

Supervised Segmentation of Volume Textures Using 3D Probabilistic Relaxation

Matthew Deighton¹ and Maria Petrou^{1,2}

¹ School of Electronics and Physical Sciences, University of Surrey,
Guildford, GU2 7XH, UK. <http://www.ee.surrey.ac.uk/CVSP/index.html>

² The Institute of Telematics and Informatics, EKETA,
PO Box 361, Thessaloniki, Greece. petrou@iti.gr

Abstract. An iterative 3D probabilistic relaxation scheme has been developed for assigning labels to voxels based on the probabilities that the voxel belongs to each one of a number of known classes. The approach takes account of the probabilities of the neighbouring voxels belonging to each class and of the likely configurations of those labels within the neighbourhood. We apply the approach to the supervised segmentation of a seismic volume. In the example, the probability that a voxel belongs to each class is provided by the application of gradient operators and statistical measures. The iterative relaxation scheme then assigns the most appropriate label to each voxel.

1 Introduction

Volumetric data are now more widespread than ever before. Medical imaging techniques such as MRI and PET routinely generate data of this form along with other fields such as seismic imaging of the Earth's crust. Often, the aim of collecting the data is to identify distinct objects or regions within a volume based on some attribute and relate it to the real world. Visualising the data in such a way to allow a human operator to carry out this task is difficult. A common approach is to first segment the volume to allow better representation of the regions of interest for visualisation purposes. Initially, 2D image processing techniques were applied to individual slices from the data. However, as computing power has increased, the data are coming to be treated as a volume. This has driven the extension of many 2D techniques to 3D and the generation of new volumetric methods.

Nikolaïdis and Pitas in [1] give illustrations of a number of techniques in this area, including filtering, 3-D DFT, segmentation, edge detection and registration.

In this paper, we propose a technique for supervised segmentation based on the texture in the neighbourhood of a voxel, using probabilistic relaxation.

2 Methodology

Kovalev et al [2] proposed the Gradient Density (GD) method as a global texture descriptor. They constructed the 3D orientation histogram of the gradient vec-

tors of all the voxels of the texture and extracted features from this histogram. We apply this method here in a 3D neighbourhood around each voxel and extract features that are assigned to the central voxel. Our features correspond to those used in [2]; namely anisotropy measure, integral anisotropy measure and local mean curvature. We extend the feature set to also include skew and kurtosis. On the basis of the values of these five features the central voxel is assigned a similarity value with one of the texture classes in the database of textures we are interested in segmenting the image into. The similarity measure is computed as

$$S_{ik} = 1 - \frac{1}{N} \sum_{n=1}^N \frac{|F_{kn} - F_{in}|}{|F_{kn}| + |F_{in}|} \quad (1)$$

where N is the number of features extracted from the 3D orientation histogram, F_{kn} is the n th feature value computed from the k th model texture in the library of textures we have, F_{in} is the n th feature value computed from the neighbourhood around voxel i . The values of S_{ik} are in the range $[0, 1]$ and so they can be used as initial probabilities with which voxel i may be assigned to class k . This similarity measure considers each voxel separately, and it does not take into consideration the labels of neighbouring voxels. Such labelling methods may be locally optimal, but globally inconsistent. We propose to use probabilistic relaxation to assign the most probable label to each voxel, given its texture properties **and** those of its neighbouring voxels.

Our 3D method closely follows the formalism of Mirmehdi and Petrou [3], who developed a probabilistic segmentation scheme for 2D images.

Let us call x_i the measurement associated with the i th voxel of a data set, with $i = 1, \dots, N$ where N is the number of voxels in the data set. For clarity we define two sets, N_0 and N_i such that $N_0 \equiv \{1, 2, \dots, N\}$ and $N_i \equiv \{1, 2, \dots, i - 1, i + 1, \dots, N\}$.

A label set Ω is defined such that $\Omega = \{\omega_1, \dots, \omega_m\}$ where m corresponds to the number of classes to which the data are to be segmented. Each voxel is assigned a label θ_i which can take any of the values from the set Ω .

Our aim is to select the most probable label θ_i for each voxel i , by considering all the information available. This can be formalised by

$$\theta_i = \arg \left\{ \max_k P(\theta_i = \omega_k | x_{j,j \in N_0}) \right\} \quad (2)$$

Using Bayes rule, we can re-write the conditional probability of a particular label assigned to a particular voxel as

$$P(\theta_i = \omega_k | x_{j,j \in N_0}) = \frac{P(\theta_i = \omega_k, x_{j,j \in N_0})}{P(x_{j,j \in N_0})} \quad (3)$$

After some lengthy manipulations, this equation can be written as

$$P(\theta_i = \omega_k | x_{j,j \in N_0}) = \frac{P(\theta_i = \omega_k | x_i) Q(\theta_i = \omega_k)}{\sum_{\omega_{\theta_i}} P(\theta_i = \omega_{\theta_i} | x_i) Q(\theta_i = \omega_{\theta_i})} \quad (4)$$

where function $Q(\theta_i = \omega_{\theta_i})$ expresses the support a particular label for the voxel in question receives from all other voxels. It is given by

$$Q(\theta_i = \omega_{\alpha}) = \sum_{\omega_{\theta_1}} \cdots \sum_{\omega_{\theta_{i-1}}} \sum_{\omega_{\theta_{i+1}}} \cdots \sum_{\omega_{\theta_N}} \prod_{j \in N_i} P(\theta_j = \omega_{\theta_j} | x_j) P(\theta_1 = \omega_{\theta_1}, \dots, \theta_i = \omega_{\alpha}, \dots, \theta_N = \omega_{\theta_N}) \quad (5)$$

From (5), we observe that the $Q(\theta_i = \omega_{\alpha})$ factor involves summing over all possible label configurations of all voxels in the data. If we assume that we have a mere 1000 voxels and 4 possible labels, the configurations over which we have to sum are 4^{1000} . This is clearly impractical.

Factor $P(\theta_1 = \omega_{\theta_1}, \dots, \theta_i = \omega_{\alpha}, \dots, \theta_N = \omega_{\theta_N})$ in (5) however, can help as it expresses the joint probability for each one of those label configurations to exist. Most of the label configurations are very unrealistic and we can assign them zero probability of existing. In addition, we realise that when we concentrate our attention to voxel i , the labels that matter are the labels of its immediate neighbours, while all other voxels could have any label they like, with virtually no influence on voxel i . We group therefore all possible configurations of labels to a few “classes”³, each “class” being characterised by the combinations of labels around voxel i . If we restrict our attention to a neighbourhood of $3 \times 3 \times 3$ voxels around voxel i , we have 26 voxels whose labels matter. With 4 possible labels, the number of configurations is still huge: 4^{27} . However, most of these “classes” are unrealistic and they can be given zero joint probability to exist. In addition, “classes” of configurations that involve more than two labels in the $3 \times 3 \times 3$ neighbourhood around a voxel are extremely rare. Without introducing much error, we can assume that these “classes” also have zero probability to exist. Then we may assemble a dictionary of allowable configurations around a voxel, assuming that, at most, two different labels will be present there.

The entries of this dictionary are created using common sense and the understanding that by and large, the world is smooth. We show the various allowable options of 3D 2-label configurations in figure 1. Most of these options represent multiple cases which can be generated by considering all possible symmetries of a $3 \times 3 \times 3$ cube. Note that each such configuration represents a whole “class” of possible labellings of the whole data cube, as for each entry in the dictionary the voxels outside the $3 \times 3 \times 3$ volume are allowed to have all possible combinations of labels they can take.

We assume that all “classes” of configurations are equally probable and so we can write

³ We use the term “class” here to distinguish it from the term class which refers to the labels we assign to each voxel during the classification process.

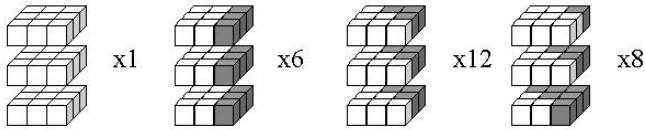


Fig. 1. Dictionary of configurations for Q-Function. There are 27 entries in the dictionary and 6 label pairs for the 4 label problem, with each label combination appearing twice: white-black and black-white. This gives a total of 324 possible combinations.

$$P(\theta_1 = \omega_{\theta_1}, \dots, \theta_i = \omega_{\alpha}, \dots, \theta_N = \omega_{\theta_N}) = \begin{cases} 0 & \text{if } (\omega_{\alpha}, \omega_{j,j \in S_i}) \notin D \\ \frac{1}{S} & \text{otherwise} \end{cases} \quad (6)$$

where S_i is the $3 \times 3 \times 3$ neighbourhood around voxel i , $(\omega_{\alpha}, \omega_{j,j \in S_i}) \equiv \Omega_{S_i}$ is the combination of labels of this neighbourhood according to the assignment ($\theta_1 = \omega_{\theta_1}, \dots, \theta_i = \omega_{\alpha}, \dots, \theta_N = \omega_{\theta_N}$), D is the dictionary of allowable combinations of labels for the $3 \times 3 \times 3$ neighbourhood and S the total number of entries in the dictionary. We see that (5) then becomes

$$Q(\theta_i = \omega_{\alpha}) = \sum_{\Omega_{S_i} \in D} \prod_{j \in S_i} P(\theta_j = \omega_{\theta_j} | x_j) \quad (7)$$

Having ignored the state of all other labels apart from those in the immediate neighbourhood of a voxel when we label it, and having done that for each voxel as if it were on its own, we may not produce a globally consistent labelling. Each voxel was allowed to be influenced by its neighbours and those by their own neighbours and so on. We may now correct for the gross classification of all possible labellings in a small number of “classes” by repeating the process in an iterative scheme. (4) and (7) now become

$$P^{n+1}(\theta_i = \omega_k) = \frac{P^n(\theta_i = \omega_k) Q^n(\theta_i = \omega_k)}{\sum_{\omega_{\theta_i}} P^n(\theta_i = \omega_{\theta_i}) Q^n(\theta_i = \omega_{\theta_i})} \quad (8)$$

$$Q^n(\theta_i = \omega_{\alpha}) = \sum_{\Omega_{S_i}^n \in D} \prod_{j \in S_i} P^n(\theta_j = \omega_{\theta_j}) \quad (9)$$

For $n = 1$ the probabilities with which each voxel is assigned each one of the possible labels are given by the similarity measure we computed using equation (1).

3 Experiments

3.1 Analysis of the Problem

Figure 2 shows our test volume of interest and figure 3 shows the texture classes to which the test volume is to be segmented. Note that the texture classes of

figure 3 have not been extracted from the test data. They are samples taken from a publically available seismic survey of the Barents Sea originally conducted by Shell. These volumes contain textures corresponding to geologically meaningful classes identified by highly trained interpreters.

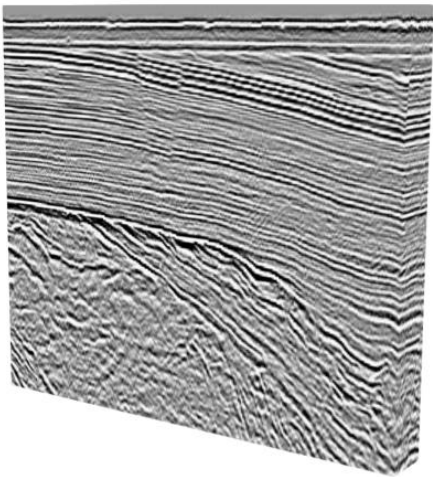


Fig. 2. Test volume

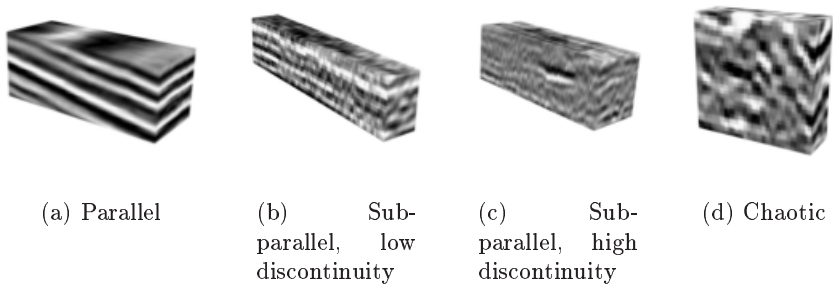


Fig. 3. Regions of interest

Figure 4 shows some slices from the test volume with the initial probabilities produced from the texture similarity measure, one output per texture class. The brighter the grey value in the display, the more probable it is that the voxel belongs to the indicated class.

Figure 5(a) shows the same slice segmented with each voxel given the label of the highest similarity. We see that this segmentation is not very “clean”, in the sense that it does not produce volumes the shapes of which may be described in a meaningful way. Applying probabilistic relaxation produces a much cleaner segmentation. Figure 5(b) shows the same slice segmented after 40 iterations of the probabilistic relaxation scheme.

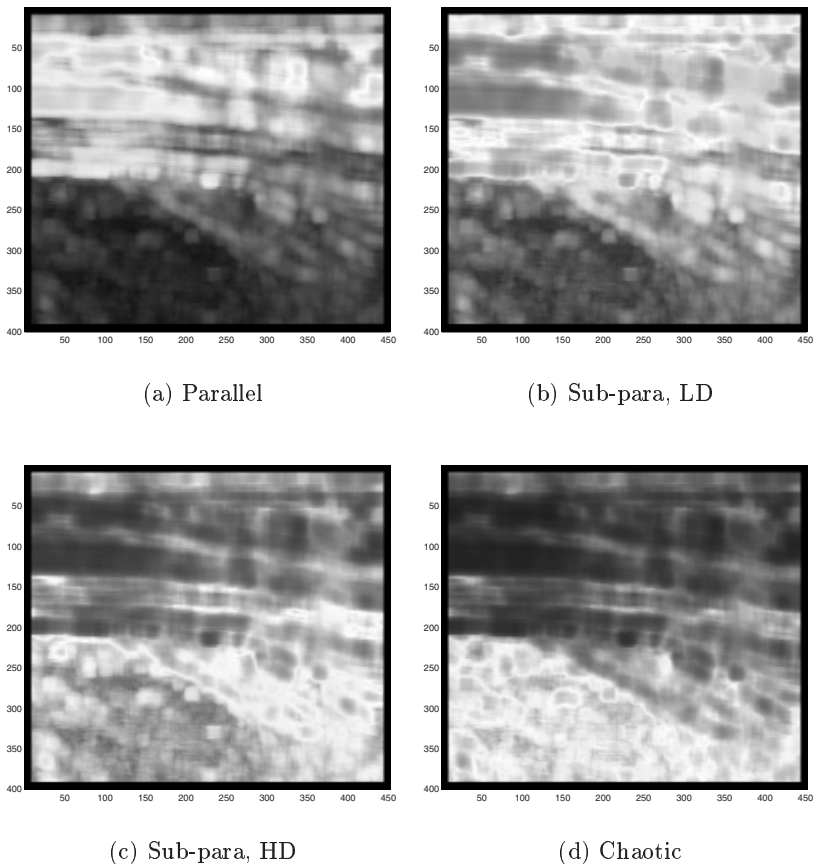


Fig. 4. Slices from Barents test volume similarity cubes; white - high similarity, black - low similarity

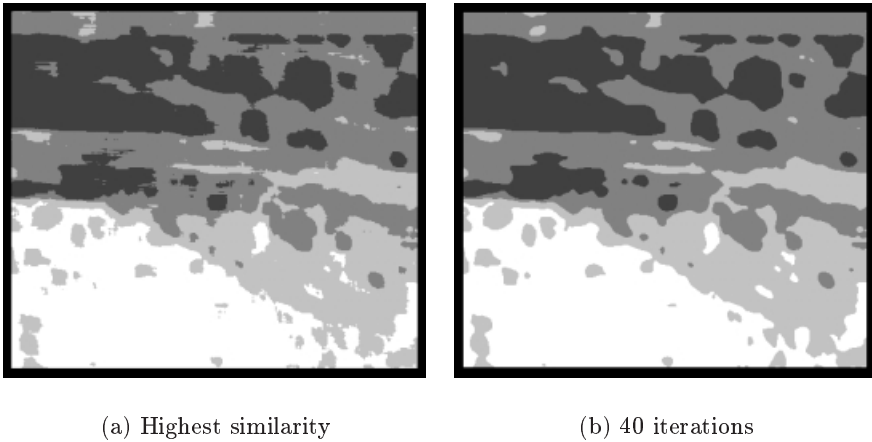


Fig. 5. Centre slices from label cubes. Black - parallel; dark grey - sub-parallel, low discontinuity; light grey - sub-parallel, high discontinuity; white - chaotic.

4 Discussion and Conclusions

We have shown how a texture measurement scheme can be combined with probabilistic relaxation to provide a clean volumetric segmentation with a seismic data example. Producing clean texture volumes is very important for the further analysis of seismic datacubes, because it allows us to identify geologically meaningful structures on the basis of their texture and shape. The probabilistic relaxation scheme is very slow, but given that such volumes usually take months to be analysed by geologists, a few hours of processing time that helps them identify the geologically meaningful regions automatically is not a significant drawback.

5 Acknowledgements

This work has been carried out under the European Commission, Directorate-General Information Society project IST-1999-20500. Many thanks to Schlumberger Stavanger Research and Seismic Area C (Norsk Hydro, Statoil, Norsk Agip and Fortum Petroleum) for their provision of data. The 3D rendering tools used to visualise the data volumes were provided by Fraunhofer Institute for Applied Information Technology.

References

1. N. Nikolaidis and I. Pitas, *3-D Image Processing Algorithms*, John Wiley and Sons, 2001.

2. V. Kovalev, M. Petrou and Y. Bondar, "Texture anisotropy in 3-D images", *IEEE Transactions on Image Processing*, Vol. 8, No 3, Pages 346–360, 1999.
3. M. Mirmehdi and M. Petrou, "Segmentation of colour textures" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 142–159, Vol. PAMI-22, No. 2, 2000.