

Drawing Analogies - Supporting Creative Architectural Design with Visual References

Gross, M. and E. Do

in *3d International Conference on Computational Models of Creative Design*

M-L Maher and J. Gero (eds)

Sydney: University of Sydney, 37-58.

1995

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



DRAWING ANALOGIES

Supporting Creative Architectural Design with Visual References

ELLEN YI-LUEN DO

*College of Architecture, Georgia Institute of Technology, Atlanta GA
30332, U.S.A.*

AND

MARK D. GROSS

*Environmental Design, University of Colorado, Boulder CO 80309-0314,
U.S.A.*

Abstract. Creative architectural designers often employ forms from other domains, retrieving, adapting, and including them in designs in whole or in part. We focus here on retrieval, searching image collections for relevant forms to support creative designing. Retrieval of images may be indexed on conceptual design features, on function, and (in domains that involve a physical artifact) on visual similarity, or shape. We propose an approach to providing information for creative design that is simple to use and integrated with the act of creative designing, which in architecture is carried out chiefly through sketching and diagramming. We argue that to inspire creativity, image collections from diverse domains can be valuable to the designer. We describe Drawing Analogies, a sketch-based visual reference program.

1. Introduction

We want to use sketching to help designers find images as visual references for creative design. Designers sketch; they seek forms from outside their design domain; and they incorporate the forms they find into their designs. These observations lead to several questions: How do designers find and use relevant forms? How might we associate a set of sketches with diverse image collections? How might we automate the visual indexing of image collections? By providing access to different collections through the act of sketching, we can help designers find interesting and sometimes useful references for creative design. We describe here a prototype computational tool that provides images based on their visual similarity to the designer's sketch. We are building design environments for architects; therefore our

examples come from architecture. However, studies of drawing in engineering design (e.g. Ullman 1989; Ferguson 1992) suggest that our approach is not inherently limited to architecture.

The first section of the paper outlines our argument, reviewing examples in architecture of visual analogy, metaphor, and 'shape borrowing'. Both in architectural pedagogy and in accounts of practice, copying and adapting forms plays a prominent role. The second section describes the Drawing Analogies program, which relies on sketching and diagrams to query and retrieve images from diverse databases. We describe our sketch based design environment and its visual reminding system, and discuss measures used to assess the similarity of diagrams. The third section discusses issues of diagram making and the use of visual references. We conclude in the fourth section with a summary and point out connections to our continuing work on constraint based and case based design.

1.1. CREATIVITY AND THE ROLE OF VISUAL REMINDING

We would develop tools to support creative design in architecture. By 'creative design' we simply mean design in which neither the solution nor an algorithm to achieve the solution are immediately obvious, in which the designer must determine an initial set of concepts as a basis for design development.

Among accounts of human creativity (e.g. Boden (1990), Koestler (1964)) several processes are frequently described: 1) combining ideas from different domains; 2) the use of visual imagination, metaphor, and analogy, and 3) expanding and varying the search space of alternatives. Our system for providing designers with images is predicated on the hypothesis that creative design in architecture employs these processes; therefore we begin by briefly reviewing these three aspects of creativity.

1.1.1. Combining ideas from different sources

An often mentioned source of creativity is the combination of ideas, or as Koestler termed it, "bisociation" (Koestler 1964). Creative ideas are derived by combining borrowed ideas from other domains with the problem at hand. For example, Koestler describes Gutenberg's invention of the printing press as combining the idea of seals used to press letters into wax with a wine press to print many letters simultaneously onto paper. Accounts of creative architectural design are replete with borrowing and adapting forms from other buildings as well as from natural and human artifacts.

Boden suggests that creativity involves "recognizing analogies" (p 12) "unusual juxtaposition of ideas" (p 30) "produced by reference" (38) solving problems, exploration and evaluation (p 47, 63). Reviewing Karmiloff-Smith's experiments on creative children's drawing, Boden concludes that the

general strategies used are "distorting, repeating, omitting or mixing parts chosen from one or more categories." (Boden 1990; p 71).

1.1.2. Visual metaphor and analogy

In many accounts of creative acts visual metaphor or analogy—'seeing as'— plays a key role, for example Kekulé's seeing benzene's carbon ring structure as a snake biting its tail, or Faraday's seeing the universe as patterned by "lines of force," which led to the electric motor (Koestler 1964). The use of analogy and metaphor features prominently in many discussions of design methods and processes, e.g. (Heath 1984; Rowe 1987). "Analogic design," says Broadbent, is the "most potent source of creative ideas in architecture." (Broadbent 1973; p 35). Architects are visually oriented and are taught to think graphically (McKim 1972; Laseau 1980). For architects therefore visual analogies are especially important and are commonly used in professional design education. Pictures and sketches of design analogs appear frequently in studio presentations. Instructors encourage students to use analogy in developing creative designs ("think of your building as a string of pearls"), applying the analogy to drive the shaping of physical form ("the rooms are pearls, a connecting path the string.") This particular analogy is used not only in design pedagogy, but also in design practice. For example, Frederick Law Olmstead described his design for a connected system of parks around Boston, Massachusetts as an "emerald necklace."

Goldschmidt (1992, 1994) has looked closely at the logic that connects sequences of architectural sketches and at the intertwined roles of drawing and visual analogy. She argues for a careful use of the term 'analogy' (as opposed to 'metaphor'): in analogy, structural rather than surface features of designs are engaged, and the analogical form 'A is to B as C is to D' can be identified. She provides examples drawn from case studies and talk-aloud protocols. From this standpoint much of what is often seen as visual analogy in architecture may be better understood as metaphor and shape borrowing. Fish and Scrivener (1990). have also examined the cognitive functions of sketching in design, with an eye toward supporting this activity with computers.

1.1.3 Expanding the search space

In the state-space model of design prevalent in much artificial intelligence work, creativity is seen as expansion of the search space. Newell, Shaw, and Simon (1962) describe creative thinking as a special case of problem solving, to be worked by heuristic search. In engineering design, Gero (1994) suggests that expanding the space of design possibilities with larger knowledge bases can inspire creativity. The search space can be expanded

both by relaxing constraints on existing parameters and by introducing new search space dimensions, viewing design not as simply search but as exploration (Gross 1987; Logan and Smithers 1993). Analogical reasoning is one way to expand the search space. For example, Gero and Maher (1992) propose an analogical mechanism for introducing design elements from other domains to encourage creative 'bisociation'. Navinchandra's (1991) CYCLOPS system finds innovative solutions, enlarging the search space by relaxing constraints and by retrieving and adapting previously stored designs.

1.2 VISUAL REFERENCES

Architectural designers use the term 'references' to mean objects from the natural and artificial world that inform their designs, for example rocks, waterfalls, trees, musical instruments, and buildings. Especially for architects, references are inherently *visual*. That is, what is important about these objects is their shape or physical form, and this information (rather than functional or behavioral models, for example) is what designers extract and employ in their designs. Visual references are often drawn from drawings, photographs, diagrams, sketches, and paintings. References seem to play a key role in supporting the three activities of creative design mentioned above : combining ideas, using metaphor and expanding the search space.

Many books about design discuss the use of visual references to explore architectural concepts. For example, *Design in Architecture* (Broadbent 1973), *Visual Notes* (Crowe and Laseau 1984), and *Poetics of Architecture* (Antoniades 1990) all mention architects' use of visual references as analogies and metaphors. Design folios and architect's personal statements often describe using visual analogies and metaphors during design development. Examples can be found in discussions of the works of famous architects, such as Le Corbusier, Frank Lloyd Wright, and Alvar Aalto (Corbusier 1958; Sekler and Curtis 1978; Wright 1943; Storrer 1993; Schildt 1989).

1.2.1 Visual reference and drawing

Architectural designers often browse collections of images as they design, and the designing frequently involves drawing, copying, tracing, transforming and incorporating reference forms. Architects and design instructors encourage students to use visual references in developing design form. For example, Le Corbusier urges architects to develop their imagination by studying and drawing natural organisms such as a class of shells (Guiton 1987). Antoniades in *Poetics of Architecture* (1990) summarizes architects' uses of natural forms and encourages designers to draw from nature to enhance creativity. Cappleman and Jordan's

Foundations in Architecture (1993) describes student designs that use insects and plant patterns. Clark and Pause's influential *Precedents in Architecture* (1985) uses different diagram types (e.g., circulation, geometry) to depict sixteen famous buildings.

Designers sketch and diagram to explore possible design solutions. Edwards suggests in *Drawing on the Right Side of the Brain* that drawing makes creative solutions to problems more accessible (Edwards 1979). An architectural sketch can explore general concepts such as projected spatial 'feel' of a design, or specific functional issues such as circulation, structure, and construction method. Sketches and diagrams are also drawn to record and analyze design concepts. In Ching's *Architecture: form space and order* (1979), diagrams illustrate how architectural elements define built space. Lockard (1977) suggests that architects draw primarily for testing, discarding, and refining "conceptual ideas," and presenting ideas to others is only a secondary concern. Aalto encouraged his students to sketch freely, to "think and draw" so those "beautiful lines" will cultivate the "eye" for form (Schildt 1989). Graves (1977) describes the roles of architectural drawing in conceptualization; he identifies the 'referential sketch' as a record of discovery and "a metaphorical base for use or transformation in later compositions".

1.2.2 Finding visual references - by visual and conceptual similarity.

If images of natural and artificial forms are so useful, how do architects find them? At least two 'reminding' (Kolodner 1993; McLaughlin 1993) mechanisms appear to be at work. Reminding may be visual, as when a shape sketched in the emerging design (perhaps doodled on a cocktail napkin) recalls a reference form, or it may be linked through a concept about the design. Examples of visual (shape) reminding include a horseshoe crab shell for the roof of Le Corbusier's Ronchamp Chapel (Corbusier 1958), palm trees for the columns of Santiago Calatrava's BCE Place Gallery (Blaser 1989), a head in agony with an open mouth for Michelangelo's Porta Pia (Chimacoff 1982) and yachts in Sydney Harbor for the shell shapes in Utzon's Opera House (Arup 1967) (Figure 1).

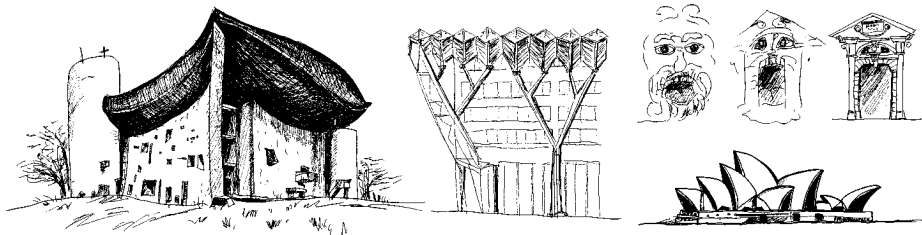


Figure 1. Examples of shape reminding: [a] Le Corbusier claimed the roof for the chapel at Ronchamp was inspired by a horseshoe crab shell; [b] Columns at Calatrava's BCE Place Gallery recall palm trees; [c] Michelangelo's Porta Pia may have derived

from a drawing of a head in agony; [d] Ove Arup, Utzon's construction partner, claimed the Sydney Opera House's shell forms reflected the image of yachts in Sydney Harbor.

On the other hand, visual references can be conceptual, that is, linked in the first place to a concept about the building, rather than to a physical form. Examples of conceptual reminding include Wright's Unitarian meeting house at Madison, Wisconsin (Figure 2a),: a church is a place for prayer, recalling the form of folded hands (Wright 1943) and the section of Kahn's Fort Wayne Performing Art Theatre, (Figure 2b) shaped like a violin in its case (Brownlee and Long 1991).



Figure 2. Examples of conceptual reminding [a] Wright claimed the form for his Unitarian meeting house derived from the image of hands clasped in prayer; [b] Many of Kahn's performing arts centers borrowed the form of a violin in its case.

1.3 Activities, media, and information in design

Architectural designing involves different activities. For example, observing designers in action, Rowe (1987) described a design process as composed of sketching concepts and images, exploring design ideas, investigating design themes, site constraints, resolution of projects, and development of concepts. Do (1993) interviewed designers about their design process and classified their activities in three broad categories: organization, conception, and fabrication. Each activity employs different kinds of external information, is carried out using different drawing techniques, and therefore calls for different computational support.

At certain times the designer's main concern is function, particularly the manipulation of objects within the context of the architectural program. Do calls these activities 'organization'. Organization drawings, typically crude sketches and diagrams, depict spatial layouts, circulation, cost and construction concerns, and zoning studies (Figure 3).

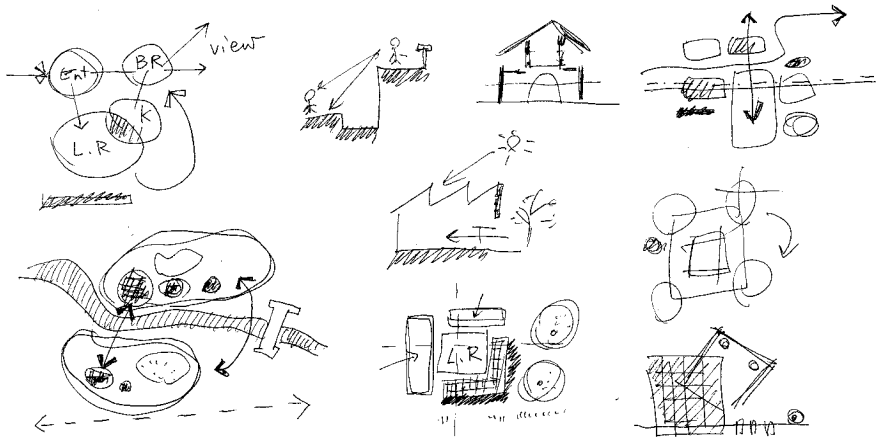


Figure 3. Organization drawings explore programmatic concerns.

More or less interposed with organization activities that aim to resolve functional concerns, the designer also searches for creative images and solutions. Do calls this activity 'conception', in which the designer sketches to explore ideas, employs metaphor and images, and often makes 'gestural' drawings of artifacts (Figure 4). Geometry is explored and principles of form such as rhythm and harmony are considered.

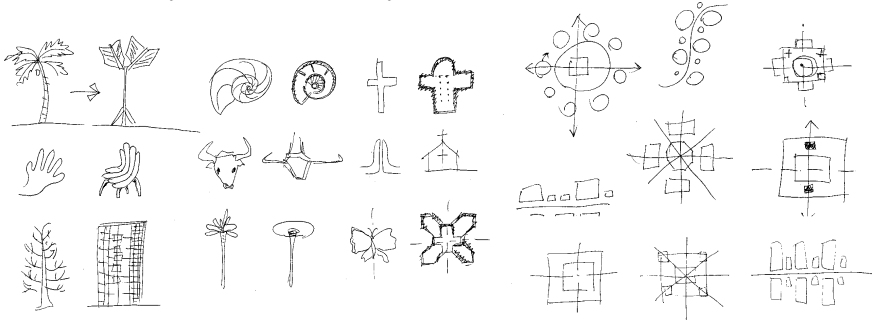


Figure 4. Conception drawings explore shapes and forms.

Usually during a later phase the design is developed further and made ready for realization, at which point construction drawings are needed. Do calls these activities 'fabrication'. During fabrication, detailing and grids are usually more definite and precise than in the first two phases, and fabrication drawings (figure 5) lead directly to the production of hard-line, mechanical, drawings.

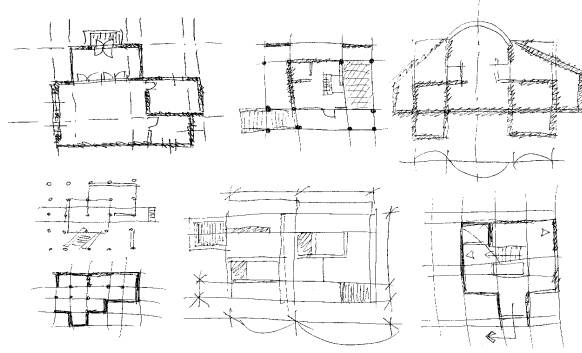


Figure 5. Fabrication drawings tend to be more definite and precise.

Computational tools have been built to support these activities, especially organization and fabrication tasks. Functional analysis, spatial layout, and knowledge based evaluation programs support organizational decision making. Commercial CAD drafting programs provide libraries of geometric elements and building components; they support fabrication. However, few systems have been built to support *conception* activities.

Databases of design information can aid designers in each of these design activities, but the nature of the needed information as well as an appropriate means of finding it may vary widely. For example, for organizational decision making, catalogs of similar designs, precedents, and post-occupancy evaluation stories (as in the case based design aid Archie (Zimring, Do et al. 1994)) may be most useful, whereas for fabrication decisions, a catalog of building components and details may be more appropriate. For conception activities, the designer may employ more widely diverse visual references.

Embedding information tools into drawing environments is a key strategy for getting designers to use them. If information is not ready to hand, a designer will be reluctant to stop drawing to look it up. For example, a usability study of Archie's key word lookup scheme (which is not embedded in a drawing or design generation environment) revealed that architects found the key word (features) search mechanism cumbersome and that using it while designing interfered with their design 'flow' (Do, Or et al. 1994). McCall and Fischer (1990, 1994) in their Phidias and Janus systems follow the strategy of embedding knowledge based critics and access to design argumentation in structured CAD environments. However, as we argue in the following section, structured CAD environments are problematic for creative designing. Especially for conception activities, but also for organization, freehand sketching is the traditional, and arguably the most natural medium; hence it should be possible to access needed information by making sketches and diagrams.

2. Sketch based query and retrieval

2.1 WHY SUPPORT SKETCHING AND DIAGRAMMING WITH COMPUTERS?

In architecture (and in other design domains as well) much conceptual and creative work is currently done using traditional media—pencil and paper, marker, and tracing sheets. Compared with conventional CAD software, pencil and paper is simply more flexible and fluid, allowing the designer to explore more freely and quickly. Therefore, computer aided design systems that support creative work could enable the designer to work in an unstructured way with a pen or pencil and emulate the advantages of paper based media. For example, with paper the designer can mark directly on the design drawing, indicating shape, line weight, color, and position simply by drawing, without interposing command sequences or menu choices. The designer can also easily copy, trace, move, and rotate drawings and drawing fragments.

However, what is the advantage to using computers to support sketching if the machine merely emulates paper media, as some sophisticated drawing and painting programs do? To support creative designing, the sketching environment might offer additional capabilities. An obvious extension is more powerful editing, allowing the user to reshape lines, to delete, group, and duplicate figures as with conventional (structured, menu-based) CAD drawing tools. But beyond tools for making and editing sketches, computing environments for creative design could begin to recognize sketch and diagram elements, providing simulation, critiquing, constraint maintenance, and knowledge based editing. These more sophisticated enhancements endow the computer based sketching environment with advantages over ‘dumb’ paper. They give reasons to begin using the machine sooner: access to relevant information during the earlier stages of design when changes in strategy are not prohibitively costly. In short, pen based computational drawing environments offer designers a way to move smoothly and incrementally from conceptual sketches to more schematic design drawings to hard line mechanical drawings, supporting the process throughout with design information.

2.2 A SKETCHING ENVIRONMENT

Recent advances in digitizer and pen computing technologies have spurred research in computational environments for hand drawn sketching, for example, Saund and Moran’s (1994) PerSketch program and Kramer’s (1994) use of translucent patches . Here we briefly describe our own computer based design environment to support hand drawn sketching and

diagramming: The Electronic Cocktail Napkin emphasizes recognition and interpretation of drawing, management of trace layers, and graphical constraint maintenance.

The Electronic Cocktail Napkin (Gross 1995) is an environment for making hand drawn sketches and diagrams. The program reads stroke information from a digitizing tablet, a mouse, or a hand held 'personal digital assistant' and attempts to recognize the hand drawn figures. It retains inking information (raw xy points and pressures) for display and for possible later re-interpretation. In addition to trainable recognition of hand drawn glyphs and parsing more complex figures built from combinations of glyphs in certain spatial relations, the Cocktail Napkin supports storage and retrieval of sketches and diagrams, simulated tracing paper, multiple users, as well as standard CAD-like editing of raw stroke data.

The user can make a set of diagrams and assemble them into a collection we call a 'sketchbook', browse previously made sketchbooks, or use the Cocktail Napkin to search for combinations of elements in certain spatial relations. To enable flexible matching, spatial relations and element types may be identified more or less specifically in the search pattern. For example, a search for the pattern [A concentric B] finds only pairs of figures A and B that have roughly the same center point (actually, whose bounding boxes are centered); [A contains B] is a more general version of the relation that simply requires B to be inside the bounds of A; and [A overlaps B] requires only that the two figures share some spatial extent (figure 6). Similarly, elements A and B can be identified in the search pattern more or less specifically, for example: 'a small circle,' 'a circle,' or 'any shape.'

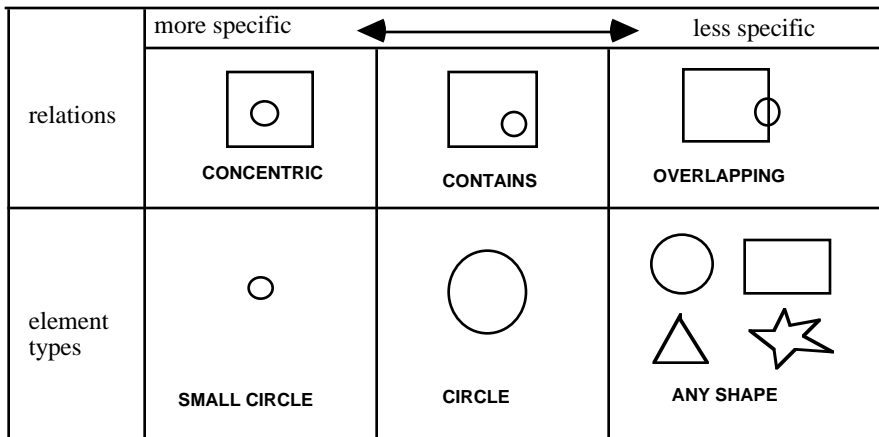


Figure 6. A hierarchy of spatial relations and element types enables flexible matching.

The user can program the Cocktail Napkin's graphical search routines to recognize new, higher-level configurations by defining replacement rules

that identify certain combinations of element types and spatial relations. For example, a user can define a collection of circles and connecting line segments as a ‘graph,’ which can in turn become a part in larger figures. A ‘row of columns’ made up of individual small boxes arranged along a horizontal line can be recognized as an element in an elevation drawing. In effect, the collection of recognition rules for configurations makes up a set of graphical rewrite rules for parsing a visual language of diagrams.

2.3 DIAGRAMMATIC SIMILARITY: SHAPE AND SPATIAL RELATIONS

We have explored several ways to compare diagrams and sketches, using element type and spatial relation information for scoring diagram similarity. The user can train new element types on the fly; the Cocktail Napkin’s initial training set includes common geometric shapes such as box, circle, line, triangle and blob. Recognition of simple multistroke elements is based on pen path, stroke and corner count, aspect ratio, and size. The program also maintains multiple readings for ambiguous elements until they can be resolved later from context; thus a closed roughly rectangular figure may be identified as ‘either a circle or a box,’ but the determination is delayed until other parts of the figure allow a higher level recognizer to resolve the ambiguity.

Spatial relations include adjacencies (such as right of, left of, directly above, below), containments (such as concentric, overlap and contains), and relations among lines (connect, intersect, tee). The similarity comparison is made among symbolic descriptions of figures, such as in figure 7.

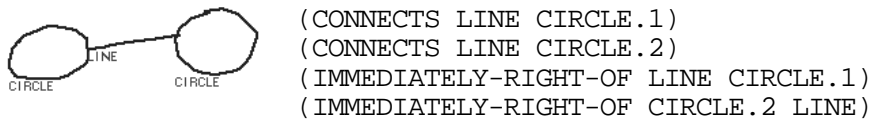


Figure 7. A simple diagram and its symbolic description.

We compute a weighted aggregate measure S , which combines several dimensions of diagram similarity:

$S = (\sum k_i s_i) / \sum k_i$, where the s_i are dimensions of similarity whose values range from 0.0 to 1.0 and the k_i are weights. Thus, the formula computes an overall similarity measure from 0.0 to 1.0. The component dimensions are as follows.

- The element-type similarity measure (s_t) compares the types of elements in each diagram. If the two diagrams have entirely different element types, the element-type similarity measure is zero; if they have exactly the same element types, that measure is 1.0.
- The element-count similarity measure (s_c) compares the number of elements in each diagram, computing the ratio of the smaller number to the

larger. Thus the measure is 1.0 if the two diagrams have exactly the same number of elements, and tends to zero as the numbers of elements in the diagrams diverge.

- The element-type-count similarity measure (s_{tc}) compares the number of elements of each type. It is 1.0 if the two diagrams have exactly the same numbers of elements of each type. It is zero if the two diagrams have no matching element types.
- The spatial relations similarity measure (s_r) compares the spatial relations in the two diagrams. The measure is 1.0 if the spatial relations (but not necessarily their element arguments) match.
- The relations&type similarity measure (s_{rt}) compares the spatial relations and the types of their element arguments. It is 1.0 if the element types and their relations match exactly.

Figure 8 shows the similarity scores of the pattern in figure 8a with several others (figures 8b, c, d, e) according to the element type, relations, and relations&type measures. For example, comparing figure 8b with 8a: 2/3 of the element types match (both have a triangle and a horizontal line), the relations match entirely (both have two 'directly-above' relations) but the combined relations&type measures scores 0, because none of the matching relations have the same element types. On the other hand, comparing figure 8c with 8a, the combined relations&type measure scores 1/2, because one of the two relations in figure 8b matches exactly with that of 8a (box directly-above line).




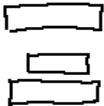

| FIGURE |  |  |  |  |  |
|------------|---|---|---|---|---|
| ELEMENTS | triangle box horizontal line | triangle row of columns horizontal line | half circle box horizontal line | box box box | line triangle box |
| RELATIONS | tri d-above box box d-above line | tri d-above row-col row-col d-above line | half above box box d-above line | box above box box d-above box | line d-above tri tri d-above box |
| SIMILARITY | element type relations relations&type | 2/3 1 0 | 2/3 1/2 1/2 | 1/3 1/2 0 | 1/3 1 0 |
| | [a] | [b] | [c] | [d] | [e] |

Figure 8. Some dimensions of diagram similarity: [a] compared with [b, c, d, e].

2.4 DIAGRAM QUERY AND RETRIEVAL OF VISUAL REFERENCES

In previous work, we built two prototype systems for querying databases of design information using hand drawn diagrams, employing the Cocktail Napkin's graphical search routines. The database for the first system was the case based design aid, Archie, which contains a collection of post-

occupancy evaluation (POE) data of courthouses and libraries. Archie was designed to remind architects in early stages of design of problems and opportunities in similar building projects. The POE information is largely text with photographs and scanned-in drawings; in Archie it is indexed by a set of key words that describe design features. We augmented Archie's feature based index with a visual bookmarking scheme that enables users to associate diagrams with stories, problems, and responses in the case base and recall those items by drawing similar diagrams on a query sketchpad (figure 9) (Gross, Zimring, and Do 1994).

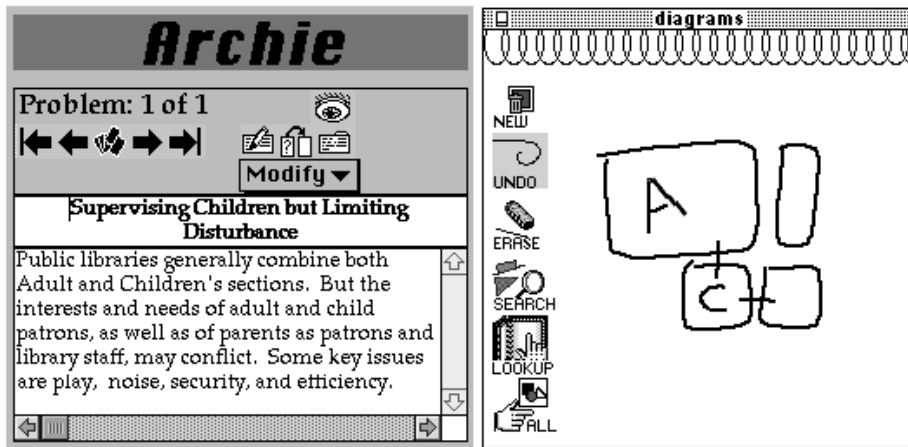


Figure 9. Querying the Archie case base. A case about the arrangement of adult (A) and children's (C) sections in library is indexed and retrieved by a diagram.

The database for the second prototype we built was the Great Buildings Collection CD ROM, which contains over 700 drawings, photos, and video clips of world famous architecture (Matthews 1994). We found in an informal experiment that designers often remember buildings by their floorplan or elevation rather than by architect or building name, and that we can rely on architects to make similar diagrams. Hence we believe a diagram query scheme will be useful for finding items in a visual collection of architecture. As with Archie's visual bookmarking scheme, a user browsing the Great Buildings Collection can draw diagrams in a sketchbook and associate these with items in the visual database, then later retrieve items by drawing a similar sketch (figure 10).

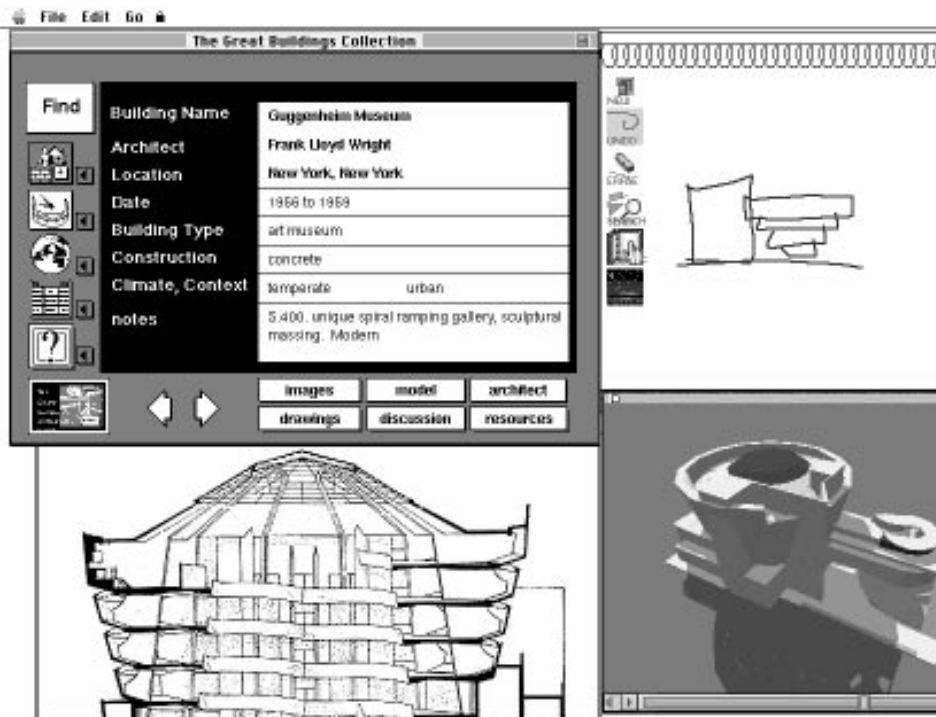


Figure 10. Querying the Great Buildings Collection CD-ROM

In both prototypes we developed the bookmarking scheme to provide designers with the option of a personalized sketchbook and to reduce the load of the system developer in creating the indices. Gradually, as designers sketch with the database they construct a diagrammatic index. We have conducted several pilot experiments (e.g. (Do 1995)) that suggest designers make similar diagrams to illustrate design concepts and buildings, and even when the diagrams have varied styles, designers can understand each other's drawings. Therefore they are likely to be able to share and exchange sketchbooks and bookmarks.

2.5 LINKING VISUAL DATABASES WITH A DIAGRAMMATIC INDEX

The two diagram index schemes described above linked specific diagrams with visual database items in a one-to-one mapping. Our more general present scheme enables several databases to be used simultaneously by virtue of a sketchbook that links diagrams with database items. The designer can paste drawings into a sketchbook, leaf through its pages, copy and modify drawings, and use the sketchbook to query the image collections. Figure 11 shows a schematic diagram of the query process.

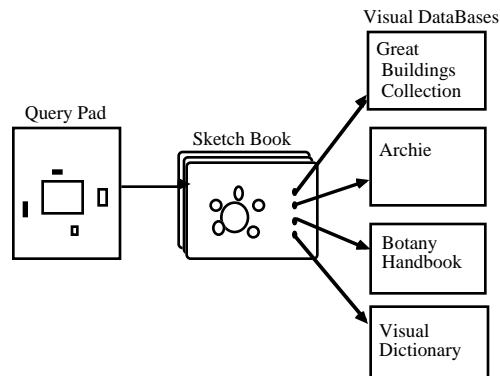


Figure 11. The sketchbook indexes several databases simultaneously, relating items in different databases that have similar shapes.

When the designer draws a diagram or a sketch on the query pad, the program's graphical similarity routines first identify the most similar sketch(es) in the sketchbook. Then the program sends database lookup calls to each of the various databases identified on that page to display their items. Figure 12 shows an example working screen, with several active databases.

For example, when designer makes a diagram of a box surrounded by four boxes in the query pad (bottom right), the Drawing Analogies program turns the sketchbook (bottom center) to the page most similar to the query, in this case, a circle surrounded by other circles. This sketchbook page link items in several databases with similar concepts of surrounding, in this case, 1) a Palladian Villa from the Great Buildings Collection (top left); 2) an Archie story concerning spatial arrangement of desks and chairs, (bottom left); and 3) a page from a collection of flower drawings (top right).

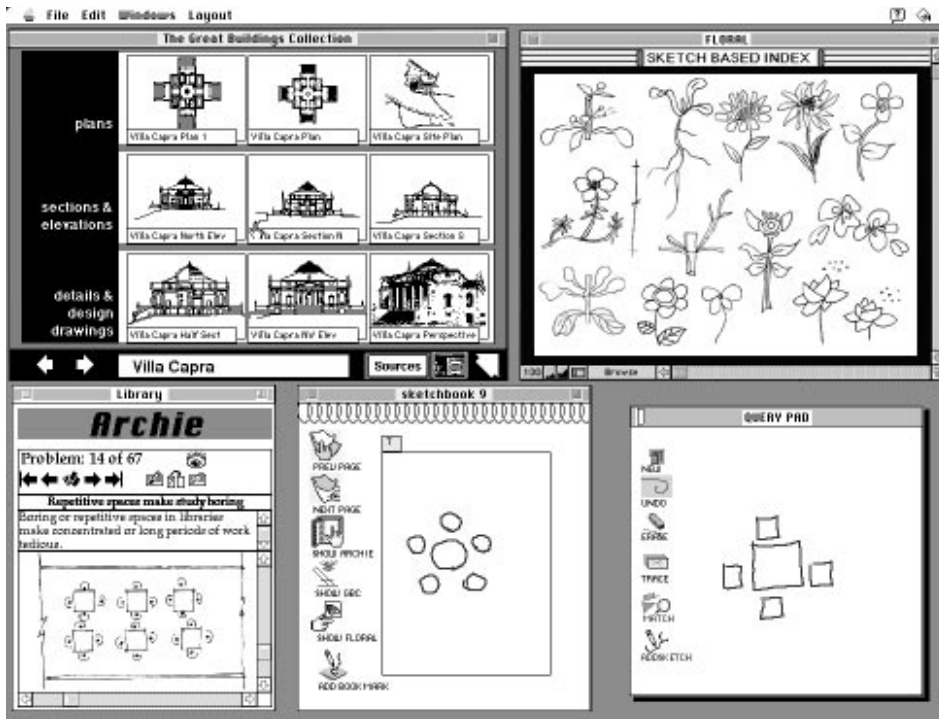


Figure 12. The Drawing Analogies working screen.

Different database items indexed and retrieved by a similar diagrams could help designers access more visual references for creative designing. For example, she might browse through magazine pictures to find ideas and make sketches to record them. A designer working with a circular floor plan might collect pictures of round shaped buildings, textiles with circle patterns, and even still life paintings of fruit baskets with apples, oranges and watermelon. Later she might incorporate those pictures into her design to produce a lobby hall with a dome roof, patterns for a rose window and floor tiles, and even perhaps a color scheme inspired by fruit.

2.6 IMPLEMENTATION

The implementation of the Cocktail Napkin drawing environment (Gross 1994a), and the basic diagrammatic indexing scheme are discussed elsewhere (Gross, Zimring et al. 1994). We summarize here the additions to handle multiple reference libraries.

A sketchbook is an ordered collection of pages, each containing a sketch or diagram as well as links to items in one or more databases. That is, each sketchbook page can be described as a tuple:

page ($S (D_1, \dots, D_k)$)

where each D_i is a set of pointers ($d_{i,1}, d_{i,2}, d_{i,j}$) to the i th database and S is a sketch containing identifiable elements ($E_{1\dots p}$) among which binary spatial relations ($R_{1,2}$ through $R_{p-1,p}$) can be found. Thus the pages of the sketchbook establish relations among the various database items in each D_i , cross-referencing items with similar diagrams.

In our Lisp implementation of the Drawing Analogies program, different software packages may be used for the various databases. For example, the Great Buildings Collection is a HyperCard application and Archie is built in Common Lisp. We use AppleEvents (a standard Macintosh interprocess communication protocol) to communicate with applications built outside Lisp. A uniform protocol between our program and the database makes it straightforward to add new databases. For each new database a Lisp function (**current-item database**) must be provided to determine the database's currently displayed item; another function (**display-related-items database page**) commands the database to look up and display items indexed on a given sketchbook page. More than one item may be indexed on a particular diagram; two standard ways are provided for the database's **display-related-items** function to handle this. The lookup function can produce a dialog asking the user to choose from retrieved references (by name or thumbnail image), or an intermediate retrieval set can be returned and the user can browse this set or further refine the query.

3. Discussion

3.1 THE PROBLEM OF MAKING DIAGRAMS FROM IMAGES

Our scheme for finding references in on-line visual databases hinges on one important condition: the images must be indexed. This can be done in two ways: by people or by machines. It may well be that using people to index visual databases by hand (drawing simple diagrams and sketches) is a more logical, trustworthy, and straightforward approach. However, lacking a sufficiently large cadre of motivated undergraduate students and the money to pay them, we have begun looking into using image processing techniques on the source images to index visual databases by machine. We have carried out preliminary explorations to determine the feasibility of automatically indexing visual databases.

We have stroke, pressure, and sequence information for the designer's hand drawn drawings; whereas for scanned reference images we have not. To some extent image processing techniques can reconstruct from a scanned image the strokes used to make a hand drawn diagram. At least in some

cases, the image can just be reduced to its most prominent forms. For example, we can decrease the resolution, increase contrast, etc., to reveal the basic lines of the image, just as one can often better apprehend the composition of a painting by squinting at it.

We used a commercially available tool suite—Adobe Inc’s PhotoShop, and Streamline—to make diagrams from digital images. PhotoShop is a ‘digital darkroom’ program that operates on raster images. Streamline is an edge detection and vectorization program used for converting scanned art (such as blueprints) into CAD drawings. Both programs provide numerous options and parameters and require considerable skill to operate effectively. We used these tools to process several different types of source materials: hand drawn diagrams, hard line diagrams, hand drawn sketches, hard line plan and elevation drawings, and color photographs. Our goal was to identify a set of operations and parameter settings for processing various types of images into a small set of poly-lines that can be read into our sketching environment using a standard graphic data exchange format. Unsurprisingly, the easiest images to convert were hand drawn and hard-line diagrams and the most difficult were photographs.

But to construct ‘appropriate’ diagrams from a hard line drawing or a photograph poses a more serious conceptual difficulty. In making diagrams from drawings and photographs, designers employ significant interpretation skills. Although we can use image processing to convert some visual images to a plausible diagrammatic form, we will be unable to cover the range of diagrams designers will make. For example, we asked architects to make diagrams of Frank Lloyd Wright’s Guggenheim Museum. Designers drew either a flat spiral or a curlicue (corkscrew) figure reflecting the three-dimensional circulation path, or they drew a set of stacked boxes reflecting the front elevation of the building, which is commonly pictured in photographs. The internal circulation path of the building has the form of a helix, but this cannot be understood easily from plan, section, or elevation drawings or from photographs. In other words, the helix and spiral diagrams rely on knowledge about the building that is not easily obtained from visual reference material.

3.2 USING REFERENCES ONCE WE FIND THEM

If finding visual references is valuable for creative design, still a sketch based reminding program must be used together with other supporting tools. Following McLaughlin’s (1993) 4-part framework, computational tools for creative design must support generating, selecting, reminding and merging of designs. The first two activities, generating and selecting design alternatives, we leave to the human designer. Although the Cocktail Napkin program supports the act of drawing, we do not try to automate the

generation of designs. Likewise, the designer can impose constraints on the drawings to limit the search space; again we leave the choice of constraints to the designer. This leaves the latter two activities, reminding and merging. We have discussed our approach to reminding; next we outline how the designer can merge and adapt retrieved reference information into a design.

After finding an appropriate or interesting visual reference, the designer incorporates it into the design in whole or in part, more or less directly. With traditional design media, this typically involves tracing or copying the image onto the design drawing at various scales. While tracing and copying, the designer may alter the reference, fragmenting, stretching, and squeezing it in response to the needs of the design. The designer may also make functional assignments from the design program to parts of the visual reference, for example matching spaces and circulation paths with features of the drawing.

To support merging and adapting, Drawing Analogies provides the designer with two ways to import a visual reference into a sketch. First, the reference material may be imported as an underlay to the Cocktail Napkin drawing, where it may be moved, stretched, rotated, and manipulated with simple paint tools. The designer can then draw on top of the reference image, using the Cocktail Napkin's semi-opaque trace layers. Second, the sketchbook diagram that indexed the reference item can be imported into the drawing environment, where it takes on the normal status of a drawing that the designer had penned herself. The designer can edit the diagram, again stretching, moving, rotating, and erasing parts, or combine the imported reference diagram with the designer's original sketch.

4. Conclusion and future work

4.1 SUMMARY: USING VISUAL REFERENCES IN CREATIVE DESIGNING

We have argued that computational support for creative design in architecture should include retrieval of visual references with similar forms. Our approach aims to support three aspects of creativity: 1) combining ideas from different sources, 2) using visual metaphor and analogy, and 3) expanding the search space to include innovative solutions. We began with a discussion of how reference images are used in creative design and the role of sketching, with examples from design research, pedagogy, and the work of famous designers. We are, of course, aware of a tendency among architects to engage in creative story-telling, inventing myths about their own design process. Utzon's sails or Le Corbusier's crab shell may simply be ex post facto rationalizations. Nevertheless, there is considerable evidence for

the use of visual reminding, metaphor, and analogy in design. We described our visual reference program, Drawing Analogies, which retrieves images with similar shapes and spatial relations. Embedded in a computational sketching environment, Drawing Analogies links a variety of databases through a sketchbook of hand drawn diagrams.

We don't mean to imply that visual reminding is everything in creative designing. For example, Heath (1984) says "Analogy is the most important, but not the only, source of form" (p 160). Borrowing and adapting shapes, visual metaphor and analogy are part, but only part, of the story. We have presented only a system for retrieval; it does not account for how references might be employed and adapted in a process of analogical reasoning. This is one topic for future work. We are also aware that design, even creative design, comprises many other important activities. For example, it also involves reasoning about function and structure: making, finding, and adapting structures to address functional requirements.

4.2. CONNECTIONS TO CONSTRAINTS AND CASE BASED REASONING

The Drawing Analogies program is integrally connected with other work we are doing in constraint based and case based support for design. For example, Stretch-A-Sketch, an extension of the Cocktail Napkin program, employs a constraint management scheme to maintain spatial relations in a diagram (Gross 1994b). The Napkin's recognizers identify key relations in the diagram (adjacencies, containments, relative sizes); these are asserted as constraints; and the constraint manager maintains them as the designer edits the drawing. A second topic for future work is to combine Stretch-A-Sketch with the retrieval of visual references. As the designer adopts a visual reference diagram she can impose the diagram's spatial relations as constraints on her design.

Providing visual references for creative design serves a larger goal: to provide information in the design environment through case libraries, appropriate and relevant to the design task at hand. We envision a suite of various information providing tools, available at different stages of design, accessed in different ways—via structured drawing, via sketching, and via key words. A third topic for future work is to try to assess the designer's intention, or focus; then to use this assessment to filter the information the system provides.

Finding diagrams with similar form can also be seen as retrieving cases that satisfy similar constraints. In Drawing Analogies, the constraints are the spatial relations and element types in the query. To be sure, spatial form is only one descriptor of designs. However, at least for the form-making, conception activities of design it is an important descriptor. Although case

libraries of designs may also be indexed by other features (e.g. function), constraints on spatial form will surely be useful for retrieving visual images.

Acknowledgments

We are grateful to the Archie Group at Georgia Tech (Craig Zimring, Janet Kolodner, and Eric Domeshek) for the use of the Archie II program. At Boulder, Ray McCall's comments on an earlier draft were illuminating. We thank Kevin Matthews and the Design Integration Laboratory at the University of Oregon for permission to reprint the screen snapshot of The Great Buildings Collection. Funding from National Science Foundation grant DMII-9313186 was instrumental in building software infrastructure for this work.

References

- Antoniades, A. C.: 1990, *Poetics of architecture: theory of design*, New York, Van Nostrand Reinhold.
- Arup, O.: 1967, Address to the Prestressed Concrete Development Group, London, on 14 January 1965. in M. Baume (ed), *The Sydney Opera House Affair*. Sydney, Thomas Nelson.
- Blaser, W.: 1989, *Santiago Calatrava, Engineering Architecture*, Basel, Birkhauser Verlag.
- Boden, M. A.: 1990, *The Creative Mind, myths and mechanisms*, London, Georgia Weidenfeld and Nicolson.
- Broadbent, G.: 1973, *Design in architecture: architecture and the human sciences*. London, New York, Wiley.
- Brownlee, D. B. and Long, D. G. D.: 1991, *Louis I. Kahn: In the Realm of Architecture*, New York, Rizzoli International Publications, Inc.
- Cappleman, O. and Jordan M. J.: 1993, *Foundations in Architecture: an annotated anthology of beginning design projects*, New York, Van Nostrand Reinhold.
- Chimacoff, A.: 1982, Figure, System and Memory: the process of design, in O. Akin and E. F. Weinel, (eds), *Representation and Architecture*, Silver Spring, MD, Information Dynamics, pp 139-157.
- Ching, F.: 1979, *Architecture: form, space & order*, New York, Van Nostrand Reinhold.
- Clark, R. H. and Pause, M.: 1985, *Precedents in architecture*, New York, Van Nostrand Reinhold.

- Corbusier, L.: 1958, *The Chapel at Ronchamp*, New York, Frederick A. Praeger.
- Crowe, N. and Laseau, P.: 1984, *Visual Notes for Architects and Designers*, New York, Van Nostrand Reinhold.
- Do, E. Y.-L.: 1993, *Stages of Conceptual Diagramming in the Design Process*. Working Paper, Georgia Institute of Technology, College of Architecture.
- Do, E. Y.-L.: 1995, What's in a Diagram (That a Computer Should Understand), in M. Tan (ed), *Proceedings of CAAD Futures '95*, Singapore.
- Do, E. Y.-L., Or, S.-W. D. et al.: 1994, *Usability Study of A Case-based Design Aid Archie*, Working Paper GIT-CS-6752-94, Georgia Institute of Technology.
- Edwards, B.: 1979, *Drawing on the Right Side of the Brain*, Los Angeles, J. P. Tarcher.
- Ferguson, E.: 1992, *Engineering and the Mind's Eye*, Cambridge, MA, MIT Press.
- Fish, J. and Scrivener, S.: 1990, Amplifying the Mind's Eye: Sketching and Visual Cognition, *Leonardo* **23**(1): 117-126.
- Gero, J. S.: 1994, Creativity and Design, in T. Dartnall (ed), *Artificial Intelligence and Creativity*, Dordrecht, Kluwer, pp 259-267.
- Gero, J. S. and Maher, M. L.: 1992, Mutation and analogy to support creativity in computer aided design, in G. Schmitt (ed), *CAAD Futures '91*, Wiesbaden, Wieweg, pp 261-270.
- Goldschmidt, G.: 1992, Serial Sketching: Visual Problem Solving in Designing, *Cybernetics and Systems* **23**: 191-219.
- Goldschmidt, G.: 1994, Visual Analogy in Design, in R. Trappl (ed), *Cybernetics and Systems '94*, Singapore, World Scientific, pp 507-514.
- Graves, M.: 1977, The necessity for drawing: tangible speculation, *Architectural Design* **6**(77): 384-394.
- Gross, M.D., Anderson, J. , et al.: 1987, Designing With Constraints, in Y. Kalay (ed), *The Computability of Design*, New York, Wiley and Sons, pp 53-83.
- Gross, M. D.: 1994a, Recognizing and Interpreting Diagrams in Design, in T. Catarci, M. F. Costabile, S. Levialdi and G. Santucci (eds), *Advanced Visual Interfaces '94*. ACM Press, pp 89-94.
- Gross, M.D.: 1994b, Stretch-A-Sketch - a dynamic diagrammer, in A. Ambler (ed), *Proc. IEEE Symposium on Visual Languages*, St Louis, MO, pp 232-238.
- Gross, M. D.: 1995, The Electronic Cocktail Napkin - working with diagrams, *Design Studies*, to appear.

- Gross, M. D., Zimring, C, and Do, E.: 1994, Using Diagrams to Access a Case Base of Architectural Designs, in J. Gero (ed), *Artificial Intelligence in Design '94*, Lausanne, Kluwer Academic. pp 129-144.
- Guiton, J.: 1987, *The Ideas of Le Corbusier: On Architecture and Urban Planning*, Paris, New York, George Braziller.
- Heath, T.: 1984, *Method in Architecture*, Chichester, John Wiley & Sons.
- Koestler, A.: 1964, *The Act of Creation*, New York, Macmillan.
- Kolodner, J. L.: 1993, *Case-Based Reasoning*, San Mateo, CA., Morgan Kaufmann.
- Kramer, A.: 1994, Translucent Patches - dissolving windows, ACM Symposium on User Interface Software and Technology (UIST 94), Marina del Rey, CA, ACM Press.
- Laseau, P.: 1980, *Graphic Thinking for Architects and Designers*, New York, Van Nostrand Reinhold.
- Lockard, W. K.: 1977, *Drawing as a Means to Architecture*, Tucson, Pepper Publishing.
- Logan, B. and Smithers, T.: 1993, Creativity and design as exploration, in J. S. Gero and M. L. Maher (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Hillsdale, NJ, Lawrence Erlbaum, pp 139-175.
- Matthews, K.: 1994, *Great Buildings Collection*, New York, Van Nostrand Reinhold.
- McCall, R. J., Bennett, P., et al.: 1994, An Overview of the Phidias II HyperCAD System, in *Proceedings of ACADIA '94 (Association for Computer Aided Design in Architecture)*, St Louis, MO., pp 63-74.
- McCall, R. J., Fischer, G., et al.: 1990, Supporting Reflection-in-Action in the Janus Design Environment, in M. McCullough, W. J. Mitchell and P. Purcell (eds), *The Electronic Design Studio*, Cambridge, MA, MIT Press, pp 247-260.
- McKim, R. H.: 1972, *Experiences in Visual Thinking*, Monterey, Calif., Brooks/Cole.
- McLaughlin, S.: 1993, Emergent value in creative products: some implications for creative processes, in J. S. Gero and M. L. Maher (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Hillsdale, NJ, Lawrence Erlbaum, pp 43-90.
- Navinchandra, D.: 1991, *Exploration and Innovation in Design*, New York, Springer-Verlag.
- Newell, A., Shaw, J. C. and Simon H. A.: 1962, The Process of Creative Thinking, in H. E. Gruber, Terrell, G., and Wertheimer, M. (eds), *Contemporary Approaches to Creative Thinking*, New York, Prentice Hall.

Rowe, P.: 1987, *Design Thinking*, Cambridge, MIT Press.

Saund, E. and Moran, T. P.: 1994, A Perceptually-Supported Sketch Editor, in *ACM Symposium on User Interface Software and Technology (UIST-94)*, Marina del Rey, CA, ACM Press, pp 175-184.

Schildt, G.: 1989, *Alvar Aalto, the mature years*, New York, Rizzoli.

Sekler, E. F. and Curtis, W.: 1978, *Le Corbusier at Work*, Cambridge, Harvard University Press.

Storrer, W. A.: 1993, *The Frank Lloyd Wright Companion*, Chicago, University of Chicago Press.

Ullman, D., Wood, S., et al.: 1990, The Importance of Drawing in the Mechanical Design Process, *Computers and Graphics* **14**(2):263-274

Wright, F. L.: 1943, *An Autobiography*, New York, Duell, Sloan and Pearce.

Zimring, C., Do, E. Y.-L., et al.: 1994, Using post-occupancy evaluation to aid reflection in conceptual design: Creating a case-based design aid for architecture, in *Proceedings of Design Decision Support System II*, Vaals, the Netherlands (in press).