Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. 1991. 311-316.

The Analysis of Resource-limited Vision Systems

Ronald A. Rensink
Department of Computer Science
University of British Columbia
Vancouver BC, V6T 1Z4, Canada

Gregory Provan
Department of Computer and Information Science
University of Pennsylvania
Philadelphia PA, 19104-6389, USA

Abstract

This paper explores the ways in which resource limitations influence the nature of perceptual and cognitive processes. A framework is developed that allows early visual processing to be analyzed in terms of these limitations. In this approach, there is no one "best" system for any visual process. Rather, a spectrum of systems exists, differing in the particular trade-offs made between performance and resource requirements.[1]

Introduction

Consider a wildcat in its natural environment. If it is to catch prey and escape from predators, the cat must not only be able to process visual information, but must also do so in real time. Its visual system is therefore best explained not only in terms of limitations on the information available to the eye, but also in terms of limitations on other resources, such as time and space.

There is an increasing awareness – especially within the more computational sub-disciplines of cognitive science – that these more general resource limitations influence many kinds of perceptual and cognitive processes. For example, Cherniak [1984] argues that classical logics cannot form the basis for cognition because such cognition is computationally intractable; this has led to an examination of heuristics by which fast reasoning could take place [Levesque and Brachman, 1985; Levesque, 1989]. Similarly, Tsotsos[1987, 1990] has argued that the processes of early vision must have at most polynomial-time complexity if they are to be carried out in real time. But although there is an increasing appreciation of the role of resource limitations (e.g., [Bylander et al., 1989; Kasif, 1986; Rosenfeld, 1987]), no general framework for discussing these issues has emerged to date.

This paper discusses some of the issues that must be addressed in developing such a framework. In particular, it focuses on the influence of resource limitations on early visual processing. Marr [1982] has made a beginning in this domain, showing how vision can be analyzed in terms of constraints that allow good use to be made of the information available in the image. We will show how this framework can be expanded to handle other kinds of resource limitations, yielding added insight into the interconnections that exist among task, algorithm, and architecture. Since many of these issues are general ones, the framework presented here will contain elements that are also applicable to other areas of perception and cognition.

Resource Limitations and Explanation

Many of the earlier analyses based on resource limitations (e.g., [Norman and Bobrow, 1975]) focused on limitations in the system architecture, for example, limited memory or channel capacity. These did not yield the insights that had originally been hoped for; indeed, it has been argued [Navon, 1984] that such limitations are inherently incapable of leading to unequivocal insights into the operation of perceptual and cognitive processes.

But architectural limitations are not the only kind that arise – more general limitations also exist, such as limits on the available information, and on the time and space allowed for a computation. These "processor-indifferent" limits are potentially more powerful than those based on architectural limitations, essentially describing the structure of the *task itself.*

Given that these general limitations must be taken into account, how might they be used to analyze the underlying mechanisms? One of the most successful approaches to date has been the computational framework put forward by David Marr [1982], in which visual processing is analyzed in terms of constraints that allow good use to be made of the information available in the image. In what follows, we will show that this framework can be expanded to accommodate not only limits on available information, but other kinds of resource limits as well, and that such a revised frame-

[1]The work of R.A. Rensink was supported by NSERC grant, via R.J. Woodham. Much of the work by G. Provan was done at the University of British Columbia, under NSERC grant A9281. The authors would like to thank Marion Rodrigues and Marc Romanycia for their comments on earlier versions of this paper. Many thanks also to Bob Woodham for his support of R.A. Rensink.

work can lead to a new understanding of several aspects of early vision.

Marr's Framework

According to Marr [1982], a complete analysis of a visual process involves three distinct levels of explanation:[2]

1. *Computational level*. Analysis at this level is entirely concerned with the specification of the task itself. This consists of two parts: (i) describing the constraints that exist between the input of a visual process and its output, and (ii) describing the reasons *why* these constraints have been chosen.
2. *Algorithmic level*. This level views explanation in terms of the representations and algorithms used for the process. More precisely, an "algorithmic" explanation is a constructive demonstration that there exists a formal algorithm sufficient to perform the required task.
3. *Implementational level*. This level is concerned with the physical substrate on which the algorithms are implemented. An "implementational" explanation is a constructive demonstration that there exists a physical system sufficient to carry out the required computation.

One of the great strengths of Marr's approach is its recognition of a "computational" level of explanation, in which emphasis is placed upon determining the *what* and the *why* of the particular operations being carried out. This has helped clarify our understanding of several processes of low-level vision, including edge detection [Marr and Hildreth, 1980], stereopsis [Marr and Poggio, 1979], and motion perception [Hildreth, 1984]. Consider, for example, the computational analysis of stereopsis. Determining the *what* consists of finding the constraints on the acceptable correspondences between features in the left and right images, and constraints on the form of the recovered surface. These constraints must be sufficient to describe a unique mapping between the image and the resulting map of disparity estimates. Determining the *why* essentially consists of a demonstration that these constraints serve to allow a satisfactory recovery of disparity estimates from the image pairs.

Hence, the stereopsis problem can be seen as the specification of a mapping from a given set of image pairs to a set of (reconstructed) surfaces. This mapping can easily be described by its "extension", viz., a list of the pairings made between individual images and surfaces. Such a description, however, does not really provide an explanation for the process, any more than a list of planetary positions over some given interval explains their motions. Explanation must involve a description of the "invariants" or "deep structure" that underlie the particular mapping that is made. The constraints sought for at the computational level provide exactly this kind of explanation. To justify the choice of a particular set of constraints (explaining *why*) requires showing that the constraints lead to an acceptable set of associations between image and scene in the world under consideration.

But although Marr's approach has helped explain several parts of low-level vision, it has not helped in our understanding of many others, e.g., color perception or texture perception [Morgan, 1984]. For example, in texture perception, it is the resources available to the processor (e.g., time and space) which are relatively scarce, rather than the information in the image. Marr's framework cannot handle such matters, since the computational level of analysis (implicitly) assumes that perception relies on processors with unlimited computational resources.[3]

Resource Limitations and Constraints

To see how these more general kinds of resource limitations influence the operation of a visual process, it is important to note that these limitations fall into three main groups:

1. *Projective limitations.* The available information in the image may be considered a basic resource acquired by the sensors of the system; the limitations on this resource stem from the way in which the scene is projected to the image. The type and amount of available information may strongly influence the kinds of computations that can be performed; if so, the process can be characterized as "data-limited" [Norman and Bobrow, 1975].
2. *Computational limitations*. A processor is also limited by many aspects of the *way* in which is operates, aspects which have no direct connection with its physical composition. Although many of these are specific to the particular computational architecture used (e.g. the particular set of elementary operations available, bandwidths, etc), more general ones also exist. It is this latter set of resources – in particular the time and space required for a computation – that will be considered here. Limitations on these resources will be referred to here as *complexity limitations*.
3. *Physical limitations*. A processor is also governed by limitations stemming from its physical make-up. Again, many of these quantities refer to the particular architecture of the processor. But limitations also arise from more general considerations, such as the matter and energy required for a given task [Bennett, 1982; White, 1988].

[2]It is important to note that Marr considers explanations at each level to be essentially independent of those at the other two [Marr, 1982, chapter 1]. For example, analysis at the algorithmic level is not concerned with ultimate purpose not does it depend on any details of implementation.

[3]Marr [1982] does consider efficiency to be important, but only once the task itself has been laid out. As such, it does not enter into the general analysis carried out at the computational level.

In order to overcome these sets of limitations, a processor must impose corresponding sets of constraints on its operation. To completely understand a given process, therefore, is to understand these sets of constraints.[4] Thus, for example, in Marr's framework, projective limitations are the only kind that the visual system is considered to grapple with. To make up for such lost information, a corresponding set of *projective constraints* [5] is needed on the mappings between image and scene; essentially, these determine which of the many possible scenes actually corresponds to a given image.

But such "processor-indifferent" explanations need not be restricted to invariants of the *form* of this mapping – there may also exist a set of constraints on the *algorithm* and *representation* used to obtain it. More generally, such "complexity" constraints describe the resources used by a given process. This in turn limits the kinds of mappings that can be made. To completely explain the form of a mapping, then, both projective and complexity constraints will usually be required. Only when computational resources are unlimited (as assumed in Marr's approach) will projective constraints alone be enough to explain a visual process.

Note that in Marr's framework, no general constraints are imposed on an algorithm, so that they often have a large element of the *ad hoc*, being based on current beliefs of psychology and physiology. But complexity constraints can provide such general guidelines, thereby substantially reducing the need for the *ad hoc* element in any particular model.

A Revised Framework

A computational explanation of a visual process, then, will include a description and justification of the projective, complexity, and physical constraints imposed to handle the corresponding types of resource limitations.[6] Different levels of explanation still exist, but are now based on the degree of generality of the constraints, rather than on issues of abstract mapping, process, and implementation:

1. *Computational level*. This includes not only the projective constraints, but also those complexity and physical constraints that are "processor-indifferent".
2. *Algorithmic level*. This involves the more specific complexity and physical constraints that are placed on the "internal" structure of the system to give the algorithm and representation a unique determination; since projective constraints have no further bearing on this matter, they are necessarily absent from this level.
3. *Implementational level*. Although not developed here, it is apparent that this level concerns the remaining constraints on the particular system being modelled.

Thus, the three levels of Marr's framework are preserved in large measure – analysis still occurs at each of the computational, algorithmic, and implementational levels. But the constraints required at the computational and algorithmic levels of analysis have been tightened up significantly, due to new sets of constraints. The most important change, however, is that analysis is based on the *generality* of the constraints. Since constraints on algorithms and implementation result from both general and more specific constraints, this provides an interesting linkage between mappings, algorithms, and implementation.

Analyzing Vision Systems

The explanation of a visual process is essentially the description and justification of the projective, complexity, and physical constraints that govern its operation. Since the use of projective constraints is already an integral part of "conventional" analysis, and physical constraints are not considered here, we will focus on the way in which complexity constraints can be used to analyze the operation of a vision system.

The theory of computational complexity (cf. [Garey and Johnson, 1979]) can be used to formalize many of the concepts pertaining to complexity constraints. Specifically, it can define the time and space requirements of particular tasks, independent of the algorithm or architecture. For the purpose of this paper, it is sufficient to distinguish tasks which can be solved quickly (e.g. the class P of tasks solvable within a time proportional to some polynomial of the input size), from more time-consuming tasks (e.g., the class of NP-complete problems, which – in the worst case – require time that increases exponentially with the size of the input).[7]

[4] In what follows, 'limitations' will be used when referring to resource limits imposed *on* the system, limits over which the process has no control (e.g., total amount of time, space, energy, etc). These must be distinguished from 'constraints', which are imposed *by* the system itself to make good use of its available resources. When talking about a system, the term 'constraint' will only be used in this latter sense.

[5] The term 'projective constraint' is meant to replace 'computational constraint' as used n Marr's framework. The alternate term is used to avoid confusion between constraints placed on the form of the "interpretation mapping" between image and scene, and the constraints on the algorithm used to compute it.

[6] In this paper, much of the focus will be on complexity constraints, since projective constraints are relatively well understood, and physical constraints add little (at least at this stage of development) to what can be learned by discussing complexity constraints.

[7] It is entirely possible that the average resource use is much more representative than the worst-case situation, and this may be used as one of the complexity measures. However, worst-case situations must still be dealt with.

Efficient Use of Resources

If the computational demands of a task exceed the resources available, it is obvious that the task will need to be reformulated. However, this reformulation may be kept to a minimum by making efficient use of the time and space that is available. As used here, the term "efficient" does not necessarily mean optimal; rather, all that is meant is that relatively little time or space is wasted.

The efficient use of resources depends greatly upon the choice of particular algorithms and representations used in a process. However, there exist a few general considerations that are relevant:

1. *Parallelism*. Perhaps the most obvious way of reducing the time required for a task is to carry it out in parallel. However, it must be noted that it is not always possible for processes to take advantage of parallelism – e.g., there can be no reduction in time for Constraint Satisfaction Problems [Kasif, 1986], since they are inherently sequential. Thus, if a task is to take advantage of parallelism, it must be such that most of its computations can be done locally. But each of these local computations must operate within the given time, and – taken as an ensemble – they must also operate within the given space.

2. *Resource trade-offs*. The specification of a desired level of performance does not uniquely determine the exact resources necessary to attain it – trade-offs between various computational resources can still be made. One well-known example of this is the trade-off between time and space: for instance, look-up tables can be used instead of computing values on demand. Thus, Goad [1983] presents an object-recognition scheme where the poses of objects are pre-computed so that viewpoint determination is speeded up. Alternatively, redundant coding can often be used to decrease processing time (see, e.g., [Arbib, 1987, pp 87-89]). Note that this use of redundant representations contrasts with Marr's approach, in which the goal is to use nonredundant (orthogonal) systems of representation as much as possible.

3. A priori *knowledge*. One final consideration that also enters into the efficient use of resources in visual processing is the possible use of "high-level" *a priori* constraints based on the particular characteristics of the objects in the scene. In many cases, higher-level constraints could significantly reduce the computational complexity of a process; if these constraints could be selectively "loaded into" lower-level processes, this could often achieve a considerable speedup of processing. The relation between early vision and later levels of processing is a complex one, and will not be discussed here, but it is worth point out that if such "downloading" of a priori knowledge does indeed occur, issue of resource use will prove to be critical for its analysis.

Performance Trade-offs

Even though a process is as efficient as possible, it may still be impossible to carry it out using the available time and space. If so, the process cannot be used; it must be replaced by one that *does* satisfy the resource constraints. The efficiency of such "approximating" processses is obtained by lowering the quality of the mapping between input and output. More generally, there is usually a trade-off between the complexity of the mapping between image and scene, and the resources required to compute the mapping. Thus, depending on available computational resources, the visual process most suitable for a particular task can range from "traditional" process that use unlimited computational resources, to "quick and dirty" systems that require only a small amount of time and space. Part of a computational explanation of a visual process is therefore to specify what the particular choice of trade-off is, and why it was made.

There appear to be some general aspects to the methods by which performance can be "gracefully" traded off for reduced computational complexity, and it is likely that these strategies will enter into many of the particular processes of early vision. A few of these strategies (together with some possible applications) will now be discussed in regards to the reduction of processing time:

1. *Reducing quantity of input*. In general, more information requires more computation time [Levesque and Brachman, 1985]. Thus, one way to reduce time is to reduce the amount of data in the input that has to be handled. For example, visual search is an NP-complete problem, requiring time that increases exponentially with the size of the input [Tsotsos, 1987]. This time can be reduced (see below) by taking advantage of the coherence and uniformity of the world to represent the original image by a smaller set of coarser-grained patterns that could be comfortably handled with the available resources. As the grain of these patterns increases, the number of distinctions that can be made in the input decreases; however, these distinctions may be quite suitable for many purposes.

2. *Reducing the quantity of output*. Given that computational complexity can be reduced by effectively reducing the amount of information in the input, a natural "dual" would be to reduce complexity by reducing the amount of information in the output. Such outputs would contain coarser-grained descriptions of the more important aspects of the scene. Note that this "coarse grain" need not always correspond to a diminished resolution in some property such as spatial location or velocity of motion; instead, the "equivalence classes" of outputs [8]

---

[8]An *equivalence class* is defined as the set of algorithms and representations which carry out the same mapping while using the same information content and computational resources.

could be based on such things as topological properties. In a very general sense, then, these outputs may be regarded as providing qualitative descriptions of the scene. For example, in Marr's theory it is assumed that the three-dimensional structures of objects in space are represented as point-by-point mappings of local depth and/or orientation. Such representations are difficult to compute, and it is possible that they are not computed at all. Indeed, it appears that more qualitative descriptions – such as descriptions of affine or ordinal structure – may provide all the information that is required by subsequent processes [Todd and Bressan, 1990].

3. *Reducing quality of the mapping*. Reducing the information in the input and output of a mapping is often sufficient to reduce complexity, but it isn't always necessary. For many processes, the availability of a spatiotopic array of processors is sufficient to allow them to be carried out in constant time. For example, a simple remapping of all intensities in an image can be done immediately on a parallel array, no matter how much information is contained in the input and output. A process can therefore trade off performance against complexity by altering the nature of the mapping itself; essentially, it is the quality of the mapping that is being traded off. This is the strategy adopted in rapid line interpretation (see below) – increased speed is obtained by reducing the validity and global coherence of the recovered scene.

Examples

To illustrate how resource constraints can help explain various aspects of visual perception, we will briefly sketch how this approach can be applied to two particular processes in early vision: visual search and the pre-attentive recovery of three-dimensional orientation from line drawings. If described in the "conventional" way, i.e., making optimal use of information, both problems are NP-complete. But the processes of early vision are generally carried out within several hundred milliseconds, making it unlikely that these problems can be formulated in this way. This suggests a shift in the way these processes should be viewed: instead of making optimal use of information, they appear instead to emphasize "quick and dirty" performance.

Visual Search

One of the first treatments of complexity in early vision was that of Tsotsos [1987, 1990], who analyzed the process of visual search. Here, the problem is to determine as rapidly as possible the presence or absence of a known target pattern in an image. Tsotsos showed that if optimal decisions are to be made, this problem is NP-complete, requiring an exponential amount of time in the worst case. This is at odds with evidence that many kinds of targets can be reliably detected within several hundred milliseconds, while others require a time directly proportional to the number of items in the image (e.g., [Treisman and Gormican, 1988]).

The first step in the analysis is to determine the extent to which time and space can be reduced while maintaining optimal detection performance. Tsotsos shows that hierarchical coding can help to minimize the resources required, but that the problem still remains NP-complete, since the target must be compared again all possible aspects of all possible subsets of the image. If visual search is to be carried out rapidly, this can only be done for a select group of image subsets (or equivalently, a select group of target patterns).

Even by defining preferred patches as convex patches with uniform properties (arising from the convexity and uniformity of objects in the physical world), the time required is still too high to be compatible with the complexity constraint. Consequently, a more radical step is taken: information is thrown away. This is done both by reducing spatial resolution in the basis set of patterns and by reducing the number of properties that can be considered at any one time. Although the completeness of the system is thereby sacrificed, these constraints do allow an architecture to be specified that is compatible with the time and space limitations generally found in biological systems. Interestingly, this architecture has many of the general characteristics of the human visual system, viz., a small set of physically separated spatiotopic maps, columnar organization of processors, and coarse coding of local properties [Tsotsos, 1987, 1990].

Rapid Interpretation of Line Drawings

Another problem in which time limitations play an important part is the rapid interpretation of line drawings. The goal of the line interpretation task is to recover the three-dimensional structure of opaque polyhedral objects from line drawings describing their projection onto the two-dimensional image plane (see e.g.,[Sugihara,1986]). Interpretations generally take the form of a labeled drawing in which each line element (or region) is assigned a unique interpretation as a three-dimensional structure (e.g., that the line has a particular three-dimensional orientation, or that it forms the boundary of the object being viewed, etc.). This process has been shown to be NP-complete [Kirousis and Papadimitriou, 1985], ruling out the possibility that it is carried out in early vision.

However, it has recently been shown (Enns and Rensink [1990, 1991]) that the three-dimensional orientation of some objects can be recovered at early stages of visual processing, within several hundred milliseconds of display onset. As in the case of visual search, then, optimal use of information is not to be expected for such a process; part of the explanation must involve complexity as well as projective constraints.

In the model proposed by Enns and Rensink, interpretation is accomplished via two parallel stages, each of which involves only a small number of steps. The first stage is carried out in parallel on each trilinear junction in the image. These junctions are places in the image where three line segments join up. Any junction may correspond to several three-dimensional structures in the scene; here, however, only the most likely interpretation is assigned. Once these initial interpretations have been established, consistency is tested by comparing the interpretations at each junction against those of their immediate neighbors. This can be done within some constant time simply by propagating the local estimates along the lines that connect the junctions.

If such a "quick and dirty" process is used in the early stages of vision, this should allow some line drawings to "pop out" of a display on the basis of the three-dimensional orientation of the block they describe.[9] Enns and Rensink [1991] show that this is exactly what happens.

These findings suggest that other "quick and dirty" processes may also be used in early vision. For example, it is possible to rapidly determine the concavity/convexity of surfaces, based on the patterns of shading in the image [Ramachandran, 1988]. If these processes are representative of the operations carried out at early levels, this will force a new look at the nature of early visual processing.

References

[Arbib, 1987] MA Arbib. *Brains, Machines, and Mathematics (2nd ed.)* New York: Springer, 1987.

[Bennett, 1982] CH Bennett. The Thermodynamics of Computation – a Review. *International Journal of Theoretical Physics*, **12**: 905-940, 1982.

[Bylander et al., 1989]. T Bylander, D Allemang, MC Tanner, and J Josephson. Some Results Concerning the Computational Complexity of Abduction. In *Workshop on Principles of Knowledge Representation and Reasoning*, 1989.

[Cherniak, 1984] C Cherniak. Computational Complexity and the Universal Acceptance of Logic. *Journal of Philosophy*, **81**(12): 739-758, 1984.

[Enns and Rensink, 1990] JT Enns and RA Rensink. Sensitivity to Three-Dimensional Orientation in Visual Search. *Psychological Science*, **1**: 323-326, 1990.

[Enns and Rensink, 1991] JT Enns and RA Rensink. Preattentive Recovery of Three-Dimensional Orientation from Line Drawings. *Psychological Review*, **98**: 335-351, 1991.

[9] Note that the result of this process will generally not be a valid reconstruction of the scene, since successful recovery can be expected only at the relatively sparse set of locations in the visual field where the projective assumptions describe the actual situation. Such a description, however, would still be valuable, since it might be expected that at least a few of its estimates would be accurate, and so guide fast-acting processes that – on average at least – will be helped more than hindered.

[Garey and Johnson, 1979] MR Garey and DS Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: WH Freeman, 1979.

[Goad, 1983] C Goad. Special Purpose Automatic Programming for 3D Model-based Vision. In *Proc. AAAI*, 349-353, 1983.

[Hildreth, 194]. EC Hildreth. *The Measurement of Visual Motion.* Cambridge, MA: MIT Press, 1984.

[Kasif, 1986] S Kasif. On the Parallel Complexity of Some Constraint Satisfaction Problems. In *Proc. Conf. Of AAAI,* 349-353, 1986.

[Kirousis and Papadimitriou, 1985] L Kirousis and CH Papadimitriou. The Complexity of Recognising Polyhedral Scenes. In *26th FOCS*, 1985.

[Levesque and Brachman, 1985] H Levesque and RJ Brachman. A Fundamental Tradeoff in Knowledge Representation and Reasoning. In RJ Brachman and H Levesue, eds., *Readings in Knowedge Rpesentation*, 41-70. Los Altos CA: Morgan Kaufman.

[Levesque, 1989]. H Levesque. Logic and the Complexity of Reasoning. Tech Report KRR-TR-89-2, Dept. of Computer Science, University of Toronto, Toronto, Canada, 1989.

[Marr and Hildreth, 1980] D Marr and EC Hildreth. Theory of Edge Detection. *Proc Roy. Soc. Lon B.*, **207**:187-217, 1980.

[Marr and Poggio, 1979] D Marr and T Poggio. A Computational Theory of Human Stereo Vision. *Proc. Roy. Soc. London B.*, **204**: 301-328, 1979.

[Marr, 1982] D Marr. *Vision.* San Francisco: Freeman, 1982.

[Morgan, 1984] MJ Morgan. Computational Theories of Vision. *Quarterly J. Exp. Psych.*, **36A**: 157-165, 1984.

[Navon, 1984] Resources: A Theoretical Soup Stone? *Psychological Review*, **91**: 216-234, 1984.

[Norman and Bobrow, 1975] DA Norman and DG Bobrow. On Data-limited and Resource-limited Processes. *Cognitive Psychology*, **7**: 44-64, 1975.

[Ramachandran, 1988] V Ramachandran. Perceiving Shape from Shading. *Scientific American*, **Aug**: 76-83, 1988.

[Rosenfeld, 1987] A Rosenfeld. Recognizing Unexpected Objects: A Proposed Approach. In *Proc. of the DARPA Image Understanding Workshop*, 620-627, 1987.

[Sugihara, 1986] K Sugihara. *Machine Interpretation of Line Drawings*. Cambridge, MA: MIT Press, 1986.

[Todd and Bressan, 1990] JT Todd and P Bressan. The Perception of 3-Dimensional Affine Structure from Minimal Apparent Motion Sequences. *Perception & Psychophysics*, **48**: 419-430, 1990.

[Treisman and Gormican, 1988]. A Treisman and S Gormican. Feature Analysis in Early Vision: Evidence from Search Asymmetries. *Psychological Review*, **95**: 15-48, 1988.

[Tsotsos, 1987] JK Tsotsos. A "Complexity Level" Analysis of Vision. *Int. J. Computer Vision*, **1**: 346-355, 1987.

[Tsotsos, 1990] JK Tsotsos. Analyzing Vision at the Complexity Level. *Behavioral and Brain Sciences*, 13: 423-445, 1990.

[White, 1988] I White. The Limits and Capabilities of Machines – A Review. *IEEE Trans*, **SMC-18**: 917-938, 1988.