Integrating Visual Search with Visual Memory in a Knowledge directed Image Interpretation System

T.P. Pridmore and S.H. Joseph

Dept. of Mechanical and Process Engineering
University of Sheffield
Mappin St.
Sheffield S1 3JD

We describe an improved implementation of ANON, a schema-driven image analysis system capable of producing high level interpretations of greyscale images of mechanical engineering drawings. The system has been extended to incorporate a drawing memory into which completed schema instances are placed and which may be accessed as an integral part of ANON's knowledge directed visual search. This memory provides a basis for the integration into a coherent whole of the piecewise interpretations previously supplied by the system. It leads to a richer description of drawing content, improves efficiency and allows ANON to deal with partially complete interpretations. The new system's structure and operation are discussed and examples shown of it's interpretation of real mechanical drawings.[1]

ANON [1,2] is a knowledge directed image analysis system capable of producing high level interpretations of greyscale images of mechanical engineering drawings. These documents (figure 1) contain the rich mixture of graphical, textual and symbolic information needed to provide a means of communication between trained engineers. As with many visual tasks, the complexity of the drawing interpretation problem is such that a systematic and extensible architecture is essential if image analysis is to remain tractable and efficient. The identification of such an architecture is the prime motivation for the present work. However, the interpretation of mechanical drawings is not just a useful testbed for experiments in vision system design; there is a real need for automatic drawing interpretation now.

Computer aided draughting (CAD) systems are widespread in the manufacturing industries. However, while new design problems are now usually addressed within CAD environments, many engineering tasks involve the modification of designs which were developed and presented on paper. In such cases a choice must be made between reworking the design using appropriate CAD tools, a lengthy and largely unproductive task, or finding some method of conversion from the old format to the new. Much thought has been given to the problem of vectorisation, the description of a drawing image in terms of an unstructured collection of line segments [3], but few systems attempt to process real images of real drawings any further. CAD descriptions normally comprise fewer, larger constructs and, consequently, most CAD/CAM applications assume a higher level of representation.

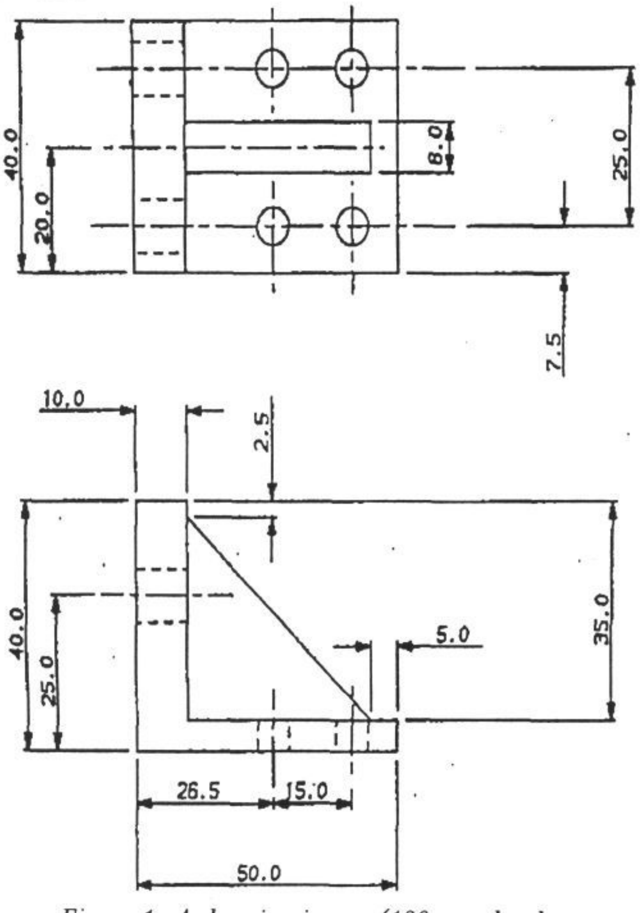

Figure 1. A drawing image (128 grey levels, 10 pixels/mm), thresholded for display only.

ANON uses schemata to describe drawing content. The system (figure 2) is modelled on the human "cycle of perception" [4]; the basis of the approach being a continuous loop in which a constantly changing world model directs perceptual exploration, determines how its findings are to be interpreted and is modified as a result. In ANON this role is taken by an instance of one of a number of schema classes. On each cycle the controlling, or "current", instance invokes appropriate members of ANON's library of image analysis routines and informs a higher level control module of the results of it's actions. The system presently maintains classes corresponding to solid, dashed and chained lines, solid and

[1] This research is supported by SERC ACME grant GRE 41092. T.P. Pridmore is now with the School of Engineering Information Technology, Sheffield City Polytechnic, Pond St., Sheffield S1 1WB.

BMVC 1990 doi:10.5244/C.4.65

dashed curves, cross hatching, physical outlines, junctions, letters, words, witness and leader lines and certain restricted forms of dimensioning. Each schema instance therefore represents a particular example of some prototypical drawing construct.

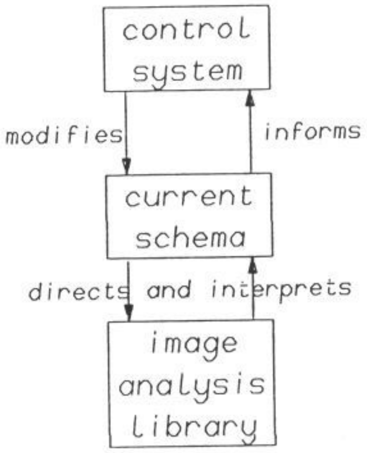

Figure 2. ANON's cycle of perception

The basic operation of the image analysis library is sequential tracking; routines are available for following straight lines [5], circles [6] and area outlines. The grey level corresponding to white paper background is first measured and its noise level assessed. These figures are then used as the basis of dynamic thresholding procedures which improve tracking on poor and variable contrast images. It should be stressed here that all of ANON's image analysis is carried out under the control of some schema instance - in the context of a particular hypothesis regarding the local content of the drawing. The image analysis library therefore provides an extendible tool box of procedures whose application varies according to context.

ANON's control module comprises a set of strategy rules written in the form of an LR(1) grammar and applied by a parser generated using the Unix utility yacc [7]. These rules define methods by which high level drawing entities may be obtained by hierarchical combination of lower level constructs. On each cycle the control system determines an appropriate modification to the current schema. Modification may mean updating an internal variable, adding new sub-parts, or replacing the instance with a new one representing a different type of construct.

Strategy rules, like string grammars, describe acceptable sequences of events; a parser is therefore a natural vehicle for their application. Moreover, LR parsing is a well understood method which provides a "simple, expressive rule format", "formal precision and interpretability" and "guaranteed consistency"; attributes noted by Rao and Jain [7] as among the advantages of logic as a representation in computer vision. The existence of well established software tools based on the technique is an added bonus.

The human perceptual cycle is thought to continue throughout life. ANON, however, needs some procedure for initiating the system given a fresh drawing and terminating it when all the relevant constructs have been found. A book-keeping module therefore directs attention towards unexplained ink marks and terminates the system when no significantly black areas remain unexamined.

ANON has been successful in extracting high-level entities from images of piece-part drawings [1,2]. The present work centres on the problems of integrating these descriptions within a hierarchical drawing memory, dealing with multiple interpretations of a single image object, merging consistent descriptions and flagging conflicts.

PROBLEMS & APPROACHES

Although at a much higher level than previous vector descriptions, ANON's schemata still provide only a piecewise representation of the drawing: if a complete interpretation is to be obtained they must be integrated into a coherent whole. Several difficulties must be faced.

ANON's basic activity is to match image primitives against schema classes in an attempt to identify appropriate schema instances. The context sensitive nature of this task, however, raises the possibility of a given group of pixels contributing to several different contexts. As contexts may intersect, so may the system's schema-based representations of drawing content.

A brief examination of Figure 1 clarifies the problem. While there exist clearly identifiable entities that are the building blocks from which drawings are constructed (hatched areas, chains, etc), these components can be combined in very many different ways. Four basic forms of dimensioning, for example, have been identified [9] and could be described by schemata. They may, however, be combined quite arbitrarily; witness lines are often shared, leaders run together, etc. Under such circumstances overlapping interpretations are unavoidable.

Overlaps may represent local ambiguities. Although each schema covers a significant number of pixels, the constructs involved are still quite small relative to the drawing. Situations therefore arise in which this comparatively local processing is unable to determine the correct interpretation.

An obvious approach would be to continue to develop ANON in the same vein: to attempt to define higher level schemata which can describe the present overlapping, consistent interpretations while rejecting those that are inconsistent. We feel, however, that it is naive to expect ever higher schemata to be able to capture the variety and complexity of interaction that is found in even the simpler mechanical drawings. McDermott [10] argues that a knowledge-based system will display erroneous behaviour for either of two reasons; (1) inadequacies in its knowledge or (2) inadequacies in its problem solving behaviour. We believe that ANON's overlapping interpretations are best addressed by considering the second possibility. In the longer term the system will need to adopt a fundamentally different problem solving technique if full integration of schemata is to be achieved. For the present we are concerned only with merging similar instances and identifying situations in which further work is necessary.

A continuum of approaches presents itself: at one extreme one could simply accumulate schemata, then delegate the integration problem to a separate post-process. This mirrors the standard separation of segmentation and interpretation adopted by, for example, VISIONS [11]. A middle road would be to have the integration process act as a demon that is invoked whenever necessary, much as frames of similar type are merged by the "exception handlers" of SIGMA [12]. The opposite extreme is to implement continuous interaction between the image interpretation and schema integration processes (cf. MAPSEE [13]), making them both part of the perceptual cycle. From our experience of ANON it would appear that much of the system's strength lies in it's ability to integrate segmentation with interpretation [1,2]; we have therefore chosen an approach at the latter end of the above continuum.

In it's original form, the system merely displayed completed schemata to the user. Hence, as interpretation progressed, a considerable amount of information was accrued but not exploited in any way. The goal of the work reported here is to make use of this knowledge by retaining complete schemata in a "drawing memory" (figure 3) which may be examined, along with the image, on subsequent perceptual cycles. During this examination, interactions are noted between the developing schema and those already completed.

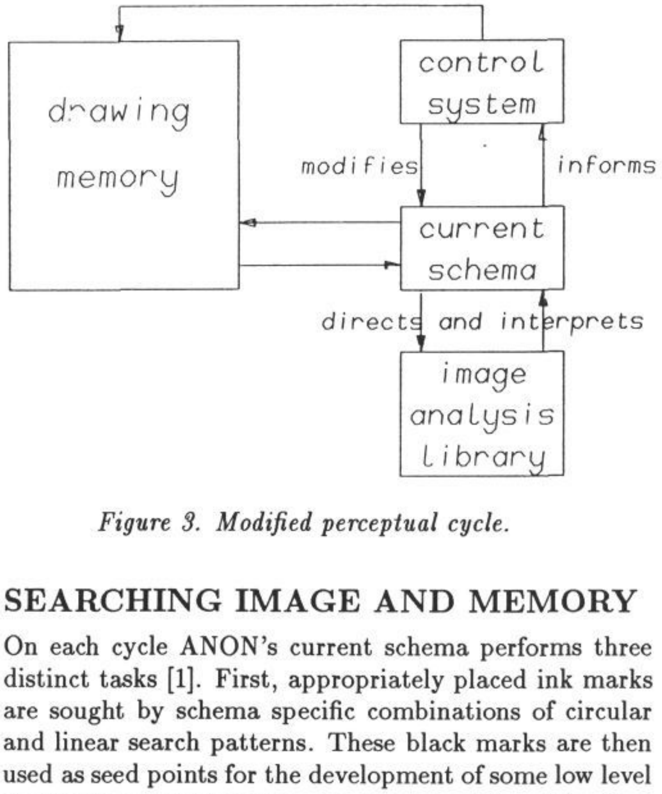

Figure 3. Modified perceptual cycle.

SEARCHING IMAGE AND MEMORY

On each cycle ANON's current schema performs three distinct tasks [1]. First, appropriately placed ink marks are sought by schema specific combinations of circular and linear search patterns. These black marks are then used as seed points for the development of some low level descriptive primitive; line, arc or area outline. The final step is to interpret that primitive in the prevailing context, this results in a symbolic label or "token" which is attached to the primitive before it is passed to the control module. How then can this sequence of events be altered or extended to incorporate drawing memory? What is needed is some point of access and some way of "tokenising" the instance(s) returned. There is clearly no need to develop schemata, complete constructs will be supplied directly from memory.

Consider access first. Since drawing memory contains schemata extracted from the grey level array, the latter provides a spatial index to the former. Furthermore, as schemata only arise from inked areas of the drawing, the ink marks identified early in each cycle provide points of access that are both natural and efficient. The current schema's knowledge of where to look for the next construct can therefore be used to initiate analysis of both the drawing image and the drawing memory. As several marks typically arise on each cycle a schema specific function must decide which candidate(s) to pursue first.

ANON's descriptions are organised hierarchically (eg. figure 4). Each schema instance contains both a geometrical summary of the construct it represents and links to other schemata representing its components. Dimensions, for example, comprise one text, one leader and two witness instances. Each witness in turn contains a root line, an (optional) chain and a junction instance marking the dimensioned point. At the leaves of the tree are instances of the system's primitives.

Drawing memory is naturally searched top down; successively lower level components are examined until an instance is encountered which the current schema interprets as worthy of consideration by the strategy grammar. Suppose, for example, that a given ink mark overlays the leader line of a previously completed dimension schema. On first examination drawing memory reveals the entire dimension set. If this is not deemed suitable the next level sub-component is considered. This is the leader instance, which comprises a line with arrowheads. Should that also be found lacking attention passes to the next level down; the line. Having reached a leaf of the schema tree, search terminates: with success if the line is acceptable, with failure otherwise. As interpretations often overlap, a given ink mark may lie on more than one schema. Drawing memory therefore supplies an ordered list of instances at each level, precedence being given (somewhat arbitrarily in this context) to those most recently stored.

This top down search for appropriate instances in drawing memory neatly complements ANON's bottom-up interpretation of the drawing image [1]; the current schema develops bottom up while incorporating the most significant acceptable constructs from memory.

During image interpretation only primitives need to be tokenised, each schema therefore incorporates a function which characterises the relative geometric and other properties of the current instance and the latest line, arc or area outline. Memory search, however, may return instances of any class. A combinatorial problem arises; each schema class needs some way of tokenising every other. In the worst case this suggests an N x N array of tokenisation functions (N=16 at present). As such difficulties place restrictions on system growth, some coherent method of avoiding the problem must be found.

One of the issues addressed during the initial implementation of ANON was how an image interpretation system which essentially operates bottom up could hypothesise and test for the existence of high level constructs in top down fashion. The solution adopted [1,2] was to employ directed bottom up processing; that is, to have the current schema predict a primitive which should be part of the desired structure. If found, this primitive seeds bottom up analysis. Should an instance of the hypothesised class be constructed the test is positive.

Recall that drawing memory is accessed through ink marks which would otherwise be used as the basis of descriptive primitives. If, having hypothesised some high level entity, an instance of the appropriate class can be retrieved from memory via these marks it seems fair to assume that a similar instance would have been extracted from the image had bottom up interpretation continued. Hence only class is considered when tokenising any high level instances returned from memory. For these constructs at least tokenisation is trivial.

Greater care is needed when tokenising retrieved primitives. These lowest level entities are typically detected by more exploratory search patterns which generate a number of possibilities at once. Under such circumstances explicit geometrical and other tests are clearly required if the correct interpretation is to be identified. The tokenisation functions used during image interpretation can, however, be applied without change to primitives extracted from drawing memory. No extension of schema classes is therefore necessary.

Our method of tokenising instances retrieved from memory is directly analogous to that employed during the examination of high-level predictions of image content by the original ANON. As noted then [1], only a limited amount of the knowledge embodied in the prediction/retrieved instance is actually exploited by the method. So far, however, this limitation does not appear to restrict the application of the technique.

In the current version of ANON constructs retrieved from drawing memory are given precedence over those extracted from the drawing image. Image analysis only proceeds when nothing worthy of further attention is supplied by memory search. One drawback of the approach is that the system can be misled by its memory. Situations arise in which a better construct is apparent in the image but not considered because the partially complete memory contains an acceptable, though sub-optimal, construct. A particularly interesting aspect of this work lies in exploring the extent to which such errors disrupt interpretation. At the present level of development no significant difficulties have been encountered, though we are currently examining alternative ways of allocating priority.

STRATEGY RULES

An interesting feature of ANON's three layer structure is its separation of geometric and strategic knowledge; the former is confined to schema classes, the latter to the strategy grammar. There is no principled difference between instances built bottom up by image interpretation and those constructed previously and now extracted from drawing memory. We have therefore been able to use the strategy rules, without change, to manage the insertion of previously stored instances into the current schema. For example, the strategy grammar includes the rule

witness : line BREAK junction {wit.instantiate()};

which states that a witness instance may be instantiated and made current by location of suitably related [1,2] line and junction instances. All that need be done to incorporate data returned from drawing memory is to add an extra clause;

witness : WITNESS;

Together, these allow a witness to be made current either bottom up or via the extraction from memory of a complete witness instance. All subsequent rules dealing with the non-terminal 'witness', eg.

dimension : leader ABUTS witness {dim.instantiate()};

apply regardless of how any given witness was obtained. ANON's interpretation of image and memory is therefore very closely integrated, not only during search and tokenisation, but at the strategy level as well.

COINCIDENCE LINKS

The practical goal of this research is the identification of interactions between the current schema and those already located and now held in drawing memory. Given the closely linked image and memory searches described above this is easily achieved. Whenever a previously stored instance is reactivated, drawing memory simply returns a copy of that instance augmented with a "coincidence link" to the original, which remains in memory. The copy may then be incorporated into whatever construct is developing at that time while retaining an explicit link to its previous, possibly conflicting, interpretation.

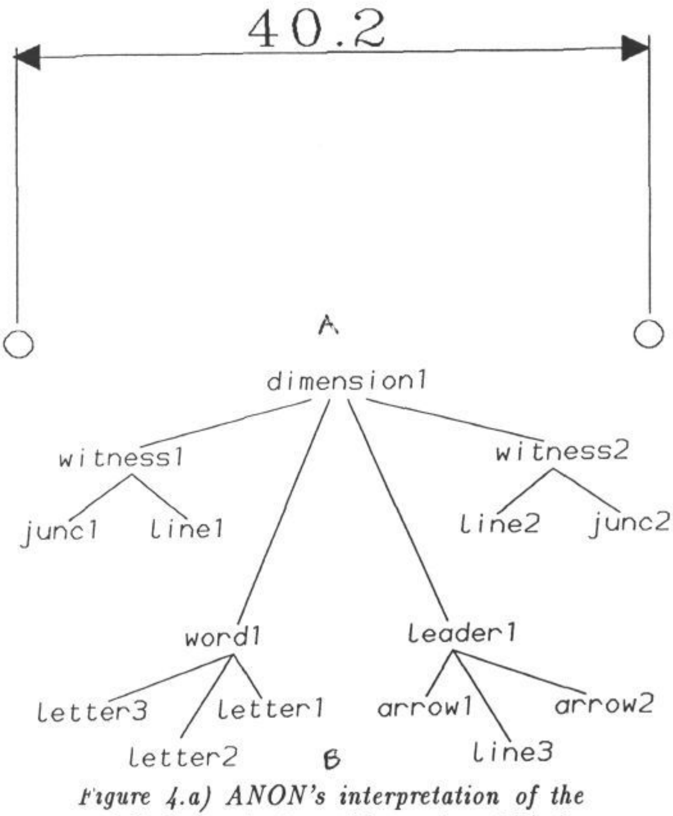

Figure 4.a) ANON's interpretation of the upper left dimensioning of figure 1 and b) the underlying hierarchical representation.

Consider the two nested dimension sets sharing a witness line at the top left of figure 1. Figure 4a depicts ANON's interpretation of the uppermost construct while figure 4b shows the hierarchical description underlying this display. Once the dimension instance is complete it is placed in drawing memory and so becomes available to the current schema on subsequent perceptual cycles. When attention falls on the second piece of dimensioning it's leader, text and one of it's witness instances are extracted from the image. The second, leftmost witness, however, is obtained, complete, from memory. Figure 5 shows the content of drawing memory after the second dimension has been completed. A coincidence link (figure 5b) is in place between the two overlapping witness instances, which are marked bold in figure 5a.

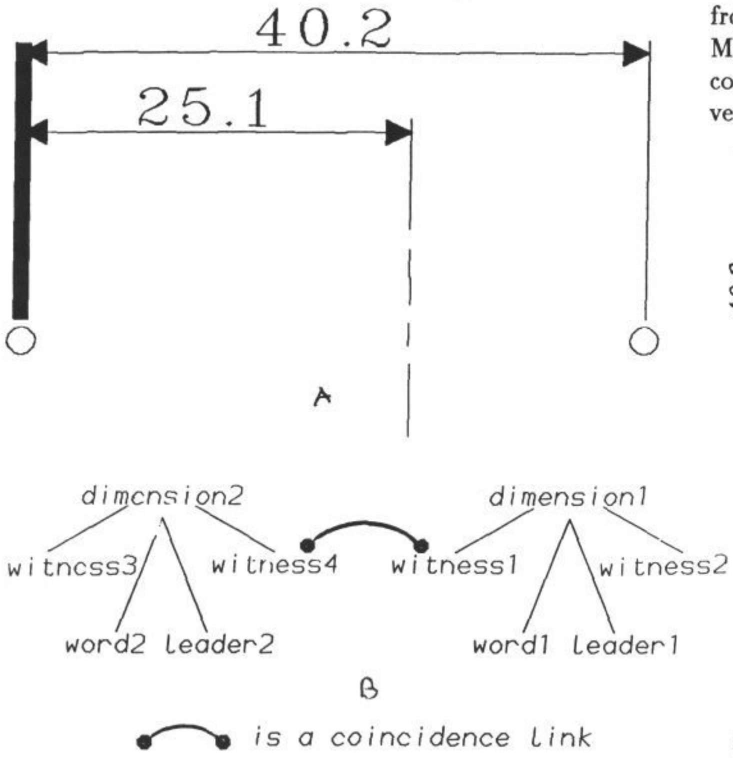

Figure 5. a) ANON's output after completeion of a second dimension instance. b) The corresponding (summarised) schema-based descrption.

Figure 6 shows ANON's final interpretation of the drawing of figure 1, coinciding instances are drawn bold. The system's description of the original drawing corresponds well to human interpretation; high level dimensioning, chain, text and physical outline schemata account for most of the image. Note the presence of coincidence links between all shared witnesses and between the junction instances appearing in both witness and physical outline schemata (marked by black dots). This association of dimension schema with dimensioned object is particularly informative, raising the possibility of computing a more complete description by propagating the given dimensions around the physical outline. Overlaps between chained line schemata contributing to centreline and witness instances are also visible to the right of the figure. Despite being only a step on the way to full schema integration, coincidence links already provide a richer description of drawing content.

PARTIAL INTERPRETATIONS AND SCHEMA FUSION

During image interpretation, the current schema seeks to extend itself by searching for further, or missing, subcomponents. As these are located they are incorporated into the current schema under the control of the strategy grammar. Only constructs at a lower level in the schema hierarchy than the current instance may be added in this way, for two reasons. First, one of our most important design goals is that ANON should exploit local context whenever possible during its analysis of the drawing image. By only allowing the current schema to accumulate lower level instances, we ensure that the schema with the greatest available context is in control. Second, we believe that the sequential processing style that arises from this restriction is natural given the task at hand. Mechanical drawings are man-made artifacts intended to communicate information according to a predefined convention: they are read rather than perceived.

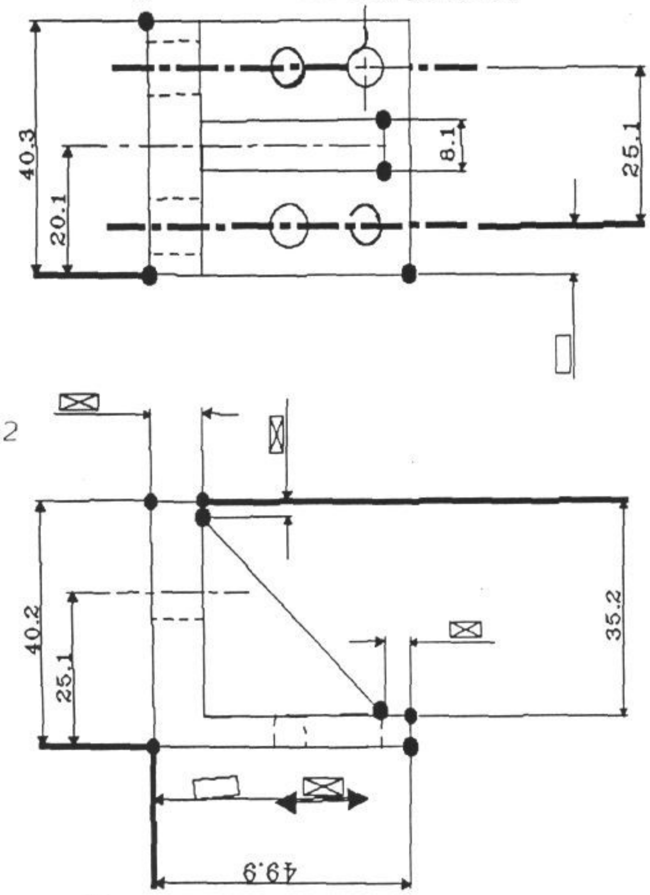

Figure 6. ANON's interpretation of figure 1. Coinciding schemata are drawn bold.

The addition of drawing memory, however, provides ANON with the opportunity to extend this operation somewhat. Top down examination of memory may result in an instance of the same class as the current schema being discovered during a search intended to locate only the next sub-component. Under these circumstances the two overlapping schemata may clearly be merged (cf. [12]) to form a single, usually larger, construct.

Suppose, for example, that a developing cross hatching instance overlaps one already held in memory. The (new) strategy rule

hatch : hatch PARALLEL HATCH {hatch.fuse()};

combines these two schemata to create a single instance representing the full extent of the known hatched region. As hatching accounts for arbitrary areas of the drawing it is impractical to expect entire regions to be extracted, reliably, in one pass. The level of strategic knowledge required would appear prohibitive. ANON's cross hatching strategies are therefore limited to seeking convex regions which may be fused together later to form more general hatched areas. This ability to locate entities piecemeal and assemble a more complete description in memory greatly improves ANON's final interpretation; schema fusion leads to fewer instances, each covering a larger portion of the image. The system is currently able to fuse hatching, physical outlines and centrelines. It should be stressed here that the strategies added to control fusion of like schemata are the only extensions made to the grammar during the course of the present work.

A similar wait-and-see approach is applied to partial interpretations. One of the drawbacks of schema based systems is that instances are frequently created but not completed. Such partially filled schemata can represent significant processing effort, but are often discarded immediately. ANON now defers its rejection of (some) partial instances until processing terminates. Those partly filled schemata that are large enough to warrant further attention (eg. dimensions lacking only text) are held in drawing memory, but labelled as partial. This gives the system every opportunity to exploit such constructs later in the interpretation. Any schemata which remain partial on termination are discarded.

CONCLUSION

Our ability to incorporate a drawing memory into ANON's perceptual cycle with only minimal extensions to the system's knowledge base is encouraging. One can characterise drawing interpretation as the search for acceptable structures in the presence of distractions; when pursuing a given hypothesis it is important not to be lead astray by the extraneous linework and symbols which typically disrupt the target entity. Although it supplies much useful information, drawing memory is also a clear source of additional distractions. That this increase in potentially misleading data does not cause ANON to fail provides further evidence of the success of the basic design.

The extension of ANON's visual search to include access to a visual memory has substantially improved the performance of the system. A richer description of drawing content is now produced which comprises fewer, larger schemata and makes explicit any (potentially) conflicting interpretations. Partial interpretations are stored and retrieved when necessary, reducing the amount of wasted interpretation effort and so improving efficiency.

Despite this, further extensions are necessary. Considerable work remains in identifying suitable methods for the resolution of conflicting interpretations. ANON's explicit representation of overlapping schemata, however, represents a significant step towards the construction of the "scene constraint graph" [cf. 9] which will form the basis of such work. The development of conflict resolution facilities appropriate to ANON is the subject of ongoing research.

In its present form ANON is perhaps closest to the aerial image interpretation system SIGMA [12]; both are schema driven, both use the same knowledge base to mediate access to the image and the developing interpretation, both give previously stored instances precedence over new constructs. The major difference in emphasis is that Matsuyama and Hwang explore decentralisation as a route to domain independent control, but keep to a comparatively restricted hierarchy of schemata, whereas ANON confines control issues to an explicit, domain dependent strategic knowledge base, but explores the problems inherent in a substantial and extendable hierarchy. This differing balance is largely due to the disparate nature of the two application areas.

REFERENCES

1. **S.H. Joseph and T.P. Pridmore** "Knowledge directed interpretation of mechanical engineering drawings" *IEEE Trans. Pami* in press.

2. **T.P. Pridmore and S.H. Joseph** "Using Schemata to Interpret Images of Mechanical Engineering Drawings" *Proc. ECAI-90* Stockholm (1990).

3. **V. Nagasamy and N.A. Langrana** "Engineering drawing processing and vectorisation system" *Computer Vision, Graphics and Image Processing* 49 pp 379-397 (1990).

4. **U. Neisser** *Cognition and Reality: Principles and Implications of Cognitive Psychology* W.H. Freeman, San Francisco (1976).

5. **S.H. Joseph** "Processing of engineering line drawings for automatic input to CAD" *Pattern Recognition* 22 pp 1-11 (1989).

6. **S.J. Cheetham** *The automatic extraction and classification of curves from conventional line drawings* PhD Thesis, University of Sheffield (1988).

7. **S.C. Johnson** "Yacc - yet another compiler compiler" *Comp. Sci. Tech. Rep.* No. 32, Bell Laboratories, New Jersey (1975).

8. **A.R. Rao and A.K. Jain** "Knowledge representation and control in computer vision systems" *IEEE Expert* pp 64-79 (1988).

9. **D. Dori** "A syntactic/geometric approach to recognition of dimensions in engineering machine drawings" *Computer Vision, Graphics and Image Processing* 47 pp. 271-291 (1989).

10. **J. McDermott** "Making Expert Systems Explicit" in *Information Processing '86* ed. H.J. Kugler, Elsevier Science Publishers (North-Holland) (1986).

11. **A.R. Hanson, and E.M. Riseman** "VISIONS: a computer system for interpreting scenes" in *Computer Vision Systems* ed. A. Hanson and E. Riseman, Academic Press (1978).

12. **T. Matsuyama and V. Hwang** "SIGMA: a framework for image understanding - integration of bottom-up and top-down analyses" *Proc. 9th IJCAI* pp 908-915 (1985).

13. **J.A. Mulder, A.K. Mackworth and W.S. Havens** "Knowledge structuring and constraint satisfaction: the MAPSEE approach" *IEEE Trans. PAMI* 10 6 pp 866-879 (1988).