

Demonstrating the Electronic Cocktail Napkin

Gross, M. D. and E. Do

*Conference Companion, ACM Conference on Human Factors
in Computing (CHI '96) ,*

pp 5-6.

Vancouver, 1996

design machine group
University of Washington
Seattle WA USA 98195-5720
<http://depts.washington.edu/dmachine>



Indexing visual databases of designs with diagrams

Mark D. Gross

Introduction

Rich visual databases of buildings and places will soon become available in electronic form. Already most universities are digitizing their slide collections and as copyright issues are sorted out, image collections will become available on CD-ROM and over the world networks. Several research prototypes have examined interfaces and applications for computer-based visual collections (Bakergem 1990; Clayton and Wiesenthal 1991), and at least one digital library of architectural images, the Great Buildings Collection, is commercially available on CD-ROM (Matthews 1994). However little work has been done to integrate these collections into computer based design processes, or on extending traditional keyword schemes to index them.

The traditional twentieth century architectural visual database—the slide library—is indexed by a list of keywords, which identify images of buildings and places by name, geographic location, architect, date built, as well as building type and style. Sophisticated indexing schemes such as the Getty Art and Architecture Thesaurus (AAT) provide detailed ways to characterize features of buildings (Getty Trust 1990). However they require of users a great deal of knowledge about the index categories and subcategories. Ultimately these schemes are based on key word search of a text index.

This paper explores the feasibility of a diagram-based visual query scheme to index visual databases for architectural design. In this scheme, an automated visual librarian program finds items in the catalog that match—more or less—a hand drawn diagrammatic query. The paper describes work toward this goal, presenting the results of two pilot experiments and a working prototype of a diagram based retrieval program. The paper is arranged as follows: First, the introduction reviews related work on visual query systems, describes ways slide libraries are currently used, and considers new uses that diagram-based indexing might make possible. The second section describes two pilot experiments that suggest there is enough similarity among architects' diagrams of designs to make diagram based indexing feasible. The third section argues that diagram based indexing is also technically feasible. It presents an 'Electronic Cocktail Napkin' program that recognizes hand drawn diagrams, and describes its mechanisms for graphical search and retrieval. The fourth section describes the application of the Cocktail Napkin program to

build a working prototype of a visual librarian. The paper concludes with a discussion of lessons learned and scale-up problems, and outline directions for further work.

Visual query systems

Visual query schemes have been proposed for visual and non visual databases. For non visual databases, a visual query scheme offers a way to describe—by composing a picture—the structure of the data being sought. A common approach uses a structured graphics editor to construct an entity-relationship diagram of the information being sought (Angelaccio, Catarci et al. 1990; Czejdo, Reddy et al. 1990).

Visual query schemes have also been proposed to access visual data (Chang and Kunii 1981). For example, applications have been built to query image databases in medicine (Assman, Venema et al. 1986), astronomy (Bordogna, Gagliardi et al. 1989), and geography (Jungert 1986; Holmes 1989). One approach is to provide the user with icons for typical image components that can be arranged spatially to compose queries (DelBimbo, Campanai et al. 1992). For example, to retrieve images with a house in the lower left, a car in the lower right, and a tree in the background, one arranges icons for these components. Other approaches that are particularly related to architecture are described in the chapters of this book.

The method described here for accessing a visual database by diagrams has the following characteristics. The content of the database, a collection of digitized images, is entirely visual. Each item in the database is indexed diagrammatically and hand-drawn query diagrams are matched against this index. Thus, queries are not composed from icons and they are not constructed with a structure-based editor. The index is prepared by hand, not automatically constructed by image processing. Unlike more formal schemes such as entity relationship diagrams, the scheme depends on the domain conventions of architectural diagramming.

Uses of a visual database

Most users—typically design instructors and architectural historians—use the university slide library in one of three ways. Most often, they come looking for a specific image that they believe is in the collection. These users know precisely what they are looking for and they can describe it using indexing terms from the catalog. A second mode of use involves less focused search -- a user seeks slides belonging to a certain category—‘wood frame houses in San Francisco from the ‘40’s’—and would like to find several examples. A third mode is browsing the catalog, either beginning from a particular query or entering the catalog at random. These three modes of use are discussed briefly below; it is suggested that diagrammatic indexing can enable new modes of use that would render the image collection more useful for design.

Finding a specific design Experienced slide library users present requests in terms of the text-based catalog: ‘plan and section drawings and photographs of Hertzberger’s Muziekcentrum concert hall in Utrecht, the Netherlands, built in the late 1970’s.’ A query like this is well-formed with respect to the text catalog and the librarian can easily locate the desired images. When the user is so well prepared to use the text index, a diagram based index is not needed. But imagine another user, who knows equally well what she is

looking for, but cannot provide the key words for the text based index: ‘a building by Le Corbusier -- can’t remember the name or the place, but the plan is like this: (Figure 1a)’. A knowledgeable slide librarian immediately retrieves the slide (Figure 1b).

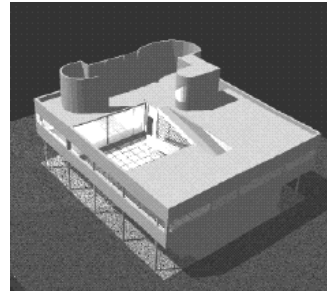
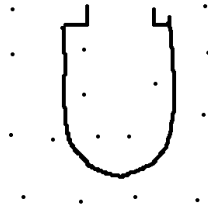


Figure 1: a Villa Savoye sketch

b Villa Savoye slide

However, most slide librarians, though skilled at negotiating the key word catalog, cannot perform content-based retrieval tasks like finding the Villa Savoye from its diagram. On the other hand, users may remember designs primarily by their form characteristics that are most easily expressed in a diagram, rather than by keywords that describe the features and attributes of designs.

Category search A second use of the slide library involves a less specific search, such as ‘find examples of courtyard houses.’ Text based searches require the right set of key words to find slides that match the description. As when searching for a particular image, the user who can formulate a query in terms of index key words can rapidly find a collection of images. If the collection that is returned is unmanageably large, the user will iteratively reformulate the query to focus on the sort of collection she had in mind. Here again the traditional way of indexing the slide library works when a key word query can be well formulated. But, for example, a designer might be looking for buildings with a certain roof form that is most easily described by a diagram. The visual librarian would retrieve images whose diagrams include that roof form.

Browsing Browsing is less common in image collections than in book stacks; nevertheless users often flip through nearby slides in the collection just to see what is there, and even open slide drawers at random to take a look. In a typical organization, nearby slides are related geographically and by designer, for example: buildings by Frank Lloyd Wright in Buffalo, New York. An advantage of the computer based index is that images can be linked together and these links can be navigated, hypertext-style, using a browser. After finding an image with a diagrammatic query, items related to it can be found by browsing relational links among images.

Use of images in design

Text based catalogs, even those such as the Art & Architecture Thesaurus that include keywords that describe content, serve historians better than designers. Yet designers are voracious users of visual references. They adopt and adapt features and fragments of designs from books and magazines. Photographs and drawings of references or

precedents are pinned up at the drawing board and included in project presentations. Nevertheless, the slide library runs a poor second to books and magazines as a source for references. It is easier to copy an image from a book or magazine than to print a slide; however this is changing as image collections go on-line. Another reason is that the library is full of books that contain collections of images, pre-selected according to a range of useful indices, for example stone houses of Normandy, buildings by van Eyck, the Bauhaus. By indexing the image collection with architectural diagrams, the slide library can be made more accessible to designers, who are already active users of images but for whom a solely text based index is an obstacle to finding relevant items.

Do architects make the same diagrams?

The success of a diagram-based indexing scheme hinges on the consistency of diagrams that users will make. If we can rely on designers to make rather the same sorts of diagrams then it is a good bet that a diagram based index will work. On the other hand, if the designers' diagrams vary widely, then a diagram based index will fail. It seems reasonable to expect designers to make similar diagrams, if only because diagramming is an essential part of architectural education and practice (Goldschmidt 1989), and diagrams are frequently used in the classroom and in textbooks (Clark and Pause 1985) to explain designs.

To explore this question two pilot experiments were performed, which provide anecdotal evidence that architects will make similar diagrams to represent designs and images. In the first experiment, fifty undergraduate design students made quick sketch diagrams of ten images. In the second experiment, experienced designers made diagrams—from memory—of three designs from the architectural canon: Le Corbusier's Villa Savoye, his chapel at Ronchamps, and Wright's Guggenheim museum.

Experiment 1: sketch from slides

This experiment asks: do designers make the same diagram of a given image? Fifty architecture students participated in the experiment, conducted at the beginning of a lecture in a design theory and methods course. They were asked to make quick sketches or diagrams of ten plans and photographs. The students were given the following instructions:

We're going to look at ten slides. The first five images will be drawings, and the second five will be photographs of buildings or places. I'd like you to make a sketch or a diagram of each slide that might help you remember the image, or that you feel in some way summarizes what you see. They will go by rather fast. You'll have a total of forty-five seconds for each slide. I'd like you to take the first fifteen seconds to look at each picture and think about it, and then use the remaining thirty seconds to make a sketch or diagram. Please be bold in making your diagrams. Please make your marks definite, not tentative. Don't rush, but try to be deliberate in making each diagram. There are no 'right' or 'wrong' ways to do this -- I'd simply like to see how you choose to represent each of these images. Are there any questions?

The subjects asked no questions and they seemed to have no difficulty carrying out the exercise. (For architecture students, sketching from slides is a typical activity). Ten images (five drawings and five color photographs) were shown. Figures 2a and b show two of the images (a plan drawing and a photo) with sketches the students made.

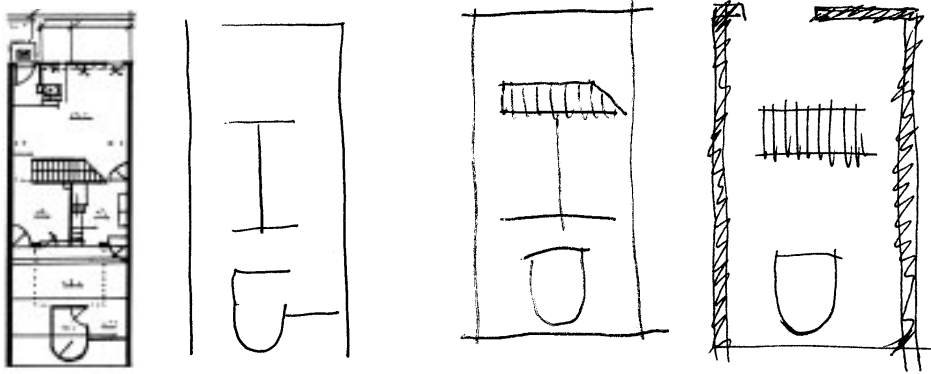


Figure 2a: Plan of Le Corbusier's worker housing at Pessac

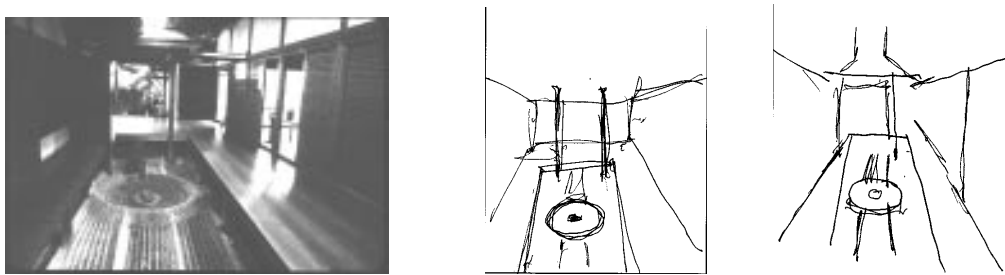


Figure 2b: Photograph of Japanese rock garden

Examination of the fifty sets of sketches reveals strong regularities in the features that the subjects chose to represent as well as the methods they used to represent each feature. Each sketch included three to five main features (Figure 3). The method used to represent each feature was chosen from two or three possible methods. In identifying features and drawing methods, simple geometric configurations were chosen that a computer program could be programmed to recognize. Most sketches of Le Corbusier's worker housing at Pessac represented the following features:

- (1) two party walls (100%)
- (2) I-shaped kitchen-stair wall configuration in the drawing's center (100%)
- (3) U-shaped enclosure at the lower part of the drawing (94%)
- (4) top horizontal closure line (92%)
- (5) bottom horizontal closure line. (48%)

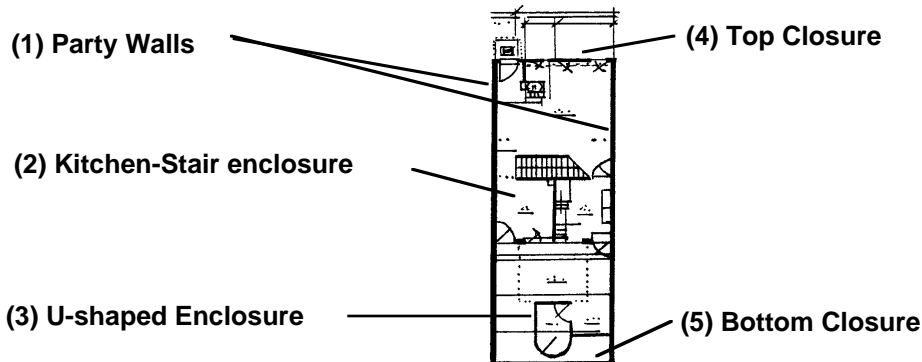
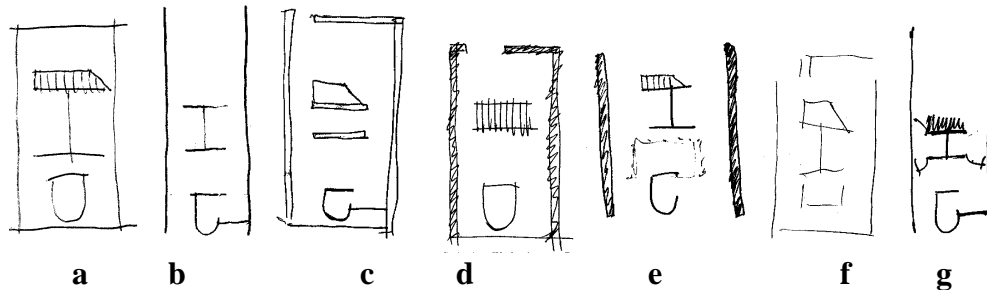


Figure 3: Features identified in Pessac plan drawing

Most subjects used one of two or three drawing methods to represent each feature. For example consider feature 2, the ‘I’ or ‘H’ shaped partition that encloses a kitchen and provides the wall for a stair. Every diagram included some representation of this feature in the center of the drawing. The most common drawing method was a three-line ‘I’ with an indication of a stair (29); a significant number (12) of sketches simply drew the ‘I’, omitting the stair (Figure 4b). Three sketches showed two walls and a stair, omitting the center wall shown vertically in the plan (Figure 4c); two sketches showed only the stair (Figure 4d) and one sketch used an altogether different drawing method.

Various methods were used to represent the stair. Of the twenty-nine sketches that indicated a stair the vast majority (23) indicated the stair as a small sequence of short lines (Figure 4e); however three simply showed a rectangle or trapezoid (Figure 4f) outlining the stair, and three used a scribble to represent the stair (Figure 4g).



Figures 4: Methods for representing the stair assembly

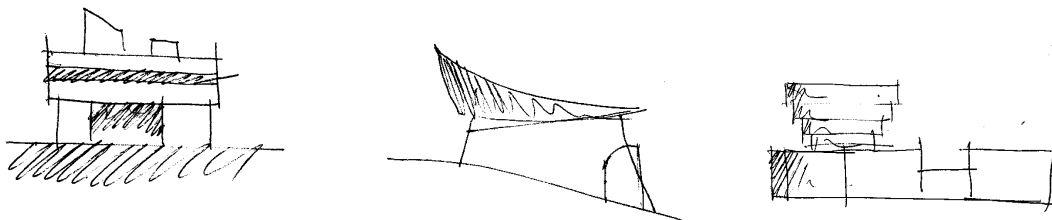
These anecdotal data suggest that architects recognize more or less the same features in a drawing or a photograph, and that they use a small set of drawing methods to represent these features. These results may seem unsurprising: designers see more or less the same things and produce more or less the same kind of marks on paper. Yet this is exactly what is needed for a diagram-based indexing scheme to work.

Diagrams from memory: Canonical diagrams for well-known designs?

The ‘sketch from slides’ experiment suggests that architects make similar diagrams to represent a given image. But in querying a visual database designers won’t be viewing and

summarizing a picture. Therefore, it is also important to explore whether architects would produce similar diagrams for the same design, from memory. Three buildings in the architectural canon were identified that many architects know well enough to diagram from memory—Le Corbusier’s Villa Savoye, his chapel at Ronchamps, and Wright’s Guggenheim museum in New York—and designers were asked to draw:

Please produce, simple diagrams or sketches of the following three buildings. Keep your diagrams simple, using a small number of lines or marks, and please do not consult any references.



**Figure 5: Designer’s diagrams (from memory) of well-known buildings
(a) Villa Savoye; (b) Ronchamps; (c) Wright’s Guggenheim Museum**

The diagrams in Figure 5 are typical examples made by an experienced architect. As with the sketches from slides (for which a great deal more data was collected), the results included several different representational schemes for each design. For example, some designers drew a diagram of the Villa Savoye plan, while others drew an elevation view. On one hand it is clear that each entry in a visual database cannot be indexed with a single diagram; on the other hand, the sketches gathered suggest that a relatively small number of diagrams will cover each design. Figure 6 shows a more complete sample of the diagrams collected.

The two pilot experiments suggest that architects will make similar diagrams of a design, that is diagrams that contain more or less the same graphical marks in the same spatial relationships. These data are encouraging that diagrams could be the basis of a successful indexing scheme for a visual database in architecture.











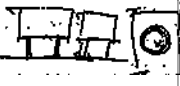



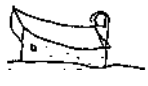


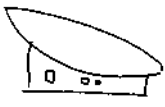

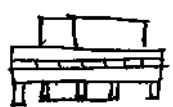

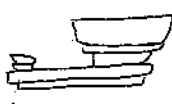















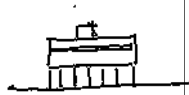



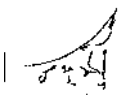



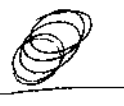
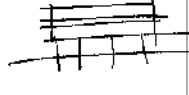
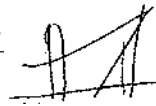

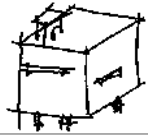
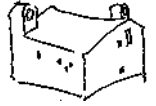
Guggenheim	Savoie	Ronchamps		Guggenheim	Savoie	Ronchamps
						
						
						
						
						
						
						
						
						

Figure 6: Diagrams of famous buildings from memory

3 The Electronic Cocktail Napkin

The Electronic Cocktail Napkin (Gross 1994) is a program that reads and recognizes simple hand-drawn diagrams, converting them to symbolic descriptions of glyphs and spatial relations. Both the simple glyphs and the higher level configurations of glyphs that the program recognizes can be easily trained by users. The program can search a catalog for diagrams that match a given input diagram. In addition to the visual database project, the program is being used to develop a prototype intelligent sketchbook and as a diagram index to access a case based aid for architectural design (Gross, Zimring, and Do 1994).

Diagram = glyphs + spatial relations

The program (written in Macintosh Common Lisp) recognizes a diagram as a set of simple symbols (glyphs) and a set of spatial relations between them. A low level recognizer identifies individual hand drawn symbols by comparing them against previously trained samples. The user can train the program to recognize new glyphs interactively, simply by drawing examples and identifying them. Therefore the elements of diagrams are not limited to a fixed set, but the set of recognizable glyphs can be easily extended.

The low level recognizer identifies hand drawn input as matching one of a stored set of characters, or glyphs, much like other similar algorithms (Rubine 1991; Zhao 1993). The algorithm used here is similar to Teitelman's and other early recognizers, described in Newman & Sproull's book on computer graphics (Newman and Sproull 1973). It uses shape, number of corners, and number of strokes to identify glyphs. It first constructs a 3x3 grid in the bounding box of the glyph. The glyph shape is the path through the grid squares traced by the pen. Corners are detected where the pen slowed down and input points are close together. The stroke count is easily computed from the pen up/down input data. The low level recognizer identifies glyphs in real time (on a Macintosh IIfx or Powerbook 160); currently it identifies about 60 different glyphs including capital letters and numbers.

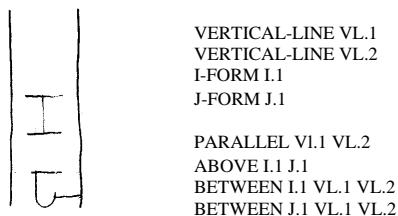


Figure 7: A diagram and its symbolic description

After recognizing the individual glyphs in a diagram, the program identifies spatial relations among the glyphs and produces a textual or symbolic description. The program looks for intersection, containment, relative sizes, adjacency, parallel, and other binary spatial relations. The symbolic description names each glyph and lists the spatial relations (Figure 7).

Higher level configurations and graphical search

The program can also be trained to identify configurations, that is sets of glyphs in particular spatial relations. For example (Figure 8), a 'tree recognizer' identifies configurations that comprise circles or boxes arranged one above the other, with line segments connecting them. A floor plan is a collection of adjacent boxes containing words (arrangements of letters in a row). A poly-line is a collection of line segments with ends that connect or tee. The program recognizes configurations by matching the set of elements and relations in the search pattern against candidate element sets in the diagram.

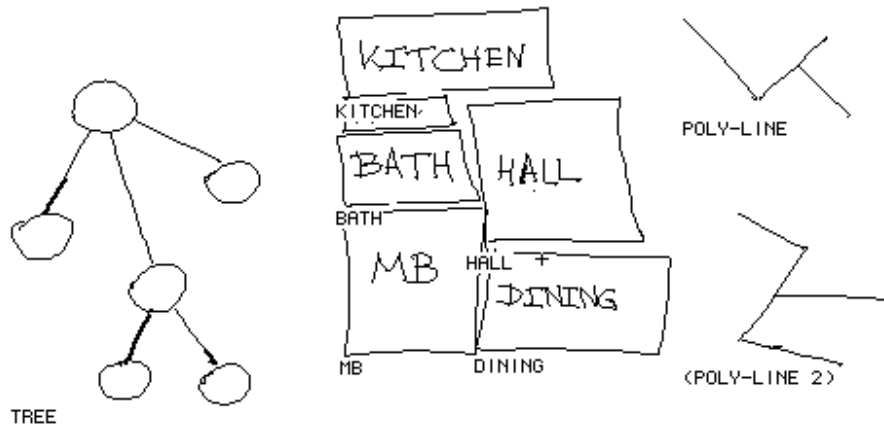


Figure 8: The program identifies configurations of elements and relations

A search pattern, or configuration, is defined using a simple interactive dialog. First a set of glyphs on the screen are selected, and the program identifies their spatial relations using the search dialog. Then the user can adjust the search pattern by (1) deleting glyphs and spatial relations that are not intended to be part of the search pattern and (2) changing the descriptions of glyphs and spatial relations to be more or less specific. For example, Figure 9 shows the search pattern identified for a configuration of glyphs named HOUSE that consists of two boxes and a triangle.

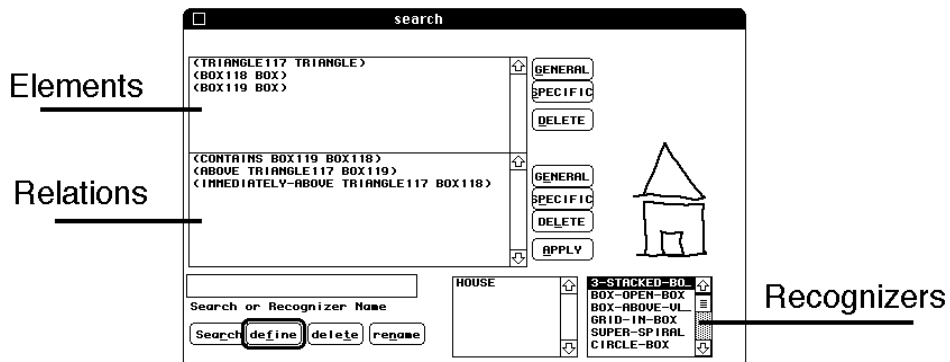


Figure 9: Search patterns are adjusted by the user

Query by diagram

A prototype query scheme was developed that uses the Cocktail Napkin program as a front end to a small visual database of ten famous buildings. The interface for retrieving images is shown in Figure 10. The user draws a diagram on a sketch pad query window. The program parses the query diagram and compares it to the diagram index. It finds and displays the closest matching image or images in the visual database.

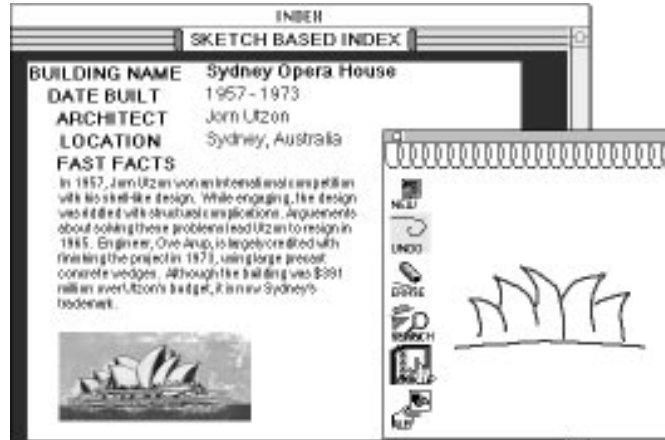


Figure 10: Diagram interface to a visual database of famous buildings

The image database in this prototype was built in FileMaker Pro. Each entry consists of a scanned photograph and a brief text description of the building. To construct the index, twenty architects and architecture students were asked to draw diagrams of these buildings from memory, and the results of this survey were used to identify two or three canonical diagrams for each building. For example, the index used three different canonical diagrams (see Figure 6) for Wright's Guggenheim Museum: a two-dimensional spiral (a diagram of the floor plan), a helix (which represents the three-dimensional form of the building), and three stacked boxes (which represents a salient feature of the front elevation). The canonical diagrams were run through the Cocktail Napkin's recognizer to produce symbolic descriptions of their glyphs and spatial relations. The diagrammatic index consists of a table of these symbolic descriptions and the names of the images.

It was convenient to identify configurations, or groupings of elements within the diagrams. For example, in making a diagram of the Parthenon, most designers drew four or five vertical lines to represent the building's columns. Therefore the Cocktail Napkin was programmed to recognize a sequence of four or five vertical lines as a configuration. In other words, constructing and recognizing intermediate level configurations the matching scheme proceeds by parsing the input diagrams. For a Parthenon diagram for example, the index contains the symbolic description:

*(and (Triangle immediately-above Vertical-lines)
(Vertical-lines immediately-above Base))*

where *Vertical-lines* is described as a collection of *Vertical-line* elements *immediately-right-of* one another, and a *Base* is either a *Box* or a *Lne*.

The collection of recognizers that identify lower level collections of diagram elements and spatial relations as instances of named configurations constitute the replacement rules of a grammar. Although the matching could take place at the lowest level of input glyphs and relations, it seems sensible to add a layer in which some simple and common configurations are recognized. By hierarchically structuring the matching the number of low-level glyphs that must be matched at one time is reduced, and diagrams are organized in a way that seems logical as well. However, matching and parsing is internal to the program and the index user need not understand how diagrams are structured or parsed.

Figure 11 illustrates the diagram-based retrieval process. The user draws a query on the sketchpad and the Cocktail Napkin's recognizers construct a symbolic description of the diagram. The matcher then compares the symbolic description against a table of previously stored symbolic descriptions. If any descriptions match, the corresponding image is retrieved from the file and displayed on the user's screen.

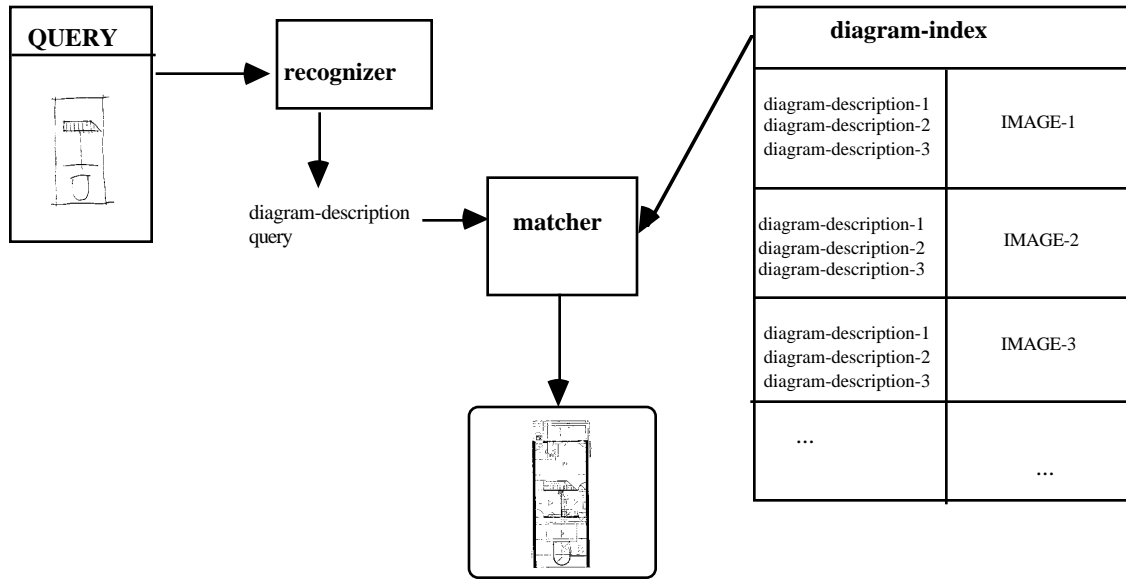


Figure 11: Overview of diagrammatic retrieval

Discussion

The two pilot experiments asking designers to draw diagrams from slides and from memory suggest that a diagrammatic index can work for indexing visual databases of architectural designs. The prototype visual librarian, built using the Cocktail Napkin program for recognizing hand drawn diagrams, suggests a technical approach that might be feasible. However, before trying to build a large and fully featured diagrammatically indexed visual database, a great deal more work should be done, both empirical work to test the conventionality of architectural diagramming, and engineering work on diagram recognition and interpretation. The scheme presented here depends intrinsically on the

order in which the strokes of the diagram are drawn. It is not easily adapted to recognize scanned images, as for example in Koutamanis's approach to retrieval (described in this volume) (Koutamanis 1995).

An obvious question is whether this approach will scale up to query collections with hundreds, or even thousands of images. As a first step in this direction, the Cocktail Napkin will be used to build a diagram scheme to query the Great Buildings Collection, which contains over 700 buildings. The diagrams used to index the prototype image collection are simple, so that many buildings would have the same diagram. That is, in drawing diagrams from memory designers cannot be expected to recall and represent subtle differences in building plans or elevations, but only to represent the most salient building features. The same simple diagram made for the Parthenon might be used to describe all Greek temples, and might even retrieve Neoclassical interpretations as well. Therefore it seems likely that a diagram index can only retrieve sets of images, to be further filtered by the user. Subsequent filtering can be done either by reformulating the diagram query on the reduced set, or by augmenting the diagram query with key word matching.

One significant obstacle has not been discussed: to access a large image collection by diagrams it would first need to be indexed with diagrams. This could be a tedious task. This paper has avoided proposing a scheme for deriving diagrams from images automatically but for very large image databases this would have to be considered. A completely automatic diagramming system that takes images as input and produces architectural diagrams seems beyond the state of the art. However, a halfway measure can be imagined, an intelligent assistant that provides a kit of useful diagramming tools for the human diagrammer. These might include edge-finding and region finding routines that process the image and provide line-drawing summaries and that simplify complex polygon lines into diagrammatic primitives. For experimental work and for reasonably small databases, it will be possible to index the images by hand.

It is also fair to question the advantage of using hand-drawn queries over composing queries by assembling icons in a structured editor. Since the approach described here translates the hand-drawn queries into symbolic descriptions of elements and spatial relations, just as the icon-based approach, any difference must lie in the pen-based human-interface. Designers are accustomed to making diagrams, so the pen-based interface provides a natural way to compose a visual query; an icon-based or structured editor approach seems to impose an unneeded barrier. Also, the diagrammatic query scheme is being developed along with other work on the use of diagrams and sketching in design. An important goal is integrating visual databases into design environments; hand-drawn diagramming may be an effective method.

Acknowledgments

Ellen Do, Aaron Fleisher, Vinod Goel, Gabriela Goldschmidt, Ray McCall, and Craig Zimring were helpful in arguing about these ideas. Kristin Mayfield Anderson helped construct and test the prototype. Thanks also to the various diagrammers, and to Alexander Koutamanis for suggesting the idea. Material support was provided by US National Science Foundation grant DMII-9313186.

References

- Angelaccio, M., T. Catarci, et al. (1990). "QBD*: A Fully Visual System." Journal of Visual Languages and Computing 1(2): 255-273.
- Assman, K., R. Venema, et al. (1986). The ISQL Language - A software tool for the development of pictorial information systems in medicine. In Visual Languages, Chang, Ichikawa, and Ligomenides (eds) 1986, Plenum. 261-284.
- van Bakergem, D. (1990). Image Collections in the Design Studio. In The Electronic Design Studio: Architectural Knowledge and Media in the Computer Age, McCullough, Mitchell, Purcell (eds). Cambridge, MA, MIT Press. 261 - 272.
- Bordogna, G., I. Gagliardi, et al. (1989). Iconic queries on pictorial data. IEEE Workshop on Visual Languages, Rome, IEEE Press. 38-42.
- Chang, S. K. and T. L. Kunii (1981). "Pictorial Database Systems." IEEE Computer (November): 13-21.
- Clark, R. H. and M. Pause (1985). Precedents in Architecture. New York, Van Nostrand Reinhold.
- Clayton, M. and H. Wiesenthal (1991). Enhancing the Sketchbook. ACADIA '91, Los Angeles, CA, ACADIA.
- Czejdo, B., V. Reddy, et al. (1990). An Interactive Graphical Query Interface for a Relational Database Management System, in Visual Languages and Visual Programming. S.K. Chang (ed) New York, Plenum. 289-302.
- DelBimbo, A., M. Campanai, et al. (1992). Using 3D Spatial Relationships for Image Retrieval. IEEE Workshop on Visual Languages, Seattle, WA, IEEE. 184-190.
- Getty Trust, J. P. (1990). Art & Architecture Thesaurus. New York, Oxford.
- Goldschmidt, G. (1989). "Problem Representation versus Domain of Solution in Architectural Design Teaching." Journal of Architectural and Planning Research 6(3): 204-215.
- Gross, M. D. (1994). Recognizing and Interpreting Diagrams in Design. Advanced Visual Interfaces '94. ACM Press. 88-94.
- Gross, M. D., C. Zimring, et al. (1994). Using Diagrams to Access a Case Base of Architectural Designs. in Artificial Intelligence in Design '94. J. Gero (ed). Lausanne, Kluwer. 129-144.
- Holmes, P. D. (1989). Visual Reply to Map-Related Queries -- A Free Space Graph Approach. IEEE Workshop on Visual Languages, Rome, IEEE. 92-98.

Jungert, E. (1986). A Conceptual Image Data model for Image Information Retrieval. in Visual languages, Chang, Ichikawa, and Ligomenides (eds) New York, Plenum. 285-301.

Koutamanis, A. (1995). Recognition and retrieval in visual architectural databases. in Visual Databases in Architecture. Koutamanis, Timmermans, and Vermeulen (eds) Avebury.

Matthews, K. (1994). Great Buildings Collection. New York, Van Nostrand Reinhold.

Newman, W. M. and R. F. Sproull (1973). Principles of Interactive Computer Graphics. New York, McGraw Hill.

Rubine, D. (1991). "Specifying Gestures by Example." Computer Graphics **25**(4): 329-337.

Zhao, R. (1993). Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors. INTERCHI '93, Amsterdam, ACM / Addison Wesley. 95-100.