

WORKING PAPER 57

A HYPOTHESIS-FRAME SYSTEM FOR RECOGNITION PROBLEMS

by

SCOTT FAHLMAN

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

December, 1973

Abstract

This paper proposes a new approach to a broad class of recognition problems ranging from medical diagnosis to vision. The features of this approach include a top-down hypothesize-and-test style and the use of a great deal of high-level knowledge about the subject. This knowledge is packaged into small groups of related facts and procedures called frames.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0005.

Working Papers are informal papers intended for internal use.

A Hypothesis-Frame System for Recognition Problems

Ph.D. Thesis Proposal - Scott Fahlman

<<The ideas expressed in this paper are still in their formative stages. I would welcome any thoughts readers might have about them, about the proposed research, or about this presentation.>>

A pattern of light and darkness strikes a person's retina and he sees a familiar face, a cow, or a cloud shaped like Texas; vibrations strike his eardrum and he hears words and sentences, a Russian accent, or the call of the Arctic Loon; a chess master looks at pieces on a board and sees a vulnerable king, an impenetrable queen-side, a likely pawn for promotion; a doctor listens to symptoms, reads lab reports, thumps the patient, and arrives at his diagnosis. These examples differ in the extent to which they are conscious acts, but they all are acts of recognition - the process of fitting some specific set of observable data (called the sample) into some loosely defined conceptual category, chosen from a large number of such categories. Because it lies at the heart of so many of our mental processes, recognition has fascinated philosophers and psychologists for centuries. Gestalt psychology, in particular, owes its existence to the cloud of mystery that surrounds the seemingly instantaneous nature of the recognition act. This thesis will be an attempt to dispel some of that mystery.

Most current computer recognition programs employ a bottom-up

deductive approach: The raw input data is carefully massaged into increasingly abstract and concise forms until, hopefully, it can be identified with a single descriptive category by something akin to a simple template match. I propose to investigate and develop a different paradigm for recognition, based upon the use of hypotheses and highly-structured global knowledge to provide the system with a strong top-down flavor. After a brief and very superficial analysis of the incoming data, the proposed system would jump to some hypothetical identification, attempt to verify this hypothesis and, if not satisfied, analyze the discrepancies for clues indicating a better hypothesis or a way to refine the current one. While investigating a hypothesis, the system will be operating within a recognition frame, a local data environment containing whatever information may be relevant to the verification, as well as a set of suggestion demons which will suggest alternative hypotheses if certain key features appear in the input or the discrepancy analysis. I hope to show that such a top-down approach can be far more efficient than the current approaches and, more important, that the top-down approach can be vastly more flexible and adaptable. Of course, such a system will be only as good as the body of knowledge it contains, but a top-down frame system will at least be capable of holding a large knowledge base and of making effective use of the knowledge in its recognition processes; this is the point upon which I pin my hopes for success. The approach I propose is quite similar to, and largely inspired by, that suggested by Minsky in his forthcoming paper on frames and vision.

It is my intention to concentrate upon the development of this

recognition paradigm itself and upon the data and control structures necessary for its success, not upon any specific domain of recognition problems. This runs counter to the current trend in AI research, which suggests that new paradigms can best be developed within the confines of some specific set of well-chosen problems, but I feel that the more general approach is indicated in this case. A number of researchers - notably the DENDRAL people, Reddy and his associates in speech, Freuder in vision, and Sussman and Pauker in medicine - have used a hypothesize-and-test style in their respective problem domains, but in all of these cases the purely local issues of the field in question have dominated the research, leaving little time for developing really good solutions to the common underlying problems. It is my hope that by avoiding the domain-specific issues and concentrating upon the similar aspects of these diverse fields, I will be able to develop a set of mechanisms that will benefit them all, in much the same way that PLANNER and CONNIVER benefited all of problem-solving. (Of course, the scale of the domain is important: Some domains are so trivial that rigid tree-search, statistical clustering, or bottom-up is entirely satisfactory; others are so amorphous that brute-force search is the only way. I am aiming at the ground between these extremes, where I believe most of the interesting problems live.)

This is not to say that my work will be completely divorced from real-world applications. Any totally abstract approach faces the constant danger of drifting out to sea on a cloud of its own definitions. Worse, the research might be successful, but could be ignored for years

because nobody understands what it is good for. I intend to use not one but several problem domains for guiding the research and for its final presentation in the thesis. The aim will be to elaborate exactly how the new paradigm will operate in each area and what advantages it holds over other methods, while glossing over all of the local details which would cause a serious diversion of effort. For instance, the thesis might describe the way a medical diagnosis system would operate, without going into the specific details of actual diseases and symptoms, or it might describe a visual recognition system for animals or faces in terms of some very ad hoc blob-description language, thus avoiding the quagmire of low-level vision. If the new system lives up to my expectations, a few such examples should be sufficient to "sell" it.

The following sections will explain and illustrate more clearly the type of system I have in mind. The details are, of course, highly tentative at this point.

Data Structure

One of the favorite slogans in AI research recently has been "Recognition should use high-level knowledge." The slow progress on this front seems to have two causes: First, it is not at all clear exactly where and how such knowledge can play a role in the usual bottom-up type of program. Second, systems which try to use too many facts usually choke on them unless they are organized in some way, and most of the

current organizational schemes involve very rigid classification systems at great cost to overall flexibility.

I envision a data base in which related sets of facts and demons are grouped into packets, any number of which can be activated or made available for access at once. These packets are very similar to CONNIVER contexts, but they do not share CONNIVER's bias toward linear growth and inclusion; a packet can contain any number of other packets (recursively), in the sense that if the containing packet is activated, the contained packets are activated as well, and any data items in them become available unless they are specifically modified or cancelled. Thus, by activating a few appropriate packets, the system can create a tailor-made execution environment containing only the relevant portion of its global knowledge and an appropriate set of demons. Sometimes, of course, it will have to add specific new packets to the active set in order to deal with some special situation, but this inconvenience will be far less than the burden of constantly tripping over unwanted knowledge or triggering spurious demons. The implementation of this packet system will be discussed below in the section on technical problems.

Hypothesis Verification

In the next sections I will describe how the system selects an initial hypothesis and moves from one hypothesis to another; first, however, let's see how the system goes about testing a hypothesis once it has one. For every recognition category the system knows about, and thus for every hypothesis it can entertain, there is an associated packet of information and demons called the recognition frame for that hypothesis. The demons are mostly for the purpose of changing the hypothesis and will be discussed later; the rest of the information consists of a definition of the category in question and auxiliary information about how to check this definition against the incoming data of the sample.

In certain very formal domains the definitions may consist of predicates to be satisfied, but far more often a category will be defined by the use of an exemplar, an archetypical model representing an ideal member of the category. This model will usually be in the form of a Winston-style network of features, and it is used in a very Winston-like way: As each feature of the sample comes in, it is matched against the corresponding part of the exemplar; any discrepancies are noted and evaluated. Each domain will have its own set of features and relations for describing the samples and exemplars, but some mechanisms of general usefulness will have to be developed. There will be various kinds of set inclusion, logical and evidentiary relationships among the features, procedures or recipes for determining the closeness of a match, demons for changing one representation into another, and so on. Sometimes all

or part of an exemplar will take the form of a scenario, a time sequence of events or actions useful for expressing such things as the normal course of a disease or the characteristic gait of a komodo dragon. Good notations for such things must be worked out. My current image is that most of the information will be declarative rather than procedural, and will be referenced by a rather simple matcher-interpreter that follows local frame information in figuring out what to do in what order, but this is very tentative.

The system begins the verification process by checking any sample features that it already has on hand - features that arrived in the first wave or were obtained while testing previous hypotheses. Then, if the hypothesis has not already been accepted or rejected, the system begins asking questions to get more information about features of the sample. The nature of these questions will vary according to the problem domain: A doctor program might order some lab tests; a vision program might direct its low-level components to look at some area of the scene more closely. Sometimes a question will recursively start another recognition process: "This might be a cow - see if that part is an udder."

The order in which the questions are asked is determined by auxiliary information in the frame. This information indicates which features are the most critical in the verification at hand, how these priorities might be affected by information already present, and how much each question will cost to answer. As each new feature of the sample is established, its description is added to a special packet of information about the sample, along with some indication of where the information

came from and how reliable it is. This packet can be taken along if the system moves to another hypothesis. Sometimes unsolicited information will be noticed along the way; it, too, is tested and thrown into the pot.

Of course, the system will practically never get a perfect match to any of its ideal exemplars. There will always be discrepancies, and not all discrepancies are created equal. Auxiliary frame information will indicate for each expected type of violation whether it should be considered trivial, serious, or fatal (in the sense that it decisively rules out the current hypothesis). Continuously variable features such as size, body proportions, or blood pressure will have a range of normal variation indicated, along with a mapping from other ranges into seriousness values. Sometimes a feature will provide no real evidence for or against a hypothesis, but can be explained by it; this, too, is noted in the frame. If there are striking or conspicuous features in the sample (antlers, perhaps) that are not mentioned in the current frame, the system will usually consider these to be serious violations; such features are evaluated according to information stored in a packet associated with the feature, since the hypothesis frame clearly cannot mention every feature not present in the exemplar.

Occasionally a feature will have a strong confirming effect: If you see it, you can stop worrying about whether you're in the right place. Usually, though, we will not be so lucky as to have a decisive test. The normal procedure, then, is to gather in sample features until either some satisfaction level is reached and the hypothesis is accepted, or until a

clear violation or the weight of several minor violations sends the system off in search of something better. (My current image of the satisfaction level is as some sort of numerical score, with each matched feature adding a few points and each trivial mismatch removing a few. Perhaps some more complex symbolic scheme will be needed for this, but right now I don't see why.) The satisfaction level can vary considerably, according to the situation: The most cursory glance will convince me that my desk is still in my office, while a unicorn or a thousand dollar bill will rate a very close inspection before being accepted.

Sometimes the sample will appear to fit quite well into some category, but there will be one or two serious violations. In such a case the system will consider possible excuses for the discrepancies: Perhaps the cow is purple because someone has painted it. Perhaps the patient doesn't have the expected high blood pressure because he is taking some drug to suppress it. If a discrepancy can be satisfactorily explained away, the system can accept the hypothesis after all. Of course, if the discrepancies suggest some other hypothesis (see section on hypothesis selection), the system will try that first and resort to excuses only if the new hypothesis is no better. Sometimes two categories will be so close together that they can only be told apart by some special test or by paying particular attention to some otherwise insignificant detail. It is a simple enough matter for both of the frames to include a warning of the similarity and a set of instructions for making the discrimination. In medicine, such testing is called

differential diagnosis.

Note that this use of exemplars gives the system an immense flexibility in dealing with noisy, confused, and unanticipated situations. A cow may formally be a large quadruped, but our system would have little trouble dealing with a three-legged cow amputee, as long as it is a reasonably good cow in most other respects. (A missing leg is easy to explain; an extra one is somewhat more difficult.) If the system is shown something that fits none of its present categories, it can at least indicate what the sample is close to, along with some indication of the major deviations from that category. A visual system organized along these lines might easily come up with "like a person, only 80 feet tall and green" or "a woman from the waist up and a tuna fish from the waist down." Under certain circumstances, such descriptions might serve as the nuclei of new recognition frames representing legitimate, though unnamed, conceptual categories.

Frame Hierarchies

An important feature of recognition frames (and of the recognition categories they represent) is that they can be organized into hierarchies. The system can thus hypothesize at many levels, from the very general to the very specific: An animal of some sort, a medium-sized quadruped, a dog, a collie, Lassie. Each level has its own recognition frame, but the frames of the more specific hypotheses include the information packets of the more general frames above them; thus, if

the system is working under the "dog" frame, the information in the "animal" frame is available as well. A specific frame may, of course, indicate exceptions to the more general information: The "platypus" frame would include the information in "mammal", but it would have to cancel the parts about live birth of young. Often a general frame will use one of the specific cases below it as its exemplar; "mammal" might simply use "dog" or "cow" as its exemplar, rather than trying to come up with some schematic model of an ideal non-specific mammal. In such a case, the only difference between hypothesizing "mammal" and "cow" would be a somewhat greater reluctance to move to another mammal in the latter case; the system would test the same things in either case.

Note that there can be many different hierarchical networks, and that these can overlap and tangle together in interesting ways: A komodo dragon is taxonomically a reptile, but its four-legged shape and its habits are closer to a dog's than to a snake's. How to represent these entanglements and what to do about them are problems that will require some further thought. Some frames are parasitic: Their sole purpose is to attach themselves to other frames and alter the effects of those frames. (Perhaps "viral" would be a better term.) "Statue-of" might attach to a frame like "cow" to wipe out its animal properties of motion and material (beef), while leaving its shape properties intact. "Mythical" could be added to animal to make flying, disappearance, and the speaking of riddles in Latin more plausible, but actual physical presence less so. Complications could be grafted onto a disease using this mechanism. There is nothing to prevent more than one parasite at a

time from attaching to a frame, as long as the parasites aren't hopelessly contradictory; one could, for instance, have a statue of a mythical animal.

Hypothesis Selection

The previous section may have made hypothesis verification sound like a formidable task, but that is because all of the most perverse cases had to be considered; in normal cases the verification is only as bad as the sum of the questions that must be asked, and these are chosen with a great deal of care and high-level guidance, so there is very little wasted effort. The system's overall performance, then, is largely a function of how quickly it can find its way to the right hypothesis. Often, of course, this is no problem at all. In familiar situations, expectations and context information can give the system a very good idea of what to try: The gray blob in the corner of my office is probably my desk; a black blob on a desk top is probably a telephone; a knee-high blob moving down the street next to a person is probably his dog. As Minsky points out, such expectations can be hung in a frame that is activated by the situation - office, street, or whatever. Such mechanisms can carry most of the load most of the time, but they are helpless in the face of surprises; for these, more powerful methods are needed.

The most promising of these methods is the use of suggestion demons, the mutated descendants of Winston's similarity networks. These live in

various hypothesis frames, and their duty is to watch for the arrival of certain highly suggestive features from outside or for the appearance of certain types of discrepancies during the verification process. When a demon sees the datum to which it is attuned, it swoops down, perhaps checks a few other conditions, and either drags the system immediately to a new hypothesis or, if it is less sure of itself, adds a suggestion to a list of hypotheses to be tried if the current one begins to look bad.

(The suggestion list idea has been developed by Freuder, among others.)

It is important to note that the suggestion demons do not have to be particularly reliable; if the system is pulled off to a bad hypothesis, this will be detected almost at once and the system can move on or return to what it was doing with little wasted effort or confusion. The demons, then, should err on the side of over-enthusiasm if they must err at all.

By placing the suggestion demons down in the various hypothesis frames instead of making them global in scope, the system in effect gets some of the demon triggering conditions tested for free. A crow-suggesting demon at the top level would have to look for some conjunction of blackness, feathers, a beak, a certain shape and size, and so on; if hung down in the "bird" frame, it would only have to look for blackness. (It may be useful to defer the firing of such hypothesis-refining demons until the system has become reasonably happy in the parent frame. Many such issues remain to be resolved.) Since firing a demon on a conjunction of conditions is difficult and costly, especially if the conjunction is of the four-out-of-five type, the savings obtained by the use of frame-bound demons can be important, but far more important is the

reduction in confusion and conflict arising from having only a few demons active at any given time. As an added advantage, the set of questions to be asked during frame verification can be augmented to include questions that are likely to turn up good demon-triggering material, especially material that will help to further refine the hypothesis. Under "animal", for instance, the system should be sure to notice the size, body proportions, shape of the head, feet, and tail, body markings, and any unusual protuberances; these are the features that will suggest what kind of animal the sample might be. In effect, the demons and questions together form a sort of search tree, but a very loose and adaptable one that is easy to augment bit by bit.

A few secondary hypothesis selecting methods should probably be mentioned in passing, methods that will be useful when the system is working in an area where it has had little experience, and where the network of suggestion demons has thus not been fully built up. First, the system could ask for help, either from a human or from other robots that have more expertise in the field at hand. Unlike bottom-up systems, the proposed system can handle hints and suggestions in a very smooth and natural way: External suggestions are treated just as though they had come from internal demons. Even if a suggestion is wrong, it can sometimes serve a useful purpose by getting the system to some node of the net from which it can find its own way to the solution. In lieu of this help, the system will have to start crawling down the tree node by node, testing the most probable nodes on a given level until a match is found, then repeating the process at each lower level until a final

identification is made or, at long last, some demon appears with a better idea. This is clearly too inefficient to be used except as a last resort, but it is always nice to have a last resort to fall back upon when all else fails. Once the identification is made, the system can consider the route it has taken and create some demons to make the task easier the next time. Interestingly enough, Sussman claims that novice doctors crawl down the diagnosis tree carefully while the experts jump to hypotheses almost at once (though the hypotheses are usually wrong and will need to be debugged).

Technical Problems

Most of the effort in this research will be devoted to elaborating the above ideas, testing them on various examples, and condensing the results into some coherent and usable form. There are, however, some technical problems that ought to be dealt with if the system is to be implemented in a reasonable way. In general, these will be treated as second-class problems: I will work a bit on them, I hope to solve them, but I will not let them bog me down. In many cases, they are problems that will almost certainly go away as progress is made on the hardware front.

The most obvious of these problems is the implementation of the packet data base. This could be implemented as a slightly modified CONNIVER, but this would be very inefficient (though perhaps good enough for testing purposes). CONNIVER operates by finding all of the matching

items in the data base, then going down the list and throwing out all of those which should not be reported as being present. Obviously, one would prefer a system in which inactive packets and the items in them are completely out of the way, perhaps even out on disk somewhere, and where system overhead is not increased as new data is added. There are some very nice implementations of this using active associative memories, and I have been playing with ways to use hash tables to create the same effect.

The recognition system clearly must be embedded in a control structure that can move easily from one hypothesis to another and back again with little wasted effort; the control structure developed for my master's thesis (BUILD) should be ideal for this. It is at present not clear whether the system will require demons that fire upon complex constellations of conditions or whether a complex priority scheme will be needed to govern demon firing. These problems will have to be dealt with later, after the problem is better understood. Both are related to the packet system, and some of the same techniques may be useful.

A Visual Example

To bring the overall operation of the proposed recognition system into somewhat sharper focus, I offer the following example of a visual recognition. We will follow our robot as it charges into its office to answer the telephone and notices the office-furniture-gray unicorn that has been placed there by a malevolent band of Luddites, trying to confuse

the machine to death. (As every Star Trek fan knows, if you confuse a robot it will smoke from the ears and die.) Thus, we will see how the machine deals with a completely expected recognition (the phone) and a complete surprise (the unicorn). This example is just meant to illustrate the style of approach used; it will be semi-incoherent, will gloss over things shamelessly, and should under no circumstances be considered as a serious attempt to describe how vision should be done.

We will assume that there is a low level vision system that runs semi-autonomously, producing crude descriptions of the scene at hand. These descriptions might be in terms of blobs in an assortment of simple convex shapes and sizes, located in various parts of the picture, connected in various ways, and with pointy or rounded corners. Each blob will have its color and perhaps its texture labeled. In general, then, the initial descriptions would be about like what you would see if you completely blurred your eyes. Of course the system can do a lot better if it wants to: Upon command from above, any portion of the scene can be expanded to "fill the screen" and far more detail can be read out. In this way the system can refine its description of any blob or look for smaller blobs that previously escaped its notice. The expanded descriptions, then, consist of some crude basic shape with successive layers of refinement added or subtracted, but with the crude description showing through most prominently for matching. (This is more or less in the style of Hollerbach.) Separate processes might monitor the scene looking for movement and estimating the distance of the blob objects by stereo or parallax, dividing blobs when necessary. Alternatively, the

system might deal not in blobs but in simply-described 3-D surfaces, probably found by a lower (but interacting) level of hypothesize-and-test. (This idea was suggested by Joel Moses.)

For the moment, at least, I do not share Minsky's view that information in the frames (verification information, though he doesn't draw the distinction clearly) should be stored in essentially the same format as the incoming information - that is, as a two dimensional picture seen from some particular viewpoint, with a variety of views of the same object being stored in different frames. Instead, I favor the use of some three-dimensional model format for the exemplars. When the system wants to check whether some exemplar, viewed from some hypothesized position, matches the scene at hand, the appropriate 2-D view is created by some computer-graphics-like technique. The position hypothesis is then fine-tuned until the match is good enough. This seems to me to be far more flexible than storing a fixed set of views for each exemplar. Note that all of this refers only to verification; the suggestion demons will certainly be looking for characteristic 2-D descriptions of various objects, especially for getting directly to often-used frames.

We join our robot as it bursts through the (hopefully open) door into its office, in response to the ringing phone. As it enters, the robot pulls out the frame for its office, notes its own orientation in that frame, and computes roughly where the major blobs should be in its field of vision. Sure enough, there is the gray desk-blob in the corner, the bright window-blob straight ahead, the pale green wall-areas, the

dark-gray floor. Curiously, there is also a large gray blob off to the right that doesn't seem to match anything in the frame. Well, it's not in the way and the phone is ringing, so the robot saves it for later. (Remember, this whole scenario takes only a second or so.)

Now the robot zeros in on the desk. From the desk model, its position in the room, and the robot's new position (it has been moving toward the desk), the system computes where the edges of the desk top should be, at least if nobody has moved the thing since yesterday. Sure enough, there they are, within an inch or so of where they should be. Guiding itself relative to the desk now, the machine continues its approach, while scanning the desk top for a black, telephone-sized blob. There it is - now which way is it pointing? The telephone frame suggests that the shiny circle of the dial is a good thing to check for or, failing that, the "ears" of the receiver. The dial is easily spotted, and from its position relative to the rest of the blob, the phone's orientation is determined. The phone is imagined in that position, everything fits, and the robot sends its hand out for the receiver.

Now, what was that gray thing I ran past? The initial impression was of a shoulder-high horizontal gray blob, suspended a couple of feet off the floor by some skinny vertical blobs of the same color. We are still in the office frame, and the gray color gets us into the office furniture sub-frame. Down swoops the table-suggesting demon which lives in this frame and watches for large horizontal elevated blobs with skinny legs. (Because the office furniture frame is active, more global demons like "animal" are temporarily de-activated or at least weakened.) The

table frame is activated, looks immediately for a flat smooth horizontal surface, and finds only curved surfaces and fuzzy textures. This is a serious violation, and it weakens the office furniture expectation, allowing more global demons to come in. The fuzzy texture calls down a demon suggesting "animal". The curves call down several demons, one of which is "animal" again. This is enough; without even waiting for the body-and-legs shape demon to add its call for "animal" the system goes off to that frame, carrying along the packet of information derived about this thing so far. Of course, if the unicorn had moved, we might have left "office furniture" and arrived at "animal" much sooner.

As the "animal" frame is entered (actually, the frame in question is probably more like "large quadruped mammal"), the system checks out the information it already knows against the exemplar, in this case a model borrowed from the "cow" frame. The fuzzy texture, the curves, the legs, the body, the size, the proportions all fit reasonably well. (The legs are a bit long. This might call down a weak "horse" demon which might even be accepted, since the choice of "cow" is arbitrary, but let's say this doesn't occur.) Now, the first thing that the animal frame wants to know is what the head looks like, but first it has to decide which end the head is on. It could take a 50/50 chance, but it finds a large bulge on one end of the body, so it sends the low-level routines off to get a better description of that area, possibly informing them that the area in question is probably an animal head.

The description comes in: A fairly long neck, rising at about a 45-degree angle; a slightly tapering rectangular head-blob, down and at

right angles to the neck; pointy, erect ears. A clear horse-head shape, and a demon kicks us into the horse frame. Now, guided by the model, details are sought out and matched up. Big eyes, nostrils, a mane. The legs are right: One of the front ones is a little forward and bent; one of the rear legs is missing, but the system decides that it is hidden behind the other leg. The tail is right. Golden horn in the middle of the forehead . . . OOPS!

Suddenly the system is popped into the "unicorn" frame, actually a parasite that shares the horse model but grafts on two critical differences: the horn and the attachment of the "mythical" parasite to "animal". The "mythical" frame is important, because it can never be happy outside of a fairy tale context; basically, it comes with a built-in serious discrepancy that must be explained away at once. In this case, no such explanation is forthcoming, since the thing is standing right there in front of the robot. Sadly, the robot returns to the horse frame where it came from. Now it must explain away the horn, so it looks carefully for tell-tale signs of artificial attachment. Aha! A rubber band stretching from the horn and running under the animal's chin. Case solved, elapsed time 1 second. (Or 1 minute or 1 day - time will tell.)

Certainly there have been many things to object to in this scenario - if there were not, this would be the thesis instead of the proposal. Hopefully it conveyed the flavor of how the new paradigm might be applied to a field like vision.