Computer Vision: Attitudes, Barriers, Counseling

Rama Chellappa

Azriel Rosenfeld

Center for Automation Research
University of Maryland
College Park, Maryland 20742-3411 USA

Abstract

The goal of computer vision is to derive descriptive information about a scene by computer analysis of images of the scene. Vision algorithms can serve as computational models for biological visual processes, and they also have many practical uses; but this paper treats computer vision as a subject in its own right. Vision problems are often ill-defined, ill-posed, or computationally intractable; nevertheless, successes have been achieved in many specific areas. We argue that by limiting the domain of application, carefully choosing the task, using redundant data (multi-sensor, multi-frame), and applying adequate computing power, useful solutions to many vision problems can be obtained. Methods of designing such solutions are the subject of the emerging discipline of Vision Engineering. *With projected advances in sensor and computing technologies, the domains of applicability and ranges of problems that can be solved will gradually expand.*

1 Introduction

The general goal of computer vision is to derive information about a scene by computer analysis of images of that scene. Images can be obtained by various types of sensors; the most common kind are optical images obtained by a (black-and-white) TV camera. An image is input to a digital computer by sampling its brightness at a regularly spaced grid of points; the resulting array of sampled values is called a digital image, the elements of the array are called pixels (short for "picture elements"), and their values are called gray levels. Given one or more digital images obtained from a scene, a computer vision system attempts to (partially) describe the scene as consisting of surfaces or objects; this class of tasks will be discussed further in Section 2.

Animals and humans have impressive abilities to successfully interact with their environments—navigate over and around surfaces, recognize objects, etc.—using vision. This performance constitutes a challenge to computer vision; at the same time, it serves as an existence proof that the goals of computer vision are attainable. Conversely, the algorithms used by computer vision systems to derive information about a scene from images can be regarded as possible computational models for the processes employed by biological visual systems. However, constructing such models is not the primary goal of computer vision; it is concerned only with the correctness of its scene description algorithms, not with whether they resemble biological visual processes.

Computer vision techniques have many practical uses for analyzing images. Areas of application include document processing (e.g., character recognition), industrial inspection, medical image analysis, remote sensing, target recognition, and robot guidance. There have been successful applications in all of these areas, but many tasks are beyond current capabilities (e.g., reading unconstrained handwriting). These potential applications provide major incentives for continued research in computer vision. However, successful performance of specific tasks on the basis of image data is not the primary goal of computer vision; such performance is often possible even without obtaining a correct description of the scene.

Viewed as a subject in its own right, the goal of computer vision is to derive correct (partial) descriptions of a scene, given one or more images of that scene. Computer vision can thus be regarded as the inverse of computer graphics, in which the goal is to generate (realistic) images of a scene, given a description of the scene. The computer vision goal is more difficult, since it involves the solution of inverse problems that are highly underconstrained ("ill-posed").

*The support of the Defense Advanced REsearch Projects Agency, through the U.S. Army Topographic Engineering Center under Contract DACA76-89-C-0019, is gratefully acknowledged, as is the help of Janice Perrone and Sandy German in preparing this paper.

A more serious difficulty is that the problems may not even be well defined, because many classes of real-world scenes are not mathematically definable. Finally, even well-posed, well-defined vision problems may be computationally intractable. These sources of difficulty will be discussed in Section 3.

In spite of these difficulties, vision systems have achieved successes in many domains. The chances of success are greatly increased by limiting the domain of application, simplifying the task to be performed, increasing the amount of image data used, and providing adequate computing power. These principles can be stated concisely as: *Define your domain; pick your problem; improve your input*; and *take your time.* They will be illustrated in Section 4.

Following these principles in attempting to solve vision problems provides a foundation for a discipline which we may call *Vision Engineering*, as discussed in Section 5.

2 Vision tasks

If a scene could be completely arbitrary, not very much could be inferred about it by analyzing images. The gray levels of the pixels in an image measure the amounts of light received by the sensor from various directions. Any such set of brightness measurements could arise in infinitely many different ways as a result of light emitted by a set of light sources, transmitted through a sequence of transparent media, and reflected from a sequence of surfaces.

Computer vision becomes feasible only if restrictions are imposed on the class of possible scenes. The central problem of computer vision can thus be reformulated as follows: given a set of constraints on the allowable scenes, and given a set of images obtained from a scene that satisfies these constraints, derive a description of that scene. It should be pointed out that unless the given constraints are very strong, or the given set of images is large, the scene will not be uniquely determined; the images only provide further constraints on the subclass of scenes that could have given rise to them, so that only partial descriptions of the scene are possible.

Computer vision tasks vary widely in difficulty, depending on the nature of the constraints that are imposed on the class of allowable scenes and on the nature of the partial descriptions that are desired. The constraints can vary greatly in specificity. At one extreme, they may be of a general nature—for example, that the visible surfaces in the scene are all of some "simple" type (e.g., quadric surfaces with Lambertian reflectivities). [Constraints on the illumination should also be specified—for example, that it consists of a single distant light source. Note that the surfaces may be "simple" in a stochastic rather than a deterministic sense; for example, they may be fractal surfaces of given types, or they may be smooth surfaces (e.g., quadric) with spatially stationary variations in reflectivity (i.e., uniformly textured surfaces).] At the other extreme, the constraints may be quite specialized—for example, that the scene contains only objects having given geometric ("CAD") descriptions and given optical surface characteristics. Similarly, the desired scene descriptions can vary greatly in completeness. "Recovery" tasks call for descriptions that are complete as possible, but "recognition" and "navigation" tasks usually require only partial descriptions—for example, identification and location of objects or surfaces of specific types if they are present in the scene.

In its earliest years (beginning in the mid-1950s), computer vision research was concerned primarily with recognition tasks, and dealt almost entirely with single images of (essentially) two-dimensional scenes: documents, photomicrographs (which show thin "slices" of the subject, because the depth of field of a microscope image is very limited), or high-altitude views of the earth's surface (which can be regarded as essentially flat when seen from sufficiently far away). The mid-1960's saw the beginnings of research on robot vision; since a robot must deal with solid objects at close-by distances, the three-dimensional nature of the scene cannot be ignored. Research on recovery tasks began in the early 1970's, initially considering only single images of a static scene, but by the mid-1970's beginning to deal with time sequences of images (of a possibly time-varying scene) obtained by a moving sensor.

By definition, recovery tasks require correct descriptions of the scene; but recognition and navigation tasks can often be performed successfully without completely describing even the relevant parts of the scene. For example, obstacles can often be detected, or object types identified, without fully determining their geometries.

Thirty-five years of research have produced theoretical solutions to many computer vision problems; but many of these solutions are based, explicitly or tacitly, on unrealistic assumptions about the class of allowable scenes, and as a result, they often perform unsatisfactorily when applied to real-world images. As we shall see in the next section, even for static, two-dimensional scenes, many vision problems are ill-

posed, ill-defined, or computationally intractable.

3 Sources of difficulty

3.1 Ill-posedness

As already mentioned, the gray levels of the pixels in an image represent the amounts of light received by the sensor from various directions. If the scene does not contain transparent objects (other than air, which we will assume to be clear), the light contributing to a given pixel usually comes from a small surface patch in the scene (on the first surface intersected by a line drawn from the sensor in the given direction). This surface patch is illuminated by light sources, as well as by light reflected from other patches. Some fraction of this illumination is reflected toward the sensor and contributes to the pixel; in general, this fraction depends on the orientation of the surface patch relative to the direction(s) of illumination and the direction of the sensor, as well as on the reflectivity of the patch. In short, the gray level of a pixel is the resultant of the illumination, orientation, and reflectivity of a surface patch. If all these quantities are unknown, it is not possible to recover them from the image. Only under limited conditions of smoothly curved Lambertian surfaces with constant albedo can one recover estimates of illuminant direction surface albedo and shape from a single image [19].

This example is a very simple illustration of the fact that most vision problems are "ill-posed", i.e., underconstrained; they do not have unique solutions. Even scenes that satisfy constraints usually have more degrees of freedom than the images to which they give rise; thus even when we are given a set of images of a scene, the scene is usually not uniquely determined.

In applied mathematics, a common approach to solving ill-posed problems is to convert them into well-posed problems by imposing additional constraints [17]. A standard method of doing this, known as regularization, makes use of smoothness constraints; it finds the solution that minimizes some measure of nonsmoothness (usually defined by a combination of derivatives). Regularization methods were introduced into computer vision in the mid-1980's, and have been applied to many vision problems [12]. Evidently, however, solutions found by regularization often do not represent the actual scene [1]; for example, the actual scene may be piecewise smooth, but may also have discontinuities, and a regularized solution tends to smooth over these discontinuities. To handle this problem, more general approaches have been proposed which allow discontinuities [16], but which minimize the complexity of these discontinuities—e.g., minimize the total length and total absolute curvature of the borders between smooth regions. In effect, these approaches [8] find solutions that have minimum-length descriptions (since the borders can be described by encoding them using chain codes). However, the actual scene is not necessarily the same as the scene (consistent with the images) that has the simplest description. Evidently, not all scenes of a given class are equally likely; but the likelihood of a scene depends on the physical processes that give rise to the class of scenes, not on the simplicity of its description, and certainly not on the simplicity of a description of its image.

3.2 Ill-definedness

It is often assumed in formulating vision problems that the class of allowable scenes is "piecewise simple"—e.g., that the visible surfaces are all smooth (e.g., planar or quadric) and Lambertian. This type of assumption seems at first glance to strongly constrain the class of possible scenes (and images), but in fact, the class of images is not constrained at all unless a lower bound is specified on the sizes of the "pieces". If the pieces can be arbitrarily small, each pixel in an image can represent a different piece (or even parts of several pieces), so that the image can be completely arbitrary. For a two-dimensional scene, it suffices to specify a lower bound on the piece sizes; but for a three-dimensional scene, even this does not guarantee a lower bound on the sizes of the image regions that represent the pieces of surface; occlusions and nearly-grazing viewing angles can still give rise to arbitrarily small or arbitrarily thin regions in the image.

Lower bounds on piece sizes are important for another very important reason: they make it easier to distinguish between the ideal scene and various types of "noise". In the real world, piecewise simple scenes are an idealization; actual surfaces are not perfectly planar or quadric or perfectly Lambertian, but have fluctuating geometries or reflectivities. [Note that these fluctuations are in the scene itself; in addition, the brightness measurements made by the sensor are noisy, and the digitization process also introduces noise.] If the fluctuations are small relative to the piece sizes, it will usually be possible to avoid confusing them with "real" pieces. [Similarly, the noisy brightness measurements—assuming that they affect the pixels independently—yield pixel-size fluctuations, and digitization noise is also of at most pixel size; hence these types of noise too should usually not

be confused with the pieces.] Of course, even if we can avoid confusing noise fluctuations with real scene pieces, their presence can still interfere with correct estimation of the geometries and photometries of the pieces.

Most analyses of vision problems (e.g., for piecewise simple ideal scenes) do not attempt to formulate realistic models for the "noise" in the scene; they usually assume that the noise in the image (which is the net result of the scene noise, the sensor noise, and the digitization noise) is Gaussian and affects each pixel independently. Examination of images of most types of real scenes shows that this is not a realistic assumption; thus the applicability of the resulting analyses to real-world images is questionable.

The problem of ill-definedness becomes even more serious if one attempts to deal with scenes containing classes of objects that do not have simple mathematical definitions—for example, dogs, bushes, chairs, alphanumeric characters, ... Recognition of such objects is not a well-defined computer vision task, even though humans can recognize them very reliably.

3.3 Intractability

Even well-defined vision problems are not always easy to solve; in fact, they may be computationally intractable [7, 5]. An image can be partitioned in combinatorially many ways into regions that could correspond to simple surfaces in the scene; finding the correct (i.e., the most likely) partition may thus involve combinatorial search. For example, even for scenes consisting of polyhedral objects, the problem of deciding whether a set of straight edges in an image could represent such a scene is NP-complete. Even identifying a subset of image features that represent a single object of a given type is exponential in the complexity of the object, if more than one object can be present in the scene, or if the features can be due to noise.

Parallel processing (e.g. [13]) is widely used to speed up computer vision computations; it is also used very extensively and successfully in biological visual systems. Very efficient speedup can be achieved through parallelism in the early stages of the vision process, which involve simple operations on the image(s); but little is known about how to efficiently speed up the later, potentially combinatorial stages. Practical vision systems must operate in "real time" using limited computational resources; as a result, they are usually forced to use suboptimal techniques, so that there is no guarantee of correct performance.

In principle, the computations performed by a vision system should be chosen to yield maximal expected gain of information about the scene at minimal expected computational cost. Unfortunately, even for well-defined vision tasks, it is not easy to estimate the expected gain and cost. Vision systems therefore usually perform standard types of computations that are not necessarily optimal for the given scene domain or vision task; this results in both inefficiency and poor performance.

4 Recipes for success

4.1 Define your domain

Well-defined vision problems should involve classes of scenes in which both the ideal scene and the noise can be mathematically (and probabilistically) characterized. For example, in scenes that contain only known types of man-made objects, the allowable geometric and optical characteristics of the visible surfaces can be known to any needed degree of accuracy. If the objects are "clean", and the characteristics of the sensor are known, the noise in the images can also be described very accurately. In such situations, the scene descriptions that are consistent with the images are generally less ambiguous (so that the problem of determining these descriptions is relatively well-posed) because of the relatively specialized nature of the class of allowable scenes. If, in addition, the number of objects that can be present is limited, the complexity of the scene description task and the computational cost of recognizing the objects are greatly reduced. For example, it has been shown [5] that when all the features in the image can be assumed to arise from a single object, the expected search cost to recognize the object is quadratic in the number of features, and the number of possible interpretations drops rapidly to one as the number of features extracted from the image increases. The number of interpretations and the search cost are much higher when the scene is cluttered, so that the object of interest may be occluded and a significant part of the data may come from other objects in the scene.

4.2 Pick your problem

Even for specialized scene domains, deriving complete scene descriptions from images—the general recovery problem—can still be a very difficult task. However, there is no reason to insist on unique solutions to vision problems. The images (further) constrain the class of possible scenes; the task of the vision system is to determine these constraints. This yields

a partial description of the scene, and for some purposes this description may be sufficient. In fact, in many situations only a partial description of the scene is needed, and such descriptions can often be derived inexpensively and reliably. A partial description may require only the detection of a specific type of object or surface, if it is present, or it may require only partial ("qualitative") characterizations of the objects that are present (e.g., are their surfaces planar or curved).

Two illustrations of the value of partial descriptions are:

a) An autonomous vehicle can rapidly and accurately follow the markers on a road; it need not analyze the entire road scene, but need only detect and track the marker edges [?, 4]. By using additional domain-specific knowledge about the types of vehicles, their possible motions, etc., significant improvements in 3-D object and motion estimation have been reported in [14].

b) An active observer, by shifting its line of sight so that the focus of expansion due to its motion occupies a sequence of positions, can robustly detect independent motion anywhere in the region surrounded by these foci [15]. In this region, independent motion is indicated by the sign of the normal flow being opposite to that of the expansion.

4.3 Improve your inputs

Vision tasks that are very difficult to perform when given only a single image of the scene generally become much easier when additional images are available. These images could come from different sensors (e.g., we can use optical sensors that detect energy in different spectral bands; we can use imaging sensors of other types such as microwave or thermal infrared; or we can use range sensors that directly measure the distances to the visible surface points in the scene). Alternatively, we can use more than one sensor of the same type—for example, stereo vision systems use two or more cameras. Even if we use only a single sensor, we can adjust its parameters—for example, its position, orientation, focal length, etc.—to obtain multiple images; control of sensor parameters in a vision system is known as *active vision* [2]. It has been shown that by using the active vision approach, ill-posed vision problems can become well-posed, and their solutions can be greatly simplified. These improvements are all at the sensor level; one can also consider improving the inputs to the higher levels of the vision process by extracting multiple types of features from the image data using different types of operators (e.g. several edge detectors).

This strategy leads to a situation where "less is required from more", i.e. where it is easier to derive the desired results if more input information is available, unlike the traditional situation where "more is required from less". Animals and humans integrate different types of sensory data, and control their sensory apparatus, to obtain improved or additional information (e.g., tracking, fixation). Obtaining additional constraints on the scene by increasing the amount of image data is evidently a sounder strategy than making assumptions about the scene (smoothness, simplicity, etc.) that have no physical justification.

Many successful computer vision systems have made effective use of redundant input data. In the following paragraphs we give three examples:

a) In [10], thermal (8.5μ-12.5μ) and visual imagery are combined to identify objects or regions such as vehicles, buildings, areas of vegetation and roads. The visual image is used to estimate the surface orientation of the object. Using the surface orientation and other collateral information such as the ambient temperature, wind speed, and the date and time of image acquisition, an estimate of the thermal capacitance of the object is derived. This information, in conjunction with the surface reflectivity of the object (derived from the visual image) and the average object temperature (derived from the thermal image), is used in a rule based system to identify the types of objects mentioned above.

b) Photometric stereo [18] is an excellent example of using more inputs to resolve the inherent ambiguities in recovering shape from shading using a single image irradiance equation. In this scheme, the viewing direction is held constant, but multiple images are obtained by changing the direction of illumination. One then generates as many coupled irradiance equations as there are illumination directions. By solving these equations, robust estimates of the surface orientation can be obtained. Photometric stereo can be very useful in industrial applications where the incident illumination can be controlled.

c) Stereo matching is the process of fusing two images taken from different viewpoints to recover depth information in the scene. The process involves identifying corresponding points or regions

in two views and using their relative displacements together with camera geometry to estimate their depths. If the baseline (the distance between the two cameras) is large, accurate depth estimates can be obtained, but at considerable added computational cost in the feature matching process. With a short baseline the cost of matching is less, but the depth resolution is low. In [11] a method is described that uses multiple stereo pairs with different baselines generated by lateral displacements of a camera. A practical system with seven cameras has been developed. This is a very good example in which, by using more inputs, the complexity of the algorithms is considerably reduced, while at the same time the results are improved.

4.4 Take your time

Since the early days of computer vision, the power of general purpose computational resources has improved by many orders of magnitude. This, combined with special purpose parallel hardware, both analog [9] and digital (VLSI), has greatly expanded the range of tractable vision tasks. The availability of increasingly powerful computing resources allows the vision system designer much greater freedom to adopt an attitude of "take your time" in vision algorithms, as well as freedom to use redundant input data. With no end in sight as regards expected improvements in computing power, the required time to solve given vision problems will continue to decrease. Conversely, it will become possible to solve problems of increased complexity and problems that have wider domains of applicability.

5 Vision engineering

Perception engineering has been defined [6] as the study of techniques common to different sensor-understanding applications, including techniques for sensing and for the interpretation of sensory data, and how to integrate these techniques into different applications. He pointed out the existence of a serious communication gap between researchers and practitioners in the area of machine perception, and proposed establishing the field of perception engineering to bridge this gap. However, he did not formulate any principles that could serve as guidelines for the design of successful machine perception systems.

We believe that the principles discussed in Section 4 can serve as foundations for an approach to computer vision that we shall refer to as *Vision Engineering*. The central task of vision engineering is to make vision problems tractable by applying the four principles: carefully characterizing the domain, choosing the tasks to be performed (breaking a given problem up into subtasks, if necessary), and providing adequate input data and adequate computational resources. We feel that these principles and their extensions will find increasing application in the design and construction of vision systems over the coming years.

References

[1] J. Aloimonos and D. Shulman. *Integration of Visual Modules: An Extension of the Marr Paradigm*, Academic Press, Boston, MA, 1989.

[2] J. Aloimonos, I. Weiss and A. Bandophadhay. "Active vision", *Intl. J. Computer Vision* **1**, 333–356, 1987.

[3] E.D. Dickmanns and V. Graefe. "Dynamic monocular machine vision", *Machine Vision and Applications* **1**, 223–240, 1988.

[4] E.D. Dickmanns and V. Graefe. "Applications of dynamic monocular machine vision", *Machine Vision and Applications* **1**, 241–261, 1988.

[5] W.E.L. Grimson. *Object Recognition by Computer*, MIT Press, Cambridge, MA, Chapter 10, 1990.

[6] R. Jain. "Perception engineering", *Machine Vision and Applications* **1**, 73–74, 1988.

[7] L.M. Kirousis and C.H. Papadimitriou. "The complexity of recognizing polyhedral scenes", *J. Comp. Syst. Scis.* **37**, 14–38, 1988.

[8] Y.C. Leclerc. "Constructing simple stable descriptions for image partitioning", *Intl. J. Computer Vision* **3**, 73–102, 1989.

[9] C. Mead. *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, MA, 1989.

[10] N. Nandhakumar and J.K. Aggarwal. "Integrated analysis of thermal and visual images for scene interpretations", *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-10**, 469–481, 1988.

[11] M. Okutomi and T. Kanade. "A multiple-baseline stereo", Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Miami, 63–69, 1991.

[12] T. Poggio, V. Torre, and C. Koch. "Computational vision and regularization theory", *Nature* **317**, 214–319, 1985.

[13] V.K. Prasanna Kumar. *Parallel Architectures and Algorithms for Image Understanding*, Academic Press, New York, 1991.

[14] J. Schick and E.D. Dickmanns. "Simultaneous estimation of 3D shape and motion of objects by computer vision", IEEE Workshop on Visual Motion, Princeton, 256–261, 1991.

[15] R. Sharma and J. Aloimonos. "Robust detection of independent motion: An active and purposive solution", Center for Automation Research Technical Report CAR-TR-534, University of Maryland, College Park.

[16] D. Terzopoulos. "Regularization of inverse visual problems involving discontinuities", *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-8**, 413–426, 1986.

[17] A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-Posed Problems*, Winston, New York, 1977.

[18] R.J. Woodham. "Photometric method for determining surface orientation from multiple images", in B.K.P. Horn and M.J. Brooks (eds.), *Shape from Shading*, MIT Press, Cambridge, MA, 1989.

[19] Q. Zheng and R. Chellappa. "Estimation of illuminant direction, albedo and shape from shading", *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-13**, 680–702, 1991.