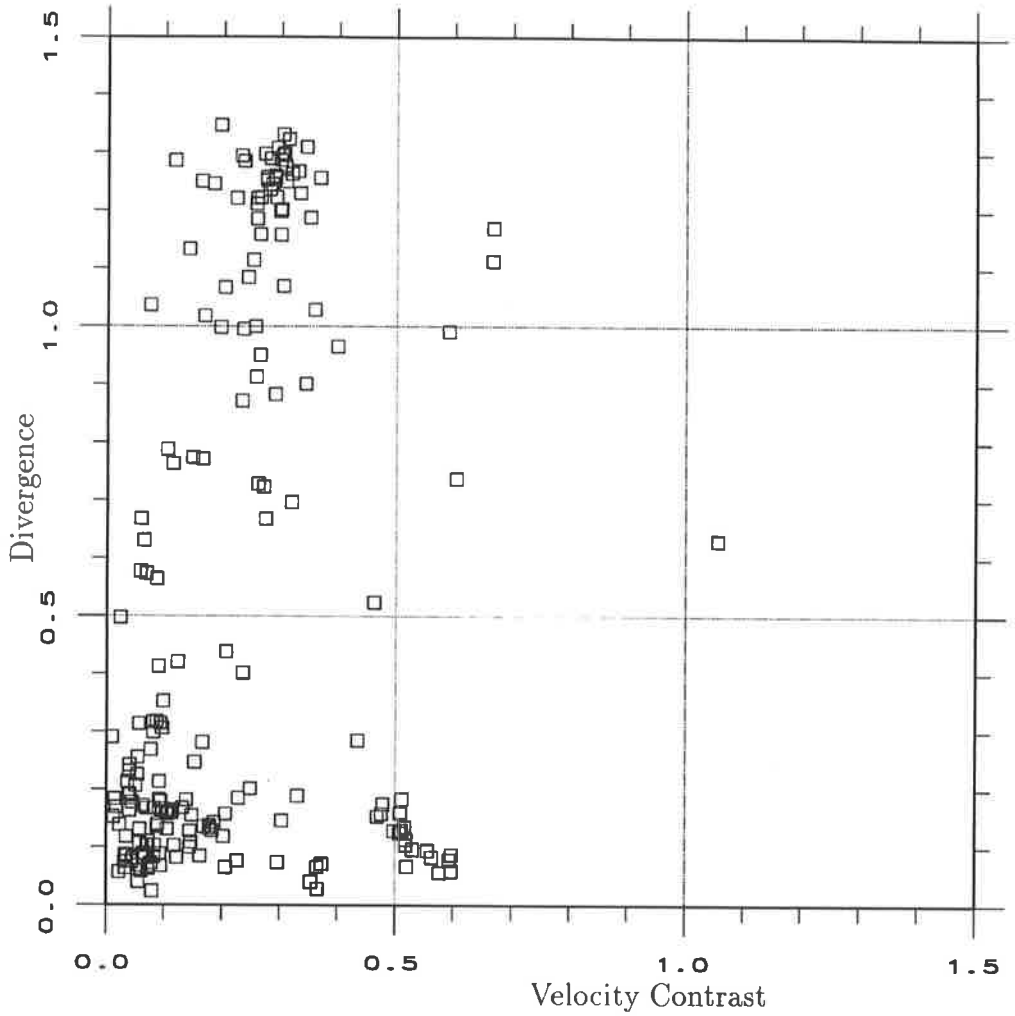

Figure 7.9: Null training points

7.11 show the distribution of each class, uncluttered with the other classes.

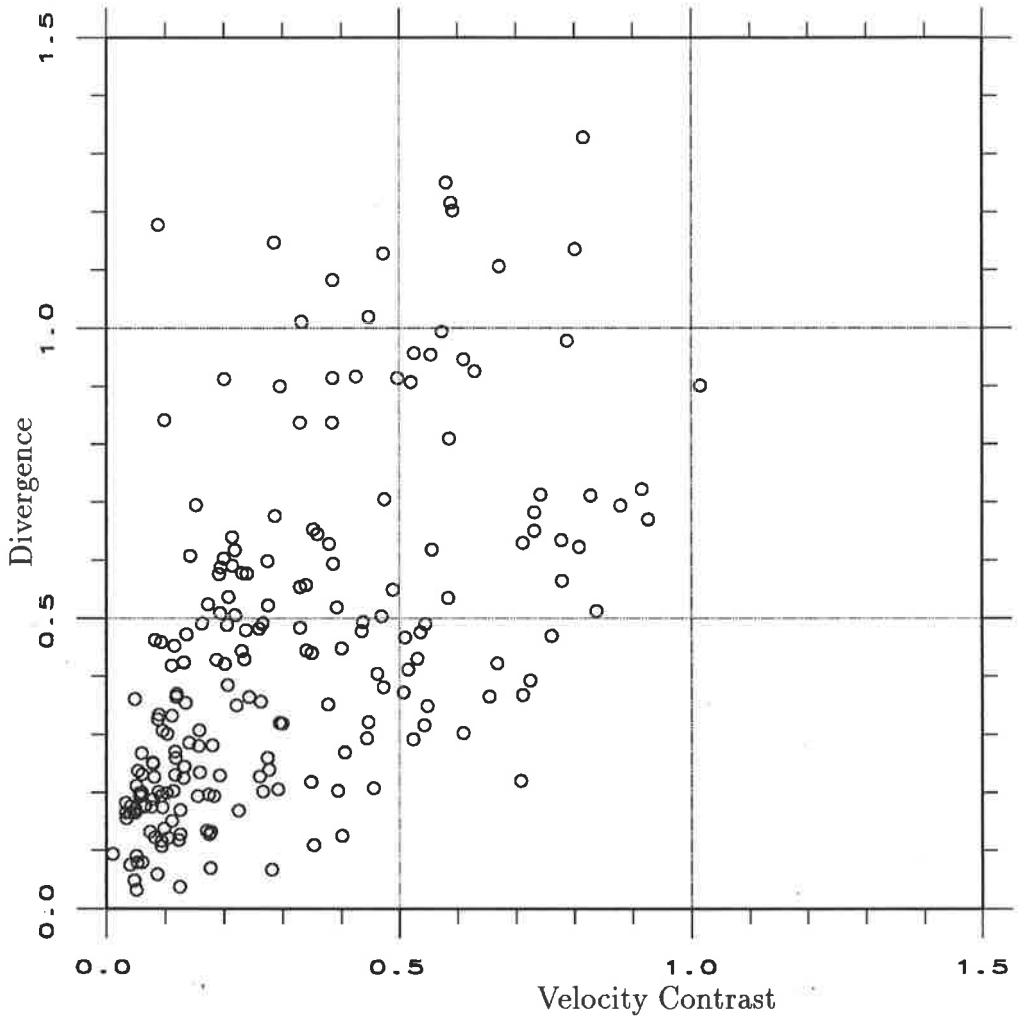

Figure 7.10: Object training points

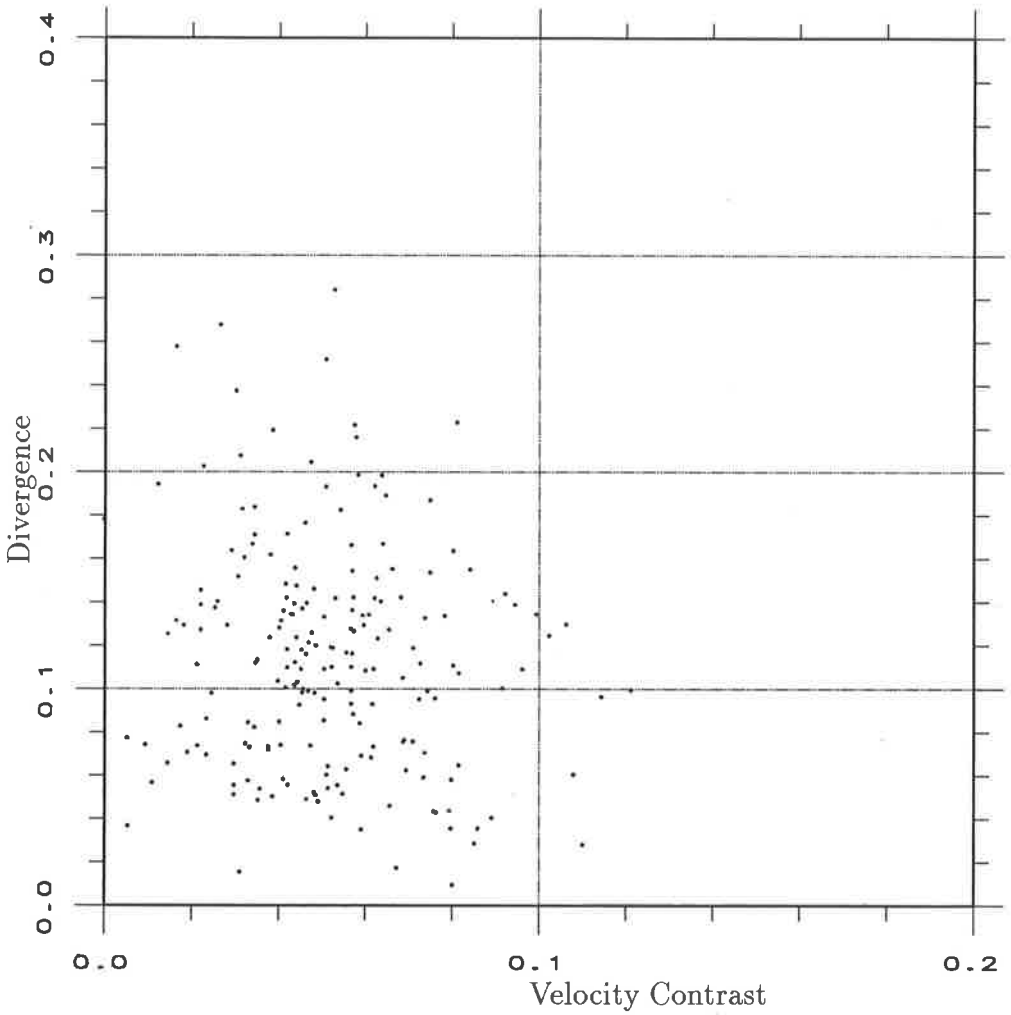

Figure 7.11: Background training points

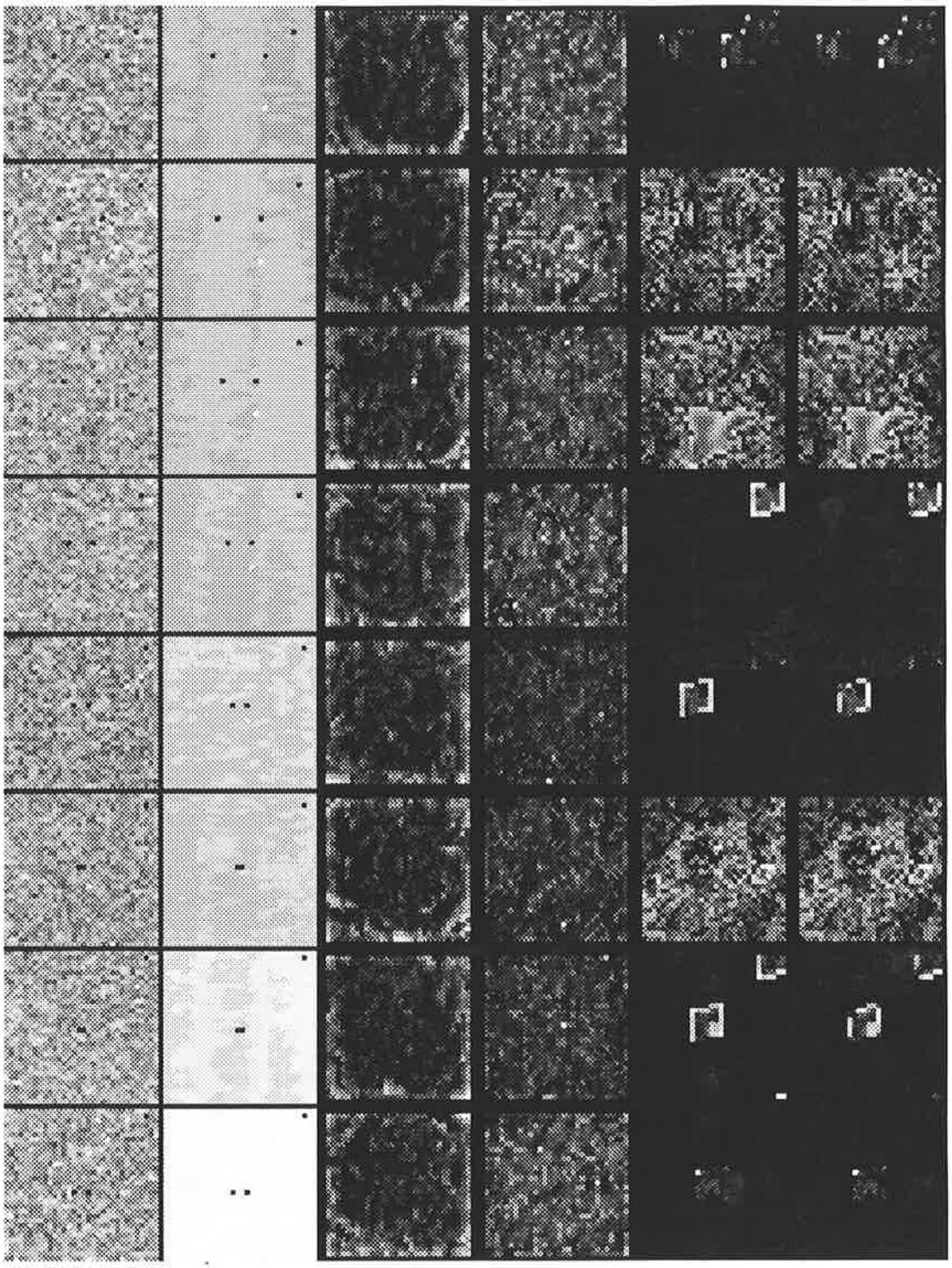

Figure 7.12: Open loop sensor 0 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

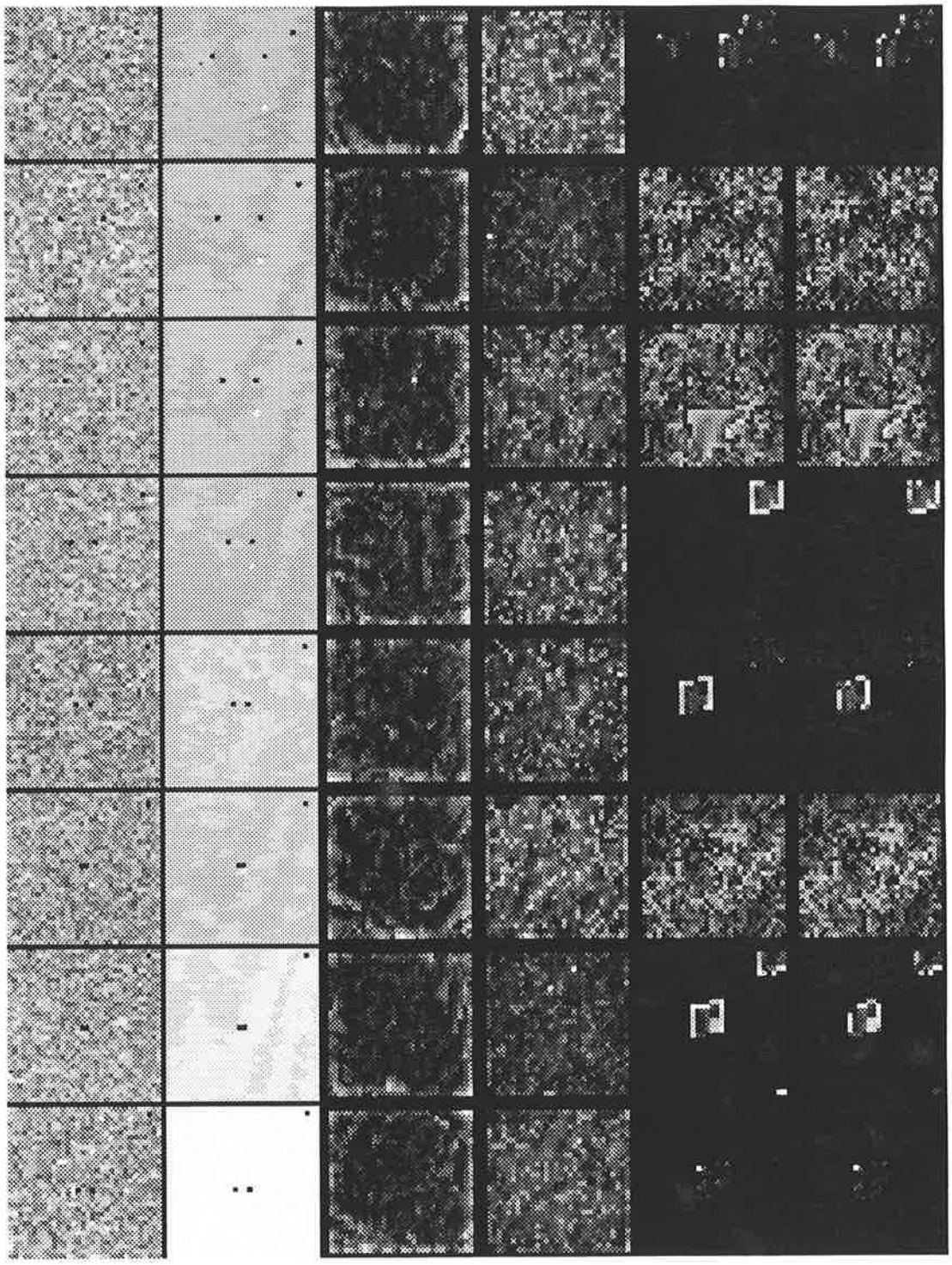

Figure 7.13: Closed loop sensor 0 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

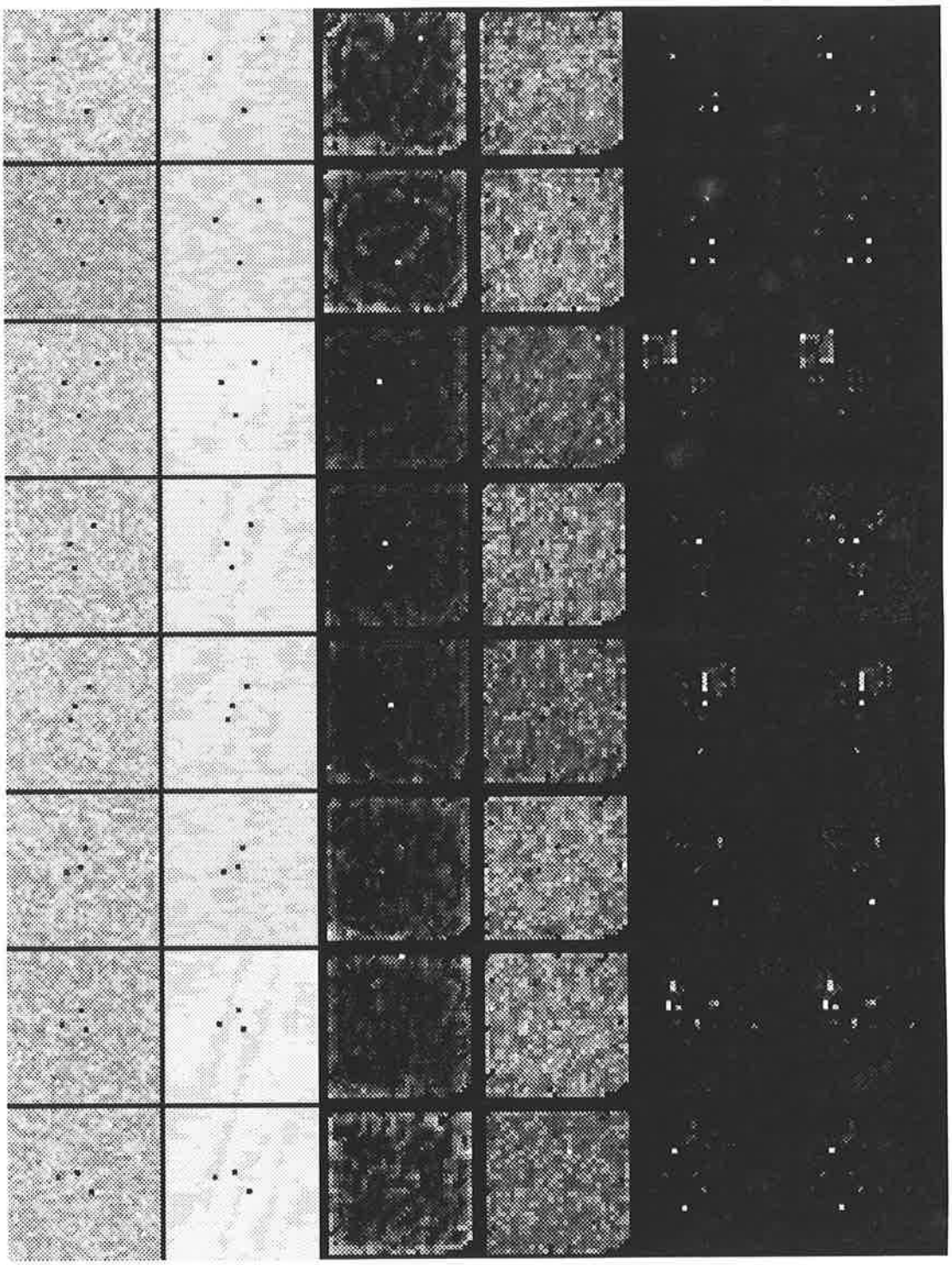

Figure 7.14: Open loop sensor 1 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

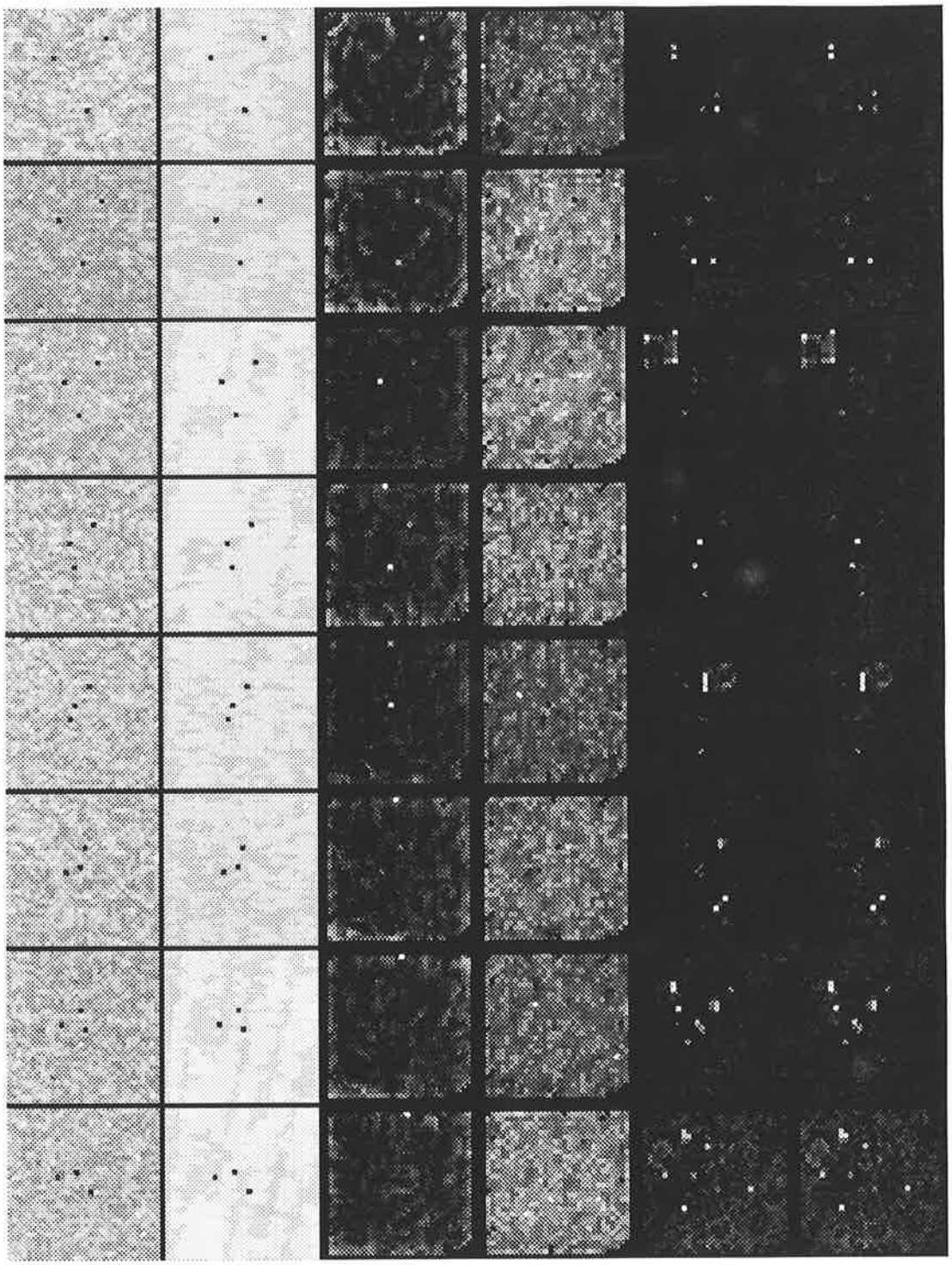

Figure 7.15: Closed loop sensor 1 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

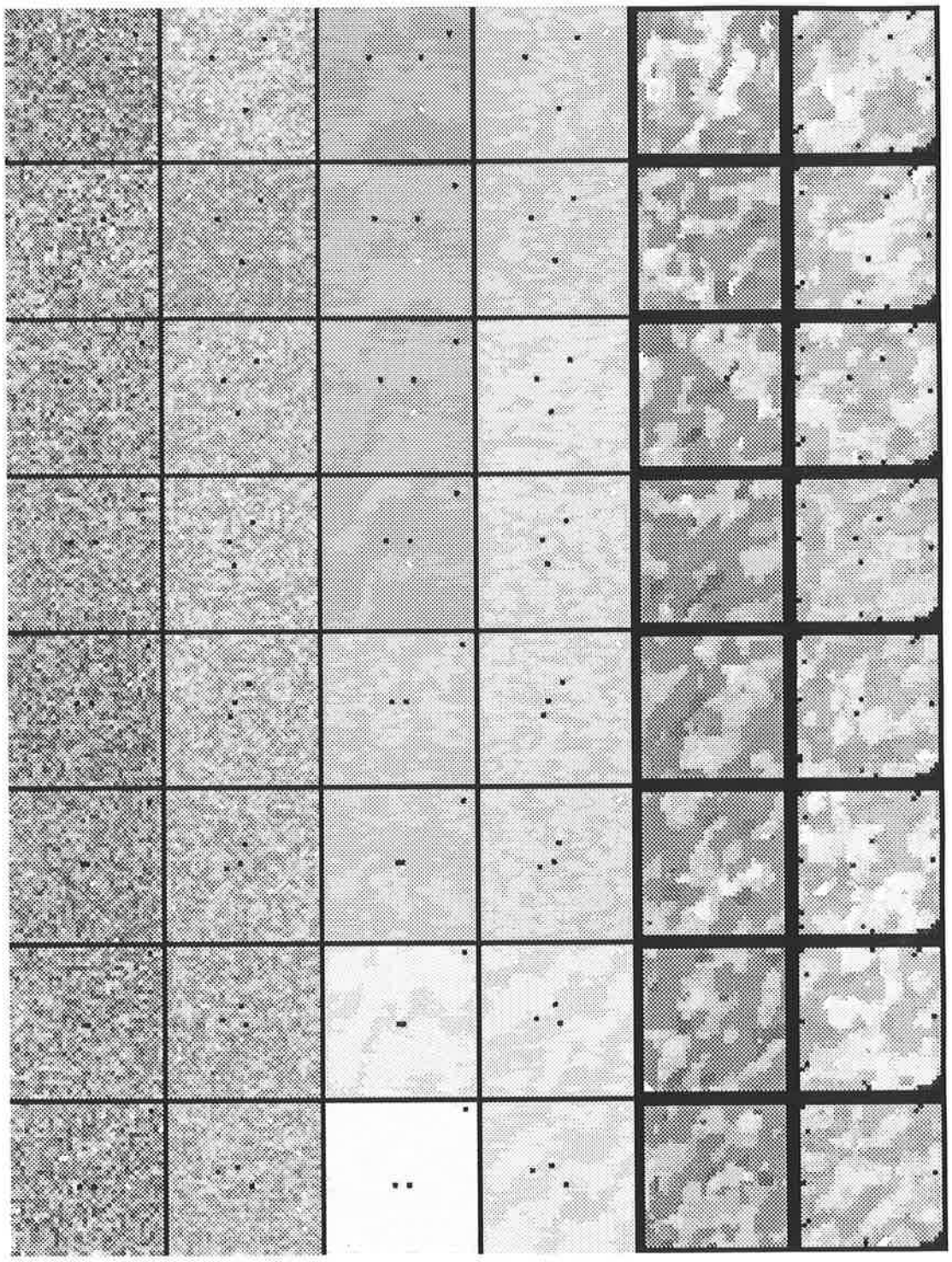

Figure 7.16: Open loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, sensor 0 expectation, sensor 1 expectation, see Figure 7.3)

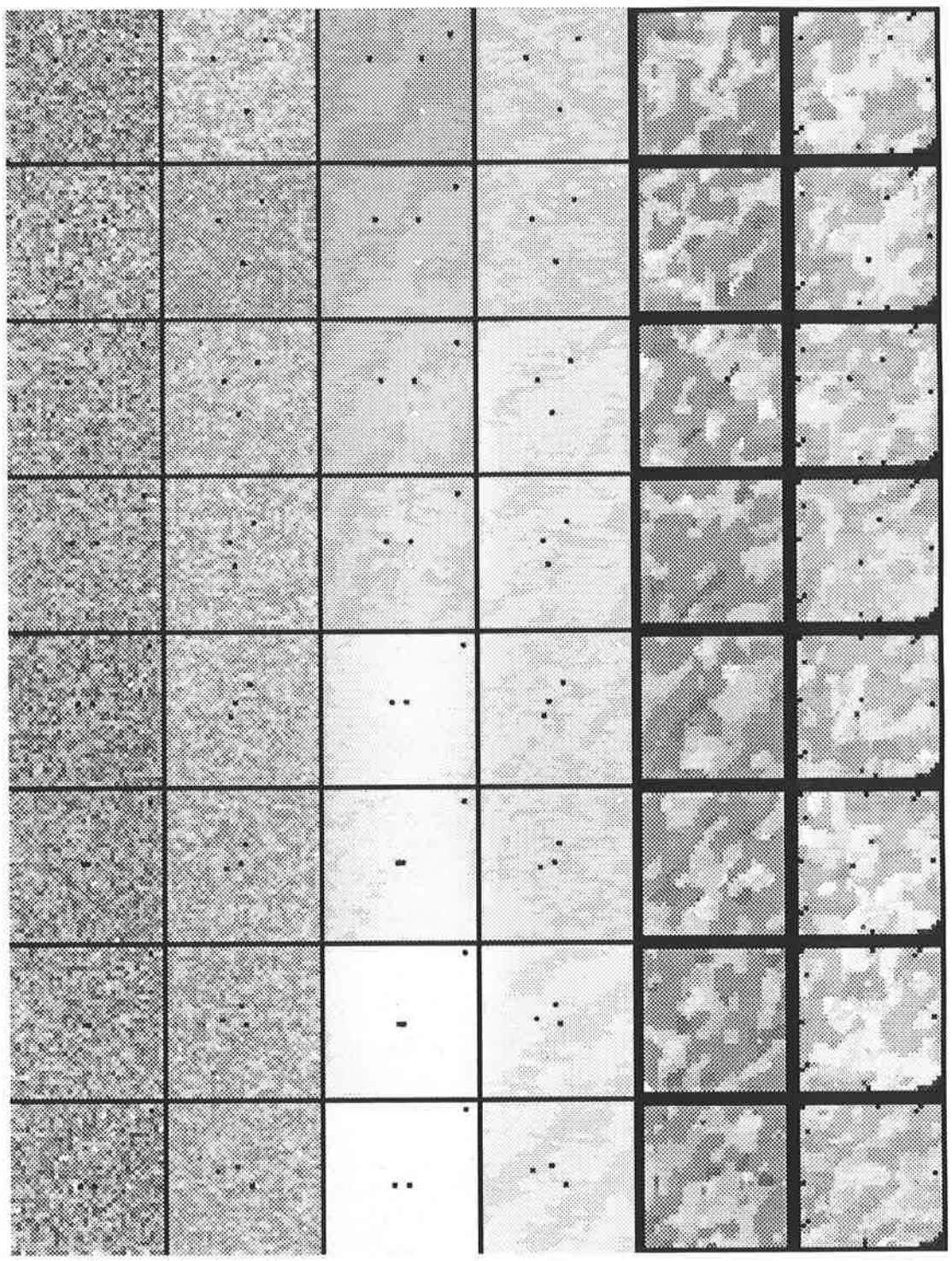

Figure 7.17: Closed loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, sensor 0 expectation, sensor 1 expectation, see Figure 7.3)

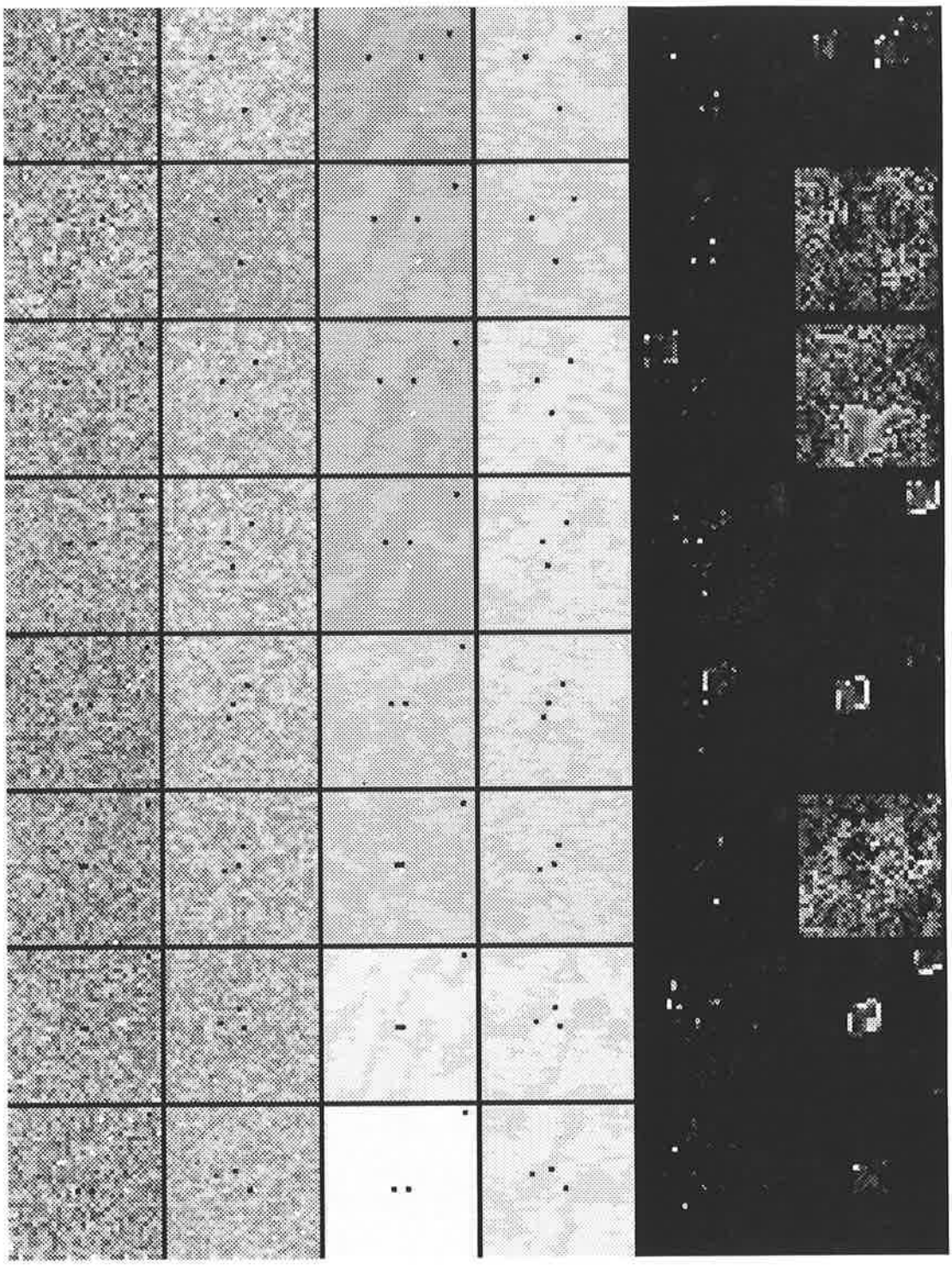

Figure 7.18: Open loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, global upper probability and lower probability, see Figure 7.4)

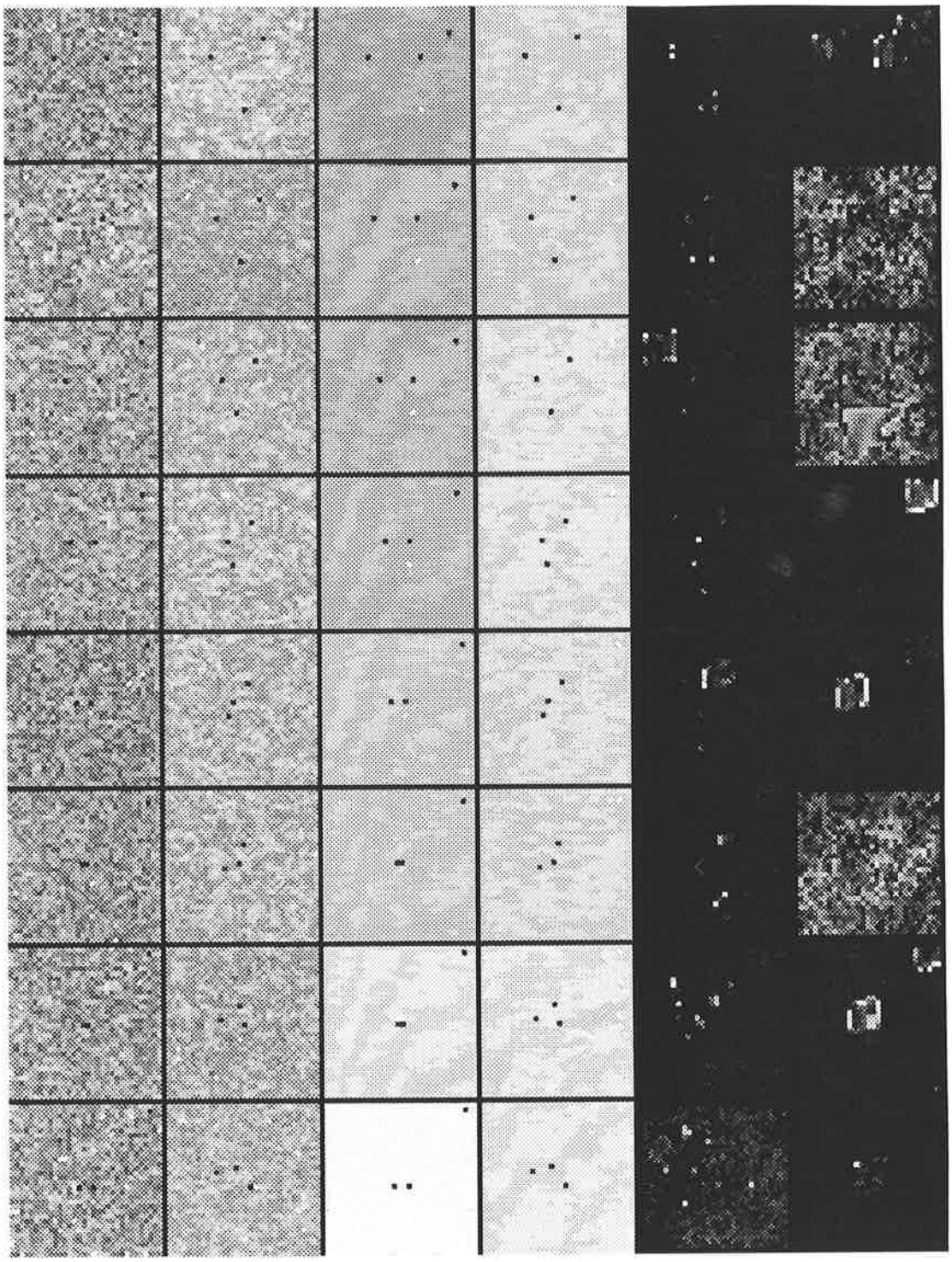

Figure 7.19: Closed loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, global upper probability and lower probability, see Figure 7.4)

At first glance there appears to be very little difference between the open and closed loop responses of the individual sensors (Figure 7.12, 7.13, 7.14 and 7.15.) It should be remembered that these pictures display only a slice of all the information available, since the augmented (three dimensional) data can be displayed only by sampling the three dimensional space in a way that a two dimensional space is achieved. For lack of a better sampling criteria the expectation probability is used as a selection tool, so that points in the same position in the augmented space as the point with the highest expected probability at each point in the two dimensional space are displayed. As Figure 7.16 and 7.17 show the expectation is not the same as the true input. Many of the objects are not shown, yet they still exist at a lower probability, this is demonstrated by comparing the object probabilities in Figure 7.12, where objects are definitely detected, and the expectation of sensor 0 in Figure 7.16, where the objects are not visible. The feedback biases the velocity determination routines, so that they have a greater chance of rejecting false objects and acquiring true objects. Consequently, an object entering the scene is recognised as an object more quickly when fusion is occurring (unfortunately this is difficult to show in a plot,) also, when the object probabilities indicate false objects, the probability assigned to these false objects is lower when the loop is closed than when it is open. Examination of sensor 1's closed loop response, Figure 7.15, indicates that there are fewer false objects than there are in sensor 1's open loop response, Figure 7.14, particularly in the first couple of time steps, where the other sensor's (sensor 0) objects are distinct.

In the final frames of sensor 1 (Figure 7.14,) three objects of similar color, and different velocity vectors cross close to each other. Since the points are close the association of a point with a particular track becomes significant. This is the problem of data association which has been avoided in this approach; however, the limitations and trade offs in data association routines for manoeuvre detection vs. track accuracy are also present in this routine. The result on the velocity estimates is that they become multimodal since there will be several consistent possible velocities. The velocity is modeled as a Gaussian distribution for computational convenience, the Gaussian is no longer a good estimate of what is actually happening and the result is an increase in the velocity variance estimate.

In the last few time steps several of the objects in sensor 0's view appear to collide, of particular significance is the collision between the objects of the same color. When more than one identical object (same color) exists at the same point then they become a composite object so that the resultant velocity estimates become that of the group rather than the individuals. In the case presented the velocity of the group is different to the velocity of both individuals.

The reduction in the velocity estimate accuracy in the final time steps reduces the consequential probability of object detection, which can be seen in both figures 7.16 and 7.17 where the object vanishes from the expectation. Sensor 1 does maintain some confidence of the objects velocity and this knowledge can be seen in the effect on sensor 0 that occurs in the final time step (Figure 7.13) where at the position of the correct object there is a faint but distinct increase in probability in relation to its neighbours. This is not seen in the open loop response (Figure 7.12.)

7.5.2 DEC5000 simulations

To reduce the computational time required in simulation the hypothesis network's training, as depicted in Chapter 3, was modified so that class distributions were assigned manually. This has the advantage that many training points can be replaced by a function so that the number of evaluations needed to be performed is reduced. It has the disadvantage that the fine detail in the class boundaries will be lost.

The class boundaries were determined by running the network on the simulation in an open loop mode and plotting a random selection of points from each class. The results are shown in figures 7.20, 7.21, 7.22 and 7.23.

The noise processes in this simulation are identical to those in the simulation performed on the CM2. The targets however are represented as points which are blurred in a gaussian manner with a variance of 0.09375 if the width of the spatial frame is 1.0.

The results obtained are presented in Figures 7.24 to 7.31

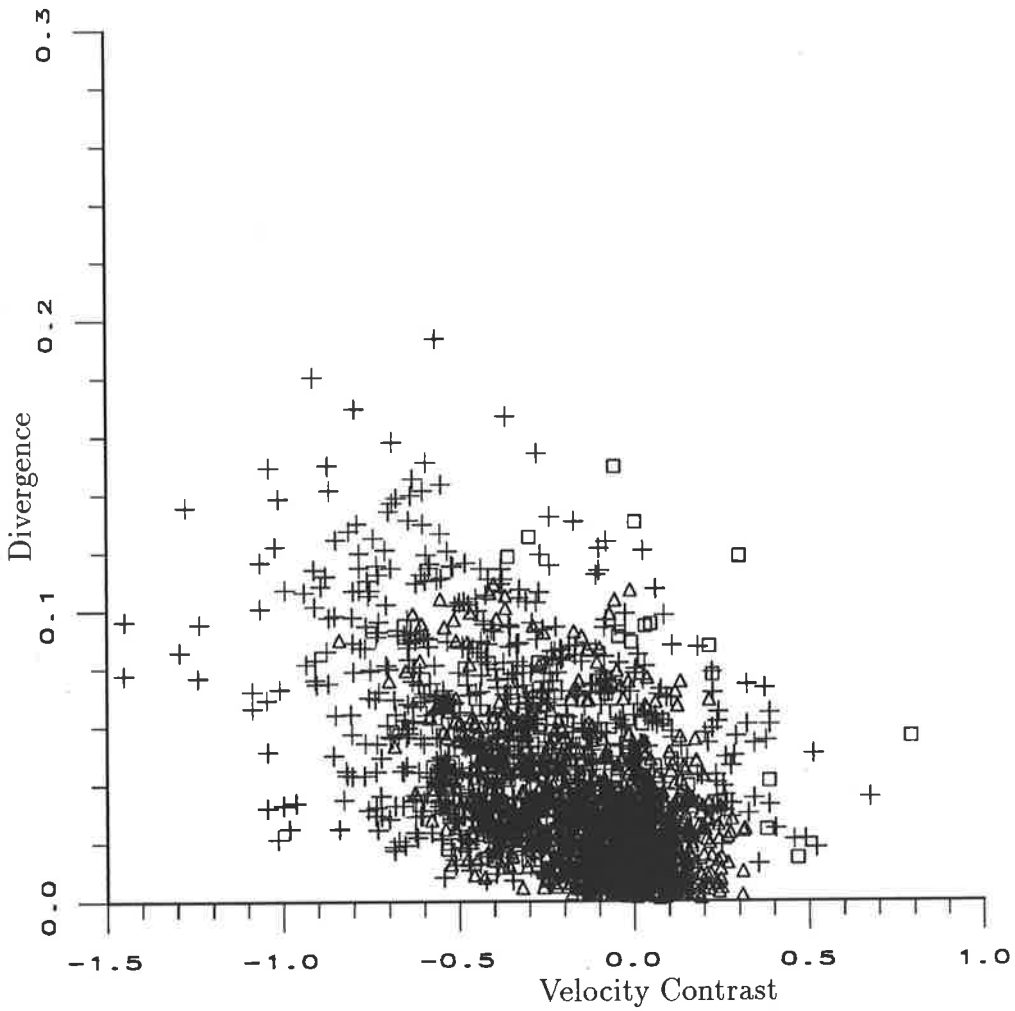

Figure 7.20: The full decision space, object training points are represented by squares, nul points by crosses, and background points by triangles.

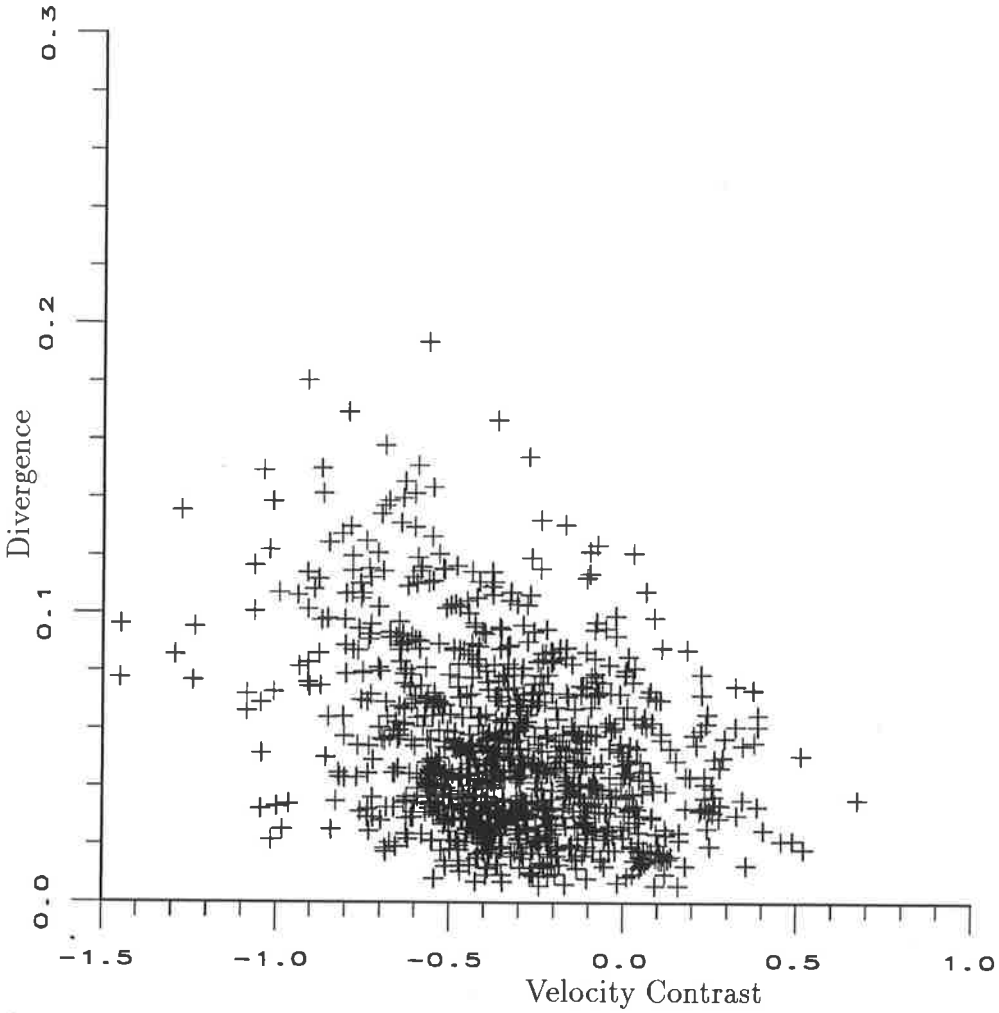

Figure 7.21: Null training points

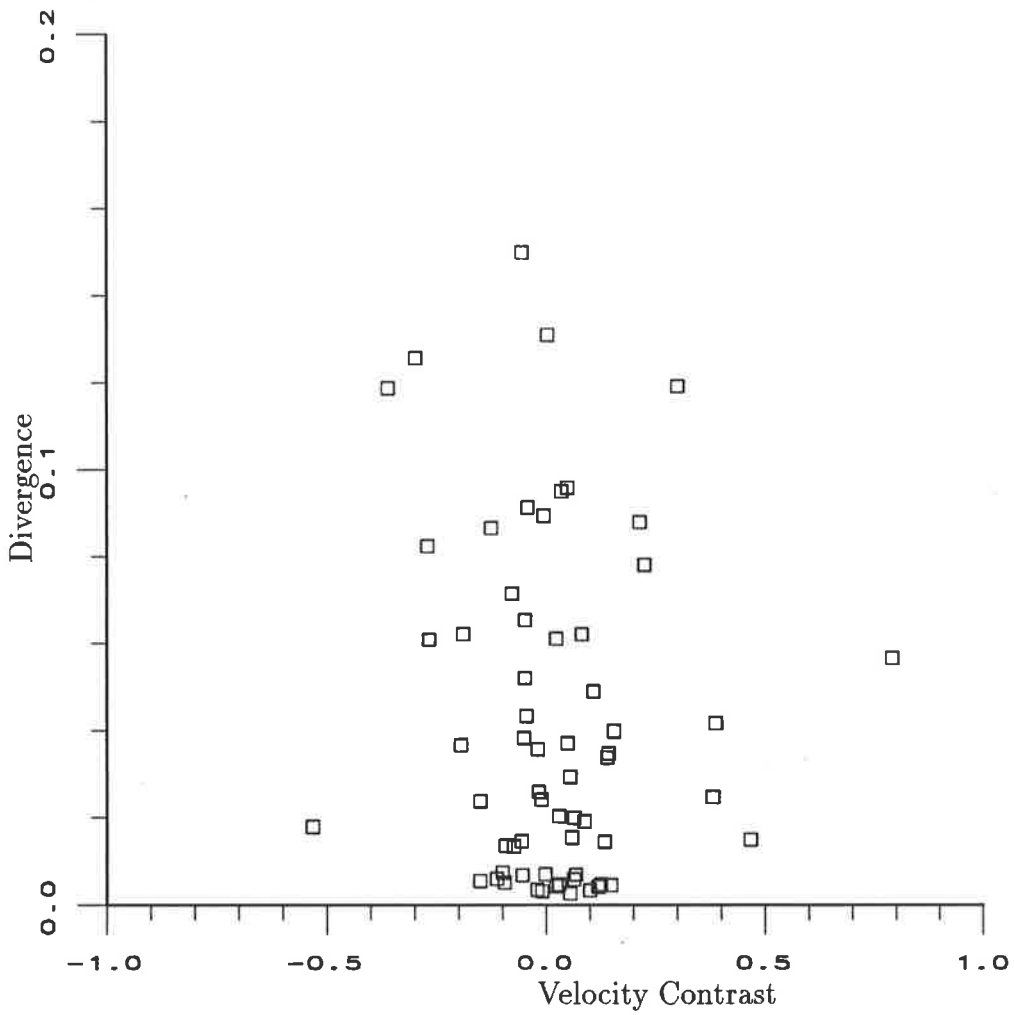

Figure 7.22: Object training points

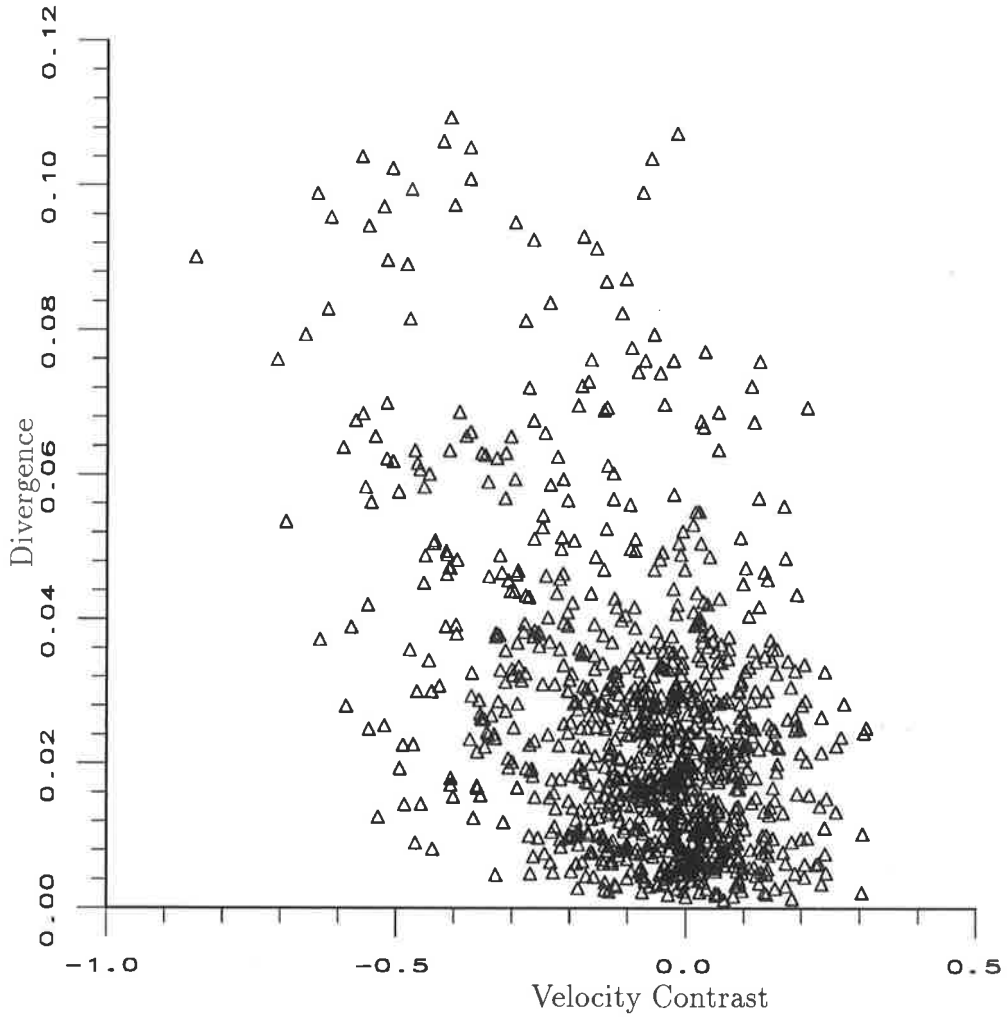

Figure 7.23: Background training points

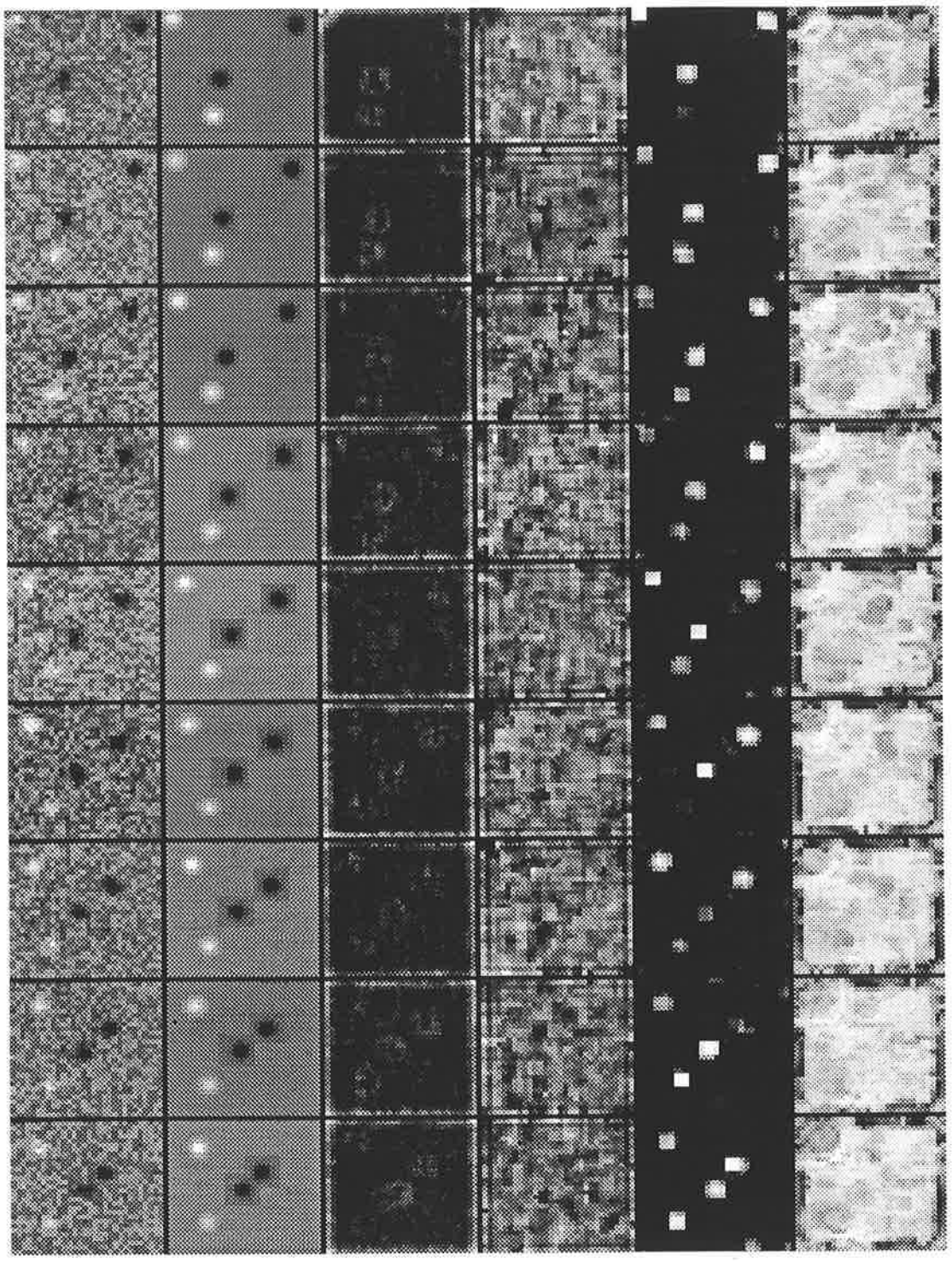

Figure 7.24: Open loop sensor 0 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

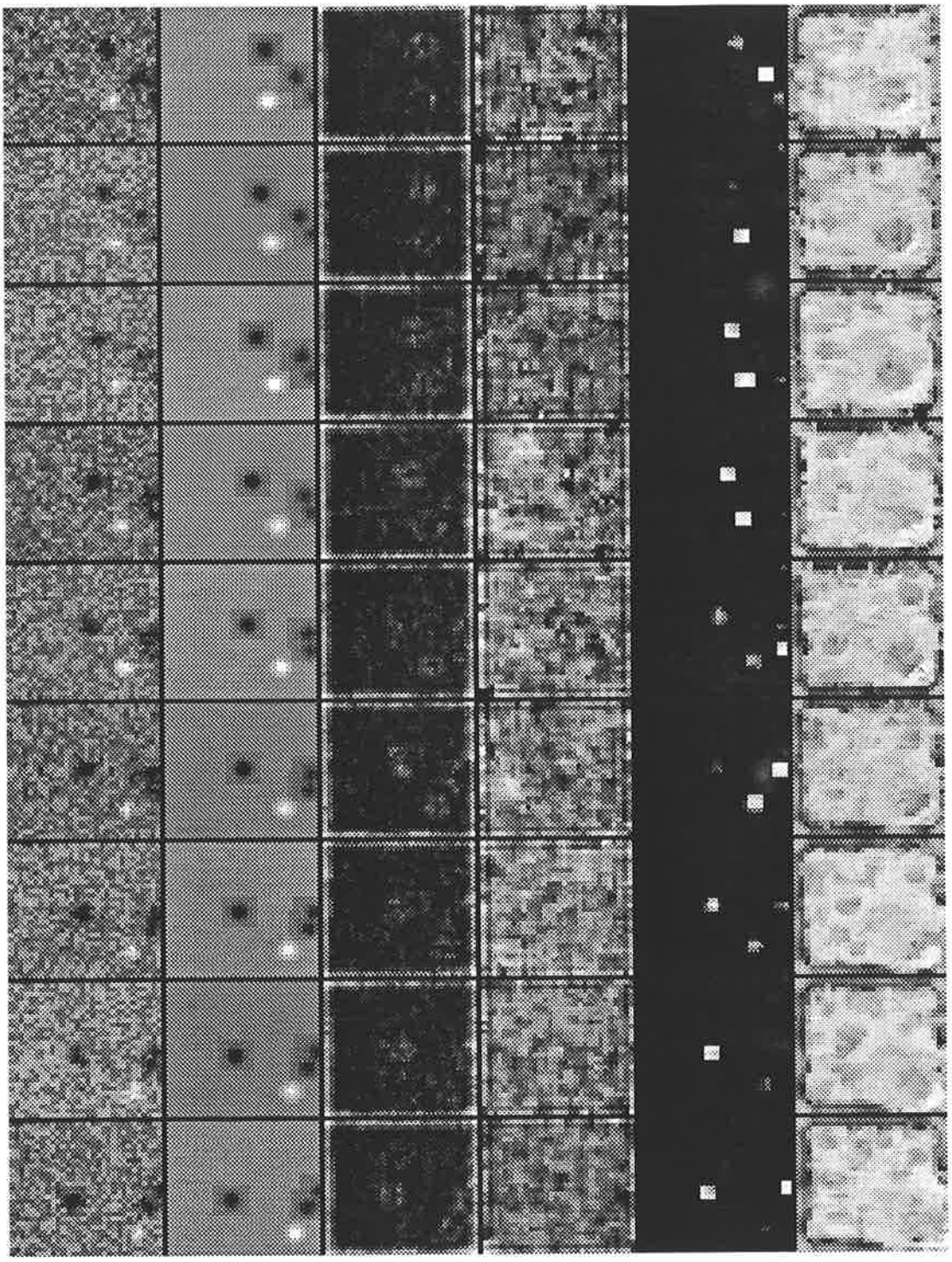

Figure 7.25: Closed loop sensor 0 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

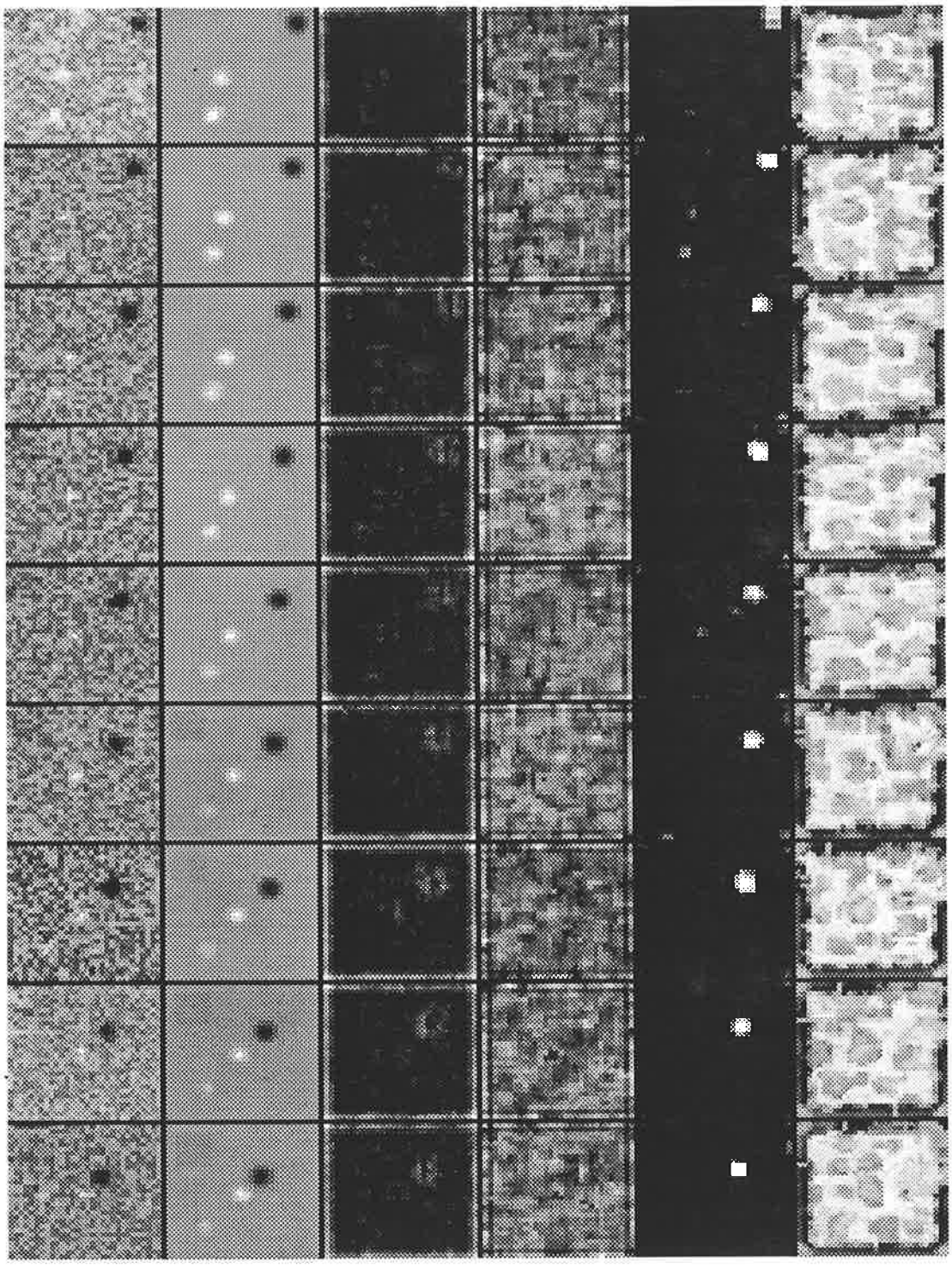

Figure 7.26: Open loop sensor 1 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

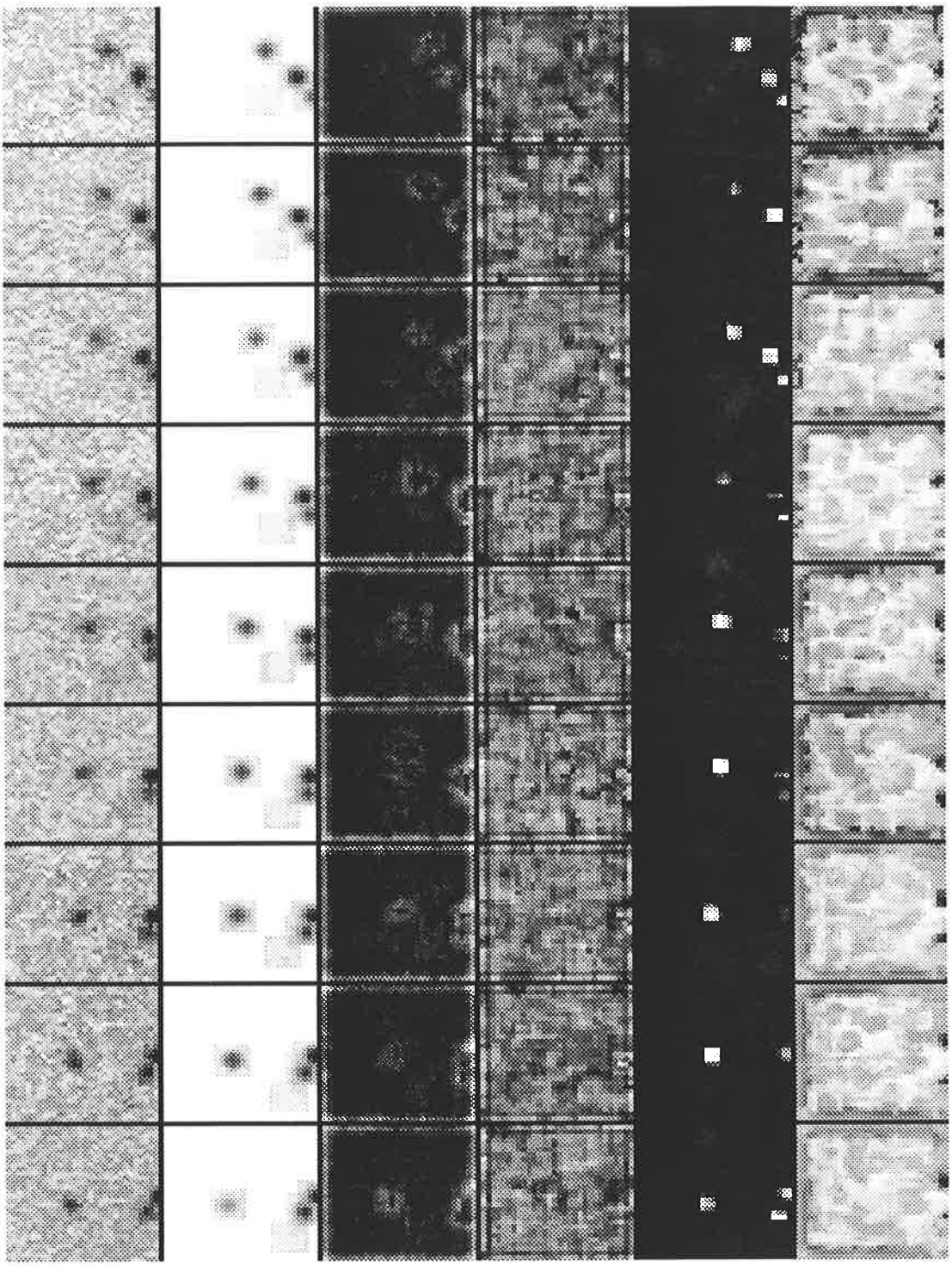

Figure 7.27: Closed loop sensor 1 response. (input, true input, velocity, velocity variance, upper probability and lower probability, see Figure 7.2)

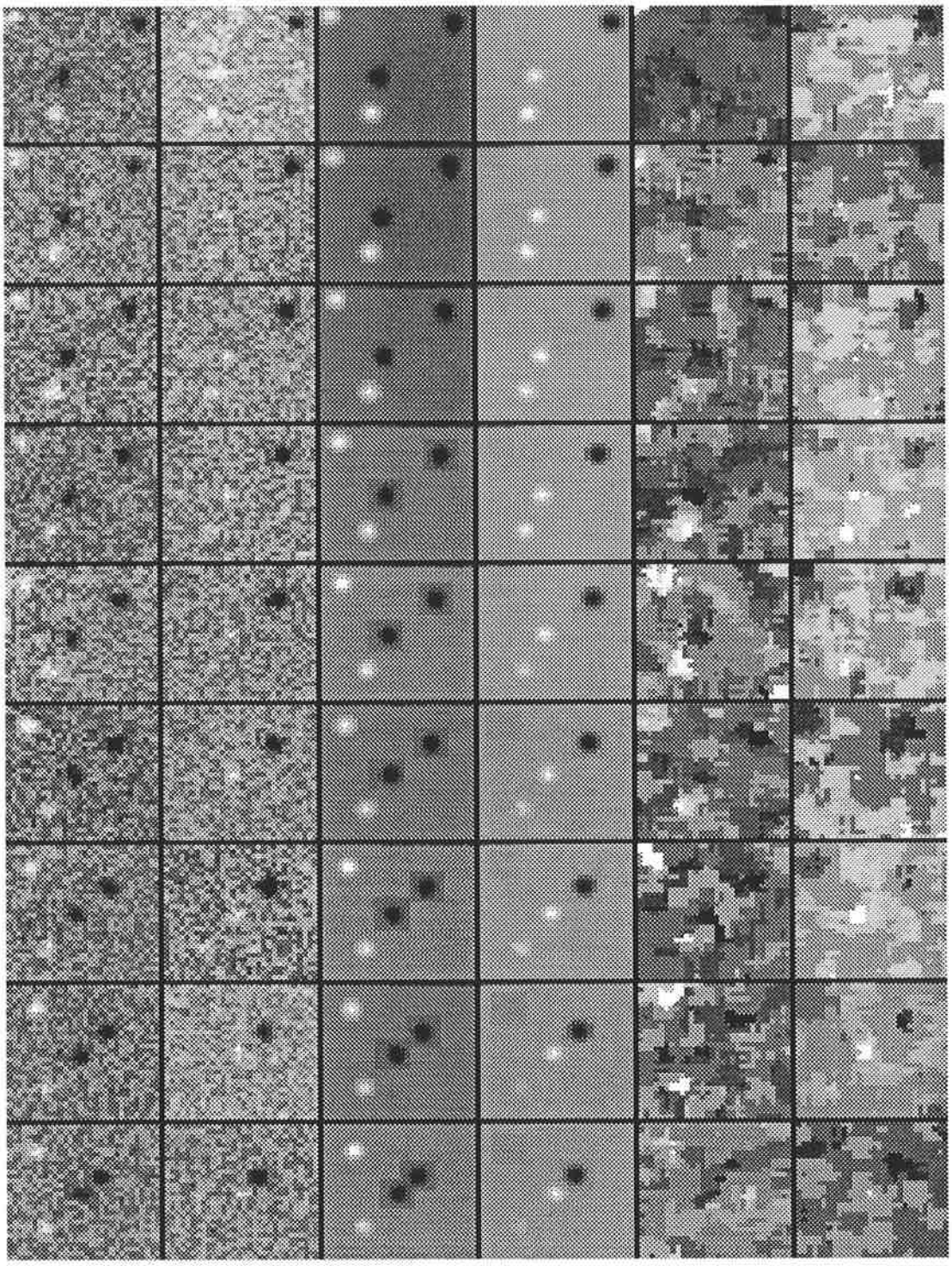

Figure 7.28: Open loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, sensor 0 expectation, sensor 1 expectation, see Figure 7.3)

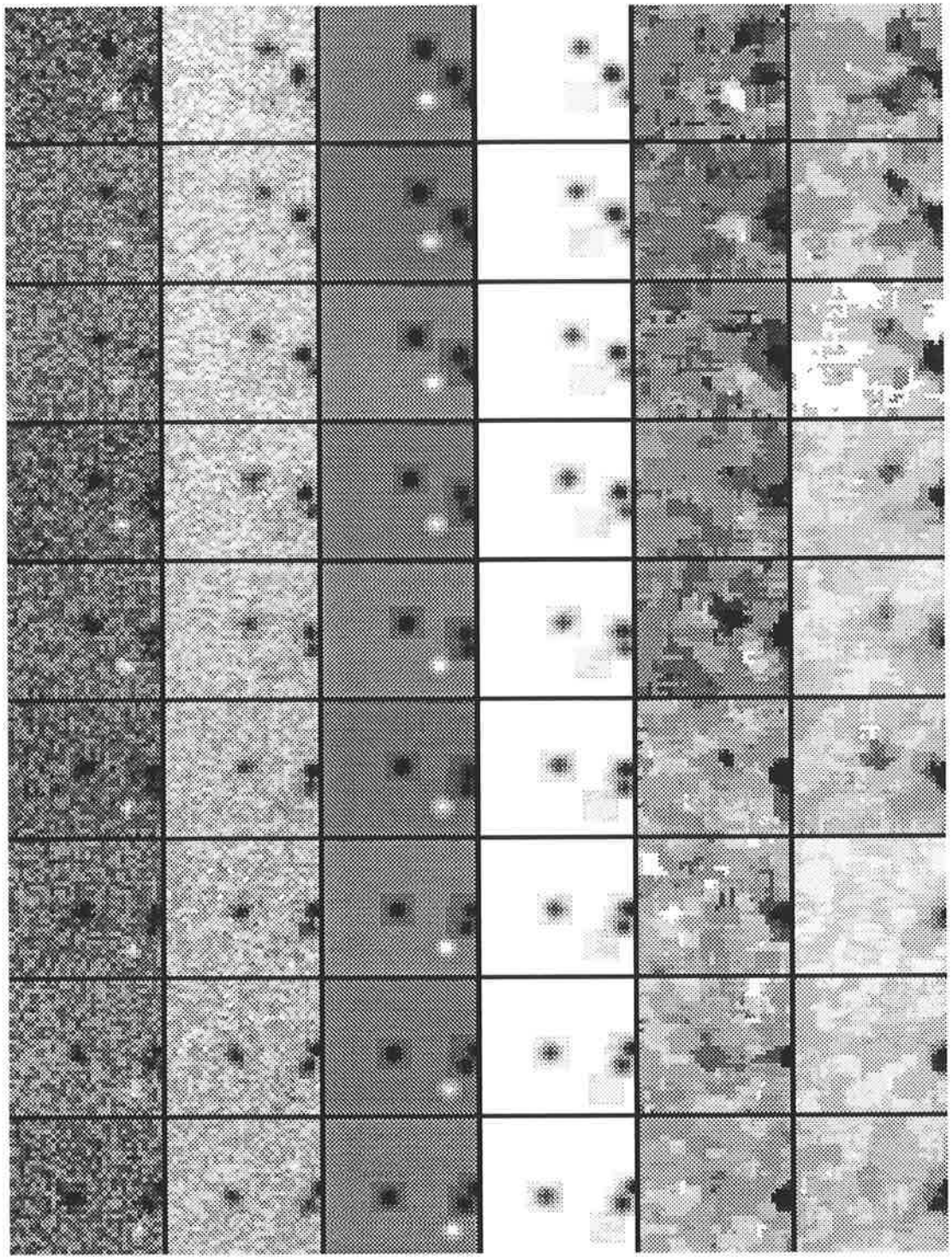

Figure 7.29: Closed loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, sensor 0 expectation, sensor 1 expectation, see Figure 7.3)

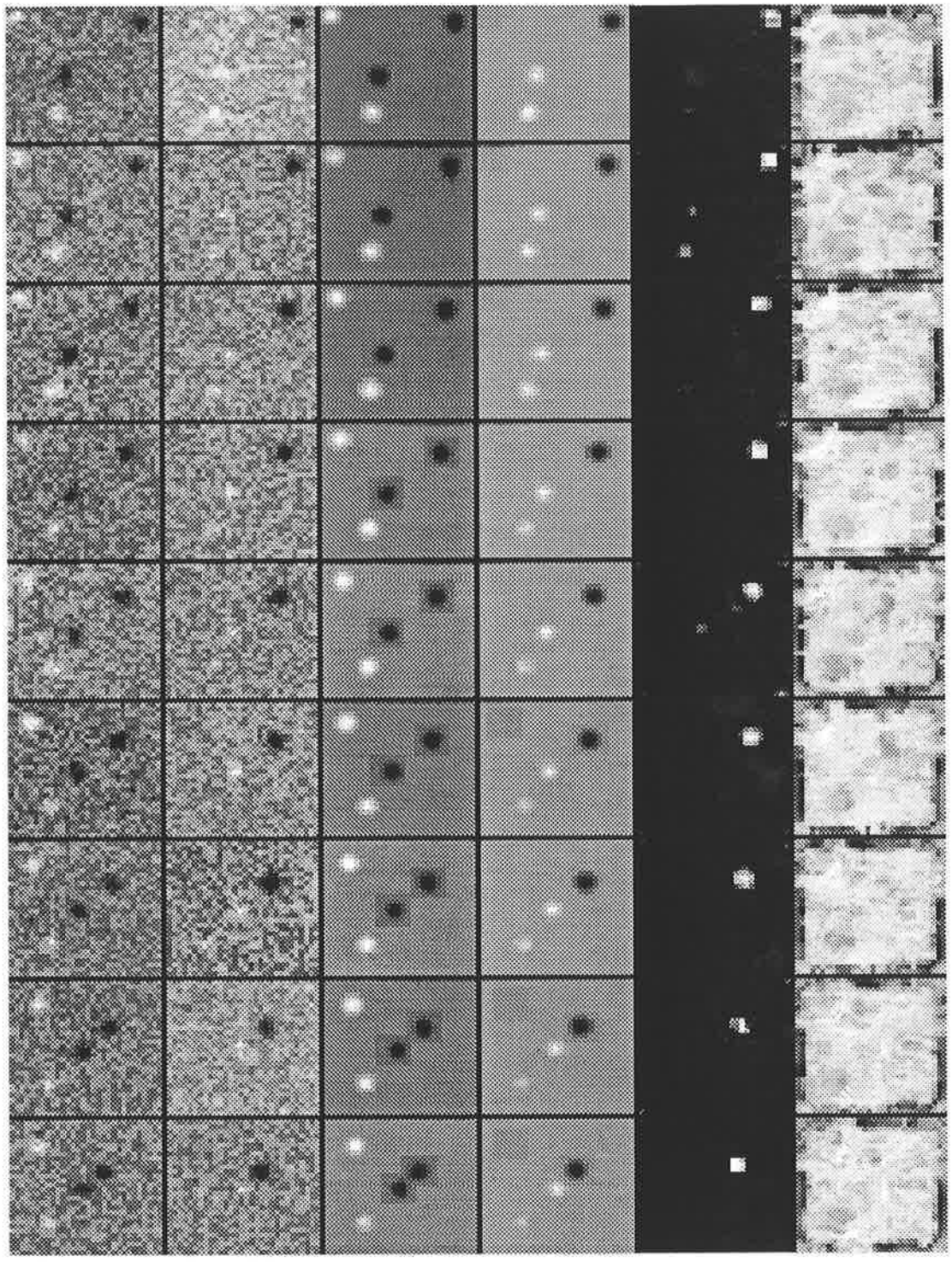

Figure 7.30: Open loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, global upper probability and lower probability, see Figure 7.4)

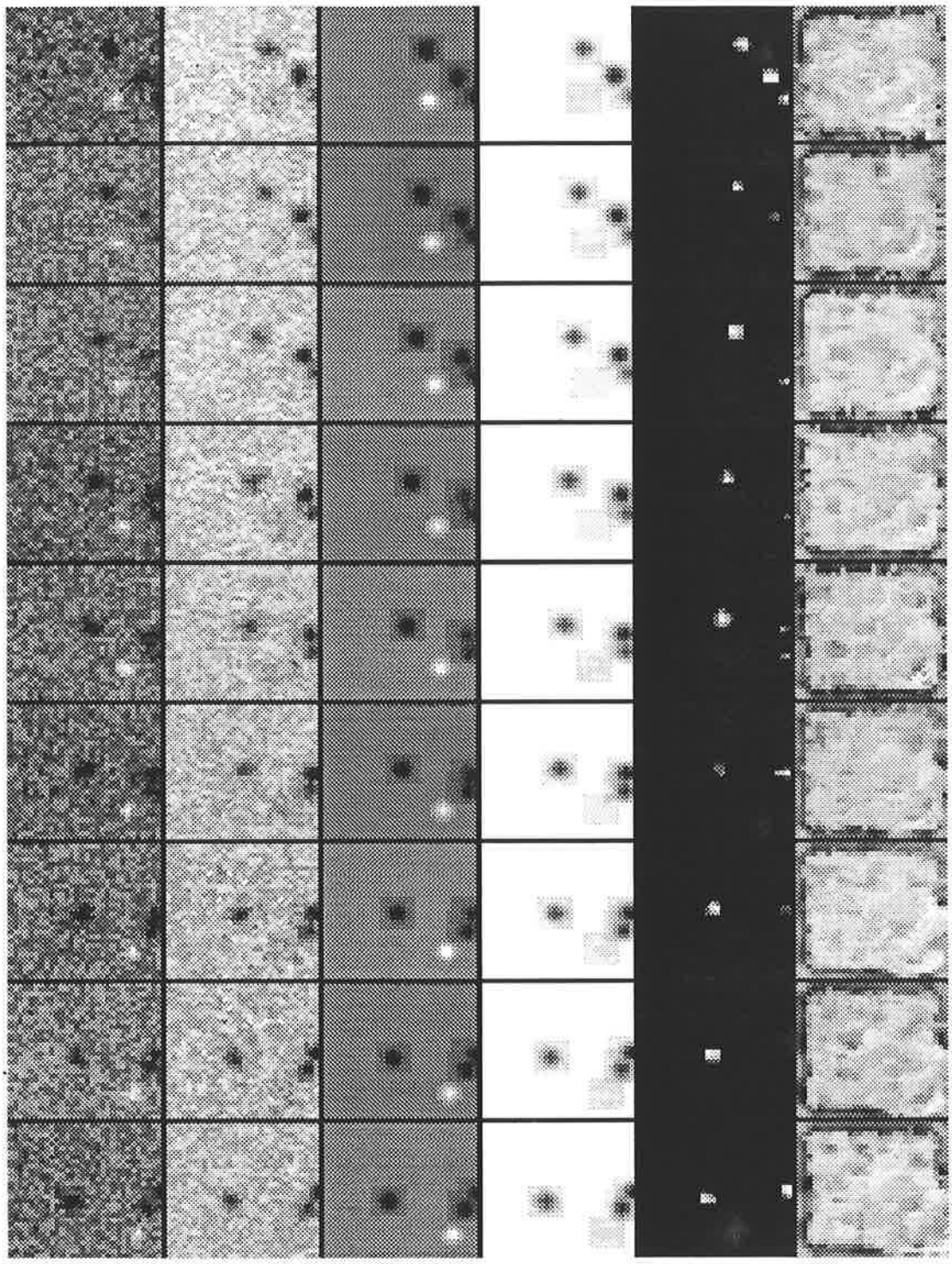

Figure 7.31: Closed loop response. (sensor 0 input, sensor 1 input, sensor 0 true input, sensor 1 true input, global upper probability and lower probability, see Figure 7.4)

The simulations on the DEC5000 have produced slightly different results, as a consequence of both the different hypothesis information and the blurring of the object point. Most noticeable is the effect on the upper and lower probabilities. The upper probability gives a good indication of the location of the object while the lower probability indicates almost nothing. The results of fusion as displayed in Figure 7.31 indicate that the common objects in the sensors has once again been detected with the objects which are at different locations between the sensors being rejected at a global level although they persist at the local level.

7.6 Simulation with Real Data

7.6.1 IR Data

Information was obtained from an IR array of a distant object. This information had a resolution which would have made processing impossible. To solve this problem, a 32 by 32 pixel subimage containing the object was extracted and quantised to 32 levels so that the simulator used in earlier parts of the experiment could be utilised. This quantisation unfortunately removes a large amount of information. Two streams of data were obtained, so that in one image the object is twice as distant as in the first, unfortunately no other information regarding the image sequences can be revealed. The simulations were performed using the classification network developed on the simulated data.

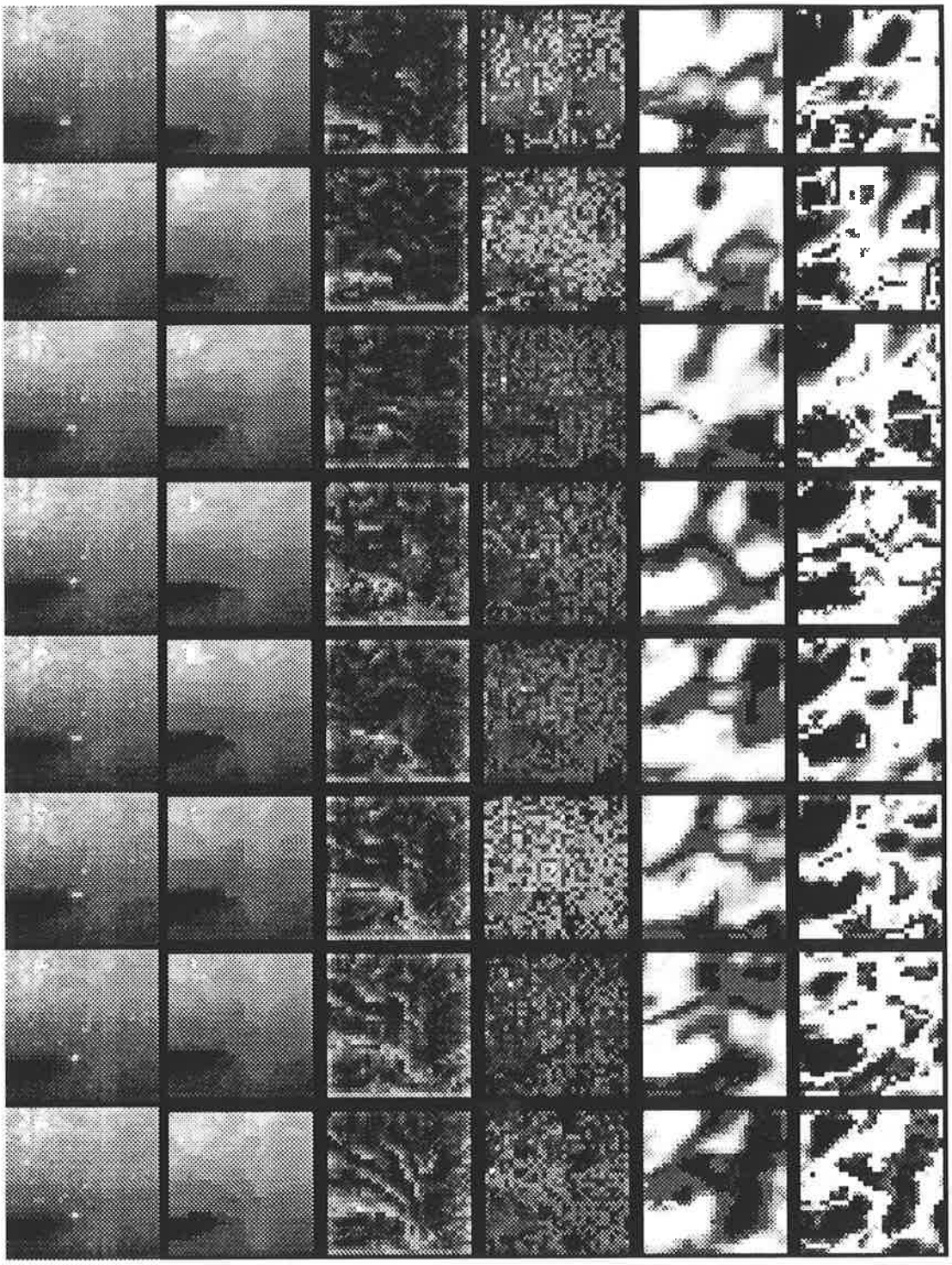

Figure 7.32: Real data, range R, (input, expectation, velocity, velocity variance, upper probability and lower probability, see Figure 7.5)

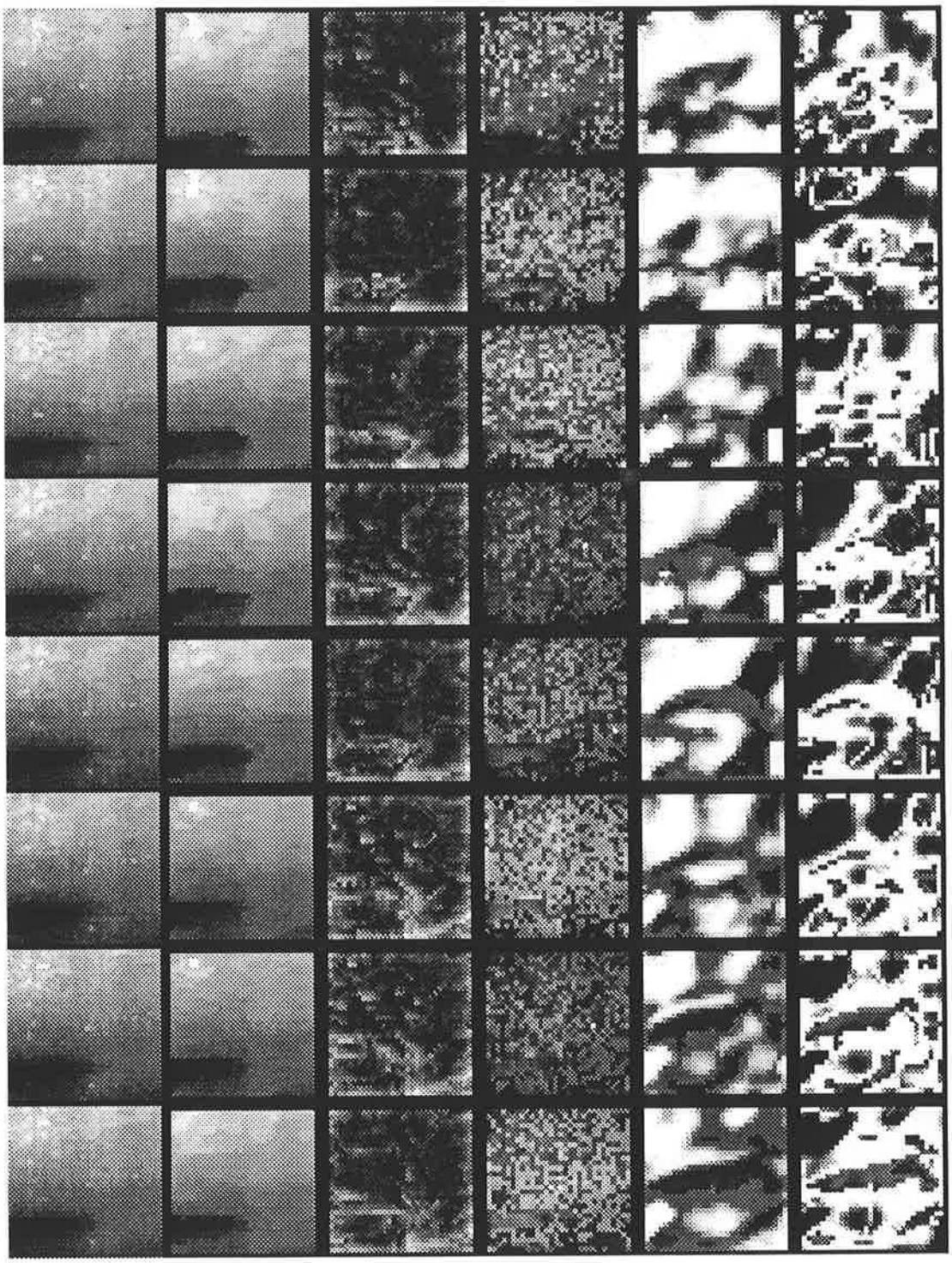

Figure 7.33: Real data, range $2R$, (input, expectation, velocity, velocity variance, upper probability and lower probability, see Figure 7.5)

It is quite difficult to analyse the results in Figures 7.32 and 7.33 since unlike the simulations the upper and lower probabilities are a totally different shape. To aid understanding two sets of new images are generated which represent the doubt in the image, and the target believability.

$$doubt = U_{object} - L_{object}, \tag{7.1}$$

$$believability = U_{object} L_{object}. \tag{7.2}$$

Target believability is not based on any statistical reasoning and is just a measure which shows when there is a reasonable level of both U_{object} and L_{object}. This measure has the effect of weighting target belief by classification knowledge.

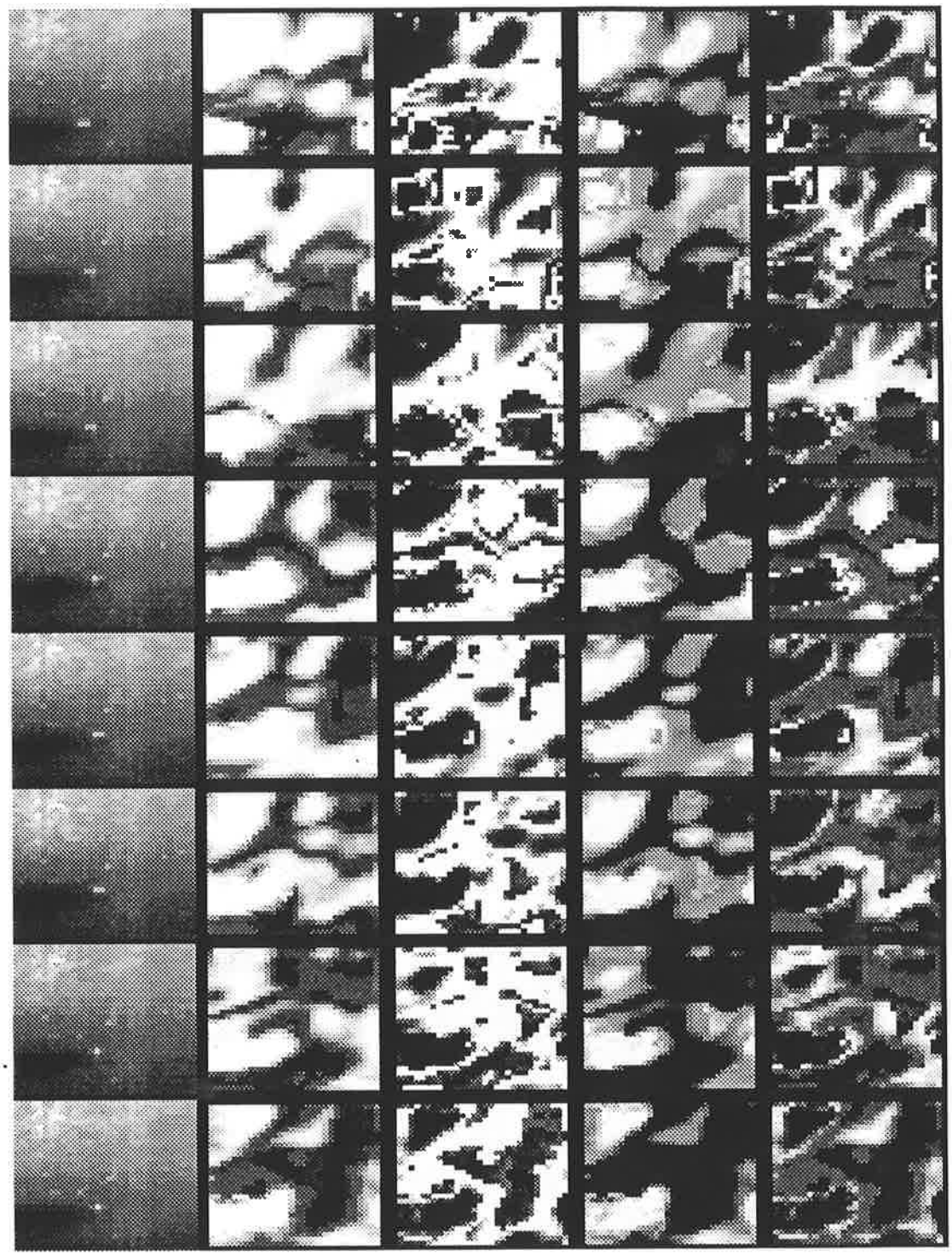

Figure 7.34: Real data, range R, (input, upper probability and lower probability, doubt, believability, see Figure 7.6)

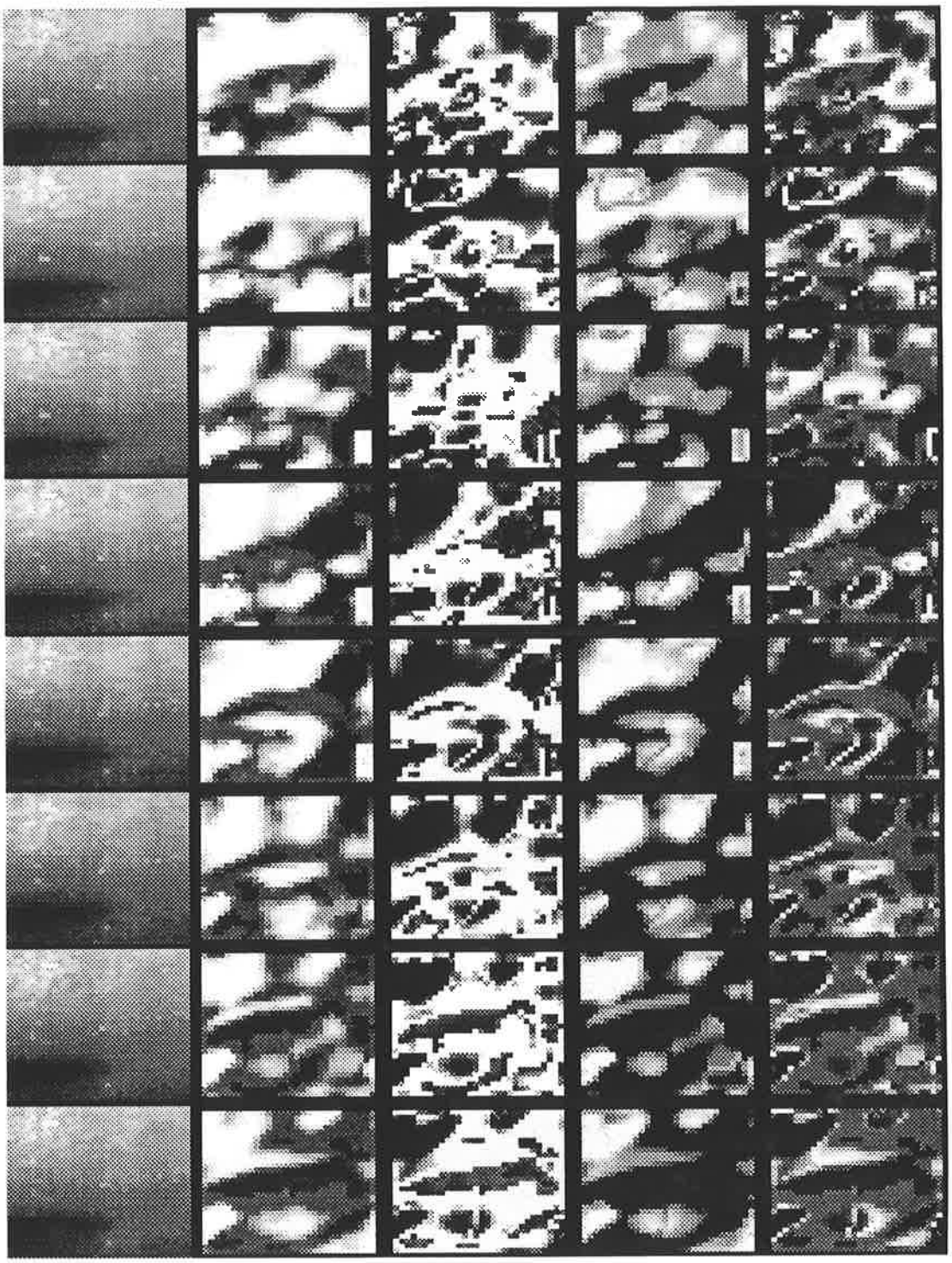

Figure 7.35: Real data, range $2R$, (input, upper probability and lower probability, doubt, believability, see Figure 7.6)

7.6.2 Sonar Data

As a test of the versatility of the approach for use with different sensor types simulation was performed on some measured sonar data. The information was available in the form of a time sequence of FRAZ diagrams which are plots of frequency verses azimuth. The first FRAZ diagram is shown in Figure 7.36. The intensity in these diagrams represents the power of the return at the particular azimuth and frequency. The information I received for simulations was quantised to 255 log power levels, so processing was forced to process log powers to avoid any further information loss by converting to power and then discretising again. The information in Figure 7.36 is power, all other processing was performed on log power information. Due to the amount of information contained in a FRAZ diagram and the computational load of the tracking algorithm, tracking was performed on individual frequency lines rather than over the full spectrum. Tracking was performed on two different frequencies which were significantly different and the results fused together as if they were different sensors, thus allowing correlated noise associated at each frequency to be removed.

The result of processing log power is that noise characteristics are modified extensively, yet this has not been allowed for. The only change made to the algorithm for this problem was a change in the dimension of the tracker, from the three dimensional tracker used in the IR simulations to a two dimension tracker for each frequency line. The features used for object hypothesis were originally left unchanged; however this actually created target hypothesis next to the targets rather than at the correct location so the trained centers were modified. The new centers were created by assuming that bright pixels corresponded to targets, which is a reasonable but incorrect assumption and the resultant training is far from perfect. The training distributions used for each of the classes are Gaussian functions with the distributions shown in Table 7.1 Despite the lack of fine tuning, the results show that the tracker does detect the objects. Figures 7.37 and 7.40 show the sequence of inputs presented to each of the trackers. Figures 7.38 and 7.41 show the conclusions about object presence from each of the frequency trackers, while Figure 7.43 shows the combined conclusions formed by fusing the hypothesis from both frequency trackers. Only the upper probabilities are shown since the information from the lower probabilities is meaningless. This is caused

Class	Velocity Contrast mean,variance	Divergence mean,variance
target	0.0,0.00032	0.03,0.006
background	0.003,0.006 0.0,0.00000001	0.025,0.025 0.0,0.00071
nul	0.003,0.006	0.025,0.025

Table 7.1: Assumed parameters of Gaussian distribution of training points used in tracking simulation of sonar data.

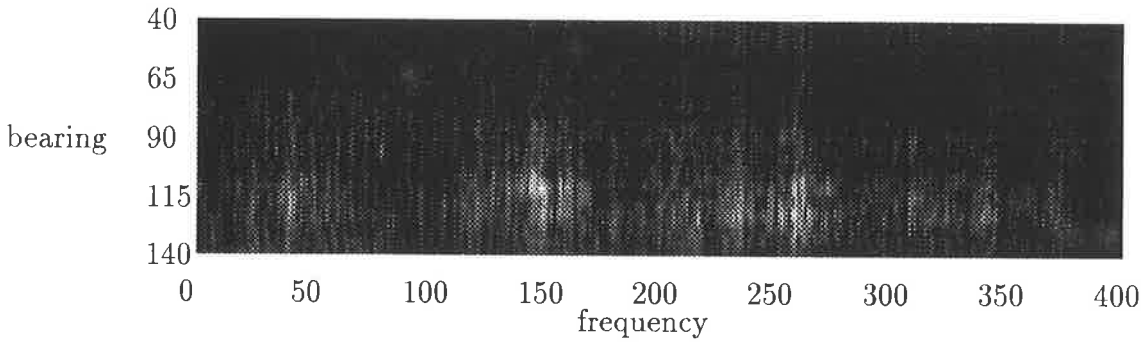

Figure 7.36: Fraz diagram representing the return signal of a sonar sensor at time zero

by the lack of tuning in the hypothesis features so that very little evidence is implicitly associated with a single hypothesis.

7.7 References

Catlett, C. E. (1992). National efforts. *IEEE Spectrum*, *29*(9), 42–55.

Drakopoulos, E. & Lee, C.-C. (1992). Decision rules for distributed decision networks with uncertainties. *IEEE Transactions on Automatic Control*, *37*(1), 5–14.

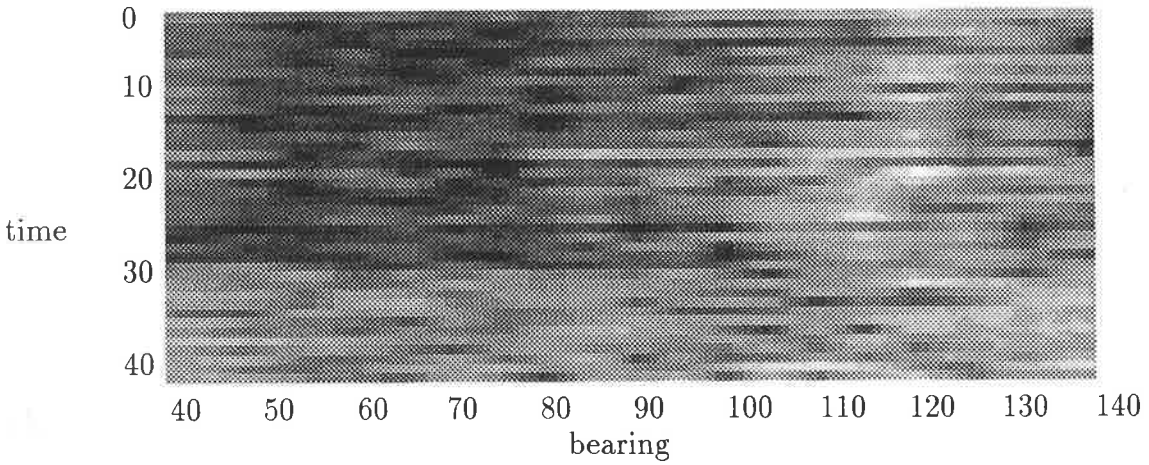

Figure 7.37: Input (log power) to the network in frequency bin 200

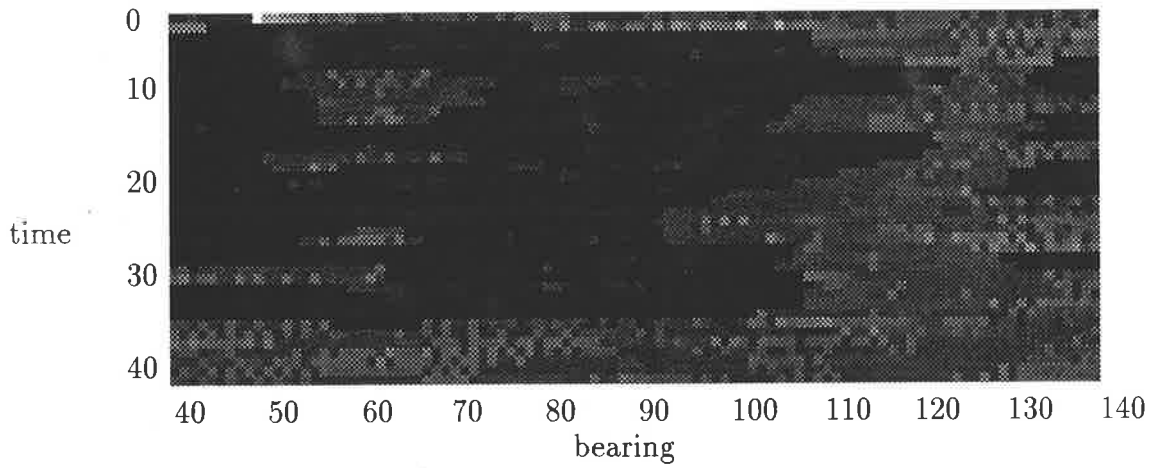

Figure 7.38: Upper probability generated by tracker using information from frequency bin 200

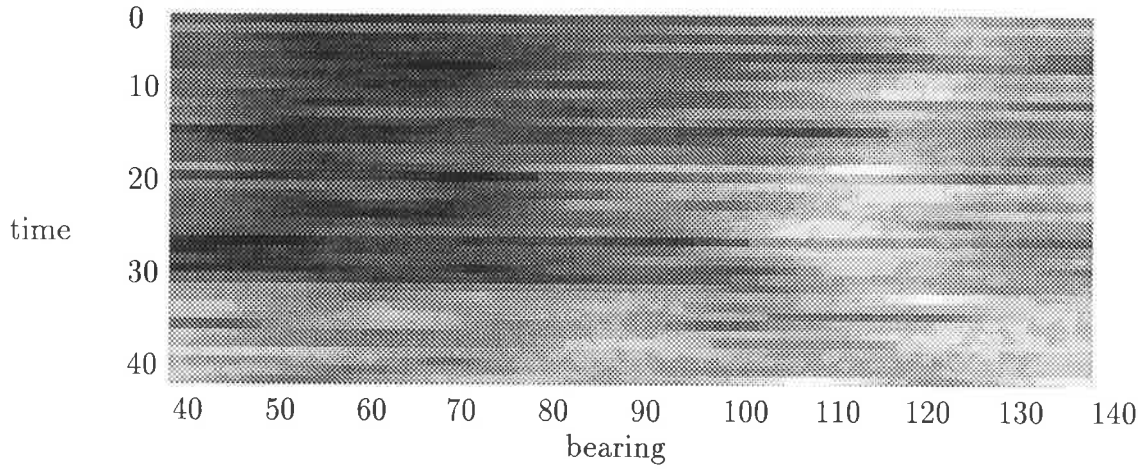

Figure 7.39: Expected signal using information in frequency bin 200

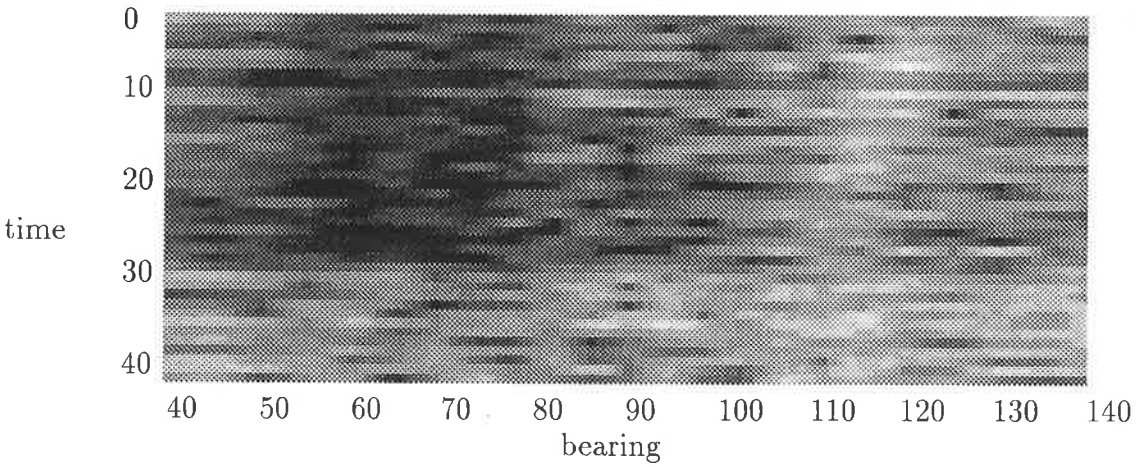

Figure 7.40: Input (log power) to the network in frequency bin 300

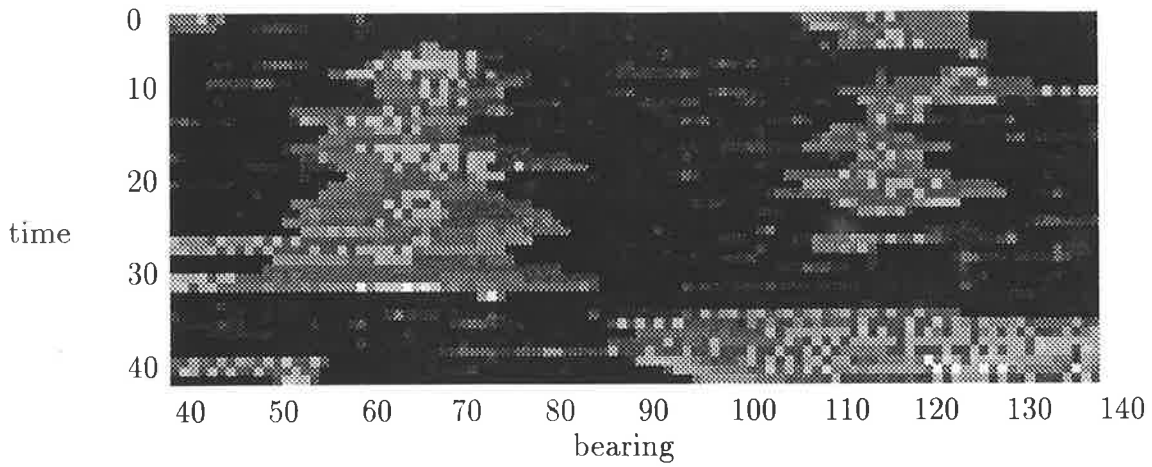

Figure 7.41: Upper probability generated by tracker using information from frequency bin 300

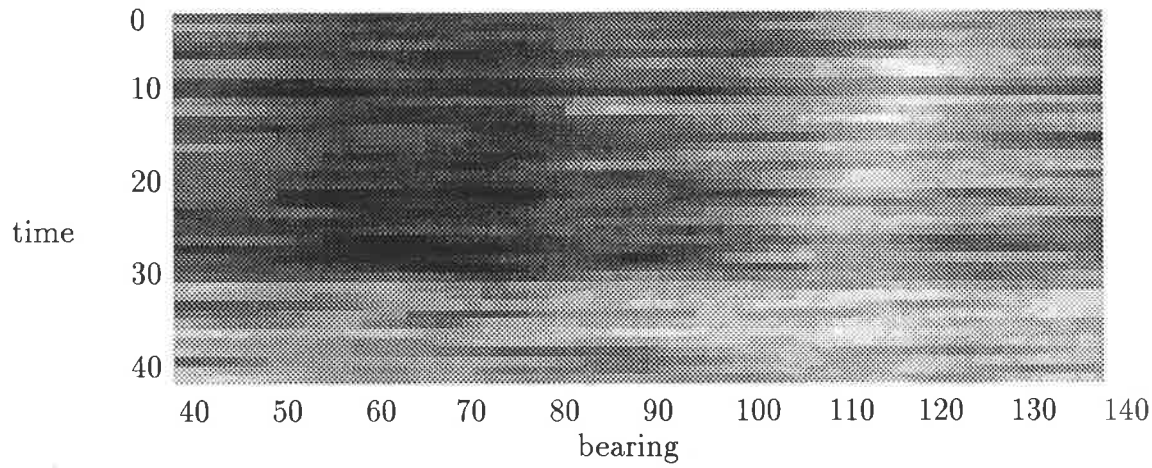

Figure 7.42: Expected signal using information in frequency bin 300

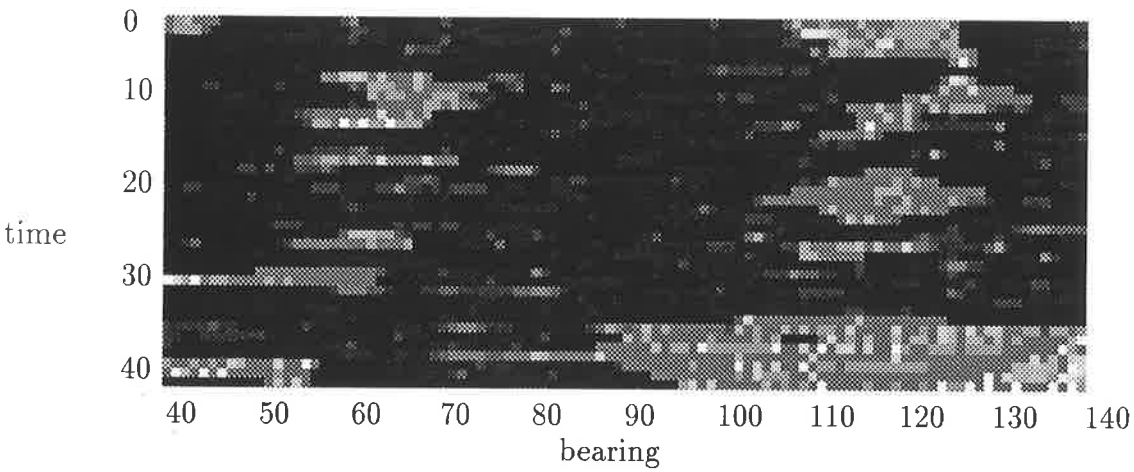

Figure 7.43: Upper probability generated by the fusion of information from trackers using information from frequency bin 200 and 300

Chapter 8

Conclusion and Future Directions

8.1 Conclusion

The problem of tracking unresolvable objects has been addressed in this thesis. Because these objects are unresolved they are difficult to distinguish from the background and consequently conventional approaches which label each point such as MHT and PDA have difficulty associating points from one time instant to the next. The approach presented in this thesis is designed for a parallel architecture to reduce the computational time involved in processing after each sensor update. Because objects cannot be selected using their features Chapter 4 presents a procedure which models the evolution of the signal without knowledge of the objects in the sensor scene. The model for the signal is a restricted statistical mapping and is determined in an iterative way so that objects which move through the image consistently over a period of time are modeled well, while temporally inconsistent objects such as noise are modeled poorly. From the model of the evolution of the signal, information about the motion of objects can be determined.

The model of signal evolution is used to classify the nature of each point in the input field. The classification is a vague process relying on incomplete prior knowledge (training) to perform the classification. Therefore to avoid jumping to conclusions when inadequate information is available a parallel procedure is presented in Chapter 3 which represents the decision with a measure of uncertainty. This representation of uncertainty allows information from multiple sensors to be combined consistently when the information is supportive, conflicting or vacuous.

The results of fusion of the information from the multiple sensors is fed back to each of the sensors to bias the adaptive signal model to the global model which reduces the model's sensitivity to coherent noise (if the sensors are located at different positions or are fundamentally different.) The feedback of information to bias early processing results in a network which is potentially unstable. The analysis of this possible instability occurs throughout the thesis and is finally brought together in Chapter 5 where the stability of the coupled system and the performance benefits are considered.

A modular network has been presented to track distant objects using noisy signals from multiple sensors. The true value of the approach is difficult to determine due to the scarcity of real information, and the lack of an ideal computer of adequate size and architecture to perform the calculations. The motivation of the approach was to derive a solution to the tracking problem which avoided some of the assumptions made using conventional approaches and this has indeed been done, but the solutions from the tracker no longer have any certainty and are really only probabilistic descriptions of target location and motion. This means that comparison with conventional approaches can at best be qualitative; however the question remains, is this approach better than conventional approaches? Although the appropriate theory and procedures have been developed to construct and evaluate the performance it has not yet been practicable to demonstrate conclusively the advantages of the approach over other procedures. I would prefer to say that the approaches are different. The approach presented in this document makes some tradeoffs which allow it to work in an environment where conventional algorithms are impractical. In situations where conventional approaches are indeed practical then the tradeoffs in my approach mean that conventional approaches are indeed superior.

8.2 Future Directions

In a practical application this approach must be improved, and there are a number of improvements which could be incorporated. Target models throughout this document are simple and improvements could be made by determining more accurate models thus improving the velocity predictions and consequently all the associated hypothesis. A conventional tracker should also be incorporated onto

the top level of the network so that tracks can be determined and labeled allowing higher level knowledge of target characteristics and consequently reinforcing some tracks and suppressing others. The addition of a conventional tracker means that possible target points, obtained from hypotheses on the optical flow, can be presented to a conventional tracker, so that the network presented in this thesis merely selects points of interest. The incorporation of such a tracker would also aid in the detection process because the resultant tracks can generate expectations which can be fused back into the overall network to bias the velocity estimations, since the track information is indeed another source of information.

Comprehensive and rigorous testing will require a means of measuring performance against other approaches. The probabilistic answers derived using the approach presented here are difficult to compare to conventional answers because of the representation of doubt. The representation difficulty can be resolved by making a decision from the probabilistic information. The doubt can be incorporated in the decision process by associating costs with correct and incorrect answers, so that when doubt exists the decision can be biased to the decision with the lowest cost. These costs are likely to vary from application to application and the performance benefits of the algorithm will also vary along with the costs so the value of this testing procedure is questionable.

The hypothesis determination network should be trained for each problem and in the case of real data this training information is difficult to obtain. To evaluate the true performance of the approach in a real application sufficient data with known classification must be obtained.

Appendix A

Convergence of Optical Flow Determination

This Appendix contains the proof of the optical flow determination algorithm presented in Chapter 4. The proof shows that the smoothed estimate of the scene $\overline{\varphi}$ converges to the true scene φ.

Given the true state of the matrix φ_t and knowledge of how the targets are truly moving it is possible to calculate the distance between the calculated and actual matrix in a probabilistic sense. It is also possible to compare the calculated transformation with the desired transformation. To evaluate the performance of the technique it is necessary to represent the true signal, and the noisy input in the augmented state space. Since the space is smooth, the noisy signal will represent a connected region (assuming that the distribution of the noise is continuous) about the true signal. To show convergence it is necessary to determine that $\overline{\varphi}(t)$ converges to $\varphi(t)$. The components of the distance between the required value of a vector $\varphi(t)$ and the estimated vector $\overline{\varphi}(t)$ can be expressed as

$$\varphi_\alpha(t) - \overline{\varphi}_\alpha(t) = (1-\epsilon)n_o + \epsilon\triangle_\alpha(t) \tag{A.1}$$

where $\triangle_\alpha(t)$ is the prediction error given by

$$\triangle(t) = \varphi(t) - \hat{\varphi}(t). \tag{A.2}$$

The convergence of this algorithm can be shown in a statistical sense so that the

first and second moments of the error between $\varphi(t)$ and $\overline{\varphi}(t)$ are given by

$$\overline{X}^{e}_{\varphi_\alpha(t)} = \mathrm{E}\left[\varphi_\alpha(t) - \overline{\varphi}_\alpha(t)\right] \tag{A.3}$$

$$= \mathrm{E}\left[(1-\epsilon)n_o + \epsilon\triangle_\alpha(t)\right] \tag{A.4}$$

$$\text{and } \overline{X}^{2e}_{\varphi_\alpha(t)\varphi_\alpha(t)} = \mathrm{E}\left[(\varphi_\alpha(t) - \overline{\varphi}_\alpha(t))^2\right] \tag{A.5}$$

$$= \mathrm{E}\left[(1-\epsilon)^2(n_o)^2 + \epsilon^2(\triangle_\alpha(t))^2\right]. \tag{A.6}$$

The error expressed by $\triangle_\alpha(t)$ is a function of the magnitude of the previous prediction errors, $\triangle(t-1)$, and uncertainty in the target manoeuvres which are modeled as a noise, n_{T_t}. We can bound $\triangle_\alpha(t)$ above so that

$$\triangle_\alpha(t) \leq k\left\|\overline{X}^{e}_{\varphi_\alpha(t-1)}\right\| + n_{T_t}. \tag{A.7}$$

When the expectation is taken with the assumption that $\|\overline{X}^{e}_{\varphi_\alpha(t-1)}\|$ and n_{T_t} are uncorrelated, which is usually valid since they are caused by different physical systems, the following identities can be derived:

$$\overline{X}_{\triangle_\alpha(t)} \leq k\|\overline{X}^{e}_{\varphi(t-1)}\| + \overline{X}_{n_{T_t}} \text{ and} \tag{A.8}$$

$$\overline{X}^{2}_{\triangle_\alpha(t)\triangle_\alpha(t)} \leq k^2\|\overline{X}^{2e}_{\varphi(t-1)\varphi(t-1)} + \overline{X}^{2}_{n_{T_t}n_{T_t}}. \tag{A.9}$$

Substituting (A.8), and assuming equality since this represents the worst case, into (A.4) the following expression is obtained after some rearranging,

$$\frac{\|\overline{X}^{e}_{\varphi(t)}\|}{\|\overline{X}^{e}_{\varphi(t-1)}\|} = \frac{(1-\epsilon)\overline{X}_{n_o} + \epsilon\overline{X}_{n_{T_t}}}{\|\overline{X}^{e}_{\varphi(t-1)}\|} + \epsilon k. \tag{A.10}$$

For a stable point

$$\frac{\|\overline{X}^{e}_{\varphi(t)}\|}{\|\overline{X}^{e}_{\varphi(t-1)}\|} = 1, \tag{A.11}$$

therefore the expected error equilibrium moment, $\|\overline{X}^{equil}_{\varphi}\|$, is given by

$$\|\overline{X}^{equil}_{\varphi}\| = \frac{(1-\epsilon)\overline{X}_{n_o} + \epsilon\overline{X}_{n_{T_t}}}{1-\epsilon k}. \tag{A.12}$$

The equilibrium point can be shown to be conditionally stable as follows. Consider the sequence of $\|\overline{X}^{e}_{\varphi(t)}\|$ expressed by rearranging (A.10) to

$$\|\overline{X}^{e}_{\varphi(t)}\| = (1-\epsilon)\overline{X}_{n_o} + \epsilon\overline{X}_{n_{T_t}} + \epsilon k\|\overline{X}^{e}_{\varphi(t-1)}\|. \tag{A.13}$$

(A.13) can be differentiated so that

$$\frac{\mathrm{d}\|\overline{X}^{e}_{\varphi(t)}\|}{\mathrm{d}\|\overline{X}^{e}_{\varphi(t-1)}\|} = \epsilon k. \tag{A.14}$$

The sequence is convergent if $-1 < \frac{d\|\overline{X}^e_{\varphi(t)}\|}{d\|\overline{X}^e_{\varphi(t-1)}\|} < 1$ (Moon, ; Andronov et al., 1966). $\epsilon k \geq 0$ by definition and so it needs to be shown that $\epsilon k < 1$ for convergence to be guaranteed.

For the second moment, substituting (A.9), and assuming equality since this represents the worst case, into (A.6) the following expression is obtained after some rearranging,

$$\frac{\|\overline{X}^{2e}_{\varphi(t)\varphi(t)}\|}{\|\overline{X}^{2e}_{\varphi(t-1)\varphi(t-1)}\|} = \frac{(1-\epsilon)^2\overline{X}^2_{n_o n_o} + \epsilon^2\overline{X}^2_{n_{T_t} n_{T_t}}}{\|\overline{X}^{2e}_{\varphi(t-1)\varphi(t-1)}\|} + \epsilon^2 k^2. \tag{A.15}$$

For a stable point

$$\frac{\|\overline{X}^{2e}_{\varphi(t)\varphi(t)}\|}{\|\overline{X}^{2e}_{\varphi(t-1)\varphi(t-1)}\|} = 1, \tag{A.16}$$

therefore the expected error equilibrium moment,

$$\|\overline{X}^{2equil}_{\varphi\varphi}\| = \frac{(1-2\epsilon+\epsilon^2)\overline{X}^2_{n_o n_o} + \epsilon^2\overline{X}^2_{n_{T_t} n_{T_t}}}{1-\epsilon^2 k^2}. \tag{A.17}$$

This equilibrium point can be shown to be stable in the same manner as was used for the first moment of the error. $\|\overline{X}^{2equil}_{\varphi\varphi}\|$ is a stable equilibrium if $-1 < \epsilon^2 k^2 < 1$. To show convergence it is therefore necessary to show that $\epsilon k < 1$.

The problem with these convergence proofs is that there is a singularity at the point where $\epsilon k = 1$. ϵ is a defined constant whiile k is a measure of the convergence of the prediction error to 0 (A.7). The analysis of this convergence is complicated and is affected by the blurring of objects with in the sensor field. The shape of this blurring function will determine the rate at which convergence is obtained.

Assuming a linear approximation to the blurring function around the equilibrium point, so that the effect of the blurring results in a linear relationship between the mapping error, $\|\overline{X}_{\triangle_\alpha(t)}\|$ and the predicted scene error, $\|\overline{X}^e_{\varphi(t)}\|$. Substituting (A.4) into (A.7) and taking the magnitude gives

$$\|\triangle_\alpha(t+1)\| = \epsilon\|\triangle_\alpha(t)\| + n_{T_t} + (1-\epsilon)n_o. \tag{A.18}$$

Figure A.1 shows this pictorially by plotting the separation between the matricies (images) in a state space representation as the matricies evolve. The

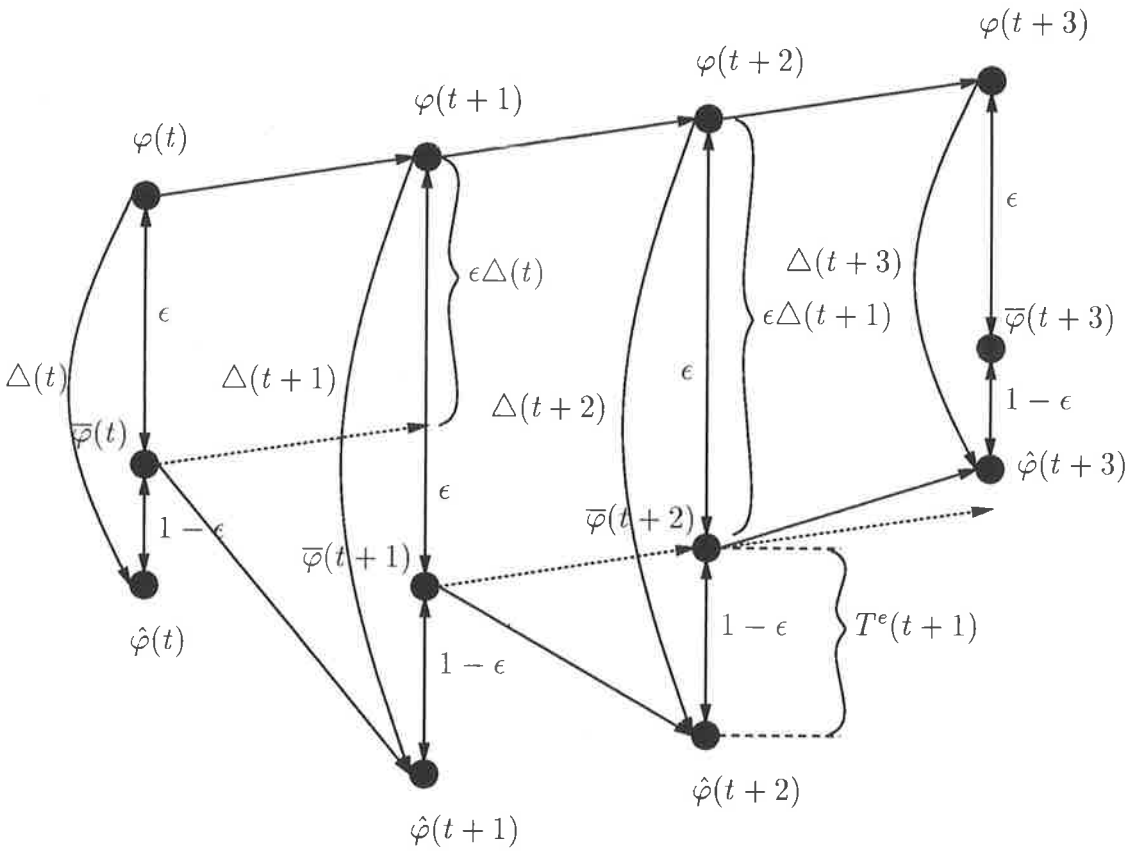

Figure A.1: Pictorial representation of the mapping function and its errors in the state space

representation in the state space is simplified from a point in an nR dimensional space to a point in a one dimensional space which represents the distance between the points in the nR dimensional space. The evolution of the states is shown progressing to the right under the influence of the mapping functions to show the relationship between the errors and how they evolve. $T^e(t)$ represents the error in the mapping parameters T. This error can be bounded so that $T^e(t) \in [\epsilon T^e(t-1), T^e(t-1)]$ These bounds correspond respectively to the situation where the mapping function is corrected perfectly to the current input sequence or is not modified at all. Using these limits and (A.18) it is possible to determine the error with respect to an initial error $T^e(0)$.

For perfect correction, we have $T^e(t) = \epsilon T^e(t-1)$,

$$\begin{aligned}
\triangle(t+1) &= \epsilon\triangle(t) + T^e(t) + (1-\epsilon)n_o \\
&= \epsilon^2\triangle(t-1) + 2\epsilon T^e(t) + (\epsilon+1)(1-\epsilon)n_o \\
&= \epsilon^3\triangle(t-2) + 3\epsilon^2 T^e(t) + (\epsilon^2+\epsilon+1)(1-\epsilon)n_o \\
&\vdots \\
&= \epsilon^{n+1}\triangle(t-n) + (n+1)\epsilon^n T^e(t) \\
&\quad + (\epsilon^n + \epsilon^{n-1} + \cdots + 1)(1-\epsilon)n_o \\
&\to_{n\to\infty} 0 + 0 + n_o.
\end{aligned}$$

For $T^e(t) = T^e(t-1)$, which represents a situation where mapping errors are not corrected,

$$\begin{aligned}
\triangle(t+1) &= \epsilon\triangle(t) + T^e(t) + (1-\epsilon)n_o \\
&= \epsilon^2\triangle(t-1) + (\epsilon+1)T^e(t) + (\epsilon+1)(1-\epsilon)n_o \\
&= \epsilon^3\triangle(t-2) + (\epsilon^2+\epsilon+1)T^e(t) \\
&\quad + (\epsilon^2+\epsilon+1)(1-\epsilon)n_o \\
&\vdots \\
&= \epsilon^{n+1}\triangle(t-n) + (\epsilon^n + \epsilon^{n-1} + \cdots + 1)T^e(t) \\
&\quad + (\epsilon^n + \epsilon^{n-1} + \cdots + 1)(1-\epsilon)n_o \\
&\to_{n\to\infty} 0 + \frac{1}{1-\epsilon}T^e(t) + n_o.
\end{aligned}$$

The result of these equations is that in the limiting case

$$T^e(t+1) \in \left[n_o, T^e(0)\frac{1}{1-\epsilon} + n_o\right].$$

The right-hand limit of $T^e(t+1)$ is the situation where no correction for mapping errors occurs, while the left hand limit is the situation where perfect correction is made for the sequence of $\overline{\varphi}$s. The true situation lies somewhere in between these cases. This demonstrates that

$$\begin{aligned}
\overline{X}_{\triangle_\alpha(t)} &\le \overline{X}_{T(0)} + k^n \overline{X}^e_{\varphi_\alpha(0)} \\
&\to_{n\to\infty} \overline{X}^e_{T(0)}\frac{1}{1-\epsilon}.
\end{aligned}$$

Since $\overline{X}^e_{\varphi_\alpha(0)}$ does not equal zero, $k \le 1$ so that $\epsilon k \le \epsilon < 1$ and the singularity in the expressions for the equilibrium point of the error does not occur. Under the assumptions given the algorithm is stable.

The assumption that the blurring function causes a linear relationship between the mapping error, $\overline{X}_{\triangle_\alpha(t)}$, and the pixel error, $\overline{X}^e_{\varphi_\alpha(t)}$, is only true for extremely small errors. In fact there are multiple possible equilibria, all of which are stable, but which exist for only finite periods of time due to the temporal inconsistency of any tracks generated by noise processes. Since these equilibrium states degenerate, tracks will not become trapped here.

The question remains, will the tracks converge correctly once a false equilibrium degenerates. Tracks are always formed in the vicinity of the correct track since mappings are attached to every discrete point in the space, so that after a false track degenerates, formation of the correct track is always possible. The true track does not degenerate so that given a probability, α of forming the correct track after a false track degenerates, it is possible to determine the probability of finding the correct track after N tracks have degenerated.

$$\begin{aligned} P(true) &= \alpha + \alpha(1-\alpha) + \alpha(1-\alpha)^2 + \cdots + \alpha(1-\alpha)^N \\ &= 1 - (1-\alpha)^{N+1} \\ &\to 1 \text{ as } N \to \infty. \end{aligned}$$

A.1 Interpretation of Convergence Proof

The convergence proof determines the convergent point for the algorithm; however the terms in the solution do not have any apparent connection to the sensor and scene characteristics. This section establishes a connection between the parameters in the convergence proof and the sensor and scene characteristics.

It is possible to represent $\overline{X_n}$, $\overline{X^2_{nn}}$, $\overline{X_{n_{T_t}}}$ and $\overline{X^2_{n_{T_t}n_{T_t}}}$ in terms of parameters which have significance to the measurement system. Consider the situation where the sensor point spread function, $G_c(\alpha)$, distributes the probability from the true position, c, to α. It shall be assumed that

$$G_c(\alpha) = \delta(\alpha - c). \tag{A.19}$$

Consequently

$$\overline{X_{n_\alpha}} = \sum_{\forall x} p_n(x)|G_{c+x}(\alpha) - G_c(\alpha)| \text{ where}$$

$$p_n(x) = \int_{x-\frac{1}{2R}}^{x+\frac{1}{2R}} \frac{1}{\sqrt{2\pi}\sigma_n} e^{-(\mu_n - y)^2/2\sigma_n^2} dy.$$

μ_n and σ_n are the mean and variance of the noise process in the real world. Using (A.19),

$$\overline{X}_{n_\alpha} = \overline{X}^2_{n_\alpha n_\alpha} = \begin{cases} p_n(\alpha - c), & \text{for } \alpha \neq c \\ 1 - p_n(0), & \text{for } \alpha = c, \end{cases}$$

$$\text{thus } \sum_{\alpha=1}^{nR} \overline{X_{n_\alpha}} = \sum_{\alpha=1}^{nR} \overline{X^2_{n_\alpha n_\alpha}} = 2n - 2np_n(0).$$

In the case of the prediction error, the objective distribution can once again be represented by $G_c(\alpha)$. The actual mapping error is gaussian due to the model defined for the transformation. Thus define the distribution

$$H_c(\alpha) = \int_{x-\frac{1}{2R}}^{x+\frac{1}{2R}} \frac{1}{\sqrt{2\pi}T^\sigma} e^{-(\alpha-(c+y))^2/2(T^\sigma)^2} \mathrm{d}y. \tag{A.20}$$

Now

$$\overline{X}_{n_{T_{t,\alpha}}} = \sum_{\forall x} p_T(x)|H_{c+x}(\alpha) - G_c(\alpha)| \text{ and}$$

$$\overline{X}^2_{n_{T_{t,\alpha}} n_{T_{t,\alpha}}} = \sum_{\forall x} p_T(x)|H_{c+x}(\alpha) - G_c(\alpha)|,$$

$$\text{where } p_T(x) = \int_{x-\frac{1}{2R}}^{x+\frac{1}{2R}} \frac{1}{\sqrt{2\pi}\sigma_T} e^{-(\mu_T - y)^2/2\sigma_T^2} \mathrm{d}y.$$

There are two parameters, $\overline{X}_{\varphi_\alpha(t)}$ and $\overline{X}^2_{\varphi_\alpha(t)\varphi_\alpha(t)}$, which have not yet been given any physical meaning. They represent the true scene statistics; however they are defined over the augmented state space, so that they are not simply the first and second moments of the scene but the first and second moments of the evolution of the multinomial distribution representing the scene, $\varphi_\alpha(t)$.

A.2 References

Andronov, A. A., Vitt, A. A., & Khaikin, S. E. (1966). *Theory of Oscillators.* New York: New York.

Moon, F. C. *Chaotic Vibrations.* John Wiley & Sons.

Appendix B

Initiation, Tracking and Association

B.1 Introduction

The initiation and association of tracks are readily acknowledged as difficult tasks. Initiation is the problem of determining the existence of a track given the information from a sequence of sensor outputs. Association is the problem of assigning information from sensors to current tracks. This chapter contains a very brief and incomplete summary of some of the more common approaches to the problem. For the purposes of this chapter it shall be assumed that the information has been filtered and then thresholded so that the returns from the sensors are points, representing possible object locations.

The review of this information has been limited since the size of the available literature is enormous, and features of the problem and the methodology chosen to tackle these problems mean that the information is of dubious usefulness. The main motivation for avoiding the approaches presented in this literature is the computational expense of scaling these algorithms to deal with the level of noise or false detections which are prevalent in the problem description.

B.2 Initiation

B.2.1 Basic Approach

A box is generated about a return point based on velocity limits (figure B.1), if in the following sensor acquisition a new point appears in the box then these two points will constitute the start of a track. Only initiate a track if there is one point in the box. This approach has numerous limitations(Evans & Mareels,

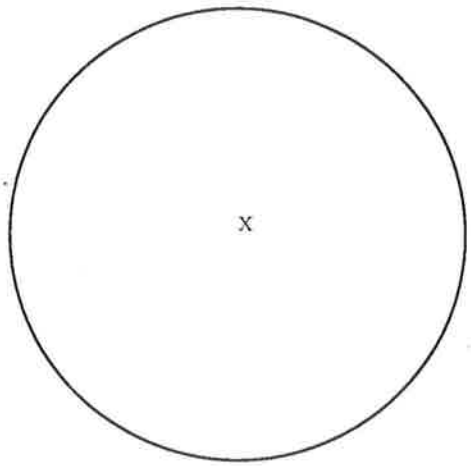

Figure B.1: Correlation box for next sample using the basic approach to track initiation

1990)

- generation of false tracks from noise
- missed tracks when multiple tracks or noise cause multiple detections within a box.

Several alternative approaches are based on similar reasoning, with modifications to remove some of the limitations:

B.2.2 Transition Matrix

This approach is an improvement on the previous approach because it decreases the chance of spurious track generation. States are assigned to each point as either hit or miss, along with an index. This enables schemes to be developed such that multiple hits are required to establish a track. As an example, 4 successive hits might be required on a particular path to establish a confirmed track, rather than the one hit required in the basic approach. Refer figure B.2(Evans & Mareels, 1990)

B.2.3 Staircase Integrator

This procedure simply counts up and down depending on whether a point was detected in the appropriate box.

$$count = count + \alpha \text{ if hit,}$$

$$count = count - \beta \text{ if miss.}$$

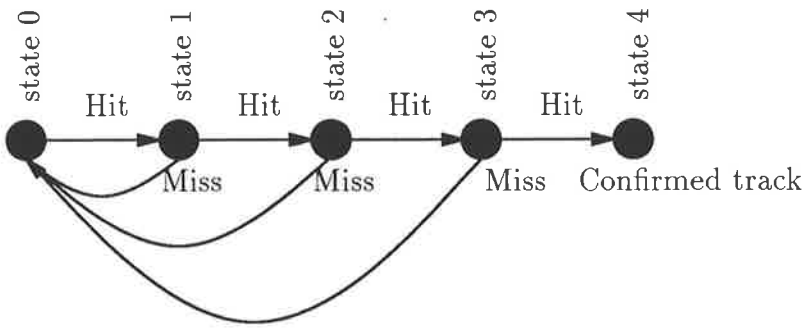

Figure B.2: State transition diagram for the transition matrix approach to track initiation

When *count* becomes larger than a certain number a track is established. If *count* is less than zero then the path is not a track.(Evans & Mareels, 1990)

B.3 Tracking

The process of tracking involves the determination of the motion characteristics of an object from a series of observations. These observations are generally noisy so that the determination of the motion parameters is essentially a random variable estimation problem. The approach which has found the greatest use for this type of problem is the Kalman filter.

There are two ways in which the information observed by sensors can be tracked.

- The sensor information points to the existence of an object at a particular location so that a label may be assigned to that point in space (ie. range and azimuth for a radar system or azimuth and elevation for an IR sensor.) The location of this label in space is then tracked by a process such as Kalman filtering. Since objects in the sensor field are indistinguishable from the background, in the application dealt with in this thesis, assigning the existence of an object at a particularly point is impossible.

- The information from the sensor varies through time so that it is possible to track the evolution of the sensor signal components. This does not assign tracks to individual objects in the scene but filters noise and providing the evolution model has sufficient parameters will determine the motion

of objects in the scene, just as the network presented in this thesis claims to do. Unfortunately the Kalman filter is linear in nature[1] which makes it inadequate for the problem tackled in this thesis. The manner in which objects can move from one pixel to another, which can result in a step change in a signal component as a function of the object's motion parameters (see example 8,) means that a linear assumption about signal evolution is inadequate.

Example 8 *Consider a simple system with three pixels arranged in a line with the object in the center pixel such that a position vector representation is given by*

$$p = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{B.1}$$

If the motion of the object is downwards then the motion vector is

$$m = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{B.2}$$

If the motion of the object is upwards then the motion vector is

$$m = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}. \tag{B.3}$$

The evolution equation for the first item of the position vector is a nonlinear function of the second item's motion parameter, since for the first item of the position vector to change the second item's motion parameter must be negative. This is true even if, as pointed out in Chapter 4, the object is not located at a single pixel but is distributed to its neighbours due to noise processes during transmission through the medium (unless the blurring function is a linear function of the distance from the center which is unlikely.)

A derivation of the Kalman filter is given in many texts, such as Anderson & Moore (1979) and Shet & Rao (1985). The resultant equations are

$$x_{k+1} = Fx_k + Gw_k$$

$$\text{and } y_k = Hx_k + v_k,$$

[1]Even the extended Kalman filter relies on linearising using Taylor's expansion about the current point.

where x is the target state vector, y is the measurement vector and w and v represent the process and measurement noise respectively. It is usually assumed that w and v are zero mean mutually dependent white Gaussian noise vectors with covariance matrices given by Q and R respectively. k is the discrete time index.

The technique relies on the conditional probability density $\Pr_{X|y}$ which is derived using Bayes' Rule(Anderson & Moore, 1979) (B.6)

$$\text{If } \Pr(B) \neq 0, \Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}. \tag{B.4}$$

$$\text{If } A_i, i = 1, 2, \ldots, n \text{ are mutually disjoint and } \cup A_i = \Omega, \tag{B.5}$$

$$\Pr(A_j|B) = \frac{\Pr(B|A)\Pr(A_j)}{\sum_i \Pr(B|A_i)\Pr(A_i)}. \tag{B.6}$$

The result is that

$$E\{x|y\} = E\{x\} + \sigma_{xy}\sigma_{yy}^{-1}(y - E\{y\})$$
$$\text{and } \sigma_{x|y} = \sigma_{xx} - \sigma_{xy}\sigma_{yy}^{-1}\sigma_{yx},$$

where σ_{ab} represents the covariance between a and b.

The trackers estimate of the target state x_k at time k, given data up to time i is designated by $\hat{x}_{k|i}$ and similarly for $\hat{y}_{k|i}$, the error in the estimate is given by $\tilde{x}_{k|i} \triangleq x_k - \hat{x}_{k|i}$ with error covariance matrix $P_{k|i} \triangleq E\{x_{k|i}\tilde{x}_{k|i}\}$, where E denotes expectation. Using the discrete time Kalman-Bucy filter the state and covariance estimate become

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + W_k\tilde{y}_k \tag{B.7}$$

$$= F\hat{x}_{k-1|k-1} + W_k\tilde{y}_k, \tag{B.8}$$

$$\text{and } P_{k|k} = P_{k|k-1} - W_kS_kW_k' \tag{B.9}$$

$$= FP_{k-1|k-1}F' + GQG' - W_kS_kW_k', \tag{B.10}$$

respectively, where the innovation vector,

$$\tilde{y}_k \triangleq y_k - \hat{y}_{k|k-1},$$

has the covariance matrix

$$S_k \triangleq E\{\hat{y}_k\hat{y}_k'\}$$
$$= HP_{k|k-1}H' + R$$

and the filter gain matrix is

$$W_k = P_{k|k-1} H' S_k^{-1}.$$

The resulting state estimate, under the above assumptions is the conditional mean

$$\hat{x}_{k|k} = E\{x_k | Y^k\},$$

where Y^k denotes the set of all vectors y_i for $i \leq k$. The m candidate measurements at time k will be denoted as $y_j, j = 1, \ldots, m$ ie.

$$Y^k = \{y_1, \ldots, y_m\} \cup Y^{k-1},$$

and the corresponding innovations are

$$\tilde{y}_j \triangleq y_j - \hat{y}, \quad j = 1, \ldots, m.$$

B.4 Association

Once a track has been established there needs to be a mechanism for deciding whether future points belong to a particular track or are the result of some noise process. If the point is deemed to belong to a particular track then the track information can be updated.

B.4.1 Simplex Approach

This approach merely associates each track with the closest point. The approaches major failing is that it is far from optimum, and in a noisy environment will generate almost useless results.

B.4.2 Branching Algorithms

This approach generates a track for each point in the correlation box associated with the previously known track point, and assigns a measure of closeness of this new track to the original. By following the alternative branches a number of possible tracks are generated; however many of these tracks are generated by noise and the measure associated with those tracks becomes small. As the branching process proceeds the unlikely track branches are eliminated to reduce the computational load. As an example consider figure B.3

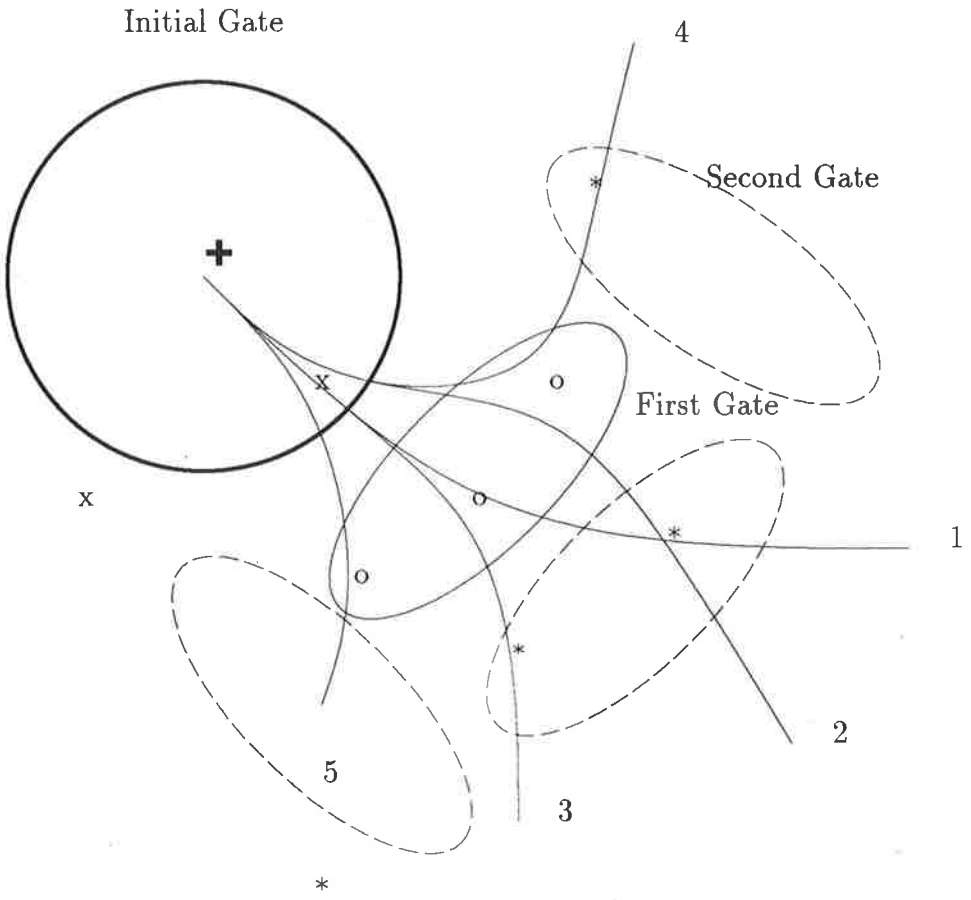

Figure B.3: Multiple hypotheses tracking. Tracks are numbered in order of probability of existence. Track 1 is the most likely, track 5 is not continued.

This approach can become computationally expensive since the branch points may branch again and again leaving lots of potential tracks which must be updated; however it does have the advantage that manoeuvring targets will generally be unable to escape from the tracker.

B.4.3 Joint Probabilistic Data Association

One of the major techniques used in tracking is *Probabilistic Data Association* (PDA), this approach can have limitations in situations with multiple targets and has led to the development of *Joint Probabilistic Data Association* (JPDA)(Fortmann, Bar-Shalom & Scheffe, 1983).

In the PDA approach the estimate of $\hat{x}$ is obtained from (B.8) using the combined innovation,

$$\tilde{y} \triangleq \sum_{j=1}^{m} \beta_j \tilde{y}_j,$$
$$\text{where } \beta_j = P\{\chi_j | Y^k\}, j = 0, 1, \ldots, m.$$

χ_j denotes the event that the jth measurement belongs to a target, and χ_0 the event that there is no detection. β_j is the probability that the jth measurement is the correct one. A full derivation is shown in Fortmann et al. (1983) along with the derivation for β_j.

B.4.4 SME Filter

The Symmetric Measurement Equation (SME) filter attempts to avoid the data association problem "by converting the measurement data into measurement equations by defining new measurements that are symmetric functionals of the original measurements."Lee & Kamen (1993) This approach utilises Kalman filtering to track objects with out requiring an association between data and tracks. It does however require that the number of valid tracks be known and that potential target points can be selected from the sensor returns via thresholding or filtering of some type. In Kamen & Sastry (1993) this approach is shown to perform fusion with no real increase in complexity. The SME approach is superior to conventional PDA filtering since by avoiding the association problem it scales in a polynomial manner rather than a factorial manner.

B.5 Optical Neural Tracking

In many instances tracking information from optical sensors is identical to the conventional tracking used in many radar systems. There are a number of alternative systems which have been developed and which were inspired by the animal visual system.

B.5.1 Visual Pursuit

This approach has been used by many early radar systems, where the intention is to maintain a signal on the tracked object. If the object moves from the center of the observation beam then there is an amplitude reduction in the returned signal. In an optical system inspired from the human visual system the sensor foveates on the object of interest. Any errors in the visual tracking will cause the object to move into the peripheral region of vision, which will indicate the size and direction of the tracking error, and enable correction. Papers by Lisberger et al. (1987) and Deno et al. (1989) explain the operation of such a system; however this approach is not capable of tracking multiple objects simultaneously and so shall not be discussed further. Gerrissen (1991) presents a mechanism of modeling the human visual search mechanism, which enables the tracking of multiple objects; however in the problem tackled in this thesis, human search strategies are far from optimal and are carried out in a serial manner, which is not desirable if rapid solutions are sought and the number of alternatives to search is large.

B.5.2 Motion Perception

Given sequences of images, animals are able to derive an apparent motion. Marshall (1990) shows an approach of generating neural networks for detection of motion given only local information from each sensor. However this relies on the objects of interest having some structure. Other papers such as Foldiak (1991) and Grossberg & Rudd (1989) perform recognition which is invariant to motion, while Marshall (1991) determines shape from motion boundaries. The essential feature of all these papers which I would like to note is that all the motion detection relies on some form of structure imposed on the object to be detected, and as such these and the many other papers on similar topics are not reviewed

further. Bouzerdoum (1993) develops an approach for determination of velocity fields modeled on the insect visual system which does not require the objects of interest to have any structure; however the objects considered are binary in nature and so once again this type of paper is not covered further.

B.5.3 Recurrent Neural Networks

Many papers have been written about *recurrent neural networks* (RNN). A good basic example is Elman (1990). The optical flow determination network I have proposed is a RNN; however to avoid the questions of convergence which can arise when training such a net, I have specifically assigned connection weights. It should be noted that I did not set out to design a RNN; however the solution turns out to fit into the general model. The majority of the literature is concerned with the capabilities of the networks and the training mechanisms. Since I have avoided these considerations an indepth review of the literature is not important.

B.5.4 Multi-Target Tracking using Interpolative Probability Fields

Neural networks have been forced to attempt solutions of almost every problem, and the problems of tracking and association are no different. The idea of using neural networks is founded on the idea that the human brain does this type of task very well; however many of the artificial networks which have been proposed for the problem solution fail in any but the simplest cases. The work of Kuczewski (1987) I believe is noteworthy and is similar to my own in some aspects of its methodology for tracking.

Kuczewski specifies a network which generates what they refer to as an *Interpolative Probability Field* (IPF). In this approach a processing element (PE) represents a point in the sensor feature space. The value of each PE is a probability of that point being included in the global solution. The probabilities are updated as a function of the correlation of all other points in the network with each point.

The network connectivity defines a relationship between neurons i and j which is simply the probability that an object exists at $(x_j, y_j, \theta_j, t_j)$ given that there is an object at $(x_i, y_i, \theta_i, t_i)$. These connections are established from a model of object and noise dynamics. The approach undergoes a competitive interaction

between neurons such that given a particular sensor input a global solution is reached.

The approach I have developed is different to this technique in a number of ways, some of which are motivated by implementation practicality, and others which are due to slightly different detection problems. The first difference is in the level of detection noise, which appears to be absent in the simulated examples. In Kuczewski (1987) there appears to be some track noise but very little noise related to the detections, as distinct from my problem where the existence of the object is the problem. This is not a problem since I believe the network in Kuczewski (1987) is capable of dealing with such situations, given that its assumption, that the number of tracks is constant does not change. My network does not determine track existence, and so such an assumption is not possible, and indeed is insupportable since I am seeking the generation of new tracks, not the maintenance of old ones. The IPF approach could be forced to seek new tracks by looking for one more track than actually exists. The other differences arise due to implementation shortcuts that I have taken to make the network presented in this thesis practical. Firstly the approach in Kuczewski (1987) maintains a history of sensor inputs which are used in the optimisation. This optimisation in time, is time consuming and the storage required to maintain such a large volume of data given the resolution of current sensor technologies is prohibitive. In essence the front end optical flow determination processing I carry out is a one step non competitive version of the approach in Kuczewski (1987), with the additional constraint that I have no knowledge of the targets approximate velocity component, and must derive such components myself from the noisy information.

B.6 References

Anderson, B. D. O. & Moore, J. B. (1979). *Optimal Filtering.* Englewood Cliffs, New Jersey 07632: Prentice Hall.

Bouzerdoum, A. (1993). The elementary movement detection mechanism in insect vision. *Philosophical Transactions of the Royal Society of London, 339*, 375–384.

Deno, D. C., Keller, E. L., & Crandall, W. F. (1989). Dynamical neural network organisation of the visual pursuit system. *IEEE Transactions on Biomedical Engineering*, *36*(1), 85–92.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 170–211.

Evans, R. J. & Mareels, I. M. (1990). *Course notes for Introduction to Tracking.*

Foldiak, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, *3*, 194–200.

Fortmann, T. E., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, *OE-8*(3), 173–184.

Gerrissen, J. F. (1991). On the network-based emulation of human visual search. *Neural Networks*, *4*, 543–564.

Grossberg, S. & Rudd, M. E. (1989). A neural architecture for visual motion perception: Group and element apparent motion. *Neural Networks*, *2*, 421–450.

Kamen, E. W. & Sastry, C. R. (1993). The sme filter approach to sensor fusion in multiple target tracking. In *Society of Photo-Optical Instrumentation Engineers*, volume 1954, pages 473–484.

Kuczewski, R. M. (1987). Neural network approaches to multi-target tracking. *Proceedings of the IEEE International Conference on Neural Networks*, *1*, 347–361.

Lee, Y. J. & Kamen, E. W. (1993). The sme filter approach to multiple target tracking with false and missing measurements. In *Society of Photo-Optical Instrumentation Engineers*, volume 1954, pages 574–586.

Lisberger, S. G., Morris, E. J., & Tychsen, L. (1987). Visual motion processing and sensory-motor integration for smooth pursuit eye movements. *Annual Review of Neuroscience*, *10*, 97–129.

Marshall, J. A. (1990). Self-organizing neural networks for perception of visual motion. *Neural Networks*, *3*, 45–74.

Marshall, J. A. (1991). Challengers of vision theory: Self-organization of neural mechanisms for stable steering of object-grouping data in visual motion perception. In Chen, S.-S. (Ed.), *Proceedings of the SPIE 1569*, pages 200–215, San Diego.

Shet, K. C. & Rao, B. V. (1985). An overview of the Kalman algorithm. *International Journal of Electronics*, *59*(6), 657–665.

Appendix C

Basic Neural Network Structures

Conventional (Von Neumann) computing architectures, although having undergone massive advances in recent years, are still inadequate at solving certain types of problems that our brains can often solve unconsciously. In an attempt to develop a computer which is capable of tackling these problems researchers are turning towards networks which are based on the brain's functionality. These networks are generally not models of the brain at all, but are constructed from numerous elementary processors, called artificial neurons, arranged in formations which are optimized for individual problems, or implementations. None of these formations model the total complexity of even the simplest brain segment.

This appendix is not intended to be a complete review of neural networks, indeed given the proliferation of network types, this would not be possible in the space. Several books are dedicated to the topic, and further information should be obtained from such sources (Rumelhart, McClelland & the PDP Research Group, 1986; Anderson & Rosenfeld, 1988; Ripley, 1993). The information presented here is not complete but explains the features of some artificial neural network architectures, which give them both desirable and undesirable features.

C.1 Backpropagation

Back propagation neural networks perform a mapping function in a parallel manner. The network structure consists of an input layer, an output layer, and possibly one or more hidden layers, as in Figure C.1 The input layer is simply a fan-out layer, while the other layers perform all the processing. The output of

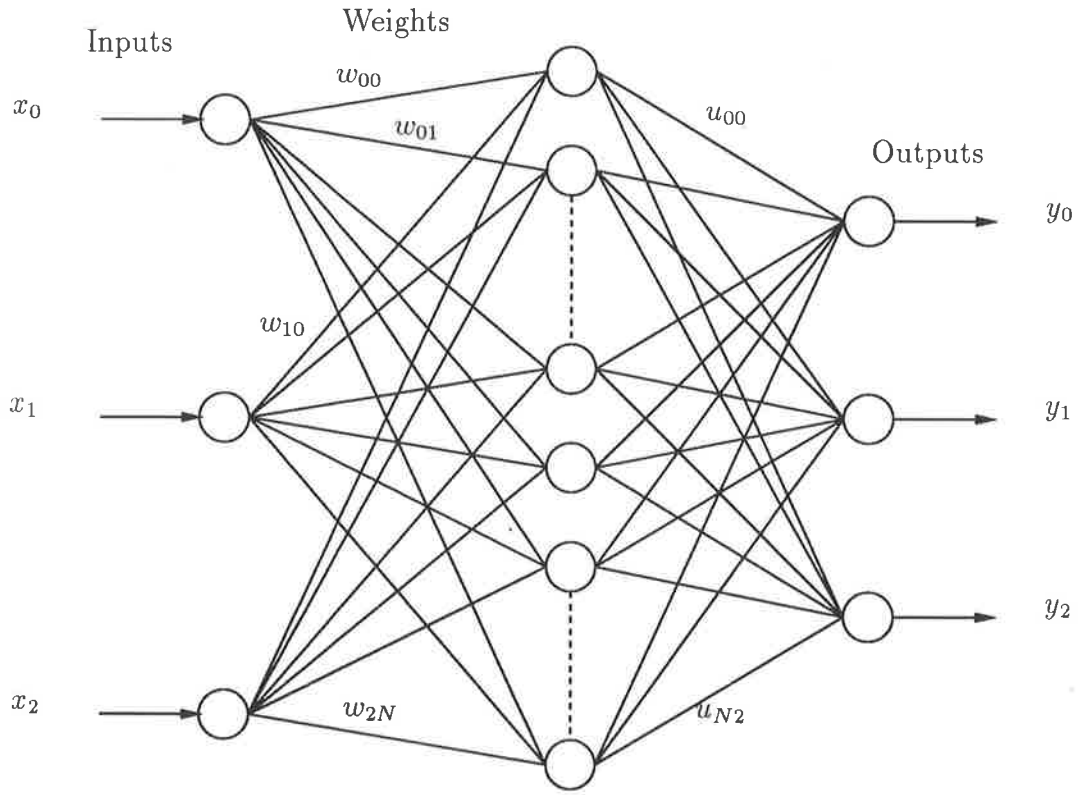

Figure C.1: Back Propagation Network Structure

each neuron performs the function

$$\text{out} = f_s(\sum_{i=0}^{N} w_i x_i).$$

The function $f_s(x)$ represents a non-linearity, normally the sigmoidal function,

$$f_s(x) = \frac{1}{1+e^{-x}}. \tag{C.1}$$

Importantly for implementation ease

$$f_s'(x) = \frac{-e^{-x}}{(1+e^{-x})^2} \tag{C.2}$$

$$= f_s(x)[1 - f_s(x)]. \tag{C.3}$$

C.1.1 Decision Boundaries

The typical recognition task involves dividing the input space into decision regions, so that the outputs specify which of the regions the input belongs to. The location of the decision regions are formed during training. Training is performed by presenting the network with samples which are known to be either inside or outside the regions. Often a large number of training examples will need to be used, especially near the boundary.

In the first hidden layer, each of the hidden neurons represents a plane in the input space,

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_N x_N. \tag{C.4}$$

Figure C.2 shows an example of a system with two inputs, where $z = 0$ represents a decision boundary, since $z > 0$ on one side of the boundary and $z < 0$ on the other. The output plane combines these decision boundaries to form regions.

Proofs exist which state that a backpropagation network with one hidden layer is capable of learning any continuous mapping (Hecht-Nielsen, 19; Cybenko, 1989; Funahashi, 1989) . In some applications where the inputs and outputs of the network may be more complex, more hidden layers can be added, these new layers act as feature extractors, and can often reduce the overall size of the network (Nguyen & Widrow, 1990) or the time required to train (Touretzky & Pomerleau, 1989).[1]

[1]By adding more layers, prior knowledge about the complexity and/or structure of the decision task is being encoded in the network structure

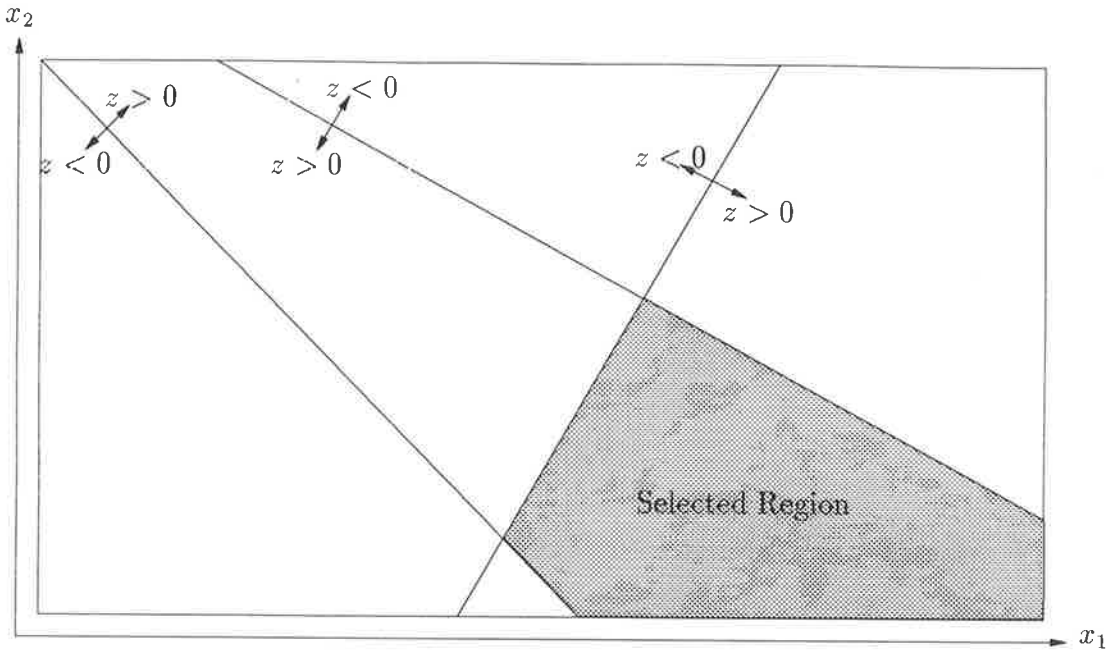

Figure C.2: Decision boundaries in a two input BPN.

The role of the sigmoid function is to threshold the planes formed by the hidden neurons. On the plane $f_s(z)$ the locus of $z = 0$ is still a straight line. However at the intersection of two or more planes, the locus of points where $z = 0$ becomes a curved approximation to the boundary. This approximation can be controlled by altering the slope of the sigmoidal function.

C.1.2 Decision Boundary Statistics

Decision boundaries will be located in such a way that they divide the hyperspace into regions. If the inputs are noise free and the only information that is required is which region of the classification space the input belongs to, then any shaped decision boundary that can be used during training is appropriate. If on the other hand the input signal is noisy or the probability that the decision is correct is required, then the decision boundary shape, (ie the Sigmoid) must be chosen to match the statistics. If the noise near the boundaries is Gaussian then use of the sigmoid gives a good estimate of the posterior probabilities (Robinson, 1992).

Bayesian Boundaries

Historically, the preferred approach to statistical decision making has relied on Bayes' theory (Bogler, 1987) which offers a highly formalized and rigorous ap-

proach to the assignment of probabilities. This is indeed the case in neural networks up to this point. It should be noted that a Bayesian Decision Network reduces to that of a perceptron if all of the hypotheses probabilities are multivariate normal density functions with identical covariance matrices, and means.(Mitiche, Henderson & Laganière, 1988) Unfortunately Bayes' approach is unable to express itself in all sensor specific levels of abstraction. In other words "Bayes' theory forces each sensor to respond with a Bayesian family of beliefs over a common level of abstraction, although most tactical sensors are inherently incapable of contributing information to this level of detail."(Bogler, 1987; Shafer, 1976) This can be demonstrated by an example.

Example 9 *Consider a case where a sensor locates 6 targets, of which one is known to be a type 'x'. From intelligence information it is known that there are 100 types of aircraft in the area, 40% of which are type 'x', the rest have not been identified. Bayes' approach will assign a prior probability of 0.6% to each type of craft(Bogler, 1987; Shafer, 1976) (since all probability must be assigned specifically,* $0.6\% = 60\%/100$*). The result of combining this data will be that craft 'x' will be identified from the selection of 6 targets with a probability of 93%.*$(93\% = \frac{40.6}{40.6+5\times 0.6})$

This seems unreasonably large considering the amount of information actually supporting the conclusion, and is caused by the need to normalize. In the approach presented in Bogler (1987); Shafer (1976) a level of ignorance is maintained so that the craft is detected with a lower (more reasonable) probability.

Unfortunately the type of function presented in Bogler (1987); Shafer (1976) cannot be represented in conventional neural network architectures. What is required is a new type of network where each connection represents two values, the probability and the ignorance.

Gaussian Boundary Statistics

If two adjacent classes are located with Gaussian probability then the probability of a point at x being a member of class a is given by

$$Pr(x) = \frac{\phi_a(x)}{\phi_a(x) + \phi_b(x)}, \tag{C.5}$$

where $\phi(x)$ is the Gaussian probability density function

$$\phi_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}. \tag{C.6}$$

The result of merging (C.5) and (C.6) is

$$Pr(x) = \frac{\frac{\sigma_b}{\sigma_a+\sigma_b}}{1+e^{\frac{(x-\mu_a)^2}{2\sigma_a}-\frac{(x-\mu_b)^2}{2\sigma_b}}}. \tag{C.7}$$

If it is assumed that $\sigma = \sigma_a = \sigma_b$ and that $\mu - \mu_a = -\mu_b$ then (C.7) reduces to

$$Pr(x) = \frac{\frac{1}{2}}{1+e^{\frac{-2\mu x}{\sigma}}},$$

which is simply the sigmoid function of (C.1).

The result, is that the sigmoid is only correct for gaussian distributed signals with identical variances. The steepness of this sigmoid is also a function of the difference of the means, so that it is possible that the shape of the sigmoid should vary between different boundaries.

Mixed Gaussian and Exponential Boundary Statistics

The Exponential density function is given by

$$\phi_E(x) = \begin{cases} ae^{-ax}, & \text{for } x > 0, \\ 0 & \text{for } x \leq 0. \end{cases}$$

When this is combined with the Gaussian probability density function (C.6) using (C.5) the resultant probability that a point at x is a member of class a is

$$Pr(x) = \begin{cases} \frac{\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}+ae^{-ax}}, & \text{for } x > 0, \\ 1 & \text{for } x \leq 0. \end{cases}$$

As can be seen from this example the expression for the correct boundary non-linearity can become overly complex when different statistic types are mixed.

C.1.3 Training

Training occurs through an adaptation of weights using error signals.(Fukushima, 1987) For each of the input patterns presented the internal weights w_i are altered to minimize the errors between the output produced by the input and the required output. For example the output to the training vector $\vec{x} = [x_1, x_2, \ldots, x_N]$ is $\vec{y} = [y_1, y_2, \ldots, x_M]$. The required output vector however is $\vec{t} = [t_1, t_2, \ldots, t_M]$, training occurs by attempting to minimize an error function, the most common of which is the *Mean Squared Error* (MSE) $E = 1/2\sum_{k=1}^{M}(e_k)^2$ where $e_k =$

$t_k - y_k$. This is averaged over the entire set of training examples according to the probabilities of their occurrences. The size of the training set may be very large, and so it will often be easier to minimize the error vector for each pattern. This has been proven to produce the same results if the patterns are presented randomly, since for a large number of training samples the errors form a statistical approximation to the MSE. Note that this assumes that the samples presented are either noise free, or that the noise is Gaussian.

Numerous schemes exist for the numerical solution of the weights. The first and possibly most used rely on gradient methods, the most common of which is simple gradient ascent, although more sophisticated approaches can rely on conjugate gradient approaches for example. These approaches have been shown to be slow, and possibly suboptimal.[2] The advantage of these slow, suboptimal adaptive routines is in the simplicity of implementation. The gradient of E with respect to the weights from the hidden layer to the output are given by

$$\begin{aligned}\frac{\mathrm{d}E}{\mathrm{d}u_{kj}} &= \frac{\mathrm{d}E}{\mathrm{d}y_k}\frac{\mathrm{d}y_k}{\mathrm{d}u_{kj}} \\ &= \frac{\mathrm{d}}{\mathrm{d}y_k}[1/2\sum_{a=1}^{M}(t_a - y_a)^2]\frac{\mathrm{d}}{\mathrm{d}u_{kj}}[f_2(\sum_{b=1}^{H}u_{kb}z_b)] \\ &= (y_k - t_k)f_2'(\sum_{b=1}^{H}u_{kb}z_b)z_j.\end{aligned}$$

If no activation is used in the output layer, then the output is linear, $f_2(x) = x$, and

$$\frac{\mathrm{d}E}{\mathrm{d}u_{kj}} = (y_k - t_k)z_j.$$

The gradient of $\frac{\mathrm{d}E}{\mathrm{d}w_{ji}}$ can similarly be defined as it is in many references.(Attikiouzel & Godfrey, 1990) The value of u_{kj} and w_{ji} are updated by

$$\delta u_{kj} = -\alpha\frac{\mathrm{d}E}{\mathrm{d}u_{kj}}.$$

The choice of α is critical since it determines the learning characteristics. If α is small then the network will learn slowly, or not at all; however if α is too large, then the network may oscillate around the correct solution, or actually diverge, so that learning will be inefficient.

[2]Other approaches such as simulated annealing may yield optimum solutions but are rarely used in practice and are not discussed here

One of the disadvantages with the Backpropagation network is that it can be overtrained so that it will recognize patterns too exactly. The result is a degradation in the generalisation performance, which means that similar patterns to those used during training, but not identical, will not be classified correctly. The problem of generalisation between trained points requires additional knowledge about the shape or smoothness of decision boundaries. These constraints can be built into the objective function of the neural network so that the problem of overtaining can be overcome. The fact that generalisation performance is normally good in the early stages of training (before overtraining) is not a feature of the neural network approach and is generally a result of smooth objective decision regions, and low network weights, so that generalisation is certainly not guaranteed.

C.2 Kohonen's Self-Organizing Feature Maps

In a Kohonen layer each neuron receives a copy of the input pattern and compares it with its vector of weights. These weights are subtractive (as distinct from backpropagation where the weights are multiplicative.) The difference at the j-th Kohonen neuron is

$$d_j = \sum_{i=0}^{N} (x_i - w_{ji})^2.$$

The node with the lowest error is the winner and represents the classification of the input pattern.

The network is trained by determining the winner for a given input pattern and modifying the winners weights towards the input vector. It is also advantageous to update surrounding neurons since this will lead to an ordered spatial relationship between processing elements.

This ordering will mean that similar inputs will be classified in similar regions of the network, so that the addition of noise will produce an output, which if not correct, will be close to the required output.

Under certain conditions Kohonen self-organising feature maps can be shown to give the same clustering as the K-means algorithm(Wu & Fallside, 1992).

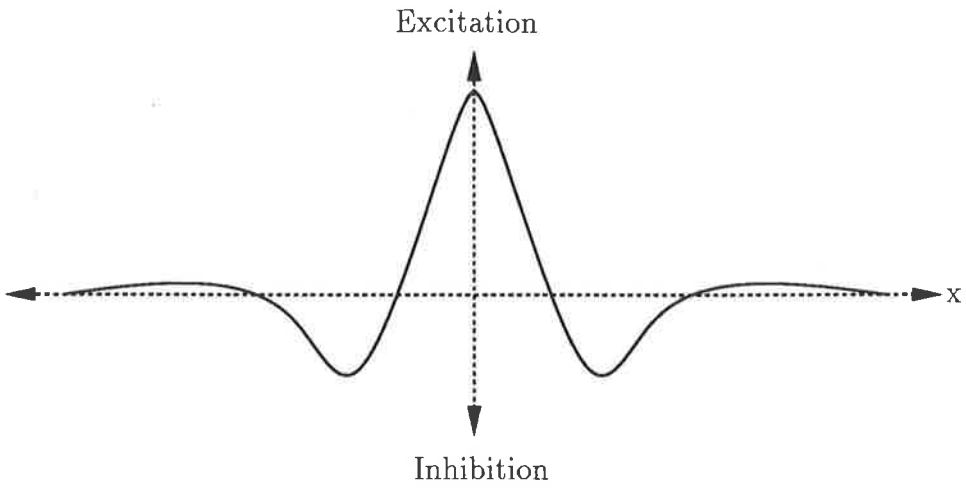

Figure C.3: Mexican Hat Function. x is the distance from the center of the hat.

C.3 Counterpropagation

The counterpropagation network is composed of three layers, an input slab, a Kohonen slab (which acts as a template or a look up table) and a Grossberg slab(which acts as an output pattern generator). This network effectively works by looking up the input pattern in the Kohonen layer and then producing the required output by multiplying the selected pattern with the Grossberg layer.

C.4 Lateral Feedback Model

The previous techniques have relied on a teacher to supply the network with the inputs and the appropriate outputs. Other structures exist which are capable of organizing themselves. In these networks neural cells are interconnected within a layer so that they receive inputs from the input layer and from adjacent neurons. This interconnection within the layer often takes the form of the Mexican hat function, as shown in Figure C.3, which contains both excitatory and inhibitory areas. The response of each neuron is given by

$$z_j = f_s \left[d_j(t) + \sum_{p=-L}^{+L} \sigma_p z_{j+p} \right],$$

where $L =$ neighborhood size. The result, after applying an input to this network, is a clustering of activity around the area of maximum excitation. This technique produces a similar network to the Kohonen network, except that the

organization produced is arbitrarily decided by the neurons during learning, and there are lateral connections within each layer.

C.5 Hopfield Model

The Hopfield model is one of the simpler (early) models of neural networks designed to perform a computing function. The model "consists of a number of mutually interconnected nonlinear devices called neurons.(Takeda & Goodman, 1986)" This network performs functions normally implemented in the major hemisphere of the brain. In some cases the advantage of rapid computation is offset by the large programming time and complexity, and conventional computer techniques may be more efficient.

The basic structure is shown in Figure C.4

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} = \sum_{j=1}^{N} T_{ij}V_j + I_i$$
$$V_i = f_s(U_i),$$

where V_i characterizes the state of the neuron, N is the number of neurons, T_{ij} are elements of the interconnection matrix and $f_s(x)$ is a nonlinear function whose form is usually that of (C.1).

Hopfield (1984) has shown that if $T_{ij} = T_{ji}$, neurons in the continuous model always change their states in such a way as to minimize an energy function defined by

$$E \equiv -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i, \tag{C.8}$$

and then stop at the minima of this function. The same is also true (Hopfield, 1982) for neurons in the discrete model if it is also assumed that $T_{ij} = 0$.

The programming of the network is performed by finding a cost function for the problem involved and by equating terms of the cost function to the energy function (C.8). This will determine the values of the connection matrix T_{ij}. There is normally more than one possible matrix as some variables may be assigned arbitrarily. The solution also depends on the number representation chosen.(Takeda & Goodman, 1986)

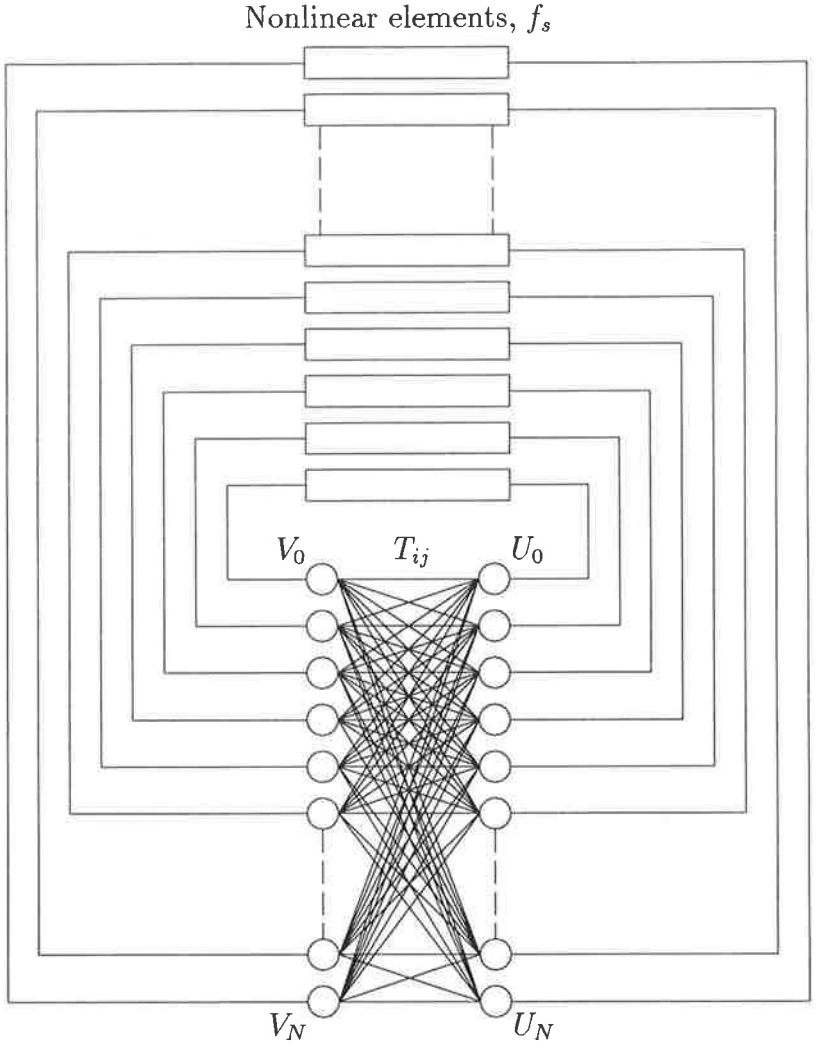

Figure C.4: The Hopfield Neural Network. The inputs, I_i, to the nodes U_i are not shown to simplify the diagram

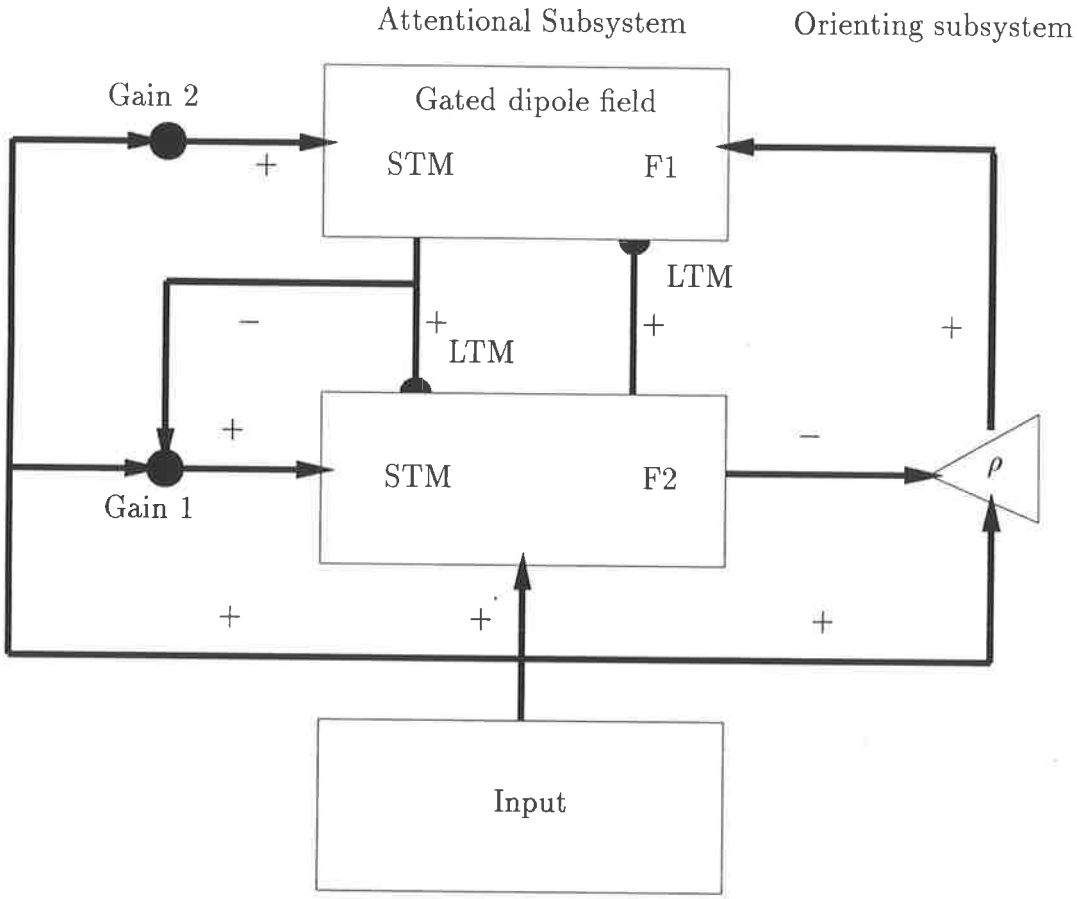

Figure C.5: A typical ART1 architecture.(Carpenter & Grossberg, 1987)

C.6 Adaptive Resonance Architectures (ART)

Adaptive resonance architectures self-organize stable recognition codes in response to arbitrary sequences of input patterns.

ART networks consist of two adaptive filters. A bottom-up filter, which encodes patterns by changing the weights or the long term memory (LTM) traces and a top-down filter which assists with code self stabilization. The bottom-up filter links a feature representation field (F1) to a category representation field (F2). These fields undergo cooperative and competitive interactions in a process of competitive learning.

There are two distinct subsystems within the ART architecture. The first is the attentional subsystem which contains the fields F1 and F2, along with the top-down and bottom-up adaptive filters. The other subsystem is the orienting subsystem which becomes active only when a bottom up input to F1 fails to match the learned active category representation at F2. In this situation the orienting system forces a reset of the active category representation at F2, and causes the attentional subsystem to proceed with a parallel search, until either a match is found or it is established that there are no matching categories.

Learning involves either the refining of a previously learnt state or the creation of a new state on previously unused nodes. The search through the recognition categories adapts as learning proceeds so that the search does not become more time consuming as the learned code becomes more complex. Search takes place only as a recognition code is being learned. The criterion for a match is adjustable by the variation of the vigilance parameter ρ, which controls the activation of the orienting subsystem. By adjusting the vigilance parameter it is possible to control the fineness of the categories. ie. by raising the vigilance the inputs will be divided into finer categories.(Carpenter & Grossberg, 1987) Figure C.5 shows a typical example of an ART1 network.

The node structure for ART networks is complicated; however the equations representing the functions at each node are relatively straight forward. For a description of the operation of an ART network and the controlling equations refer to Carpenter & Grossberg (1987). For more information on the theory behind the ART networks consult Carpenter & Grossberg (1987); Cohen & Grossberg (1983); Ellias & Grossberg (1975); Grossberg (1980a); Grossberg (1980b); Gross-

berg (1969); Grossberg (1972); Grossberg (1973); Grossberg (1978b); Grossberg (1978a)

C.7 Distributed Associative Memory

"There are no memory locations for stored items in the neural realms; all information is superimposed as specific state changes on the same memory medium, and the coding of these items is distributed over a large area. Such a distribution was previously only encountered in optical holography."(Kohonen, 1987) [3]

It is possible to create a network based on holographic theory and implemented as a hologram.

Long range connections are time variable and distributed all over the network. A matrix N represents all the feedback of synaptic weights. With an input the matrix N is modified by

$$N(t) = N(0) + \alpha \sum_{k=0}^{t} y(k)y(k)^T,$$

where $y(k)$ is the value of y, assumed to be constant over a period indexed by k. and α is a constant. Thus the changes in the matrix N represent the memory traces stored in the network. Reading this memory can only be done associatively. By partially activating the network (key) which then spreads into the other parts (recollection).(Kohonen, 1987)

C.8 Neural Implementation

A great deal of work has been performed in an attempt to implement neural networks, since to gain the benefits of the neural algorithms special parallel processors are required. The implementation of these massive parallel architectures in VLSI, has been made difficult by the interconnectivity of each processor and the planar technology which is prevalent in today's integrated circuit manufacturing. One of the solutions to this problem is the use of optical interconnects, and if the processing elements can be replaced with optical elements then a complete optical neural network will be possible.

[3] This has been shown to be only partially correct since certain types of knowledge are located at specific locations in the brain; however in that position the information is distributed over a number of neurons.

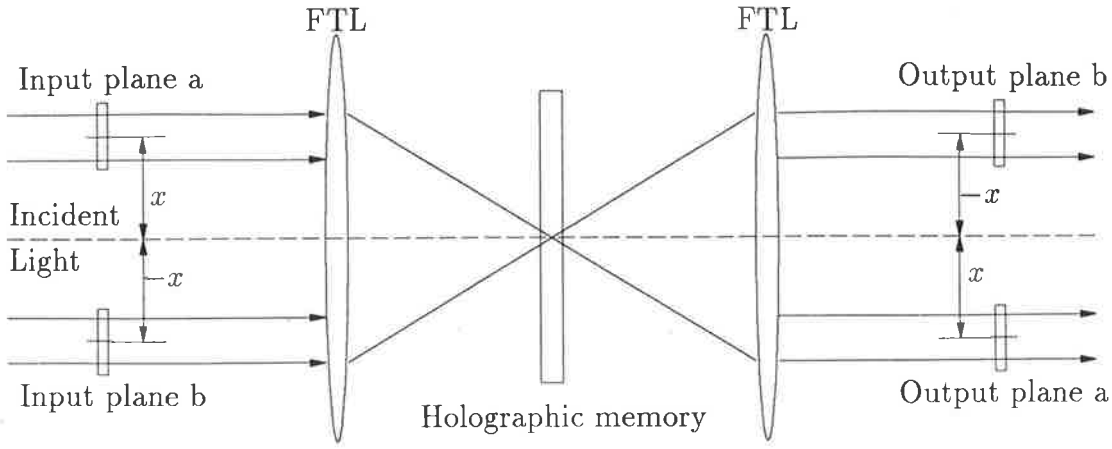

Figure C.6: Optical System for associative memory

It is now possible to implement many of the conventional logic functions using optics, thus implying that digital simulations can be implemented optically by replacing digital components with optical ones. Optical implementations have the additional advantage of being able to perform some complex operations such as Fourier transforms with simple components. The remainder of this appendix addresses some of the ways in which neural nets can take advantage of optical devices.

C.8.1 Holographic Associative Memories

Figure C.6 shows an Optical System for associative memory. Assume that the input patterns are

$$a(x,y) * \delta(x - x_0) \text{ and}$$
$$b(x,y) * \delta(x + x_0),$$

where $*$ expresses convolution.

The hologram is then exposed in the process known as memorizing, its transmission T is

$$\begin{aligned} T(f_x, f_y) = &|A(f_x, f_y)|^2 + |B(f_x, f_y)|^2 \\ &+\bar{A}(f_x, f_y)\bar{B}(f_x, f_y)e^{-j4\pi x_0 f_x} \\ &+\bar{A}(f_x, f_y)\bar{B}(f_x, f_y)e^{j4\pi x_0 f_x}, \end{aligned} \tag{C.9}$$

where f_x and f_y are spatial frequencies. The notation $A, B, \bar{A}$ and $\bar{B}$ are the Fourier transformation and its complex conjugate of patterns a and b.

If $\beta(x,y)*\delta(x+x_0)$ is applied to the input plane then the following information is obtained:

$$\bar{a}(x',y') = F^{-1}\{T(f_x,f_y)B(f_x,f_y)e^{-j2\pi x_0 f_x}\} \tag{C.10}$$

$$= \beta(x,y) * \{\bar{a}(x,y) \star a(x,y) \tag{C.11}$$
$$+\bar{b}(x,y) \star b(x,y)\} * \delta(x'+x_0)$$
$$+\beta(x,y) * \{\hat{a}(x,y) \star b(x,y)\} * \delta(x'+3x_0)$$
$$+a(x,y) * \{\hat{b}(x,y) \star \beta(x,y)\} * \delta(x'-x_0), \tag{C.12}$$

where F^{-1} is the inverse Fourier transformation, and $\star$ represents correlation.

Consequently the pattern $\hat{a}$ will appear on output- plane a

$$\hat{a} = a * (\hat{b} \star \beta) \tag{C.13}$$

This is the third term of (C.12). Similarly

$$\hat{b} = b * (\hat{a} \star \alpha) \tag{C.14}$$

if α is incident on input-plane a. if the output $\hat{b}$ is used as the input β the following expression is obtained.

$$\hat{a} = a * \{\hat{b} \star [b * (\hat{a} \star \alpha)]\} \tag{C.15}$$

hence by a two step process the recalled pattern matches the initial pattern programmed at a.

This process can be expanded so that there are more than 1 pair of patterns stored on the hologram. In this case the pattern returned is the one which has the highest correlation with the input pattern.(Mada, 1987) For a binary coded input pattern constructed of n dots the capacity of the memory should be $\frac{n^2}{2}$.

C.8.2 Optical Resonators

Various associative memories are possible using optical resonators, one of these is the ring resonator in which information is stored in a holographic medium. This medium may be a photographic plate (for read only) or a photorefractive medium (which enables dynamic learning).

An information bearing coherent light beam (called an object) is incident on the recording medium. The object is transformed to O' by the T operator which

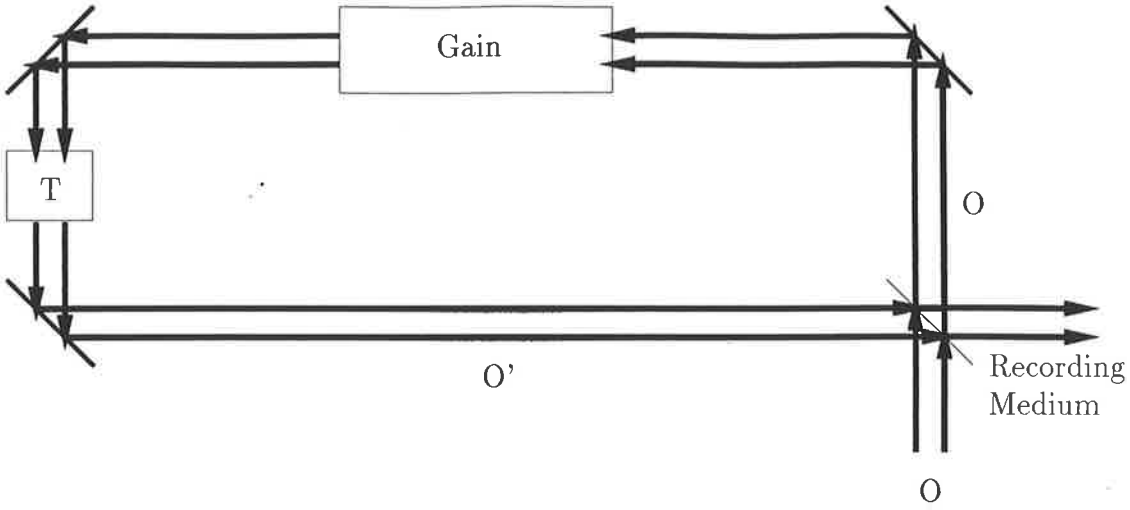

Figure C.7: Optical ring resonator

then interferes with the original beam at the recording medium. The medium records this interference pattern and effectively becomes a diffraction grating, which scatters O' back into the path of O. Thus O is reconstructed from O'.

An example of this type of system is an optical ring resonator, which is shown in figure C.7. The holographic recording medium performs the actions of the neurons, while the operator T determines the connections between the neurons, but not the strength of the connection. Thus the nature of T determines the the group properties, such as rotational symmetries, of the stable states of the network.

By using several inputs, several modes can be stored. With the addition of the photorefractive gain medium, modes will compete and cooperate. Oscillation of one mode tends to inhibit oscillation of all other modes. Recall of an object is achieved by biasing the competition with an input signal. The result is that the stored object most resembling the input will be recalled.(Anderson, 1986)

C.9 References

Anderson, D. Z. (1986). Optical resonators and neural networks. *AIP Conference Proceedings*, *12*, 151.

Anderson, J. A. & Rosenfeld, E. (Eds.). (1988). *Neurocomputing: Foundations of Research.* MIT Press.

Attikiouzel, Y. & Godfrey, K. R. L. (1990). *Artificial Neural Networks Workshop.* The University of Western Australia: The Institute of Electrical and Electronic Engineers.

Bogler, P. L. (1987). Shafer-Dempster reasoning with applications to multisensor target identification systems. *IEEE Transactions on Systems, Man & Cybernetics, SMC-17*(6), 968–977.

Carpenter, G. A. & Grossberg, S. (1987). Art 2: Self-organisation of stable category recognition codes for analog input patterns. *Applied Optics, 26,* 4919–4930.

Cohen, M. A. & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man & Cybernetics, SMC-13*(5), 815–826.

Cybenko, G. (1989). Approximations by superpositions of a sigmoidal function. *Mathematics of Control, signals and Systems, 2,* 303–314.

Ellias, S. A. & Grossberg, S. (1975). Pattern formation, contrast control, and oscillations in the short term memory of shunting on- enter off-surround networks. *Biol. Cybernetics, 20,* 69–98.

Fukushima, K. (1987). Neural network model for selective attention in isual pattern ecognition and associative recall. *Applied Optics, 26,* 4985–4992.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2,* 183–192.

Grossberg, S. (1969). On the serial learning of lists. *Mathematical Biosciences, 4,* 201–253.

Grossberg, S. (1972). A neural theory of punishment and avoidance, i: Qualitative theory. *Mathematical Biosciences, 15,* 39–67.

Grossberg, S. (1973). Contour enhancement, short term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics, LII no. 3,* 213–257.

Grossberg, S. (1978a). Competition, decision, and consensus. *J. of Mathematical Analysis and Applications*, *66*, 470–493.

Grossberg, S. (1978b). *A Theory of Human Memory: Self Organization and Performance of Sensory Motor Codes, Maps, and Plans*, volume 5, pages 233–374. New York: Academic Press.

Grossberg, S. (1980a). Biological competition: Decision rules, pattern formation, and oscillations. *Population Biology*, *77*(4), 2338–2342.

Grossberg, S. (1980b). How does a brain build a cognitive code. *Psychological Review*, *87*(1), 1–51.

Hecht-Nielsen, R. (19??). Theory of the backpropagation neural network. *??*, *??*, I–593–I–605.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Science*, *79*, 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational roperties like those of two-state neurons. *Proceedings of the National Academy of Science*, *81*, 3088–3092.

Kohonen, T. (1987). Adaptive, associative, and self-organizing functions in neural computing. *Applied Optics*, *26*, 4910–4918.

Mada, H. (1987). Adaptive, associative, and self-organizing functions in neural computing. *Applied Optics*, *14*, 2063–2066.

Mitiche, A., Henderson, T. C., & Laganière, R. (1988). Decision networks for multisensor integration in computer vision. *Society of Photo-Optical Instrumentation Engineers*, *1003*, 291–299.

Nguyen, D. H. & Widrow, B. (1990). Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, *10*, 18–23.

Ripley, B. D. (1993). Statistical aspects of neural networks. In *Proceedings of 1992 Séminaire Européen de Statistique*. Chapman and Hall.

Robinson, T. (1992). Practical network design and implementation. Cambridge Neural Network Summer School.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton: Princeton University Press.

Takeda, M. & Goodman, J. W. (1986). Neural networks for computation: number representations and programming complexity. *Applied Optics*, *25*, 3033–3046.

Touretzky, D. S. & Pomerleau, D. A. (1989). Whats hidden in the hidden layers. *Byte*, 227–233.

Wu, L. & Fallside, F. (1992). Fully vector quantized neural network-based code-excited non-linear predictive speech coding. *submitted to IEEE Transactions on Signal Processing.*

Appendix D

Data Fusion

"It often happens in science that while data are scarce, interpretation seems easy, but as the number of data grows, consistent argument becomes more and more difficult"[1]

D.1 General

Data Fusion is generally divided into three levels(Labuz, 1988; Waltz, 1981; White & Llinas, 1990)

1. Alignment, Correlation and Assignment, Tracking, and Attribute fusion.
2. Situation assessment and Threat assessment.
3. Sensor Control.

In this document I am interested primarily in the process labeled as level 1. The advantages of the fusion process can be

- Robust operational performance,
- Extended spatial coverage,
- Extended temporal coverage,
- Increased confidence,

[1]The quotation is from p. 180 of Debate about the Earth: Approach to Geophysics through Analysis of Continental Drift (Revised Edition), published in English in 1970, Hitoshi Takeuchi, Seiya Uyeda, and Hiroo Kanamori.

- Reduced ambiguity,
- Improved detection,
- Enhanced spatial resolution,
- Improved system reliability,
- Increased dimensionality.

D.2 Fusion Techniques

The fusion technique should be capable of dealing with uncertainty. Table D.2 shows how each of the requirements in table D.1 is handled by the various techniques. For a fuller description of the methods see Waltz (1981); Llinas (1989); Waltz & Llinas (1990); White & Llinas (1990). Some of the basic fusion methodologies are explained briefly here. The terminology used to refer to aspects of data fusion approaches has varied throughout its development with different researchers using different terminologies. Waltz & Llinas (1990) contains a good description of many of the terms and their historical development along with brief explanations and examples.

D.3 Positional Data Fusion

The state of the network is deduced using an optimal estimator which is intended to minimize the error with respect to some optimality criterion, using knowledge of:

- the system and measurement dynamics,
- the assumed statistics of the system noises,
- the measurement errors and the initial conditions.

This technique has the advantage that

- Minimization of error in a well-defined statistical sense.
- Utilization of all measurement data.
- Utilization of prior knowledge.

Requirement	Description
1	The combination rules should not assume independence of evidence.
2	The combination rules should not assume exhaustiveness/exclusiveness of hypothesis.
3	There should be explicit representation of the amount of evidence supporting/refuting a hypothesis.
4	There should be explicit representation of the reasons supporting/refuting a hypothesis.
5	The representation should accommodate varying levels of detail in uncertainty.
6	There should be explicit representation of consistency and detection of conflict.
7	There should be explicit representation of ignorance.
8	There should be a clear distinction between ignorance and conflict.
9	There should be a second order of measure of uncertainty.
10	The representation should be natural, and understandable to the operators.
11	The representation should be a closed form.
12	Pairwise comparisons of uncertain knowledge should be feasible.
13	The system should be able to explain the reasoning/combination process.

Table D.1: Requirements used in table D.2

Method	Requirement												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Modified Bayesian	N	N	N	N	N	N	N	N	N	Y	Y	Y	N
Confirmation	N	N	E	N	N	N	Y	N	N	N	N	Y	N
Upper/Lower Probs	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Evidential Reasoning	N	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Probability Bounds	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Fuzzy Logic	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Evidence Space	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N
Reasoned Assumptions	Y	Y	N	Y	N	Y	N	N	N	Y	Y	N	Y
Endorsements	Y	Y	N	Y	N	Y	N	N	N	Y	Y	N	Y

Table D.2: Success of various techniques to requirements, Y=Yes, N=No and E=Either Yes or No

The disadvantages are

- Sensitivity to erroneous statistics.
- High computational demands.

D.4 Attribute Data Fusion

These types of fusion do not rely on the existence of a metric or closeness of fit as in Positional Data Fusion. The approaches are inferential in nature and can be divided into two categories, statistical and heuristic, where statistical methods (e.g. Bayesian Inference) are defined as those requiring numerical interpretation, while heuristic approaches (e.g. Expert Systems) don't.

D.4.1 Classical Statistical Inference

This technique is based on empirical probability, the technique selects a "null" hypothesis H, makes observations of the input data and determines two error probabilities

$$\mu_1 = \Pr(\Pr(\text{Observations}|H) \leq \alpha|H) \text{ and}$$
$$\mu_2 = \Pr(\Pr(\text{Observations}|H) > \alpha| \sim H).$$

ie. The variable μ_1 is the probability of rejecting H when H is in fact true, and μ_2 is the probability of accepting H when H is in fact false. The value of μ_1 can be decreased by varying α but at the expense of μ_2 and vice versa.

This technique has the advantage that

- Any known distribution of data can be exploited fully.
- Either μ_1 or μ_2 can be decreased but not both.

The disadvantages are

- The population distribution function must be known or at least approximated.
- Only one hypothesis, and its negation, can be considered at any time.
- Joint distributions become complex.
- Prior probabilities are not taken into account.

D.4.2 Bayesian Statistical Inference

Bayesian inference is based on Bayes' formula which can be stated as. Let $H_i, i = 0, \ldots, n-1$, be disjoint sets with nonzero probability, and

$$C \subseteq \cup_{i<n} H_i$$

then

$$\Pr(H_i|H) = \frac{\Pr(H_i)\Pr(H|H_i)}{\sum_{j<n}\Pr(H_j)\Pr(H|H_j)}.$$

The multiple hypotheses and event version of this expression can easily be determined; however unless certain independence assumptions regarding hypothesis-event and event-event relationships can be made then the number of probabilities which need to be estimated becomes excessive. See Mitiche et al. (1988); Brown et al. (1989); Waltz & Buede (1986); Waltz (1981); Selzer & Gutfinger (1988); Thomopoulos & Okello (1988); Nahin & Pokoski (1980) for examples.

This technique has the advantage of

- Probability of a cause can be computed from observed evidence.
- Analysis of an exhaustive (and exclusive) set of causes can be performed. (This is required)
- Multiple events can be handled.

The disadvantages are

- Defining prior probabilities of causes can be difficult.
- Numerous potential causes result in a heavy computational load.
- The multiple event situation is difficult to handle unless independence assumptions are made.
- Uncertainty is not represented.
- The Hypotheses must be mutually exclusive and mutually exhaustive.

D.4.3 Dempster-Shafer Statistical Inference

The Dempster-Shafer technique generalizes Bayesian inference by associating each hypothesis with a "plausibility" factor. See Garvey et al. (1981); Bazzi (1988); Bogler (1987); Selzer & Gutfinger (1988); Chatterjee & Huntsberger (1988); Cohen et al. (1985); Huntsberger & Jayaramamurthy (1987) for examples.

Let Θ be a set (space) of primitive hypotheses H and $Prod(\Theta)$ be the set of all subsets of Θ. A probability assignment over the space Θ is a mapping

$$b : Prod(\Theta) \rightarrow [0,1] \quad \text{such that} \sum_{H \subseteq \Theta} b(H) = 1.$$

The total amount of belief in a hypothesis H in Θ is $Bel(H) = \sum_{X \subseteq H} b(X)$. The "Plausibility" of H is defined as $Pl(H) = 1 - Bel(\Theta - H)$ or the amount of belief not currently assigned to "not H", which is the same as the maximum amount of belief in H which might be assigned at some future point. The belief interval of H is $[Bel(H), Pl(H)]$

The advantages of this technique are

- A general level of uncertainty is represented.
- Probabilities range over intervals rather than points.
- Hypotheses need not be mutually exclusive.

The disadvantages are

- There is a potential for weak decisions.
- Multi-Sensor, multi-target situations are complex.
- The initial data requirements may be difficult to realize.

D.4.4 Fuzzy Set Theory

The advantages are

- The approach is very flexible.
- It is definable within formal mathematics.

The disadvantages are

- The membership functions are generally defined heuristically for military problems.
- Appropriate data bases have yet to be developed

D.4.5 Cluster Analysis

This technique determines a classification based on similarity of data vectors. See Stewart (1988); Rajapakse & Acharya (1991); Maren et al. (1989); Lindgren et al. (1986).

This technique has the advantage that

- The need for knowledge or estimation of the underlying processes is reduced.
- Useful in the development of new systems where no knowledge is available.

The disadvantages are

- A Training cycle is often required.
- Computationally demanding.
- There are many techniques of clustering from which to choose.

D.4.6 Figure of Merit

Features are collected, organised, weighted and then summed to form a number representing the merit of a particular hypothesis. See Cain et al. (1989).

This technique has the advantage that

- It is simple.
- Flexible formulation of correlation measure.

The disadvantages are

- It is not founded in formal process models.
- Development and validation are cumbersome.
- Not sensitive to environmental observations.

D.4.7 Templating

Observations are compared to established templates to determine the support for an event or conclusion. See Miltonberger et al. (1988); Garvey et al. (1981); Shaw et al. (1988); Vogel & Chan (1988); Durrant-Whyte ().

This technique has the advantage that

- It is simple and straightforward to develop template structures.
- Data representation is straight forward as is the integration of large amounts of data.
- Granularity of templates is selectable.

The disadvantages are

- The construction is inflexible.
- Data bases are complex and difficult to manage.
- Ad hoc weighting strategies can make interpretation difficult.

D.4.8 Artificial Intelligence Methods

These techniques generally employ some form of pattern matching, and use heuristic rules to evaluate hypotheses. See Sikka & Varshney (1989); Capocaccia et al. (1988); Eggleston & Kohl (1988); Johnson et al. (1989); Garvey et al. (1981); Waltz & Buede (1986); Rauch (1984).

D.5 The Fusing Process

One of the problems associated with fusion of multiple sensors, is that of alignment of the data. One way of solving this problem is to use multiresolution techniques (Stewart, 1988). This technique applies a coarsening of the data so that only the most significant information from each sensor is resolvable. If the significant features that are resolved by each sensor are the same then the data spaces can be aligned. Once alignment has occurred the coarsening is gradually removed with successively finer adjustments being made to the data alignment in a relaxation type process.

D.5.1 Parameter Fusion

The simplest method of fusing data from two sensors is to increase the dimension of the network by providing all the inputs from the sensors to the network. This approach has the advantage that all the data is available to consider, but it increases the computational problems associated with implementation since there is now more data, some of which may disagree, so that multiple alternatives will need to be evaluated. A greater problem exists if any sensor fails since this will change the inputs such that the evaluation network will need to be changed.

An alternative approach will fuse the data from various sensors to generate a probability function. This unfortunately increases the data which is connected, or stored, at each neuron, but the system can now have sensors added or removed from it without destroying operation but only changing the level of certainty.

Another alternative is to form filters or logical functions between the different sensed data and then fuse the operated data prior to processing. This can be done by adding layers within the network which perform this filtering. This type of network is biologically inspired. The optic tectum of the Rattle Snake appears to have six types of bimodal neurons used to fuse visual and infrared information, these neurons form the following filters (Ajjimarangsee & Huntsberger, 1988; Newman & Hartline, 1982)

- OR
- AND
- Visible Enhanced Infrared (VEI)
- Infrared Enhanced Visual (IEV)
- Visible Inhibited Infrared (VII)
- Infrared Inhibited Visual (IIV)

This type of approach can be extended for other fusable sensor types. Statistics of the sensor certainties may be incorporated either by weighting the results from different filters or possibly by incorporating the statistical decision making network mentioned earlier into the filtering networks.

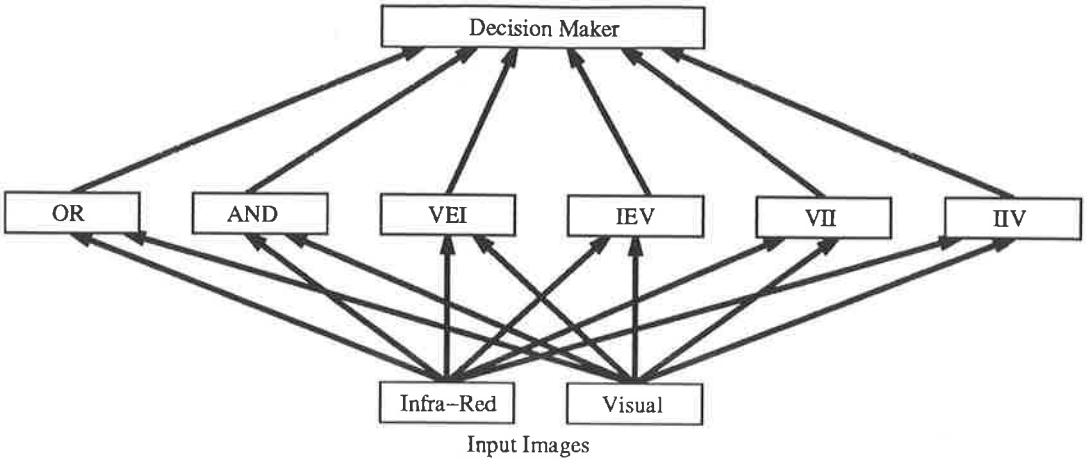

Figure D.1: Fusion Network Employing Filtering

D.5.2 Symbolic and Parameter Fusion

In surveillance systems the measured data is often incomplete and noisy. Intelligence information is also often incomplete, or uncertain. One possible way of fusing these types of data is to take advantage of the recollection and noise immunity of a Bidirectional Associative Memory (BAM) network. If the network is trained to convert measured data at the first port into a symbolic representation of the intelligence at the other then, if the measured data is presented at one port, and a symbolic interpretation of the intelligence at the other, then the network will attempt to find a consistent result. If there is no consistent result then the network will try to determine the best solution from the presented information.

D.6 References

Ajjimarangsee, P. & Huntsberger, T. L. (1988). Neural network model for fusion of visible and infrared sensor outputs. *Society of Photo-Optical Instrumentation Engineers, 1003*, 153–160.

Bazzi, Y. A. (1988). Reasoning under uncertainty with dependency information. *Society of Photo-Optical Instrumentation Engineers, 1003*, 188–193.

Bogler, P. L. (1987). Shafer-Dempster reasoning with applications to multisensor target identification systems. *IEEE Transactions on Systems, Man & Cybernetics, SMC-17*(6), 968–977.

Brown, D. E., Pittard, C. L., & Martin, W. N. (1989). Neural network implementations of data association algorithms for sensor fusion. *SPIE (Sensor Fusion II)*, *1100*, 126–135.

Cain, M. P., Stewart, S. A., & Morse, J. B. (1989). Object classification using multispectral sensor data fusion. *SPIE (Sensor Fusion II)*, *1100*, 53–60.

Capocaccia, G., Damasio, A., Regazzoni, C. S., & Vernazza, G. (1988). Data fusion approach to obstacle detection and identification. *Society of Photo-Optical Instrumentation Engineers*, *1003*, 409–419.

Chatterjee, P. S. & Huntsberger, T. L. (1988). Comparison of techniques for sensor fusion under uncertain conditions. *Society of Photo-Optical Instrumentation Engineers*, *1003*, 194–199.

Cohen, P., Davis, A., Day, D., Greenberg, M., Kjeldsen, R., Lander, S., & Loiselle, C. (1985). Representativeness and uncertainty in classification systems. *The AI Magazine*, 136–149.

Durrant-Whyte, H. F. Sensor models and multi-sensor integration, 303–312.

Eggleston, P. A. & Kohl, C. A. (1988). Symbolic fusion of mmw and ir imagery. *Society of Photo-Optical Instrumentation Engineers*, *1003*, 20–27.

Garvey, T. D., Lowrance, J. D., & Fischler, M. A. (1981). An inference technique for integrating knowledge from disparate sources. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, *1*, 319–325.

Huntsberger, T. L. & Jayaramamurthy, S. N. (1987). A framework for multi-sensor fusion in the presence of uncertainty. In *Proceedings of the 1987 workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 345–350.

Johnson, D., Shaw, S., Reynolds, S., & Himayat, N. (1989). Real-time blackboards for sensor fusion. *SPIE (Sensor Fusion II)*, *1100*, 61–72.

Labuz, J. (1988). Coordinated tracking of multiple point targets with multiple cameras. *Society of Photo-Optical Instrumentation Engineers*, *1003*, 263–269.

Lindgren, A. G., Gong, K. F., & Graham, M. L. (1986). Data fusion in a multisensor-multicontact environment. *Proceedings of the 20th, IEEE silomar Conference on Signals, Systems and Computers*, 214–220.

Llinas, J. (1989). *Data Fusion and Multi-Sensor Correlation.* Seminar Technology Training Corporation.

Maren, A. J., Pap, R. M., & Harston, C. T. (1989). A heirarchical data strusture representation for using multisensor information. *SPIE (Sensor Fusion II), 1100*, 162–178.

Miltonberger, T., Morgan, D., & Orr, G. (1988). Multisensor object recognition from 3d models. *Society of Photo-Optical Instrumentation Engineers, 1003*, 161–169.

Mitiche, A., Henderson, T. C., & Laganière, R. (1988). Decision networks for multisensor integration in computer vision. *Society of Photo-Optical Instrumentation Engineers, 1003*, 291–299.

Nahin, P. J. & Pokoski, J. L. (1980). Nctr plus sensor fusion equals iffn or can two plus two equal five? *IEEE Transactions of Aerospace and Electronic Systems, AES-16*(3), 320–337.

Newman, E. A. & Hartline, P. H. (1982). The infrared vision of snakes. *Scientific America, 246*, 116–127.

Rajapakse, J. & Acharya, R. (1991). Neuromorphic model for information fusion. In *1991 International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2397–2400, Mississauga, Canada. Imperial Press Limited.

Rauch, H. E. (1984). Probability concepts for an expert system used for data fusion. *The AI Magazine*, 57–60.

Selzer, F. & Gutfinger, D. (1988). Ladar and flir based sensor fusion for automatic target classification. *Society of Photo-Optical Instrumentation Engineers, 1003*, 236–246.

Shaw, S. W., deFigueiredo, R. J. P., & Krishen, K. (1988). Fusion of radar and optical sensors for space robotic vision. In *IEEE International Conference on Robotics and Automation*, pages 1842–1846.

Sikka, D. I. & Varshney, P. K. (1989). A distributed artificial intelligence approach to object identiffication and classification. *SPIE (Sensor Fusion II), 1100*, 73–84.

Stewart, C. V. (1988). Local parallel models for integration of stereo matching constraints and intrinsic image combination. *Society of Photo-Optical Instrumentation Engineers, 1003*, 178–187.

Thomopoulos, S. C. A. & Okello, N. N. (1988). Distributed detection with mismatched sensors. *Society of Photo-Optical Instrumentation Engineers, 1003*, 255–262.

Vogel, M. A. & Chan, T. C. (1988). Model-derived multisensor target discrimination. *Society of Photo-Optical Instrumentation Engineers, 1003*, 247–254.

Waltz, E. & Llinas, J. (1990). *Multisensor Data Fusion.* Artech House.

Waltz, E. L. (1981). Computational considerations for fusion in target identification systems. *Proc. NAECON*, 492–497.

Waltz, E. L. & Buede, D. M. (1986). Data fusion and decision support for command control. *IEEE Transactions on Systems, Man & Cybernetics, SMC-16*(6), 865–879.

White, F. E. & Llinas, D. J. (1990). Data fusion: The process of ccc i. *Defense Electronics*, 77–83.

Appendix E

Dempster-Shafer Reasoning

To many people Dempster-Shafer Reasoning appears to be unfathomable. This is partly due to the nature of the reasoning, partly due to an evangelical support or rejection of the approach by some authors which leads to a misleading, obscure or even incorrect explanation of the topic, and most importantly because there are several techniques used to determine the parameters (the labeling of which is of course also inconsistent.) All the techniques generate identical results, which appear different, and leave the reader wondering which approach is correct.

E.1 Dempster-Shafer Problem Formulation

E.1.1 Dempster-Shafer Theory Summary

A derivation and proof of most of the topics listed here can be found in (Shafer, 1976; Kyberg, 1987; Tessem, 1992; Pal et al., 1992; Bouchon & Yager, 1986; Pearl, 1988). Consider a set of hypotheses, $A_1 \ldots A_4$, on some information. It is possible that the information may be vague and may not specifically indicate which hypothesis is true. For example the information may not be adequate to determine that A_1 is true or that A_2 is true, but may provide enough information to state that the combined hypothesis, $A_1 \cup A_2$ is true. The set of all the hypotheses, and all the possible combinations of hypothesis is called the frame of discernment and is signified by θ. To each of the elements in θ a weight m is attached which represents the relative mass of evidence explicitly supporting that hypothesis or group of hypotheses. If a percentage of the evidence supports hypothesis A_1 then that is reflected in the weight attached to that hypothesis, m_1, this specific evidence does not affect the weight attached to the combined hypothesis, $A_1 \cup A_2$, represented by m_{12} since the evidence has already been

attributed. If another portion of the evidence supported either A_1 or A_2 or both, but was not precise enough to determine which, then it would not affect the value of m_1 or m_2 but the value of m_{12}. Not all the knowledge needs to be attributed, where evidence cannot be attributed to any subset of the frame of discernment the evidence is attributed to the frame itself, m_θ. The sum of the ms should be 1. Table E.1 contains an example with four hypotheses. It can be seen that the weight is attached to each hypothesis or when there is some uncertainty it is attached to a combination of hypotheses as is the case for m_{12}. The lower measure which shall henceforth be known as *support* or *belief* (and labeled as $S(.)$—some texts use $Bel(.)$) is the amount of weight which can be attributed specifically to a hypothesis. The upper measure, henceforth known as *plausibility* (and labeled as $Pl(.)$—some texts use $P^*(.)$), is the amount of evidence which can not be specifically attached to the negation of the hypothesis, or in other words the amount of evidence which may support the hypothesis.

If two independent sources of information are to be combined which are given by $m_{A_1}, \ldots, m_{A_k}$ and $m_{B_1}, \ldots, m_{B_l}$ where the sum of the masses for each information source is 1 then the evidence can be combined using Dempster's orthogonal sum defined by (E.2). This is shown pictorially in Figure E.1 where the lightly shaded regions represent $A \cap B$ while the black region represents some contradicting evidence. The information can be processed in the same manner as Table E.1 by defining a new frame of discernment, C and normalising for the conflicting evidence. Before normalisation the values of m_{C_i} are equal to the area of the rectangle in Figure E.1

To give an example of what Figure E.1 might mean consider the hypothesis *it is a plane* $(A \cap B)$ given the evidence from two sources A and B, where A_2 and B_3 represent the existence of wings, B_1 the existence of propellers and A_4 of jets. Since it is not possible to have both propellers and jets, $A_4 \cap B_1 = \phi$.

$$S(A \cap B) = S(A) \oplus S(B), \tag{E.1}$$

$$= \frac{\sum_{A_i \cap B_j \subset A \cap B} m_{A_i} m_{B_j}}{1 - \sum_{A_i \cap B_j = \phi} m_{A_i)} m_{B_j}}. \tag{E.2}$$

The denominator of this expression renomalises the result to ignore the effect of conflicting evidence.[1] Orthogonal sums of multiple frames can be formed in any

[1]The renormalisation is perhaps the most controversial aspect of Dempster's combination rule and has come under a lot of criticism.

Hypothesis	weight	lower measure, $S(.)$	upper measure, $Pl(A)$
A_1	m_1 0.2	m_1 0.2	$1 - m_2 - m_3 - m_4$ $-m_{23} - m_{24} - m_{34} - m_{234}$ 0.4
A_2	m_2 0.2	m_2 0.2	$1 - m_1 - m_3 - m_4$ $-m_{13} - m_{14} - m_{34} - m_{134}$ 0.4
A_3	m_3 0.1	m_3 0.1	$1 - m_1 - m_2 - m_4$ $-m_{12} - m_{14} - m_{24} - m_{124}$ 0.1
A_4	m_4 0.2	m_4 0.2	$1 - m_1 - m_2 - m_3$ $-m_{12} - m_{13} - m_{23} - m_{123}$ 0.4
$A_1 \cup A_2$	m_{12} 0.1	$m_1 + m_2 + m_{12}$ 0.5	$1 - m_3 - m_4 - m_{34}$ 0.7
$A_1 \cup A_3$	m_{13} 0.0	$m_1 + m_3 + m_{13}$ 0.3	$1 - m_2 - m_4 - m_{24}$ 0.5
$A_1 \cup A_4$	m_{14} 0.1	$m_1 + m_4 + m_{14}$ 0.5	$1 - m_2 - m_3 - m_{23}$ 0.7
$A_2 \cup A_3$	m_{23} 0.0	$m_2 + m_3 + m_{23}$ 0.3	$1 - m_1 - m_4 - m_{14}$ 0.5
$A_2 \cup A_4$	m_{24} 0.1	$m_2 + m_4 + m_{24}$ 0.5	$1 - m_1 - m_3 - m_{13}$ 0.7
$A_3 \cup A_4$	m_{34} 0.0	$m_3 + m_4 + m_{34}$ 0.3	$1 - m_1 - m_2 - m_{12}$ 0.5
$A_1 \cup A_2 \cup A_3$	m_{123} 0.0	$m_1 + m_2 + m_3$ $+m_{12} + m_{13} + m_{23}$ 0.6	$1 - m_4$ 0.8
$A_1 \cup A_2 \cup A_4$	m_{124} 0.0	$m_1 + m_2 + m_4$ $+m_{12} + m_{14} + m_{24}$ 0.6	$1 - m_3$ 0.8
$A_1 \cup A_3 \cup A_4$	m_{134} 0.0	$m_1 + m_3 + m_4$ $+m_{13} + m_{14} + m_{34}$ 0.6	$1 - m_2$ 0.8
$A_2 \cup A_2 \cup A_4$	m_{234} 0.0	$m_2 + m_3 + m_4$ $+m_{23} + m_{24} + m_{34}$ 0.6	$1 - m_1$ 0.8
θ	m_θ 0.0	1	1

Table E.1: Hypothesis combination using Dempster-Shafer theory

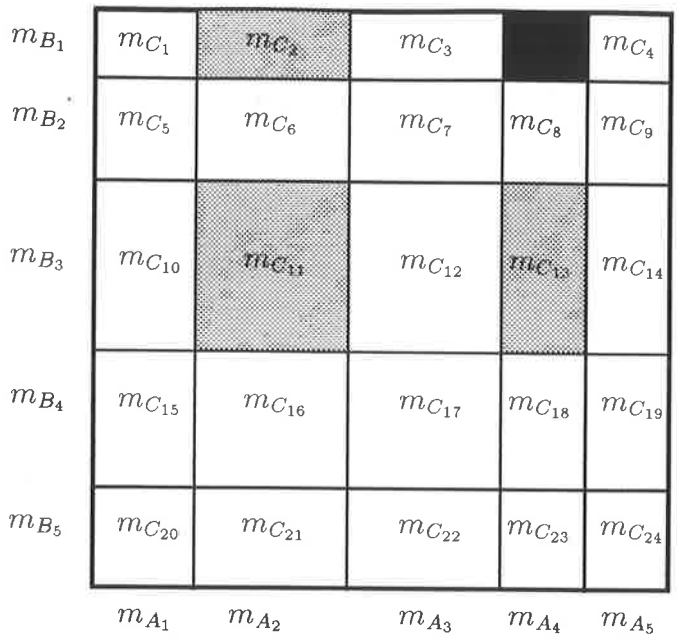

Figure E.1: Orthogonal Sum of Belief Functions.

association ie.

$$\begin{aligned} S &= S_1 \oplus \cdots \oplus S_n, \\ &= (S_1 \oplus S_2) \oplus S_3 \oplus \cdots \oplus S_n, \\ &= ((\ldots(S_1 \oplus S_2)\ldots) \oplus S_{n-1}) \oplus S_n. \end{aligned}$$

A vacuous hypothesis represents complete ignorance ie.

$$S(A) = \begin{cases} 0, & \text{if} \quad A \neq \theta, \\ 1, & \text{if} \quad A = \theta. \end{cases}$$

A Bayesian hypothesis has the property that

$$S(A) + S(\bar{A}) = 1. \tag{E.3}$$

Some properties of Dempster's rule for two belief functions S_1 and S_2 over θ are

- if S_1 is vacuous, then S_1 and S_2 are combinable and

$$S_1 \oplus S_2 = S_2.$$

- if S_1 is Bayesian, then S_1 and S_2 are combinable and $S_1 \oplus S_2$ is Bayesian.

The big advantage of Dempster-Shafer reasoning over Bayesian inferencing is the representation of a range of probabilities. The upper probability, or plausibility

of a hypothesis H, represented notationally by $Pl(H)$ is the maximum amount of probability that could be associated with H,

$$Pl(A) = \sum_{B \cap A \neq \phi} m_B,$$

while the minimum amount of support available for a hypothesis is given by

$$S(A) = 1 - Pl(\bar{A}). \tag{E.4}$$

It should be noted that for the Bayesian case (E.3) and (E.4) imply that $Pl(A) = S(A)$ so that there is no doubt in the association of the probability.

E.1.2 Combination Rules

Multiple hypothesis can be combined by the process of Dempster-Shafer reasoning, using the following rules (Garvey et al., 1981). Considering the case for two hypothesis, A and B, the hypothesis A is denoted by $A_{S(A),Pl(A)}$, where the subscripts represent the support of A, $s(A)$, and the plausibility of A, $p(A)$. In (E.6) the subscripts $S_1(A)$, $S_2(A)$, $Pl_1(A)$ and $Pl_2(A)$ refer to the same hypothesis from different sources. In all the other cases the hypotheses are different and may or may not come from a different source. The statements to the left of the $\rightarrow$ allow the statement on the right to be inferred.

$$A \subset \theta \rightarrow A_{0,1}, \tag{E.5}$$

$$A_{S_1(A),Pl_1(A)} \ , \ A_{S_2(A),Pl_2(A)} \rightarrow A_{S(A),Pl(A)}, \tag{E.6}$$

$$A_{S(A),Pl(A)} \ , \ B_{S(B),Pl(B)} \rightarrow (A \cup B)_{S(A \cup B),Pl(A \cup B)}, \tag{E.7}$$

$$A_{S(A),Pl(A)} \ , \ B_{S(B),Pl(B)} \rightarrow (A \cap B)_{S(A \cap B),Pl(A \cap B)}, \text{ and} \tag{E.8}$$

$$A_{S(A),Pl(A)} \ , \ B_{S(B),Pl(B)} \rightarrow (A|B)_{S(A|B),Pl(A|B)} \tag{E.9}$$

where

$$S(A) = \max(S_1(A), S_2(A)), \tag{E.10}$$

$$Pl(A) = \min(Pl_1(A), Pl_2(A)), \tag{E.11}$$

$$= 1 - S(\overline{A}), \tag{E.12}$$

$$S(A \cup B) = S(A) + S(B) - S(A \cap B), \tag{E.13}$$

$$Pl(A \cup B) = 1 - S(\overline{A \cup B}), \tag{E.14}$$

$$S(A \cap B) = S(A) + S(B) - S(A \cup B), \tag{E.15}$$

$$Pl(A \cap B) = 1 - S(\overline{A \cap B}), \tag{E.16}$$

$$S(A|B) = \frac{s\left(A \cup \overline{B}\right) - s\left(\overline{B}\right)}{1 - s\left(A \cup \overline{B}\right)} \text{ and} \tag{E.17}$$

$$Pl(A|B) = \frac{p(A \cap B)}{Pl(B)} \tag{E.18}$$

(E.5) is the vacuous hypothesis where no information is available. (E.9) is not uniformly accepted due to some ongoing arguments over the nature of belief functions and probability. If Belief is seen from a frequentists approach then this is almost certainly wrong (Shafer, 1992), while if the frequentist view of probability is abandoned then the combination rules can be almost arbitrary. This type of combination is separated by some authors as a special case, sometimes called conditioning, and because of the variety and arbitrariness of the combination strategies often leads to paradoxes, and consequently to lots of criticism of the entire approach.

Often the information is in a form which enables simpler combination rules to be used, for instance the data may be consonant. Consonant hypotheses arise when propositions imply other propositions are also correct. In this case it makes sense to commit all the support attributed to a hypothesis to the hypotheses which it implies. As an example consider a contour map of a hill. The contours bound a region where support can be given for the belief that the ground is higher than the contour level. Obviously any support that the ground is higher than a particular level implies that the ground is higher than lower contours. Consequently support for the higher level is committed to the lower level as well.

With this simplification (E.10) to (E.16) can be simplified to

$$\begin{aligned} S(A \cup B) &= \max(S(A), S(B)), \\ Pl(A \cup B) &= \min(1, Pl(A) + Pl(B)), \\ S(A \cap B) &= \max(0, S(A) + S(B) - 1) \text{ and} \\ Pl(A \cap B) &= \min(Pl(A), Pl(B)). \end{aligned}$$

The modifications to (E.7) for more than 2 hypotheses is straightforward so that

$$s(A_1 \cup A_2 \cup \cdots \cup A_n) = \sum_{I \subset \{1,\ldots,n\}} (-1)^{|I|+1} S(\cap_{i \in I} A_i) \tag{E.19}$$

$$= \max(S(A_1), S(A_2), \ldots, S(A_n)) \quad \text{if the hypotheses are consonant.} \tag{E.20}$$

E.1.3 Statistical Representation

One of the major difficulties is in the nature of the statistics for tracking. Dempster-Shafer reasoning relies on set theory, yet the information from tracking sensors is continuous in nature, so that there are effectively an infinite number of infinitely small hypotheses. (The frame of discernment is infinite) There is no problem in generating the orthogonal sum of belief functions since the summation of the products between two frames can be altered to an integration. The problem occurs when both the level of support and plausibility are required. It is possible to select the most plausible point[2], for the track algorithm; however since the frame is continuous, it is not possible to use (E.4) to determine the level of support for this choice since $\bar{A}$ represents the entire space excluding a point. This problem can be overcome if it is assumed that the sensors have a finite resolution, thus imposing a coursening to the frame.

Another property which could be used in the generation of track information is consonance. Even if consonance is only local to a portion of the frame, then given that portion of the frame, new expressions can be constructed for Pl and

[2]It may be that the most plausible point is not actually the best point to use. The most supported or possibly some combination of supportability and plausibility could be used to determine the optimum point.

hence S.

$$Pl(A) = \frac{\max_{x\epsilon A}\,\phi(x)}{\max_{x\epsilon\theta}\,\phi(x)},$$

$$\therefore \quad S(x) = 1 - \frac{\max_{x\epsilon\bar{A}}\,\phi(x)}{\max_{x\epsilon\theta}\,\phi(x)}.$$

These expressions are far easier to deal with, but as was pointed out, they are relevant only in particular situations.

E.2 Dempster-Shafer Validity, To Believe or not to Believe

The use of belief functions has not become wide spread due to a number of unresolved issues. These are summarized in Pearl (1990); Shafer (1991) with a number of examples where the theory is reported to fail. This paper has stimulated many responses such as (Dubois & Prade, 1988; Smets, 1992; Wilson, 1992; Wasserman, 1992; Pearl, 1992; Shafer, 1992), where discussion of both the theory and the examples in Pearl (1990) are refuted. The common reasons for the rejection of the belief approach to reasoning will be presented in the following subsections.

E.2.1 Probability and Belief

There is a distinction between knowledge and evidence which is pointed out in Pearl (1990) and Dubois & Prade (1988) where knowledge is defined to be "judgements about the general tendency of things to happen," while "evidence describes one situation in its peculiarities." The application of belief functions is often aimed specifically at the treatment of incomplete evidence, while it is perceived to be deficient in the treatment of incomplete knowledge. Shafer (1992) opposes this distinction as misleading, and points to some basic misunderstandings of what probability and consequently belief functions actually are. In Dubois & Prade (1988) it is pointed out that this is not necessarily true since it is possible to consider the determination of the belief function as an optimization.

What is questionable is that "everything relies on the postulate that probability theory is relevant for expressing subjective uncertainty, independently of any frequentist interpretation."(Dubois & Prade, 1988) In Smets (1992) this argument is pursued further by expressing knowledge as a double level structure,

where belief functions are used to express knowledge at a *credal* level, and where probability functions are used at a *pignistic* level. The credal level precedes the pignistic level, where beliefs are maintained and updated, with mapping to the pignistic level occurring only when a decision must be made. This argument certainly makes some sense when comparing it to the way we think intuitively, our beliefs are generally held with very little regard for probability, and it is only when forced to make a decision that we may map our beliefs to a probabilistic frame of reference.

Wasserman (1992) and Shafer (1982) make reference to a notion of exchangeability between belief and frequency, which has not apparently been pursued in the notion of belief functions but which applies in Bayesian analysis in the situation when an infinite sequence of observations is exchangeable (DeFinetti, 1964).

E.2.2 Updating Beliefs

Another major source of opposition to belief functions is in the dubious way that evidence is updated. The normalization factor used in Dempster's rule of conditioning is readily seen as a point of contention (Pearl, 1990; Dubois & Prade, 1988; Smets, 1992; Wilson, 1992; Wasserman, 1992). Some authors (Smets, 1992) choose to disregard the normalization completely by defining problems to be of an open world type, rather than a closed world formulation where the knowledge of the frame structure is complete, but this may not be the best approach since although this will certainly form belief functions which bound the true result, precision may be lost which causes the effective region of doubt to be large making conclusions impossible.

E.2.3 The Consequence of Uncertain Frames

Bayesians "use the principle of indifference or a generalization, maximum entropy"(Wilson, 1992) to allow for the incorporation of incomplete knowledge. However, one very common type of ignorance is in the nature of the frame of discernment. It is possible to define frames which are well known such as considering the result of a die; however in some cases the frame may not be well defined, consider the classification of a bird. Should the frame consist of a class for each type of living bird, or should it also consider classes of birds which are now ex-

tinct. The selection of either frame and then the distribution of probabilities via the process of indifference will produce different results. In the same situation using belief functions, the evidence will be allocated identically irrespective of the frame chosen. The situation of complete ignorance may appear extreme but extremely complex classifications are almost certain to present some ignorance, such that estimation of the priors may be in error by an order of magnitude, which means that the posterior probabilities can be trusted at best to be within an order of magnitude to the correct values (Wilson, 1992).

E.2.4 Coherence

The propagation of wayward priors in the Bayesian case has analogies in belief theory due to the non-uniqueness of the belief functions. This is why it is believed that belief functions are not coherent. Coherence essentially defines the efficiency of a model. If a number is assigned to an event, then that number makes sense if actions taken from the results of the number lead to sensible results. These assignments are correct if and only if the the number assigned is a probability measure (Wasserman, 1992). As pointed out in Smets (1992) it is possible to form maps between beliefs and probabilities, but how are these maps formed?

As more information is obtained the posterior Bayesian distribution converges to the truth. Information about the asymptotic nature of belief functions has not as yet been determined although it has apparently been claimed by Eddy (1990) that only probability measures and limited types of belief functions are possible as limits.

E.2.5 Implications

Many of the issues regarding the use of belief functions is yet to be determined, and in some respects the use of the theory at all is controversial. The theory does lead, in the simplified cases in which I am using it, to nice implementable functions. The belief functions I am using are really upper and lower probabilities so that the credal and pignistic representations are equivalent. Also since the beliefs are probabilities they must be consistent. The representation of the frame is such that it is known exactly so that normalization procedures make sense.

E.3 References

Bouchon, B. & Yager, R. R. (1986). *Uncertainty in Knowledge-Based Systems.* Number 286. Berlin, Heidelberg, New York, London, Paris, Tokyo: Springer-Verlag.

DeFinetti, B. (1964). *Foresight: its logical laws, its subjective sources*, pages 93–158. New York, London, Sydney, Toronto: John Wiley & Sons.

Dubois, D. & Prade, H. (1988). *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* New York: Plenum Press.

Eddy, W. F. (1990). Approximating belief functions in a rule based system. In *Proceedings of the 1st International Conference on Statistical Computing*, volume 3.

Garvey, T. D., Lowrance, J. D., & Fischler, M. A. (1981). An inference technique for integrating knowledge from disparate sources. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, *1*, 319–325.

Kyberg, H. E. (1987). Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, *31*, 271–293.

Pal, N. R., Bezdek, J. C., & Hemasinha, R. (1992). Uncertainty measures for evidential reasoning i: A review. *International Journal of Approximate Reasoning*, *7*(3), 165–183.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Pearl, J. (1990). Reasoning with belief functions: an analysis of compatibility. *International Journal of Approximate Reasoning*, *4*(5/6), 363–389.

Pearl, J. (1992). Rejoinders to comments on "reasoning with belief functions: An analysis of compatibility". *International Journal of Approximate Reasoning*, *6*(3), 425–443.

Shafer, G. (1976). *A Mathematical Theory of Evidence.* Princeton: Princeton University Press.

Shafer, G. (1982). Belief functions and parametric models (with discussion). *Journal Royal Statistics Society B*, *44*, 322–352.

Shafer, G. (1991). Perspectives on the theory and practice of belief functions. *International Journal of Approximate Reasoning*, *4*(5/6), 323–362.

Shafer, G. (1992). Rejoinders to comments on "perspectives on the theory and practice of belief functions". *International Journal of Approximate Reasoning*, *6*(3), 445–480.

Smets, P. (1992). Resolving misunderstandings about belief functions. *International Journal of Approximate Reasoning*, *6*(3), 321–344.

Tessem, B. (1992). Interval probability propagation. *International Journal of Approximate Reasoning*, *7*(3), 95–120.

Wasserman, L. (1992). Comments on shafer's "perspectives on the theory and practice of belief functions". *International Journal of Approximate Reasoning*, *6*(3), 367–375.

Wilson, N. (1992). How much do you believe? *International Journal of Approximate Reasoning*, *6*(3), 345–365.