# Interaction of Control and Knowledge in a Structural Recognition System

Eckart Michaelsen, Michael Arens, and Leo Doktorski

Research Institute for Optronics and Pattern Recognition FGAN-FOM
Gutleuthausstr. 1, 76275 Ettlingen, Germany
{michaelsen|arens|doktorski}@fom.fgan.de

**Abstract.** In this contribution knowledge-based image understanding is treated. The knowledge is coded declaratively in a production system. Applying this knowledge to a large set of primitives may lead to high computational efforts. A particular accumulating parsing scheme trades soundness for feasibility. Per default this utilizes a bottom-up control based on the quality assessment of the object instances. The point of this work is in the description of top-down control rationales to accelerate the search dramatically. Top-down strategies are distinguished in two types: (i) Global control and (ii) localized focus of attention and inhibition methods. These are discussed and empirically compared using a particular landmark recognition system and representative aerial image data from GOOGLE-earth.

## 1 Introduction

Structural recognition of patterns or objects is an option in cases where the structure of the target patterns is the property which distinguishes them from clutter or background best, i. e., for patterns and objects where no obvious or simple numerical features are at hand which promise satisfying recognition performance of machine learning or statistical methods. Structural recognition of patterns or objects should also be considered if the structure of the targets is the desired output, i. e., if recognition is meant as automatic pattern understanding. Last but not least structural recognition of patterns or objects may be beneficial if the number of training examples is low – or none are given at all – while there is knowledge accessible on the constructive elements and rules that define the targets, i. e., if the learning data consist of handbooks, CAD-models, thesauri, or ontologies respectively. Production systems give an approved formalism for structural recognition of patterns or objects.

The idea to employ high-level knowledge for the guidance or control of low-level image analysis processes has ever been conceived since the days of early work of Kanade [5] and, e. g., Tenenbaum and Barrow [13]. Performed on single images – often termed image understanding, compare [12] – the work of Neumann [4] and colleagues can be seen as a prominent and current work in the field of top-down control of low-level image analysis. Work on exploitation of high-level knowledge in form of expectations or anticipations on time-varying imagery –

|  | left-hand | constraint | right-hand |
|---|---|---|---|
| $p_{prolong}$ | longline | collinear and overlapping | line ... line |
| $p_{stripe1}$ | road | parallel and $6.75m < d < 11.25m$ | longline longline |
| $p_{stripe2}$ | halfhighway | parallel and $16m < d < 19m$ | longline longline |
| $p_{dstripe}$ | highway | parallel and $6.0m < d < 8.4m$ | halfhighway halfhighway |
| $p_{cross}$ | bridge | crossing the T | road highway |

**Table 1.** Example production system for recognizing bridges over highways.

i. e. videos – have been reported by the groups of Dickmanns [3] and Nagel [1], to name only two. Declarative production rules as means to code knowledge for aerial image understanding has been introduced, e. g., in the SIGMA system [8]. This includes a discussion on intelligent control of the search – being aware of the dangerous combinatorics inherent in such formulations.

## 2 Knowledge-based Recognition by Production Systems

Context-free constrained multi-set grammars are discussed in [7] particularly with regard to graphical languages and computer interfaces. The basic idea is generalizing the generative string grammars by replacing the concatenation constraint by a more general constraint to be tested in higher dimensional space – such as a picture. This generalized grammar can then be employed to parse an image – taking basic image features such as lines as input and trying to derive complex structures: elements of language defined by the grammar. Next to their symbolic name the instances then have attributes – such as locations, orientations, etc. Basically, we have a finite set of non-terminals $N$ and terminals $T$ and a finite set of production rules $p : \mathtt{A} \rightarrow \Sigma$ where only context free forms are allowed, i. e., $\mathtt{A} \in N$ and $\Sigma = \mathtt{BC}$, $\mathtt{Bc}$, or $\mathtt{bc}$ with $\mathtt{B}, \mathtt{C} \in N$ and $\mathtt{b}, \mathtt{c} \in T$. In [10] recurrent subsystems of the form $\{p : \mathtt{A} \rightarrow \mathtt{bb}, q : \mathtt{A} \rightarrow \mathtt{Ab}\}$ are approximated by short-cut productions $s : \mathtt{A} \rightarrow \mathtt{b} \ldots \mathtt{b}$ for particular classes of constraints – such as adjacency or collinearity. Here clustering techniques or accumulator methods such as the Hough transform are included in a declarative way into the knowledge representation. Actually, this can not only be done on the terminal level but also with non-terminals yielding the form $s : \mathtt{A} \rightarrow \mathtt{B} \ldots \mathtt{B}$. The constraint is tested on a set of instances of the object class in $\Sigma$ where the size of this set is not fixed. In order to distinguish this form we call it *cluster_form*, while the classical production are said to have *normal_form*. An example production system is given in Table 1. For the classical language of such systems $P$ one root object $\mathtt{R} \in N$ must be reduced from a set of primitive instances $S$:

$$L_{reduce} = \{S : \{R\} \overset{*}{\rightarrow}_P S\} \tag{1}$$

where the asterisk has the usual meaning of successive right-to-left reduction using productions from $P$. In recognition tasks from out-door scenery usually clutter objects have to be tolerated. Thus the language is defined as the set of all primitive object sets $S$ where a set $X$ is reducible that contains a root object:

$$L_{cluttered} = \{S : R \in X \xrightarrow{*}_P S\}. \tag{2}$$

### 2.1 Recognition Using the Approximate Any-time Interpreter

The language definitions given in Section 2 are of combinatorial nature. To the best of our knowledge there exists no interpretation algorithm of polynomial computational complexity for such systems. In order to assure feasibility in the presence of input data containing several thousands of primitives – as they are commonly segmented from input images or videos – approximate interpretation is crucial. Algorithm 1 adapted from [6] gives such method in particular for $L_{cluttered}$. It works on a constantly growing database (DB) of object instances which is initialized by the set of primitive terminal instances $S$. It associates working hypotheses $(o, p, h, \alpha)$ with each instance $o$ in the DB where $p$ is the partner class (i.e. $\Sigma = op$ or $po$), $h$ is the hypothesis type (left-hand side) and $\alpha$ is the priority. All newly inserted instances cause a working hypothesis initially with $p, h = nil$ and $\alpha$ set from the quality of the object instance $o$. When such a hypothesis is encountered appropriate clones are constructed with $p$ and $h$ according to the productions in the system. At this point the priority can be changed.

As can be seen from Algorithm 1 production of *normal_form* are treated in the usual combinatorial way leading to a branching search tree. Productions of *cluster_form* are treated differently: Only the maximal set is reduced. All sub-sets which also may fulfill the constraint are not reduced. This violates the soundness. Here the algorithm only gives an approximate solution. For the justification of this see [10] and the definition of the maximal meaningful gestalt in [2]. Cluster analysis is often iterative. Here this means that for a *cluster_form* production $s : \mathtt{A} \rightarrow \mathtt{b} \ldots \mathtt{b}$ hypotheses $h_1 = (\mathtt{b}, \mathtt{b}, \mathtt{A}, \alpha)$ and $h_2 = (\mathtt{A}, \mathtt{b}, \mathtt{A}, \alpha)$ are permitted – the first for triggering objects b and the latter for triggering objects A. For justification see again [10]. An example is given with $p_{prolong}$ in Table 1. For each object *longline* a hypothesis is formed for further prolonging it. When such hypothesis triggers, a more accurate query can be posed giving a larger set of line objects to feed the production.

### 2.2 Re-evaluation Strategies Control the Search

For all kinds of search control it is demanded that every object concept should have a quality attribute with it assessing the saliency, relevance, or importance respectively. This allows comparing instances of the same object concept. In order to control the search using Algorithm 1 the importance of hypotheses interrelated with instances of different object concepts have to be compared, e. g., by weighting factors. So much for the data-driven bottom-up control. But more can be done. The importance of the hypotheses can be re-assessed with

```
repeat
    sort(queue);
    set_of_hypothesis = choose_best_n(queue);
    foreach trigger_hypo $\in$ set_of_hypothesis do
        if p=nil then
            foreach q where trigger_obj $\in$ right-hand side do
                adjust_priority(q);
                append_queue(trigger_elem, q, new_priority);
            end
        else
            actual_query = construct_query(trigger_hypo);
            candidate_set = select_DB(actual_query);
            switch p of type do
                case normal_form
                    foreach partner $\in$ candidate_set do
                        p:new_elem $\leftarrow$ (trigger_elem, partner);
                        insert_DB(new_elem);
                        construct_null_hypo(new_elem);
                    end
                end
                case cluster_form
                    p:new_elem $\leftarrow$ candidate_set;
                    insert_DB(new_elem);
                    construct_null_hypo(new_elem);
                end
            end
        end
        remove_queue(trigger_hypo);
    end
    foreach newly inserted element do
        re_evaluate all hypotheses;
    end
until root R found OR timeout OR queue=$\emptyset$ ;
```

**Algorithm 1**: Approximate Any-time Interpreter for Production Systems.

respect to the state of the DB reached. We distinguish two different classes of such importance calculation.

A very simple example for **global priority control** is delaying hypotheses $(o, p, h, \alpha)$ for triggering objects $o$ until the set of partner objects $p$ in the DB is not empty anymore. In the example given in Section 3 this is included in all variants. When there are multiple hypotheses $(o, p, h_1, \alpha)$, ..., $(o, p, h_k, \alpha)$ with the same triggering instance $o$ there often will be a preference for one of these inhibiting the others. For the system given in Table 1 actually three hypotheses are formed for objects *longline*, namely for $p_{stripe1}$, $p_{stripe2}$ and also for $p_{prolong}$ (see Section 2.1). According to the principle of maximal meaningful gestalts [2] the hypotheses for further prolonging has high preference over the others. Only after the prolongation production failed to produce any new instance the

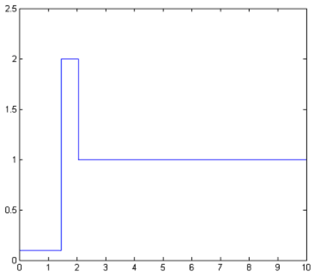

**Fig. 1.** Local re-evaluation function automatically derived from 'higher' productions.

other hypotheses regain their original priority. This is included in all our systems. Furthermore, in production systems with a hierarchy on the non-terminals higher priorities $\alpha$ can be chosen with rising hierarchy. In the example reported in Section 3 the priorities were chosen as linear functions of the quality $\alpha_{qual}$ in overlapping intervals with ascending hierarchy *hie* using appropriate offset $v$ and factor $w$:

$$\alpha_{min}(hie) \leq \alpha = v + (w\alpha_{qual}) \leq \alpha_{max}(hie) \tag{3}$$

This leads to a depth first search characteristic. More dynamically, a histogram of instance numbers for each symbol of the DB can be acquired with low computational effort. Hypotheses with $h$ being a frequent object type may be punished and rare ones rewarded. In the beginning of an interpretation run this also leads to a depth-first search characteristic. Later with rising computation time the search will get broader. For the sake of simplicity such control has not been used in the present system.

**Local priority control:** With every hypothesis tested there is a triggering object instance, and this instance is located somewhere in the corresponding attribute space. Also during the last execution of the *foreach* trigger_hypo block new object instances have been constructed. Of course while inserting them to the DB it was tested whether they are not already present there. The re-evaluation is based on the relation between the newly inserted objects and the triggering objects of the hypotheses. Since here all pairs of new objects and hypotheses have to be considered this causes considerable computational effort. In Table 2 these extra costs are shown in the column titled *top down*. The first possibility for such control is *general local inhibition*: other hypotheses of the same production with their corresponding triggering object located close to the instance at hand will lead to similar queries. Such control has been preliminarily investigated with a very simple toy production system in [11] using a smooth re-evaluation function. A similar strategy is appositely termed as *anticipation* or focus of attention: It uses declarative knowledge from productions that contain both, the newly built object and the object resulting from the triggered hypothesis. Figure 1 gives an example: On the abscissa here we have distance of

|  | success | partial | time out | failure | production | sorting | top down | complete |
|---|---|---|---|---|---|---|---|---|
| **with** | 15 | 13 | 6 | 7 | 8.22 | 8.18 | 0 | 17.25 |
| **without** | 22 | 11 | 2 | 6 | 2.03 | 2.80 | 3.76 | 8.89 |

**Table 2.** Number of fully/partial successful cases **with** and **without** anticipation control strategy and average runtimes in seconds.

a location from a newly built object *long-line*. All hypotheses building long-line objects are re-evaluated. If the corresponding objects are very close their priority will be multiplied by 0.1. The focus of attention is at a particular distance interval. Here the weighting factor will be 2. Otherwise, their priority will not be changed. Qualitatively, this is similar to the smooth re-evaluation functions used in [11], but has more parameters and is more focused – recall that this has to be defined for all combinations. Interesting here is that the particular distance and the width must be taken from "higher" productions. In the example the priority of hypotheses with triggering objects *line* is doubled because an object *longline* has been constructed in the vicinity and another such object is needed here in order to form an object *road*. The construction of the focus of attention uses the constraint predicate of the production forming road-stripe objects. This means that declarative knowledge and control knowledge interact – a dangerous but successful thing.

## 3 Experiments

The simple production system used here (compare Table 1) is designed to recognize bridges over highways. Knowledge – such as the expected width of single carriage ways used in production $p_{stripe2}$ (and in the localized top-down control) – can be obtained from Wikipedia. Experiments where made with a set of images taken during an evaluation run with a test-bed based on Google earth [9]. These images where picked blindly, i. e., the operator sees on the map layer of Google maps that there should be a landmark consistent with the model at that location. One of these 41 images actually doesn't contain a bridge – it is present in the map, but in the corresponding image there is a construction under way with the bridge removed. Some other pictures turn out to be difficult: Constructions under way, low contrasts, shadows, similar structures running parallel to the autobahn, etc. Not every failure of the system may be crucial depending on the task at hand. We distinguish for classes of recognition behaviour: (1) full success – the model has been instantiated properly with corresponding contours in the image, the location is precise; (2) semi success – the model has been instantiated partially correct, the localization is good enough for the task at hand; (3) no instantiation of the model could be achieved in the time limit – set in this experiment to 60 seconds; (4) false instantiation of the model to contours that are really caused by something else. Table 2 gives the success rates with and without the localized anticipation top-down control function automatically

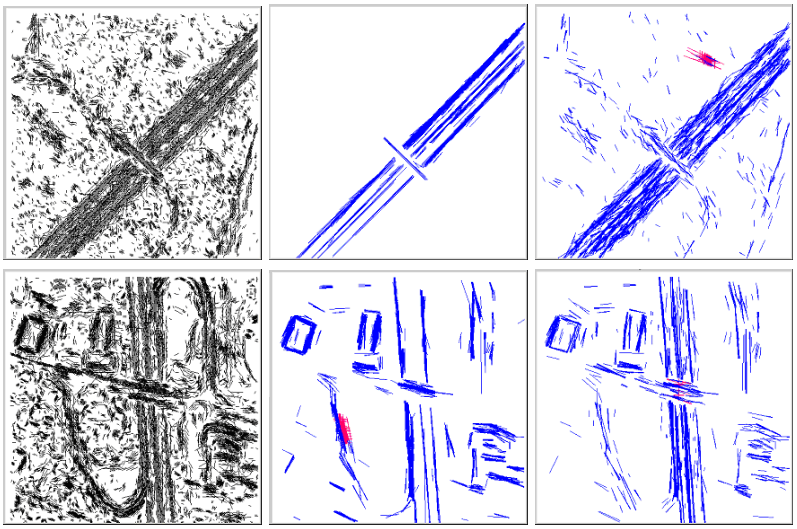

**Fig. 2.** Left terminal primitives for this image, center result with bottom up control, right result with localized top down control, upper row image 17 lower row image 4

derived from higher productions and indicated in Figure 1 (right). In particular in time critical applications we have fewer problems with recognition failure due to time out constraints. The anticipation control produces only one such error – the other time-out is correct because that is the image which contains no bridge. Whereas, without this control five time-out failures are produced – and also one correct time-out. All times have been obtained on the same server machine running eight processors at 3GHz. The instantiation of the productions is run in parallel threads – thus the time spent on them is not the major part. The administration of the process queue – in particular sorting and the control handling – is done serially. We observed one more failure of type (4) with the default control than with the anticipation control. This cannot be regarded as significant given the small test sample set. Because this is the most important type of failure for the assessment of the robustness of the system for the task at hand we would still like to learn something on this topic from this experiment. We therefore give two examples, where one control works and the other one fails. Example results can be seen in Figure 2. The sets of primitive objects are displayed in the left column (usually between 10.000 and 20.000 instances per picture). The upper example is from a forested area, the lower from a urban environment. In the forest example the south eastern contour of the bridge is very weak. Accordingly, the default control concentrates on the strong autobahn contours – until the time is over. This example counts as a failure of type (3) in table 1. On the same data the anticipation first instantiates the major contours of the autobahn. Then it spreads its focus of attention more into the forest. It also instantiates the critical weak contour of the bridge. However, before this can trigger any hypotheses other parallel structures in the forest are found to be consistent with the model. We counted this as a type (4) failure. Thus we see that the anticipation automatic is not always necessarily better in every single example. The lower example contains many adjacent salient structures that may well be confused with the landmark. Important contours of the autobahn are much weaker than some other contours caused by buildings. Such structures at the south-western part of the image happen to fit the autobahn model. Some other road structure then forms the desired "crossing the T". This is a clear type (4) failure. Nothing here corresponds correctly and the landmark position is localized completely wrong. On the very same data the anticipation control gives a full type (1) success. A fairly tight cluster of landmark instances is found in the correct position (in fact some of them also refer to the shadow of the bridge – which we tolerate as still correct).

## References

1. M. Arens and H.–H. Nagel: *Quantitative Movement Prediction based on Qualitative Knowledge about Behavior.* KI – Künstliche Intelligenz 2/2005, pp. 5–11.
2. A. Desolneux, L. Moisan, J.–M. Morel: *From Gestalt Theory to Image Analysis.* Springer–Verlag, Berlin, 2008.
3. E. Dickmanns: *Expectation-based Dynamic Scene Understanding.* In: A. Blake and A. Yuille (eds.) Active Vision. MIT Press, MA, USA, 1993, pp. 303–335.
4. L. Hotz, B. Neumann, and K. Terzic: *High-Level Expectations for Low-Level Image Processing.* In: A. Dengel, K. Berns, T. M. Breuel, F. Bomarius, and T. Roth-Berghofer (eds.): Proc. 31st German Conf. on Artificial Intelligence (KI–2008) Kaiserslautern, Germany, Sept. 23–26, 2008. LNCS 5243, Springer–Verlag, Berlin 2008, pp. 87–94.
5. T. Kanade: *Model Representations and Control Structures in Image Understanding.* In: R. Reddy (ed.): Proc. 5th Int. Joint Conf. on Artificial Intelligence (IJCAI77), Cambridge, MA, USA, Aug. 1977. William Kaufman 1977, pp. 1074–1082.
6. K. Lütjen: *BPI: Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung.* In: G. Hartmann (ed.): Mustererkennung 1986, 8. DAGM–Symposium, Paderborn, 30. Sept. – 2. Okt. 1986, Informatik Fachberichte 125, Springer–Verlag, Berlin 1986, pp. 164–168.
7. K. Marroitt, B. Meyer (eds.): *Visual Language Theory.* Springer–Verlag, Berlin 1998.
8. T. Matsuyama, V. S.–S. Hwang. *SIGMA a Knowledge-Based Aerial Image Understanding System.* Plenum Press, New York, 1990.
9. E. Michaelsen and K. Jäger: *A GOOGLE-Earth Based Test Bed for Structural Image-based UAV Navigation.* FUSION 2009, Proc. on CD, Seattle, WA, USA, ISBN 978-0-9824438-0-4, pp. 340–346.
10. E. Michaelsen, L. Doktorski, and M. Arens: *Shortcuts in Production Systems – A way to include clustering in structural Pattern Recognition.* Proc. of PRIA-9-2008, Nischnij Nowgorod, ISBN 978-5-902390-14-5, Vol. 2, pp. 30–38.
11. E. Michaelsen, L. Doktorski, and M. Arens: *Making Structural Pattern Recognition Tractable by Local Inhibition.* VISAPP 2009, Proc. on CD, Lisboa, Portugal, ISBN 978-989-8111-69-2, Vol 1, pp. 381–384.
12. H. Niemann: *Pattern Analysis and Understanding.* Springer–Verlag, Berlin 1989.
13. J. M. Tenenbaum and H. G. Barrow: *Experiments in Interpretation Guided Segmentation.* Artificial Intelligence Journal 8:3(1977) 241–274.