Application Of The Schema Mechanism To Learning Visual Tasks

Henry Minsky

December 23, 1992

1 Introduction

This thesis describes the construction of a system which learns to perform a class of useful operations, *visual tasks*, on high-resolution two-dimensional binary images, using a learning algorithm based on Drescher's Schema Mechanism [Dre91]. The system is composed of two major components; a synthetic visual system, and the Schema Mechanism learning system(Section 2). The synthetic visual system provides a rich set of primitive operations, implemented as *vision modules* for detecting classes of *visual artifacts*, which the learning system can manipulate and reason about. The visual system which I propose to implement is based in part on Ullman's ideas about visual routines[Ull84], as well as work by Mahoney on image chunking[Mah87], and Shashua on computation of saliency[Sha88]. The learning system will be responsible for the creation and purposeful manipulation of visual routines to solve visual tasks. It is also designed to discover and construct progessively higher level representations of visual structures, in order to perform progressively more abstract tasks. The synthetic visual system is designed to operate on high-resolution (300-400 dpi) page-sized binary images.

There has been little progress in applying learning to computer vision problems. The main cause of this can be blamed on a lack of a good framework for general purpose learning and action. Currently, computer vision research tends to be done with a standalone mentality; the algorithms and systems are designed to solve a specified task in a self-contained fashion, without worrying about what other agency will be using the output from the system. These systems have built in to them a large set of assumptions, about both the task domain and about how the system will interface to other clients. These sets of assumptions have become inseperable from one another in research areas like object recognition and image segmentation. The result is systems which are *monolithic*; they are structured to solve a specific high-level task, and in a way which is self-contained and opaque to external agencies.

This thesis describes a different approach from the monolithic systems approach to bulding a system which can learn to perform a class of general purpose problems involving computer vision. Several factors guide the design of this system. Foremost, we are designing the visual processing system with a specific client in mind who needs to use its results, namely the Schema learning mechanism. The visual system which we construct consists of a large number of simple primitive modules. Where possible, the vision modules all operate on common data structures, usually two-dimensional maps or masks [Mah92]. This provides some degree of composability between different vision modules, which helps the learning system to construct higher level composite visual routines. Vision modules can be dynamically wired into pathways which respond to the visual artifacts specific to a given task or sub-task (Section 6). Because I wish to construct a system which can learn to recognize, abstract, and manipulate graphic information in a wide range of visual task domains, I believe that it is unwise to push too much domain-specfic expertise into a particular vision module. This is not to say that a vision module cannot have complex or powerful special-purpose functionality, just that where possible, this functionality should be made accesible to, and composable with, other vision modules.

The following three sections illustrate some examples of proposed visual tasks. An overview of the schema learning framework is given, and then a more detailed scenario of schema learning on a visual task is presented.

1.1 Example Visual Tasks: Text-Image Separation And Identification

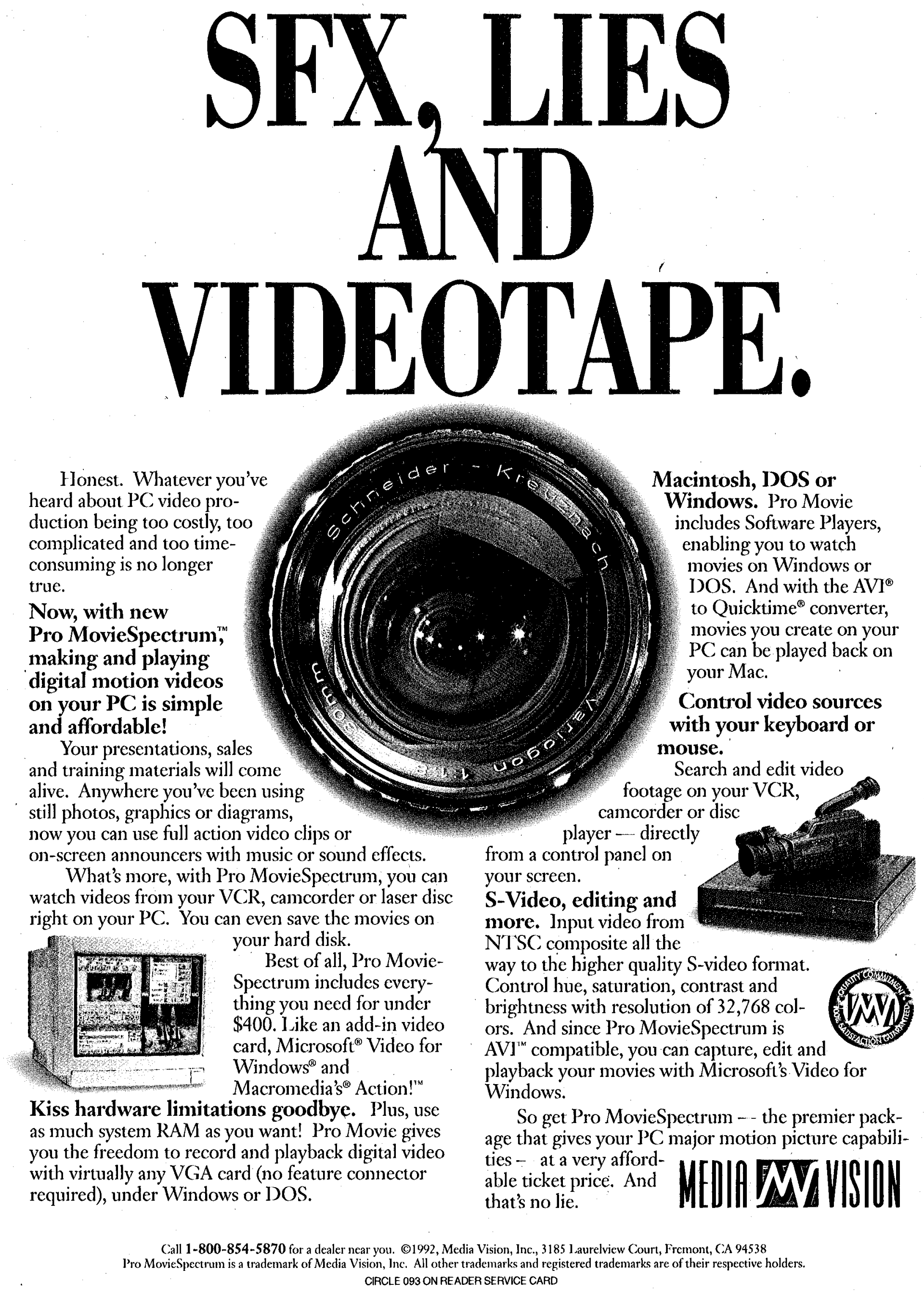
Figure 1 shows an example of a binary image from a man-made source. It consists of mostly text, laid out in regions, with descriptive pictures and graphics added for embellishment and visual appeal.

Text-image segmentation is an example of a visual task. It involves identifying regions of the page which contain certain types of visual entities, such as text strings, halftone pictures, and line art. Figure 2 shows an example of the task performed on a the magazine advertisement page image. The actions necessary to produce this segmentation are composed out of the primitive operations defined by the synthetic visual system. A more detailed enumeration of primitive image-processing operations which I propose to include in the visual system can be found in Section 7.1.

The method by which these tasks are carried out involves the use and construction of procedures which are carried out in the visual system, something which Ullman called *visual routines*. Visual routines are sequences or subroutines which instruct the system to apply primitives or other routines in the visual system. While some set of visual routines will be built into the system, it is necessary for the learning system to have the ability to construct new compound visual routines which adapt to new task requirements.

An example of a visual routine to remove text from the magazine advertisement image is shown in Figure 3. This routine uses operations which act in parallel over the entire image. Thus, it does not make use of any focus of attention or other local operations. This operation is thus a sort of preattentive routine, which would serve to draw the attention to the figures or text, for further processing by more specialized routines.

These simple open-loop image-parallel routines are somewhat of a shotgun approach, and where the image is not homogenous, they will tend to produce many unwanted artifacts. In



Schneider - Kreuznach
Variogon 1:1.8

SFX, LIES AND VIDEOTAPE.

Honest. Whatever you've heard about PC video production being too costly, too complicated and too time-consuming is no longer true.

Now, with new Pro MovieSpectrum™, making and playing digital motion videos on your PC is simple and affordable!

Your presentations, sales and training materials will come alive. Anywhere you've been using still photos, graphics or diagrams, now you can use full action video clips or on-screen announcers with music or sound effects.

What's more, with Pro MovieSpectrum, you can watch videos from your VCR, camcorder or laser disc right on your PC. You can even save the movies on your hard disk.

Best of all, Pro MovieSpectrum includes everything you need for under $400. Like an add-in video card, Microsoft® Video for Windows® and Macromedia's® Action!™

Kiss hardware limitations goodbye. Plus, use as much system RAM as you want! Pro Movie gives you the freedom to record and playback digital video with virtually any VGA card (no feature connector required), under Windows or DOS.

Macintosh, DOS or Windows. Pro Movie includes Software Players, enabling you to watch movies on Windows or DOS. And with the AVI® to Quicktime® converter, movies you create on your PC can be played back on your Mac.

Control video sources with your keyboard or mouse.

Search and edit video footage on your VCR, camcorder or disc player — directly from a control panel on your screen.

S-Video, editing and more. Input video from NTSC composite all the way to the higher quality S-video format. Control hue, saturation, contrast and brightness with resolution of 32,768 colors. And since Pro MovieSpectrum is AVI™ compatible, you can capture, edit and playback your movies with Microsoft's Video for Windows.

So get Pro MovieSpectrum -- the premier package that gives your PC major motion picture capabilities – at a very affordable ticket price. And that's no lie.

MEDIA VISION

Call **1-800-854-5870** for a dealer near you. ©1992, Media Vision, Inc., 3185 Laurelview Court, Fremont, CA 94538
Pro MovieSpectrum is a trademark of Media Vision, Inc. All other trademarks and registered trademarks are of their respective holders.
CIRCLE 093 ON READER SERVICE CARD

Figure 1: An example binary image input to the synthetic visual system.

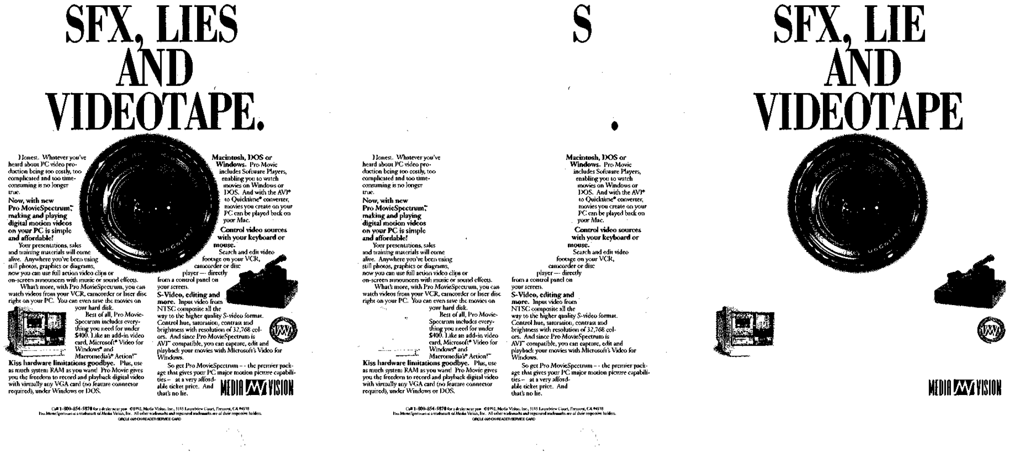


Figure 2: A sequence of erosions and dilations which is (anti)selective for the text lines

this case the routine was hand tuned by me to do a good job separating out the text. The point is that these routines are not required to be perfect, but rather they should serve as building blocks from which to form approximations to different target concepts in a task domain.

```
▷ Remove small text-like stuff
  $im4.1 \leftarrow \textbf{Reduce8x}(im1,2)$ ▷ thresholded reduction by 8
  $im4.3 \leftarrow im4.1$ ▷ copy image 4.1 to 4.3
  $im4.3 \leftarrow \textbf{Open}(im4.3,\mathrm{OCTAGON})$ ▷ remove small spots
  $im4.3 \leftarrow \textbf{FillToMask}(im4.3, im4.1)$ ▷ fill using 4.1 as a bounding mask
  $im2.3 \leftarrow \textbf{Expand4x}(im4, 3)$ ▷ expand to half original resolution
  $im2.3 \leftarrow \textbf{Dilate}(im2, 3,\mathrm{OCTAGON2X})$ ▷ dilate isotropically
  $im2.3 \leftarrow \textbf{Intersect}(im2.3,im2.1)$ ▷ mask with original image

  end
```

Figure 3: Pseudo-code for visual routine for removing text

The visual routine in Figure 3 is written in an open-loop style, which is to say that it does not use any sensory input feedback from intermediate visual system results. It is the job of the Schema mechanism to coordinate the application of these primitives, which are similar to motor output routines, with sensory feedback from the image, in order to derive more flexible task-driven sequences of routines.

The application of a given visual routine or sequence of visual routines is not intended to produce a robust solution to a visual task. The purpose of visual routines should not be to solve any given complex task completely and reliably. The visual routines should be looked at in two ways; as advanced specialized feature or "evidence" detectors which probe for the existence of various types of structure in the image, and as graphic-image manipulation routines through which the system can side effect its environment in order to accomplish the goals of a given task.

The Schema mechanism will be responsible for planning and executing sequences of visual routines, using feedback from the visual system at many intermediate points in the computation to guide the course of further processing.

In many cases of real printed pages, the text and image boundaries are fairly unambiguous, and it is seemingly an effortless task to do the segmentation. Where they are ambiguous is often intentional, in order to add visual appeal to an advertising layout or artistic graphic. The task specification is so broad that the distinction between text and graphic is not always sharp, as in the case of the large headline text "SFX, LIES AND VIDEOTAPE." in Figure 1.

Text-image segmentation requires some model of what distinguishes text from images, in terms of the primitive operations available to the visual system. I envision the system creating a set of visual routines which help it find evidence for the presence of text-like or image-like properties of regions of the page, and building progressively higher level models of what distinguishes the two.

Ultimately, I would like to see if the learning system can be taught by a series of well chosen examples to develop a set of visual routines which robustly separate out the text from the images. By robust, I mean that the system should develop models and behaviors which can be usefully adapted to new cases which it has not encountered before. It should gain some measure of "common sense" with respect to the structures it expects to find in an image.

1.2 Example: Separating Connected Components By Size

Figure 4 shows an example of a binary image derived from silhouettes of real objects. These objects have no internal texture or structure, but have complex contours. A visual task on this image might be to pack the objects into as small a space as possible, to find all objects enclosed within other objects, or to put into correspondence the objects which are most similar in shape, by some metric of functional similarity.

Another simple visual task would be to distinguish between sets of very broad classes, such as hollow and solid objects, square and round, wide and tall, or big and small objects. In each case, a sequence of progressively more difficult tasks would be presented to the system,

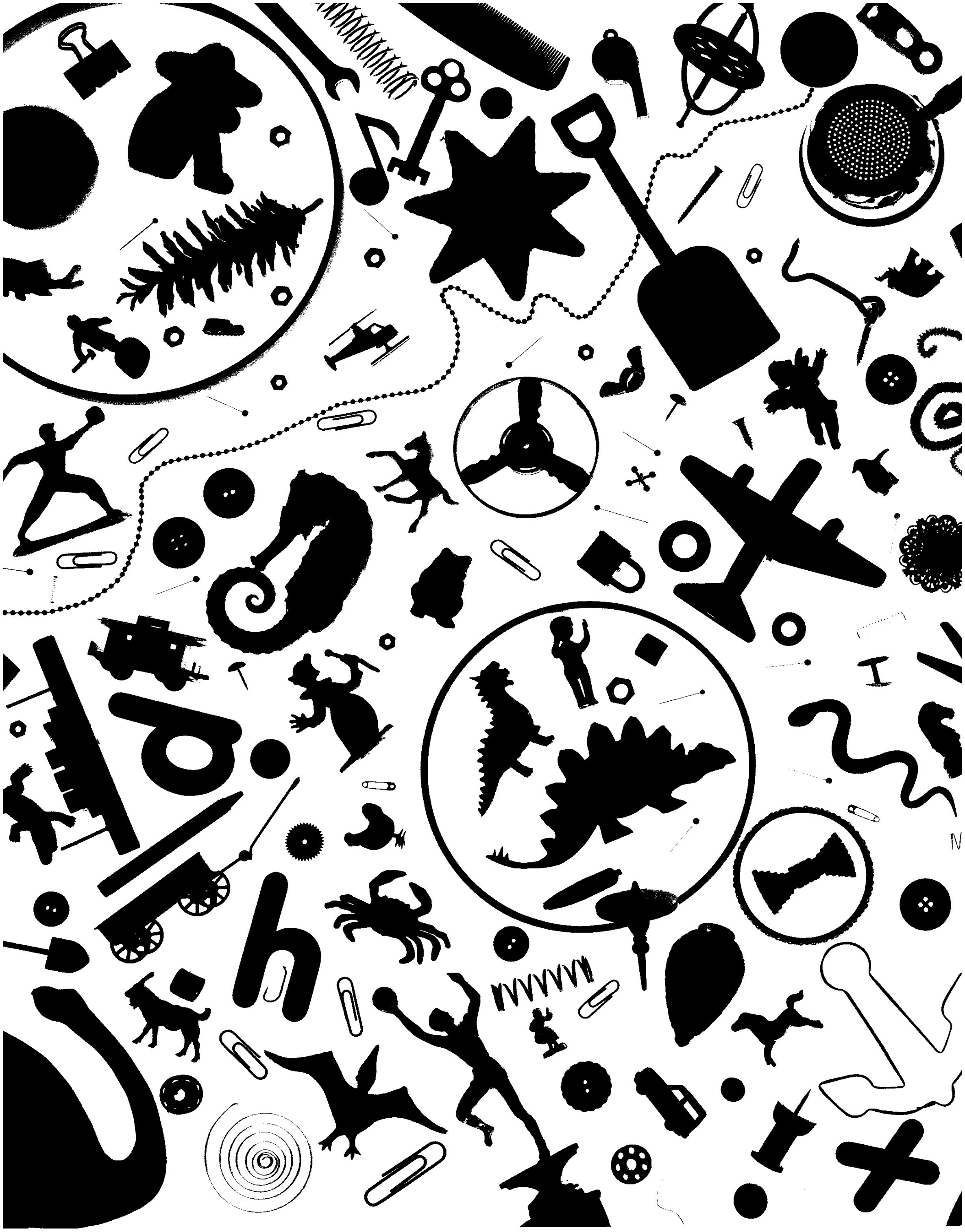


Figure 4: An example binary image input to the synthetic visual system.

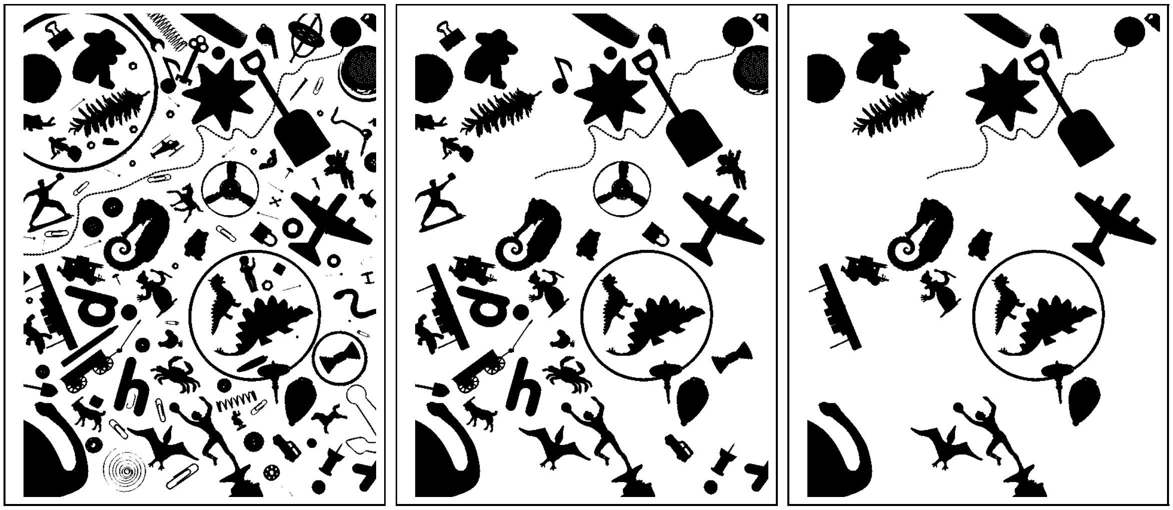


Figure 5: Output of an object-size filter visual routine

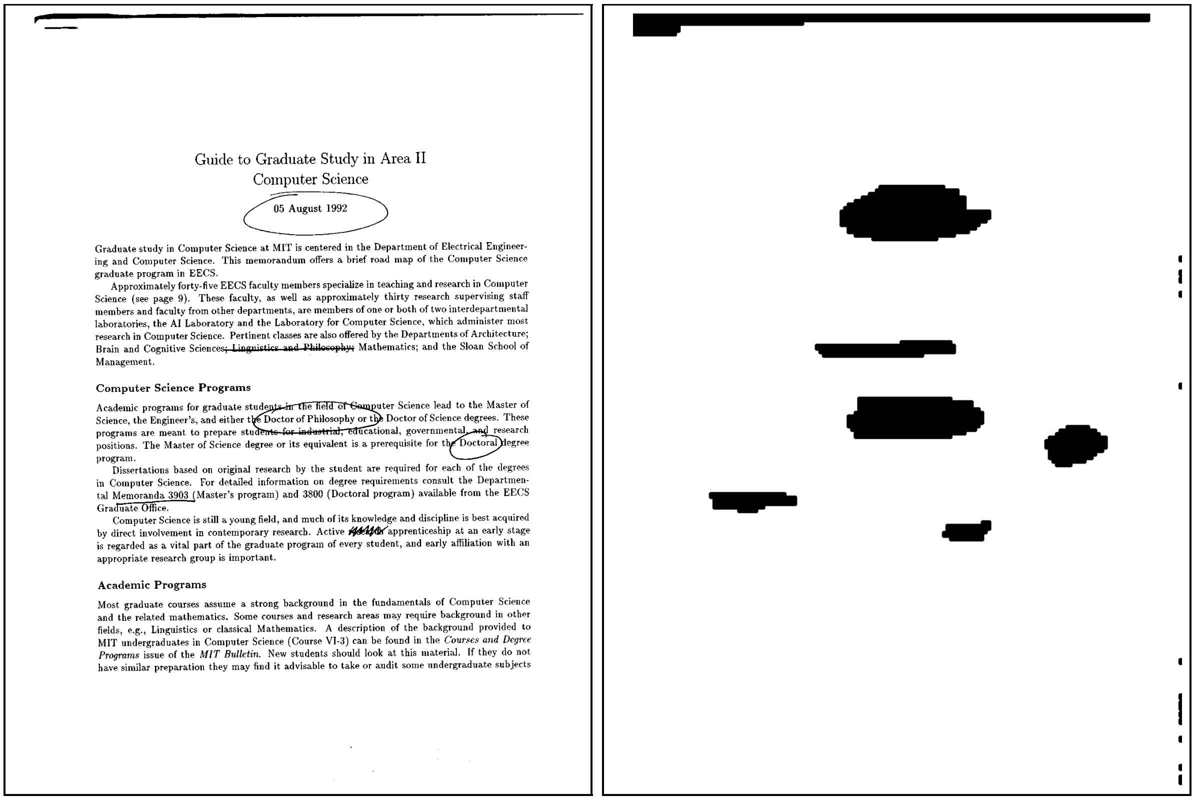
in order to allow the schema mechanism to create some sort of generalization of the object classes. Figure 5 shows the output of a hand-made visual routine to separate extract the large solid items from the picture in Figure 4.

The concept of size, in this case, is approximated by a routine which erodes the image isotropically until connected components start vanishing. This is of course not the only definition of size, but it is a *functional definition.* The system can define such properties on images by applying a sequence of operations, and categorizing the resulting changes with respect to some set of sensory inputs. In this case, number of connected components, or percentage reduction in area, would be simple metrics for the size property with respect to isotropic erosion. Other metrics of size could be computed in serial fashion, such as how far the gaze must travel to reach the edge of a connected region.

An interesting higher-level task would be to learn to recognize a bimodal distribution of object sizes in an image. That is, if you present the system with an image with "big" and "small" things which are sized relatively with respect to one another, the task is to discover which are the larger objects, or to sort the objects spatially by size. Since the "big" objects in one picture could be the "small" ones in another, depending on the distribution on objects in the scene, this would involve the system developing a scale-invariant metric for judging relative size (or at least a large set of special cases). Development of this kind of reasoning based on concrete spatial properties should be quite useful for the next higher level of qualitative reasoning which we want the system to achieve.

1.3 Example: Separating Handwritten Marks From Machine Printed Text

People take advantage of the human visual system's ability to pick contours out of background when they mark a sheet of paper with a circle or an arrow. In order to be attentive to



Guide to Graduate Study in Area II
Computer Science

05 August 1992

Graduate study in Computer Science at MIT is centered in the Department of Electrical Engineering and Computer Science. This memorandum offers a brief road map of the Computer Science graduate program in EECS.

Approximately forty-five EECS faculty members specialize in teaching and research in Computer Science (see page 9). These faculty, as well as approximately thirty research supervising staff members and faculty from other departments, are members of one or both of two interdepartmental laboratories, the AI Laboratory and the Laboratory for Computer Science, which administer most research in Computer Science. Pertinent classes are also offered by the Departments of Architecture; Brain and Cognitive Sciences~~; Linguistics and Philosophy;~~ Mathematics; and the Sloan School of Management.

Computer Science Programs

Academic programs for graduate stude~~nts in the field of Com~~puter Science lead to the Master of Science, the Engineer's, and either the Doctor of Philosophy or the Doctor of Science degrees. These programs are meant to prepare stud~~ents for industrial~~, educational, governmental~~, and~~ research positions. The Master of Science degree or its equivalent is a prerequisite for the Doctoral degree program.

Dissertations based on original research by the student are required for each of the degrees in Computer Science. For detailed information on degree requirements consult the Departmental Memoranda 3903 (Master's program) and 3800 (Doctoral program) available from the EECS Graduate Office.

Computer Science is still a young field, and much of its knowledge and discipline is best acquired by direct involvement in contemporary research. Active ~~research~~ apprenticeship at an early stage is regarded as a vital part of the graduate program of every student, and early affiliation with an appropriate research group is important.

Academic Programs

Most graduate courses assume a strong background in the fundamentals of Computer Science and the related mathematics. Some courses and research areas may require background in other fields, e.g., Linguistics or classical Mathematics. A description of the background provided to MIT undergraduates in Computer Science (Course VI-3) can be found in the *Courses and Degree Programs* issue of the *MIT Bulletin*. New students should look at this material. If they do not have similar preparation they may find it advisable to take or audit some undergraduate subjects

Figure 6: The result image is the output of a visual routine which starts with a call to a visual primitive that selects long contours with low curvature. A primitive contour filling routine is then applied.

the same sorts of cues as humans, the synthetic visual system must have some sort contour finding routines. An example of such a routine is shown in Figure 6.

This task is an example of dependence on a fairly high-level and specific primitive routine which is sensitive to a particular class of visual artifacts. In this case, the long smooth contours are semantic precursors of object boundaries, something which probably occurs a lot in nature, and is very useful for perceiving real physical objects. But, crucially, we give the system the routine for sensing the artifact, boundary contours, rather than sensing the assumed generic "object". The output of the contour finding routine is both tunable and composable. It can be used to construct a wide variety of higher-level visual routines, which may have very different task domain constraints. For example, it may be that contours along with texture sensors are needed for one task, whereas only contours and orientation information are needed for another task. We do not want to lock up the access to all the important semantic precursors in one "black box".

2 The Schema Mechanism

A more detailed learning example will be shown in Section 5, but first it is necessary to define some of the technical vocabulary used in the system. The next section gives a brief overview of the proposed learning system, and the following section describes both the visual system and the learning system in more detail.

Drescher's *schema mechanism* is a computational framework which attempts to model Piaget's constructivist account of the emergence of intelligence in children. This model asserts that elementary concepts, such as the existence of objects, are constructed by an individual as development progresses. The individual's representation of reality is constantly updated and enlarged to account for empirically observed relations in the world.

The Schema Mechanism contains an engine which is used for deducing the existence of entities or causal relationships from their manifestations, as well as for learning the results of actions performed upon these entities, and ultimately reasoning in terms of an extended vocabulary which incorporates these entities and actions as new primitives.

The basic Schema Mechanism can be viewed as a containing several key elements from different branches of artificial intelligence research; it contains a simple and efficient inductive learning engine, based on Drescher's Marginal Attribution algorithm, a powerful abstraction mechanism based on the learning system, and a strategic/reactive planner which uses the stuctures which are built by the learning engine. The results of the learning engine are available to the planner through the creation of perceptual *synthetic items*, and the results of the planner are available to the learning system through the construction of higher-level *synthetic actions*.

An overview of the schema mechanism comes from Drescher's introduction to his thesis [Dre91]:

> The schema mechanism controls, and receives sensory information from, a body. Based on its interaction with the world, the mechanism discovers regularities in the world, expressed in some existing representational vocabulary to make additional empirical discoveries expressible. The schema mechanism uses the knowledge it acquires to guide its actions, both for the sake of specific goals, and in order to gain further knowledge.
>
> The mechanism expresses regularities as *schemas*, each of which predicts some effects of an action under specified circumstances; the mechanism expresses concepts as binary state elements called *items*, each of which can be *on* or *off* to assert or deny that some condition holds. Each item can have an associated *value*; an item's value can influence the selection of those actions which, according to extant schemas, may achieve the state designated by the item. The mechanism thus follows what we might call a *prediction-value* paradigm (see section 9.1), in contrast with a situation-action paradigm: the mechanism does not directly learn what action to take in a given situation, but rather learns what would happen next for each of several possible actions. It may then select what action to take based in part on the value of an achievable result.
>
> The schema mechanism is principally concerned with empirical learning and with concept invention. For each of these intertwined processes, I identify a foundational

problem, and propose and demonstrate a partial solution.

- The foundational problem in empirical learning is that the variability of the effects of the same action in different circumstances makes an action's results hard to notice as such in the first place. A solution to the empirical-learning problem, implemented by the schema mechanism's *marginal-attribution* facility, is to use sensitive statistical measures to alternate between discovering a highly unreliable result, and then seeking conditions with respect to which the result follows more reliably.
- The foundational problem in concept invention is the need to define radically novel concepts—ones that designate entities fundamentally different from any that were previously represented (as, for example, a physical object is a much different sort of thing than a visual image or a tactile sensation). A solution is to define a new concept as the *potential to evoke a manifestation*, where the manifestation is described in terms of previously designated concepts; the schema mechanism's *synthetic items* define such concepts.

The schema mechanism is a mechanism for empirically discovering reliable production-like if-then rules, but also for creating new (higher-level) perceptual items and actions (synthetic items). At its most basic level of behavior, it uses exploration and experimentation to map out the state-space of its world, but more importantly it creates new state-items whenever something "interesting" seems to be happening. This is the first and most basic difference between schema mechanism and reinforcement learning; the state space grows extremely rapidly in schema mechanism learning. While this is a disaster for conventional reinforcement learning, it is a necessary part of the bootstrapping in order to form abstractions and empirical models of the environment.

I hope to show that in each task domain, there are abstractions which the system can make which allow its performance to improve. These abstractions are crucial to allowing the system to perform robustly on certain tasks, and in fact to perform at all on other tasks.

[A summary of the schema mechanism should go here]

- schemas are context, action, results
- primitive items, actions
- values (good, bad, pleasure, pain)
- synthetic items, compound actions
- marginal attribution
- controllers
- delegated value? what was this?
- marginal attribution: control of production of schemas

- teaching/learning by example ??

As applied to learning to operate the synthetic visual system, the schema mechanism is fundamentally a mechanism for learning action routines which *reliably* produce a resulting change in the state of an image, with respect to some set of perceptual state items. The power of the schema mechanism comes from its ability to define new perceptual state items based on the reliable routines which it identifies. This in essence allows the system to develop perceptual routines to define the features, and ultimately objects, relevant to a given task, while ignoring other features or perceptual inputs which are irrelevant.

More advanced work:

There are numerous unanswered questions about many aspects of the schema mechanism. There are critical issues having to do with more precise control over the production and garbage-collection of schemas. Also the control of attention of the system while attempting to teach it new tasks.

The schema mechanism described by Drescher can only remember things in terms of chains of schema, which means that all of its experience is embedded in a very procedural fashion. It may be desirable to add a more holistic associative event memory, in which entire or partial state vectors of the system (i.e., active state items or schemas) are stored in a temporally organized memory table. This would allow the system to retrieve partial mental states when needed, rather than relying on the more nearly linear schema activation mechanism. This would allow it to reason about remembered situations where some elements of the situation were only incidentally associated with a situation, with respect to the known or active schemas at the time. As in "Oh yeah, there *was* a man in a green car parked at the corner right before the building blew up".

The planning mechanism, as implemented with using action controllers, need to be reexamined. As described in his thesis, the simple backward-chaining planning of the system is not capable of solving a large class of important planning problems. A possible extension suggested by Drescher, *subactivation* of schemas, would allow the system to "imagine" the consequences of hypothetical actions, and thus greatly increase the space of possible solutions to problems.

2.1 Differences Between Schema Learning and Classical Inductive Concept Learning

The Schema mechanism incorporates a form of inductive concept learning, but has a far more complete intelligence model than is found in classical induction learning systems.

The nature of the abstractions formed by the Schema mechanism, while based on a form of induction learning [?][?], are fundamentally different and more powerful. Classical induction learning seeks to find new predicates on an attribute state space, given in a hypothesis language, which can classify an example instance's membership in a concept space. The induction learning algorithms assume that there is a teacher somewhere who will always assign a set-membership label to each training example. They then try to find the optimal strategy for generalizing over the examples in order to produce an optimal discriminator hypothesis, where optimality is defined by some combination of probabilistic error rates, and possibly a minimal size of the hypothesis. The inductive learning theorists have defined exceedingly complex combinatorial complexity bounds on the performance of concept learning algorithms over arbitrary instance spaces. I think that their efforts are somewhat misguided, however, because no matter how optimal a concept learning algorithm is made, I do not think that by itself it is very useful for anything approaching human-level intelligence.

The Schema mechanism uses an exceedingly simple inductive concept learning method. The marginal attribution mechanism effectively produces simple conjuctive hypotheses. The goal-driven spreading activation system, which is compiled into synthetic actions, effectively produces a concept space which is equivalent to conjuctive normal form over boolean literals; the literals being the perceptual state items of the Schema system.

The important point is that the construction of “concepts”, in the classical learning sense, is not sufficient to do anything useful with. The Schema mechanism tries to optimize its “concept” predicates, (the context field of the schemas) only as far as is needed to increase the predictive reliablity on the action taken schema above some (empirically set) threshold. Beyond this very concrete goal, there is no further need to improve the performance of the induction mechanism.

Unlike the classical induction learning model, the Schema mechanism does not rely soley on an external teacher to classify every “training” percept into its proper class. The training is directed by the relevance phase of the marginal attribution system, which identifies potential classes, and then the relevance phase poses simple conjuctive predicates to see if they improve the schema’s reliabilty. The application of the induction machinery is specifically to enhance the usefulness of schemas, and hence to make them more reliable building blocks to reason with. However, the knowledge produced is more than just simply declarative; the procedural nature of the schemas (and their synthetic items, and composite actions) is what gives the system an ability to know “what to do” to achieve desired result states, rather than just how to classify them.

2.2 Schema Learning vs. Reinforcement Learning

The system needs to construct and maintain internal state variables in order to model its environment. The Schema Mechanism contains support for automatically maintaining the state of internal synthetic items, by using supporting evidence in the form of currently or recently successful schema activations.

This is because certain tasks require the system to represent the state of the components of a scene in some internal mental model. If the system does not have the correct abstractions (or approximately correct, since there is no objectively ”correct” representation of objects), it will be unable to even begin to approach solving the task. If it cannot fully perceive and distinguish the elements of the scene which are relevant to accomplishing a task, then its performance can never improve beyond a very basic level. This is the problem of ”perceptual aliasing” [Whi92], which all reinforcement learning systems suffer from, and which is beginning to be recognized as a serious shortcoming in those systems. Certain workaround solutions have been proposed, which involve trying to avoid getting into areas in the perceptual-motor goal space where aliasing seems to exist. This solution is really just a band-aid, and by using it a reinforcement learning system can do damage-control, but cannot overcome the fundamental lack of correct internal abstractions for the task. The Schema Mechanism, on the other hand, implicitly works to solve the perceptual aliasing problem by confronting it directly, because its abstraction mechanism is automatically applied whenever a case of perceptual aliasing seems to be occuring, as indicated when it finds it is having occasional but unreliable success at activating a schema.

3 The Sensory-Motor System

The bottom-level of the sensory system is made up of a battery of primitive image processing routines which are always active, and run (conceptually) in parallel on the raw image data. These primitive sensory routines extract a base level of properties and features which will allow the system to bootstrap its visual behavior. The output of the sensory battery on a test image, a horizontal and vertical stripe pattern, is shown in Figure 7

The raw output of a base sensory perceptor routine is another image, with the same amount of data as the raw input image. This data must be squished down to a couple of bits of state to be fed into the marginal attribution system. Table 1 lists the conversion routines used to perform this data reduction. For some routines, a simple area metric is used; the change in number of ON pixels after the primitive operation is applied. For others, the number of pixels which changed value is used, in others the change in number of connected components is used.

The choice of base-level sensory primitives is somewhat arbitrary at this point. Using an empirical approach, I have selected a set of the primitives which I found useful in performing the image segmentation task manually, as well as some extra "distractor" primitives which increase the size of the search space and hopefully make a slightly less cooked domain example.

3.0.1 Primitive Sensory-Motor Actions

Table 9 lists the primitive "sensory-motor" actions available to the system. The actions are much closer to the sensory domain than the motor domain; most of them involve application of specific morphological filters or feature extractors. Nonetheless, they are actions in the sense that they must be deliberately invoked by the system.

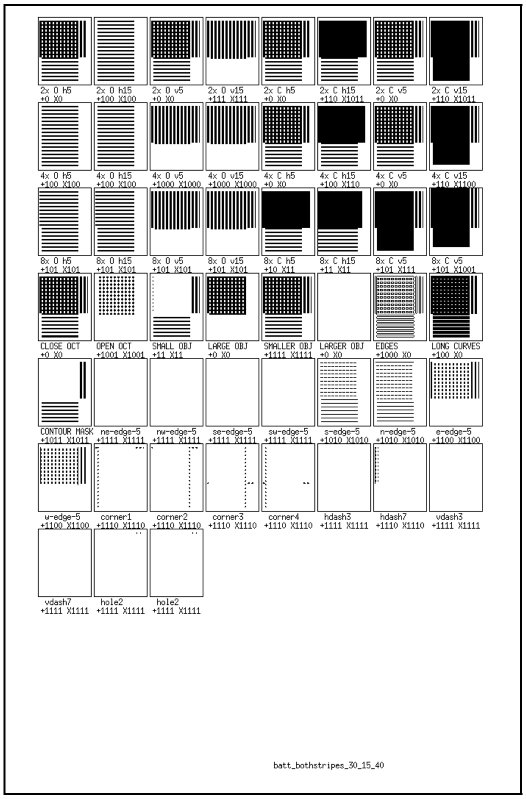


Figure 7: Each frame represents the raw output from a primitive sensor, applied to a test image of partialy crossed vertical and horizontal stripe . The raw output is converted to a few bits of state for the marginal attribution mechanism, indicated the D and X labels on the frames.

Name	**Scale**	**Meaning**
Open[V5]	2,4,8	Open for 5 pixel vertical line
Open[V15]	2,4,8	Open for 15 pixel vertical line
Close[H5]	2,4,8	Close by 5 pixel horizontal line
Close[H15]	2,4,8	Close by 15 pixel horizontal line
Close[Oct]	4	Close by octagonal kernel
FindObjects[1,3]	4	Find large objects routines
FindEdges]	4	Create edge map
LongCurves]	2,4,8	Create salient edge map
ContourMask]	4	Create map of solids from contours
DirectedEdge[NE,NW,SE,SW]	4	Open for directed edges
Corner[UL,UR,LL,LR]	4	Open for rectangular corners
Open[Hdash3]	4	Open for dashed lines with period 5
Open[Hdash7]	4	Open for dashed lines with period 7
Open[Vdash3]	4	Open for dashed lines with period 5
Open[Vdash7]	4	Open for dashed lines with period 7

Table 1: Bottom-level sensory routines. These are run continuosly, unconditionally, and in parallel.

Name	**Scale**	**Meaning**
BleedAB[...]	2,4,8	Intersection of two directional smears
BleedABC[...]	2,4,8	Intersection of three directional smears
CASeedFill[]	2,4,8	Seed fill from focus of attention
ExtractCenterCC[]	2,4,8	Extract connected component from focus of attention
FindPlaceFor[]		Find open space for object at focus of attention
NestingLevel[]		Label regions with respect to how deeply they are nested

Table 2: Additional proposed sensory routines, to be added as the system is developed.

Name	**Meaning**
Open	kernel ...
Close	kernel ...
Shift	IOR a shifted copy of image onto itself
ShiftXOR	Xor a shifted copy of image onto itself
FindObjects	Find large connected components
Corners	Convolve for corners
Junctions	Convolve for T joint, Y joint
FindEdges	Extract edges
SeedFill	Seed fill from center of attention
XOR	XOR an image with another
LeaveMarker	Leave a marker at focus of attention
ReturnMarker	Return attention to a marker

Table 3: Example task sensory-motor primitives.

4 Visual Tasks And Visual Routines

5 Example Of A Learning Task: Text-Image Segmentation

This section gives an example of the use of learning in a visual task. The task is to learn to extract text regions from other regions on a printed magazine page. For simplicity, in this example we will restrict the training examples to Roman text.

The approach is to allow the system to experiment in order to learn the effect of the application of its primitive image processing operators under different context conditions. The context conditions are themselves defined by a combination of primitive visual sensory routines, and self-constructed synthetic perceptual items.

In order to carry out the task reliably, the system will need to make use of both *image-parallel* and *sequential attention-directed* methods. The examples which follow demonstrate some plausible strategies which the system can be taught to use to coordinate its image processing using these two approaches. Image-parallel routines are used to flag candidate regions, followed by sequential routines to focus the attention on a region, follwed by possibly more image-parallel routines to confirm the initial hypothesis.

We will postpone the discussion of how the system can be trained to use its focus-of-attention machinery, and concentrate first on simple single-region images. The idea is to train the system to discriminate the two kinds of regions, text and graphics, by first letting it explore the behavior and properties of the different image regions. The construction of

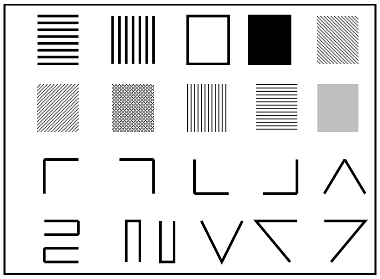


Figure 8: The system's experimental use of the primitive actions on these simple patterns leads to generation of the first layer of perceptive schemas.

these basic schemas will allow more robust and redundant representations to be built at a higher level to classify and discriminate different structures in an image.

5.1 Building Image-Parallel Schemas

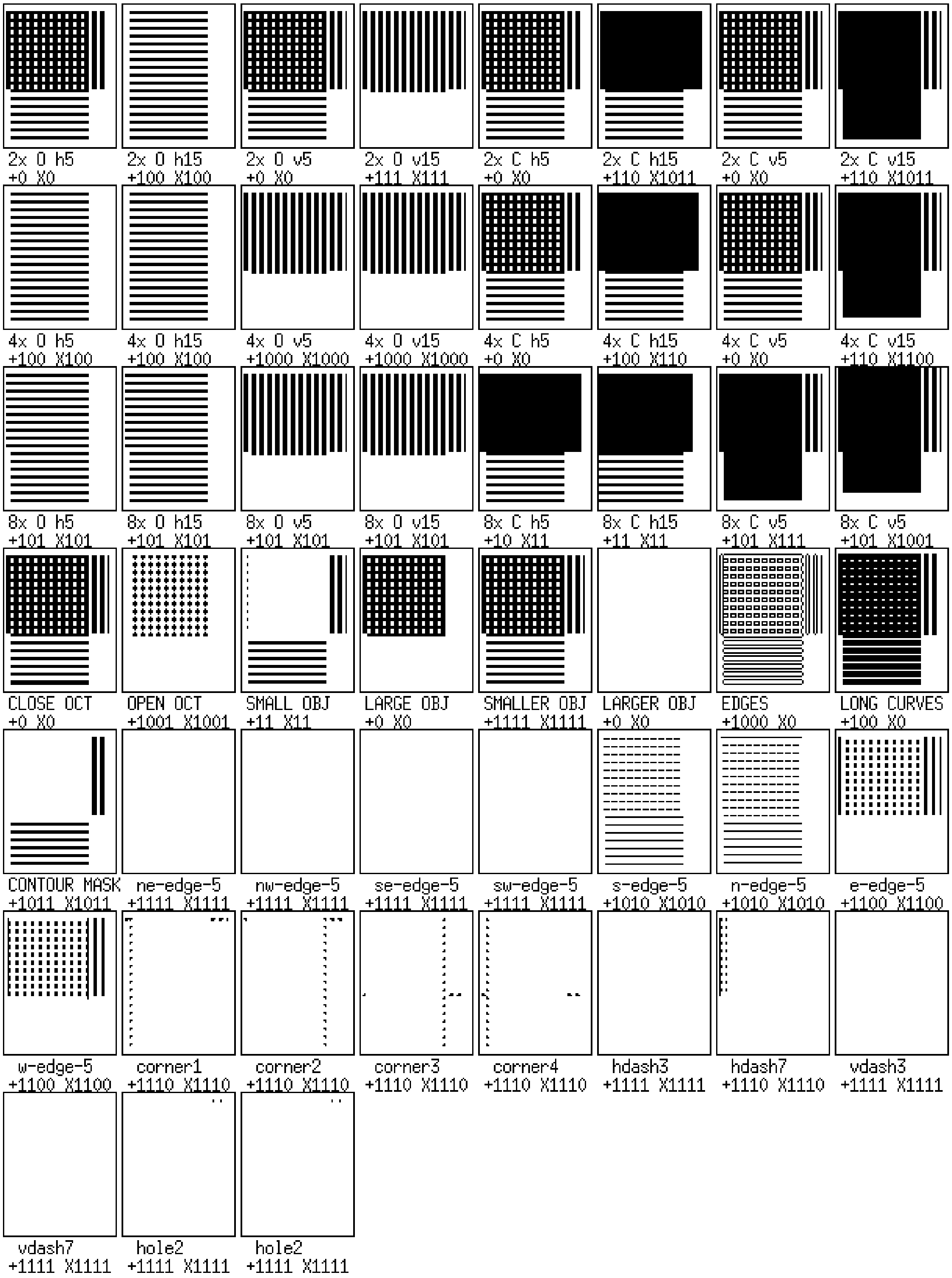
Before being asked to perform tasks on complex images, the system will first be presented with simple homogenous images. The general approach in this example will be to encourage the system to learn the properties of highly simplified image structures, such as solid regions, parallel lines, closely spaced marks, etc., and allow it to build schemas in this simplified domain. The system will then be presented with more complex images, which it can nonetheless quickly (in terms of primitive visual actions) reduce to the previous simpler class of image structures. The idea is to allow the system to gradually build up the complexity of the scenes which it can understand, by relying at each step on simple reductions to previously discovered reliable schemas.

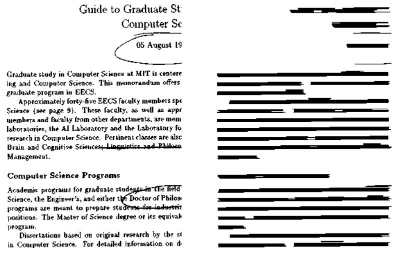
In order to apply the schema mechanism to build a model of "text-like" regions, it is necessary that there be actions which can be taken on the regions, whose outcomes, in terms of perceptual state items, are distinguishable from non-text regions.

5.1.1 Learning Of Base Level Schemas

The system will first be trained on patterns of solid horizontal bars, solid vertical bars, sparse grid patterns, and solid black regions (Figure 8). The first job of the marginal attribution mechanism is to construct sets of schemas which apply in the presence of the various image structures. Examples of the effects of the primitive actions are shown in Figure 9. For example, the horizontal bar pattern is eroded by increasingly larger vertical kernels, and progressively becomes susceptible to complete erasure by the OpenV5 primitive sensory routine at larger and larger scales. Complete erasure of the image will cause a negative transition of the OpenV5 sensory bit. This kind of transition is directly usable by the marginal attribution system, to build the base level schemas which begin to differentiate the different input images.

One of the things which distinguishes regions of horizontal text is that they are selectively preserved or destroyed under certain operations. For example, a visual routine which


2x O h5
+0 X0
2x O h15
+100 X100
2x O v5
+0 X0
2x O v15
+111 X111
2x C h5
+0 X0
2x C h15
+110 X1011
2x C v5
+0 X0
2x C v15
+110 X1011
4x O h5
+100 X100
4x O h15
+100 X100
4x O v5
+1000 X1000
4x O v15
+1000 X1000
4x C h5
+0 X0
4x C h15
+100 X110
4x C v5
+0 X0
4x C v15
+110 X1100
8x O h5
+101 X101
8x O h15
+101 X101
8x O v5
+101 X101
8x O v15
+101 X101
8x C h5
+10 X11
8x C h15
+11 X11
8x C v5
+101 X111
8x C v5
+101 X1001
CLOSE OCT
+0 X0
OPEN OCT
+1001 X1001
SMALL OBJ
+11 X11
LARGE OBJ
+0 X0
SMALLER OBJ
+1111 X1111
LARGER OBJ
+0 X0
EDGES
+1000 X0
LONG CURVES
+100 X0
CONTOUR MASK
+1011 X1011
ne-edge-5
+1111 X1111
nw-edge-5
+1111 X1111
se-edge-5
+1111 X1111
sw-edge-5
+1111 X1111
s-edge-5
+1010 X1010
n-edge-5
+1010 X1010
e-edge-5
+1100 X1100
w-edge-5
+1100 X1100
corner1
+1110 X1110
corner2
+1110 X1110
corner3
+1110 X1110
corner4
+1110 X1110
hdash3
+1111 X1111
hdash7
+1110 X1110
vdash3
+1111 X1111
vdash7
+1111 X1111
hole2
+1111 X1111
hole2
+1111 X1111



Guide to Graduate St
Computer Sc

05 August 19

Graduate study in Computer Science at MIT is centere
ing and Computer Science. This memorandum offers
graduate program in EECS.
Approximately forty-five EECS faculty members spe
Science (see page 9). These faculty, as well as appr
members and faculty from other departments, are mem
laboratories, the AI Laboratory and the Laboratory fo
research in Computer Science. Pertinent classes are also
Brain and Cognitive Sciences~~; Linguistics and Philoso~~
Management.

Computer Science Programs

Academic programs for graduate stude~~nts in the field~~
Science, the Engineer's, and either the Doctor of Philos
programs are meant to prepare stude~~nts for industri~~
positions. The Master of Science degree or its equival
program.
Dissertations based on original research by the st
in Computer Science. For detailed information on d

Figure 10: A sequence of erosions and dilations which is sensitive to the presence of text lines

selectively preserves text-like structures, under certain preconditions, is the sequence shown in Figure 10. The first operation is a morphological `close` by a small horizontal-line kernel, followed by an `open` for a somewhat larger horizontal bar. This routine tends to smooth over small inter-letter or inter-word gaps, and thus runs the letters and words together, while preserving the inter-line white space. This pair of operations causes the text lines to become solid horizontal bars, a simpler and more directly recognizable structure.

This resulting image is then completely obliterated by an open or erode operation for a vertical kernel which is taller than the text height. The sequence of operations thus serves to form a crude discriminator for the presence of text-like strucuture in a region. They are not very robust or reliable, and certainly not scale or rotation invariant. Nonetheless, they serve to form a first order approximation to a concept of "text" regions. A simpler conception of horizontal regions is one which omits the first close operation, and thus depends on the source image to contain only solid regions, rather than any fine texture. This might form an earlier precursor to the sequence shown above.

An important point is that the previous example described a routine which was selective for a certain subset of attributes of text in a region. It is this incompleteness which causes the routine to fall short of being a reliable detector of the desired concept. The routine is a first order approximation to the correct concept, however. It can serve usefully as a basis upon which the system builds its next, more accurate, set of schemas to approximate the desired concept.

Numerous other simple routines can be constructed by the system to distinguish text from graphics in a number of different cases. Simple pixel-density measurements can give useful information. Since the primitive routines described are not scale invariant, a number of schemas will need to be created, based on different absolute size properties of the training images.

5.2 Detailed Training Sequence For Learning Horizontal-Bars Schema

The following is a sequence of training examples which can be presented to the system in order to encourage it to develop useful schemas for the detection of text-like horizontal bars in a region. The training sequence is designed to push the system to develop useful ways to detect some basic geometric structures and textures. First, examples of stripes and solid regions are presented. The low-level hardware applies a fixed set of primitive operators to copies of the image in parallel. These results are used at the first level of schema production.

- The system is presented with a sequence of images of horizontal and vertical bars, with varied thicknesses and spacings. By default, the system's primitive sensory apparatus automatically performs a base set of morphological operations for a set of kernels, which consists of things like lines and edges at several rotations, corners and line terminations, and periodically spaced arrays. These feature detectors are run at several spatial scales of the image. Many of these feature detectors will cause some response when applied to the images, in terms of how the quantity of pixels in the result image which differ with the pixels in the source.

- The first set of schemas which need to be built are those which distinguish horizontal lines from vertical lines. The sensory feedback from the image-processing operations gives responses which fall into several categories; the XOR of the result image with the source will give a measure of the amount which the operation acted on the image. The act of doing an operation and noticing that it produces no change in an image should be an event noted with interest by the schema mechanism. The sensory feedback should also be noting when an operation removes all pixels, or drastically changes the number of black pixels.

  The detection of a change in the number of connected components is also of interest. The morphological OPEN operation gives feedback on a kind of local connected component information. If there are regions smaller than the kernel size, then they will be erased by the operation, whereas larger regions will remain largely unchanged.

  The sequence of a CLOSE followed by an OPEN gives

base level: black from white, primtive. horizontal from vertical. solid from hollow/textured eqv of solid and dilated texture?/

6 Visual Artifacts

The structure of the world is projected as images on the retina by the reflection and refraction of light from objects in the environment. These projections are like shadows, having form without substance, but containing information about the structure and placement of the source objects. In these images there are many artifacts can serve to delineate and uniquely identify objects or properties of objects.

These artifacts include collinear discontinuities at object occlusion boundaries, texture discontinuities at object edges, parallel lines on surface contours, and coherent motion fields. They are visual manifestations of the structure and properties of physical objects, and of the relations between physical objects. In this proposal, *visual artifacts* will refer to visual cues which allow a system to infer the existence of underlying structure.

Central to this thesis is the idea that a properly constructed visual system should be able to extract these visual artifacts in a form suitable for interface to a learning system. In [Pen86], Witkin and Tenenbaum introduce the idea of *semantic precursors* [Pen86]:

> ...when we strongly perceive a structural relationship, we are implicitly asserting that there is a corresponding causal relationship. Whatever the relationship means, it means *something.*
>
> For example, having discovered a compelling parallelism relationship, we may be free to interpret it in many ways – a curved surface, a waving flag, etc.–but we can never dismiss it. As we proceed from the level of primitive structure to a high-level semantic interpretation, the bare assertion of non-accidental parallelism may evolve into a more specific assertion of surfacehood, and the detailed shape description of the parallel curves may be reinterpreted as the shape of a surface. But to whatever the primitive relationship is eventually attributed, it is almost certain to survive in *some* form to the highest levels of interpretation. In effect, perceived structural relationships are "semantic precursors," deserving and demanding explanation by subsequent interpretation. In this role, the primitive structural description provides constraints on subsequent interpretation–in the form of facts that ought to be explained–and also provides a bootstrap by which higher-level interpretations may be obtained, because the structural relationships resemble the underlying causal ones.

This thesis will examine the question of whether the schema mechanism is capable of learning to use semantic precursors in the form of visual artifacts, in order to interpret images and perform tasks, and in fact to carry out the bootstrapping to the creation and use of successively higher-level representations. The proposed visual system is not, however, a passive set of feature detectors or filters for visual artifacts. The visual system is much like a sensory-motor system, which the schema mechanism must learn to operate. This paradigm of *internal active perception* is explained in more detail in Section 6.3.

Figures 11 (a,c,d) from [Mar84] show examples of a visual artifact object occlusion; of collinear terminations of lines which indicate boundaries [1].

6.1 Visual Entities: Elements Of Images

6.2 Are Pictures Of Things As Good As Things?

In Drescher's thesis, the computer infant was placed in a world which was a highly simplified model of the real world. The system had a crude visual system, as well as tactile feedback from a hand. The simulated objects in the microworld were analogs of things sort of like toy blocks in the real world.

[1] Figure 11 (d) shows a counterexample to a proposed explanation for the perception of a circular disk, that the visual system looks for the intersection of extensions of radial spokes.

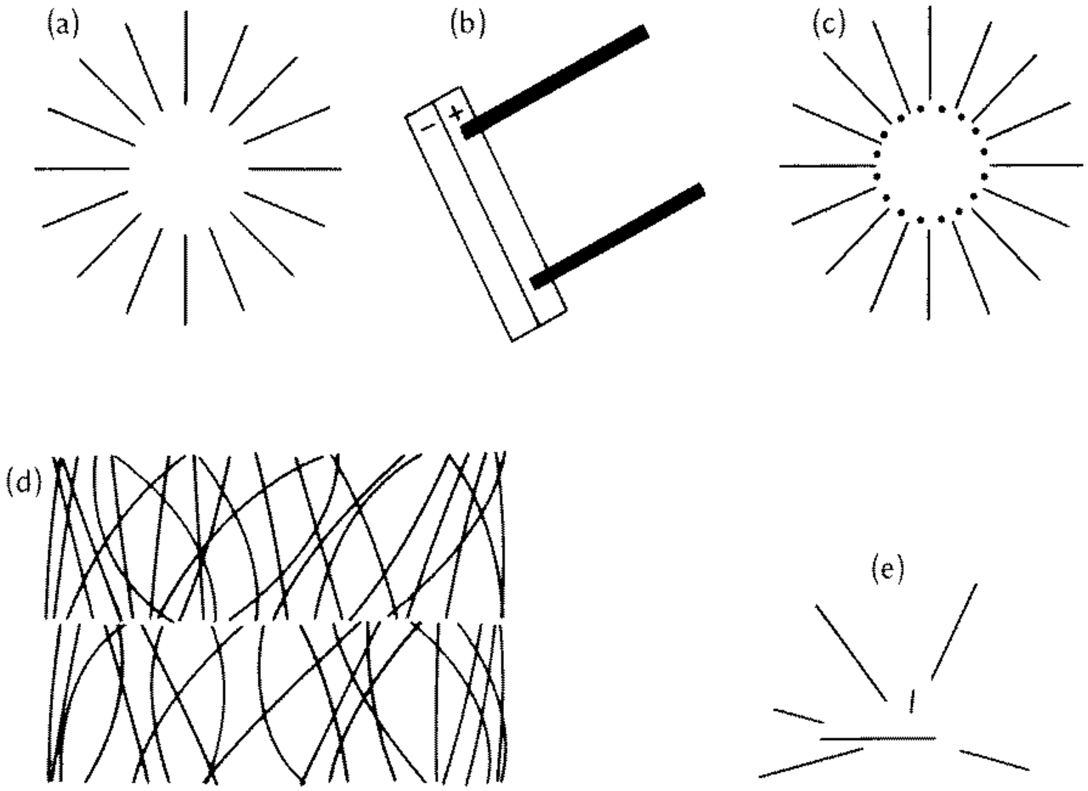


Figure 11: Salient contours at occlusion boundaries [Mar84].

The microworld which I am proposing, while having a much richer visual system, does not have any direct analog to physical tactile or mechanical properties of objects. In this sense it is very divergent from the real physical world. I am thinking of adding a 2-1/2 dimensional semantics to the microworld, whereby the system can manipulate two-dimensional shapes, which have a stacking property, such that a shape can be placed either in front of or behind another object. This will allow the possibility of interesting learning tasks, such as the perception of occlusion boundaries and the concept of objects which exist but are not necessarily visible.

Without the semantics associated with a 2-1/2 or three dimensional world, I do not think that it is possible to expect the system to develop any concept of partially obscured objects, since there is no perceptible action which corresponds to covering or uncovering an object.

6.3 Internal Active Perception

The human visual system is made to find and collect evidence of things, through their manifestation as visual artifacts, to allow the other agencies in the brain to reason about the consequences of their existence.

The visual system does operations on images, and constructs intermediate visual results which are not directly accessible to the central reasoning system (i.e., as schema state items). It does however feed out distilled information, which can be used directly by the schema mechanism. These distillations are either one-bit or a scalar value with a small number of bits. Some example visual system sensory outputs are simple binary flags to indicate whether any bits remain on after an operation, or a low-resolution scalar of count of the on-pixels in a region or the size or area of a connected-component.

The application of a visual primitive or routine can be viewed in some contexts as a probe the environment (image). For example, an operation such as filtering for horizontal line segments or filtering for objects with holes will produce a resultant intermediate result image. The new image is not directly accessible to the learning system, for it contains as many pixels as the source image, but a count of the remaining "on" pixels gives a measure of the response of the operation to the image back from the visual system.

When the visual system is architected in this manner, I call it *internal active perception.* The implication is that for any meaningful information to be extracted, an image must be probed in a purposeful manner by the application of sequences of primitive visual system control operations. The state space of possible application sequences is enormous, and exponential. The hope is that these probe sequences can be incrementally constructed or improved by the system itself, using the schema mechanism. The system must explore its own internal world as well as the external world.

6.4 Visual Communication

While the human visual system may have been developed to deal with naturally occurring information, humans today frequently use abstracted forms of visual information, in written material, signs, diagrams, and other sorts of visual communication. Diagrams and signs are convey information by taking advantage of the capabilities of the existing human visual system to segment and organize the picture into virtual objects and relations between those objects.

Visual tasks are tasks which involve perception and recognition of visual entities in images, and modification of the image by the system to achieve desired a desired goal configuration of entities in the image.

Visual entities are defined with respect to a given task. They correspond to real physical things at only the most basic level; a coherent structure of space, composed of arrangements of visual artifacts which hold certain repeatable or invariant properties with respect to certain visual-system transformations. It is the job of the visual system to provide primitive image operators and routines which allow a useful and richly descriptive set of attributes of visual entities to be sensed. It is the job of the schema mechanism to learn to perceive the entities which are relevant to a given visual task, and ultimately to perceive the world as relations between these entities.

6.5 Learning Representation And Actions

6.5.1 Learning vs. Programming

7 Visual System Architecture

multi-scale operations

7.1 Image Processing Primitives

7.2 Texture Primitives

7.3 Pop-out Effects

One useful skill of the human visual system is the ability to pick out a single exceptional feature from a uniform background, such as shown in Figure 12.

This effect could be implemented using a suppression mechanism to filter strong responses before the image is presented to the attention mechanism. The strong, spatially uniform response of some of the oriented edge detectors to the majority of slanted lines in Figure 12 could be used to disqualify them from submitting maps to the attention mechanism.

7.4 Scale and Rotation Invariance; A Bad Idea?

Many approaches to object recognition use scale, translation, or rotation invariant transforms [ref?]. I don't believe that this is generally necessary or desirable for the type of visual learning which I wish to have my system perform. Vital information is discarded when one of the invariant transforms is applied; i.e, how big the object is, its location, or its orientation.

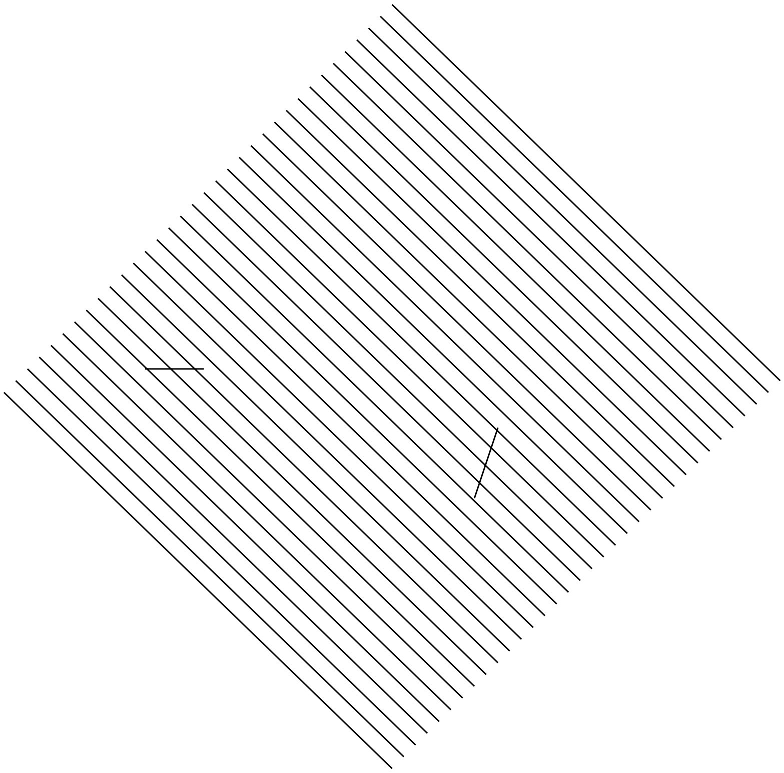


Figure 12: An example of a pop-out effect.

I think it is good for the system to notice *when* an image *is* invariant under some primitive visual routine or tranform, but that is a very different kind of information. Real objects do occur at different distances, positions, and rotations, and it is necessary at some level to efficiently normalize out these effects, in order to recognize these multiple views as coming from the a single object. But I believe that that this ultimate goal can be achieved through the use of many simple and *weakly invariant* operations, such as the parallel horizontal line filter shown in Figure 10. This filter is applied over an entire image regions, and so is in a sense translation invariant. It happens to produce a map of responses with the absolute spatial information preserved about pixels which responded. The learning system will look first at the strength of the overall response, and then later perhaps will focus its attention on the regions of high response, or will mask them out in a parallel fashion using boolean image-to-image operations. Nevertheless, it is a very simple operator that extracts just one very particular piece of visual evidence from the scene. It is my hope that the application of many of these simple operators can build a much more robust description of the scene than the application of a single complex transform, such as the Hough transform.

My intuition is that the focus of attention reflexes, like saccadic eye motion, tend to normalize objects to the center of the field of view as quickly as possible, when detailed discrimination is needed. Image-parallel operations are used probably used to generate likely positions of interesting features, but the object cannot be carefully indentified if it is not close to its "normal" position. In the case of rotation, I believe that the most common rotations are probably learned as individual cases, an multiple stored views are associated with a single object, rather than a single rotation invariant description. In some cases, it is necessary to tilt the head, or rotate the eyeball, in order to bring the image into line with a known stored description. When this is not possible, it can be very difficult to recognize objects in unfamiliar orientations; for example, try reading this page upside down.

7.5 Image Register Architecture

The basic architecture of the visual system will be constructed from a set of image-registers. Operations will be provided to scale an image or mask up or down, or to copy a region of one register into another. Image registers at several different scales will be provided.

Some set of the image registers, the *primary image registers* are dedicated input registers; they are constantly updated in parallel from the raw visual input source. In the case of a real-time or time-varying sequence of images, the primary input registers are updated as fast as possible. In the case of viewing a single static image, such as a document, some of the primary input registers are associated with the current focus of attention, and are updated as the attention point is moved by the system. Each primary input registers has a permanently associated image operator function associated with them, although some of the parameters may be varied under the agents control. Examples are an edge detection unit, a blob detection unit, a set of oriented edge maps, etc.

The *secondary image registers* are able to perform operations under the control of the learning system. These are useful for intermediate storage for learned visual routines. The image resgister architecture allows certain sets of registers to be pipelined together, to produce visual datapaths. Visual routines are generally composed this way.

7.6 Implementation On A Workstation Class Computer

References

[Dre91] Gary Drescher. *Made-up Minds.* MIT Press, Cambridge, MA, 1991.

[Mah87] James V. Mahoney. Image chunking: Defining spatial building blocks for scene analysis. Technical Report TR-980, MIT AI Lab, Cambridge, MA, 1987.

[Mah92] James Mahoney. Signal geometry. Unpublished technical report, Xerox PARC., 3333 Coyote Hill Rd., Palo Alto CA, 1992.

[Mar84] David Marr. *Early Processing of Visual Information*, chapter 1. Ablex, 1984.

[Pen86] Alex P. Pentland, editor. *From Pixels To Predicates*, chapter 7 On Perceptual Organization, pages 149–169. Ablex, 1986. Chapter by: Andrew P. Witkin, Jay M. Tenenbaum.

[Sha88] Amnon Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. Technical ReporAI Memot AIM-1061, MIT AI Lab, Cambridge, MA, 1988.

[Ull84] Shimon Ullman. Visual routines. *Cognition*, 18:97–159, 1984.

[Whi92] Steven D. Whitehead. *Reinforcement Learning for the Adaptive Control of Perception and Action.* PhD thesis, University of Rochester, Rochester, New York, February 1992.