Comparing Probabilistic and Geometric Models On Lidar Data

Roberto Fraile and Steve Maybank
The University of Reading, UK

September 2001

Abstract

A bottleneck in the use of Geographic Information Systems (GIS) is the cost of data acquisition. In our case, we are interested in producing GIS layers containing useful information for river flood impact assessment.

Geometric models can be used to describe regions of the data which correspond to man-made constructions. Probabilistic models can be used to describe vegetation and other features.

Our purpose is to compare geometric and probabilistic models on small regions of interest in lidar data, in order to choose which type of models renders a better description in each region. To do so, we use the Minimum Description Length principle of statistical inference, which states that best descriptions are those which better compress the data. By comparing computer programs that generate the data under different assumptions, we can decide which type of models conveys more useful information about each region of interest.

1 Introduction

High density sources of information, such as lidar, compare with traditional topographic surveys on the vast amount of information available. Automation in the processing of lidar data is not only required for reasons of speed and accuracy, but it also helps to find new ways of understanding the data. The initial assumption is that the best descriptions of objects are the shortest ones, when those descriptions are built taking into account the context information required to reproduce those objects [4, 5] Our purpose is to produce constructive models, in the form of computer programs, that can reproduce the data, are as compact as possible, convey the knowledge we have about the data, and which we can compare using a single measure, their length in bits. This approach is based in the theory of Kolmogorov Complexity, most popular under the perspective of the Minimum Description Length (MDL) principle [8].

A proportion of past work in reconstruction from range images [1, 2] starts from the assumption that the range data represented surfaces with continuity properties. This is not the case with the lidar data we have, representing not only topographic features of terrain but also buildings and vegetation, which are not suitable for representation in terms of curvatures.

In this paper present work towards the selection of appropriate models, with examples on the classification of lidar data using a narrow family of models. Similar work in applications of MDL has spanned over a wide range of applications, for example [3] is a review of MDL from the point of view of machine learning. See [6] for an application to computer vision.

The interest of this approach is in the way it could help to handle increasingly complex models, by helping in the comparison between heterogenous families of models, and in the explicit use of prior knowledge.

We introduce the principle and report an application in which lidar images are segmented in quadtrees [9] and the resulting cells are classified.

2 Model Selection

Constructive models allow us to compare between very heterogenous alternatives. A constructive model leads to a description of the data, which has a length in bits. This length is weakly dependent on the language in which the description is written. If that description is close enough to the Kolmogorov complexity [4] of the data, which is the result of compressing the data as much as possible, then we are obtaining a measure of the complexity of the data.

The data D is represented by a program that corresponds to its structure P, and some error E which we expect to be small. The description of D is P together with S. It is the size of P and S what we use as measure of complexity. If the structure chosen to represent the data is the appropriate one, then the size of the error should be small. But it could be the case that the structure is not very correct but very simple, and still leading to a small description.

These are the families of models we are looking at

geometric representing buildings, dykes and any other feature usually formed by straight lines combined in simple forms. Our aim is to describe geometric features using programs that reconstruct the features, and short codes to represent the error.

probabilistic representing vegetation, areas that are better described by giving the probability distribution, with non-zero standard deviation, that generated them. Our aim is to identify the distributions involved.

The features of interest in our data are characterised in very simple geometric and probabilistic terms, compared to the study of range images in general [1]. But the models that represent such types of features are fundamentally very different: a geometric model that describes well the shape of a building in terms of facets and edges, will fail to describe accurately the shape of vegetation; a probabilistic distribution, more appropriate for vegetation, will not capture most essential characteristics of built environment.

In order to compare such models, they must be defined in a generative form: in our case, we have implemented them as computer programs that can generate the data. The data is described using a computer program to generate it. That means that the model itself is encoded, and its size taken into account. This is an important feature of this method. When the models to compare are fairly similar, the size of the model is irrelevant, and reduces to Maximum Likelihood. This method departs from plain Bayes when the size of the model varies and affects our decision on which model is best.

Probabilistic models consist on data that is draw from a particular distribution. This is equivalent to assume that the data is described in shorter form by a code that associates shortest programs to the data with higher probability. For example, we use a uniform distribution in our experiments, and we implement it by encoding all data using the same amount of bits.

In the case of geometric models, in particular, we expect the model to approximate the data fairly well. In our experiments we consider flat surfaces. To encode the errors for such a model we use the $\log^*$ code [7], which is just one case of a code that associates shorter programs to shorter numbers while filling the tree of codewords.

3 Experiments

The pilot site of our project is a 25 km stretch of the river Váh in Slovakia, chosen for the purpose of flood simulation and impact assessment. Our experiments are centered in a patch of terrain around the canal that include in a cross section. The features under consideration are industrial buildings and vegetation.

These experiments were carried in two steps. First the image was segmented into a quadtree using a homogeneity criterion, then the resulting quadtree cells, of different sizes, were classified according to a description length criterion.

4 Segmentation

The first step in the labeling is the segmentation of the images. The segmentation structure are quadtrees, which are recursive division of a cell c into four equal cells c_1, c_2, c_3, c_4, whenever a homogeneity test is negative over the cell. The first cell is the complete image. Quadtrees were chosen expecting to obtain somehow a transition between probabilistic models (quadtrees with small cells) to geometric models, which in our case correspond to flat areas of the terrain and large tiles.

Two main types of homogeneity test were tried in the segmentation (see Figure 1 and below); the simplest one is the variance test, $v(c)$, in which the cell is divided whenever the variance is above a threshold. The second type of test consists on describing the area as if it was flat with small perturbations, $\log^*(c)$, and dividing the cell if the amount of bits per pixel is above a threshold. The third type of test consists on describing the cell c as if it a flat area with small perturbations, $\log^*(c)$, encoding the mean value and the small perturbations using a special code, then doing the same considering now the four sub-cells independently $\log^*(c_1) \ldots, \log^*(c_4)$. Both descriptions of the same data are compared and those shortest one is chosen: if dividing the cells leads to a shortest description, then the recursion goes on.

Only the first two tests lead to significant results, both $v(c)$ and $\log^*(c)$ produce clusters of small cells in the vegetation areas and lines of small cells in the edges. The $\log^*$ description did not lead to a significative improvement over the variance.

This leads to an illustration of the fact that a variety of common model selection methods are in fact computable approximations of the Kolmogorv Complexity. The variance function $v(c)$ corresponds to an encoding in the similar way that our

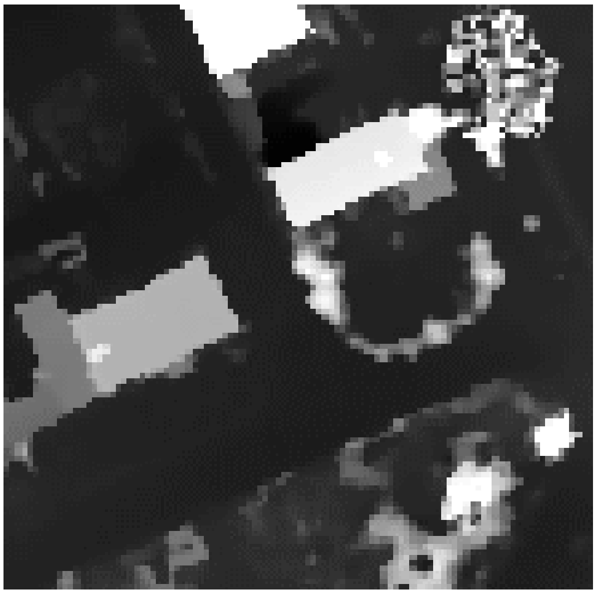

Figure 1: Lidar tile 100 meter side

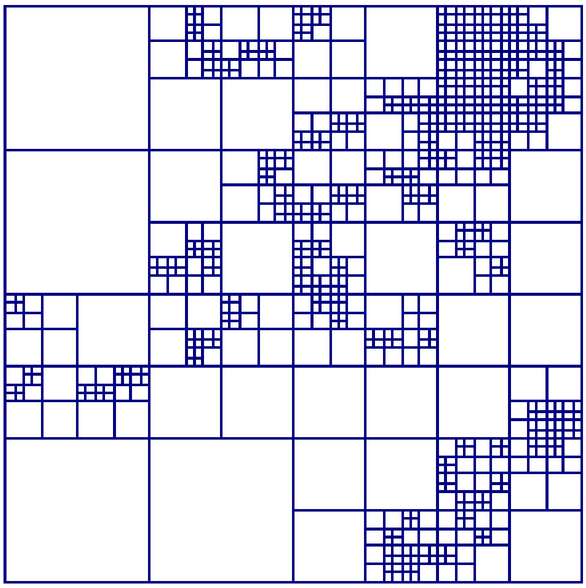

Figure 2: Quadtree segmentation of Figure 1 based on variance, cells with variance below 20000 are not split

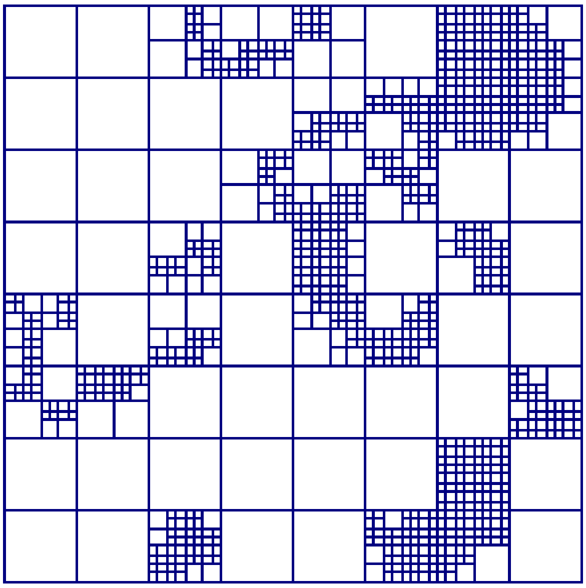

Figure 3: Quadtree segmentation of Figure 1 based on description size using $\log^*$, cells with 14 bits per pixel or less are not split

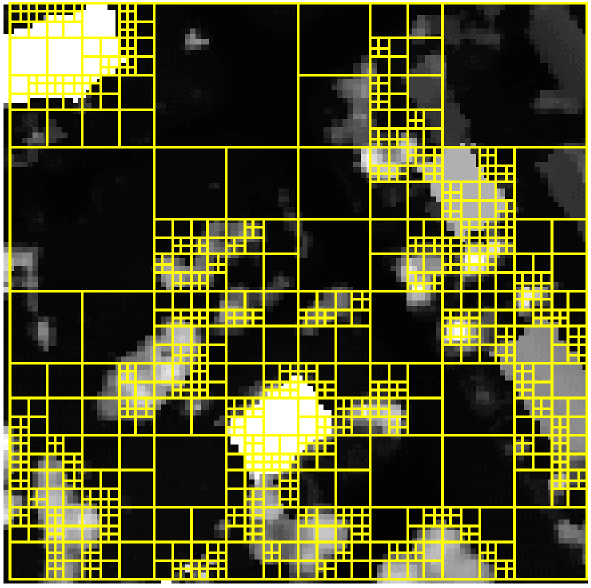

Figure 4: A lidar tile (100 meter side) after variance-based segmentation into a quadtree

$\log^*(c)$ encoding:

$$v(c) = \frac{\sum_i (s_i - \mu)^2}{n}$$

(where μ is the mean and $c = \{s_i, \ldots, s_n\}$ the data available), is the code length when each datum is represented using $(s_i - \mu)^2$ bits. The actual $\log^*$ function we have used is:

$$\log^*(c) = \frac{\sum_i \log^*(s_i - \mu)^2}{n}$$

5 Classification

For the classification, a wider family of models were compared: each of them was used to describe the cells, and the model that produced the shortest description of the cell was used to label it. Two families of models were considered: the data was produced by a uniform distribution, which means that it is evenly distributed within its range of values, or the data was flat with small perturbations, subject to short description by $\log^*$.

Two models were dominant, and are used to label Figure 5. The cells that correspond to the vegetation and building corners were better described by the $\log^*$ code. The rest of the cells, including flat areas, were better described by the assumption of a uniform distribution. This fact contradicts the fact that $\log^*$ should encode better flat areas, and it is just a direct effect of the size of the cell.

6 Conclusion

We are applying the concept of length of description to compare very heterogenous models when interpreting lidar data. This method would also help in handling models of varying complexity.

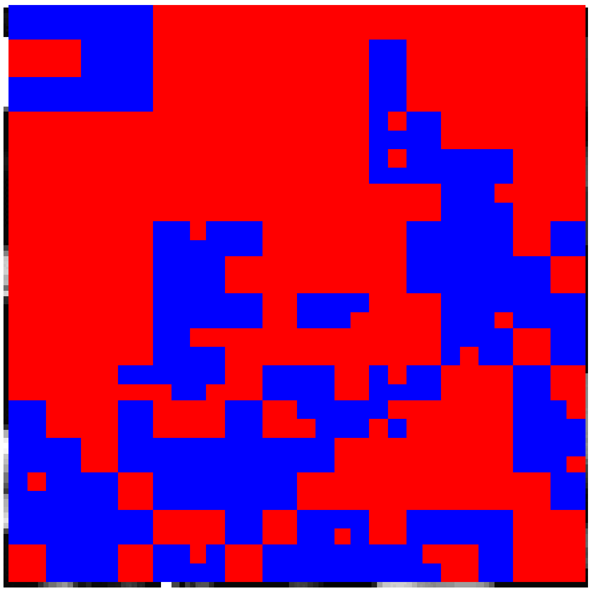

Figure 5: A classification of the pixels from the segmentation in Figure 4, according to the shortest description. Large (lighter) blocks are better represented by a uniform distribution, small (darker) blocks's description is shorter when using a $\log^*$ code

Applications of this technique are in the development of extensible, flexible methods for off-line feature extraction, as a result of the ability of this algorithm to compare between very heterogenous models.

We also expect this sort of metrics to be robust against outliers, a requirement for automatic feature extraction.

We are also looking for alternatives for the segmentation, such as region growing, that produce structures with a clearer geometric meaning. This would produce complex models that better represent the reality, while keeping a grasp on a wider range of models by means of the code length metric.

REFERENCES

[1] Farshid Arman and J. K. Aggarwal. Model-based object recognition in dense-range images — a review. *ACM Computing Surveys*, 25(1), March 1993.

[2] Brian Curless. *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, 1997.

[3] Peter D. Grnwald. *The Minimum Description Length Principle and Reasoning Under Uncertainty*. PhD thesis, CWI, 1998.

[4] Ming Li and Paul Vitányi. Inductive reasoning and Kolmogorov complexity. *Journal of Computer and System Sciences*, 44:343–384, 1992.

[5] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2nd edition, 1997.

[6] S. J. Maybank and R. Fraile. Minimum description length method for facet matching. In Jun Shen, P. S. P. Wang, and Tianxu Zhang, editors, *Multispectral Image Processing and Pattern Recognition*, number 44 in Machine Perception and Artificial Intelligence, pages 61–70. World Scientific Publishing, 2001. Also published as special issue of the International Journal of Pattern Recognition and Artificial Intelligence.

[7] Jonathan J. Oliver and David Hand. Introduction to minimum encoding inference. Technical Report 4-94, Department of Statistics, Open University Walton Hall, 1994.

[8] Jorma Rissanen. *Encyclopedia of Statistical Sciences*, volume 5, chapter Minimum-Description-Length Principle, pages 523–527. Wiley New York, 1983.

[9] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. PWS Publishing, second edition, 1999.