Perceptual Grouping Using Gestalt Laws: Coding as Production System, Running with Data-Driven Control

E. Michaelsen

FGAN-FOM
Gutleuthausstr.1, 76275 Ettlingen, Germany
mich@fom.fgan.fom.de, www.fom.fgan.de

Abstract The laws of gestalt-perception play an important role in human vision. Psychological studies identified *similarity, good continuation, proximity* and *symmetry* as important inter-object relations that distinguish perceptive gestalts from arbitrary sets of clutter objects. Research in mimicking this behaviour for machine vision has a long history. This contribution focuses on coding these principles in production systems. Such systems capture declarative knowledge. The procedural details are defined as control strategy for the interpreter. Here this unit implements an assessment driven approach. Often an exact solution is not feasible while approximately correct interpretations of the data with the production system are sufficient. One such way to interpret given data and a given production system in any-time manner is the accumulative assessment driven control. There even has been work on speeding this search by associative memory structures and special parallel hardware. Recently the work focuses on coding such search in a machine-independent manner (e.g. in MATLAB). Here an example from the automatic extraction of man-made structure from high resolution SAR-image data is given.

1 Introduction

Psychological studies have been indicating for a long time now that human perception tends to group individual objects fulfilling certain relations into entities of higher order [14]. Fig. 1 illustrates some of these inter-object relations namely *similarity*, *good continuation*, *proximity* and *symmetry*. Obviously an object of higher more abstract hierarchical level is constructed. The information content is compressed in its description and its members are clearly distinguished from other objects which are perceived as background. Research in mimicking this behaviour for machine vision also has a quite remarkable history [5, 7]. Particularly, the application of such technique to image retrieval tasks has proven successful [4] (using also the very strong constraint of parallelism as grouping relation for man-made structure). There is joint work from psychologists, artificial-life researchers, neurophysiologists, Darwinists and computer vision experts to derive these principles from co-occurrence statistics of natural images and the principles of evolution of species [2]. Lately, tensor voting has been proposed to constitute the interaction between local entities close to the filter- or feature extraction level [8]. On the other hand perceptual grouping may also be performed on higher symbolic reasoning level. In [3] Guo et al. give striking examples for the utility of grouping higher order objects they named textons. These have many more attributes than just a location and orientation. They use Gibbs-fields and Marcov chains to describe the inter-texton relations, to analyse given images and to generate new images. It also has been demonstrated that perceptual groups may well be subject to meta-grouping [12] and also that the objects to be grouped may well be on such high level like 3D-houses (simple gabled parametric model) along a 3D-street [10]. Perceptual grouping may be helpful on any level of a complex structural recognition approach.

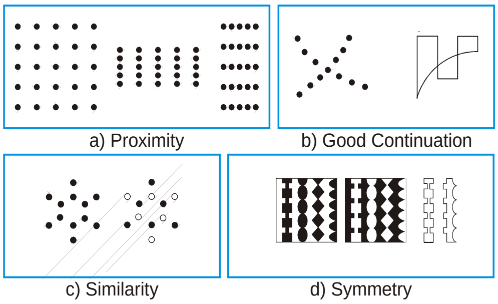

Fig.1 Examples of Gestalt relations

Formally, a relation φ between an object T and its parts S_i as well as the constraint π imposed on the parts can be written as production $p=((S_1,\ldots, S_n), T, \pi, \varphi)$ of a coordinate grammar [11, 9]. Actually, π is a predicate defined on the coordinates of the generated side $(S_1,\ldots, S_n)$ while φ is a function on the same space calculating the coordinates of the single object of the reduced side T. A structural recognition approach often requires more than one production. In such situation the productions may be simply listed as production system $\mathscr{P}=\{S, A, P\}$ where S is a finite set of object symbols, A is the attribute domain (containing things like image coordinates, orientations, number of members etc.) and P is a finite set of productions. It is preferred to display their interaction as production net [9, 13]. Fig 2 shows the example system used for this contribution. The syntactic structure of this formalization becomes apparent looking at an example production like

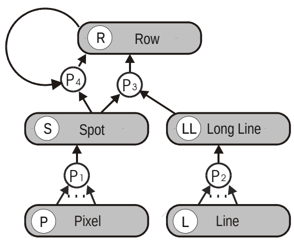

Fig2. Production net

p_4 = ((**Row**, **Spot**), **Row**, *good_continuation*, *append*),

which allows recursive generation or parsing (because of the cyclic net structure). Other productions like the example

p_2 = ({**Line** … **Line**}, **Long_Line**, *collinear & overlapping*, *regression*)

define the constraint on a sub-set of arbitrary size.

The production-system or production-net rationale postpones major problems of Gestalt-driven machine perception to the control unit of the production interpreter. Particularly, the two production forms given above may lead to excessive requirements for storage capacity and processing time. The number of objects **Row** reducible by repetitive use of production p_4 may well grow exponentially with the number of objects **Spot** in an image. This may be eased using a contextual production like p_3 = ((**Spot**, **Long_Line**), **Row**, *proximity*, *copy*), that restricts the search to potential building outlines and also limits the orientation possibilities – objects **Row** with only one member inherit the position from the preceding **Spot** and the orientation from the preceding **Long_Line**.

A production with a set of arbitrary size on the generation side – like p_1 – will pose a power-set search problem if used in the reducing direction. Given a large generated set of objects **Line** from one object **Long_Line** many of the sub-sets will also be valid for reduction with p_1. In other words: There may be many possibilities to split a given contour or to kick out “outliers”.

Therefore coding such productions in a system like PROLOG and running them on non-toy data is doomed to produce large amounts of irrelevant information or fail completely. However, there are alternatives. While a complete list of solutions may not be feasible, a list of some good or important interpretations of the data with the production system may be sufficient for the task. In such case it is recommended to explicitly declare what is meant by “good” or “important”.

The assessment of a **Pixel** object results from the response of the spot-operator used as well as the assessment of a **Line** object results from the edge ore line filter chosen by the operator. For the **Spot** object the total mass of all its predecessors was used (its centre of gravity also specifies its position). A **Long_Line** object is assessed by its length and the residual error of its parts. The assessment of an object **Row** depends on the number of members and the regularity of their positioning.

2 The Accumulating Interpretation Cycle

Given a production system $\mathcal{P}=\{S, A, P\}$ a working element is defined as quadruple $e=(s, i, as, pm)$ where s is a symbol from S, i is an object instance index, as is an assessment and pm is a production module index. Assessments may be taken from a symbolic discrete ordered domain like *{very good, good, average, bad, very bad}* or from a continuous ordered interval like $[0, 1]$. A production module is always triggered by a particular object instance. It contains code that queries the database for partner instances which fulfil the constraint relation π of the production given the triggering object instance. Usually search regions are constructed (e.g. a long stripe shaped region with the triggering **Line** instance in the centre for p_2). If the query results in a non-empty set the module will create new instances according to the functional part φ of the production. Some productions need more than one module (e.g. p_4 may be triggered by a **Row** instance or by a **Spot** instance requiring different queries). The set of module indices is expanded by *nil*. Always when a new instance is created – either by an external feature-extraction process or by one of the production modules – also a corresponding new working element is added using this module index *nil* (meaning that there is no module assigned yet). The set of working elements is called the queue. It is sorted occasionally (e.g. every 100 interpretation cycles) with respect to the assessments. The central control unit (AI-people call it dispatcher) always picks working elements from the good end of the queue. If the module index of an element is *nil* it will be replaced by new working elements with appropriate module indices (recall that each connection from a symbol to a production in the production-net corresponds to a production module, i.e. a possibility to be tested). If there is a non-*nil* module index attached this module will be triggered by the corresponding object instances. Modules may be run in parallel on different processors. The dispatcher can start picking elements from the queue the moment the first primitive instances are inserted. It terminates inevitably when the queue happens to run empty. But usually it will be terminated before, either by external processes or the user, or by limiting the number of cycles or time.

3 From BPI to MATLAB

The BPI-system designed for the accumulative interpretation of images has proven successful on many applications where perceptual grouping capabilities were desirable for almost 20 years [6,9,10,12,13]. BPI (Blackboard-based Production-systems for Image-analysis) was a compiler language providing special means useful for coding production modules. Special care was taken for set operations and particularly for associative access to the database organized in blackboard architecture. Data were taken from as different domains as oblique and aerial images, thermal IR-videos, SAR-data and laser range finder data. Many modules could be used with only slight modifications for these different data types. The BPI queue mechanism included not only sorting according to a finite symbolic assessment attribute but also the definition of focus of attention areas. Special associative access hardware memory was constructed and widely used with BPI. Massive parallel single instruction multiple data machinery was tested. Because of these issues the attribute values were restricted to discrete finite domains. This has proven a serious obstacle. Today, however, the most important disadvantage of this system is its machine dependency (the assembler routines in its kernel are designed for VAX/VMS).

Production systems like outlined in Section 1 and the modules and interpretation cycle explained in Section 2 can be coded in any computer language. MATLAB has proven a good choice because the database and the queue can be stored in arrays which will be automatically enlarged on demand. Also there are convenient graphical output means, access to all relevant information for debugging and large ready coded tool boxes for mathematical sub-tasks. Primarily, the use of MATLAB may foster the acceptability of the approach for other researchers or applicators.

4 Extracting Scatterer Rows from SAR-Images

As an example application the extraction of presumably man-made salient rows of strong scatterers in aerial X-band SAR images is chosen. The development of synthetic aperture radar systems has lead to very small sample sizes on the scene (in the order of a decimetre) [1]. In most of these small scene portions there is only low back-scattering, so that the image appears dark compared to traditional SAR images with several meter pixel sizes where at most some scatterers where present in almost every pixel. Some pixels, however, give a response several orders of magnitude higher than the dark background leaking also into adjacent pixels. These contain special structure like rectangular corner reflectors formed by window-sills or metallic water gutters. Fig. 3 shows a section of such an image taken from urban terrain (the campus of the University of Karlsruhe). It also displays an automatic interpretation of it using the production system outlined in Section 2.

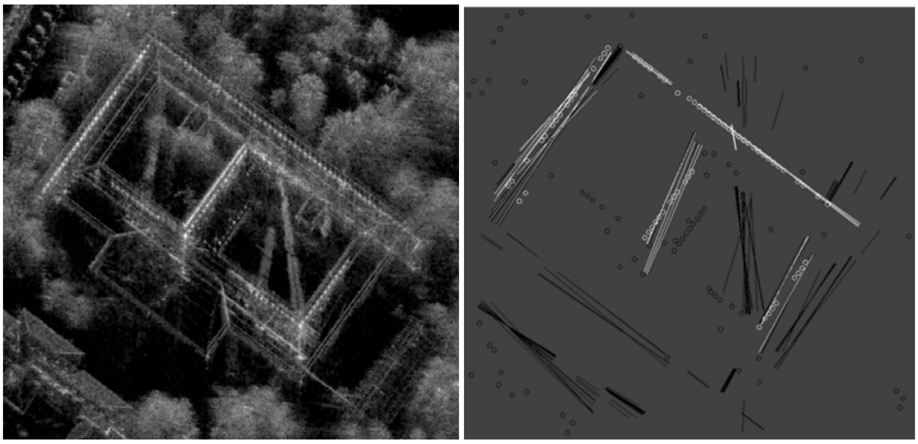

Fig. 3: Left: Section from a high-resolution X-band SAR image (enhanced by noise reduction filter); right: Result of grouping after 10,000 interpreter cycles (white); objects rejected as clutter (black)

The system obviously concentrates on the same salient rows that also absorb most attention of human subjects. In doing so it discriminates the large building complex well from vegetation and clutter.

5 Conlusion

Gestalt mechanisms in human perception will draw attention to the salient rows of scatterers in good continuation even if the human subject is not familiar with SAR imagery nor has any idea of what is the image content. Anybody will perceive man-made structure like buildings instantaneously in such images although the details of SAR-image formation are not easy to understand and lead to awkward phenomena like lay-over. The same Gestalt mechanisms that were constructed by nature to fit the perception tasks of animal and man in its natural environment also work with other pictorial data coming from completely different sources. This indicates evidence that these principles reflect very general and important laws. So for any desired automatic recognition on pictorial data Gestalt mechanisms should be considered, at least next to the domain specific features.

Coding these principles in declarative productions eases the designer from the labour of specifying in detail the sequence of actions of a digital machine or the connectivity of a neural network that causes such behaviour.

Acknowledgement

SAR data were recorded by the PAMIR experimental system of FGAN - Prof. Dr. Ender and Dr. Brenner (both FGAN-FHR).

References

1. J. H. G. Ender, A. R. Brenner, *PAMIR - a wideband phased array SAR/MTI system*, IEE Proceedings - Radar, Sonar, Navigation, vol. 150, no. 3, 2003, pp. 165-172.
2. W. S. Geisler, J. S. Perry, B. J. Super, D. P. Gallogly, *Edge Co-occurrence in Natural Images Predicts Contour Grouping Performance*, Vision Research, Vol. 41, 2001, pp. 711-724.
3. C.-E. Guo, S.C. Zhu, Y. N. Wu, *Modelling Visual Patterns by Integrating Descriptive and Generative Methods*, IJCV, Vol. 53, No. 1, 2003, pp. 5-29.
4. Q. Iqbal, J. K. Aggarwal, *Applying Perceptual Grouping to Content-Based Image Retrieval: Building Images,* Proc. IEEE Int. Conf.. on Comp. Vis. and Pat. Rec. 1999, Fort Collins, Col., 1999, pp. 42-48.
5. D. G. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer, Boston, 1985.
6. K. Lütjen, *BPI: Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung*, in: G. Hartmann (ed.), *Mustererkennung 1986*, DAGM 86, Springer, Berlin, 1986.
7. D. Marr, *Vision*, Freeman, San Francisco, 1982.
8. A. Massad, B. Babós, B. Mertsching, *Application of the Tensor Voting Technique for Perceptual Grouping to Grey-Level Images*, in: L. Van Gool (ed.), *Pattern Recognition*, DAGM 2002, Springer, LNCS 2449, Berlin, 2002, pp. 306 -313.
9. E. Michaelsen, *3D Coordinate Grammars*, in: B. Girod, H. Niemann, H.-P. Seidel (eds.), *3D Image Analysis and Synthesis ,96*, Infix, Sankt Augustin, 1996, pp. 81-85.
10. E. Michaelsen, U. Stilla, *Problems in Geometric Modelling and Perceptual Grouping of Man-Made Objects in Aerial Images*, International Archives of Photogrammetry and Remote Sensing, Vol. 33, Part B3, pp. 577-583.
11. D. L. Milgram, A. Rosenfeld, *A Note on Grammars with Coordinates*, in: F. Nake, A. Rosenfeld (eds.), *Graphic Languages*, North Holland, Amsterdam, 1972, pp. 187-194.
12. U. Stilla, E. Michaelsen, U. Soergel, *Perceptual Grouping of Regular Structures for Automatic Detection of Man-made Objects – Examples from IR and SAR*, IGARSS, IEEE, 2003.
13. U. Stilla, E. Michaelsen, K. Lütjen, *Automatic Extraction of Buildings from Aerial Images*, in: F. Leberl, R. Kalliany, M. Gruber (eds.), *Mapping Buildings, Roads and other Man-made Structures from Images*, IAPR-TC7, Wien, Oldenburg, 1996, pp. 229-244.
14. M. Wertheimer, *Untersuchungen zur Lehre von der Gestalt II*, Psychol. Forsch., Vol. 4, 1923, translated as *Principles of Perceptual Organization* in: D. Beardslee, M. Wertheimer (eds.), Princeton, N. J., 1958, pp 115-135.