UC Irvine
ICS Technical Reports

Title
Recognition by directed attention to recursively partitioned images

Permalink
https://escholarship.org/uc/item/4st2n2qb

Author
McNulty, Dale M.

Publication Date
1988

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

Archives
Z
699
C3
no. 88-08
c. 2

The University of California

Irvine

RECOGNITION BY DIRECTED ATTENTION TO RECURSIVELY PARTITIONED IMAGES

Technical Report #88-08

Dale M. McNulty

Department of Information and Computer Science
University of California
Irvine, CA 92717
ARPAnet Address: MCNULTY@ICS.UCI.EDU
714-856-6360

Abstract

A learning/recognition model (and instantiating program) is described which recursively combines the learning paradigms of conceptual clustering (Michalski, 1980) and learning-from-examples to resolve the ambiguities of real-world recognition. The model is based on neurophysiological and psychological evidence that the visual system is analytic, hierarchical, and composed of a parallel/serial dichotomy (many, see conclusions by Crick, 1984). Emulating the experimental evidence, parallel processes in the model decompose the image into components and cluster the constituents in much the same way as the image processing technique known as moment analysis (Alt, 1962). Serial, attentive mechanisms then reassemble the decompositions by investigating spatial relationships between components. The use of attentive mechanisms extends the moment analysis technique to handle alterations in structure and solves the contention problem created by combining the two learning paradigms. The contention results from a disagreement between the teacher and the model on what constitutes the salient features at the highest level of the symbol. There are four cases ZBT must handle, two of which result from the disagreement with the teacher. The parallel/serial dichotomy represents a vertical/horizontal tradeoff between the invariant and variant features of a domain. The resultant learned hierarchy allows ZBT to recognize structural differences while avoiding problems of exponential growth.

This research was supported in part by the office of Naval Research under grants N00014-84-K-0391 and N00014-85-K-0854, by the Army Research Institute under contract MDA903-85-C-0324, and by the National Science Foundation under grants IST-81-2-685 and IST-85-12419.

...tion: This Material
...y be produced
...by Copyright Law
...Title 17 U.S.C.)

1.0 Introduction

Machine recognition is difficult in simple image environments, but the problem is compounded in a real-world paradigm where objects undergo various types of transformations. The problem can be viewed as a comparison of similarities and differences from one viewing to the next. There are two important aspects to the problem.

First is the difficulty of understanding the scanned raster data. This part of the problem can be simulated by pressing one's eye against a television screen and attempting to identify what is being viewed by interpreting the dots that compose the image. Viewed at a distance, the contents of the image, such as the call letters identifying the broadcast station, might be discernible, but at the closer viewpoint, things are confusing - the letters cannot be deciphered from the complexity of dots that compose them.

A simple approach to this problem, as the previous statement implies, is to back away from the detail and focus attention on the larger, less detailed constructs. This seems to work for a human, but implementation in a machine vision system is problematic. How does a machine back away from the raster image it holds in memory?

And, if a machine can be made to effectively back away from the image it is analyzing, how does it cope with the variability of a real-world domain, variability easily overlooked by a human viewer, but a formidable problem to the machine vision system. The difficulty is that, from one viewing to the next, various types of image alterations and interference can make visually identical objects, such as the call-letter characters, appear to be different, and different objects appear to be the same. The characters can vary in size, can be rotated, and can even be structurally altered by "snow" and other distortions of the image. This is the second part of the machine vision problem: despite variations in images the machine must identify the differences and similarities between objects such that the learned knowledge is predictive of the environment.

Neurophysiological and psychological data suggest that biological vision systems solve these problems in an analytic, hierarchical fashion (Julesz, 1962, 1984, Crick, 1984, Moran & Desimone, 1985, Treisman, 1985). First, raster-like data, converted from continuous tone data by discrete elements in the back of the eye, is canonically decomposed or grouped in the early visual system according to fixed primitives such as: contrast, spatial orientation, and color (summarized in Van Essen & Maunsell, 1983). Then, in later processing stages, the features are reassociated to compose recognizable objects. Psychological experiments suggest that while decomposition is an effortless, parallel process (Julesz, 1984), the reassembly is a serial process requiring attentive resources (Treisman, 1985).

This paper describes a learning, recognition model, ZBT, (and instantiating program) that achieves some success on simple, paradigmatic examples by emulating the accumulating neurophysiological and psychological evidence that the visual system is analytic, hierarchical, and composed of both parallel and serial processing elements (Crick, 1984).

ZBT's processing begins with the least detailed level of the visual form and progresses to the most detailed level. At each level a two-step, matching process is invoked. In the first step, parallel processes select an aspect of the image and decompose it into meaningless, component blobs according to simple, fixed primitives. Other parallel mechanisms then compute features of the blobs that are invariant to the real-world transformations of translation, rotation, and scale and use these features to index into every level of the memory hierarchy simultaneously. This is a parallel memory match that isolates candidate recognitions.

In the second step, attention is focused serially on the spatial relationships between component blobs. This is the serial portion of the match where candidate recognitions are verified.

ZBT's two steps coincide with two different paradigmatic approaches to learning: conceptual clustering and learning from examples. The first step, clustering, is performed in the absence of environmental feedback to group data according to the invariant primitives. The second step is ZBT's method of conforming to the environment's classifications which can conflict with the clustering process by negating their groupings. The conflict creates two ambiguous situations during the match process: objects can appear the same (to ZBT) but be different, or appear different and be classified the same. In reality, the ambiguities result from a mismatch of measures of similarity chosen by system and environment at the highest level of object recognition.

One solution to the conflict is to reorganize memory but ZBT takes a different approach. ZBT attempts to resolve the conflict by searching for other levels of detail that might disambiguate. It does this by recursing on the level of detail where the conflict occurred. The recursive combination of the two steps creates an hierarchical memory organization where the vertical levels of the hierarchy represent increasing amounts of image detail and horizontal relationships within levels represent the spatial relationships between the features that compose a level. Indexing each level of detail in the hierarchy with the invariant measures helps ZBT avoid the problem of exponential growth frequently associated with the extension of learning systems.

2.0 The Problem

Briefly stated, the problem considered is the difficulty of learned object recognition that confronts biological vision systems which must cope with imperfect environments, and, thus, must cope with potentially infinite perceptual domains. The problem has been simulated in this research with the corresponding problem of recognizing binary, raster-scanned images under certain transformations. The following assumptions detail the research paradigm:

(a) Input to the system is a series of images representing the target symbols to be learned/recognized. Presented singly, each image is a binary, raster array (bitmap) such as might result from scanning a continuous tone, or gray scale, drawing of the respective symbol[1]. The recognition problem is simulated in this work with the raster-scan problem because of the apparent correspondence between scanning technology and the earliest processes of biological vision systems. Scanning systems, typified by charge coupled devices (CCD), quantize reflected light, mapping light intensity onto a matrix of numerical values, where each location (pixel) represents the amount of light collected by a sensor in that respective area. In a similar way, sensing elements in the retina of the eye, innervated by different types and quantities of light, map the continuous light tones onto an array of discrete values. The problem has been somewhat simplified in this work by assuming that the continuous tone images are thresholded to produce binary arrays such that pixels are either 1 (on) or 0 (off) representing background.

(b) The system must be capable of incremental learning behavior (see Schlimmer & Fisher, 1986 for motivation). That is, it must base its responses on the incremental acquisition of knowledge.

[1] A future direction is to interface the system to a document scanner. For the present, however, the raster representations have been created by hand; placing 1s for black and 0s for white (background).

(c) The system must be easily extensible. In other words, the system must be capable of easily learning new symbols without increasing learning/recognition times exponentially.

(d) The system should tolerate four possible real world image variations:

1) Translation: A symbol's location in the image field can vary from one experience to another.
2) Scaling: A symbol's size can also vary from experience to experience. (However it is assumed that the size should always be large compared to the resolution.)
3) Rotation: 2-dimensional (2D) rotations of less than 45 degrees should not impede recognition. While biological vision systems seem to tolerate rotation, no psychological model explaining it or suggesting its limitations has yet to emerge. The assumption in this work is that some amount of rotation must be tolerated, but the difficulties of rotational confusion (e.g., rotate a "d" and it eventually becomes confused with a "p") must be avoided, thus rotations are limited to 45 degrees.
4) Simple structural alterations: Variations in structural characteristics as might result from omissions (due to occluding noise or sloppy printing) are allowed. Figure 1 illustrates three symbols selected as typical of one type of structural alteration and used throughout the examples presented here to explain ZBT's operation. Object1 is a well-formed capital letter "A" composed of three straight-line strokes, labeled (for purposes of the discussion): left side, right side, and brace. Object2 is an alteration of Object1 in which the connection between the right end of the brace and the right side has been broken. Object3 is identical to Object2 except that the break between the brace and side is wider.

(e) Optionally, each image (as input) can be accompanied by a label identifying the category assigned to the represented symbol. The category is the class, specified by the teacher or environment, that contains all of the forms assigned the same label. For example, Object1, Object2, and Object3 of Figure 1 would, typically, all be labeled as members of Class A and Object4 would typically be labeled as a member of Class Y. Labels, as defined by the teacher, are assumed to be consistent.

(f) If a label does not accompany the input, the system should include, as part of its output, a statement specifying the class the current symbol it is perceived to be a member of.

(g) The domain of symbols is not limited (except for practical limitations of implementation discussed later - see Appendix II).

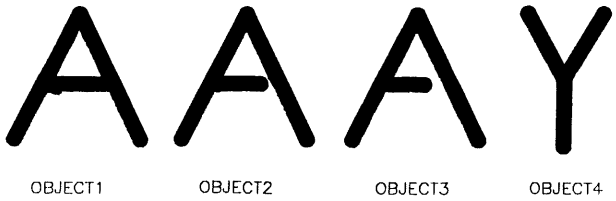

Figure 1: Example Structures

3.0 The Model - An Overview

The goal of the research described here is an investigation of the role of attention in real-world, image recognition problems. The result has been an incremental learning/recognition theory, ZBT, and its program instantiation. The purpose of this paper is to describe the operation of the

ZBT model in detail. This will be done in a later section when the computations are delineated, but, first, an overview of the model's operation and its relevance to certain experimental data provide a perspective for the detail that follows.

ZBT is based on a number of ideas which have been culled from the pertinent physiological and psychological data. These points are discussed later, but can be summarized as:

- The visual system is modular; different areas of the visual system appear to analyze an image in different ways (summarized in VanEssen & Maunsell, 1983).
- The information between modules is somehow reassociated to form a whole.
- Some recognition processes are automatic and seem to operate in parallel (many including: Julesz, 1984).
- Some recognition processes require attentive resources that appear to operate in a serial manner (ibid.).
- Different levels of recognition detail can be brought to bear on an image at different times (see examples discussed later).

Consistent with these points, ZBT can be characterized by the following two step process:

1) Parallel decomposition of the current attentive area
2) Serial focus of attention on the resultant components

The two steps correspond to two different learning paradigms: conceptual clustering and learning from examples[2]. The conceptual clustering is embodied in parallel processes that cluster the data according to feature similarities perceived by the program. Unfortunately, as we'll see, the resultant grouping may not agree with the grouping preferred by the environment. In that case ZBT invokes the serial, attentive mechanisms to resolve the disagreement between the two modes of operation.

The disagreement typically occurs at the highest or least detail level of the objects in question. By recursively combining the two learning paradigms, ZBT attempts to expose, and in turn learn, detail that will unambiguously identify the nature of the disagreement. That is, the recursive, two-step process, alternating between decomposition and attention to relationships between the components of the decomposition, zooms in on the detail that distinguishes one visual object from another. An example serves to illustrate.

Working on one image at a time[3], ZBT attempts to identify the symbol contained within each image[4]. To accomplish this, ZBT first isolates the object contained within the raster display of the current image. For example, when viewing an image of Object1, ZBT first isolates the "A" form from the rest of the raster background. This form is the level of least detail associated with Object1. This level is also the level of greatest aggregation since the components that define the object are aggregated into one composite structure[5].

---

2 The input requirements dictate that ZBT can not always depend on environmental feedback, but it must also respond appropriately when it is available. Since the fundamental difference between the two learning paradigms is that the latter incorporates environmental feedback and the former does not, ZBT combines the two types of learning. The two paradigms are covered in a later discussion.

3 Typically, ZBT's operation is studied on a sequence of images where a sequence can be 2 images or it can be hundreds. ZBT begins a sequence with no prior knowledge of previous experiences. As an incremental learning system, it builds its knowledge base incrementally with each image. ZBT's operation on a sequence will be explained later.

4 As a conceptual clustering system ZBT stores its knowledge of an object whether or not the category of the object is provided with the image.

5 Later, when more structural detail is required, ZBT will decompose Object1 into 3 constituent parts, however, at the highest level, ZBT ignores the 3 component strokes, focusing instead on the composite form.

ZBT next abstracts the experience of the composite form by computing certain values, called moment invariants, on the isolated raster array. These values represent the unique physical features of the raster form, invariant to the transformations of translation, scale, and rotation.

Using the invariant values as indices (pointers), ZBT references memory. The mechanics of this are discussed later, but the essence is that the invariants serve to cluster, or group, abstracted experiences in memory locations on the basis of features that are independent of the stated transformations. Thus, the result of a memory reference in this example is a reminding of any previous experiences possessing physical features equivalent to the Object1 form, even if those previous experiences were translated, rotated, or scaled versions of the Object1 form.

If no label is provided with the image, the results of the memory reference dictate that one of two possible actions should be undertaken[6]:

1) If no comparable experience is found in memory (i.e., in this example, if ZBT does not possess in memory an experience comparable to the least detail level of Object1), then ZBT abstracts the current experience, stores it in memory, and indexes it (so that it can be referenced later).
2) If a comparable experience is found, ZBT reports the category associated with the match as the likely category of the current experience.

If, on the other hand, a label is provided as input, the situation can become confounded by two possible conflicts:

3) The current experience can possess a label identical to a stored experience that is not physically similar to the current experience. This is comparable to saying that ZBT has experienced two physically different forms that have the same label.
4) The current experience can remind ZBT of a previous, similar experience that possesses a label different from the current experience. This is comparable to assigning two different labels to a single experience.

ZBT attempts to resolve these conflicts by looking for more detail in the present image to distinguish it from the conflict. It does this by first attending to spatial relationships and then recursing. Therefore, in this example, ZBT would decompose the top level form of Object1 into its three constituent parts, reference memory, and serially attend to the spatial relationships between them. If no physical aspect differentiating the current structure from the conflicting structure is found, ZBT would then choose one component (i.e., focus attention) and recurse. In this way, ZBT builds a memory hierarchy that represents different levels of image detail at each level of the hierarchy. Consistent with the evidential points, ZBT can, therefore, apply different levels of its recognition memory to different aspects of the visual scene.

A more detailed example provides greater insight into ZBT's operation, but first a more thorough discussion of the experimental data and other modeling attempts is presented.

4.0 The Evidence: Psychological and Neurophysiological Contributions

4.1 Image Decomposition and Features

In the early visual system, physiologists have identified a modular organization, demarcated by the distinct elements of the physical world to which the modules are sensitive (see VanEssen & Maunsell, 1983, for a summary). Illustrative of this modularity are groups of cells that map the physical world in elementary detail by responding to very specific visual features (Hubel & Wiesel, 1962, 1968, and 1977, and Moran & Desimone, 1985). For example, researchers have

[6] The numbering coincides with four cases presented later.

identified cells in Area 17, the first visual area of the neocortex, that fire only in the presence of contrast lines possessing a specific orientation within the visual frame. Other cells have been found that map specific spatial frequencies and still other cells respond to other elementary aspects of the environment, such as motion and color.

Compared to the early visual system, later stages of visual processing become progressively more abstract. In other words, as information moves through the visual system, specific image detail (e.g., location within the visual field) is discarded and increasingly more complex information replaces it (Bruce, Desimone, & Gross, 1981, and Perrett, Rolls, & Caan, 1982).

Psychological evidence supporting canonical decomposition in the early visual system was first suggested by Beck (1967) who reported an apparent visual grouping based on three specific segregating features, or primitives: line orientation, contrast, and color. He found, for example, that subjects typically segregate one contrast area from another contrast area and tilted "T"s from straight "T"s. He contrasted those groupings with simple spatial relationships which his subjects did not easily distinguish, such as "T"s from "L"s.

Since Beck, additional psychological evidence has been reported supporting an analytic visual system. Among others reporting such evidence are Julesz (1983) and Treisman (1982, 1983, Treisman & Gelade, 1980, Treisman & Schmidt, 1982, Treisman & Paterson, 1984) who have built more detailed cases for specific decomposition mechanisms. Additionally, both have demonstrated what appears to be a processing dichotomy that distinguishes the analytic portions of the visual system from attentive aspects of processing.

4.2 Attention

In Treisman's visual search paradigm, non-saccadic[7] reaction times are recorded as subjects attempt to determine if a specific target symbol is present in an image containing various distractor symbols. By comparing reaction times across many tasks, Treisman has demonstrated that when certain visual structures are unique in the image they consistently "pop out" from other features. She argues that pop-outs are visual features or primitives of visual decomposition. Consistent with Beck's findings, Treisman's primitives include contrast, line orientation, and color.

However, Treisman (1982) has gone beyond Beck to demonstrate a relationship between features and conjunctions of features. She has shown that, regardless of the type of target feature and the number of distractors present in the image, the time to recognize a "pop-out" is constant, while the time to recognize conjunctions of features is proportional to the number of distractors present. For example, Treisman reports that a red "P" will pop out from a field of blue "P"s, and a red "K" will pop out from a field of red "O"s, but identifying a red "L" among a field of red "T"s requires a search time proportional to the number of red "T"s present. This demonstrates the dichotomy that seems to exist in the visual system. Individual features can be effortlessly recognized (Julesz, 1975) but combinations of features require additional processing.

Treisman and others (including: Julesz, 1962 and 1983) have proposed two visual processes to account for the data. According to Treisman, one visual process is preattentive, parallel, and feature based. The other is a serial focusing of attentive resources on combinations of features.

[7] Eye movement as a factor in focused attention is eliminated by limiting viewing times to 200 milliseconds or less.

The contrast between the parallel and serial resources is further demonstrated by another Treisman (1985) experiment where she found that portions of images can be camouflaged, or masked, preattentively by manipulating spatial structure. The camouflage is constructed by grouping objects on the basis of feature similarity to form a different level of detail or aggregation. The effect is to camouflage levels of perceptual detail. An example of this phenomenon is illustrated by the image recreated in Figure 2. When viewing that image in Treisman's paradigm, subjects consistently fail to report the existence of the "odd" items[8].

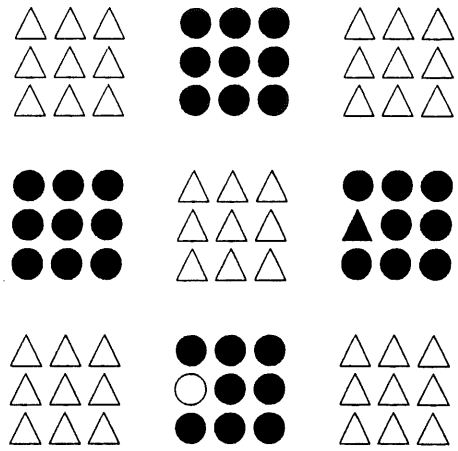

Figure 2:
A reproduction of a Treisman masking image

The camouflage is only temporary, however, because if allowed to view the image for longer periods of time, the subjects eventually locate the masked items. The amount of time required to unmask a camouflaged object is proportional to the time it would take for a serial search of the major groups (9 in this case) and not proportional to the total number of objects contained in the image as it is when the grouping is not present (there are 81 objects in this example). That is, even with the grouping, the camouflage effect is not comprehensive because the masked objects will popout if the viewer is allowed to focus attention on the proper level of detail (aggregation).

Treisman's masking phenomenon seems similar to the real-life phenomenon of visual "mistaken identity." Examples of mistaken identity include the embarrassing experience of mistaking a stranger for a close acquaintance and the experience many have had of approaching the wrong car in a parking lot, intent on driving away in it as if it were their own.

The commonality between Treisman's observations and cases of mistaken identity is the apparent variability in the amount of information used to match the stored experience with the

[8] The phenomenon may not manifest itself during a leisurely viewing of the image reproduced here. The outline circle is often noticed quickly in a nonchalant atmosphere, however, much to the chagrin of the author, dozens of leisurely viewings can take place before a viewer realizes that the image has two masked objects. Example: the author noticed the outlined circle within seconds of the initial exposure, but the solid triangle was only discovered much later.

current image. In each case, the viewer eventually notices that there is a difference between the observed image and the matched image, but initially that detail is overlooked[9].

One explanation for these phenomena is the existence and use, during recognition, of more than one level of knowledge, where a "level" of knowledge denotes the amount of detail learned about an object. Treisman's results and the examples suggest that the application of variable amounts or different types of knowledge involves the use of attention.

5.0 Some Approaches to Modeling the Evidence

ZBT builds on and extends some previous approaches to recognition. The following section presents a synopsis of the major modeling contributions to ZBT.

5.1 Classifying Images

One approach to image recognition is to view it as a pattern classification problem (Duda & Hart, 1973). In that paradigm, recognition is a search for functions capable of distinguishing one pattern from another.

Initial classification attempts made use of match functions in the form of templates (there are many examples but Selfridge & Neisser, 1960 is interesting because the authors compare their model to emerging neurophysiological data). Template matching is deceptively simple. First, an exact copy of the target structure is formed, then an automaton is directed to search through the entire image for a match. The search takes place in the following manner. If an "A" is the target structure, a template composed of the exact pixel pattern of the "A" form to be searched for is first generated. This pattern of pixels is then compared, bit by bit, to the first bits in the image. If unsuccessful, the search proceeds by moving the template over one pixel location and attempting the match again. This process continues across and down the entire image until the pattern is encountered or the search runs out of image to compare.

The obvious disadvantages of this technique are that search time increases with the size of the template, the size of the image, and the image resolution (this creates a significant paradox since higher resolution images are often required in order to distinguish essential image detail). Efficient search techniques (e.g. the Boyer & Moore search algorithm, 1977) and today's cheap, parallel processing hardware can largely negate these problems, but the real difficulties associated with matching techniques remain. They are not easily extended to handle new domains, especially domains including real-world problems such as those discussed earlier.

Extension of a template system is straightforward in a limited domain of perfect data, but in the real world it is exponentially difficult. Every acceptable rotation of every target symbol requires another template; every possible change in scale requires another template; every pixel difference, in every structural alternative, must be accounted for with another matching template. The use of tri-state logic (i.e., on, off, and don't care) can diminish the significance of individual pixel changes, but the fact that each structural alternative can be rotated and scaled still means that an exponential number of templates are required. Despite this, and possibly demonstrating the overall dearth of comprehensive recognition capability, template matching

[9] It's not pertinent to this discussion, but mistaken identities differ from the Treisman masking experiments in that Treisman's subjects are constrained by the viewing time. Mistaken identities appear to be more of a lazy recognition because viewing time is generally not constrained and yet the viewer chooses not to use everything he or she has learned about the target object, settling instead for a match with a different amount of detail.

was, until recently, the technique of choice for most OCR (optical character recognition) systems. Template matching is now being replaced by the technique of feature analysis.

Feature analysis springs from attempts by researchers to discover dependable mathematical classifiers, a classification technique based on segmentation functions (see Grunland, 1978). To date, no single segmentation classifier, nor group of functions, has been found capable of identifying complex objects such as a tank or a face. There has, however, been some success identifying lower level components such as contrast edges, vectors, and arcs (see Pavlidis, 1978 and Brady, 1982 for summaries).

Feature analysis capitalizes on these small successes by combining the ability to identify low-level features with the techniques of statistical decision theory (see Lewis, 1962 for comments and a brief summary). Labeled "statistical recognition" by some, the term "feature analysis" is used here to include the following steps:

1) Various aspects of the image are enhanced or de-emphasized. Examples include thresholding, noise removal, edge separation, and edge enhancement.
2) The image is decomposed, or segmented, into entities using various computational methods (see summaries of techniques by Rosenfield, 1978 and Brady, 1982).
3) Descriptors are used to characterize the decomposed entities in terms of features or attributes. Descriptors and the corresponding attributes they describe are usually selected for their invariant qualities over certain classes of image manipulation (see Hu, 1962 for a discussion of algebraic invariants).
4) The decomposed entities are classified, or grouped, by their feature descriptors (see Duda & Hart, 1973, and Fukanaga, 1972, for complete presentations of classification techniques).

This approach has the following advantages over template matching:

- Translational effects are negated by mapping image (raster) data onto a symbolic data base.
- Scaling and rotation can also be negated by applying certain classification techniques on the proper invariant descriptors (discriminate analysis is an example of such a technique).
- Learning and extensibility are possible.

5.2 Moment Analysis

One type of feature analysis, which ZBT builds on, employs moments as the invariant feature descriptors. (Hu, 1962, discusses the theoretical underpinnings of moment analysis, their invariant qualities, and the computation of moments of gray scale images. Alt, 1962, covers moment analysis on alphabetic characters and presents experimental data showing the effectiveness of various orders of moments. Hall et al., 1975, 1976, Wong et al., 1976, and Dudani et al., 1977 cover the applicability of moment invariants to different domains.)

The concept of a moment can be summarized as follows: the $(p+q)^{th}$ order moment on a continuous probability distribution function (pdf) f(x, y) is defined as the integral:

$$M_{pq} = \iint x^p y^q f(x,y)dxdy \qquad \text{where } p,q = 0,1,2, \ldots$$

Since our concern is scanned images, the formula can be rewritten as follows to reflect the fact that a raster matrix (i.e., the image format ZBT accepts as input) is a discrete version of a pdf:

$$M_{pq} = \sum x^p y^q$$

The equation appears formidable, but calculating these values for a raster matrix is not difficult, only tedious. In effect, computing a moment on a bitmap array means simply summing all the black (if white is ground) pixels in the pertinent portion of the matrix. If a computer is utilized and the computer is instructed in an algebraic computer language (such as FORTRAN, Pascal, or C), the calculations can be implemented by looping through the two dimensions that define the array and performing the operation indicated for that order of moment. For example, if the moment being calculated is the 0th, the program simply loops through the entries of the matrix adding 1 to a sum every time an "on" pixel is encountered. The other order moments are calculated similarly, although there are algorithmic shortcuts that can be implemented.

The meaning of a moment is not immediately obvious, therefore, a couple of examples will be covered to provide a feeling for what a moment computed on a raster array represents. Assuming that a single object or symbol is represented in a raster array (image), the first summation, or 0th order moment, is the count of pixels that constitutes the pattern or form of the object. That is, the 0th order moment represents the sum of pixels composing the object. For example, if the form in question is the "A" form of Object1, the 0th order moment is computed by counting every pixel that makes up the "A" form[10]. Calculating the 1st order moment identifies the center of gravity of the area summed by the 0th order and the 2nd order moment defines the moment of inertia. The latter is the theoretical tendency for the symbol to rotate about its center of gravity[11].

The actual interpretation of moments is less important than the following two significant points[12]:

- The moment sequence, M_{pq}, uniquely defines $f(x,y)$ and conversely, $f(x,y)$ is uniquely determined by M_{pq} (see Hu, 1962 for a presentation of algebraic invariants, a description of The Uniqueness Theorem, and the correspondence of moments to the raster domain).
- Normalizing and referencing the moment calculations to the center of gravity (i.e., forming the central moments) make the higher order moments invariant to various manipulations including: scale, translation, and rotation (see Hu, 1962 for the method of calculation, proofs of invariance, and a description of application to symbol recognition).

Therefore, the technique of moment analysis works in this manner. First, the system is trained on the domain of objects by exposing the system, one at a time, to the raster representations of

[10] For the reader not familiar with raster representations, this number depends on the resolution of the scan. A typical value, assuming a 12 point character and a scan resolution of 300 dots per inch, would be a value between 1000 and 2000.

[11] Assuming the presence of a gravitational field, if a weight is attached to the outer rim of a bicycle tire and the tire is suspended by the ends of its axle, the wheel will rotate until the weight lies at rest at the bottom of the circumference. In the same way, if a paper cutout equivalent to an object represented by a raster image is suspended by a frictionless device through its center of gravity it will rotate to equilibrium. The 2nd order moment predicts this tendency.

[12] These attributes do not hold for every pdf but do for the conditions under which ZBT works.

the symbols to be learned and the classes[13] that the symbols belong to. As per the previous description, working on each image, the system must isolate the object from the background (i.e., segment the image), calculate the moments of the isolated area, and store the moments with the associated class.

Later, after training, when exposed to a raster symbol, the system will follow the same procedure, however, in the absence of category information from the teacher, the system will look through its memory for a previous, similar experience by comparing the stored moments of each class with the newly calculated moments. A match indicates that the new experience is probably of the matched class.

Alt (1962) demonstrated the usefulness of this approach by showing that it is possible to distinguish the standard 35 textual characters (26 alphabetic and 9 numeric) independent of translation, scale, rotation, certain affine mappings (i.e., those preserving x and y, that is, squeezing and stretching), changes in proportion, bending, and "reasonable" amounts of random noise. Other successful applications include: handprinted characters (Casey, 1970), and interpreting medical x-rays (Hall, Crawford, & Roberts, 1975).

Despite the Uniqueness Theory moment analysis has limitations. For example, Lambert (1969) reported 95% accuracy distinguishing a set of printed characters but encountered greater difficulty when multiple fonts were learned. Hall, et al. (1976), Wong & Hall (1976), and Wong, Hall, & Rouge (1976) reported some success utilizing the technique to match optical scenes with radar images but the domains were very limited. And, Dudani (1977) had similar problems with another complex, real-world domain, aircraft identification. The limitations reported by these authors are of two sorts: difficulties attributable to the technique of feature (moment) analysis and problems caused directly by limitations in the moments themselves.

There are two weaknesses in the feature analysis technique: (a) segmentation difficulties, and (b) problems caused by structural modifications not invariant to the specified image alterations. Both difficulties should be anticipated. The Uniqueness Theorem prescribes that except for the specified invariant modifications (i.e., translation, scale, and rotation), different structural forms will have different moments associated with them. To understand the problem, consider a moment analysis system attempting to distinguish the forms of Object1 and Object2 (see Figure 1). Object2 is an incompletely formed version of Object1 that might result from sloppy printing or noise occluding the reading (scanning) process. The moments calculated for Object1 will not typically be the same as those calculated for Object2. Therefore, if the system has been trained on Object1 (i.e., exposed to the object and told that the respective class is A), then it is likely that an unlabeled occurrence of Object2 would not be associated with the same class as Object1. Similarly, if the system is then exposed to an unlabeled Object3 (i.e., after learning that Object1 and Object2 are both members of Class A), it is unlikely that the system will associate the new object with either Object1 or Object2. In other words, training a moment analysis system to allow errors of omission (beyond a minor amount, see Alt, 1962) requires training the system on every minor, yet acceptable, change. This can be, at best, time consuming and, at worst, an exponential problem.

A comparable difficulty exists if characters are not segmented consistently from one viewing to another or if segmentation does not present a complete character. The latter is illustrated by typical attempts to segment the letters "i" and "j". These are are seldom segmented as whole characters (i.e., dot and stroke together). This is because typical segmentation techniques make

[13] The class of the object can also be viewed as the value of the object or the response the system is supposed to use when the object is recognized. For example, when detecting an "A" form the system might respond by outputting the ASCII representation of an "A". This is a typical response of an OCR system.

use of low-level computations that make decomposition decisions on the basis of very local conditions. An example of such a process is a process called connectivity which simply isolates the areas of uniform intensity (e.g., all of the touching "on" pixels). Decomposition by connectivity will segment the dot of the "i" as one connected area and the main stroke of the character as another connected area. The problem is that if segmentation doesn't present the descriptor with the entire symbol, the two components must somehow be reassociated later. That puts a burden on the later stages of processing that was not intended to be part of feature analysis.

The nature of moments themselves create two other types of difficulties. An analysis of Table 1 (reproduced from Wong & Hall, 1978) reveals one type. That table contains the logarithms of the seven moments computed for three images - the original image and two possible transformations, one scaled and one rotated. Comparing the moments, one sees that the values do not match from one column to the next[14]. The variation is caused by the digital encoding of data. That is, a scaled or rotated image can vary from the original image in the number of and locations of pixels. The result is that, contrary to their advertised quality, moments do not necessarily match precisely from transformation to transformation, and, thus, a simple matching procedure is not possible. Moment analysis systems usually overcome this problem by employing correlation techniques to determine the closest match between new and stored experiences. (Moment correlation is not germane to this discussion because ZBT employs a different technique. However, the interested reader is referred to Wong & Hall, 1978 for a simple correlation procedure and Duda & Hart, 1973 for other approaches.)

Original Image	Scaled Image	Rotated Image
6.24993	6.22637	6.25346
17.18015	16.95439	17.27091
22.65516	23.53142	22.83652
22.91954	24.23687	23.13025
45.74918	48.34990	46.13627
31.83071	32.91619	32.06803
45.58951	48.34356	46.01707

Table 1
A comparison of the log values of invariant moments of three images - original, a version scaled by 2, and a rotated version (reproduced from Wong & Hall, 1978)

Because the invariants are not precisely invariant, the opposite problem also exists. That is, because the system must accept some slack in the moments, it is difficult to distinguish a variation between moments of similar objects from the natural but close variation between two dissimilar objects. What's more, the problem is compounded as the recognition domain increases because as additional characters are added to the database, more variations in characters are learned and the individual features of dissimilar characters can overlap. For example, consider the difficulty of distinguishing characters with simple distinctions, such as "6" from "b" and "I" from "1", across multiple fonts. The subtle distinctions between many versions of these characters begin to overlap and the discriminating categories begin to overlap. Therefore, the more the system learns, the more difficult recognition can become.

[14] These values were chosen as relatively indicative of the problem. The differences should not be taken literally since absolute values can vary from one domain to another depending on resolution.

In summary, moment analysis can perform usefully in certain domains but it does not appear to be universally adoptable. The basic limitation of moment analysis is that the technique places too much pressure on the moments to distinguish an object from every other one. Confusion between objects can result despite what appear to the human to be very simple differences between the two confused objects.

ZBT extends the demonstrated capabilities of moment analysis and overcomes its weaknesses by augmenting it with a second step that focuses on structural detail. In the first part of the process, ZBT maps experiences into memory clusters invariant to scale, translation, and rotation. This much of ZBT's operation is similar to other moment analysis systems except that a simple match process is utilized instead of a more complex correlation procedure. That is, ZBT compares the moments of the current experience with those in memory and any stored experiences possessing moments within a definable range[15] of the newly calculated moments are candidate matches. If a candidate is unique then the category of the match is reported. If the candidate is not unique or if there are labeling mismatches, as mentioned earlier, then the second part of the ZBT process is required to resolve the problems caused by the limitations of clustering by moments. In the second part of the process, ZBT focuses on other aspects of the current experience, looking for structural detail in the image which can resolve the situation. The basis for the second step is best understood after looking at some other contributions to ZBT.

5.4 Modeling Attention

Recent theorists trace their roots to Julesz (1962) and Neisser (1967) who proposed a two-mode visual system: preattentive and attentive. Neisser further suggested that the system is an analytic one utilizing parallel processes to decompose the image along the distinct dimensions of color, movement, and contour.

Building on Neisser's model, Treisman (1985) has suggested that the preattentive, parallel processes decompose the image into feature maps such that if a feature is unique to a map, it will "pop out". According to Treisman's model, unique features will be processed effortlessly and in less time than conjunctions of features that require a serial search through the feature maps by attentive processes. Crick (1984) characterized this attentive process as a searchlight-like selection process and proposed specific cell assemblies in the reticular complex of the thalamus that might represent features and conjunctions of features. In a related model, Koch and Ullman (1985) suggest that features are coalesced in a "saliency map" which tracks the highest frequencies of firing among all the various feature maps. A winner-take-all network then selects the highest of the high as the focus of attention.

ZBT has a common basis with these models in the neurophysiological and psychological data. Additionally, it proposes a specific mechanism that details the use of attention in the handling of novel experiences and the reassembly of the decomposed features. Further, ZBT proposes a specific hierarchical structure to account for the varying levels of recognition detail apparent in the visual system.

ZBT's hierarchy is similar to the hierarchical model of the visual system sketched out by Marr (1978, 1982). The initial step in Marr's model, computation of the primal sketch, relies on ordered stages of spatial primitives to describe different scales of spatial organization. In this way, Marr wished to account for varying levels of visual detail or he put it (Marr, 1978),

[15] Typically this range (slack value) is plus or minus 0.2% of the moment value. This range has been determined empirically for this scanning domain. As the resolution of the scanner or the size of the symbols to be recognized changes, the range must be adjusted proportionally.

"summarize lesser subparts of an object, leaving them unspecified until they are needed." Marr suggested the use of two types of spatial primitive: convolution and zero-crossing computation. The latter is the actual decomposition mechanism. The former is a method for isolating different levels of processing detail prior to decomposition. By convolving with different patterns, Marr believed he could control the level of organization detail exposed by decomposition

Marr did not go far enough, however. He neither mentioned how the ordering should proceed, nor did he specify how to choose, from the infinite number of possible convolutions, the single convolution that would provide the appropriate amount of detail for each particular situation. Marr also failed to distinguish attentive from preattentive processes in his model. This is less of a concern than the previous points, however, since the primal sketch computation seems very much preattentive and other parts of the model (i.e., 2½D and 2D computations) could be construed as serial.

Building on Marr's concepts, while remaining consistent with the neurophysiological and psychological data, ZBT proposes a specific use of decomposition and attention to enable ZBT to incrementally learn new experiences. ZBT's hierarchy is the result of combining a specific attentional mechanism with different learning mechanisms. The contribution to ZBT's learning mechanisms comes primarily from the machine learning literature, however, the use of attention in learning has received very little study there.

Something of an exception to that statement is the EPAM system (Feigenbaum & Simon, 1963, 1964), an early example of visual learning. Although not intended as a model of attention, EPAM is pertinent to this discussion because of its selective use of features.

EPAM was designed to model human performance on the nonsense syllable task made famous by Ebbinghaus (1913). There are two phases of operation. In the first phase, EPAM is presented with pairs of 3 character syllables until criterion is reached. In the second phase, EPAM is presented one syllable of a learned pair and it must determine the correct association. This would be a simple memorization task for a computer except that whole representations are not learned by EPAM. Instead, it adds features to its discrimination net on the basis of what is already there.

The features employed by EPAM are hand-coded to represent the domain of characters and fed to EPAM as input. This is comparable to an innate, feature-based decomposition by a visual system, except that the system's authors determine which features will be employed. The limitation of this feature-based system is that EPAM's knowledge of structural relationships is limited to knowledge of the serial relationships between characters in the learned syllables.

5.5 Learning Spatial Knowledge

A more comprehensive spatial knowledge is employed by an unnamed collection of programs assembled by Winston (1975) to learn structural concepts from scenes of actual and "near-miss" examples. The initial program, or step, in the process is a decomposition of the 3D scene based on Mahalaba's (1969) principles of vertex recognition. In the second step, a program by Guzman (1968) classifies regions by analyzing the intersections. Finally, a series of programs by Winston sort the classifications, group the objects, and look for structural differences between groups. The system employs concept descriptions which are described as hierarchies, or networks of nodes and arcs, where arcs are labeled with the nominal structural features it detects. By comparing the differences between networks, Winston's system learns which components comprise a concept (e.g., "arch").

The structural similarity between an arch and an "A" suggests that Winston's programs might be adaptable to real-world character recognition. It might be possible, for example, for his programs to learn that an "A" belongs to Class A in the same way it learns characteristics of an "arch". Further, since Winston is capable of generalizing structural variances (e.g., the concept of an "arch" is independent of the form of the top portion of the "arch" structure), his programs might also be capable of generalizing on structural variances in "A"s (e.g., an "A" form belongs to Class A regardless of whether the middle brace joins the right side or not). To explore this possibility, consider the consequences of Winston's programs learning an "A" in light of the problems of real-world recognition.

The initial stages of the programs (Mahalaba's decomposition and Guzman's classification) map the 3D input data onto abstract data structures. This process negates the effects of translation, but unfortunately the representations rely on lengths of line sections to resolve z-axis (i.e., the 3rd dimension) positions. That makes the system sensitive to scale. The problem could be eliminated for purposes of ZBT's problem domain (since ZBT's domain is only 2D) by eliminating the use of nominal terms related to size. Therefore, with a little modification the system can be made impervious to translation and scale. Rotation is a more difficult matter, however.

Winston's system learns that an "arch" is composed of a brick "support-by" two bricks that must not "marry". In ZBT's real-world, recognition domain, an "A" cannot be represented comparably for a number of reasons. First, concepts such as "supported-by," "above," "left-of," and other references to direction or orientation are not rotation invariant. The problem is easily understood by considering the difficulty of recognizing an arch that has undergone a 45 degree rotation. If the memory representation that will be employed to match the visual experience describes an "arch" with orientation dependent terms such as those above, there is a match problem. In a rotated arch, the two component bricks really no longer "support" the third and the top brick is not necessarily "above" the left side brick.

Another representational problem is that concepts such as "marry", that describe a meeting between two objects, do not contain the expressive power required to distinguish a structure, such as a Class A form, from other possibilities. For example, "marry" may be satisfactory to describe the single type of contact in Winston's domain, but it does not describe the two types of contact present in an "A". That is, if we label, for purposes of this discussion, the three component parts of the "A" as the left side, right side, and brace, then the two side components contact each other at their respective end-points and the end-points of the brace contact each side at some midpoint location on the sides. The term "marry" could be used to represent arch or triangle, but could not distinguish a triangle from an "A".

Alternative structural concepts could be proposed to solve this representation problem. For example, two types of "marry" could be proposed. The first one, called "end-to-end", could be used to describe the relationship between the two sides of the "A" where their endpoints meet at the top. However, describing the brace-to-side relationship is more difficult. A second type of marry, called "end-to-midsection", could be postulated, but that does not describe which side component is involved.

The problem is apparent if we try to represent the form in Figure 3 using the "end-to-midsection" relation (assuming no other concepts are added to the representation). To represent this form, it is useful to distinguish the two "ends" of each component and each "end's" relationship with its specific partner[16]. When two sibling components are related, as they are in

[16] Components at the same level of detail as the three components composing an "A" are called siblings throughout this discussion.

the case of this form or the forms in Class A, each component can be in a different relationship with the other component. That is, a "midsection" of the side of an "A" can contact an "end" of the brace (a characteristic of members of the A Class) or a "midsection" of the brace can contact an "end" of a side (a characteristic of the form in Figure 3). As will be discussed later, ZBT's solution is to describe both aspects of the relation: the what and where.

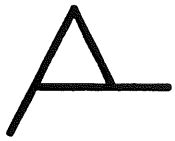

Figure 3:
An alternative form using "end-to-face"

There are other problems adopting Winston's approach to a real-world recognition domain. In Winston's system a human must specify the concepts (e.g., brick) which are in turn used internally to represent other learned concepts (e.g., arch). This is a twofold problem. First, human intervention creates a data entry problem. Second, the programs must be made aware of every potential concept, even if those concepts are otherwise unknown. This limits the number of objects that can be learned. For example, the programs must know about bricks even though bricks themselves might be unimportant except as components of the concept. ZBT represents objects hierarchically. At each level of detail it uses amorphous, unnamed entities (called blobs for purposes of discussion) to fill structural spots. Blobs go unnamed[17] until the system receives feedback to indicate that a label should be associated.

Another potential difficulty associated with Winston's system and not discussed by him, is that of extensibility. A recognition in Winston's system requires a complete perusal of the data base. Like the template matching systems discussed earlier, there are an exponential number of variations of even a single character in the real-world problem domain. ZBT employs a multi-level, hierarchical indexing scheme to alleviate this problem. Winston doesn't discuss this problem or the use of indices.

5.6 Combining Two Learning Paradigms: Learning from Examples and Conceptual Clustering

Contrasted with Winston's learning from examples system are conceptual clustering systems that do not utilize feedback - at least not overt feedback (i.e., obvious to the viewer). In the learning from examples paradigm the system receives teacher or environmental feedback designating the class of the viewed subject. A typical task for such a system is to associate the provided class information with the labeled object. In other words, the system attempts to group objects on the basis of the information provided by the teacher.

[17] Blobs are unnamed except for purposes of user tracking. That means that the program will generate random names and assign them to blobs so that users can distinguish one from another, but internally this is not necessary because the program only needs to identify a blob by its location in memory. As we'll see the location in memory is defined by the moments of the blob.

A different approach to learning is taken by conceptual clustering systems (Michalski, 1980). The conceptual clustering task is to group attributes of objects without environmental feedback. This is a similarity based learning since the system attempts to maximize intracluster similarities and intercluster differences by internal standards which are generally based on some perceived similarity in the attributes (features) of the objects.

Another learning approach, as yet undocumented, combines the two paradigms of conceptual clustering and learning from examples. During part of its existence such a system must operate in a supervised mode, receiving category information from its teacher, and learning how to respond to its environment. At other times, it must operate in the absence of feedback but nonetheless storing and classifying its experiences for retrieval at a later time.

Combining the two paradigms is not just a matter of adding the two processing components. There is a problem of conflict created by the combination that this type of system must contend with. The problem is that environmental feedback may, at any time, negate the groupings created by conceptual clustering. For example, it is not unreasonable to hypothesize that this type of learning system, in the absence of feedback and based on structural similarities and differences, might group Object1 with (a comparably shaped) "triangle" form and group Object2 separately. Having formed that grouping, environmental feedback would likely indicate that Object1 and Object2 should have really been grouped together and that triangles are the separate class. The result would be a necessary reorganization of memory.

This illustrates the potential contention between conceptual clustering and learning from examples. As a learning from examples system, the clustering results can be partially or completely negated by environmental feedback. The reason for the contention is that the similarity based clustering engine is grouping the objects on the basis of attributes not necessarily important to the environment.

ZBT combines conceptual clustering with learning from examples. Conceptual clustering, the parallel, preattentive portion of the two-step process, decomposes the image and groups the experiences invariant to translation, scale, and rotation. The grouping allows ZBT to store experiences for future reference even when environmental feedback is not provided.

Illustrative of the usefulness of the clustering process is a test performed on ZBT that showed the program capable of successfully distinguishing different fonts without labeling information. The test was conducted in this way. First, ZBT was exposed, one at a time, to the unlabeled characters of two different fonts and two transformed images of each of the font characters (one scaled and one rotated). Then ZBT was given the same untransformed characters and told what classes they belonged to. The internal groups formed by ZBT in the absence of the labeling information was completely consistent with the groups specified by the labels[18]. In other words, the attributes utilized by ZBT adequately clustered the characters and their transformed versions according to the environment's choices despite the fact that category information was not initially provided.

There are, however, other situations in which the clustering mechanisms do not adequately group the data. Quinlan (1986) observed that in a clustering system where internal metrics are employed to judge the quality of the fit there is a simple dichotomy relative to the ability of the attributes to segregate the data: attributes can adequately or inadequately distinguish objects within the object space.

[18] This is to be expected since it simply confirms the findings of a number of authors (including: Alt, 1962, and Lambert, 1969).

Contrasted with clustering systems are models such as ZBT that combine the two paradigms. When the two paradigms are combined the feedback creates four cases which ZBT must handle (the matrix of Table 2 summarizes):

Case 1) Same category, same attributes. That is, the reminded experience has the same attributes and was assigned the same label as the new perception. The labeling is consistent with the attributes.

Case 2) Different categories, different attributes. The category assigned to the remind differs from that provided with the new experience, but there is no conflict since the attributes indicate there should be a difference. The labeling is again consistent with the attributes.

Case 3) Same category, different attributes. The new experience is perceived to have features different from the learned experience, but the teacher has indicated that they are really the same category. From ZBT's perspective the attributes are ambiguous because two categories exist with the same label; the attributes ambiguously distinguish the meaning provided by the environment.

Case 4) Different categories, same attributes. The new experience and the reminded experience are perceived to have the same features (from ZBT's point of view), but have been labeled differently. ZBT views this as a case of ambiguous labels because one experience has two meanings.

Cases 1 and 2 correspond to Quinlan's definition of "adequate" since the attributes adequately segregate the objects. Cases 3 and 4 are examples of inadequate attributes, or conflicts between the grouping preferred by the conceptual clustering processes and the grouping preferred by the environment.

Each of these conflicts could require a reorganization of memory, however ZBT attempts to resolve them by attending to previously undetected detail in the image, looking for some aspect that might distinguish the labeling conflict. By serially investigating the spatial relationships that define structures and recursing on components, ZBT's memory organization is less of a reorganization of the existing clusters and more of a hierarchical addition describing the detail of the object thus far attended to. The conflicts and how ZBT resolves them are detailed in the next section.

	SAME ATTRIBUTES	DIFFERENT ATTRIBUTES
SAME CATEGORY	CONSISTENT	AMBIGUOUS ATTRIBUTES
DIFFERENT CATEGORIES	AMBIGUOUS LABELING	CONSISTENT

Table 2:
Comparing the retrieved memory experience (remind) and new experience by category and attributes.

6.0 ZBT: Learning Levels of Structural Detail by Recursive Direction of Attention

The following section describes how ZBT handles the four cases by describing its operation on a series of images. The raster forms contained within the images were chosen to conform to certain processing limitations, which are discussed in Appendix II, and chosen to be representative of a kind of problem potentially encountered in a real-world recognition task.

6.1 Description of the Example Images

The recognition domain in these examples has been chosen from the standard alphabet of English/Latin characters and consists of the forms illustrated in Figure 1. Characters were chosen over other less-known symbols because they are more easily described in text and more familiar to the reader. Additionally, given the precedent of character recognition as an accepted research task, alphabetic characters serve as a base-level performance standard and departure point for ZBT's behavior. The latter is especially important because it means that ZBT can build on the performance of previous moment analysis systems by adopting that approach as part of its structure[19].

ZBT's handling of the four cases is illustrated with the following sequence of images:

Illustrating Cases 1 and 2:
Image 1) Object1, labeled Class A
Image 2) A translated and scaled Object1, unlabeled

Illustrating Case 3):
Image 3) Object2, labeled Class A
Image 4) Object1, labeled Class A
Image 5) Object3, unlabeled

Illustrating Case 4):
Image 6) Object4, labeled Class Y

An example of ZBT's runtime output is included in Appendix I.

6.2 System Operation

ZBT can be summarized with the following functional flow:

1) In parallel, decompose the current attentive area of the image into component blob(s)
2) In parallel, compute invariants of the decomposed blob(s).
3) In parallel, reference memory with the invariants of the blob(s) for similar experiences (reminds).
4) If the reminds are unique, the recognition is complete
else, (if there are two or more component blobs) serially investigate spatial relations of the component blobs and compare the relationships with those of the reminds.
5) If the spatial relationships of the component blobs and those of the reminds match, and there is no label conflict, the recognition is complete
else, recurse.

[19] This was demonstrated in tests previously described.

The following discussion elaborates this brief description.

6.2.1 Cases 1 & 2: Attributes Are Adequate (Invariants Handle the Scale and Translation Problems)

ZBT's operation on the first two images illustrates high level decomposition and the usefulness of the invariant clustering. The detailed explanation of that operation will be presented, but first as a guide to what follows a summary is presented in outline form.

The first image consists of Object1, a raster representation of a well-formed "A", with an assigned Class of A. Encountering the first image, ZBT will:

1) decompose, or segment, the image into a single blob form (i.e., isolate the object form within the raster image).
2) compute the invariants (moments) of the component blob.
3) use the invariants to index into memory for comparable experiences.
4) find no similar experiences.
5) store the new experience in memory.

The second image also contains an Object1 form with an assigned Class of A, but this time the raster representation of the form has been physically relocated in the image (translated) and diminished in size (scaled) (see Figure 4). The result is an image of an "A" which is about 9% smaller than the original and centered at a higher location. Encountering this image, ZBT will:

6) follow steps 1-3 as above.
9) be reminded of the "A" experience encountered in Image 1.

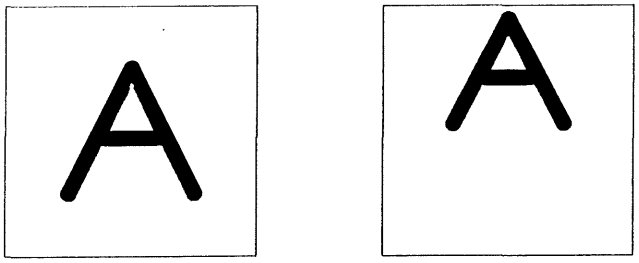

Figure 4:
Representation of the first two images presented to ZBT. Image 2 is a copy of Image 1 translated and scaled.

6.2.1.1 Image Decomposition and Invariant Calculation

An image is composed of a raster form in some portion of the visual field. ZBT begins by identifying that form. It does that by decomposing the image to isolate the form from the rest of the field. The result is the outline form, or most abstract level of the structure.

Recalling the television analogy presented earlier, the first decomposition is comparable to moving away from the television screen until image detail (such as the number of pixels

composing the symbol) is hidden and only shapes of information can be discerned. Therefore, the first decomposition of the first image isolates the "A" form of Object1[20].

The decomposition processes by which the "A" form is isolated are low-level. The mechanisms do not have knowledge of the high-level "A" form. Unlike decomposition schemes which attempt to segment an image into human recognizable entities (i.e., the types of mathematical classification systems mentioned earlier), ZBT's mechanisms decompose an image into amorphous, unlabeled entities, generically called "blobs[21]." ZBT's blobs are isolated within the image by primitive decomposition techniques analogous to Julesz's (1983) textons of density, width, and length and consistent with other data including that from Hubel & Wiesel (1962) and Treisman (1982)[22].

As part of the decomposition process, ZBT computes the moments of the segmented blob. (The technique of calculating moments was described previously. For greater detail see Alt, 1962 and Wong & Hall, 1978, who present very good descriptions of moment calculations.) The moments are used as indices to both store and retrieve experiences in memory. Since the moments are feature based, invariants cluster blobs in memory according to feature similarity. This clustering allows ZBT, in the absence of feedback, to quickly reference memory for previously experienced, comparable decompositions. The following discussion describes how grouping and retrieval take place.

6.2.1.2 Clustering of Experiences and Reminding

Computationally the moment invariants are treated as indices[23], clustering the related experiences within the memory space according to invariant similarities in structure as defined by the moments themselves. This allows ZBT to reference memory without searching the entire contents. To illustrate, consider a hypothetical memory composed of two invariants indexing two learned experiences labeled Class E and Class F (see Figure 5). The two indices distinguish these experiences in the following manner. If a new blob experience results in a value of 12.5 for Invariant1, then the 12.5 track in that group will be activated. Since this track connects both the "E" and "F" experiences, the two learned experiences are not distinguished. However, if the new experience also has a value of 3.2 associated with Invariant2, then ZBT will only be reminded of the "E" experience since only that experience has those two indices uniquely in common.

---

[20] The results of the second decomposition will be covered later when ZBT finds it necessary to zoom on the greater detail of the three components that constitute the highest level form it has just isolated.

[21] The term "blob" was used because it is meant to represent a possibly meaningless, amorphous visual entity. An example of a meaningless entity is the grouping experienced by most viewers of the many "masking" examples presented by Treisman. The grouping that masks the lower-level objects is label-less. That is, the masking structures have not and need not be assigned a cognitive meaning. In the same way, an aggregation of features could be an apple, or it could be a feature of the apple which is salient to recognizing the apple but go unlabeled to the viewer.

[22] The actual mechanisms of decomposition are de-emphasized in the ZBT model. While the model assumes a decomposition consistent with the experimental results of Julesz, there is no evidence to support any specific decomposition mechanisms for those primitives. Therefore, the computational details of ZBT's decomposition mechanisms are included in Appendix II for the interested reader but are not discussed in the main body.

[23] Thus, moment, invariant, and index are virtually synonymous terms. The exception is that not all indices are invariants. Another type of index, the label index, is introduced later. When the term "index" is used it refers to invariant indices. When there is a possibility of confusion label indices will be identified explicitly.

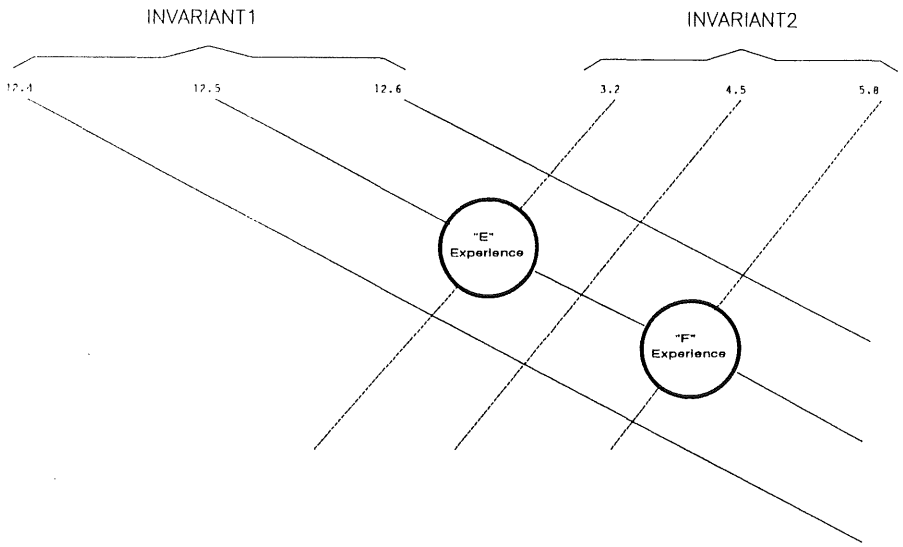

Figure 5:
Example memory illustrating the clustering of memory experiences by invariant indices.

In general terms, an index is one of the computed moments and consists of a group of parallel tracks pointing into memory. Each track represents one value of the many that that particular index can have. A track, therefore, points to the locations that have that particular value for that particular index.

When referencing memory, the invariant values activate one track within each of the indices. The intersection within memory of the values of the different indices is the memory location containing the stored experiences uniquely possessing those specific moments. If the invariants segregate the experiences completely, only one experience will be identified by each collection of indices. That is, an experience composed of unique features (invariant to transformations of translation, scale, and rotation) will pop out in the fashion Treisman (1985) described.

In the present example, after the invariants of the top level Object1 blob are computed, memory is referenced. If a similar experience (i.e. one with feature characteristics comparable to the top-level Object1 blob) had been previously experienced, the indices would point to the knowledge of that experience, but since this is the first experience, no comparable experiences exist (i.e., there are no experiences with the same invariants) and ZBT is not reminded of anything. Therefore, ZBT records the experience of the highest level blob of Object1 at the location in memory where all of the values of the blob's invariants intersect. ZBT also creates a label index (i.e., an index equal to the provided category label), points it at the same memory location, and creates a return pointer from the memory location to the index. This index differs from the invariant indices described earlier in that its value is the label itself and not a computed moment. The relevance of this index and the double ended pointer will be discussed later.

Figure 6 summarizes the results of the first Object1 experience with a diagram of memory contents. Diagrammatically, a node (circle) represents a stored experience. A stored experience represents an abstraction of the original raster experience containing pointers which define a blob's relationships with its relatives. Stored experiences do not contain verbatim image data such as the original raster form which defined the blob.

Pointers are represented diagrammatically by arrows. There are four types of pointers used in ZBT. An experience is pointed to and points back to the label assigned by the environment. Bold font arrows represent label indices. Vertical pointers, represented in the diagrams with normal font arrows, indicate father/son relations (i.e., the relationships between different levels of detail). Horizontal pointers, also indicated with normal font arrows, represent sibling relationships or the relationships between components at the same level of detail. The significance of horizontal pointers will be illustrated later when ZBT zooms for detail.

The three pointers discussed so far are usually double ended pointers. That means that these pointers can be traversed in both directions. For example, if provided the label of Class A by the environment, ZBT can identify all of the memory locations and thus all of the experiences associated with that category. In the same way, once ZBT has been reminded of a previous experience it can identify the associated label(s) by following the pointer(s) up the hierarchy.

The fourth type of pointer is the single ended pointers used in Figure 6 to represent the invariants that point to the experience (for clarity, the indices have been omitted from later diagrams). In actuality, the indices do not point at the experience but point at the memory location that holds the experience. Thus, a memory location potentially represents a cluster of stored experiences where a cluster can consist of one or more abstracted experiences. Clusters (or memory locations) are indicated by dashed ellipses, except that, for clarity, memory locations are typically not distinguished from the experiences they hold unless it is important to do so.

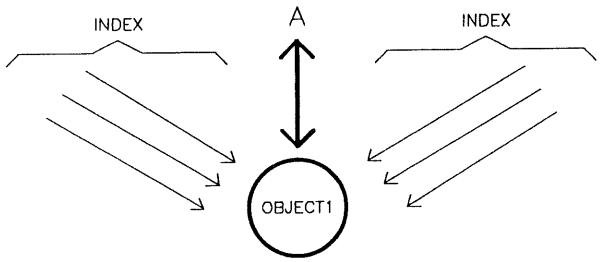

Figure 6:
Memory contents after the first Object1 experience

ZBT has now completed processing on Image 1 and begins on Image 2. It's encounter with the second image is handled in much the same way as the first image. The second image is decomposed to reveal the highest level blob of the scaled and translated Object1. Invariants of the blob are computed and memory is referenced. The invariants of the new experience (i.e., the scaled and translated Object1) are the same as the original[24] (Hu,1962 and Alt 1962), therefore, when referencing memory ZBT is reminded of the first Object1 experience.

ZBT has now resolved the second image. That is, it has matched the present experience with a known experience. If the present image was not accompanied by a label, ZBT would follow the double ended pointer from the remind (Object1 experience) to the associated label (Class A),

[24] The invariants are not necessarily equal, but the matching algorithms allow for the specified amount of slack in the manner described earlier.

report Class A as the likely category of the present experience, and terminate with memory unchanged. On the other hand, if a label is provided with an image, ZBT must take other actions.

As previously mentioned, conforming to the category information provided by the environment causes a potential problem for ZBT. In the absence of feedback, ZBT could assume that the clustering mechanisms will group "A"s with "A"s, "Y"s with "Y"s, etc, but by conforming to the environment's classifications ZBT must assume that its clusters will not always be adequate. The relationship between Object1, Object2, and Object3, as explained earlier, is an example of this problem.

The possibility of a conflict forces ZBT to check for agreement between the labels of the respective experiences. Thus, after memory has been referenced, ZBT compares the label of the new experience (if provided), with the label associated with the reminding (if any). A match confirms the original clustering and a mismatch indicates an inadequacy of the clustering attributes. As previously discussed, there are two mismatch situations. They are reviewed again here prior to explaining how ZBT handles each one of them:

Case 3: The current experience and the matched experience possess the same label, but appear different to ZBT (i.e., have different moments). For example, two seemingly different experiences, such as Object1 and Object2, are accompanied by the same label (Class A).

Case 4: The current experience and the matched experience are labeled differently, but appear the same (i.e., have the same moments). An example, a common problem for many feature analysis programs but unusable here because of implementation constraints (see Appendix II), is the difficulty of distinguishing a "6" from a "b" across many fonts. Since ZBT is not capable of handling this example, another one will be used to illustrate Case 4. That one, a somewhat artificially induced example, utilizes Object4 (labeled Class Y). The artificial inducement is a minor modification to ZBT's moment comparison routine to allow the moments of Object4 to be accepted as equivalent to the Object1 experience[25].

In both mismatch situations, the invariant measures of similarity at the highest level of the objects are unsatisfactory to group them in the classes the environment would prefer. ZBT handles both situations almost identically. It looks for another level of detail which might distinguish the experiences.

The following discussion utilizes Images 3, 4, and 5 to illustrate how Case 3 is handled and then, Image 6 to illustrate how Case 3 differs from that of Case 4. In each section, as a guide to the detailed operation that is discussed there, ZBT's operation is summarized in outline form. Additionally, because ZBT's behavior builds on the knowledge accumulated during its experiences with Images 1 and 2, ZBT's operation on the first image is recalled for comparison before outlining its operation on Images 3, 4, and 5.

[25] At first this appears to be an artificial exercise, but in reality the nature of invariant moments, as previously discussed, dictate that this is a very real situation. The nature of the minor modification to the matching routines shows just how real it is (see later footnote).

6.2.2 Case 3: Same Class, Different Attributes

As previously described, encountering Image 1 (an image of Object1 labeled Class A), ZBT did the following:

1) Decomposed the image into a single blob form.
2) Computed the invariants (moments) of the component blob.
3) Used the invariants to index into memory for comparable experiences.
4) Found no similar experiences.
5) Stored the new experience in memory and terminated.

Encountering Image 3 (an image of Object2 labeled Class A), ZBT will:

6) repeat steps 1-5 on the new image.
11) discover that the label of the new experience is the same as a previous experience.
12) (because the stored experience possessing the same label is not a remind of the current experience), decompose the current attended area (a single blob) into three component blobs.
13) index memory with the three component blobs.
14) find no similar experiences in memory.
15) serially attend to each component and its spatial relationship to its siblings.
16) Store the new experiences in memory.

Encountering Image 4 (a repeat image of Object1 labeled Class A), ZBT will:

16) repeat steps 1-3 on the new image.
17) be reminded of the top-level Object1 experience.
18) decompose the current attended area into three component blobs.
19) index memory with the three blobs.
20) be reminded of the three decomposed blobs of Object2.
21) serially attend to each component and compare sibling relationships to those of the experiences matched in memory.
22) find a difference in a spatial relationship and take steps based on the difference it finds.

ZBT's operation on Image 5 is comparable to what it does with Image 4, but since Image 5 will not be accompanied by a label the results are different. This is explained after ZBT's zooming behavior is illustrated with Images 3 and 4.

6.2.2.1 The Object2 Experience

ZBT proceeds on Image 3 in the same way it operated on Image 1 (an image of Object1). Decomposition reveals the form that defines the highest level of Object2. Memory is referenced, but there are no reminds of this experience (because the moments of Object2 differ from those of Object1). ZBT stores the new experience in a unique memory location corresponding to the invariants calculated. Next, ZBT uses the label provided with the current image and references memory looking for experiences possessing the same label. ZBT discovers that the top-level Object1 experience was indexed by a Class A label.

Since ZBT was not reminded of the stored experience that possesses the label identical with that of the current experience, ZBT has encountered an ambiguity of attributes. That is, the attributes indicate that two different perceptions have been encountered, but the experiences have been assigned the same category. That means that the features (attributes) appear to disagree with the provided label. ZBT must now look for something to differentiate the new experience from the stored experience. It begins by decomposing the top-level Object2 blob which it is currently viewing.

Decomposition of the current blob (i.e., the second decomposition of the image, see Appendix II for computational details) reveals three blobs (see Figure 7). The three blobs correspond to the strokes which compose the original, top-level blob. ZBT orders the component blobs by contrast for serial investigation (i.e. the one possessing the lowest contrast correlation, either increasing or decreasing, will be investigated first), computes the invariants associated with each blob, and simultaneously references memory with the invariants of the three blobs.

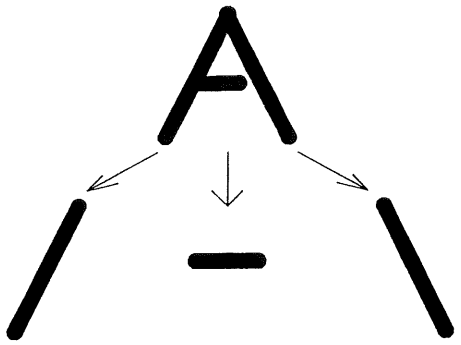

Figure 7:
The second decomposition of Object2.

There are no reminds of the three components of Object2, therefore, ZBT records the new experiences of the three constituent blobs. It does this by placing them in memory as separate experiences according to their respective invariants (see Figure 8), connecting them with two-way pointers to the higher level blob of which they are components, and interconnecting them with two-way, horizontal pointers to indicate their sibling relationships.

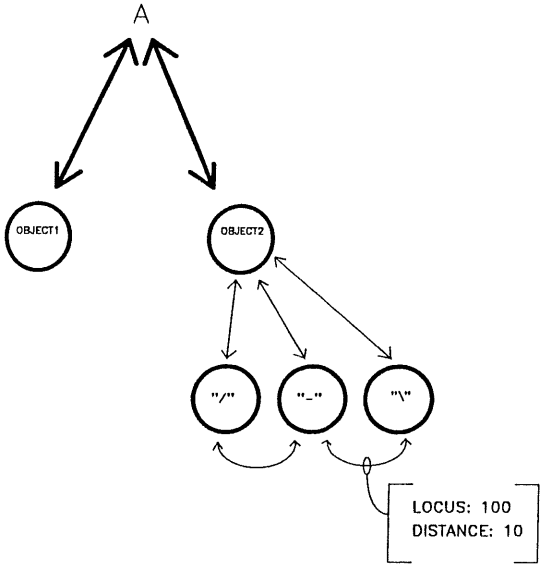

Figure 8:
Memory contents after the Object2 experience. Horizontal relationships are indicated, but only the one spatial relationship discussed in the text is shown.

ZBT then investigates the relationship each component has with each of its siblings. While many possible structural relationships are possible, ZBT has had success in this domain by comparing simple proximal associations between sibling blobs. ZBT's proximal associations consist of the following information describing each relationship a blob has with a sibling:

SIBLING: The sibling (identified by memory location) to which this relationship pertains.
LOCUS: The point of focus of the relationship, identified as the spot on the blob closest to the pertinent sibling and normalized as a percentage of the length of the major axis of the blob.
DISTANCE: The distance between the locus and the closest point on the sibling (again, normalized).

The actual method of computing these values is not critical to ZBT's operation, but the details are included in Appendix III.

In the present example, ZBT records for each component blob two relationships with each of its two siblings for a total of six recorded relationships. However, in order to avoid confusion, the following discussion and the associated diagrams focus on only the most pertinent of the six relationships, the one between the right end of the brace and the right side. As part of the information associated with the brace of Object2, ZBT records the following for the brace component (see Figure 8):

SIBLING: right side[26]
LOCUS: 100
DISTANCE: 10

These data reflect the fact that the end of the brace (i.e., a location 100% of the way from the reference end) is close to but does not contact the right side. The distance separating the two is the span between the loci of the blob and its sibling, computed as a percent of the length of the blob. In this case, that distance is 10.

At this point, ZBT has no comparable structural information for Object1. Thus, it can go no further in its attempt to identify the difference between Object1 and Object2.

6.2.2.2 The Second Object1 Experience

ZBT's situation after its encounter with Image 3 can be summarized this way. It has knowledge of the top-level Object1 experience. It also has knowledge of the top-level Object2 experience and has zoomed to the spatial relationships that define the second level of detail within Object2. However, while processing the previous image, ZBT was unable to compare the newly decomposed, second level detail of Object2 with that of Object1 because it lacked the corresponding second level information of Object1. The reason is that while processing the first images of Object1, its attention was not drawn to the lower level detail, thus it only recorded the top-level of Object1. Now that the need for greater detail of Object1 has arisen, another Object1 experience (the next image in the sequence) will allow it to collect the information necessary to isolate the differences between the two objects.

Encountering Image 4 (a second image of Object1 labeled Class A), ZBT isolates the highest-level of the Object1 form, references memory, and is reminded of the previous Object1 experience. It then finds that the reminded experience shares its label with another experience

[26] As a reminder, the terms brace and side are used in this explanation as a matter of convenience for the reader. In the program, true to the definition of a blob, those entities are not labeled except by memory location.

and that the ambiguity between the two experiences hasn't been resolved yet. Therefore, ZBT "zooms" on Image 4 in another attempt to resolve the confusion.

The highest level blob of Object1 is decomposed (revealing three constituent blobs), invariants of the three blobs are calculated, and memory is referenced for experiences comparable to the three blobs of level 2. ZBT detects three reminds corresponding to a match with each of the three constituent blobs[27].

This illustrates the multi-level indexing which allows ZBT to detect candidate recognitions anywhere in memory in a single reference. A memory reference is a one step process whereby any experiences stored at the intersection of the converging indices is a reminding of a similar previous experience. Although not apparent in this example, the fact that ZBT matched the three blobs composing the second level of detail of the two objects with the one step process is significant. The significance is illustrated by an example. Consider the situation after ZBT has experienced many different forms of Class A objects from many different fonts. At the second level of detail, all of these forms can vary considerably. Each of the three characteristic strokes can vary by thickness, length, curvature, etc.. The result is that within the second level of Class A, a large number of candidates could exist. Searching for the correct matches at that level could be an exponential problem except that ZBT's multi-level indexing scheme allows it to select the most likely candidate blobs at any level in a single step, regardless of the level of decomposition of the current object.

Now that ZBT has been reminded of the second level of the Object2 experience, it must verify that this object is the conflicting one and compare sibling relationships to complete the match. It begins by looking from the reminds up the hierarchy (using the double-ended pointers) for the label associated with the reminds. When it finds that the reminds have the top-level of Object2 as an ancestor and that Object2's label is shared with Object1, it knows it has matched both levels of the two experiences. However, since no difference has yet to be detected, ZBT must look further in the image.

Successive decompositions have not isolated the differences between the two similarly labeled experiences, therefore, as it did on the previous image, ZBT now serially inspects the current decomposition (i.e., the second decomposition of Object1) for relationships which structurally define the higher level. It discovers the following relationship between the brace and the right leg:

SIBLING: right leg
LOCUS: 100
DISTANCE: 0

Comparing the spatial relationships of the reminds with those of the current blobs, ZBT finds a difference in the distance value (see Figure 9). ZBT has now detected the break in the "A" which distinguishes Object1 from Object2.

[27] This example assumes that the three component blobs of Object1 and Object2 are equivalent enough that one set will cause the reminding of the entire other set. As previously discussed, this may or may not be the case; one or more component blobs may not match. Therefore, ZBT actually employs a more complex matching technique when only a subset of the component set cause of reminding. The more complex situation has been avoided because it complicates the explanation and because in practice the simple case does occur.

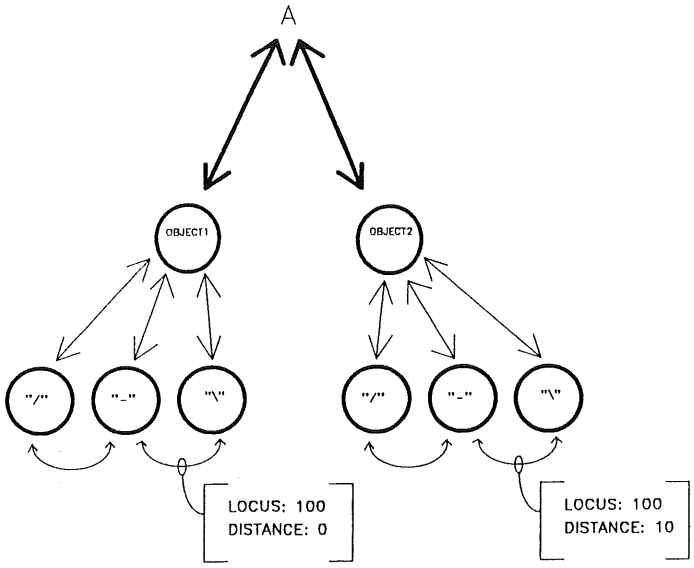

Figure 9:
Memory contents after the second Object1 experience and prior to coalescing the two experiences.

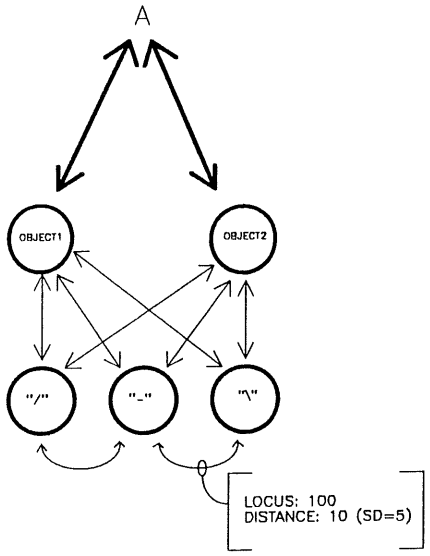

Figure 10:
Memory contents after coalescing the Object1 and Object2 experiences.

Knowing (based on feedback) that it is supposed to act as if these structures are really the same, ZBT coalesces the two knowledge structures by recording the detected difference as a cumulative statistical value. That is, the two data structures are merged into one structure where the detected difference is represented as the maximum DISTANCE experienced and the standard deviation (SD) of the current DISTANCE (taking the old value as the average). Thus,

the second Object1 experience results in the following values for the critical relationship (see Figure 10):

SIBLING: right side
LOCUS: 100
DISTANCE: 10 (sd = 5)

6.2.2.3 The Object3 Experience (Handling Structural Variations)

The next image in the sequence, Image 5, contains an unlabeled Object3 (i.e., a form identical to Object2 except containing a larger break between the brace and right side). Since this image is not accompanied by a label, ZBT must try to determine the class of the object it contains.

Processing begins, as before, on the top-level which causes no remindings because its moments differ from those of Object1 and Object2. The second decomposition reveals three blobs that cause a reminding of the coalesced second level knowledge of Object1 and Object2. ZBT then checks the spatial relationships of the three constituent blobs and finds that there is a close match. Comparing sibling relationships, ZBT detects that the DISTANCE value of the pertinent relationship is different from the corresponding stored value. ZBT then compares the new DISTANCE with the stored value to see if it exceeds an allowable measure. It does this by summing the stored DISTANCE with the associated SD value. There are two possibilities ZBT must deal with.

If the DISTANCE value of the current experience exceeds the sum (i.e., the break exceeds 15), ZBT will record the present experience in memory as a new experience and report that there is no known category for the present object[28].

On the other hand, if the DISTANCE value is less than or equal to the summed value, ZBT will accept the current experience as a member of Class A. In this case, memory will be updated to reflect a new maximum value and associated standard deviation for the pertinent relationship. Consequently, the class of the remind is reported as the likely category of the current experience. ZBT determines the class by progressing up the hierarchy from the three matched blobs until it finds the associated label.

6.2.3 Case 4: Different Class, Same Attributes

Image 6, an image of Object4, helps to illustrate ZBT's handling of Case 4. In this example, Object4 will be perceived as identical to Object1[29] but the label included with Object4 will not match the reminded experience of Object1. The conflict for ZBT in this case is that a category, as defined by the environment, has two different labels or meanings. Although conceptually

---

[28] ZBT actually reports that there was a close match to the A category, but this aspect of ZBT's operation has not received a great deal of attention. Work has instead focused on experimentally verifiable aspects of the model. The concept formation literature has not, as yet, addressed the specific concept formation task confronting ZBT. Previous experiments have largely concentrated on the perception of well-formulated, natural concepts such as cups, bowls, birds, and animals (e.g., Labov, 1973). The statistical approach was incorporated into ZBT because of evidence presented by a number of researchers that subjects prefer an all-or-none concept formation strategy (e.g., Trabasso & Bower, 1968, and Bruner, Goodnow, & Austin, 1956) and that the natural categories formed by people do not seem to have fixed boundaries (McCloskey & Glucksberg, 1978).

[29] As previously mentioned, if the moments of two experiences are within 0.2% of each other they are considered identical. Operating with this criteria ZBT would normally segregate the Object1 and Object4 experiences, however, by increasing the range to .31% they overlap and Case 4 can be demonstrated. Note that, Object1 and Object4 were selected from one of the fonts in ZBT's test repertoire. The close correspondence does not hold across other fonts in the repertoire.

different than Case 3, ZBT attempts to resolve this situation in a similar fashion. How it does this is outlined in the next section.

6.2.3.1 The Object4 Experience

Encountering Image 6 (an image of Object4 labeled Class Y), ZBT does the following:

1) Decomposes the image into a single blob form.
2) Computes the invariants (moments) of the component blob.
3) Uses the invariants to index into memory for comparable experiences.
4) Is reminded of an experience (i.e., the top-level of the Object1 experience).
5) Stores the new experience in memory at the same location as the Object1 experience.
6) Discovers that the label of the new experience is different from the remind.
7) Decomposes the single blob into 3 component blobs.
8) Computes the invariants of the component blobs.
9) Indexes into memory with the 3 component blobs.
10) Finds no similar experiences in memory. (It now has a difference between the two top-level experiences.)
11) Stores the 3 new experiences in memory after serially attending to each component and its spatial relationship with its sibling.

Notice that ZBT detected the difference between the two objects when referencing memory with the decomposed blobs of the second level of Object4. In the previous example, Case 3, ZBT had to compare spatial relationships of the components at that level in order to detect the difference. In most other regards ZBT's activity on this example, and the resultant memory structures, are very similar to what has been described previously. Figure 11 summarizes the final memory structures (the broken ellipse clusters two experiences).

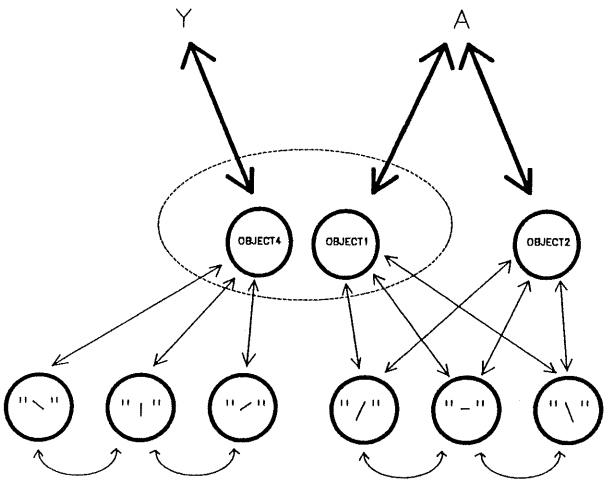

Figure 11:
Memory contents after Image 6

7.0 Summary, Discussion, and Future Directions

Cases of mistaken identity and, possibly, identification of shadows, image outlines, and cartoon caricatures are examples of recognitions that occur without reference to every nuance of detail available in the stored (learned) image. Contrasted with these are recognitions that require greater attention to detail in order to distinguish structural similarities and differences that define whether objects are distinctly different from each other or simply structural alterations that belong to the same class. The ZBT incremental learning and recognition model proposes an hierarchical memory to account for this dual functionality. The hierarchy results from an recursive application of two learning paradigms: conceptual clustering and learning-from-examples. The combination extends the image processing technique known as moment analysis to handle simple, structural transformations in a real-world, character recognition domain.

ZBT's hierarchical dichotomy corresponds to an apparent parallel/serial dichotomy in the human visual system (many references including Julesz, 1962 and Treisman, 1985). First, parallel ZBT processes decompose the image along dimensions of contrast and line orientation (consistent with psychological and neurophysiological evidence presented by many authors including Van Essen & Maunsell, 1983, Julesz, 1982, and Treisman, 1985). Then, corresponding to the serial portion of the dichotomy, ZBT serially attends to the spatial relationships between components to reassociate the decomposed constituents.

A decomposition is a simple segmentation of the image into meaningless (unlabeled) blobs. Unlike classification schemes that attempt to decompose an image directly into cognitively identifiable objects, ZBT places less burden on the decomposition mechanisms by simply requiring consistent segmentation into unlabeled blobs. Thus, the decomposition mechanisms do not have to determine what is salient in the image. Saliency is specified by the environment and ZBT has to make the association.

Blobs are stored in memory as abstracted experiences. ZBT abstracts the blob experiences by computing the moments of the original raster forms. The theory underlying ZBT does not propose that the visual system necessarily utilizes the technique of moment analysis, but it does propose that some type of innate mechanisms perform an abstraction of the data such that the experiences can be clustered in memory invariant to some set of image transformations. In response to the problem paradigm stated here, ZBT utilizes moments because they are invariant to the transformations of translation, scale, and rotation. The actual invariants utilized by nature may be less than this.

The invariants cluster blobs in memory on the basis of feature similarity. Thus, ZBT can acquire and retrieve experiences in the absence of environmental feedback. However, in the presence of feedback, ZBT's invariant measures of similarity may conflict with those of the teacher and negate the grouping. That is, combining conceptual clustering and learning from examples can cause a problem. The problem is a conflict or disagreement between the way the environment chooses to classify experiences and the attributes ZBT utilizes to cluster them. From ZBT's point of view, there are two types of conflict: a label can ambiguously represent two categories or a category can ambiguously have two labels.

Instead of redoing its clusters to satisfy the environmental classifications, ZBT refines the approximately correct grouping by decomposing the current level and serially searching for variable structural information among the decomposed constituents.

If the spatial relationships do not disambiguate the image, ZBT recursively zooms on constituent blobs until memory is matched or it exhausts zoom levels in the image. The recursive combination of conceptual clustering and learning from examples builds a is-a and

part-of hierarchy that distinguishes two types of visual feature. Vertically, levels of the hierarchy represent levels of feature detail that are invariant to the chosen transformations (i.e., translation, scale, rotation). Horizontally, the relationships represent features that are allowed to vary in the domain (i.e., spatial relationships).

Viewed another way, a recognition in ZBT can be considered a two step search, first, acting like a moment analysis system, ZBT identifies candidate recognitions by matching invariant features. Then it verifies the candidates by re-associating the previously decomposed features. The recursive addition of the second part of the process overcomes the limitations of moment analysis discussed earlier, including: problems associated with segmentation, difficulties of matching variant structural alterations, and the dilemma of segregating overlapping dissimilar moments while matching moments of similar objects.

7.1 Future Directions

Interesting questions have come out of the work on ZBT. Initially, the issues were related to ZBT's organization and functioning. For example, in the early stages of development ZBT's decomposition mechanisms required a great deal of attention. The reason is that in order to implement a complete working model, some type of decomposition technique was required but the experimental literature does not support a specific computational implementation. There is experimental evidence suggesting specific types of decomposition (e.g., segmentations based on local density and line orientation - see the summary by VanEssen and Maunsell, 1983 and arguments presented by Julesz & Bergen, 1983 and Treisman, 1985), but no evidence suggesting how to obtain them. The computational method eventually chosen has proved adequate for the simple class of problems described here, but it will not allow ZBT to handle multiple aggregation problems like that illustrated in Treisman's masking experiments. A more comprehensive decomposition approach has been proposed, but experimental evidence supporting it is currently scanty. Support for the new approach is being pursued in the literature and through the possible undertaking of new experiments.

Another issue pointed out by experiments with ZBT concerns ZBT's concept formation capability. The difficulty is that psychologists have not yet tested the specific concept formation task that ZBT confronts (illustrated by the Object1/Object2/Object3 example presented earlier). There are significant questions related to this that affect the design of ZBT. For example, if a person (machine) has had a single experience with a particular object (e.g., Object1) and if a second experience of the object includes a minor but definite alteration of the object's structure (e.g., Object2), is the new form recognized as equivalent to the original or must the system be told that a break is acceptable? In other words, as the structure of a definite object is varied, at what point does it cease to be the original object and take on a new identity? This question and related ones are currently being considered for experimental testing.

As ZBT matured, attention turned to an exploration of its performance and limitations. There are two aspects of ZBT's performance to consider: its ability to handle invariant transformation and its ability to cope with variant transformations.

There was a two-fold rationale for employing moments as the invariant indices. First, there is the opportunity to build on the success of documented moment analysis systems. As previously described, ZBT's clustering appears to be equivalent to that reported for other moment analysis systems (specifically Alt, 1962 and Lambert, 1968), however, it is not known how ZBT will perform when a substantial number of objects have been learned. This continues to be tested with additional fonts of well-formed characters and their transformations.

The second rationale behind the use of moments was the need in ZBT to provide some type of invariance to image translation, rotation, and scale. As previously mentioned, the ZBT model does not propose that the visual system makes use of moments. It only proposes that there are some set of transformations which the visual system is invariant to. Moments are one specific computational technique for achieving that quality. Although moments may not be computed in the visual system, the work on ZBT has shown the usefulness of using moment-like invariant properties as indices into the knowledge base in order to simulate the pop-out quality.

While investigation continues into ZBT's preattentive handling of invariant image qualities, work continues on its attentive aspects and their ability to resolve the variants aspects of image transformation. So far, ZBT has been tested on about six different examples of "broken" symbols. ZBT's success with these examples indicates that it does extend the capabilities of moment analysis, but, again, it is not known how well it will perform when a large number of such objects have been learned. Other aspects of attentive processing that need to be tested include: ZBT's ability to handle multiple alterations per symbol and the specific spatial relationships required to resolve the possible structural alterations.

Aknowledgements

I am thankful to a number of people who took the time to read this paper and comment, including Vince Brown, Kurt Eiselt, and Laura Yoklavich. I am also thankful for Laura's help in creating the drawings.

References

Alt, F. (1962). Digital Pattern Recognition by Moments. In G.L. Fischer, et al. (Eds.), Optical Character Recognition. Spartan, Washington, D.C.

Beck, J. (1967). Perceptual grouping produced by line figures. Percept. Psychophys. 2, 491-495.

Boyer, R.S. & Moore, J.S. (1977). A fast string searching algorithm. Comm. of the ACM, 20, 10, 762-772.

Brady, M. (1982). Computational approaches to image understanding. Computing Surveys, 4, 1, 3-72.

Bruce, C., Desimone, R. & Gross, C.G. (1981). J. Neurophysi., 46, 369-384.

Bruner, J.S.. Goodnow, J., & Austin, G.A. (1956). A Study of Thinking. Wiley, New York.

Casey, R. G. (1970). Moment normalization of handprinted characters. IBM J. Res. Development, 14, Sept., 518-527.

Crick, F. (1984). Function of the thalamic reticular complex: The searchlight hypothesis. Proc. Nat. Acad. Science, 8, 4586-4590.

Duda, R.O. & Hart, P.E. (1973). Pattern Classification and Scene Analysis. Wiley-Interscience, New York.

Dudani, S.A, Kenneth, K.J., and McGhee, R.B. (1977). Aircraft identification by moment invariants. IEEE Trans. on Computers, C-26, January, 39-46.

Ebbinghaus, H. (1913). Memory: A contribution to experimental psychology (translated by Ruger. H.A. & Bussenues, C.E., 1913). Teachers College, Columbia University, New York.

Fukanaga, K. (1972). Introduction to Statistical Pattern Recognition, Academic Press, New York.

Granlund, G.H. (1978). In search of a general picture processing operator. Comp. Graphics and Image Proc., 8, 155-173.

Hall, E.L., Wong, R.Y., Chen, C.C, Sadjadi, F., & Frei, W. (1976). Invariant features fro quantitative scene analysis. Final Report, Image Processing Institute, Department of Electrical Engineering, University of Southern California.

Hall, E.L., Crawford, W.O., and Roberts, F.E. (1975). Computer classification of pneumoconiosis from radiographs of coal workers. IEEE Trans. Biomedical Engineering, BME-22, November, 518-527.

Hall, E.L., Wong, R.Y., Chen, C.C., Sadjadi, F., and Frei, W. (1976), Invariant Features for Quantitative Scene Analysis, Final Report. Image Processing Institute, Department of Electrical Engineering, University of Southern California, July.

Hu, M. K. (1962). Visual pattern recognition by moment invariants. IRE Trans. Information Theory, IT-8, February.

Hubel, D.H. and Wiesel, T.N. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. J. Physiology. 77, 281-285.

Hubel, D.H. and Wiesel, T.N. (1968). Receptive fields and functional architecture of monkey striate cortex. J. Physiology. 160, 106-154.

Hubel, D.H. and Wiesel, T.N. (1977). Proc. R. Soc. London Ser. B, 198, 1-59.

Julesz, B. (1962). Visual pattern discrimination. IRE Trans. Inform. Theory, IT-8:84-92.

Julesz, B. (1975). Toward and axiomatic theory of preattentive vision. Sci. American, 232, 34-43.

Julesz, B. & Bergen, J.R. (1983). Textons, the fundamental elements in preattentive vision and perception of textures. Bell Syst Tech J, 62, 1619-1645.

Koch, C. & Ullman, S. (1985). Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiol., 4.

Labov, W. (1973). The boundaries of words and their meanings. In C. N. Bailey and R. W. Shuy (Eds.), New Ways of Analyzing Variations in English. Georgetown Press, Washington, D.C.

Lambert, P. F. (1969). Designing Pattern Categorizers with Extremal Paradigm Information. In S. Watanabe (Ed.), Methodologies of Pattern Recognition. Academic Press, New York.

Lewis, P.M. (1962). The characteristic selection problem in recognition systems. IRE Transactions on Information Theory, February, 171-178.

Lowerre, B. & Reddy, R. (1980). The HARPY speech recognition understanding system. In Lea, Trends, 340-360.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information. W.H. Freeman, San Francisco.

Marr, D. & Nishihara, H.K. (1978) Representation and recognition of the spatial organization of three dimensional shapes, Proc. Roy Soc Lond B, 200, 269-294)

McCloskey, M.E. & Glucksberg, S. Natural categoriess: Well-defined or fuzzy sets?. Memory and Cognition, 6, 462-472.

Michalski, R. (1980). Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and Algorithm for Partitioning Data into Conjunctive Concepts. International Journal of Policy Analysis and Information Systems, 4, 3, 219-243.

Moran, J. & Desimone, R. (1985). Selective attention gates visual processing in the extrastriate cortex. Science, 229, 782-784.

Nakayama, K. & Silverman, G. H. (1986). Serial and parallel processing of visaul feature conjunctions. Nature, 320, pp. 264-265, March 20.

Neisser, U. (1967). Cognitive Pshychology. Appleton-Century-Crofts. New York.

Pavlidis, T. (1978). Survey: A review of algorithms for shape analysis. Comp Graphics and Image Proc, 7, 243-258.

Pavlidis, T.. (1982). Algorithms for Graphics and Image Processing. Computer Science Press, Rockville, MD.

Perrett, D.I., Rolls, E.T. & Caan, W. (1982). Exp. Brain Res, 47, 329-342.

Quinlin, J. R. (1986). The effect of noise on concept learning. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, Volume II. Morgan Kaufman, Los Altos, CA.

Rosenfield, A. (1978). Survey: Picture Processing 1977. Computer Graphics and Image Processing, 7, 211-242.

Schlimmer, J.C. & Fisher, D. (1986). A Case Study of Incremental Concept Induction. Proceedings Fifth National Conference on Artificial Intelligence, pg. 496-501.

Selferidge, O. & Neisser, U. (1960). Pattern recognition by machine. Scientific American, 203, 60-68.

Trabasso, T.R. & Bower, G.H. (1968). Attention in Learning. Wiley, New York.

Treisman, A. (1985). Preattentive Processing in Vision. Computer Vision, Graphics, and Image Processing. Academic Press, New York.

Treisman, A. (1982). Perceptual grouping and attention in visual search for features and for objects. J. Exp. Psychol. Human Percept, 10, 194-214.

Treisman, A. (1983). The role of attention in object perception. In O.J. Braddick & A.C. Sleigh (Eds.), Physical and Biological Processing of Images. Springer, New York.

Treisman, A. & Gelade, G. (1980). A feature integration theory of attention. Cog. Psychol., 12, 97-136.

Treisman, A. & Paterson, R. (1984). Evergent features, attention and object perception. J. Exp. Psychol. Human Percept., 10, 12-31.

Treisman, A. & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. Cog. Psychol., 14, 107-141.

Van Essen, D.C & Maunsell, J.H. (1983). Trends Neurosci., 6, 370-375.

Winston, P.H. (1975). Learning Structural Descriptions from Examples. In P.H. Winston (Ed.), The Psychology of Computer Vision. McGraw-Hill, New York.

Wong, R.Y., Hall, E.L. (1978) Scene matching with invariant moments. Computer Graphics and Image Processing, 8, 16-24.

Wong, R.Y., Hall, E.L., and Rouge, J. (1976). Hierarchical search for image matching. Proceedings of IEEE Conference on Decision and Control, December.

APPENDIX I:

Runtime output of ZBT test

Top-Level
* (date)
Date is: 9-18-87 14:34

* (sup)

Begin test of ZBT.
Clear memory (C) or accumulate? c
Look at a single image(I) or a sequence(S)? s
Choose images (C) or automatic selection (A)? c
(Separate with spaces, end with ')CR')? (triangle1 triangle2 triangle1 triangle3)

SUPERVISOR:
Passing TRIANGLE1 to ZBT; image 1 in sequence with assigned label TRIANGLE.
In cognitive terms this image can be roughly described as:
A WELL FORMED TRIANGLE

ZBT:
Zooming on the initial or highest level (level 0) of the image.
***** Begin parallel operations *****
Decomposing level 0.
Computing connectivity.
Single blob detected.
TRIANGLE1 decomposed into: BLOB-0
Computing invariants of BLOB-0.
Checking for reminds of BLOB-0.
Experience BLOB-0 caused no reminding.
***** Begin serial operations *****
Only one sibling, no relationships to investigate, attend complete.
Recording level 1.
No similar memories exist; record this experience in unused memory area.
Recording BLOB-0.
Indexing BLOB-0 under label TRIANGLE.
There were no reminds and no comparably labeled experiences.
Therefore, there is no conclusion concerning this image; and no reason to zoom.

SUPERVISOR:
Passing TRIANGLE2 to ZBT; image 2 in sequence with assigned label TRIANGLE.
In cognitive terms this image can be roughly described as:
A TRIANGLE WITH A SMALL BREAK IN THE LOWER RIGHT CORNER

ZBT:
Zooming on the initial or highest level (level 0) of the image.
***** Begin parallel operations *****
Decomposing level 0.
Computing connectivity.
Single blob detected.
TRIANGLE2 decomposed into: BLOB-1
Computing invariants of BLOB-1.
Checking for reminds of BLOB-1.
Experience BLOB-1 caused no reminding.
BLOB-0 has also been identified with this category.
There are no reminds, that is, the levels attended to so far do not
index to a previously stored experience, however, there are experiences
stored under the label TRIANGLE.
This is a conflict since similarly labeled experiences should
cause similar reminds.
Attempt to resolve the conflict by looking for previously unattended
differences between the current image and BLOB-0
Only one sibling, no reason to attend.
Recording level 1.
No similar memories exist; record this experience in unused memory area.

Recording BLOB-1.
Indexing BLOB-1 under label TRIANGLE.
Zooming on level 1.
***** Begin parallel operations *****
Decomposing level 2.
Computing connectivity.
Single blob detected.
Same decomposition as previous level.
Computing by projections.
Searching for blobs which conform to the horizontal.
Found the following horizontal blobs: BLOB-2
Searching for blobs which conform to the vertical.
None were found.
Searching for blobs which conform to the diagonals.
Found the following diagonal (lower-left to upper right) blobs: BLOB-3
Found the following diagonal (lower-right to upper left) blobs: BLOB-4
BLOB-1 decomposed into: BLOB-4 BLOB-3 BLOB-2
Computing invariants of BLOB-4.
Computing invariants of BLOB-3.
Computing invariants of BLOB-2.
Checking for reminds of BLOB-2.
Experience BLOB-2 caused no reminding.
Checking for reminds of BLOB-3.
Experience BLOB-3 caused no reminding.
Checking for reminds of BLOB-4.
Experience BLOB-4 caused no reminding.
There are no reminds, that is, the levels attended to so far do not
index to a previously stored experience, however, there are experiences
stored under the label TRIANGLE.
This is a conflict since similarly labeled experiences should
cause similar reminds.
Attempt to resolve the conflict by looking for previously unattended
differences between the current image and BLOB-0
Look for the structural differences at this level of zoom which
can resolve the conflict.
***** Begin serial operations *****
Looking for spatial relationships among siblings BLOB-2 BLOB-3 BLOB-4
Attending to BLOB-2 and its relationship to BLOB-3.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-2 0.0
BLOB-3 0.0
Attending to BLOB-2 and its relationship to BLOB-4.
Proximal relationship detected
Distance = 2.0
Locus points (relative to total length and measured from lower and left):
BLOB-2 1.0
BLOB-4 0.0
Attending to BLOB-3 and its relationship to BLOB-2.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-3 0.0
BLOB-2 0.0
Attending to BLOB-3 and its relationship to BLOB-4.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-3 1.0
BLOB-4 1.0
Attending to BLOB-4 and its relationship to BLOB-2.
Proximal relationship detected
Distance = 0.0
Locus points (relative to total length and measured from lower and left):
BLOB-4 0.0

BLOB-2 9.23077F-01
Attending to BLOB-4 and its relationship to BLOB-3.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-4 9.23077F-01
BLOB-3 9.23077F-01
Recording level 2.
Ancestor to this level is BLOB-1.
No similar memories exist; record this experience in unused memory area.
Recording BLOB-2.
Recording vertical links.
Record pointer from BLOB-2 to ancestor BLOB-1.
Record reverse pointer from ancestor BLOB-1 to BLOB-2.
Recording horizontal links.
No similar memories exist; record this experience in unused memory area.
Recording BLOB-3.
Recording vertical links.
Record pointer from BLOB-3 to ancestor BLOB-1.
Record reverse pointer from ancestor BLOB-1 to BLOB-3.
Recording horizontal links.
No similar memories exist; record this experience in unused memory area.
Recording BLOB-4.
Recording vertical links.
Record pointer from BLOB-4 to ancestor BLOB-1.
Record reverse pointer from ancestor BLOB-1 to BLOB-4.
Recording horizontal links.
The zoom level for the current experience now exceeds the maximum stored level of the conflict.
It doesn't make sense to zoom further on this image.

SUPERVISOR:
Passing TRIANGLE1 to ZBT; image 3 in sequence with assigned label TRIANGLE.
In cognitive terms this image can be roughly described as:
A WELL FORMED TRIANGLE

ZBT:
Zooming on the initial or highest level (level 0) of the image.
***** Begin parallel operations *****
Decomposing level 0.
Computing connectivity.
Single blob detected.
TRIANGLE1 decomposed into: BLOB-5
Computing invariants of BLOB-5.
Checking for reminds of BLOB-5.
Experience BLOB-5 caused a reminding of BLOB-0.
BLOB-1 has also been identified with this category.
The current experience appears to be TRIANGLE, but there is a conflict since there is another experience in the same category.
***** Begin serial operations *****
Only one sibling, no relationships to investigate, attend complete.
Recording level 1.
Zooming on level 1.
***** Begin parallel operations *****
Decomposing level 2.
Computing connectivity.
Single blob detected.
Same decomposition as previous level.
Computing by projections.
Searching for blobs which conform to the horizontal.
Found the following horizontal blobs: BLOB-6
Searching for blobs which conform to the vertical.
None were found.
Searching for blobs which conform to the diagonals.

Found the following diagonal (lower-left to upper right) blobs: BLOB-7
Found the following diagonal (lower-right to upper left) blobs: BLOB-8
BLOB-5 decomposed into: BLOB-8 BLOB-7 BLOB-6
Computing invariants of BLOB-8.
Computing invariants of BLOB-7.
Computing invariants of BLOB-6.
Checking for reminds of BLOB-6.
Experience BLOB-6 caused a reminding of BLOB-2.
Checking for reminds of BLOB-7.
Experience BLOB-7 caused a reminding of BLOB-3.
Checking for reminds of BLOB-8.
Experience BLOB-8 caused a reminding of BLOB-4.
There are no reminds, that is, the levels attended to so far do not
index to a previously stored experience, however, there are experiences
stored under the label TRIANGLE.
This is a conflict since similarly labeled experiences should
cause similar reminds.
Attempt to resolve the conflict by looking for previously unattended
differences between the current image and BLOB-1
Look for the structural differences at this level of zoom which
can resolve the conflict.
***** Begin serial operations *****
Looking for spatial relationships among siblings BLOB-6 BLOB-7 BLOB-8
Attending to BLOB-6 and its relationship to BLOB-7.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-6 0.0
BLOB-7 0.0
Attending to BLOB-6 and its relationship to BLOB-8.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-6 1.0
BLOB-8 0.0
Attending to BLOB-7 and its relationship to BLOB-6.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-7 0.0
BLOB-6 0.0
Attending to BLOB-7 and its relationship to BLOB-8.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-7 1.0
BLOB-8 1.0
Attending to BLOB-8 and its relationship to BLOB-6.
Intersection relationship detected.
Locus points (relative to total length and measured from lower and left):
BLOB-8 0.0
BLOB-6 9.23077F-01
Attending to BLOB-8 and its relationship to BLOB-7.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-8 9.23077F-01
BLOB-7 9.23077F-01
Recording level 2.
Experience BLOB-5 caused a reminding of BLOB-0.
Ancestor to this level is BLOB-0.
Experience BLOB-6 caused a reminding of BLOB-2.
Matched BLOB-6 to BLOB-2.
Experience BLOB-7 caused a reminding of BLOB-3.
Matched BLOB-7 to BLOB-3.
Experience BLOB-8 caused a reminding of BLOB-4.
A spatial structural difference has been detected in the
BLOB-6/BLOB-8 relationship (compared to the previous BLOB-2/BLOB-4).

Distance relationships are different.
Label TRIANGLE matches previous relationship.
Category TRIANGLE must allow this difference.
Recording wildcard for BLOB-2/BLOB-4 distance relationship.
The image has been disambiguated.

SUPERVISOR:
Passing TRIANGLE3 to ZBT; image 4 in sequence with assigned label UNLABELED.
In cognitive terms this image can be roughly described as:
A TRIANGLE WITH A LARGE BREAK IN THE LOWER RIGHT CORNER

ZBT:
Zooming on the initial or highest level (level 0) of the image.
***** Begin parallel operations *****
Decomposing level 0.
Computing connectivity.
Single blob detected.
TRIANGLE1 decomposed into: BLOB-9
Computing invariants of BLOB-9.
Checking for reminds of BLOB-9.
Experience BLOB-9 caused no reminding.
***** Begin serial operations *****
Only one sibling, no relationships to investigate, attend complete.
Recording level 1.
This experience is unlabeled, therefore zoom for other detail
that might provide a match.
Zooming on level 1.
***** Begin parallel operations *****
Decomposing level 2.
Computing connectivity.
Single blob detected.
Same decomposition as previous level.
Computing by projections.
Searching for blobs which conform to the horizontal.
Found the following horizontal blobs: BLOB-10
Searching for blobs which conform to the vertical.
None were found.
Searching for blobs which conform to the diagonals.
Found the following diagonal (lower-left to upper right) blobs: BLOB-11
Found the following diagonal (lower-right to upper left) blobs: BLOB-12
BLOB-5 decomposed into: BLOB-12 BLOB-11 BLOB-10
Computing invariants of BLOB-12.
Computing invariants of BLOB-11.
Computing invariants of BLOB-10.
Checking for reminds of BLOB-10.
Experience BLOB-10 caused a reminding of BLOB-2.
Checking for reminds of BLOB-11.
Experience BLOB-11 caused a reminding of BLOB-3.
Checking for reminds of BLOB-12.
Experience BLOB-12 caused a reminding of BLOB-4.
Reminds have a common label of TRIANGLE.
This appears to be a TRIANGLE.
Look for the structural differences at this level of zoom which
might resolve.
***** Begin serial operations *****
Looking for spatial relationships among siblings BLOB-10 BLOB-11 BLOB-12
Attending to BLOB-10 and its relationship to BLOB-11.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-10 0.0
BLOB-11 0.0
Attending to BLOB-10 and its relationship to BLOB-12.
Intersection relationship detected

Locus points (relative to total length and measured from lower and left):
BLOB-10 1.0
BLOB-12 0.0
Attending to BLOB-11 and its relationship to BLOB-10.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-11 0.0
BLOB-10 0.0
Attending to BLOB-11 and its relationship to BLOB-12.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-11 1.0
BLOB-12 1.0
Attending to BLOB-12 and its relationship to BLOB-10.
Intersection relationship detected.
Locus points (relative to total length and measured from lower and left):
BLOB-12 0.0
BLOB-10 9.23077F-01
Attending to BLOB-12 and its relationship to BLOB-11.
Intersection relationship detected
Locus points (relative to total length and measured from lower and left):
BLOB-12 9.23077F-01
BLOB-11 9.23077F-01
Recording level 2.
Experience BLOB-9 caused a reminding of BLOB-0.
Ancestor to this level is BLOB-0.
Experience BLOB-10 caused a reminding of BLOB-2.
Matched BLOB-10 to BLOB-2.
Experience BLOB-11 caused a reminding of BLOB-3.
Matched BLOB-11 to BLOB-3.
Experience BLOB-12 caused a reminding of BLOB-4.
Wildcard match of BLOB-12 to BLOB-4.
Ancestor of reminds is BLOB-1.
Label of ancestor is TRIANGLE.
This image must be a TRIANGLE.
The image has been disambiguated.

SUPERVISOR:
ZBT test complete.
NIL
*

(dribble)

APPENDIX II:

Computational Details of ZBT's Segmentation Procedures

Appendix II

There are two mechanisms that make up a decomposition in ZBT. The first mechanism isolates a raster form from the surrounding background. This is necessary to identify which portion of the image the moments are to be calculated on and, thus, normalize the moments to that area (as opposed to unnormalized across the entire area). ZBT employs a simple connectivity analysis to accomplish this. That is, during the first part of decomposition, ZBT isolates connected areas of common pixel values. Since ZBT deals exclusively with binary images, that means that ZBT groups all of the on pixels (i.e., the black area) that are touching each other. Additionally, ZBT places a one pixel contrasting boundary around the area. This procedure is effectively a segmentation by simple contrast and not inconsistent with experimental evidence suggesting some type of decomposition by contrast in the visual system (e.g., Julesz's, 1983 segregation based on local density of visual features).

The second decomposition mechanism is consistent with other data suggesting the existence of primitives based on the orientation of spatial lines (many including: Hubel & Wiesel, 1962, Beck, 1967, Treisman, 1982, and Julesz, 1983). A decomposition by orientation is composed of the following steps:

1) Calculate the projections.
2) Group the projections.
3) Find the boundary which encompasses the most pertinent group, thus, defining the segmented blob.
4) Compute the invariants of the blob.

A projection is a mapping from an n-dimensional space to an m-dimensional space. If n is 3 and m is 2, the projection eliminates the depth information from a 3D image. This is a common computation in graphics applications. If n is 2 and m is 1, as it is when handling two-dimensional images, the resultant vector contains the sums of black pixels counted through the respective discrete parallel lines. For example, the horizontal projection of a 5x5 matrix is calculated by counting across (in the y direction[1]) the five edge (or x) elements (see Figure). Five sums or counts of black pixels result. From the vector, ZBT then computes the standard deviation of the sums and identifies the boundary locations. Boundaries are calculated in the following manner. A search begins at each end of the vector. It terminates when the first value one standard deviation above noise level is encountered. When calculating a horizontal projection, these values define the horizontal boundaries (or x values of the blob; call them x_1 and x_2). The vertical boundaries of a horizontal projection are found by searching along x_1 and x_2 in the original array until a black pixel is encountered. These then constitute the y values. The boundaries define the blob coincident with the projection of that orientation.

Figure:
Horizontal projection on a 5x5 matrix

Four types of blob are identified by ZBT. The four types correspond to the four directions: horizontal, vertical, lower-left to upper-right diagonal, and lower-right to upper-left diagonal. Vertical projections are computed in a fashion inversely comparable to the horizontal method. Diagonal strokes are somewhat more difficult, but are done in much the same way. Given the difficulty of computing the other, off-diagonal projections and the time to process (an important consideration when observing hundreds of tests) a decision was made to limit ZBT to the four projections mentioned. The only consequence of the limitation, observed so far, is that symbols, that will undergo a second level decomposition, must be chosen such that their components are formed on one of the four axes (i.e., curved components are not detected in the

[1] This usage of coordinates may be confusing to the reader because many are accustomed to x being the direction across and y the up and down direction. The reverse was utilized in this work to be consistent with a large portion of the graphics and image processing literature.

second decomposition)[2]. The projection method was not chosen arbitrarily. It was chosen to resemble cells discovered in the early visual system (Hubel & Wiesel, 1962). These cells fire only in the presence of lines of a specific spatial orientation. Additionally, these cells fire with a frequency proportional to the spatial frequency of the image lines.

ZBT employs the two mechanisms in a sort of competition. Each mechanism is performed in parallel on each area of attention. The results of the two are compared with each other and with the previously chosen decomposition. ZBT prefers the decomposition with the fewest components as long as it does not match the previously selected segmentation. On the examples presented here, that means that connectivity will be chosen in the first decomposition of the image (since each image presented here is a single entity in the visual field) and the result will be a single segment containing all of the black pixels which define the top-level form. After the results of the second decomposition are compared, ZBT will choose the results of the decomposition by orientation since the segment formed by contrast is the same as the previously selected decomposition. Thus, the second decomposition on the image will reveal the three component blobs of the "A" (see Figure 7 in the main text) or the two components of the "T" or "V".

[2] This does not conflict with ZBT's ability to handle entire alphabets, as previously stated. Recall that the moments of the first decomposition of characters will segregate them adequately (Alt, 1962). The limitation of this implementation of projection calculation only manifests itself on the second decomposition.

APPENDIX III:

Computation of the Spatial Relationships between Siblings

Appendix III

ZBT calculates the LOCUS and DISTANCE values for a blob and its sibling in the following fashion. Referencing the raster image, one of the projections composing the blob and one of the projections making up the sibling are selected (see Appendix II for a description of projections) as a candidate pair to represent the entire blob/sibling relationship. For each of these candidates the locus of each line and the distance between loci is calculated.

The locus points of the two lines are determined by comparing the their endpoints (in terms of raster location) for intersection (see Pavlidis, 1982, page 329). If the lines are intersecting, the intersecting point that lies on the line of the blob becomes a candidate LOCUS and the point where the intersection occurs on the line of the sibling becomes the point of comparison for computing DISTANCE.

If the lines are not intersecting then the imaginary points of intersection are determined. This is accomplished by extending both lines to the edge of the image and computing the intersecting point using the new endpoints (i.e., where the lines terminate at the edge of the drawing). Since it may only be necessary to extend one of the two lines to achieve an intersection, the imaginary point of intersection is compared to the natural endpoints of the two lines. If this point falls on either line then it is used for the locus of that particular line.

If, on the other hand, the intersecting point is not on a line, the endpoint of the line closest to the point of intersection is selected. The closest endpoint is determined by comparing the natural endpoints to the imaginary point of intersection. The closest one is selected as the locus.

The locus associated with the line of the blob is selected as the locus of this pair, but first each of the values are normalized as a percentage of the entire length of the line from the point of origin of the projection. This value is obtained by calculating the distances from the locus to the two endpoints. The ratio of the distance closest to the point of origin and the sum of the distances is recorded as a percentage value representing the candidate locus of the line. Once the two locus values have been determined the distance value of this pair is simply the distance between the two loci.

The distance and locus values are computed for each of the blob/sibling line pairs. The pair with the smallest distance value (0 means intersecting) is selected to represent the blob/sibling relationship. The distance and locus of that pair is then recorded as DISTANCE and LOCUS respectively.