Perceptual Organization as a Foundation for Intelligent Sketch Editing

Eric Saund, James Mahoney, David Fleet, Dan Larner, Edward Lank
Palo Alto Research Center
3333 Coyote Hill Rd., Palo Alto, CA 94304
{saund, mahoney, fleet, larner, elank}@parc.com

Abstract

This paper discusses the design of intelligent sketch editing tools exploiting intermediate levels of visual interpretation known as *Perceptual Organization*. Sketches are much more than an accumulation of strokes: they convey information largely through spatial configurations and patterns *among* markings. We propose that the detection of this visual structure, based on the principles of Perceptual Organization, will make sketch editing tools more useful for a wide variety of hand-drawn and formatted diagrammatic material. We review our efforts which are embodied in a perceptually supported sketch editing tool, called ScanScribe.

Introduction

One application of sketch understanding is to facilitate entering and editing of hand-drawn and formatted figures. Through recognition of image structure, intelligent editors can make it easier for users to select collections of markings that correspond to salient or semantically meaningful entities. Additionally, recognized image structure can be used to modulate the functioning of edit operations, such as by snapping or otherwise enforcing geometric constraints.

Most existing sketch-based systems rely heavily on prior constraints imposed from a targeted application domain. Domain-dependent strategies have been adopted to various degrees in, for example, the design of websites [Lin, et al 2000], the drawing of architecture sketches [Gross 1996], and the entry of mechanical diagrams [Alvarado 2001]. By reducing the space of possible interpretations, domain constraint serves to simplify the system's problem of recognizing structure in raw content and command data. While highly constrained systems do support editing and entry tasks in the domain for which they were devised, these constraints become barriers to sketching outside narrow limits. In a sketch system that interprets and re-renders all four-sided figures as rectangles, it can become impossible to draw a rhombus.

Conversely, unconstrained tools for drawing and editing sketches do not deal with sketch data in meaningful ways. Paint-style programs enable unlimited editing on a pixel-by-pixel basis, but offer little support for selecting salient collections of pixels in sketches beyond rectangle drag and perhaps lassoing selection commands. Digital ink editors (e.g., [Pedersen et al 1993]) sometimes manage to perform simple temporal grouping of strokes, but few of them break strokes into salient fragments or form more complex groups.

To build intelligent sketch editing tools for situations in which there is no *a priori* identified application domain, we believe that the early levels of Perceptual Organization in the human visual system provide inspiration for an approach to the extraction of salient visual structure. Classically, Perceptual Organization has been concerned with the Gestalt Laws of Grouping, such as *smooth continuation*, *common fate*, *symmetry*, *similarity*, *proximity*, and *closure* [Koffka 1922, Wertheimer 1923, Kanizsa 1979, Witkin and Tenenbaum 1983]. More recently, Computer Vision researchers have attempted to model these and related processes computationally (see e.g. [Stevens 1978, Lowe and Binford 1983, Zucker 1983, Boyer and Sarkar, 1999]). Although Perceptual Organization for general visual scenes will remain a daunting challenge for quite some time, intermediate visual structure in many sketches and diagrams is more readily accessible.

Application and Test Platform: The ScanScribe Document Image Editor

Our group has been developing this idea in the context of a perceptually-supported document image editor, called *ScanScribe*, which builds on our earlier work in this area while it extends the range of image material that can be manipulated and the kinds of image structure recognized. Like PerSketch [Saund and Moran, 1994], ScanScribe is designed to manipulate primitive, or *prime* objects, as well as groupings or collections of these, called *composite* objects. These relations can occur as lattice structures and are not limited to tree hierarchies, as shown in Figure 1. While PerSketch was targeted at digital ink manipulation, ScanScribe is designed to work also with image regions as manipulable and groupable objects.

By virtue of placing users in the loop, in an interactive editing application, some burden of accuracy is removed from recognition technologies. Accordingly, UI techniques

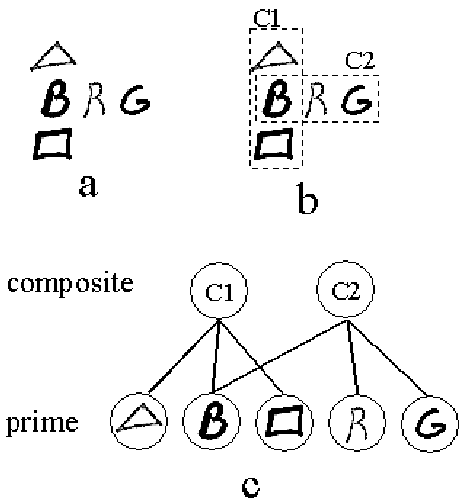

Figure 1: A lattice structure enables prime objects to belong to multiple composite object groups. a. Image. b. Sensible groups. c. Grouping structure.

must enable users to visualize or quickly access visual structures inferred by the system, and to repair or work around mistakes in recognition. To the extent that automatic structure recognition can succeed, we aim to endow the ScanScribe image editor with WYPIWYG (What You *Perceive* Is What You Get) capability that will make editing complex collections of perceptually salient markings as effortless as pointing and clicking.

Key Problem: Facilitating Selection of Meaningful Image Material

ScanScribe is designed to facilitate *selection* of salient image material. Once selected, standard operations of cut, copy, delete, translate, rotate, scale, etc. can all be invoked. We regard the difficulty of selecting image objects that are perceptually meaningful to the user as the primary shortcoming of existing image editors.

By default, when digital ink or a scanned document is loaded into ScanScribe an initial processing step separates foreground markings from background. Thereafter, the background is treated as transparent so that foreground objects may be moved in proximity to one another without occlusion by background white pixels. Users may select image objects by any of four means: rectangle drag, lasso, polygon enclosure, and clicking on established image objects. These selection methods are integrated and may be used seamlessly without the user having to appeal to a menu or otherwise specify any particular selection mode.

Selection of composite objects, or collections of image primitives that form groups, is done by repeatedly clicking on any foreground object. Repeated clicks cycle through the groups that the clicked prime object belongs to. The general question of how to present multiple interpretation options in a graphical user interface remains open. Our experience to date suggests this method to be quite adequate for allowing the user to access and select among multiple interpretations of image structure as long as the number of groups is not too great, and they are all perceptually salient and meaningful to a reasonable degree.

This basic interface supports the use of composite groups even in the absence of any automatic recognition. Once a collection of prime objects has been selected and acted upon, a new composite object is automatically formed to represent this group. The group now becomes available and can be selected by clicking some number of times on any of its prime constituents, depending on its priority in each prime object's list of supported composite objects. Conversely, a group, represented by a composite object, is automatically broken up when a subset of its constituent prime objects are removed from proximity to their former locations, for example by deleting them or dragging them far away.

Groups can also be built and destroyed manually by invocation of explicit "Group" and "Ungroup" commands. Thus the basic ScanScribe editor provides means for users to take ultimate control over any image structure maintained by the underlying program. With this platform we are now prepared to explore and introduce recognition methods that will form groupings automatically on the basis of perceptual organization or other principles.

Finding Visual Structure at the Level of Perceptual Organization

From Computer Vision we discern two basic approaches to beginning to identify visual structure in images. The first is to apply linear or nonlinear operators such as tuned filters of varying scales, exhaustively across the image. Examples are edge and bar detectors [Freeman and Adelson 1991], and spatiotemporal filters for motion detection [Barron et al 1994]. The second approach is to decompose the image into geometric primitives, then perform grouping and matching operations on these to identify composite features and objects. The latter approach leads more directly to workable solutions for a sketch input domain consisting of thin, relatively sparsely distributed curvilinear markings on a unform background. We note however that some "sketchy" styles of drawing such as those illustrated in Figure 2 would be better suited to filter-style detection techniques, and fall outside the capabilities of our current approach.

Segmentation to form Prime Image Elements

Aside from reliance on relatively clean linework, our initial processing stage is indifferent to whether the input is a scanned image or digital ink. This stage is designed to segment prime image objects of two types: (1) relatively straight thin curvilinear fragments unbroken by junctions, and (2) spatially extended regions which are to be treated as unarticulated two-dimensional "blobs". This roughly corresponds to distinguishing the stroke elements of line art from the distinct characters and words of printed and cursive text. While this distinction is not a fundamental one, it is convenient to make and arguably reflects a categorization into the natural kinds that occur in hand-drawn material. People readily decompose sketching activity into diagrammatic “drawing”, and symbolic “writing”. The differing appearance of text and line art pops out to human observers at a glance. From a scene analysis standpoint, different kinds of grouping operations are suited to recognizing textual versus diagrammatic structure; the rules and procedures underlying the analysis of figural shapes, encirclings, arrows, etc., can be rather different from those needed to detect the words, lines, and columns by which text is structured.

At least two challenging research questions confront the segmentation stage. The first concerns the definitional boundary between curvilinear prime objects and blob prime objects. By and large, handwritten and printed text consist of complexly shaped agglomerations of relatively short strokes or stroke segments, while line art consists of relatively longer strokes. Frequently, the shape features of text are commensurate in scale with the width of strokes or font size, while the characteristic radii of curvature, distances between junctions, and spacing of linework in diagrams is substantially greater than the stroke width of lines. Most diagrammatic material satisfies a “thin-line” model: neither visual appearance nor geometric structure and relationships are substantially changed by thinning foreground objects down to skeletons. The thin-line model is manifestly violated by most handwritten and printed text, as witnessed by the degradation of image quality often to the point of unreadability when thinning is performed on text images. The intention of segmentation is to deliver as curvilinear prime objects that image material for which a thin-line model is appropriate, and leave as blob objects that material for which it is violated.

While frequently the curve fragments comprising line art are readily distinguished from blob prime objects, some image objects are ambiguous. In Figure 3 the same image object serves in one context as a fragment of line art, and in a different context as part of a character. It is critical that any categorization be soft enough for classified prime image objects to participate under multiple kinds of structure identification processes.

Conversion of line art into curve objects such as line segments, arcs, splines, or chain code curves is known, in the field of graphics recognition, under the general rubric of “vectorization”. Separation of line-art from blob image objects is handled by ad hoc algorithms. ScanScribe employs a conglomeration of techniques including standard methods

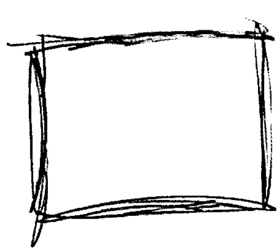

Figure 2: “Sketchy” styles of linework are not segmented into sensible primitives through simple line thinning and tracing techniques.

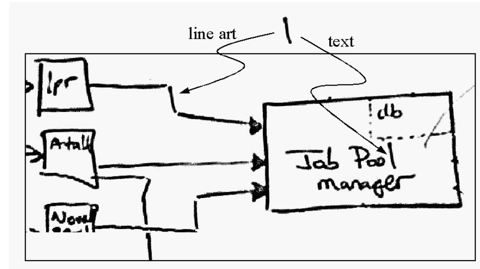

Figure 3: The same image primitive can serve as part of line art or part of a text string in different contexts.

of image morphology, thresholding, boundary tracing, statistical classification, thinning, junction detection, tracing, and corner detection.

A related challenge in segmentation of prime image elements concerns touching or overlapping perceptually distinct image objects, for example a long curvilinear line that touches text. This remains an outstanding problem in the field of document image analysis. The approach we are investigating involves iteration on the steps of fragmentation, then classification of the resulting parts, until prime fragments are returned that appear not to involve mixtures of types as indicated by statical measures on contour properties.

Grouping to form Composite Structures

In ScanScribe, significant perceptual structure in sketches, handwritten notes, or for that matter any document image, is represented by groups of prime image objects represented by composite objects in the structure lattice. While our particular focus is on groups representing structure at the level of Perceptual Organization, this framework supports many recognition methodologies and representational ontologies, including matching to domain-specific databases of shapes, relations, and semantic interpretations.

Some structure recognition algorithms may operate in stages, forming more complex or “higher level” composite objects with the help of intermediate level objects found in turn by grouping primitives. As such, an ontological choice is to be made about the hierarchical structure of support relations. Figure 4 illustrates a simple example in which a “rectangle” is defined as comprising four “line segments”. If the objects at the prime layer themselves do not fulfill the requirements for the rectangle’s “line segment” slots, then some additional computation may be required to introduce an intermediate object creating a line segment out of smaller pieces. There are two ways of handling this, “hierarchical” versus “flat”. In the hierarchical approach the support lattice can grow to any depth. In a flat approach links maintain relations directly between composite objects and their prime level support. In our design of ScanScribe we found it much simpler to manage these as flat relations.

Our approach to detecting perceptual structure follows a basic two-stage strategy employed widely in the Computer Vision field. Within this strategy fall many technical and design options, among which notable aspects of our approach

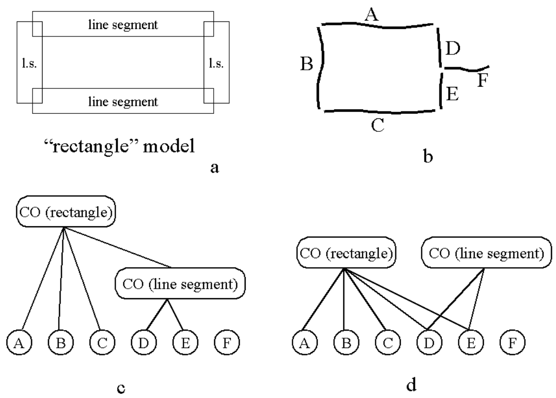

Figure 4: a. "Rectangle" model. b. Prime objects labeled in image data. c. Hierarchical lattice structure showing support relationships between composite objects and prime objects. d. Flat lattice structure.

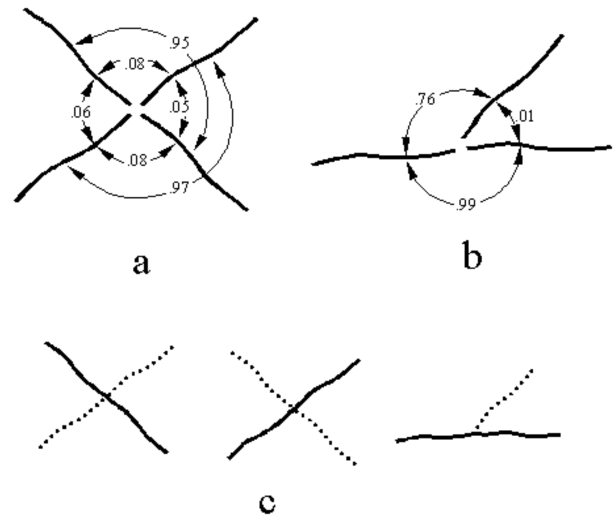

Figure 5: a-b. Hypothetical alignment scores for pairs of curve fragments. 1.0 is perfect alignment. c. Lack of ambiguity makes it easy to decide which pairs should be grouped into composite objects (solid curves).

are discussed below.

The first generic stage is construction of a graph of object relations. Pairs or n-tuples of image objects are identified based on proximity and/or internal attributes. These relations give rise to graph structures, where the nodes are image objects and the links indicate relations. The links may be attributed with properties of the relation, such as distance, relative orientation, etc. Initially these computations are performed on prime image objects, but in subsequent grouping stages these can be composite objects as well. The graph may be used immediately, or it may be massaged based on its local and aggregate properties. For example, two very different forms of this graph massaging are iterative relaxation [Hummel and Zucker 1983] and spectral graph partitioning [Weiss 1999].

The second generic stage is traversal of the graph to collect groups of transitively linked image objects satisfying certain properties. This traversal may include various forms of tracing, coloring, matching, and search.

To date we have applied this framework in separate processes dedicated to identifying visual structure in line art and in text.

Line Art Analysis

Line art analysis operates primarily on curvilinear type prime image objects, or curve stroke fragments. Many of the relevant spatial relations among stroke fragments occur at and among their endpoints. Sets of nearby curve fragment ends are found by clustering. Links are established pairwise among proximal curve fragment ends and labeled with measures of the geometric configuration between them. A score is given for how well each pair forms a corner configuration, and for how well the curve fragments align with one another. This score is based the relative orientations, distances, and other aspects of the local geometry.

We have explored ways of searching for three kinds of perceptual level structure over the link graph of stroke ends. These are based on the Gestalt principles of smooth continuation, continuity, and closure.

Smooth Continuation Paths Smooth continuation search involves tracing paths through the link graph, accumulating stroke fragments that align with one another, end-to-end. These paths reflect the path of the pen across other image material, and commonly reflects semantically significant as well as perceptually salient imagery. The interesting questions are how to decide which paths to follow, and how to handle ambiguity. Figure 5 illustrates. Suppose that alignment score varies from 0 (poor alignment) to 1 (perfect alignment). The pairwise alignment scores shown indicate affinity for each curve fragment to join with the others, that is, to establish a local path through the link graph. Clearly, the pairwise score alone does not dictate whether a particular potential join will be perceptually preferred. The context of other curve fragments must be taken into account. In certain situations a robust strategy is available whereby pairs of curves are joined whose preference for joining with one another exceeds their preference for joining with any other. The clear X in Figure 5a offers such a case. Figure 5b poses a somewhat more equivocal choice. Here, one pair of curves mutually prefer to join with each other, while a third comes into play. However, the alignment score for this third curve is much worse than for the mutually preferring curves, and this join can be safely ruled out. Conditions such as these enable global search routines to construct extended curves by straightforward tracing, yielding the composite curve objects shown in Figure 5c.

Figures 6a and 6b however pose truly ambiguous situations. Local evidence offers no indication as to which of several possible paths through the link graph gives rise to a perceptually salient curve continuation. Because ScanScribe's prime/composite object lattice supports multiple overlap-

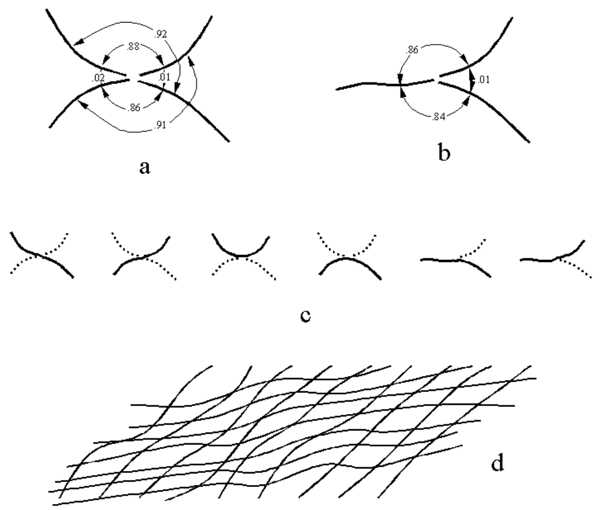

Figure 6: a-b. Hypothetical alignment scores for pairs of curve fragments. 1.0 is perfect alignment. c. Local ambiguity supports alternative composite objects with overlapping support. d. Compounding of local ambiguity can lead to combinatorial explosion in possible number of alternative smooth-continuation paths.

ping interpretations, one approach to these situations is to permit search routines to construct multiple composite curve objects accepting alternative paths, as in Figure 6c. The drawback of this approach is that it can lead to combinatorial explosion in the number of global paths constructed, as in Figure 6d. Heuristic termination criteria offer a stopgap solution. A more principled basis for handling curvilinear paths might involve management of how aggressive or conservative the search is in evaluating local joins as unequivocal versus ambiguous.

Stroke Continuity Paths A second kind of perceptually salient line-art object occurs as a coherent stroke that happens to contain sharp corners. Where such a change of direction presents a clear continuation of the curve, it is meaningful to construct a global path proceeding through such a corner. However, because the corner is in and of itself a perceptually salient potential stopping point for the curve, it is sensible to construct, and make available to users, alternative composite curvilinear objects both bounded by sharp corners, and proceeding through sharp corners. This follows the strategy used in the PerSketch system [Saund and Moran 1994].

Closed Contour Paths A third kind of important perceptual structure in line art is based on the principle of closure. Closed regions are known psychophysically to be perceptually salient to human observers. In sketches and hand-drawn diagrams, closed regions represent individual physical objects; conceptual objects; groupings or collections; logical or other abstract relations; emphasis; looping paths or circuits; symbols and characters (or fragments thereof); and tabular

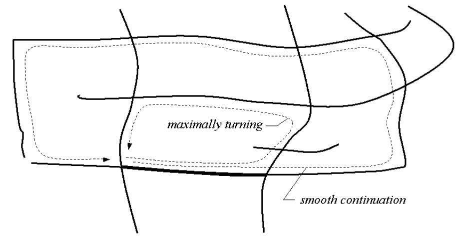

Figure 7: We observe two kinds of figural closure. A *maximally-turning* closure path traces the smallest figure possible. A *smooth-continuation* closure path prefers smooth continuation traces through junctions. Other closed paths through the seed (thick contour fragment) are perceptually insignificant.

cells.

Our detection of closed curvilinear paths reflects an observation that there are actually two kinds of perceptually significant closed path, those that tend to obey the rule of smooth continuation, and those that tend take a turn at every opportunity to enclose the smallest possible region. See figure 7. Accordingly, we construct additional links in the link graph reflecting two kinds of preference for tracing. One kind of link scores the local preference for tracing from one node (curve fragment) to the next under a search for smooth closed paths. A second kind of link scores local preferences reflecting search for maximally turning paths. A bidirectional search procedure conducts best-first search that starts from a seed and explores the link graph in both directions until either a geometrically closed path is found or termination criteria are met. See [Saund 2001] for details.

Interaction of Curvilinear Path Types The most straightforward option for making available curvilinear paths found by the three preceding tracing/search strategies is to make available to the ScanScribe user all constructed composite curvilinear path objects. However, there can be interactions among these such that some paths become less perceptually apparent than others. The recognition of this fact permits either pruning of the options, thereby removing perceptually insignificant choices, or at least demotion of less salient options in the ordering in which they are offered to users by repeated clicking on foreground prime objects.

Figure 8 offers an example. In Figure 8a, the two squares are visually prominent as is the rectangle. A click on the X would sensibly select either the square or the rectangle. In Figure 8c, although closed path search will deliver an analogous set of segments forming squares, the fact that the rectangle's bisector participates in a smooth continuation path reduces its perceived membership as the side of a square. The larger rectangle should be preferred by a user's click on the X. Figure 8e shows that this effect becomes even stronger if the placement of Segment A gives it no special geometric status, e.g. as a perpendicular bisector of the sides of the rectangle. The analysis of such considerations can become quite involved and to date we have not undertaken to formulate a comprehensive approach within ScanScribe.

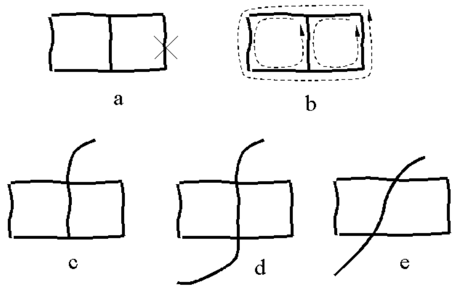

Figure 8: a Both a rectangle and two squares are visually apparent in this figure; all be made available as composite objects (dotted lines in b). These are be found by smooth-continuation search and maximally-turning search in our closed path finding algorithm. c, d. When the bisector becomes part of an extended smooth continuation path the prominence of the squares is reduced. e. The maximally-turning closed paths become even less salient when the bifurcating segment lacks special geometric placement.

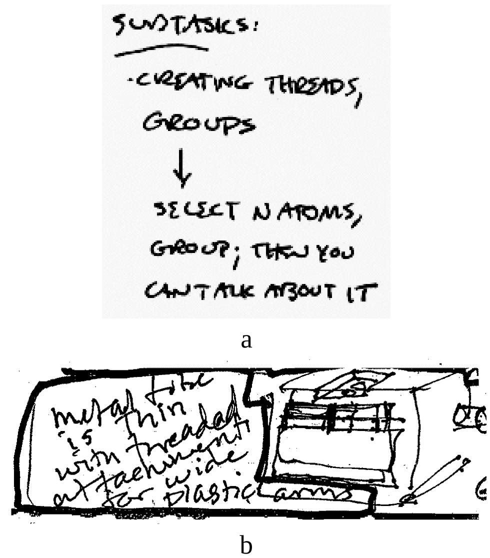

Figure 9: Samples of handwritten text that is moderate (a) and high (b) in difficultly of grouping into words and text lines. Difficulty in a is due to similarity in between-word and within-word character spacing.

Text Layout Analysis

Page layout analysis for printed documents is a major topic of investigation in the field of document image analysis [Nagy 2000]. Universally, strong assumptions are made about the uniform rectilinear arrangement of printed pages, which do not hold for hand-drawn sketches and notes. Much work has also been done on the recognition of hand-printed and cursive characters and words, known in the applications trade as ICR ("Intelligent Character Recognition"). Typically this is done after some preprocessing step or user interface constraint has acted to isolate words.

Our concern in ScanScribe is with the organization of markings into words, lines, columns, tables, and other arrangements of textual information (regardless of script or language), leaving any actual character or word recognition to third-party modules. The difficulty of this problem ranges from relatively easy, for neat, cleanly written, well-separated, horizontally aligned lines of text, to nearly indecipherable cramped, skewed, scratchings with ascenders and descenders overlapping multiple lines. See Figure 9. An issue often neglected is the identification of text groups regardless of the size at which they occur. Size variation arises both from differences in the physical size of writing, and from sensor characteristics such as scanner resolution or spatial sampling of a stylus.

For expediency, the current version of ScanScribe is designed with the assumption that blob prime image objects roughly correspond to individual characters, groups of touching characters within a word, or entire words (e.g. cursive words). In other words, we assume that the segmentation stage has not delivered blob objects corresponding to multiple words on different text lines linked by touching ascenders and descenders.

Following the generic framework outlined above, links are formed between blob prime objects, in this case based on a scale-normalized measure of proximity. Next, grouping takes place through a series of steps involving hierarchical agglomerative clustering, formation of intermediate stage "proto-word" objects from stable clusters, formation of links among these using local proximity and orientation information, and finally assembling collections of proto-words into extended structures that most often correspond to lines of text. The details of this process are continuously in flux as we attempt to improve its performance. A sample result is shown in Figure 10.

As occurs within the realm of line art grouping, text layout analysis raises issues of ambiguous interpretations, multiple valid interpretations, and the interaction of structure found within and between different interpretation modules. Figure 11a shows a case where the context of one text line influences the membership of a blob in another. In Figure 11b a slightly different configuration of these elements yields overlapping, equally salient groups. Figure 11c presents a case in which, by its shape, a prime object demands to be considered for both line art grouping and text grouping. The context of surrounding line art weakens its perceived membership in a text group.

These examples illustrate the kinds of complexities that our future research must grapple with in order to offer users just those groupings, represented by composite objects in the structure lattice, that correspond to the image objects they might want to select, and no spurious others.

Conclusion

At this writing ScanScribe is just beginning to make its way into the hands of (non-researcher) users. While our current line art and text structuring operations are incomplete and of variable performance, they represent an initial attempt

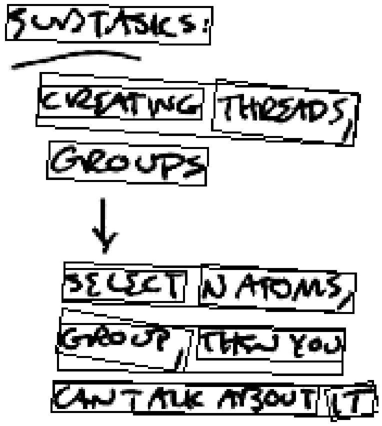

Figure 10: Groups found by ScanScribe's current text grouping algorithm are shown by their bounding boxes

.

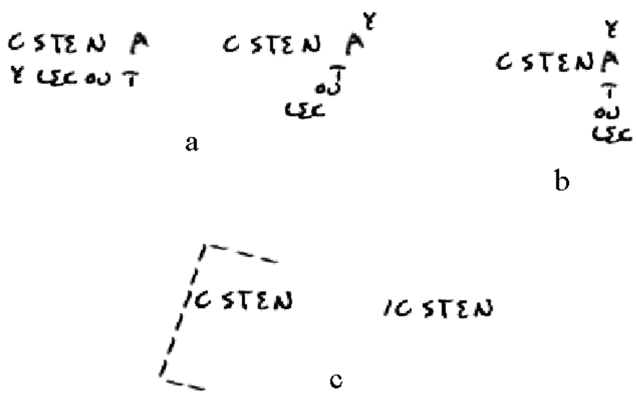

Figure 11: The salience of different groups formed by blob prime objects depends on surrounding context.

to integrate a suite of systematic perceptual grouping facilities into a useful document image editor capable of supporting everyday work. ScanScribe is positioned to become a platform and testbed for further progress on fundamental sketch recognition problems. As such it could amplify knowledge-intensive sketch-based applications for targeted domains [Forbus et al 2001].

By focusing on the detection of visual structure at the level of Perceptual Organization, we propose that intelligent sketch editing tools will become useful for a wide variety of hand-drawn and formatted diagrammatic material. Moreover, we suggest that this step will also lead to a solid foundation for recognizing targeted, domain-specific visual objects and syntax. By endowing sketch editors with perceptual-level WYPIWYG abilities, we recognize that sketches are much more than an accumulation of strokes. They are a particular form of visual scene in which significant structure does not occur on an element-by-element basis, but is found in a rich array of emergent spatial relationships and patterns abstracted from the basic data level.

Acknowledgments. We thank Thomas P. Moran for substantial contributions in the conceptual and initial design stages of this project.

References

Alvarado, C., and Davis, R. 2001. Resolving ambiguities to create a natural computer-based sketching environment. *Proc IJCAI*, Seattle, Vol.2: 1365-1371.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S. 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43-77.

Boyer, K., and Sarkar, S., eds., 1999. *Computer Vision and Image Understanding* Special Issue on Perceptual Organization, 76(1).

Forbus, K., Furguson, R.W., and Usher, J.M. 2001. Towards a Computational Model of Sketching. IUI '01, Santa Fe, New Mexico.

Freeman, W., and Adelson, E.H. 1991. The design and use of steerable filters. *IEEE Trans. PAMI*, 13:891-906.

Gross, M. 1996. The electronic cocktail napkin - computer support for working with diagrams, *Design Studies*, 17(1):53-70.

Hummel, R.A., and Zucker, S.W. 1983. On the Foundations of Relaxing Labelling Processes, *IEEE Trans. PAMI*, 5(3): 267-287.

Kanizsa, G. 1979. *Organization in Vision: Essays on Gestalt Perception*, Praeger, New York.

Koffka, K. 1922.'Perception: An introduction to Gestalt-theorie. *Psychological Bulletin*, 19: 531-585.

Lin, J., Newman, M., Hong, J., and Landay, J. 2000. DENIM: Finding a tighter fit between tools and practice for web site design. *Proc ACM CHI*, The Hague, 510-517.

Lowe, D., and Binford, T. 1983. Perceptual Organization as a Basis for Visual Recognition, *Proc. AAAI-83*.

Nagy, G. 2000. Twenty years of document image analysis in PAMI. *IEEE Trans. PAMI*, 22(1): 38-62.

Pedersen, E., McCall, K., Moran, T, and Halasz, F. 1993. Tivoli: An electronic whiteboard for informal workgroup meetings. *Proc ACM CHI*, 391-398.

Saund, E. and Moran, T. 1994. A perceptually supported sketch editor. *Proc ACM UIST*, Marina del Rey, 175-184.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Transactions PAMI*, 22(8):888-905.

Stevens, K. 1978. Computation of locally parallel structure. *Biological Cybernetics*, 29:19-26.

Weiss Y. 1999. Segmentation using eigenvectors: a unifying view. Proceedings IEEE International Conference on Computer Vision, Corfu, Greece, 975-982.

Wertheimer, M. 1923. Laws of Organization in Perceptual Forms, in Ellis, W., ed, *A source book of Gestalt psychology*, Routledge & Kegan Paul, London, 1938.

Witkin, A.P., and Tenenbaum, J.M. 1983. On the role of structure in vision, in Beck, J., Hope, B., and Rosenfeld, A. (eds.), *Human and Machine Vision*, Academic Press, 481-543.

Zucker, S. 1983. Computational and Psychophysical Experiments in Grouping: Early Orientation Selection, in Beck, J., Hope, B., and Rosenfeld, A. (eds.), *Human and Machine Vision*, Academic Press, 545-567.