Linguistics, Morristown, NJ, April 1991), pp. 15–20.
39. K. W. Church, in *Proceedings of the 2nd Conference on Applied Natural Language Processing* (Association for Computational Linguistics, Morristown, NJ, February 1988), pp. 136–143. For an application of this algorithm, see R. Weischedel, M. Meteer, R. Schwartz, J. Palmucci, *Applying Probabilistic Models to Knowledge-Based Algorithms in Natural Language Processing, Technical Report* (Bolt, Beranek, and Newman, Cambridge, MA, 1991).
40. S. DeRose, *Computational Linguistics* **14**, 31 (1988).
41. W. Francis and H. Kucera, *Frequency Analysis of English Usage: Lexicon and Grammar* (Houghton Mifflin, Boston, MA, 1982).
42. E. Brill, personal communication.
43. M. R. Brent and R. Berwick, in *Proceedings of the DARPA Workshop on Spoken Language Systems* (Morgan Kauffman, Palo Alto, CA, February 1991), pp. 342–345.
44. M. R. Brent, in *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics* (*EACL*), Berlin (Association for Computational Linguistics, Morristown, NJ, April 1991), pp. 222–226.
45. D. Hindle, in *Proceedings of the Association for Computational Linguistics Conference*, Pittsburgh (Association for Computation Linguistics, Morristown, NJ, June 1990), pp. 268–275.
46. F. Smadja and K. McKeown, in *Proceedings of the Association for Computational Linguistics Conference*, Pittsburgh (Association for Computation Linguistics, Morristown, NJ, June 1990), pp. 252–259.
47. M. Liberman, *Guidelines for the Linguistic Data Consortium*, Report prepared for the Defense Advanced Research Projects Agency (DARPA), May 1991. See also the announcement by DARPA in the Commerce Business Daily Announcement (Department of Commerce, Washington, DC, 23 July 1991).
48. H. J. Hutchins, *Machine Translation: Past, Present and Future* (Ellis Horwood, Chichester, 1986).
49. M. Nagao, *Machine Translation: How Far Can It Go?* (Oxford Univ. Press, Oxford, 1989). [Translation by N. D. Cook of *kikei hon'yaku wa doko made kano ka* (Iwanami Shoten, Tokyo, 1986).]
50. S. Nirenberg, Ed., *Machine Translation: Theoretical and Methodological Issues* (Cambridge Univ. Press, Cambridge, 1987).
51. J. Slocum, Ed., *A Survey of Machine Translation: Its History, Current Status and Future* (Cambridge Univ. Press, Cambridge, 1985).
52. Y. Wilks, in *Proceedings of the International Forum for Translation Technology*, IFTT (IFTT, Oiso, Japan, April 1989), pp. 56–62.
53. R. M. Kaplan, K. Nutter, J. Wedekind, A. Zaenen, in *Proceedings of the European Association for Computational Linguistics* (*EACL*) *Conference* (Association for Computational Linguistics, Morristown, NJ, April 1989), pp. 272–281.
54. S. M. Shieber and Y. Schabes, in *Proceedings of the International Conference on Computational Linguistics* (*COLING-90*) (University of Helsinki, Helsinki, August 1990), vol. 3, pp. 253–258.
55. A. Abeille, Y. Schabes, A. K. Joshi, in *Proceedings of the International Conference on Computational Linguistics* (*COLING-90*) (University of Helsinki, Helsinki, April 1990), vol. 3, pp. 1–6.
56. J. Tsujii and K. Fujita, in *Proceedings of the European Association for Computational Linguistics* (*EACL*) *Conference* (Association for Computational Linguistics, Morristown, NJ, April 1991), pp. 275–280.
57. J. G. Carbonell, in *Machine Translation*, S. Nirenberg, Ed. (Cambridge Univ. Press, Cambridge, 1987), pp. 68–89.
58. H. L. Sommers, J. Tsuji, D. Jones, in *Proceedings of the International Conference on Computational Linguistics* (*COLING 90*) (University of Helsinki, Helsinki, August 1990), pp. 271–276.
59. J. Tsujii and M. Nagao, in *Proceedings of the International Conference on Computational Linguistics* (*COLING-88*) (John von Neumann, Society for Computing Sciences, Budapest, August 1988), pp. 688–693.
60. H. L. Sommers, in *Proceedings of a Workshop on Machine Translation* (University of Texas, Austin, TX, June 1990).
61. P. F. Brown *et al.*, in *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation* (Carnegie Mellon University, Pittsburgh, PA, June 1988).
62. W. A. Gale and K. W. Church, in *Proceedings of the Association for Computational Linguistics* (*ACL*) *Conference* (Association for Computational Linguistics, Morristown, NJ, June 1991), pp. 177–184.
63. This work was partially supported by ARO grant DAAL03- 89-0031, DARPA grant N00014-90-J-1863, and NSF STC grant DIR-8920230. I want to thank E. Brill, B. Cheikes, R. Frank, D. Griesbach, S. Heyner, A. Kroch, M. Liberman, M. Marcus, R. Pito, P. Resnik, O. Rambow, Y. Schabes, M. Steedman, M. Walker, B. Webber, and R. Weischedel for their valuable help in the preparation of this article.

Computer Vision

YIANNIS ALOIMONOS AND AZRIEL ROSENFELD

The field of computer vision is devoted to discovering algorithms, data representations, and computer architectures that embody the principles underlying visual capabilities. This article describes how the field of computer (and robot) vision has evolved, particularly over the past 20 years, and introduces its central methodological paradigms.

VISION IS THE MOST POWERFUL SENSE FOR MANY LIVING organisms, including humans. We take it so much for granted, because it is ordinarily so effortless, that we often fail to seriously consider how it works. Students of visual perception work in diverse fields, including neuroanatomy and physiology, psychology, computational and robot vision, and engineering. But researchers in different fields ask different questions about vision. Some ask empirical questions: How are existing biological visual systems actually designed? On the other hand, scientists and engineers try to answer theoretical and normative questions. The theoretical question in vision is, What is the range of possible mechanisms underlying perceptual capabilities in vision systems? The normative question is, How should a particular class of vision systems (or robots) be designed so that it can efficiently perform a set of specific visual tasks? The three types of basic questions do not in general have the same answers.

A very large part of the human brain is devoted to visual perception (*1*) (Fig. 1). Computational algorithms are implemented in this massive network of neurons; they obtain their inputs from the retina, and produce as output an "understanding" of the scene in view. But what does it mean to "understand" the scene? What algorithms and data representations are used by the brain? Analogously, given a set of images acquired by a TV camera, what computer architectures, data structures and algorithms should we use to create a machine that can "see" as we do? (*2–4*) (Fig. 2).

Many organisms possess visual capabilities, and their visual systems are not structured in the same way; moreover, they live in different environments and use vision for different purposes. But although a given visual capability, say for obstacle avoidance, is not necessarily implemented in the same way in the fly, the rat, and the human, the principles underlying this ability may be the same. It is these principles that are the subject of research in computer vision. As our understanding of visual principles advances, we can build robots that perform various tasks through the use of vision.

The authors are at the Center for Automation Research, University of Maryland, College Park, MD 20742–3411.

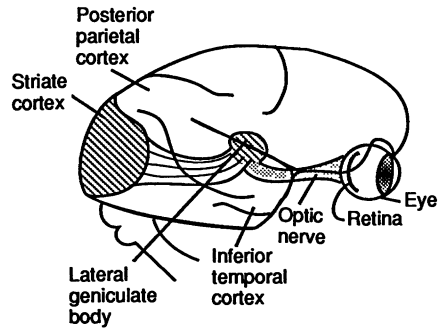


Fig. 1. Visual areas in the brain. The eye forms an optical image of the scene on the retina, an array of photoreceptors which discretely sample the image. Other layers of cells associated with the retina perform various types of local computations on the sampled image. The processed image is transmitted in parallel along the optic nerve, through the lateral geniculate body, to the striate visual cortex, where additional types of local operations are performed; these operations are sensitive to the presence of local features such as spots, bars, and edges in the image. The outputs of these operations are transmitted to other areas of the cortex, particularly to the posterior parietal cortex and the inferior temporal cortex, where global properties of the image appear to be analyzed.

The Goal of Image Understanding

What is vision for? Why do organisms have vision and why do we want to equip robots with it? We use vision (and other senses) to interact with our environments and survive—to navigate and avoid obstacles, to recognize and pick up objects, to identify food and danger, friends and enemies. In other words, we use vision to perform visual tasks; we engage in many kinds of behaviors that are guided by visual inputs.

How can we study the principles of visual perception? Should we study individual tasks? Because the nature of a task depends on the agent and the environment, what kind of agent should we assume in this study—an insect, a human, a robot? Such questions demonstrate that one of the hardest problems we encounter in the study of visual perception is what questions to ask.

A set of specific questions was formulated in the work of the late David Marr (*5*), who suggested that we can study the principles of visual perception by considering the purpose of vision as describing scenes. In other words, we should regard the task of vision as being the construction of a detailed representation of the physical world, independent of the tasks under consideration. This viewpoint is prevalent; for example, we read in a recent survey article (*6*, p. 389) that "The goal of an image-understanding system is to transform two-dimensional data into a description of the three-dimensional spatiotemporal world . . . [Such a system] must infer 3D surfaces, volumes, boundaries, shadows, occlusion, depth, color, motion, ..."

Regarding the central goal of vision as scene recovery makes sense. If we are able to create, using vision, an accurate representation of the three-dimensional (3D) world and its properties, then using this information we can perform any visual task. Because we will know where obstacles are, we can avoid them; because we will have an accurate representation of an object's properties, we will be able to match it against models in a database of possible objects and recognize it. The recovery methodology also allows us to study vision in isolation—something which is very desirable in the early stages of development of any new field.

The next section describes research on the problem of recovery, which has given rise to some interesting mathematical problems. It should be noted that treating recovery as a central vision problem raises a theoretical question: What properties of a scene can be recovered by means of vision? The study of the recovery problem tells us about the mathematical relationships between properties of the image and properties of the physical world. It does not necessarily tell us how to build a vision system for a specific purpose, because in order to perform a given visual task we may not need to fully recover the scene; but it does shed some light on the problems of designing visual systems.

Scene Recovery: A Theory of Computer Vision

If we need to treat vision as a recovery problem, it is necessary to make the image formation process explicit. Consider the abstract model of an observer in Fig. 3. The light going through a pinhole camera creates an image, that is, an array of light intensities (in digital form the intensities are represented by numbers) (Fig. 4). The problem of recovery is the "inverse optics" problem; optics maps the world onto the image, vision attempts to invert the process. It is important to note that recovery tasks and their applications (navigation, recognition, and so on) become easier if one has a specific model for the class of scenes in question. In this article, however, we will focus on techniques based on general scene models and on general-purpose tasks that visual systems might need to perform.

The early years (1955–1970). The earliest work in the field dealt with the analysis of single images of static scenes. A great deal of effort was expended on images of "scenes" that are (approximately) two-dimensional (2D): documents, micrographs (where, because of the shallow depth of field, the image is an "optical section" of the specimen), and images of the earth's surface taken from high altitudes (in which terrain relief is negligible); the interpretation of such images is usually called "pattern recognition," not computer vision.

When work on robot vision began in the 1960s, it initially concentrated on the so-called "blocks world," that is, the scene was assumed to consist of a set of polyhedra. Because the image is a perspective projection of the scene, geometric analysis yields useful relationships between parameters of the block edges in the image and the 3D structure of the blocks. Such relationships can be used to recover geometric properties, such as the concavity or convexity of an edge [see (7) for a review].

At the same time a lot of work was done on what was called "low-level" processing, much of it devoted to the extraction of "important" intensity changes (edges) in an image. In a blocks-world image these should correspond to depth and slope discontinuities or to shadow boundaries. Edge detection was usually achieved by convolving images with local operators and thresholding the results. Finding homogeneous or smooth regions, which is essentially complementary to edge finding, was thought to have the potential of isolating image regions that were the

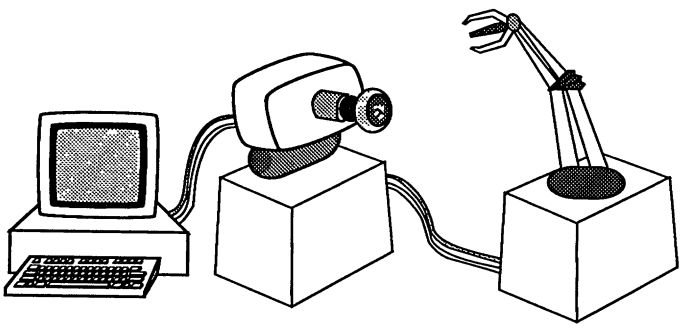


Fig. 2. Hardware for robot vision. The TV camera forms an optical image of the scene on an array of photoreceptors. The sampled image is sequentially scanned and stored in a frame buffer, from which it is read into the computer memory. The processing unit of the computer can randomly access the memory and perform arbitrary computations on the image data. The results of these computations can be used to control the manipulator.

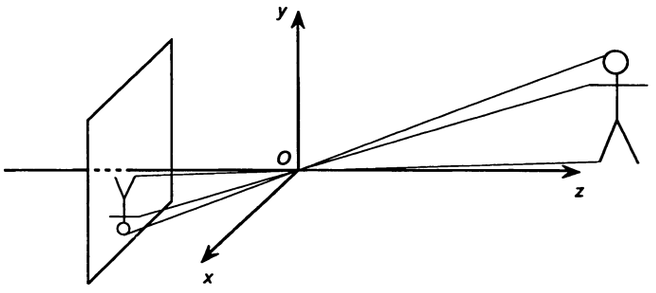


Fig. 3. Geometry of vision. The optical image is formed by perspective projection through a point (a pinhole, or the nodal point of a lens).

images of surface patches with some physical significance. It was soon realized, however, that physically significant parts of a scene cannot be identified solely by analyzing the gray level intensities in the image. By the early 1970s it had become clear that low-level vision could not generally derive useful scene descriptions from a single image, because even seemingly simple problems such as edge detection are in fact very complex.

At that time, which was a period of rapid progress in the development of artificial intelligence, it was suggested that "high-level" knowledge about the scene could be used in conjunction with low-level visual processing to introduce additional constraints. To experiment with such ideas, "complete" vision systems were constructed (*6*) that used information at all levels, including both general knowledge about the imaging process as well as domain-specific information. By and large, however, the performance of these systems was not impressive. Many researchers therefore abandoned the system building approach and concentrated on the study of specific visual abilities, possibly corresponding to identifiable modules in the human visual system (*8*).

1970–1985: Modules and uniqueness. During the 1970s the field of computer vision became more mathematically sophisticated. Marr proposed a paradigm in which a vision system is conceptualized as a collection of individual autonomous components, or modules, each of which performs a different computational task (*5*). The low-level modules operate directly on the image data in order to recover useful 2D descriptions. The middle-level modules use these descriptions to perform 3D recovery; and the high-level modules use the results of recovery to reason about the world.

Low-level vision modules are devoted to extracting "simple" representations of the image intensity array that have some general physical significance. Tasks of particular interest at the low level are image restoration (that is, estimation of the true intensities in a degraded image); edge detection; segmentation into homogeneous regions; and texture representation.

Low-level modules operate on the image intensities and make no use of higher level knowledge about the scene. Some attempts were made to introduce such knowledge into edge detection and segmentation processes by treating them as image labeling processes and making use of local consistency constraints on the labels (*9–10*); but this approach did not provide a sufficiently flexible means of representing and integrating global knowledge.

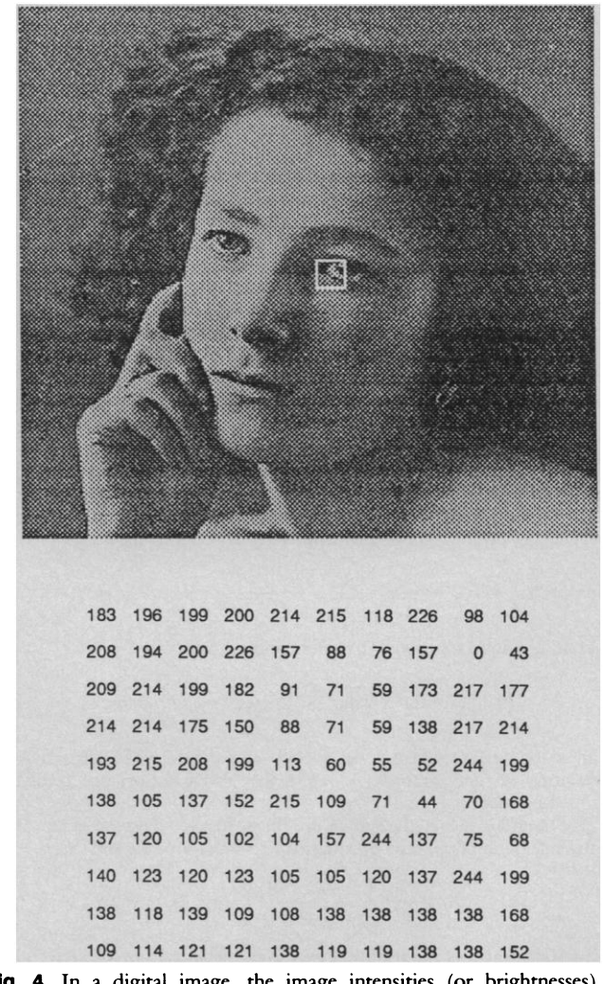
Middle-level modules use the results of the low-level modules as well as the image itself to recover the shapes, colors, spatial locations, and motions of objects in the scene. These modules make use of various cues in the image, such as shading, texture, contours, and motion. During this period many mathematical techniques were developed for describing object geometry (*11*) and computing scene properties on the basis of various types of information present in images (*12–15*).

It turns out, however, that nearly all of these low- or middle-level visual tasks are ill-posed problems (*16–17*); they are underconstrained and so do not have unique solutions. For example, consider the problem of recovering surface orientation from shading [the "shape from shading" problem (*18*)]. If we assume that reflectance is Lambertian (the object reflects light equally in all directions), the intensity I at a point (x,y) of the image is

$$I(x, y) = \rho \frac{1 + p\, p_s + q\, q_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s{}^2 + q_s{}^2}} \qquad (1)$$

where ρ is a constant (the albedo) that depends on the surface material; $(p_s,q_s,-1)$ is the direction of the light source; and $(p, q,-1)$ is the normal at the surface point whose image is (x,y). Thus measuring the intensity at any image point gives us one equation and two unknowns (p,q). Hence we cannot solve the shape from shading problem unless we impose additional constraints on the scene [see, for example, (*19*)]. The same is true for every recovery module.

In general, suppose that we want to recover some quantity ϖ which is a function of position in the image and which satisfies the equation $L(\varpi) = 0$. Because in many cases the equation $L(\varpi) = 0$ is not enough to determine ϖ, we need to make a further assump-



183	196	199	200	214	215	118	226	98	104
208	194	200	226	157	88	76	157	0	43
209	214	199	182	91	71	59	173	217	177
214	214	175	150	88	71	59	138	217	214
193	215	208	199	113	60	55	52	244	199
138	105	137	152	215	109	71	44	70	168
137	120	105	102	104	157	244	137	75	68
140	123	120	123	105	105	120	137	244	199
138	118	139	109	108	138	138	138	138	168
109	114	121	121	138	119	119	138	138	152

Fig. 4. In a digital image, the image intensities (or brightnesses) are discretely sampled, and the sampled values are quantized to a discrete set of values, usually represented by integers. The elements of the resulting array of numbers are called pixels, and their values are called gray levels. In the figure, the array of numbers represents the array of intensities in the boxed portion of Sarah Bernhardt's eye.

tion about the scene. Let this assumption be of the form $S(\varpi) = 0$; this equation constitutes an additional constraint. Such constraints can be of various forms and can be related to various properties of the scene relevant to the quantities being computed. For example, if we need to compute surface shape, $S(\varpi) = 0$ can impose the condition that the surface is smooth, in other words that shape variations are locally small, based on the fact that most surfaces are piecewise smooth. In color processing, $S(\varpi) = 0$ can require that surface reflectance be describable with a small number of basis functions (*20*), based on the small number of retinal pigments. In the processing of general nonrigid motion, $S(\varpi) = 0$ can require that the deviation of the motion from rigidity be small. In segmenting an image into parts, $S(\varpi) = 0$ can require that the segmentation be as simple as possible with respect to some complexity measure (*21*).

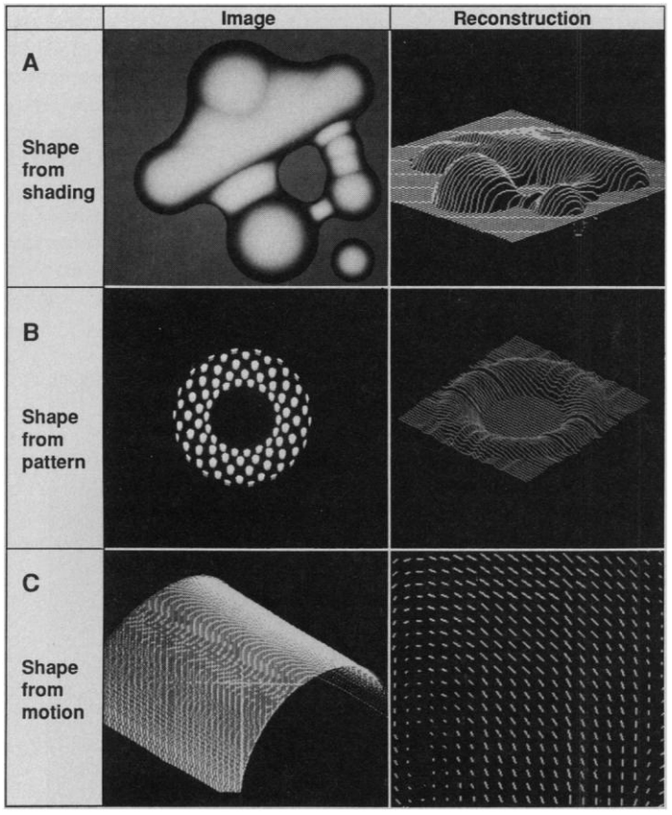


Fig. 5. Recovery of surface geometry from image cues: (**A**) shading, (**B**) pattern, and (**C**) motion. The left column shows the image (or image motion field); the right column shows the reconstructed surface. In (A) the intensity at every image point, assuming a Lambertian reflectance model, is

$$I(x, y) = p\,\frac{pp_s + qq_s + 1}{\sqrt{1 + p^2 + q^2}\,\sqrt{1 + p_s^2 + q_s^2}} \equiv R(p, q).$$

To obtain a unique solution, we minimize the functional

$$\iint_{\text{image}} \left\{ (I - R)^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) \right\} dxdy;$$

this gives the surface which is as smooth as possible while satisfying the constraint $I = R$. The parameter λ weighs the relative importance of the two terms of the functional.

In (B), assuming that all the surface markings ("texels") have the same surface area, the area of an image texel S_I is related to the shape of the surface by

$$S_I = \frac{S_W}{d^2}\,\frac{1 - Ap - Bq}{\sqrt{1 + p^2 + q^2}} = R(p, q),$$

where S_W is the area of the surface texel, d is its distance from the viewer, and (A,B) is the centroid of the image texel. We obtain a unique solution by minimizing the same functional.

In (C), assuming that we know how every point of a rotating object moves on its (orthographic) image, the surface shape (p, q) is related to the local image motion by a quadratic expression of the form $f(p, q) = 0$. As before, we obtain a unique solution by minimizing the same functional.

During the 1980s, Poggio and his colleagues (*16*) suggested that ill-posed visual recovery problems can be solved with the technique of regularization. [For simplicity and for consistency with the mathematical theory of regularization (*22*), we shall call the additional constraint $S(\varpi) = 0$ a "smoothness" condition even though it may not actually express the smoothness of the desired quantity ϖ. In general, $S(\varpi) = 0$ expresses the fact that some function of ϖ should be small.] The solution is then obtained by minimizing a functional of the form $\int L^2(\varpi) + \lambda S^2(\varpi)$. In other words, we find a solution that is as smooth as possible and at the same time satisfies the constraint $L(\varpi) = 0$. The coefficient λ determines the relative importance of smoothness in the solution. Figure 5 shows examples of surface recovery from shading and pattern cues, with the use of various functionals.

The difficulty with regularization is that it tends to smooth over discontinuities [places where the constraint $S(\varpi) = 0$ is violated]; but the visual world is rich in discontinuities. Another problem with regularization is that we need a systematic way of choosing a value for the coefficient λ. If λ is small, the solution involves less smoothing over discontinuities, but it then tends to be more sensitive to noise.

Some current research areas (1985–present): Discontinuities and active vision. In the real world, the function ϖ that we need to recover has discontinuities and discontinuous derivatives. Standard recovery techniques cannot deal fully with this situation.

One approach to dealing with discontinuities is to first segment the image (*23*) into homogeneous regions and then to regularize within each region. However, segmentation is not a solved problem. One of the reasons for attempting to recover the quantity ϖ is to facilitate segmentation.

Another approach is to divide the image into boundary and nonboundary points. Assume there is a known probability that a random point is a boundary point and that at boundary points all values of S are equally likely. At nonboundary points, minimizing $\int [L^2 + \lambda S^2]$ is acceptable except that we do not excessively penalize large S (because large S means a probable boundary point). Thus we can minimize, for example, $\int [L^2 + \lambda g_T(S)]$ where $g_T(S) = \min(S^2, T^2)$. Here T is a threshold depending on the fraction of points that are discontinuities.

This problem has been studied in the case where ϖ has discrete range (for instance, if ϖ is a binary function) and the domain is a discrete lattice (*24*). This approach was later extended to the case where ϖ is real-valued (*25*) and to a continuous domain (*26*). The minimization problem is solved with Monte Carlo techniques or deterministic approximations to them such as the mean-field approximation. In (*27*) the problem was solved by a continuation method called "graduated nonconvexity." The solution is generally not unique. All these methods of finding the solution are either not guaranteed to converge to a global minimum or cannot be known to be reasonably efficient. The assumption that all $S^2 < T^2$ are equally good is also questionable.

Other approaches have been proposed (*28–29*). Some of these approaches first find the boundary points (at which λ can be set equal to 0); others make the amount of smoothing depend on the gradient magnitude (we smooth more where the gradient is small); still others allow some smoothing at boundary points, but only in the gradient direction ("oriented smoothness"). These theories can be augmented to incorporate the assumption that boundaries are smooth except at a few corner points, or other assumptions about boundary shape.

Table 1. How observer activity simplifies recovery problems (*32*).

Problem	Passive observer	Active observer
Shape from shading	Ill-posed problem. Needs to be regularized. Even then, unique solution is not guaranteed because of nonlinearity.	Well posed and stable. Linear equation; unique solution.
Shape from contour	Ill-posed problem. Has not been regularized up to now in the Tichonov sense. Solvable under restrictive assumptions.	Well-posed problem. Unique solution for either monocular or binocular observer.
Shape from texture	Ill-posed problem. Needs some assumption about the texture.	Well-posed problem. No assumption required.
Structure from motion	Well posed but unstable. Nonlinear constraints.	Well posed and stable. Quadratic constraints, simple solution methods, stability.

Another approach (*30*) is based on the observation that the errors and smoothness measures of nearby points are correlated and that we should therefore use terms involving the derivatives of L and S in the minimization problem. This makes it unnecessary to make a rigid binary distinction between boundary and nonboundary points and to recover ϖ while smoothing as little as possible over discontinuities.

Research will continue on the solution of ill-posed problems in which the quantity to be recovered is a function with discontinuous derivatives and better algorithms will be developed. However, it should be pointed out that any recovery technique must be based on assumptions about scene and noise models. The dependence on a noise model leads to serious difficulties, because standard noise models are not adequate to describe images of real scenes. A scene often contains "clutter" which is hard to model. Simple geometric and photometric models for a class of 3D scenes do not give rise to simple models for the 2D images of these scenes. For example, a quadratic Lambertian surface gives rise to a trigonometric shading function in the image. Statistically stationary surface markings in the scene do not necessarily yield stationary intensity fluctuations in the image; conversely, the stationarity of an image property is not necessarily due to the stationarity of the corresponding scene property. In spite of these difficulties, the quest for noise-insensitive recovery algorithms continues and can be expected to lead to a series of increasingly robust methods.

It was observed in the mid-1980s (*31–32*) that many visual recovery problems become easier if the observer is active, that is, it can control its visual apparatus—for example, by making (known) "eye movements." It turns out that most of the low- and middle-level vision problems become much easier to solve (and often even become well-posed) for an active observer. Examples of this phenomenon are shown in Table 1 (*32*). Observer activity is also a central theme in the design of "animate" or "purposive" vision systems, which will be discussed below.

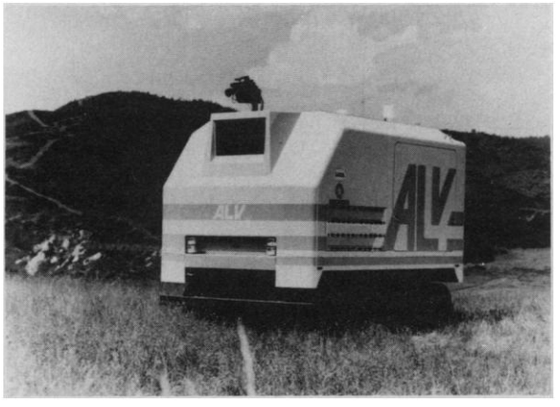


Fig. 6. The Autonomous Land Vehicle, a project sponsored at Martin Marietta Corp. by the Defense Advanced Research Projects Agency (DARPA). The vehicle carried TV cameras and computers, and drove itself along a road network using information about road geometry derived from the TV images.

Recognition and navigation. After the 3D structure of the scene has been recovered from the images, the information can be used in a variety of ways. Objects of given types can be detected and located in the scene by finding parts of the scene that match stored object descriptions. (Of course, in many cases this can be done without first recovering 3D structure.) Knowing the structure of the scene allows a mobile robot (or a robot manipulator) to move around while avoiding obstacles. Object recognition and navigation are the two major areas of application of 3D computer vision. Many systems have been designed for recognizing given classes of objects from their geometric descriptions (*33*). Other systems have demonstrated successful control of robot movements using visual feedback; recent demonstrations involve autonomous outdoor vehicles that can drive on roads under computer control (*34–36*) (Fig. 6).

A New Paradigm: Purposive Vision

In recent years the realization has grown that it is very difficult to create an accurate 3D description of the visible world from images. The recovery paradigm, which regards a vision system as a set of low- and middle-level modules that recover the structure of the scene, and that provide input to high level modules which can then reason about the scene, has not led to the design of successful vision systems, that is, systems that robustly perform recognition or navigation tasks by means of vision.

General 3D scene recovery is a very hard problem. Many recovery problems are inherently unstable [see, for example, (*37–38*)]. In order to correctly recover we may need to formulate new classes of scene and noise models. General recovery would provide a powerful theoretical basis for solving recognition and navigation problems. However, we should not assume that practical results will flow out of successful theories rather than vice-versa. In the past, it has at least as often been the case that successful theories have been constructed on the basis of engineering observations.

For many of the problems we need to solve using vision, complete and accurate recovery of the scene is not necessary. Brooks (*39*) has suggested that it is not necessary to achieve artificial intelligence before we can build successful robots. On the contrary, Brooks claims that we can achieve AI by building robots, starting with simple ones and progressing to more complex ones. He has demonstrated how to build simple robots that have primitive behaviors (see his article in this issue).

Coming back to fundamentals, we should once again ask the basic question: What is vision for? Why does an organism (or a robot) need vision? Obviously, organisms use vision to accomplish various

tasks—for example, to recognize danger, food, and so on. Organisms have goals and purposes, and visual information makes it possible, or easier, to achieve these goals. It has been suggested (*40*) that perhaps vision can be more readily understood in the context of the vision-guided behaviors that the system (the organism or the robot) is engaged in.

This suggestion leads to the important realization that specific vision-guided behaviors may not require a very elaborate representation of the 3D world. If we are looking for an object that can be used for a certain purpose, we may only need to recognize some of its qualitative characteristics; we don't need to know its exact shape. Similarly, if we need to find a path out of a room, we don't need to know the exact shapes of all the pieces of furniture in the room. For many vision-guided behaviors, the visual processing needed is relatively simple; it does not require extensive numerical analysis, but involves only simple qualitative techniques that provide yes/no answers about the scene. For example, an active observer can robustly detect independently moving objects in its vicinity and can estimate their trajectories, using simple analyses of the image motion field (*41*). The results of this partial recovery can be used to control various behaviors, such as dodging or catching an object.

This new paradigm of task-oriented, or purposive, vision, emphasizing the study of specific vision-guided behaviors, will accelerate progress in the field and will lead to systems having robust, reliable performance. At the same time, the paradigm can be used to study the theory of visual perception by developing and analyzing generic vision-based behaviors. However, the paradigm still lacks theoretical foundations, including a formal definition of a visual agent and the dependence of behavior on agent characteristics (size, mobility, and so forth); a formal definition of behavior (as a sequence of perceptual events and actions); and a calculus of behaviors or purposes that can generate new behaviors by combining existing behaviors or by learning and that can provide the basis for controlling the agent; and a corresponding repertoire of visual routines (*42*). The paradigm treats vision as part of a larger system, with increasing emphasis on high-level reasoning about the world, and will require interdisciplinary approaches. The study of vision in organisms and computers continues to be a rich source of interesting research problems.

Summary

Those living organisms that have vision exhibit impressive abilities to interact with their environments. This performance constitutes a challenge to computer (and robot) vision; at the same time, it serves as an existence proof that the goals of computer vision are attainable.

The theoretical foundations of computer vision are not yet fully developed; better models for noisy, cluttered real-world scenes, and better ways of solving ill-posed problems in the presence of discontinuities, are needed. Meanwhile, more attention should be paid to tasks that require only partial descriptions of the scene, because such tasks tend to be better-posed and less computationally costly. Organisms seem to require only partial scene descriptions (of various types) in order to perform visually guided behaviors; similarly, the ability to construct appropriate partial scene descriptions from images may be all that a computer or robot vision system needs to function successfully in its environment.

Computer vision techniques have many practical applications in such domains as document processing, industrial inspection, medical imaging, remote sensing, reconnaissance, and robot guidance. There have been successes in many of these domains, but many tasks are still beyond our current capabilities. These potential applications provide major incentives for continued research.

REFERENCES AND NOTES

1. J. P. Frisby, *Seeing: Illusion, Brain and Mind* (Oxford Univ. Press, New York, 1979).
2. D. H. Ballard and C. M. Brown, *Computer Vision* (Prentice Hall, Englewood Cliffs, NJ, 1982).
3. B. K. P. Horn, *Robot Vision* (McGraw-Hill, New York, 1986).
4. A. Rosenfeld and A. C. Kak, *Digital Picture Processing* (Academic Press, New York, ed. 2, 1982).
5. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (Freeman, San Francisco, 1982).
6. J. K. Tsotsos, in *Encyclopedia of Artificial Intelligence*, S. Shapiro, Ed. (Wiley, New York, 1987), pp. 389–409.
7. M. Brady, *ACM Comput. Surv.* **14**, 3 (1982).
8. The existence of such modules is supported by the study of patients with visual disabilities resulting from brain lesions (*43*) and by psychophysical experiments in which a particular source of information is isolated. Notable examples are Land's demonstrations on the computation of lightness (*44*) and Julesz's demonstration of stereoscopic fusion without monocular cues (*45*).
9. L. S. Davis and A. Rosenfeld, *Artif. Intell.* **17**, 245 (1981).
10. O. D. Faugeras and M. Berthod, *IEEE Trans. PAMI* **3**, 412 (1981).
11. J. J. Koenderink, *Solid Shape* (MIT Press, Cambridge, MA, 1990).
12. S. Ullman, *The Interpretation of Visual Motion* (MIT Press, Cambridge, MA, 1979).
13. T. S. Huang, Ed., *Image Sequence Analysis* (Springer, Berlin, 1981).
14. J. Aloimonos and C. M. Brown, in *Advances in Computer Vision*, C. M. Brown, Ed. (Erlbaum, Hillsdale, NJ, 1987), vol. 1, pp. 115–163.
15. T. Poggio, E. B. Gamble, J. J. Little, *Science* **242**, 436 (1988).
16. T. Poggio, V. Torre, C. Koch, *Nature* **317**, 314 (1985).
17. M. Bertero, T. A. Poggio, V. Torre, *Proc. IEEE* **76**, 869 (1988).
18. B. K. P. Horn and M. J. Brooks, Eds., *Shape from Shading* (MIT Press, Cambridge, MA, 1989).
19. R. T. Frankot and R. Chellappa, *IEEE Trans. PAMI* **10**, 439 (1988).
20. A. C. Hurlbert and T. A. Poggio, *Science* **239**, 482 (1988).
21. Y. G. Leclerc, *Intl. J. Comput. Vision* **3**, 73 (1989).
22. A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-Posed Problems* (Winston and Wiley, Washington, DC, 1977).
23. R. M. Haralick and L. G. Shapiro, *Comput. Vision, Graphics, Image Process.* **29**, 100 (1985).
24. S. Geman and D. Geman, *IEEE Trans. PAMI* **6**, 721 (1984).
25. J. L. Marroquin, S. Mitter, T. Poggio, *J. Am. Stat. Assoc.* **82**, 76 (1987).
26. D. Mumford and M. Shah, in *Image Understanding 1989–1990*, S. Ullman and W. Richards, Eds. (Ablex, Norwood, NJ, 1990), pp. 19–43.
27. A. Blake and A. Zisserman, *Visual Reconstruction* (MIT Press, Cambridge, MA, 1987).
28. H. H. Nagel and W. Enkelmann, *IEEE Trans. PAMI* **8**, 565 (1986).
29. D. Lee and T. Pavlidis, *ibid.* **10**, 822 (1988).
30. J. Aloimonos and D. Shulman, *Integration of Visual Modules: An Extension of the Marr Paradigm* (Academic Press, Boston, 1989).
31. R. Bajcsy, in *Proceedings of the Third Workshop on Computer Vision: Representation and Control* (IEEE Computer Society Press, Washington, DC, 1985), pp. 55–59.
32. J. Aloimonos, I. Weiss, A. Bandopadhay, *Intl. J. Comput. Vision* **2**, 333 (1988).
33. T. O. Binford, *Intl. J. Robotics Res.* **1**, 18 (1982).
34. A. M. Waxman *et al.*, *IEEE J. Robotics Autom.* **3**, 249 (1987).
35. C. Thorpe, M. H. Hebert, T. Kanade, S. A. Shafer, *IEEE Trans. PAMI* **10**, 362 (1988).
36. E. D. Dickmanns and V. Graefe, *Mach. Vision Appl.* **1**, 241 (1988).
37. A. Verri and T. Poggio, in *Proceedings of the 1st International Conference on Computer Vision* (IEEE Computer Society Press, Washington, DC, 1987), pp. 171–180.
38. M. E. Spetsakis and J. Aloimonos, in *Proceedings of the DARPA Image Understanding Workshop* (Morgan Kaufmann, San Mateo, CA, 1990), pp. 271–283.
39. R. A. Brooks, *Achieving Artificial Intelligence Through Building Robots*, AI Memo 899, Artificial Intelligence Laboratory (Massachusetts Institute of Technology, Cambridge, MA, 1986).
40. D. H. Ballard, *Artif. Intell.* **48**, 57 (1991).
41. J. Aloimonos, in *Proceedings of the DARPA Image Understanding Workshop* (Morgan Kaufmann, San Mateo, CA, 1990), pp. 816–828.
42. S. Ullman, *Cognition* **18**, 97 (1984).
43. M. J. Farah, *Visual Agnosia—Disorders of Object Recognition and What They Tell Us About Normal Vision* (MIT Press, Cambridge, MA, 1990).
44. E. H. Land and J. J. McCann, *J. Opt. Soc. Am.* **61**, 1 (1971).
45. B. Julesz, *Foundations of Cyclopean Perception* (University of Chicago Press, Chicago, 1971).

46. This work was funded in part by DARPA (ARPA order no. 6989, through Contract DACA76-89-C-0019 with the U.S. Army Engineer Topographic Laboratories), NSF (under a Presidential Young Investigator Award, grant IRI-90-57934), Alliant Techsystems, Inc. and Texas Instruments, Inc. The authors thank B. Burnett and J. Perrone for their expert help in preparing this paper.