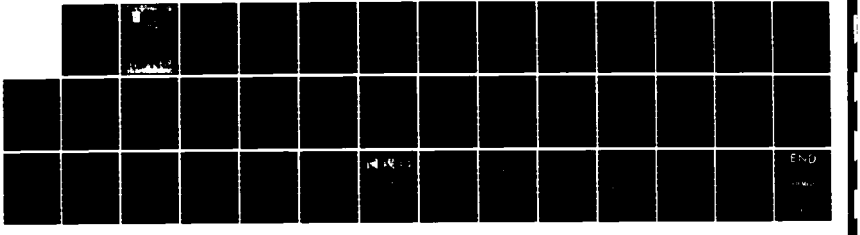
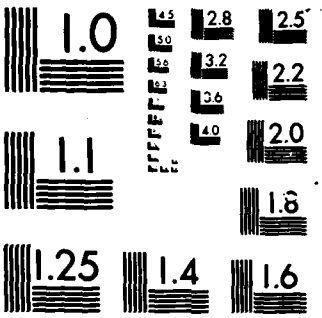
AD-A163 749 THE IMPLICATIONS OF BOLTZMANN-TYPE MACHINES FOR SAR (SYNTHETIC APERTURE R..(U) ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) S P LUTTRELL JUN 85 1/1

UNCLASSIFIED RSRE-MEMO-3815 DRIC-BR-98262 F/G 17/9 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RSRE

MEMORANDUM No. 3815

ROYAL SIGNALS & RADAR ESTABLISHMENT

THE [illegible]-KOETZMANN-TYPE MODEMS FOR [illegible]: A PRELIMINARY SURVEY

[illegible]: S P Luttrell

MEMORANDUM No. 3815

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3815

TITLE: THE IMPLICATIONS OF BOLTZMANN-TYPE MACHINES FOR SAR DATA PROCESSING: A PRELIMINARY SURVEY

AUTHOR: Dr S P Luttrell

DATE: June 1985

SUMMARY

We propose that Markov random field models (MRFs) be used as a framework within which to construct models of synthetic aperture radar (SAR) images. We clarify the relationship between this class of models and the Boltzmann machine (BM) of artificial intelligence. We then generalise the BM training procedure and use it to train MRF models. Using this technique we investigate the ability of a simple MRF texture model to learn a texture by maximising a relative entropy objective function. We find that the marriage of MRF models with the BM training procedure is fruitful.

LIST OF CONTENTS

1. INTRODUCTION

The purpose of this memo is to advocate the use of Markov random field (MRF) models in the analysis of SAR imagery. In addition we wish to promote the idea of generating suitable MRF models by a "training" procedure. We select a training procedure which has been used in the past to train Boltzmann machines (hence the title of the memo).

The organisation of the memo is such that all mathematical details are relegated to appendices in the order in which they are encountered in the body of the memo. It is an indication of the mathematical flavour of this new field that the appendices constitute a large part of this memo. However we hope that the discussion (in the form of words and pictures rather than equations) will capture the interest of the non-mathematical reader nonetheless.

In section 2 we outline the problems which we are faced with in modelling SAR imagery, and we propose that a general model-building framework (MRFs) be used instead of the plethora of existing models. In section 3 we give an overview and an interpretation of MRFs. In section 4 we give a brief description of Boltzmann machines (BMs) in order to make clear the connection with MRFs (a BM is a special case of a MRF). In section 5 we generalise the idea of a BM by taking advantage of their connection with MRFs. This enables us to design (in principle) special purpose BMs which can incorporate prior knowledge of the type of model that needs to be built. In section 6 we give some numerical simulations of a particular class of special purpose BM in order to demonstrate their ability to learn a "texture".

2. MODELS OF SAR IMAGES

A SAR image consists of a set of complex numbers (I and Q) which forms a member of the ensemble of all possible SAR images of the scene under surveillance. If one has no prior knowledge of the content of the scene, then one can not predict which member of the ensemble of possible SAR images will occur. Such uncertainties can be expressed in terms of a statistical and/or structural model of the scene. The most obvious of these uncertainties is the randomness of the phase of the radiation which is scattered from each elementary scatterer; this leads to the notorious speckle which degrades all coherent images. Other uncertainties derive from our ignorance of the underlying cross section of the elementary scatterers, and our ignorance about how they are clustered together to form the scene.

A primary task in SAR image processing is to formulate a model which can encode our prior knowledge about SAR scenes. In the Sensor Information Processing Section (SIPS) of BSI there is a program of basic research which is achieving significant success in describing some aspects of SAR images. We can partition a SAR image into two components which we describe as "clutter" and "targets". The "clutter" cross section can be modelled as a birth-death-migration process, which leads to a cross section described by correlated gamma statistics. When illuminated with coherent radiation such a cross section gives rise to K distributed scattered radiation (Oliver, 1985). The "target" cross section can be modelled as a localised strongly scattering region. This allows targets to be super-resolved (Luttrell and Oliver, 1984; Luttrell and Oliver, 1984/5; Luttrell, 1985).

The above two models are the simplest that give non-trivial results. Thus "clutter" can (and does) have a more complicated structure than that implied by a correlated gamma distribution model, and "targets" can (and do) have an internal structure which is not described by the target model used above. We could adopt one of two distinct approaches in order to develop the research program further, either

i. Develop detailed specific models for each type of clutter and target, or

ii. Develop a general framework (or methodology) within which clutter and/or target models could be built (semi)automatically.

Hitherto (i) has been used as the investigative tool of SAR research, and it has enjoyed a lot or success. However the great variety of clutter/target scenarios encountered in SAR imagery suggests that (ii) might ultimately be a more profitable line of investigation.

We propose that the novel approach in (ii) above) be adopted in order to begin to model the less well understood aspects of SAR scenes. We require a very general model-building framework in order that we do not unnecessarily constrain the resultant models. We therefore propose that Markov random field models be used as the basic framework for SAR image models; these are described in the next section.

3. MARKOV RANDOM FIELD MODELS

The conventional notion of a "Markov chain" is one of a system evolving from state to state under the influence of a stochastic matrix. We could, in principle, use a Markov chain to generate a SAR image by building the image up in a natural sequence. This would involve starting with a structureless image, and by using an exceedingly complicated stochastic matrix we could generate a Markov chain of images with increasing complexity (ie structure). If the stochastic matrix faithfully represents the processes that occur in nature (eg trees growing, humans enclosing land with hedges, humans building roads and towns, etc), then the images in the Markov chain should begin to resemble realistic SAR images. In a sense we would obtain SAR images by a process which is akin to "evolution". If we know what the relevant SAR image generating stochastic matrix was, we could (in principle) calculate the properties of SAR images. However, we do not begin to pretend that we know how to construct such a stochastic matrix, and so we shall have to model directly the properties of SAR Images without recourse to an "evolutionary" model.

The simplest suitable Markov model for SAR images will have two spatial dimensions and no time dimension. The construction of two spatial dimensional Markov models has been explained very clearly in the literature (Geman and Geman, 1984); they are a special case of "Markov random field" (MRF) models. The origin of the term MRF is as follows: "random field" is used because the image pixels can take values which are best described by probabilistic constraints; "Markov" is used because the permitted probabilistic constraints have a limited (spatial) range.

The field variables take values which are derived using a probability density function (PDF) which is conditioned by the values of the field variables in some defined "neighbourhood". Thus an MRF is defined by a complete set of neighbourhood interactions. The diagram in Figure 1 conveys the idea of an MRF pictorially. The class of models which can be so constructed is very

large, it includes; "Ising type" models, texture models, neural models (in artificial intelligence), etc. The neighbourhood interactions may be written in the form of a "potential", which determines a Gibbs distribution. This is not a trivial result - it follows from the equivalence between Gibbs distribution and MRFs. A more detailed description of MRFs is to be found in Appendix 1.

The simplest way in which we can implement an MRF model for describing SAR data is to regard the pixel values of a SAR Image as field variables whose values are mutually correlated. The observed correlations are obtained from the underlying neighbourhood interactions of an MRF model. However, the problem of deducing what underlying interactions are needed in order to generate a set of observed correlations is very subtle indeed; this is an example of the so-called "inverse problem".

Composite MRF models have been explored with some success for synthetic data sets (Geman and Geman, 1984). A composite MRF model is constructed by introducing an additional set of "hidden" field variables in order to augment the usual set of "visible" field variables; see Figure 2 for a pictorial representation. The pixel values are identified with the visible field variables, and boundary information (in Geman and Geman's model, for instance) is identified with the hidden field variables. This is a very natural procedure because the positions of boundaries certainly influence the pixel values of the data, but such influence is indirect. Thus the boundary information is regarded as "causal factors" which influence the form taken by the observed image. This approach expresses the nature of the mutual correlations amongst the pixel values in a far more satisfactory manner than any model which does not introduce the notion of boundaries. More generally we observe that any hidden variable which has an (indirect) influence on the correlations amongst the pixel values must be included if an economical model is to be constructed. For the purpose of SAR image analysis we would like to identify the primitive casual factors which give rise to the observed SAR image. The specification of these alone would then suffice to describe the image, and an extensive data compression would have been achieved.

We shall now prepare the ground for constructing a composite MRF model to describe the processes which give rise to SAR image data. However we are not in the privileged position of knowing every detail of the MRF interactions which can explain the correlations present in SAR data. Therefore we must extend the composite model to include hidden field variables/interactions whose exact structure has yet to be decided. Such additional structure could then be determined by "training" the MRF model on some SAR data. This type of approach would at best be a form of inspired model building in which one rapidly determined the hidden variable structure of the MRF, and at worst it would be a phenomenological MRF model in which a set of "unphysical" hidden variables/ interactions conspired to emulate the required correlation structure of the pixel values. A derivation of a suitable objective function (called "relative entropy") which should be maximised in order to train an MRF is contained in Appendix 2. We are currently investigating other objective functions which are more specific about what should be learnt during training.

4. THE BOLTZMANN MACHINE

An extreme case of the uncommitted MRF type of model arises if we have no prior knowledge of the MRF structure and interactions whatsoever - everything must be acquired by the training procedure. Such an MRF would be a very flexible means for emulating arbitrary MRFs. This is the methodology adopted by many in the "connectionist" approach to modelling of the (human) brain. Specifically a "Boltzmann machine" (BM) has been proposed (Hinton et al, 1984), which

has the structure of an Ising spin model MRF. We shall call the spins "units" (after Hinton et al) in the following discussion. A summary of BM operation, and a review of current literature on the subject has been given by Bounds (1985).

The units of a BM have imposed on them a prior neighbourhood structure, but the specific interaction strengths are not determined a priori. The prior neighbourhood structure is chosen at random when one is completely ignorant of the required MRF structure, and the neighbourhood interactions are permitted to take the form of a quadratic potential only (for simplicity). The diagram in Figure 3 shows such a BM structure. The BM thus has a non-committal neighbourhood structure, and a simple form of neighbourhood interaction; little (or no) prior knowledge has been encoded into the machine. The flexibility of a BM is greatly enhanced by deeming that only a few of its units are "visible"; the remainder are deemed to be "hidden". The hidden units then indirectly influence the behaviour of the visible units. This is completely analogous to the hidden boundary information influencing the visible pixel values in the composite MRF model of the previous section. However the hidden units of a BM do not in general have a physical interpretation, although Hinton et al refer to them as "elemental hypotheses".

An advantage of the BM structure is that because the behaviour of the hidden units is not directly observed, their neighbourhood interactions can be adjusted in whatever way is required for them to induce a desirable high order correlation amongst the visible units. Therefore a training procedure must regard the visible units as "supervised" and the hidden units as "unsupervised". A derivation of the gradients of the relative entropy objective function (used for training) with respect to the various neighbourhood interaction strengths is given in Appendix 3. The derivation which we present is more general than it needs to be for a BM, because the results are also used later on when more complicated machine structures are considered. Hinton et al (1984) minimise MINUS the relative entropy to train their BM - this is equivalent to our procedure.

The fidelity with which a BM can represent the correlation structure required of its visible units depends on the number of hidden units, and on the size of the neighbourhoods which are introduced. Increasing the size of either (or both) of these quantities will enhance the BM's capabilities. However the non-committal nature of the BM structure means that it can be an exceedingly inefficient means of generating some types of correlations. For instance there could be correlations present which are very simply expressed in terms of a few complicated MRF interactions, but which have a very complicated expression in terms of the simple BM interactions. In the next section we describe a way of "custom designing" parts of the BM to deal with known correlations (prior knowledge).

5. THE GIBBS MACHINE AND THE HYBRID GIBBS MACHINE

A limitation of the (canonical) BM is that one can not incorporate prior knowledge of the required correlations into the machine's structure. The number of hidden units which is required to emulate a given correlation structure is much greater (in general) than if such correlations had been built in at the outset. Of course the raison d'être of the BM is that it has a (fairly) non-committal internal structure, but this should not discourage us from attempting to optimise its performance in particular cases.

In the extreme cases where we know all of the relevant variables (visible and hidden) and their interactions we may write down the full MRF model - no training is required. Alternatively we may know the variables and their

neighbourhood structure - training is the required to determine the interaction strengths. The gradients of the objective function used during training are given by the same expression as was used for the canonical Boltzmann machine (see Appendix 3). Whether or not such a machine needs to be trained, it clearly has a more general internal structure than the Boltzmann machine described by Hinton et al (1984). Following the terminology which is used in statistical thermodynamics we suggest that the term "Gibbs machine" (GM) may be used to describe a machine which has an arbitrary internal structure. This terminology is also consistent with that of "Gibbs distributions" as described in Appendix 1. An example of a Gibbs machine is shown in Figure 4.

In the more usual case we know only some of the relevant variables and their interactions, which enables us to write down only an incomplete MRF model. The presence of additional (but unknown) variables/interactions makes it necessary to incorporate some (non-committal) BM into the associated GM in order that the values taken by the visible units can have the correct correlation properties. This is discussed in more detail in Appendix 4. The resulting machine is still a Gibbs machine, but it now consists of two parts:

i. A custom designed part which incorporates prior knowledge, and

ii. A non-committal part which gives the machine the ability to be taught correlations which were not anticipated in the prior knowledge.

An example of such a machine is shown in Figure 5. We shall call (ii) a "graft", because it serves to extent the capabilities of the machine beyond that which prior knowledge alone permits. Clearly if the graft proves to be unnecessary, then the interactions between it and (i) will be severed during training; this will happen if all the correlation structure of the visible units has been anticipated by the prior knowledge encoded in (i). Conversely the graft will interact strongly with (i) if the prior knowledge is insufficient to specify the correlation structure. We propose to use the term "hybrid Gibbs machine" or "hybrid" to describe a machine which has the heterogeneous structure given by (i) plus (ii).

The hybrid has a structure between that of the full MRF model (where we know all the variables/interactions in advance), and that of the BM model (where we know nothing in advance). The graft helps to capture whatever structure there is present IN ADDITION TO that specified by the prior knowledge. Ideally when training is complete we would like to interpret the interactions present within (ii) (and the interactions between (i) and (ii)) in order to increase our understanding (prior knowledge) of the correlation structure. This is equivalent to determining what elementary hypothesis corresponds to each unit's state, which is not a simple task in general. This difficulty arises because it is the cooperative behaviour of the units which captures correlation structure; a unit can not operate alone in a meaningful fashion. Thus elementary hypotheses (if indeed they exist) are spread out over many units of the machine. We have not resolved this problem at present, but we are devoting considerable research into elucidating what is learnt by a hybrid Gibbs machine during training; the results will be reported elswhere.

An interesting possibility which arises in the context of hybrid Gibbs machines is that of progressively growing a graft. At the outset we envisage a rudimentary machine which does not capture the correlation structure which is required of it very well (even after training). We could then "grow" the machine in a selective fashion by adding on suitably fashioned and placed pieces of graft. The form and position of such grafts would be suggested by the same training procedure which is normally used. However in this case we are positively

encouraging the birth of MRF interactions which DO NOT EXIST YET. This possibility is the most compelling MRF building strategy which we have found to date, and it is currently receiving much attention. Again the results will be reported elsewhere.

A possible criticism of the learning capabilities of a BM is that it can take an exceedingly long time to acquire a useful set of internal connection strengths by observing training set data. However we observe that this objective is damning only for completely non-committal (canonical) Boltzmann machines, because they have no prior knowledge of what they need to represent and so they need to acquire EVERYTHING from the training set. Our scheme introduces prior knowledge in an effort ot circumvent this (and other) problems by presenting the machine with a partial representation before it has observed the training set.

6. NUMERICAL SIMULATIONS

This work is in its infancy, so our numerical results are a preliminary demonstration of what can be achieved by incorporating prior knowledge into a Boltzmann machine. In order to show the relevance of the technique to SAR image processing we shall define a standard training set of signals. Such a set must possess the basic textural properties that SAR data possess, for instance a correlation length. For simplicity we shall define a class of 2D signals which is a generalisation of the 1D random telegraph signal (RTS); this is dealt with in detail in Appendix 5. The MRF structure of an RTS is shown in Figure 6. Some examples of isotropic homogeneous 2D RTSs are shown in Figure 7, and some examples of mirror-symmetric homogeneous 2D RTSs are shown in Figure 8(a), 8(b) and 8(c). In all cases we have noted the corresponding value(s) taken by the RTS probability parameters below the diagram. The diagrams shown in each part of Figure 8 have a sequence which follows a straight line in the corresponding four dimensional parameter space. We have generated all these 2D textures from Gibbs distributions by using the Metropolis algorithm. This is achieved in a fashion which is closely related to the work of Cross and Jain (1983). Our ability to generate training sets of signals by using the Metropolis algorithm allows us to simplify the training procedure which is derived in Appendix 3; this is explained in Appendix 6.

We shall confine our attention to the pure Gibbs machine (GM). By definition this has an MRF structure which is identical to that which generates the RTS training set. For an RTS defined on a square lattice with an 8-fold neighbourhood (see Appendix 5 and Figure 6) this implies that the Gibbs machine has a square lattice with an 8-fold neighbourhood. The RTS is not defined in such a way as to involve hidden variables, so the Gibbs machine will not have hidden variables; it is not a composite MRF. For an isotropic homogeneous RTS there is a single parameter p to be learnt during training, and for the more general mirror-symmetric homogeneous RTS there are four parameters p_1 to p_4 to be learnt (see Appendix 5). Training a MRF which has an identical structure to that which generates the training set is a fair test of the performance of the relative entropy objective function, ie it is a test which has been divorced from problems which arise due to an inappropriate choice of MRF structure to be trained.

Results for the isotropic homogeneous RTS are shown in table 1. A 10 by 10 (toroidal) lattice was chosen for both the RTS and the GM. The potentials were parameterised as in equation (A5.5), and the RTS probability parameter p_{RTS} and GM probability parameter p were chosen separately. The gradient in equation (A6.3) could then be calculated by SEPARATELY estimating the average of the potential for the GM (the first term) and for the RTS (the second term). The second term does not involve the GM because there are no hidden units. The estimates of the average of the potential were obtained (in both cases) by driving the MRF into equilibrium using the Metropolis algorithm, and then averaging over all 400 pairs of distinct neighbouring nodes on the 100 node (toroidal) lattice. Equilibrium was assumed to have been attained after 10 full raster scans of the Metropolis algorithm, and the average was then built up over 10 successive full raster scans. The number of scans is insufficient to generate a fair set of realisations from the RTS when long range (anti)correlations are present (eg p = 0.35 or p = 0.65 in Figure 7), but this does not affect the corresponding average much. This is because the "missing" realisations have the same statistical properties as those which are generated. The estimates of the gradients of F so obtained were converted into gradients with respect to the probability parameter p, and the results are shown in table 1. The tabulated G is effectively the contribution of each nearest neighbour pair of nodes to the total G. Clearly G is very good at distinguishing between the various members of the isotropic homogeneous class of RTSs.

Similar results for the non-isotropic homogeneous RTS (still using a pure GM) would require us to give the partial derivatives of G with respect to four probability parameters. We cannot give results as comprehensive as those in Table 1, so we shall restrict ourselves to partial derivatives which are related to the textures shown in Figure 8 only. In each case a 32 by 32 (toroidal) lattice was used for both the RTS and the GM. 10 full raster scans of the Metropolis algorithm were used to drive the RTS and the GM into equilibrium, and the averages of each of the four potentials were built up over 10 successive full raster scans by adding up all 1024 (= 32 x 32) occurrences of each potential on the lattice. The estimates of gradients of G were then converted into grad(G) measured in $\underline{p}$ space. The results are presented in Tables 2(a), 2(b) and 2 (c) which correspond to Figures 8(a), 8(b) and 8(c) respectively. We can see that grad(G) "points" in the roughly the correct direction in $\underline{p}$ space, so that if F were used as an objective function for training the GM to imitate the RTS then learning would be rapid. This is borne out by running a "steepest ascent" algorithm to locate the maximum of G.

A particular point of interest in Table 2 is the increase in gradient as one moves further away from a match between the RTS and the GM. G measures the logarithm of the probability that the GM will produce textures belonging to the RTS class chosen (see Appendix 1), so the behaviour of the gradient implies that this probability decays faster than exponentially as the GM and RTS become mismatched. A glance at each of Figures 8(a), 8(b) and 8(c) reveals that this is reasonable, because the various textures which we are comparing have very little "overlap" even when they are "adjacent" in $\underline{p}$ space. Therefore the overlap between textures must decrease extremely rapidly with separation in $\underline{p}$ space. We may estimate this decrease in probability by crudely integrating the results given in Table 2. For example (see Table 2(a)) the probability that $\underline{p} = \underline{p}_E$ produces a texture which looks like that produces by $\underline{p} = \underline{p}_A$ is crudely estimated as

$$P \simeq \exp[- 0.1 \times (0.2+0.6+1.0+1.6) - 0.1 \times (0.3+0.6+1.0+1.6) - 0.1 \times (0.5+1.1+1.4+1.7) - 0.1 \times (0.2+0.2+0.6+0.2)]$$

$$\simeq 0.25$$

This is the probability per lattice site pair per realisation of the texture (see Appendix 1 for more details).

In order to relate our results to synthetic aperture radar (SAR) applications, we must incorporate speckle into the textures. In the simple binary RTS texture model we must regard the RTS as "hidden", and instead a speckled RTS is "visible". This is a composite MRF model in which the textural organisation is performed at a deeper level than the observed quantities. Speckle itself is effectively a multiplicative noise process, which may be modelled using a transition probability between 1 and 0 (and vice versa) which occurs when viewing the (binary) RTS. Thus the 8-fold neighbourhood lattice which holds the RTS is coupled (unidirectionally) to 0-fold neighbourhood lattice which holds the observed speckled RTS. This structure is depicted in Figure 9. We shall use a unidirectional transition probability $p_{speckle}$ to control the "depth" of the speckle (relative to the separation of the binary RTS levels). Thus $p_{speckle} = 0$ will correspond to having no speckle present, and will produce the same results as in Figures 7 and 8 and Tables 1 and 2. In what follows we shall set $p_{speckle} = 0.1$, which degrades the textures to a significant extent.

Figures 10(a), 10(b) and 10(c) show speckled textures corresponding to those in Figures 8(a), 8(b) and 8(c) respectively. Note that the random numbers used in the Metropolis algorithm leading to Figure 8 are different from those leading to Figure 10, and so the underlying RTS has different values in each case. The effect of speckle is seen to degrade the textures in such a way that it becomes much more difficult to distinguish between them; this is a common problem in SAR image analysis. We shall investigate the learning ability of a GM which has an identical lattice structure to the RTS (and speckle). For this purpose we shall use the relative entropy measure again, and we shall assume that the GM already knows the value of $p_{speckle}$ so that the $\underline{p}$ must be learnt as before. The results are presented in Tables 3(a), 3(b) and 3(c) which correspnd to Figures 10(a), 10(b) and 10(c) respectively. The results in Table 3 should be compared with those in Table 2 (which were derived in the speckle-free case). As before the gradient points in roughly the correct direction in $\underline{p}$ space, and so G would be a good objective function to use during training. However the estimated gradients are much smaller than they were in the speckle-free case. This is obviously because the OBSERVED overlap between textures is greater when speckle is included, so texture discrimination is more difficult than before. The probability that $\underline{p} = \underline{p}_E$ produces a texture which looks like that produced by $\underline{p} = \underline{p}_A$ is crudely estimated as

$$P \simeq \exp[- 0.1 \times (0.1+0.3+0.3+0.2) - 0.1 \times (0.1+0.3+0.3+0.2)$$
$$- 0.1 \times (0.3+0.5+0.3+0.1) - 0.1 \times (-0.1+0.1+0.4+0.4)]$$
$$\simeq 0.68$$

Again this probability is per lattice site per texture realisation. This result is 2 to 3 times larger than the equivalent speckle-free result, so the texture overlage is correspondingly greater. If we were to increase $p_{speckle}$ then the overlap would be even greater, and the ability to discriminate between the textures would be reduced further.

In order that the GM learns steadily we must estimate the gradient of G in such a way that it is not dominated by statistical errors in the estimation procedure. Clearly this becomes more critical as the texture overlap increases

and the (ideal) gradients become smaller. Any procedure for discriminating between textures will suffer from this problem, but the grad(G) has the advantage of having a rigorous information theoretic justification (see Appendix 1).

7. CONCLUSIONS

We have seen how the relative entropy measure G may be used as an objective function to train a Gibbs machine to emulate (or capture) a texture. Furthermore we have seen that this measure is robust with respect to presence of speckle noise; its performance gracefully degrades as the effect of speckle becomes worse. We have not explored in detail the other possible ways of using G (eg Boltzmann machine, hybrid Gibbs machine, etc). Bounds (1985) has examined G in the context of Boltzmann machines, and work in progress (Luttrell) is directed towards understanding how to use G and other objective functions in the (necessarily more difficult) context of hybrid Gibbs machines.

We propose that an MRF structure is ideal for modelling SAR images, and that G is a very good measure to use for training the model. Whilst the results presented in the last section are concerned solely with the very simple 2D RTS textures, we anticipate that there are no major problems in extending the results to more general signal classes. In the language of the 2D RTS this amounts to increasing the dimensionality of $\underline{p}$ space, and so introducing more types of potential into the MRF model. The precise mix of pure Gibbs machine and pure Boltzmann machine which we should use must depend on the signal class and what we know about it (prior knowledge). A problem which can (and does) arise in more complicated problems is the possibility that the training process might run into a local (but not global) maximum of G. This and other problems might be dealt with by "annealing" the MRF representation (Hinton et al, 1984).

When we have obtained MRF representations of the various texture classes which occur in SAR imagery, we may perform many useful tasks:

i. We can generate synthetic textures corresponding to any of the training classes. This is achieved by "running" the relevant MRF by using the Metropolis algorithm.

ii. We can classify SAR textures by comparing them against the MRF representations already learnt. This is achieved by identifying which MRF has grad(G) closest to zero (for instance).

iii. We can define a texture specific target detection threshold. This is achieved by using a trained MRF to define our best approximation to the corresponding texture PDF. This enables us to define a suitable threshold.

iv. We can investigate how to refine our understanding of each texture by examining the trained MRFs. This is speculative because it is not clear in general how to perform such an interpretation of an MRF structure; this is the subject of future work.

Together these properties constitute a powerful argument in favour of using MRF models and the relative entropy objective function in the context of SAR image analysis.

8. ACKNOWLEDGEMENTS

I thank Dr R K Moore for introducing me to Boltzmann machines and Markov random field models, and I thank Dr R J A Tough for stimulating theoretical discussions on Markov Random Fields.

APPENDIX 1 - Markov Random Fields (MRFs), The Metropolis Algorithm and Gibbs Distribution - MRF Equivalence

The core concept in the development of Boltzmann machines is that of Markov random fields (MRFs). Historically the Boltzmann machine has not been developed using the language of MRFs (Hinton et al, 1984). However we find that the very general nature of MRFs makes it easier to understand the essence of what a Boltzmann machine is. Furthermore it makes it obvious how to extend the structure to more general types of machines.

The following discussion should be read in conjunction with Figure 1. A MRF is built on a mathematical object called a "graph". Loosely speaking a graph is a collection of "nodes", and a specification of what the "neighbourhood" of each node is. The neighbourhood of a particular node is those nodes which are deemed to be directly linked to a particular node. Thus a graph can be visualised as a mesh-like construct with a line passing between each pair of nodes which are neighbours. Some familiar graph-like structures are: a spider's web, the London underground, a street map, etc. Some graphs (eg street maps) have some asymmetric links (eg one way streets).

A MRF can be defined on a graph by associating a (possibly vector valued) variable with each node. The value of each such variable is influenced ONLY by the values of the variables at nodes in the (graph) neighbourhood. Such influence between the variables is encoded in the form of neighbourhood conditional PDFs (NCPDFs). Thus the joint values of the variables in the neighbourhood of a particular node is used to define a PDF. This PDF in turn defines the probability that the variable at the node of interest can take each possible value which is accessible to it. A complete set of consistent NCPDFs defines the properties of a particular MRF. Mathematically this can be summarised as follows. A NCPDF has the form $P(X_k \mid X_j, j \in N_k)$ where X_k is the value taken by the variable at node k, and N_k is the set of nodes which are neighbours of node k. The joint PDF will be denoted by $P(\underline{X})$.

In general it is not possible to solve for $P(\underline{X})$ given the set of NCPDFs (although in special cases it is possible, eg gaussian NCPDFs); we have to resort to numerical simulation. The Metropolis algorithm is universally used for generating $P(\underline{X})$ from the set of NCPDFs. The algorithm is astoundingly simple for the complicated task it has to perform! Put most simply the algorithm simultaneously updates each X_k according to its associated NCPDF. Thus the updated values each have a probability distribution which is determined solely by the previous values of the variables in the neighbourhood. Such parallel operation of the Metropolis algorithm is not essential. Nodes can be selected seqentially (raster fashion, at random, etc) and their associated variables updated sequentially. The sequence of joint states $\underline{X}$ which is generated by such a probabilistic algorithm converges to a limit distribution in which each $\underline{X}$ is selected with the correct probability as given by $P(\underline{X})$; $P(\underline{X})$ has been generated from the set of NCPDFs. Apart from some subtle points of convergence the computational problem of passing from a set of NCPDFs to the associated $P(\underline{X})$ is solved by the Metropolis algorithm as presented above.

The formulation of a model as a MRF specified by a set of NCPDFs may not be convenient in practice. For instance one may wish to derive $P(\underline{X})$ from some other quantity. This possibility can be released by invoking the equivalence between MRFs and Gibbs distributions (Kindermann and Snell, 1980). The essence of this equivalence is summarised as follows. A PDF is a Gibbs distribution with respect to a graph if

$$P(\underline{X}) = \frac{1}{Z} \exp(-U(\underline{X})) \qquad (A1.1)$$

where

$$U(\underline{X}) \equiv \sum_c V_c(\underline{X}) \qquad (A1.2)$$

The $V_c(\underline{X})$ must have the property that they each depend on X_j which are members of a single "clique" of the graph. A clique is a set of nodes which are all mutual (graph) neighbours. The Z in equation (A1.1) is a normalisation factor (partition function). The statement of the MRF-Gibbs distribution equivalence is then

> If $P(\underline{X})$ is a Gibbs distribution with respect to a graph
>
> then $\underline{X}$ is a MRF with respect to the same graph.

We can now specify the properties of $P(\underline{X})$ by specifying a graph structure and a set of $V_c(\underline{X})$. Equations (A1.1) and (A1.2) are used to build the $P(\underline{X})$ from the $V_c(\underline{X})$. Such a prescription allows us to write down $P(\underline{X})$ explicitly in terms of a set of elementary "interactions" $V_c(\underline{X})$. Such an explicit $P(\underline{X})$ is deceptively simple in form; the joint statistics of the X_k can not be deduced from the form of the $V_c(\underline{X})$ by elementary calculations. We have to recover the NCPDF structure from $P(\underline{X})$ and then REBUILD $P(\underline{X})$ by using the Metropolis algorithm if we wish to obtain explicit results. The NCPDFs are easily obtained from $P(\underline{X})$, with the MRF-Gibbs distribution equivalence ensuring that the neighbourhoods which are obtained are the same as those of the graph which was used to define $P(\underline{X})$ in the first place. In essence there are two alternative formulations of the problem:

i. Specify the NCPDFs directly, or

ii. Specify the $V_c(\underline{X})$, and hence the $P(\underline{X})$, and hence the NCPDFs.

Both of these are then numerically simulated using the Metropolis algorithm. Formulation (i) is useful if the probablistic interactions amongst the variables (on the graph) are directly known. Formulation (ii) has a more physical flavour where a set of "potentials" is used to specify the interactions amongst the variables.

APPENDIX 2 - Relative Entropy

A Boltzmann machine (and its generalisations) operates by using an internal MRF to emulate an external PDF. Clearly it would be very useful to have a measure of the quality of such an emulation, because this could then be used as an objective function in an optimisation procedure designed to "train" the Boltzmann machine.

Information theory gives us a rigorous measure of the extent to which an a posteriori PDF $p(\underline{x})$ is more committal than an a priori PDF $q(\underline{x})$

$$G \equiv - \int d\underline{x}\ p(\underline{x}) \log [\ p(\underline{x})/q(\underline{x})\ ] \qquad (A2.1)$$

This measure has been given various names such as: relative entropy, cross entropy, directed divergence, expected weight of evidence. However we shall use the term "relative entropy" because this conveys the (correct) notion that $q(\underline{x})$ is being used as a reference PDF.

There is much confusion about the origin and correct way of using G, so we shall give a simple derivation of G from elementary principles. The type of argument which we shall use is similar to that used by Shannon in deriving the entropy expression in the context of communication theory (ie information theory). We shall not assume anything about the role of $p(\underline{x})$ and $q(\underline{x})$ in order to separate such considerations from the basic task of comparing the PDFs.

We wish to define a measure of the similarity (or the difference) between the two PDFs. The most useful type of measure would have an operational definition; ie it would relate to some practical situation. The simplest such situation which we can conceive of is one where states $\underline{x}$ are selected at random with a frequency given by a PDF. Let us denote a sequence of such states by the term "chain". Clearly each possible choice of PDF will give rise to its own characteristic set of such chains, and each such set of chains could be used to characterise the PDF which generated it. Thus we shall use the structure of the set of chains which are generated by a PDF in order to provide an operational definition of the properties of the PDF. An important caveat which must be mentioned before we develop the associated theory is that each state in a chain must be chosen independently of all the others; only the generating PDF is permitted to influence the choice of state. This restriction can be removed at the cost of moving to a higher dimensional "super-state" space in which each super-state represents a correlated chain of states, but where the super-states themselves are statistically independent. We shall not consider this more sophisticated analysis here other than to note that it provides the means for analysing correlated sequences of states (such as time series). This method is closely related to "block entropy" or "Renyi entropy".

In order to facilitate the theoretical development we shall discretise the state $\underline{x}$; this is equivalent to dividing state space (phase space) into cells which are labelled by an index. Thus state space is covered by a set of non-overlapping cells, except possibly for a set of measure zero (with respect to both $p(\underline{x})$ and $q(\underline{x})$). Thus the PDFS are discretised in the following fashion

$$p(\underline{x}) \longrightarrow \{p_1,\ p_2,\ \ldots\ldots,\ p_m\} \qquad (A2.2)$$

$$q(\underline{x}) \longrightarrow \{q_1,\ q_2,\ \ldots\ldots,\ q_m\} \qquad (A2.3)$$

where

$$\sum_{j=1}^{m} p_j \equiv 1 \tag{A2.4}$$

and

$$\sum_{j=1}^{m} q_j \equiv 1 \tag{A2.5}$$

If we generate a chain of N states using the p_k to define the state probabilities, then the number of times state j occurs in the chain is approximately Np_j. As $N \longrightarrow \infty$ this approximation becomes more and more accurate. Therefore Np_j is the average frequency of occurrence of state j in chains generated by the p_k. A similar comment applies to the q_k. These frequencies completely specify the statistical properties of infinite length chains of states, and so will be used to construct an operational measure of the properties of PDFs. We shall call the chains which have the average frequencies of occurrence of states "likely chains", and those with other frequencies of occurrence of states "unlikely chains".

The probability of occurrence of a particular chain of N states generated by the p_k is given by

$$\pi(\underline{p},\underline{n}) \equiv (n_1)^{p_1}(n_2)^{p_2} \ldots\ (n_m)^{p_m} \tag{A2.6}$$

where state j occurs n_j times, and

$$\sum_{j=1}^{m} n_j \equiv N \tag{A2.7}$$

The number of ways in which such a chain can be generated is given by a multinominal coefficient

$$W(\underline{n}) \equiv \binom{N}{n_1,\ n_2,\ n_3,\ \ldots\ n_m} \tag{A2.8}$$

$$\equiv \frac{N!}{(n_1)!\ (n_2)!\ \ldots\ (n_m)!} \tag{A2.9}$$

The total probability of generating such a chain is given by the product $\pi(\underline{p},\underline{n})W(\underline{n})$.

In order to obtain an operational measure of the similarity of the set of p_k and the set of q_k we shall measure the probability that the q_k could generate "likely chains" of the set generated by the p_k, and vice versa. Thus we imagine that a set of likely chains has been generated by one PDF, which is then used as a standard against which to measure the chain generating performance of another PDF. The probability that the q_k can generate a p_k-likely chain of length N is given by

$$P(\underline{q},N \mid \underline{p}) \equiv \pi(\underline{q},N\underline{p})\ W(N\underline{p}) \tag{A2.10}$$

This expression may be recast by using Stirling's approximation for log(z!) (where z is large). Thus we arrive at

$$\log(\ P(\underline{q},N \mid \underline{p})\ ) = -N \sum_{j=1}^{m} p_j\ \log(\ p_j/q_j) \tag{A2.11}$$

We can recover the continuum analogue of equation (A2.11) by the following replacements

$$\sum_{j=1}^{m} p_j \cdot \longrightarrow \int d\underline{x}\ p(\underline{x}). \tag{A2.12}$$

and

$$p_j/q_j \longrightarrow p(\underline{x})/q(\underline{x}) \tag{A2.13}$$

Thus

$$\log(\ P[q(\underline{x}),N \mid p(\underline{x})]\ ) = N\ G \tag{A2.14}$$

or equivalently

$$P[q(\underline{x}),N \mid p(\underline{x})] = \exp(G)^N \tag{A2.15}$$

where the definition of G in equation (A2.1) has been used. Equation (A2.15) gives the probability that $q(\underline{x})$ generates a $p(\underline{x})$-likely chain of length N. A corresponding formula can be obtained where the roles of $q(\underline{x})$ and $p(\underline{x})$ are interchanged.

The G measure is used in relative entropy determination of an a posteriori PDF $p(\underline{x})$, when constriants in the form of expectation values are available, and a prior estimate $q(\underline{x})$ of the PDF is available. This procedure ensures that the $p(\underline{x})$ which is chosen as the solution generates a set of likely chains which maximise the probability (subject to the constraints) that members of a set can be generated by $q(\underline{x})$. This is an eminently reasonable criterion for selecting a $p(\underline{x})$, and it has been shown to follow from a few elementary consistency axioms (Shore and Johnson, 1980).

The G measure has been used to determine the degree of similarity between the "visible" part of a Boltzmann machine's PDF and the required PDF. The probability that the visible units' PDF generates a likely chain from the set generated by the required PDF should be maximised. Thus in equation (A2.1) the $p(\underline{x})$ is the required PDF and the $q(\underline{x})$ is the visible units' PDF; G should then be maximised with respect to the $q(\underline{x})$ which the Boltzmann machine can generate. This prescription gives rise to the standard Boltzmann machine training procedure (Hinton et al, 1984).

APPENDIX 3 - General Training Procedure

The procedure whereby a Boltzmann machine is trained is just maximisation of an objective function G with respect to the neighbourhood interaction strengths permitted by the machine's internal structure. The details of how this is achieved for the canonical Boltzmann machine have been given already (Hinton et al, 1984). Here we give the generalisation of this method which enables us to train an arbitrary MRF structure to emulate a required PDF.

Let us group the visible (hidden) units together into a vector $\underline{V}$ ($\underline{H}$), and let us denote the required PDF by $p(\underline{V})$. The PDF which is generated by the MRF structure of the generalised Boltzmann machine will depend on both $\underline{V}$ and $\underline{H}$, and we shall denote it by $q(\underline{V},\underline{H})$. It takes the form

$$q(\underline{V},\underline{H}) \equiv \frac{1}{Z} \exp\left[-\sum_{j=1}^{r} w_j U_j (\underline{V},\underline{H})\right] \qquad (A3.1)$$

where the equivalence between MRFs and Gibbs distribution has been used. The $U_j(\underline{V},\underline{H})$ form a set of permitted potentials, and the w_j give the overall strength with which each U_j contributes to $q(\underline{V},\underline{H})$. The Z factor normalises the PDF (it is the partition function). The canonical Boltzmann machine uses a special case of equation (A3.1) where each element of $\underline{V}$ and $\underline{H}$ is a binary variable, and the $U_k(\underline{V},\underline{H})$ are each restricted to taking one of the five forms: V_i, H_i, V_iV_j, H_iH_j, V_iH_j. The binary nature of the underlying variables makes the canonical Boltzmann machine equivalent to an Ising spin model. The PDF over $\underline{V}$ alone is obtained by integrating $q(\underline{V},\underline{H})$ over all $\underline{H}$. Thus

$$q(\underline{V}) \equiv \int d\underline{H}\; q(\underline{V},\underline{H}) \qquad (A3.2)$$

In order to maximise G (see equation A1.1) we need

$$\frac{\partial G}{\partial w_i} = \int d\underline{V}\; p(\underline{V}) \frac{\partial}{\partial w_i} \log\left(q(\underline{V})\right) \qquad (A3.3)$$

However

$$\frac{\partial}{\partial w_i} \log\left(q(\underline{V})\right) = \frac{1}{q(\underline{V})} \frac{\partial}{\partial w_i} q(\underline{V})$$

$$= \frac{1}{q(\underline{V})} \int d\underline{H}\; \frac{\partial}{\partial w_i} q(\underline{V},\underline{H}) \qquad (A3.4)$$

The gradient with respect to the w_i which is required in equation (A3.4) is given by

$$\frac{\partial}{\partial w_i} q(\underline{V},\underline{H}) = q(\underline{V},\underline{H}) \left[\bar{U}_i - U_i(\underline{V},\underline{H}) \right] \qquad (A3.5)$$

where $\bar{U}_i$ is the mean value of $U_i(\underline{V},\underline{H})$ when states $(\underline{V},\underline{H})$ are selected with a frequency corresponding to $q(\underline{V},\underline{H})$. On substituting equation (A3.5) into equation (A3.4) we obtain

$$\begin{aligned} \frac{\partial}{\partial w_i} \log(q(\underline{V})) &= \frac{1}{q(\underline{V})} \int d\underline{H}\, q(\underline{V},\underline{H}) \left[\bar{U}_i - U_i(\underline{V},\underline{H}) \right] \\ &= \bar{U}_i - \int d\underline{H}\, q(\underline{H}|\underline{V})\, U_i(\underline{V},\underline{H}) \\ &= \bar{U}_i - \bar{U}_i(\underline{V}) \end{aligned} \qquad (A3.6)$$

where $\bar{U}_i(\underline{V})$ is the mean of $U_i(\underline{V},\underline{H})$ when states $\underline{H}$ are selected with a frequency corresponding to $q(\underline{H}|\underline{V})$. Substituting the result in equation (A3.6) into equation (A3.3) gives

$$\begin{aligned} \frac{\partial G}{\partial w_i} &= \int d\underline{V}\, p(\underline{V}) \left[\bar{U}_i - \bar{U}_i(\underline{V}) \right] \\ &= \bar{U}_i - \int d\underline{V}\, p(\underline{V})\, \bar{U}_i(\underline{V}) \end{aligned} \qquad (A3.7)$$

Thus the gradient of the relative entropy with respect to one of the Gibbs parameters depends on the difference between

(i) The average value of the associated piece of the potential with the MRF selecting states according to the full Gibbs distribution $q(\underline{V},\underline{H})$, and

(ii) The average value of the associated piece of the potential with the MRF selecting states according to the conditional Gibbs distribution $q(\underline{H}|\underline{V})$, and with states $\underline{V}$ being selected according to the required PDF $p(\underline{V})$.

If the components of $\underline{V}$ and $\underline{H}$ are binary variables, and the $U(\underline{V},\underline{H})$ are restricted to the five forms permitted for a canonical Boltzmann machine, then the result in equation (A3.7) is the same as that already obtained for Boltzmann machines (Hinton et al, 1984).

Equation (A3.7) is fairly simple (in principle) to implement in a training procedure. Because $q(\underline{V},\underline{H})$ is a Gibbs distribution it defines a MRF structure, so the Metropolis algorithm may be used to calculate the expectation values of any sample function. Such calculations must be run for long enough to allow the Metropolis algorithm to generate a sufficiently long chain of states that the limit distribution (ie equilibrium) is reached. Simulated annealing may be used to accelerate the approach to the limit distribution (Hinton et al, 1984). The first term in equation (A3.7) may be estimated by this procedure because the relevant PDF $q(\underline{V},\underline{U})$ has the form of a Gibbs distribution. The second term in equation (A3.7) is not so simple because it involves the PDF $q(\underline{H}|\underline{V})p(\underline{V})$. The $q(\underline{H}|\underline{V})$ piece has the form of a Gibbs distribution and so the Metropolis algorithm may be used to

generate chains of states $\underline{H}$ for a given $\underline{V}$. For instance the $\underline{V}$ may be the members of a training set of states which are used to fix suitable $\underline{V}$ before running the Metropolis algorithm to generate chains of states from $q(\underline{H}|\underline{V})$. The required expectation values can be estimated if the Metropolis algorithm is allowed to reach its limit distribution SEPARATELY for each training set vector $\underline{V}$, and if the results are then summed over the $\underline{V}$ (Hinton et al, 1984). This assumes that the $\underline{V}$ are importance sampled from $p(\underline{V})$; if they are not then the sum over the $\underline{V}$ must be suitably weighted.

We may derive expressions for higher derivations of G by using the results given above. As an example we may obtain the second derivative as

$$\frac{\partial^2 G}{\partial w_i \partial w_j} = [\ \bar{U}_i \bar{U}_j - \int d\underline{V}\ p(\underline{V})\ \bar{U}_i(\underline{V})\ \bar{U}_j(\underline{V})\ ]$$

$$- [\ \overline{U_i U_j} - \int d\underline{V}\ p(\underline{V})\ \overline{U_i U_j}(\underline{V})\ ] \qquad \text{(A3.8)}$$

where the same notation has been used as in equation (A3.7). The Metropolis algorithm is used to estimate the second derivative in an analogous fashion to the first derivative.

An interesting situation arises if the $P(\underline{V})$ itself is a Gibbs distribution; then the Metropolis algorithm can be used to estimate ALL the required expectation values. This is treated in Appendix 6.

APPENDIX 4 - Incorporating Prior Knowledge

The canonical Boltzmann machine structure is non-committal insofar as it does not presume that there is any particular correlation structure in $p(\underline{V})$. Appendix 3 contains the training procedure for arbitrary $p(\underline{V})$. In neither case is the Boltzmann machine primed with information which will assist it in emulating $p(\underline{V})$. However the types of situation where prior knowledge is available are so widespread that they deserve to be considered.

Prior knowledge is usually partial and can consist of snippets of information of various kinds such as

(i) The observed data is "coupled to" a common source (hidden variable)

(ii) There are probabilistic constraints on the source and its couplings to the observed data.

This type of prior knowledge can very conveniently be expressed in an MRF structure. The simplest example of this is when we express PART of the structure of $p(\underline{V})$ in terms of a Gibbs distribution. Specifically if we may express our prior knowledge about $p(\underline{V})$ by the following replacement

$$p(\underline{V}) \longrightarrow p_1(\underline{V},\underline{H}')\ .\ p_2(\underline{V},\underline{H}',\underline{H}'') \qquad \text{(A4.1)}$$

where $p_1(\underline{V},\underline{H}')$ has a known graph/interaction structure, and where $p_2(\underline{V},\underline{H}',\underline{H}'')$ is a residual (and unknown) PDF. The canonical Boltzmann machine can be modified into a hybrid which incorporates

(i) The graph/interaction structure of $p_1(\underline{V},\underline{H}')$, and

(ii) A "graft" of canonical Boltzmann machine graph nodes which serve to emulate the effect of the unknown $p_2(\underline{V},\underline{H}',\underline{H}'')$.

Note that in the special case where $P_1(\underline{V},\underline{H}')$ alone is a perfect model for generating $p(\underline{V})$ then the graft is not necessary. A hybrid is depicted in Figure 5.

APPENDIX 5 - Random Telegraph Signals

In order to investigate the ability of a Boltzmann machine, a Gibbs machine or a hybrid Gibbs machine to capture the correlation structure of the signals in a training set, it is convenient to define a standard training set. In one dimension the simplest non-trivial signal class is the random telegraph signal (RTS). Such a signal is binary valued and its statistical properties are completely specified by the transition probability p, which is the probability that the value (0 or 1) that the signal takes at time t_{n+1} is different from that which it takes at time t_n. The 1D RTS is thus a Markov chain with a stochastic matrix given by

$$S \equiv \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \tag{A5.1}$$

We have introduced the RTS because it is the simplest signal structure which has a non-zero correlation length L. From elementary considerations L may be estimated by using

$$L = \begin{cases} O(1/p) & 0.0 < p \leq 0.5 \\ O(1/(1-p)) & 0.5 \leq p < 1.0 \end{cases} \tag{A5.2}$$

The first (second) case in equation (A5.2) corresponds to an RTS which is (anti)correlated in value at adjacent times. p=0.5 will give rise to a pure noise signal, ie no correlations whatsoever.

We may formulate the 1D RTS as a Gibbs distribution (see Appendix 1). The probability of a particular chain of length N occurring is

$$P_N(x_1, \ldots, x_N) = \frac{1}{2}\, q(x_2,x_1)\, q(x_3,x_2) \ldots q(x_N,x_{N-1}) \tag{A5.3}$$

where

$$q(x_i,x_j) \equiv \exp(-U(x_i,x_j)) \tag{A5.4}$$

and

$$U(x_i,x_j) \equiv \log(1-p) + \log(p/(1-p))\,(x_i-x_j)^2 \tag{A5.5}$$

and x_k is the state of the RTS at time k. The factor 1/2 in equation (A5.3) arises because the limit distribution of the Markov chain occupies the 0 and 1 states with equal probability. From equations (A5.3), (A5.4) and (A5.5) we see that the probability of occurrence of an arbitrary length chain is made up of a product of terms each of which has the form of a Gibbs distribution. Some particular points to note are:

(i) The form of the Gibbs distribution is symmetric with respect to reversal of the time variable. This is because the Gibbs distribution which we have constructed does not have the Markov chain's transient behaviour (which is causal) included in its specification.

(ii) The form of the Gibbs distribution is translation invariant. This follows from the homeogeneity of the definition of the 1D RTS.

(iii)The properties of the RTS reside in a "potential" $U(x_k, x_{k+1})$ which exists BETWEEN adjacent times t=k and t=k+1.

Remark (iii) above may be used in order to generalise the 1D RTS to a 2D RTS (and higher dimensions if required). Thus we may create a 2D graph with nodes placed on a square lattice. We shall define the 2D neighbourhood of a node to be the 8 nearest neighbours (N, S, E, W, NE, SE, SW and NW) on the lattice, see Figure 6. We shall define a Gibbs distribution by analogy with the 1D RTS above. Thus each pair of neighbouring nodes will have an associated potential which takes the form given in equation (A5.5), and which gives a contribution to the Gibbs distribution as in equation (A5.4). The Gibbs distribution is formed by taking the product of these separate contributions for all distinct pairs of neighbouring nodes. This definition is "isotropic" insofar as it has the same symmetry as that of a square lattice.

We may use the Metropolis algorithm to generate realisations of the 2D RTS from its Gibbs distribution (see Appendix 1). These may be used to generate sets of correlated signals for training purposes. Some examples of 32 by 32 2D RTSs are shown in Figure 7 for various values of p in equation (A5.5).

The potential defined in equation (A5.5) gives rise to a MRF structure which is a special case of the autobinomially distributed MRF of Cross and Jain (1983). Our RTS model corresponds to their second order binary model with suitably chosen values for the model parameters.

A more general 2D RTS can be constructed by removing the isotropy condition, but retaining translation invariance and mirror symmetry. This permits four independent types of potential to appear in the Gibbs distribution, one for each possible direction in the square lattice. Let us denote these by p_1, p_2, p_3 and p_4 (corresponding to E/W, N/S, NE/SW and SW/NE directions respectively). Again the Gibbs distribution is formed by taking the product of contributions from all distinct pairs of neighbouring nodes. However for this more general case each potential must be selected according to which of the four directions the corresponding node pair defines.

APPENDIX 6 - Training Procedure for Gibbs Distributions

This appendix assumes the results of Appendix 3. If the required PDF $p(\underline{V})$ itself is derived from a Gibbs distribution, then it must have the form

$$p(\underline{V}) = \int d\underline{H}' \; p(\underline{V},\underline{H}') \qquad (A6.1)$$

where

$$p(\underline{V},\underline{H}') \equiv \frac{1}{Z'} \exp\left[-\sum_{j=1}^{r'} w'_j U'_j(\underline{V},\underline{H}') \right] \qquad \text{(A6.2)}$$

which should be compared with equation (A3.1). The $\underline{H}'$ are a new set of hidden variables which contribute to the form of $p(\underline{V})$ in an analogous fashion to the affect of the $\underline{H}$ of $q(\underline{V})$ in Appendix 3. Equation (A3.7) may be written out in full as

$$\frac{\partial G}{\partial w_i} = \int d\underline{V}\, d\underline{H}\, q(\underline{V},\underline{H})\, U_i(\underline{V},\underline{H}) - \int d\underline{V}\, d\underline{H}\, d\underline{H}'\, q(\underline{H}|\underline{V})\, p(\underline{V},\underline{H}')\, U_i(\underline{V},\underline{H}) \qquad \text{(A6.3)}$$

There are two Gibbs distributions which contribute to the gradient

(i) $q(\underline{V},\underline{H})$, and

(ii) $q(\underline{H}|\underline{V})\ p(\underline{V},\underline{H}')$

Distribution (i) was considered in Appendix 3, but distribution (ii) is new (insofar as it is now a Gibbs distribution). There are three types of variables to consider: $\underline{H}$, $\underline{V}$ and $\underline{H}'$. However the NCPDFs can be recovered from the Gibbs distributions, and the Metropolis algorithm ALONE can be used to estimate both the integrals in equation (A6.3).

Of course there is no need to use a Boltzmann machine to capture the required correlation structure (ie $p(\underline{V})$) if one has an explicit representation of $p(\underline{V})$ in terms of a Gibbs distribution. However this formulation is very useful when the "training sets" of interest can be approximated by a Gibbs distributions, for then the behaviour of Boltzmann machines can be investigated much more conveniently. Furthermore the fact that the entire training scheme can be achieved by using a Metropolis algorithm alone means that the limit distribution (equilibrium) of the conditional PDF $q(\underline{H}|\underline{V})$ does not have to be reached separately for each $\underline{V}$ (see the discussion following equation A3.7)).

TABLE 1

Table of dG/dp for various p_{RTS} and p corresponding to Figure 7.

p	P_{RTS} 0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.2	0.0	0.3	0.8	2.4	2.7	3.7	4.2
0.3	0.0	0.0	0.7	2.1	2.4	2.7	3.1
0.4	-0.5	-0.5	0.1	1.3	1.6	2.0	2.3
0.5	-1.8	-1.8	-1.1	0.0	0.2	0.5	0.9
0.6	-2.2	-2.1	-1.7	-0.3	0.0	0.3	0.6
0.7	-2.8	-2.2	-2.1	-0.6	-0.3	0.0	0.4
0.8	-4.2	-4.2	-3.5	-1.3	-0.9	-0.5	0.0

TABLE 2(a)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 8(a)

$\underline{p}_A$ = (0.7,0.3,0.7,0.3) $\underline{p}_B$ = (0.6,0.4,0.6,0.4) $\underline{p}_C$ = (0.5,0.5,0.5,0.5)

$\underline{p}_D$ = (0.4,0.6,0.4,0.6) $\underline{p}_E$ = (0.3,0.7,0.3,0.7)

$\underline{p}_{RTS}$	grad(G($\underline{p}$)) $\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$	$\underline{p} = \underline{p}_D$	$\underline{p} = \underline{p}_E$
$\underline{p}_A$	(0.0,+0.1, -0.1, 0.0)	(+0.2,-0.3, +0.5,-0.2)	(+0.6,-0.6, +1.1,-0.2)	(+1.0,-1.0, +1.4,-0.6)	(+1.6,-1.6, +1.7,-1.2)
$\underline{p}_B$	(-0.3,+0.4, -0.7,+0.1)	(-0.1, 0.0, -0.1, 0.0)	(+0.3,-0.3, +0.5,-0.2)	(+0.7,-0.7, +0.7,-0.6)	(+1.1,-1.2, +0.9,-1.4)
$\underline{p}_C$	(-0.8,+0.8, -1.3,+0.4)	(-0.4,+0.4, -0.6,+0.2)	(-0.1,+0.1, 0.0, 0.0)	(+0.3,-0.3, +0.2,-0.4)	(+0.7,-0.7, +0.3,-1.2)
$\underline{p}_D$	(-1.2,+1.2, -1.6,+0.9)	(-0.8,+0.7, -0.9,+0.7)	(-0.4,+0.4, -0.3,+0.5)	(-0.1,+0.1, 0.0,+0.1)	(+0.2,-0.3, +0.1,-0.6)
$\underline{p}_E$	(-1.4,+1.5, -1.7,+1.6)	(-1.0,+0.9, -1.0,+1.4)	(-0.7,+0.6, -0.4,+1.1)	(-0.4,+0.4, -0.1,+0.7)	(+0.1, 0.0, -0.1, 0.0)

TABLE 2(b)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 8(b).

$\underline{p}_A$ = (0.5,0.5,0.3,0.7) $\underline{p}_B$ = (0.4,0.6,0.4,0.6) $\underline{p}_C$ = (0.3,0.7,0.5,0.5)

$\underline{p}_{RTS}$	grad(G($\underline{p}$)) $\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$
$\underline{p}_A$	(0.0, 0.0, -0.2,+0.2)	(+0.4,-0.5, -1.0,+0.5)	(+1.1,-1.2, -1.9,+0.3)
$\underline{p}_B$	(-0.3,+0.3, +0.9,-0.6)	(0.0, 0.0, +0.1,-0.1)	(+0.6,-0.6, -0.9,-0.2)
$\underline{p}_C$	(-1.0,+1.0, +2.2,-0.2)	(-0.7,+0.7, +1.2,+0.2)	(-0.2,+0.1, +0.2,+0.1)

TABLE 2(c)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 8(c).

$\underline{p}_A$ = (0.7,0.7,0.3,0.3) $\underline{p}_B$ = (0.6,0.6,0.4,0.4) $\underline{p}_C$ = (0.5,0.5,0.5,0.5)

$\underline{p}_D$ = (0.4,0.4,0.6,0.6) $\underline{p}_E$ = (0.3,0.3,0.7,0.7)

	grad (G($\underline{p}$))				
$\underline{p}_{RTS}$	$\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$	$\underline{p} = \underline{p}_D$	$\underline{p} = \underline{p}_E$
$\underline{p}_A$	(0.0,-0.1, +0.1, 0.0)	(+0.4,+0.3, -0.3,-0.4)	(+1.5,+1.5, -1.4,-1.5)	(+1.9,+1.9, -1.9,-2.0)	(+2.0,+2.2, -2.8,-2.8)
$\underline{p}_B$	(-0.6,-0.6, +0.7,+0.6)	(-0.1,-0.1, +0.2,+0.1)	(+1.0,+1.0, -1.0,-1.0)	(+1.4,+1.4, -1.5,-1.5)	(+1.5,+1.6, -2.2,-2.2)
$\underline{p}_C$	(-1.9,-1.9, +1.8,+1.8)	(-1.3,-1.3, +1.3,+1.2)	(-0.2,-0.1, +0.1,+0.1)	(+0.1,+0.2, -0.3,-0.3)	(+0.1,+0.3, -0.9,-0.8)
$\underline{p}_D$	(-2.1,-2.1, +2.2,+2.2)	(-1.4,-1.5, +1.6,+1.5)	(-0.3,-0.3, +0.5,+0.4)	(-0.1, 0.0, +0.1, 0.0)	(-0.1, 0.0, -0.5,-0.4)
$\underline{p}_E$	(-2.1,-2.0, +2.8,+2.8)	(-1.5,-1.3, +2.1,+2.1)	(-0.2,-0.3, +0.9,+0.9)	(-0.2,+0.2, +0.6,+0.6)	(-0.1,+0.1, 0.0,+0.1)

TABLE 3(a)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 10(a).

$\underline{p}_A$ = (0.7,0.3,0.7,0.3) $\underline{p}_B$ = (0.6,0.4,0.6,0.4) $\underline{p}_C$ = (0.5,0.5,0.5,0.5)

$\underline{p}_D$ = (0.4,0.6,0.4,0.6) $\underline{p}_E$ = (0.3,0.7,0.3,0.7)

	GRAD(G($\underline{p}$))				
$\underline{p}_{RTS}$	$\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$	$\underline{p} = \underline{p}_D$	$\underline{p} = \underline{p}_E$
$\underline{p}_A$	(0.0, 0.0, +0.1, 0.0)	(+0.1,-0.1, +0.3,+0.1)	(+0.3,-0.3, +0.5,-0.1)	(+0.3,-0.3, +0.3,-0.4)	(+0.2,-0.2, +0.1,-0.4)
$\underline{p}_B$	(-0.1,+0.1, -0.4,+0.2)	(-0.1,+0.1, -0.2,+0.1)	(+0.2,-0.2, +0.2, 0.0)	(+0.3,-0.3, +0.2,-0.4)	(+0.3,-0.3, +0.1,-0.6)
$\underline{p}_C$	(-0.2,+0.2, -0.5,+0.2)	(-0.2,+0.2, -0.4,+0.1)	(0.0, 0.0, 0.0, 0.0)	(+0.2,-0.2, +0.1,-0.3)	(+0.3,-0.3, +0.1,-0.6)
$\underline{p}_D$	(-0.3,+0.2, -0.6,+0.2)	(-0.3,+0.3, -0.4,+0.2)	(-0.2,+0.1, -0.1,+0.2)	(0.0, 0.0, 0.0,-0.1)	(+0.1,-0.1, +0.1,-0.3)
$\underline{p}_E$	(-0.2,+0.1, -0.4,+0.3)	(-0.3,+0.3, -0.4,+0.3)	(-0.2,+0.3, -0.1,+0.5)	(-0.1,+0.2, 0.0,+0.3)	(0.0, 0.0, 0.0,+0.1)

TABLE 3(b)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 10(b).

$\underline{p}_A$ = (0.5,0.5,0.3,0.7) $\underline{p}_B$ = (0.4,0.6,0.4,0.6) $\underline{p}_C$ = (0.3,0.7,0.5,0.5).

$\underline{p}_{RTS}$	grad(G($\underline{p}$)) $\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$
$\underline{p}_A$	(0.0, 0.0, +0.2,-0.2)	(+0.3,-0.3, -0.5,+0.2)	(+0.6,-0.6, -0.7,-0.1)
$\underline{p}_B$	(-0.2,+0.2, +0.8,-0.6)	(0.0, 0.0, +0.1,-0.1)	(+0.4,-0.5, -0.4,-0.3)
$\underline{p}_C$	(-0.4,+0.4, +1.3,-0.7)	(-0.4,+0.3, +0.8, 0.0)	(-0.2,+0.1, +0.2,+0.2)

TABLE 3(c)

Table of grad(G($\underline{p}$)) for various $\underline{p}_{RTS}$ and $\underline{p}$ corresponding to Figure 10(c).

$\underline{p}_A$ = (0.7,0.7,0.3,0.3) $\underline{p}_B$ = (0.6,0.6,0.4,0.4) $\underline{p}_C$ = (0.5,0.5,0.5,0.5)

$\underline{p}_D$ = (0.4,0.4,0.6,0.6) $\underline{p}_E$ = (0.3,0.3,0.7,0.7)

$\underline{p}_{RTS}$	grad(G($\underline{p}$)) $\underline{p} = \underline{p}_A$	$\underline{p} = \underline{p}_B$	$\underline{p} = \underline{p}_C$	$\underline{p} = \underline{p}_D$	$\underline{p} = \underline{p}_E$
$\underline{p}_A$	(+0.2,+0.1, -0.4, 0.0)	(+0.4,+0.3, -0.5,-0.3)	(+0.8,+0.7, -0.8,-0.7)	(+0.2,+0.3, -0.4,-0.4)	(0.0,+0.2) -0.4,-0.4)
$\underline{p}_B$	(-0.1,-0.1, -0.1,+0.2)	(0.0, 0.0, -0.1, 0.0)	(+0.5,+0.5, -0.5,-0.5)	(+0.2,+0.3, -0.3,-0.3)	(0.0,+0.2, -0.4,-0.5)
$\underline{p}_C$	(-0.2,-0.2, +0.1,+0.2)	(-0.8,-0.8, +0.7,+0.8)	(-0.1,-0.1, +0.1,+0.1)	(0.0,+0.1, -0.2,-0.2)	(-0.1,+0.1, -0.4,-0.5)
$\underline{p}_D$	(-0.3,-0.3, +0.1,+0.3)	(-0.9,-0.9, +0.8,+0.9)	(-0.1,-0.2, +0.2,+0.2)	(0.0,+0.1, 0.0, 0.0)	(-0.2,+0.1, -0.3,-0.4)
$\underline{p}_E$	(-0.2,-0.1, +0.1,+0.2)	(-0.8,-0.8, +0.8,+0.9)	(-0.1,-0.2, +0.4,+0.9)	(+0.2,-0.1, +0.4,+0.5)	(0.0,+0.1, -0.1,-0.1)

FIGURE CAPTIONS

1. General MRF Structure

2. Composite MRF Structure

3. Boltzmann Machine Structure

4. An example of a Gibbs Machine Structure

5. Hybrid Gibbs Machine Structure

6. 8-fold neighbourhood square lattice used for generating 2D RTSs

7. Homogeneous isotropic 2D RTSs

8(a), 8(b) and 8(c).

Homogeneous mirror-symmetric 2D RTSs

9. 8-fold neighbourhood square lattice linked unidirectionally to a 0-fold neighbourhood lattice used for generating 2D speckled RTSs

10(a), 10(b) and 10(c).

Homogeneous mirror-symmetric 2D RTSs with speckle noise

REFERENCES

Bounds, D. G., 1985, RSRE Memo 3826

Cross, G. R., and Jain, A. K., 1983, IEEE PAMI-5, No. 1, 25-39

Finley, I. P., and Wood, J. W., 1985, RSRE Memo 3790

Geman, S., and Geman, D., 1984, IEEE PAMI-6, No. 6, 721-741

Hinton, G. E., Sejnowski, T. J., and Ackley, D. H., 1984, Technical Report CMU-CS-84-119 (Carnegie-Mellon University)

Kindermann, R., and Snell, J. L., 1980, Contemporary Mathematics, 1

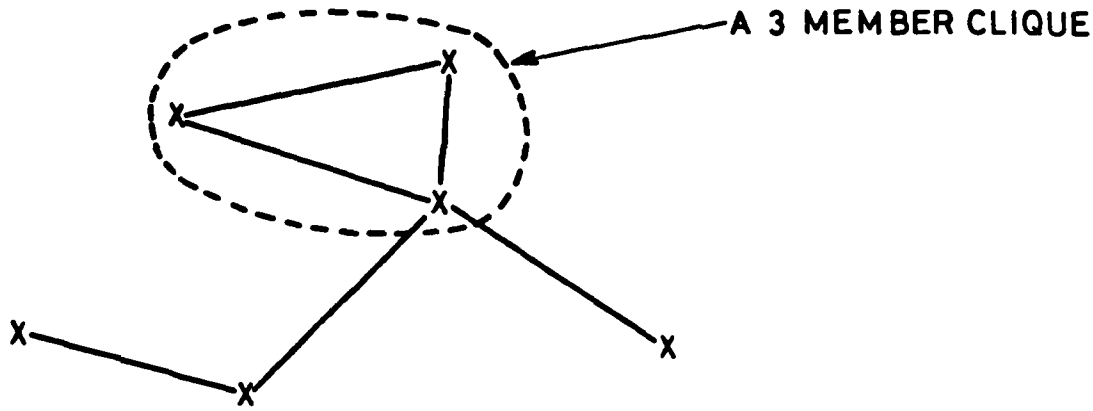
Luttrell, S. P., and Oliver, C. J., 1984, Unpublished MOD data

Luttrell, S. P., and Oliver, C. J., 1984/5 Unpublished MOD data

Luttrell, S. P., 1985, Optica Acta, 32, 703-716

Oliver, C. J., 1985, to be published in Optica Acta

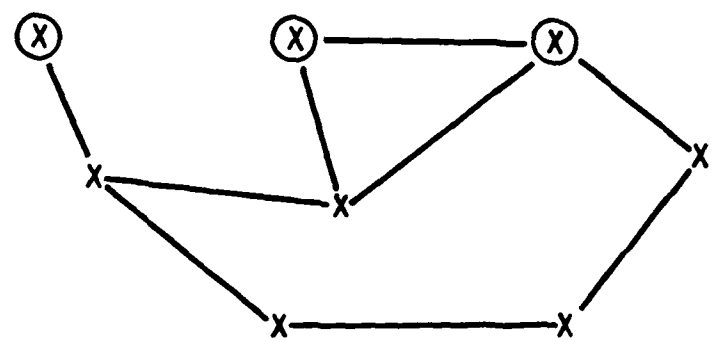
Shore, J. E., and Johnson, R. W., 1980, IEEE IT-26, 26-37



X ≡ NODE OR FIELD VARIABLE

—— ≡ LINK BETWEEN NEIGHBOURING NODES

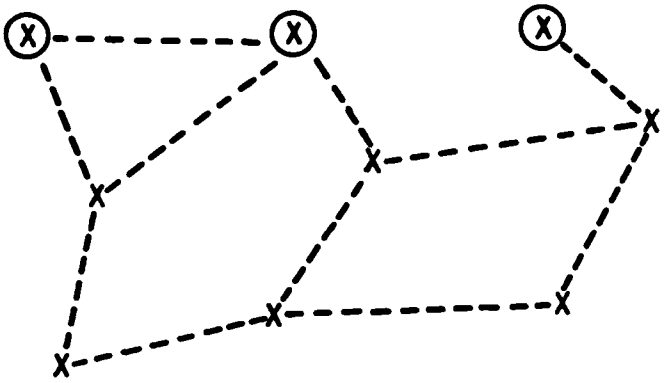
FIG. 1



Ⓧ ≡ VISIBLE FIELD VARIABLE

X ≡ HIDDEN FIELD VARIABLE

FIG. 2

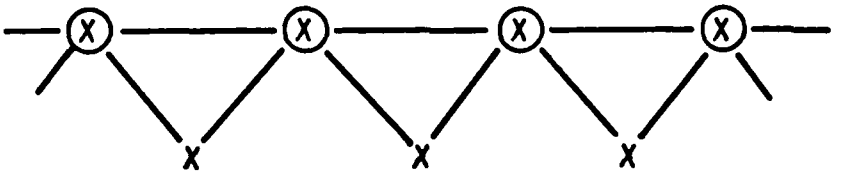


(X) ≡ VISIBLE UNIT (BINARY)

X ≡ HIDDEN UNIT (BINARY)

----- ≡ LINK BETWEEN NEIGHBOURING UNITS (QUADRATIC POTENTIAL)

FIG. 3

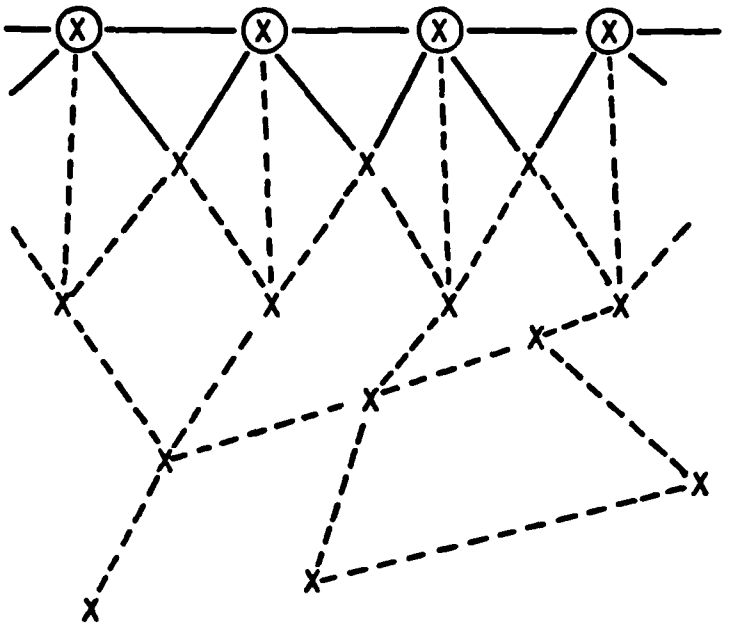


(X) ≡ PIXEL INTENSITY (VISIBLE)

X ≡ BOUNDARY INFORMATION (HIDDEN)

—— ≡ A PRIORI KNOWN LINKS

FIG. 4



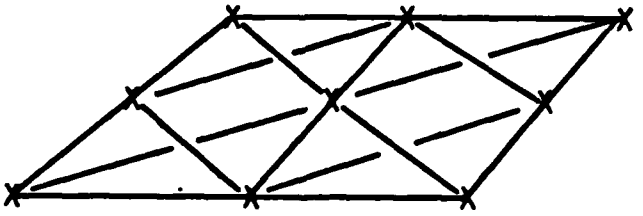
(X) ≡ VISIBLE UNIT

X ≡ HIDDEN UNIT

——— ≡ A PRIORI KNOWN LINKS

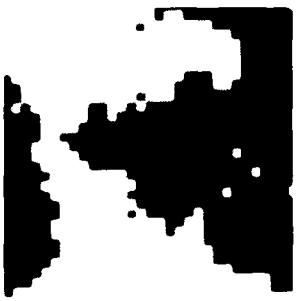
----- ≡ NON-COMMITTAL LINKS
(see FIG. 3)

FIG. 5

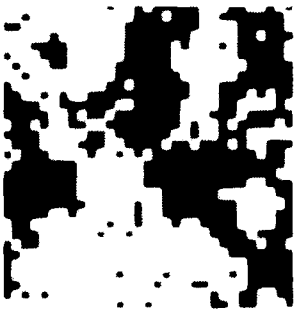


8 - FOLD NEIGHBOURHOOD
SQUARE LATTICE

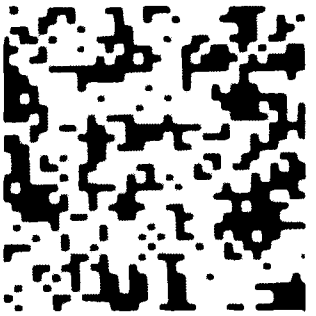
FIG. 6



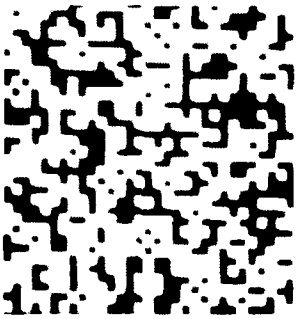
p = 0.35



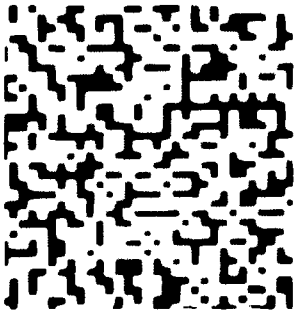
p = 0.40



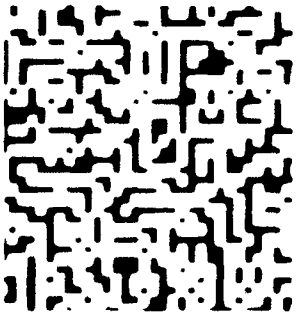
p = 0.45



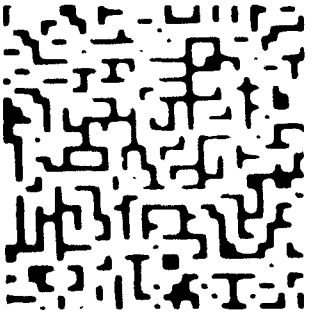
p = 0.50



p = 0.55



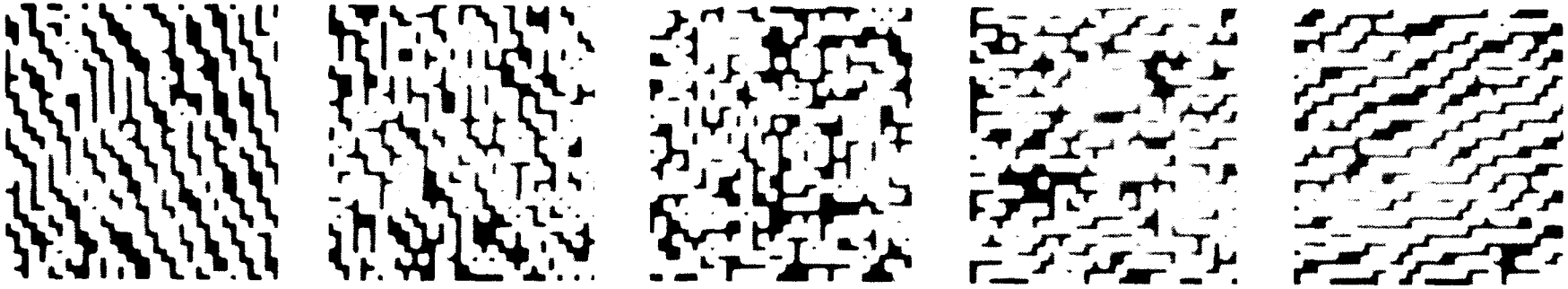
p = 0.60



p = 0.65

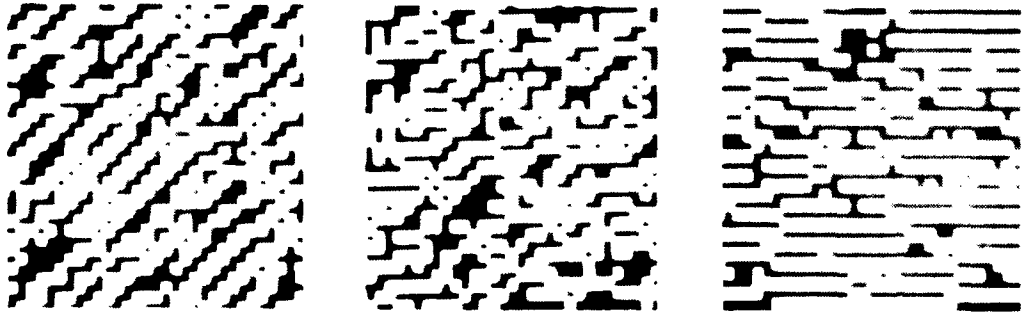
FIG 7

FIG 8a



0.7,0.3,0.7,0.3 0.6,0.4,0.6,0.4 0.5,0.5,0.5,0.5 0.4,0.6,0.4,0.6 0.3,0.7,0.3,0.7

FIG 8b



0.5,0.5,0.3,0.7 0.4,0.6,0.4,0.6 0.3,0.7,0.5,0.5

FIG 8c

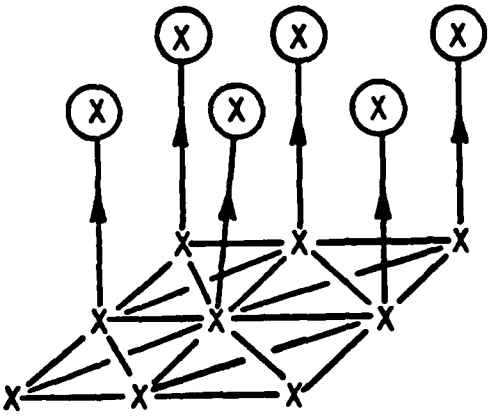
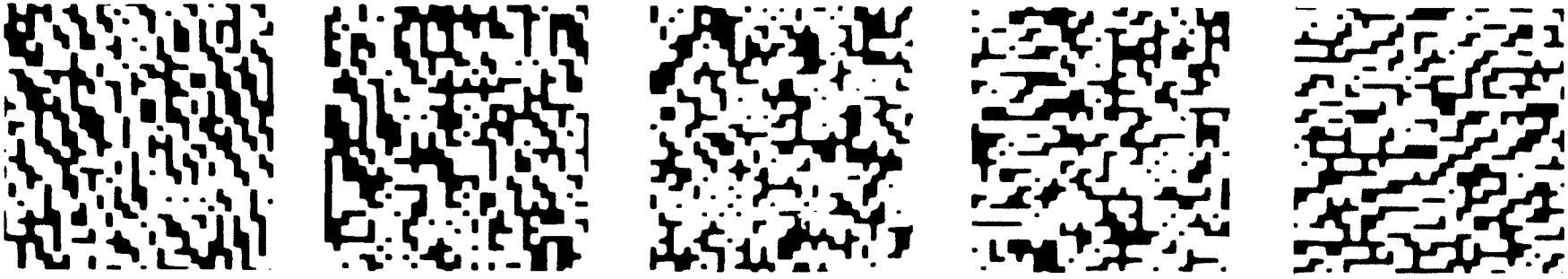


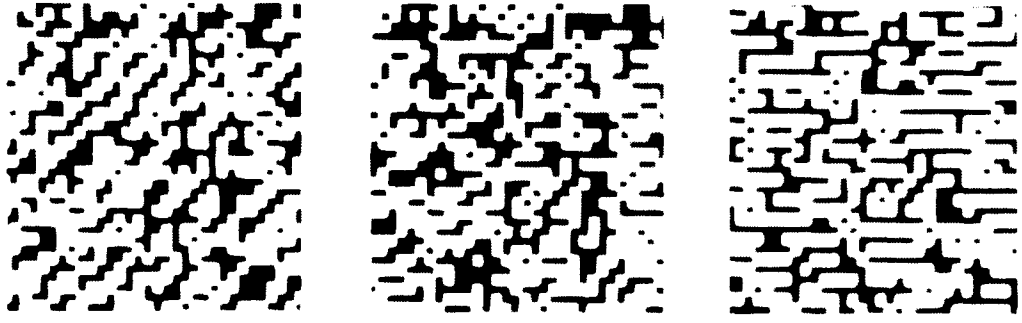
FIG. 9

FIG 10a



0.7,0.3,0.7,0.3 0.6,0.4,0.6,0.4 0.5,0.5,0.5,0.5 0.4,0.6,0.4,0.6 0.3,0.7,0.3,0.7

FIG 10b



0.5,0.5,0.3,0.7 0.4,0.6,0.4,0.6 0.3,0.7,0.5,0.5

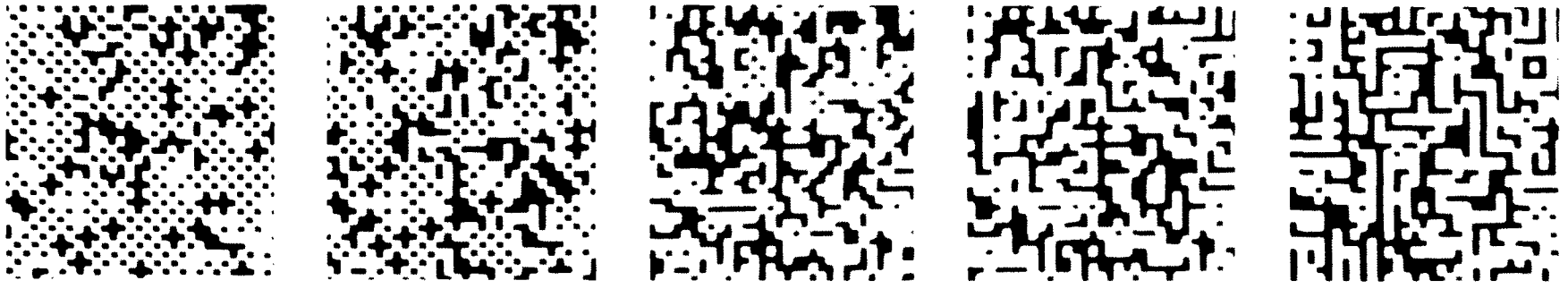


FIG 10c

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..... UNCLASSIFIED .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference MEMORANDUM 3815	3. Agency Reference	4. Report Security Classification U/C
5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS AND RADAR ESTABLISHMENT		
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location		

7. Title
THE IMPLICATIONS OF BOLTZMANN-TYPE MACHINES FOR SAR DATA PROCESSING: A PRELIMINARY SURVEY

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.
LUTTRELL S P				

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement
UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract

We propose that Markov random field models (MRFs) be used as a framework within which to construct models of synthetic aperture radar (SAR) images. We clarify the relationship between this class of models and the Boltzmann machine (BM) of artificial intelligence. We then generalise the BM training procedure and use it to train MRF models. Using this technique we investigate the ability of a simple MRF texture model to learn a texture by maximising a relative entropy objective function. We find that the marriage of MRF models with the BM training procedure is fruitful.

END

FILMED

3-86

DTIC