

# Neural Network Based Learning of Local Compatibilities for Segment Grouping

D. Rivière<sup>1,2</sup>, J.F. Mangin<sup>1</sup>, J.M. Martinez<sup>2</sup>, F. Chavand<sup>3</sup>, and V. Frouin<sup>1</sup>

<sup>1</sup> Service Hospitalier Frédéric Joliot, Commissariat à l’Energie Atomique, Orsay, France

<sup>2</sup> Service d’Etudes des Réacteurs et de Mathématiques Appliquées, Commissariat à l’Energie Atomique, Saclay, France

<sup>3</sup> Université d’Evry CEMIF Systèmes complexes, Evry, France

**Abstract.** This paper addresses the automatic inference of a Gibbs distribution dedicated to segment grouping through relaxation labeling. The behavior of this method is studied through the detection of a road-like network from a noisy set of segments extracted from an image during a preprocessing step. Linking segments are added to this set to recover lost road parts. The whole segment set is organized in a relational graph and the road network restoration is modeled as a labeling process. The solution is defined as the labeling maximizing a Gibbs distribution constructed from a set of local costs computed for each graph clique. These cost functions, corresponding to interaction potentials, are learned automatically using multi-layer perceptrons. Supervised learning is performed over a training data set using only binary teaching output, “good” or “bad” configuration example. Several neural networks are used to overcome the problem of the variable complexity of clique configurations.

## 1 Introduction

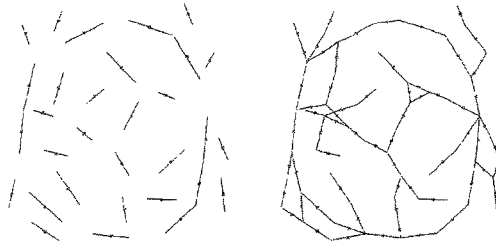
Artificial vision systems usually include a hierarchy of processing units. The lowest levels deal with local feature detection while the highest ones are in charge of pattern recognition. In this paper we address an intermediate level of processing which consists in correcting errors of the lowest level of processing using contextual information. The paper focuses on the detection of a road network from a set of segments extracted from the image but the approach could be extended to other similar problems.

Various approaches have been proposed in literature to address this kind of problems [3]. In this paper, a method of relaxation labeling based on a Markovian random field model is discussed [1]. Such a model has been recently proposed in [8] to overcome the difficulties induced by speckle noise in SAR images. Local contextual a priori information is embedded in interaction potentials used to recover the full road network from a very noisy set of segments. These potentials rely on different parameters which are set according to ad hoc considerations. In this paper we propose a general scheme which consists in using multi-layer perceptrons to construct similar potentials from a supervised learning on a set

of manually labeled road networks. Different simulations are proposed to show the efficiency of this strategy.

## 2 Segment Graph

In this section we discuss how we build a graph from an image. We suppose that a set of segments has been extracted from images (see fig. 1. Let us assume that roads can be recovered through addition of new segments linking extracted ones, and deletion of other segments stemming from other image structures. Hence, a set of new connection segments is added to the initial set following simple rules. A segment is added between two others if the distance between them is not too long and the angle they are forming is not too large. In the following a weight parameter is attached to each segment: high (1) if it is part of the former set, low (0) if it is a new connection segment. A finest weight set could stem from an evaluation of the segment detector. Once all segments have been obtained, a segment graph can be built, with each node representing a segment, and relations between nodes showing if nodes share an extremity. The cliques of this graph are sets of segments which share the same extremity. Note that configurations where three segments form a triangle clique can exist but are not used for potential definition.



**Fig. 1.** Connection of segments. Each segment is marked here by a dot on its center to allow better visualization

## 3 Interaction Potentials

We now address the problem of labeling the segment graph in order to recover the road network (1 for road, 0 for non road). We assume that local contextual a priori information is sufficient to solve this problem. Therefore, the solution is defined as the labeling corresponding to the maximum of a Gibbs distribution. This distribution is inferred from a set of interaction potentials attached to each maximal clique of the segment graph:  $p(l) = \frac{1}{Z} e^{-\sum_{c \in C} V[c](l)}$ , where  $l$  is a labeling,  $C$  the clique set,  $V[c]$  the potential of clique  $c$ , and  $Z$  a normalization

constant. Hence the problem reduces to the minimization of a sum of potentials acting as cost functions. Triangle cliques are endowed with 0 potential and do not influence the distribution. Now we have to define cost functions which will model the local nature of a road network.

The system has to know that a road is most of the time a low curvature line, that crossings (ie cliques involving more than two nodes with label 1) are unusual but do exist, and that end points (cliques with only one node with label 1) are rare (in fact they usually only happen at the edges of the image). So a cost function will be built, that will favour usual situations by giving them a low cost, and penalize unusual configurations by assigning them a high cost. The cost function has to be bounded because improbable situations *can* happen and must be allowed even if penalized. This cost function can be regarded as the opposite of a plausibility. One way of defining it is to try to put our human knowledge in the system, by modeling. This was the choice of Tupin [8]. But this implies humans are able to teach the system a good model. In fact only very general rules, like those given above, can be modeled this way. Tupin used a model driven by four parameters which were adjusted by hand. The final result was good but it could not be used for other applications since the rules and the parameters would not be appropriate. Another approach is to build an adaptive system which will learn to decide if a given local configuration is or is not plausible and then output a cost value. This is the choice we made in the present work.

## 4 Learning Interaction Potentials

Each segment of the graph can be described by its local characteristics (for instance its length, coordinates of center point, orientation, weight and label). The characteristics of all segments of a clique have to be put together into a black box to compute the cost, or potential of the clique. Such a black box can be filled with a lot of statistical methods since the output cost can be related to a probability density.

So far we used a supervised method, the common multi-layer perceptron, trained with a standard back-propagation algorithm [7,6]. We consider that all clique configurations in manually labeled segment graphs are “good” and systematically use 0 (low value cost) as teaching output for these cliques. We provide also bad examples which are obtained by randomly changing labels of good clique configurations. High cost value (1) is used as teaching output for these examples. Giving such extreme binary teaching outputs might seem inappropriate, but in fact “bad” examples balance “good” ones so that in frontier regions the cost output will be intermediate.

An important problem lays in coding the input to the black box. Kim and Yang [2] used a similar approach for the labeling of an image over-segmentation, but only pair interaction of adjacent regions were considered. Here the data is a clique configuration which is described by a large set of parameters. The number of parameters is not the same from one clique to another, depending on the number of segments involved in it. Some of these parameters make more sense

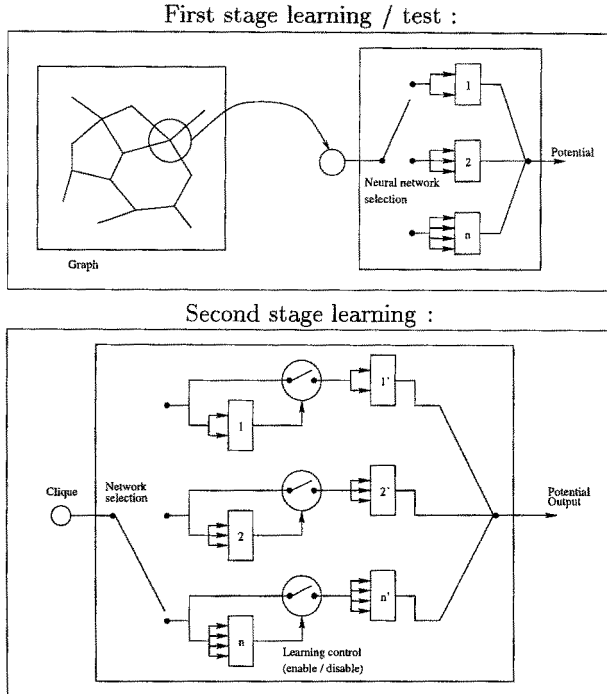
than others, and some will be better used if given in another formulation. Our choice was to build several neural networks to overcome the problem. Indeed, attempts to summarize all cases in a constant number of parameters led to bad results. As a general principle, the number of input parameters must be as small as possible: for our problem, absolute coordinates of the segments in the clique are useless: only relative positions (angles for example) are meaningful. This has the advantage of making the representation invariant in translation and rotation. Giving too many inputs to a neural network results in useless computation, slower and even less efficient learning. In our case important information stem from segments whose labels are 1, because these segments are potential roads. Therefore information stemming from label 0 segments will be used only to evaluate fully “no road” clique configurations.

We constructed first one network for each configuration set corresponding to a fixed non zero number of “road” segments (see fig. 2). This can also be seen as separating syntactically different cases: “end point”, “no crossing”, “T-crossing”, “X-crossing” and so on. The corresponding inputs are, for each “road” segment, its length and weight. The labels have not to be specified because “no road” segments are forgotten. Angles between segments are also provided. Segments are described in a consistent order: clockwise for example, starting at random in the learning phase, and with the longest first during the minimization process which allows a consistent potential definition. The last configuration set corresponding to a no road configuration requires a different construction. We define a set of networks similar to previous ones but defined for a fixed “no road” segments number. As a matter of fact, if all segments have the label 0 their relative organization may be more important than in the former case. Indeed, if two segments share a similar direction, a road could have been missed, and the configuration should be penalized.

Dimensioning the neural networks has been done iteratively by trying different numbers of hidden units, learning and testing them on different data sets. The networks which obtained the lowest total sum of clique potentials on the test data set were kept. Networks with 1 or 2 hidden layers were tried. Depending on the network (and its number of inputs) networks with either one hidden layer of about 10 neurons or two hidden layers of about 5-8 neurons on the first hidden layer and 2-5 on the second had the best results for our application.

The way we create “bad” configuration cliques for training introduces a bias in the learned cost function: sometimes realistic configurations appear this way and should be good. Then “good” configurations are reliable, but “bad” configuration sets can include errors. To overcome this we made a two-stage learning. After a first training, another set of equivalent neural networks is trained with the same data set, from which “bad” configurations evaluated “good” by the first networks are dropped out.

Another point is that in this work learning had to be done with a few simulations. This implies that the configuration space is not well sampled. To allow good generalization, the sample set is noised differently during each training iteration.



**Fig. 2.** (up): From cliques to local potentials: the appropriate neural network is chosen depending on the complexity of the clique configuration. (down): Second stage learning scheme: a first network set is used to control learning of a second set. When the first network disagrees with the teaching output for a “bad” configuration, learning is disabled for this clique

As an adaptive process, the system can be specialized for different patterns. For the road detection application for instance mountain roads allow more curvature but fewer crossings than urban roads. To specialize the system, using an appropriate data set for learning is sufficient. Moreover another system could be specialized to river detection: rivers have their own characteristics which are a bit different from roads. Then competition between a road system and a river system could be used as a recognition tool to discriminate between roads and rivers, which may all be mixed up by Tupin’s model.

## 5 Global Detection: Simulated Annealing

The optimal labeling for the whole graph should include mainly low cost configurations. A clique is usually endowed with several low cost configurations which are discriminated through the search for a global optimum. The optimal labeling corresponding to the minimal energy configuration of the underlying random field is obtained using simulated annealing. The state space is explored using the following topology: during each iteration, the segment set is randomly

partitioned in singletons, pairs and triplets. Then segment subsets are sequentially updated using the Gibbs sampler algorithm [1]. The annealing temperature follows a geometric decreasing with 0.965 reason. Using this fast approximate of the simulated annealing scheme which does not wait for stabilization at a constant temperature, convergence was obtained after about a minute calculation time on a Ultra-1 SUN calculator.

## 6 Results

The behavior of the method described above has been studied using simulated road networks. Two sets of data were used. The first only represented closed curves without any crossings. Learning was performed on five different images (see fig. 3), using 1000 learning iterations for each of the two steps. The generalization results on other images of the same type (closed curves) were good: in most cases a closed curve was obtained after detection. In some cases it was not exactly the same as the expected one, but the proposed one might have been good (see fig. 4). The energy of the proposed solution gives an idea of the performance of the system: the lower the final energy, the best the solution is. On trained graphs of about 150-200 cliques, the final energy was generally less than 2 (on a maximum equal to the number of cliques, since each had an energy between 0 and 1). On test graphs of approximately the same size the energy was between 1 and 7.

However a problem sometimes happened when testing graphs containing some atypical and incorrectly evaluated data (generally shorter or longer segments than those in the training set): the system sometimes ended with a graph with all zero label segments, but with regard to the learned data these examples were out of the scope of the learned problem.

The second set of images we tried were representing grids which can figure a urban road network in a part of a city. In this case roads were quite straight, there were X-crossings, no T crossings, and end points on the edges of the images. This trial obtained encouraging results too, although not as perfect as for simple closed lined, because too many end points were allowed in the learning data (see fig. 5). Perhaps it would improve to add an additional input to all neural networks providing the number of label 0 nodes in the clique to allow end points only on the edges of images (where cliques actually contain a single node).

## 7 Conclusion

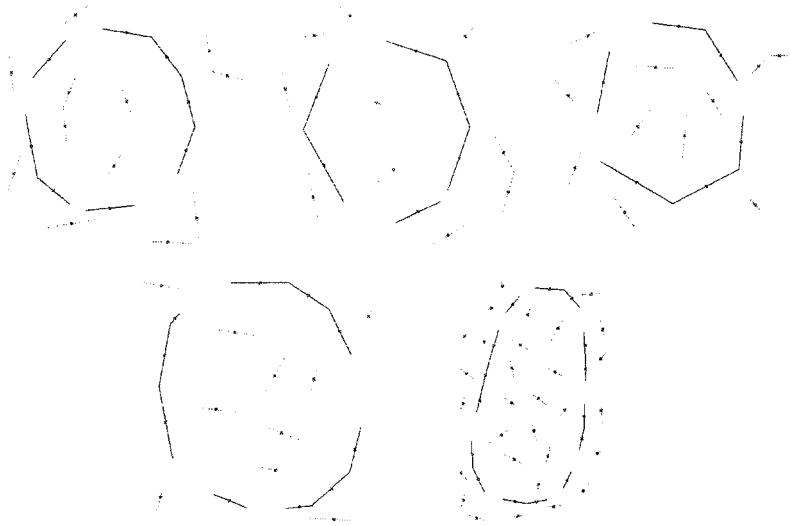
The method we have presented here is an extension of the road detection system of Tupin [8] used on SAR images. The main improvement we proposed is a learning system for local compatibilities which allows more flexibility on the applications. Our system uses multi-layer perceptrons but other adaptive techniques could be used instead. Several networks had to be used to deal with the coding of cliques with different structures. First experiments are especially encouraging. In the future, we hope to improve the discrimination power of the

system by adding for each network a self-evaluation of the reliability of the cost through the configuration space. We could also increase the number of neural networks using other configuration codings. Each clique configuration would then be evaluated by several experts, a supervisor making the final decision.

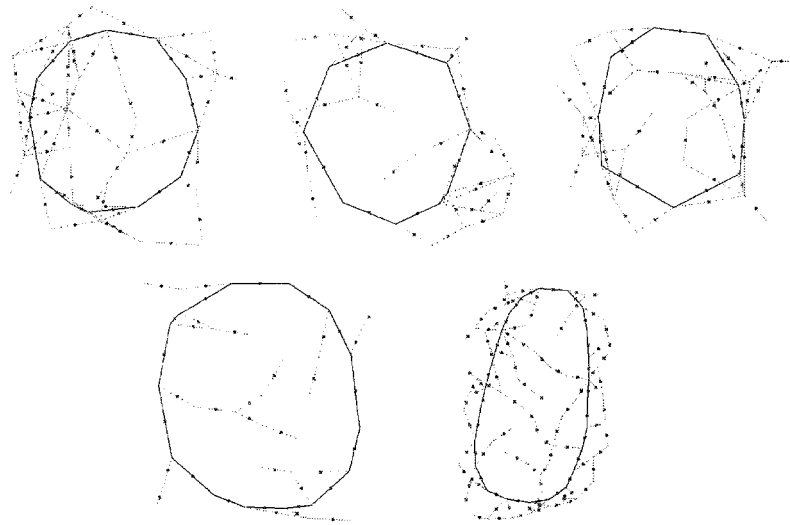
We plan to use similar approach with 3D networks stemming from medical images. For instance this approach could be used to recover the vascular system from angiographic data or to identify cortical folds from MRI data [5, 4].

## References

1. S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE PAMI*, 6:721–741, 1984.
2. I.Y. Kim and H.S. Yang. Efficient image labeling based on Markov random field and error backpropagation network. *Pattern Recognition*, 26(11):1695–1707, 1993.
3. J. Kittler and J. Illingworth. Relaxation labeling algorithms - a review. *Image Vision Comput.*, 3(4):206–216, 1985.
4. A. Manceaux-Demiau, J.-F. Mangin, J. Régis, O. Pizzato, and V. Frouin. Differential features of cortical folds. *CVRMed-MRCAS'97 LNCS 1205*, Springer Verlag, pages 439–448, 1997.
5. J.-F. Mangin, J. Régis, I. Bloch, V. Frouin, Y. Samson, and J. Lopez-Krahe. A MRF based random graph modelling the human cortical topography. *CVRMed'95, LNCS 905*, Springer, pages 177–183, 1995.
6. W. S. McCulloch and W. A. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
7. F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
8. F. Tupin, H. Maitre, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky. Linear feature detection on SAR images: Application to the road network. *IEEE Geoscience and Remote Sensing, in press*, 1997.

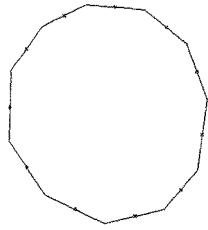


Learning database for the closed line case, extracted segments

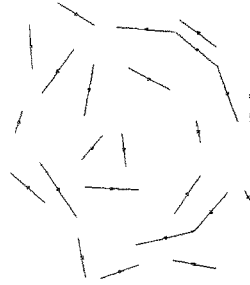


**Fig. 3.** Learning database for the closed line case, full segment set, black: road, gray: noise or connection

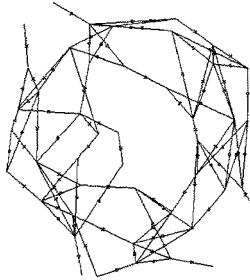




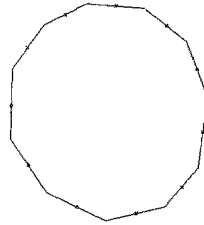
Ideal image



Segmented image

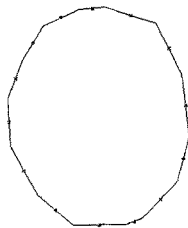


Graph with linking additions

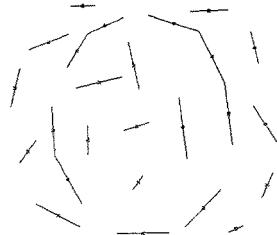


Final graph

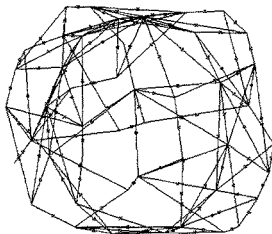
Closed line generalization results, example 1. Detection result is identical to the ideal model.



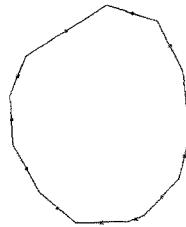
Ideal image



Segmented image



Full graph



Final graph

**Fig. 4.** Closed line generalization results, example 2. Detection result differs slightly from the ideal model, but is close to it, and a closed curve is obtained, confirming the model has learned a general principle.

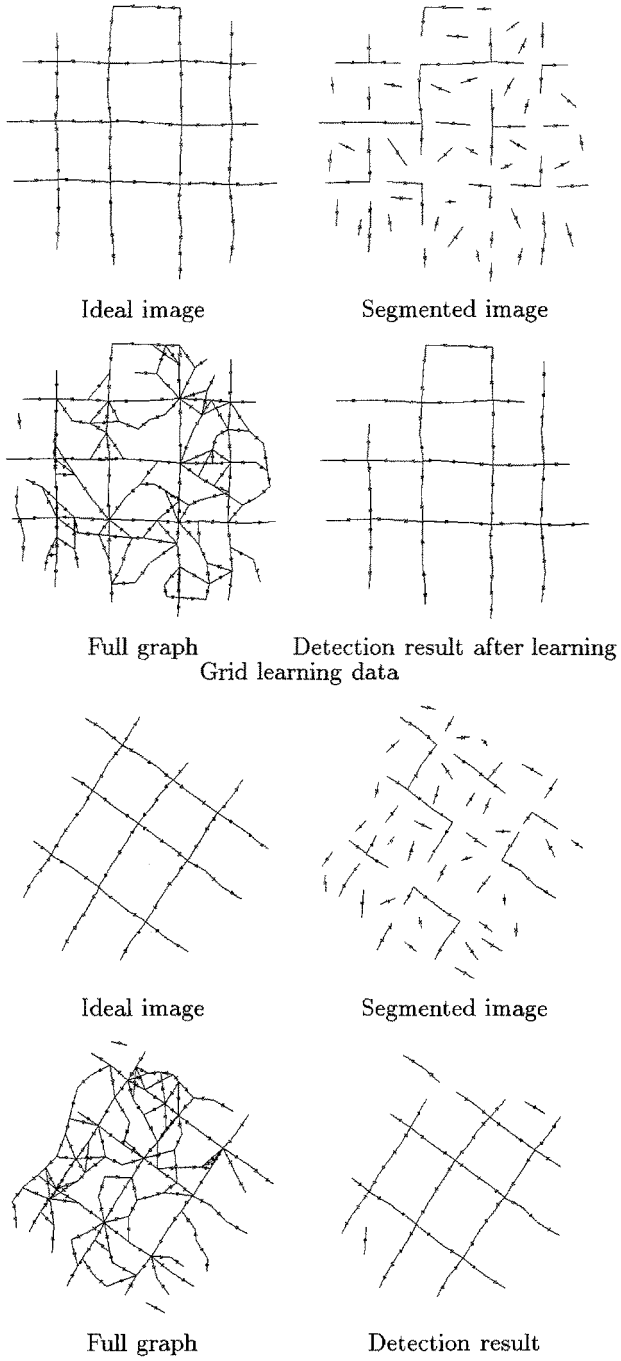


Fig. 5. Grid generalization example. Learning has been performed on a single image (up image). (down) generalization result.