

Learning Spatial Grammars for Drawn Documents Using Genetic Algorithms

Simon J. Hickinbotham* and Anthony G. Cohn†,
University of Leeds, Leeds LS2 9JT, UK

E-mail: a.g.cohn@leeds.ac.uk

Abstract

The problem of object recognition may be cast into a spatial grammar framework. This system comprises three novel elements: a spatial organisation of line features, an efficient two dimensional parsing engine, and a genetic algorithm learning routine that induces spatial grammars. Labelling the spatial organisation of feature pairs allows the terminal symbols of the spatial grammar to be defined, and constrains the search space of the feature parser. A genetic algorithm approach is then used to induce appropriate grammars using a supervised learning routine. Early results show that similar foreground and background features can be discriminated using this approach.

1 Introduction

Object recognition is one of the central challenges in document analysis and related fields. This is a difficult problem, since there is a potentially overwhelming number of low level feature candidates, and no obvious way in which a feature hierarchy can be assembled from them [2, 5]. To overcome these difficulties, we are developing a framework for supervised learning of line segment pairs. We divide the recognition task into two stages. The first stage uses a novel model of perceptual organisation (PO) for 2D line relationships. Features from this stage are fed into a learning stage, which builds a hierarchy of features to arrive at a labelling of the object.

In our representation, connectedness of features is established via the perceptually organized relationships, and the hierarchy of the structure is induced via the learning phase. The advantages of this approach are twofold. Firstly, there is no need for a manual interpretation of the data, which is difficult given the gestalt nature of the patterns to the human

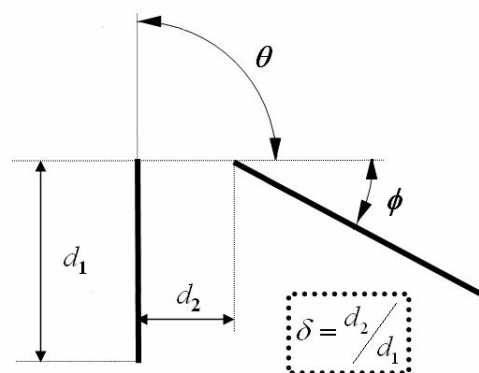


Figure 1. Perceptual organisation of lines.

eye, and secondly, the hierarchy is more likely to be robust since it is derived directly from the data.

Genetic Algorithms (GAs) which induce grammars are sparsely reported in the literature. Works most relevant to the task in hand include Wyard [8] and Kammeyer and Belew [6], who both used a variable-length chromosome, which we have also adopted. The methodology for handling this family of GAs is not mature, yet there are interesting parallels between spatial grammars and biological genomes that were noted by Burke et al [1]. Biological genomes vary in length during evolution, they contain genes that are independent of position, contain non-coding regions, and may contain duplicative or competing genes.

In previous work by the authors [4], diagrams were recognised by parsing a spatial grammar that was specified by hand. One of the key difficulties with this approach is that hand-crafted grammars become increasingly difficult to design as the complexity of the objects to be recognised increases.

2 Methodology

Perceptual Organisation is a means of controlling computational expenditure in the recognition process. Lowe [7], introduced quantitative measures for PO, where emphasis

*Now at YCCSA, University of York, E-mail: s.jh@cs.york.ac.uk

†This work was carried out as part of the VISTA project, sponsored by the DTI, UK. The authors thank EDF energy for the provision of the data sets.

was placed on collinearity, parallelism and endpoint proximity. We have devised a new way to organize line feature hierarchies for PO in 2D figures, where a richer representation is likely to be more stable.

2.1 Hierarchical perceptual grouping of line segments

We detect lines in scans of electricity network drawings following [4]. The first ingredient of our perceptually organised feature set is the spatial relationship operator. There are many ways of defining such operators. Examples include Fidler et al. [2], who used a similar structure to organise their search through perceptual space, and Jin and Geman [5], who used a hand-coded grammar to drive a number-plate recognition scheme. A novel feature of our approach is that we do not require connectedness. Instead, we define a search radius, based on the size of the feature, within which features can be placed via a spatial relationship operator. Spatial relations between entities are binned along three axes, thus forming a three dimensional grid of cells into which relationships can be binned. The three measurements are: θ , the angle formed from the centreline of the shortest feature; δ , the distance between the features as a proportion of the length of the shortest feature; ϕ , the angle between the two centrelines. These features are illustrated in figure 1. Processing commences by identifying all relative line pairings with particular values of θ , δ , and ϕ . Only these relationships are passed into the grammar for processing.

We use a minimum bounding rectangle (MBR) to specify the spatial extent of any feature in our hierarchy. MBRs are simple to compute and can encapsulate any feature from a simple line to a complete figure. However, we need a means of robustly defining the orientation of the MBR. This is a simple labelling process. The narrower sides of the rectangle form the “head” and the “tail” of the feature. The “head end” is then the side with the highest number line endings nearest to it. The top left panel in figure 2 illustrates this. Features are represented by a grey rectangles. A circle on the centre of one side of each rectangle indicates the head end of each feature. Individual line pairings are thus iteratively grouped to form a feature hierarchy. Recognition follows from attention via the parsing process. We have devised a specification for a grammar that captures these relationships, which is described in the following section. This perceptual grouping model greatly reduces the search space for the processing hierarchy.

2.2 2D spatial grammars

We express our hierarchical organisation via a Stochastic Context Free Grammar in Chomsky Normal Form (CNF).

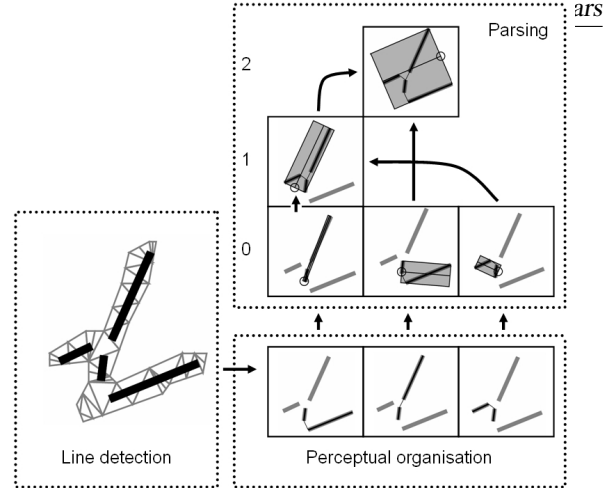


Figure 2. Parsing a perceptually organised feature set.

Spatial relationships between elements are specified via the addition of a spatial relationship operator to the rule definition. Our 2D grammar expands the canonical CNF to include a relationship operator. Terminal rules take the following form:

$$A \rightarrow \underset{\theta, \delta, \phi}{\alpha} \quad (1)$$

where α is effectively a pointer to a line pairing described in the previous section. Nonterminal rules operate in a similar manner:

$$A \rightarrow \underset{\theta, \delta, \phi}{B \bowtie C} \quad (2)$$

The relationship operator \bowtie specifies the spatial relationship that exists between the two elements either side of the operator. Each particular operator specifies θ , δ and ϕ . We assign a probability to each production rule following Kammerer and Belew [6].

2.3 Parsing 2D grammars

We adopt a bottom-up strategy to learning the grammars of the line segments. We employ a modification to the CYK algorithm [3] to parse incomplete grammars that are produced by the GA. Our modification tests for the spatial relationships between all instances of symbols at a particular parsing step. Although this has the potential to make the computational burden too heavy, we only test when it is possible that the probability for the production exceeds the

2.4 METHODOLOGY
 current probability for the same symbol. This is relatively cheap to compute, and so allows parsing within acceptable time constraints.

An example of a spatial parse is illustrated in figure 2. Perceptual groupings of detected lines, specified as pertinent by the grammar, are shown in the bottom right of the diagram. These are passed into the parsing algorithm, where they form the lowest level of the feature hierarchy. Higher levels of the hierarchy are then generated by a process of combination. Note that symbols do not have to be consecutive in the layers of the hierarchy – it is impossible to arrange representations of 2D adjacency on a 1D line. This is the ultimate reason for the cost of 2D parsers – one has to search through relevant rows of the hierarchy to find candidate symbols for the r.h.s. of a rule.

With our grammar and parser to hand, the remaining ingredient of our learning strategy is the induction method. This is described in the following section.

2.4 Encoding spatial rules for learning via Genetic algorithms

We have chosen to use an ASCII-coded string for our genetic sequence so that rules can be manually written and inserted into strings for testing purposes. There is no appreciable difference between coding a synthetic genome in binary vs. ASCII characters [8]. Our encoding scheme is identical to that described in [6], with the addition of a straightforward symbolic code for the spatial relationships which distinguish our spatial grammar from one-dimensional grammars.

We chose a fixed population size of 160 individuals by empirical trial. Individual genomes have a lifetime of one generation. We use a roulette selection approach to choose parents for subsequent generations, whereby the chance of an individual being selected to become a parent is proportional to its fitness. One individual is generated per mating.

For crossover, we pick two random places in the parents and swap the strings at these points. The mutation rate is 0.003 for each element of the DNA string. Mutation can be insertion, substitution or deletion. Each has an equal chance of occurring.

We use a supervised learning scheme for two reasons. Firstly, it is easier to assess the performance of the approach using supervised learning, since we can directly specify the features that the algorithm is meant to learn (compared with unsupervised learning). Secondly, supervised learning allows us to define a start symbol via the fitness function.

The populations were initialised such that each individual encoded a grammar consisting pair of terminal symbols which randomly referred to a spatial relationship, along with a nonterminal production rule which combined the symbols.

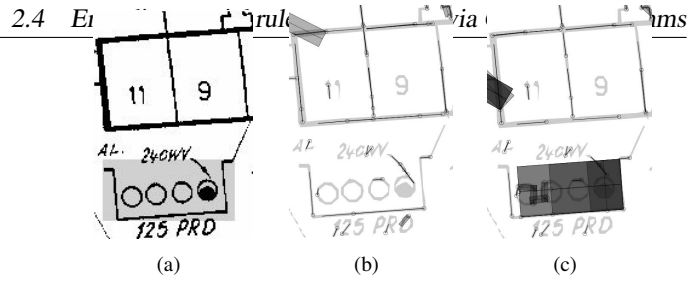


Figure 3. (a): Part of scan of a utility map. The target feature is shaded in grey. (b): Example parse from the first generation. (c): Productions for a grammar with fitness 0.83

The rule probabilities are all 1.0 since each symbol in the grammar is unique. Our fitness function rewards the production of grammar rules and the amount of the image the productions cover. We use the area of the foreground T_f and background T_b regions in the image as normalising elements in our fitness calculations. We use the minimum bounding rectangle of each production to calculate how much of T_f and T_b are accounted for (“covered”) by the grammar. P_f is the area of the foreground covered by the parse, and P_b is the area of background covered by the parse. Early experiments indicated that early generations of a run commonly evolved DNA strings with no production rules in them. To counter this phenomenon, we partitioned a small part of the fitness to reward DNA with parsable rules. This is a constant ρ , which we set to 0.05, so if a DNA string contains parsable rules, ρ is added to the fitness score. Our fitness function $F()$ thus takes the form:

$$F() = \rho + ((1 - \rho) * \frac{P_f}{T_f} * (1 - \frac{P_b}{T_b})) \quad (3)$$

Specifying start symbols during the induction process is a difficult challenge, since there is no rule regarding adjacency as there is with 1D grammars – a particular feature can have extent in any direction on the page, and can encompass features which are not necessarily part of the target. Supervised learning allows us to find start symbols via the fitness function described above. We iteratively assign every symbol in the grammar to be the start symbol and calculate fitness for each. The symbol which produces the fittest parse then becomes the start symbol S for the grammar encoded by the DNA string of each individual. Finally, to combat the phenomenon of “bloat”, we set a maximum DNA string length of 500 bases. Offspring longer than this were truncated. This was a necessary step, since very long DNA strings encoded grammars with unacceptably long parse times.

REFERENCES

Experimental Evaluation

An example subfigure is shown in figure 3(a). It is clear that the low level line features are present in both the foreground and background regions of the image. Figure 3(b) shows the initial perceptual grouping specified by the first-generation grammar described above. The nonterminal rule does not fire because the two features are not perceptually organised. Figure 3(c) shows the nonterminal productions from a grammar with fitness 0.83 from generation 160 of a typical evolutionary trial. A large proportion of the foreground is covered by the grammar, which explains the high fitness value.

It is interesting to note some further statistics regarding this induced grammar. Firstly, 10 rules are used to create productions for the test figure, but there are 12 rules (including duplicates) in the grammar. All of the rules that are used in the parse have the same left-hand side symbol. Only one of these is a nonterminal rule, but this rule is recursive, specifying $E \rightarrow E \bowtie_{3,0,2} E$. The spatial relationship for this rule describes a right-angled corner, with the shorter feature approximately half the length of the longer one.

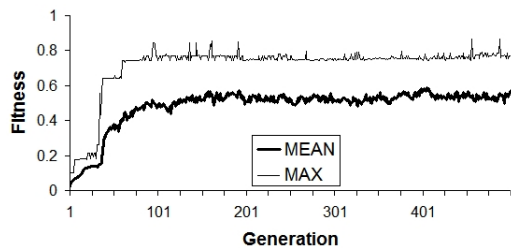


Figure 4. Change in fitness.

Figure 4 shows the evolution of 500 generations using the fitness measure in equation 3. Maximum fitness is reached by a single individual in the 160th generation, although the average fitness for the population is still low. This average fitness increases until the 150th generation, where it stabilises at around 0.5. It is clear that the GA is capable of delivering improvements in fitness, but the fitness measure we use at present delivers limited spatial grammars, possibly since we do not reward grammars that have desirable qualities such as depth of parsing, or diversity of symbols.

We tested the fitness of the learned grammar on similar data to the training data. An example of parses on two subdiagrams is shown in figure 5. The average fitness measure on 10 test images was 0.44. Relatively little of the image background was parsed, but there remained a large proportion of foreground that went unlabelled. This contrasts with our work on hand-crafted grammars, which often completely missed some target subdiagrams.

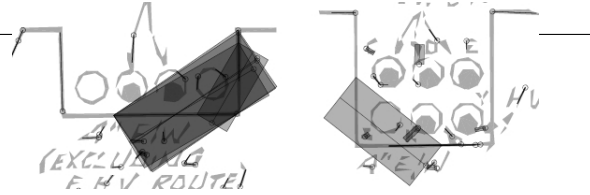


Figure 5. Parsing test data

4 Conclusion

The approach described above builds appropriate partial parses of image data and delivers perceptually organised feature hierarchies. Now that a complete system has been developed, efforts could be focussed on improving the quality of the learned grammars. The main focus would be to experiment with a range of fitness functions, and other encodings with a view to shortening the learning process and making the active parts of the grammar more dominant in the genome. In addition, exploration of a more robust perceptual grouping model, where the rather crude binning strategy is replaced with a cost function between the spatial relations specified by the grammar and the candidate feature pairings merits exploration. Evaluation of our approach with a range of low-level feature detection strategies such as skeletonisation, filtering and Hough transforms is another obvious next step.

References

- [1] D. Burke, K. D. Jong, J. Grefenstette, and C. Ramsey. Putting more genetics into genetic algorithms. *Evolutionary Computation*, 6(4):387–410, Winter 1998.
- [2] S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *Proc. CVPR*, pages 182–189. IEEE Computer Society, 2006.
- [3] R. C. Gonzales and M. G. Thomason. *Syntactic Pattern Recognition: An Introduction*. Addison-Wesley, Reading, MA, 1978.
- [4] S. J. Hickinbotham and A. G. Cohn. Knowledge-based recognition of utility map sub-diagrams. In *Proc. ICDAR*, pages 213–218. IEEE Computer Society, 2007.
- [5] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *Proc. CVPR*, pages 2145–2152. IEEE Computer Society, 2006.
- [6] T. Kammeyer and R. K. Belew. Stochastic context-free grammar induction with a genetic algorithm using local search. In R. K. Belew and M. Vose, editors, *Foundations of Genetic Algorithms IV*. Morgan Kaufmann, 3–5 1996.
- [7] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.
- [8] P. Wyard. Context free grammar induction using genetic algorithms. In *Grammatical Inference: Theory, Applications and Alternatives, IEE Colloquium on*, pages 11/1–11/5, 1993.