

**Beyond the Low-hanging Fruit:
Information Technology in Architectural
Design, past, present, and future**

Mark D. Gross, Ellen Yi-Luen Do, and Brian R. Johnson

Proc. Association of Collegiate Schools of Architecture
(ACSA) Technology Conference 2000, Cambridge MA.

2000

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

Beyond the low-hanging fruit:
Information Technology in Architectural Design
past, present and future

Mark D. Gross

Ellen Yi-Luen Do

Brian R. Johnson

Department of Architecture

University of Washington

Seattle WA 98195-5720

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

Abstract

Today's commercial CAD software is the product of years of research that began in the 1960's and 1970's. These applications have found widespread use in the architectural marketplace; nevertheless they represent only the first fruits of research in computer aided design. New developments based on research in human-computer interaction (HCI), computer-supported collaborative work (CSCW), and virtual reality (VR) will result in a next generation of tools for architectural design. Although preliminary applications to design have been demonstrated in each of these areas, excellent opportunities remain to exploit new technologies and insights in service of better design software. In this paper we briefly examine each of these areas using examples from our own work to discuss the prospects for future research. We envision that future design technologies will develop from current and traditional conventions of practice combined with forward looking application of emerging technologies.

In HCI, pen based interaction will allow architects to use the pencil again, without sacrificing the added power of computer aided design tools, and speech recognition will begin to play a role in capturing and retrieving design critique and discussion. In CSCW, a new generation of applications will address the needs of designers more closely than current general purpose meeting tools. In VR, applications are possible that use the technology not simply to provide a sense of three-dimensional presence, but that organize design information spatially, integrating it into the representation of artifacts and places.

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

Overview: Where we are today?

The last thirty years have seen remarkable advances in the development and application of information technology in architecture. As a result, design studios in schools and professional firms have adopted computer-based tools in their work practice, using software products such as AutoCAD, 3D Studio, Form•Z, and PhotoShop. A new generation of students is learning to use these tools in both traditional and innovative ways. We have come to expect realistic rendering and walk-through animations in project presentations.

These computer-aided design tools, although a remarkable departure from pencil and tracing paper, basswood and chipboard of the recent past, merely represent the commercialization of the first generation of CAD development. They represent the low-hanging fruit, the most obvious and easy applications of information technology in architectural design. We have hardly exhausted the possibilities of information technology in architectural design. Some of the most effective and exciting developments are yet to come.

These mainstream CAD software products that now are standard tools in the studio and office are the result of a ten to fifteen year research and development cycle. It's important for schools of architecture to understand this cycle in order to anticipate future changes and to recognize where their own technology efforts best fit.

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

In the late 1960s and early 1970s in the era of mainframes and mini-computers only the largest firms such as SOM could afford to invest in computer aided design, for example the development of the AES system for architecture-specific applications. Only a few companies such as ComputerVision were producing CAD software systems. At that time, a handful of academic laboratories were engaged in research in computer aided design, for example the ABACUS lab at the University of Strathclyde, the Architecture Machine Group at MIT, and Eastman's group at Carnegie Mellon. These research efforts began to reach the marketplace during the early 1980s as the first versions of products such as AutoCAD. A similar story can be told for digital photography and raster based image processing (e.g., PhotoShop) and Geographic Information Systems (ArcInfo). In each case exploratory academic research projects predate commercial development by ten to fifteen years.

This may seem obvious to those who have witnessed the development of today's Computer Aided Design tools. However, many students who grew up in the age of computing take these products for granted and assume that CAD tools just come from the software store. They do not realize the ten to fifteen year research and development cycle. The CAD industry will only invest in research and development that pays off in a short time period: at most within 3 years, and perhaps only within the next release of their product. Therefore it is important to invest in the future by continuing longer term research in computer aided design. Schools of architecture, who can afford to take a longer view, have a unique and important role to play in this research.

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

The remainder of this paper focuses on several areas of research in design computing where significant exploratory work has already been done but in which there is the possibility of advances in the coming few years that could profoundly affect the discipline and practice of architectural design. By no means do we mean that these are the only areas of opportunity; rather they reflect the interests of the authors. These areas are: (1) human-computer interaction, (2) collaborative design environments, and (3) the application of virtual reality technology in architecture. We elaborate on each of these areas, using examples from our current research.

Human-Computer Interaction

After more than thirty years of development in computer aided design, it is nothing short of incredible that most architects are using a mouse and menu interface to construct and edit designs. Early design systems (in the 1960's and early '70's) employed light pens or digitizing tablets for drawing input and system commands. With the adoption of the personal computing model developed at Xerox PARC, however, structured menu and mouse (WIMP) interfaces become the norm, combined with the older command line interfaces inherited from mainframe computing systems and embodied in the CPM and DOS operating systems. Despite an active program of research throughout the 1980's and 1990's in human-computer interface (HCI) design, most CAAD software today retains the look and feel of the mid-'70's interfaces developed at Xerox PARC.

Thanks to advances in hardware design (low cost LCD and inexpensive cordless digitizing, for example) and in software recognition technologies the time is now ripe to

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

rethink the human-machine interface in computer aided architecture design. Among the many emerging interaction modalities are pen interaction, speech, gesture, audio, haptics, 3D sensing, and immersive visual environments. We focus our discussion on the first two areas, pen interaction and speech, because these are primary within the tradition of architectural design and therefore may form appropriate 'transition' technologies for next-generation human-computer interaction in computer aided design.

Drawing and pen input

Despite the now wide availability of inexpensive digitizing tablets, freehand input with a stylus—drawing—remains largely unexploited as a means of interacting with computer aided design software. Rather, CAD programs demand structured input and editing using pull-down menus, selection from tool palettes, and entering text into command lines.

There are, of course, advantages to these more structured input modalities; most of the advantages accrue, however, to the software developer rather than to the designer. For example, by requiring the designer to structure input the developer avoids sticky problems of accommodating imprecision, vagueness, and uncertainty.

In a series of prototype projects based on our Electronic Cocktail Napkin system (Gross 1996) we have explored possible roles for freehand diagramming in design as well as technological issues that must be resolved. The basic Electronic Cocktail Napkin platform entails a trainable recognizer for freehand figures and an end-user programmable visual language parser for configurations. It uses an internal representation for uncertainty and ambiguity as well as the means to resolve ambiguity

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

and uncertainty from contextual information (Gross and Do 1996). Demonstration applications have been built for diagrammatic indexing of design case bases and visual collections (Gross 1995), input to simulation programs, and for layout of Web pages, and as a means for creative "shape based" reminding (Do and Gross 1995). The recognition of context offers a way to identify the task that a designer is working on and to offer an appropriate tool (Do 1998). All these prototypes explore drawing as a means to think about designs diagrammatically and conceptually, as opposed to understanding drawing as a means to directly generate the shape or form of designs.

Figure 1. Electronic Cocktail Napkin: Sketch based retrieval from visual databases

In contrast to the diagrammatic and symbolic approach taken in the Electronic Cocktail Napkin projects we built two prototypes to explore the direct generation of three-dimensional form from freehand drawing input. Digital Clay (Schweikardt 1998) explored the application of an old machine vision algorithm (Huffman-Clowes) to derive three-dimensional geometry from a two-dimensional projected sketch. The program walks the lines of the sketch to identify convex and concave edges, then assigns three dimensional coordinates to vertices, and finally generates a three-dimensional model that the designer can view and edit. The program can make assumptions about hidden edges and surfaces, or it can leave that for the designer to specify later. A different approach to this problem is followed in (Tolba, Dorsey et al. 1999); the well known SKETCH System

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

(Zelevnik, Herndon et al. 1996) offers an alternative view based on gesture rather than drawing.

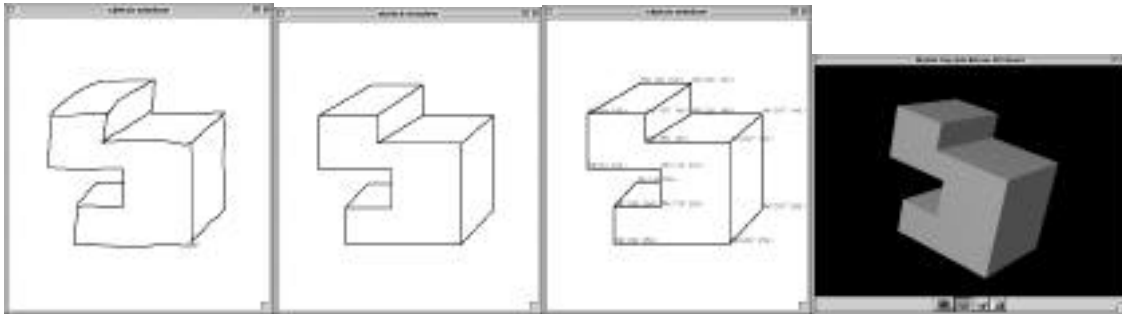
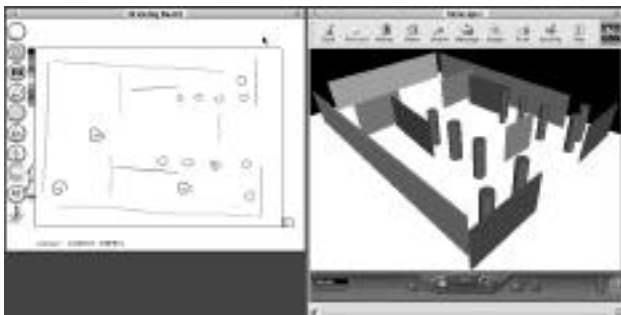


Figure 2. Digital Clay -- Stages of interpreting a sketch to VRML

"Sketch-VRML" allows a designer to quickly generate a virtual world that can be explored using a VRML browser. The designer draws a floor plan that may include lines to indicate walls, circles or small squares for columns, and other symbols for furniture. The Sketch-VRML program generates—by extruding the plan on the fly and inserting design elements from a library— a three dimensional VRML world that reflects the designer's 2D sketch. Interacting with 3D worlds by sketching is an emerging area of research; see for example (Igarashi, Kadobayashi et al. 1998).



**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

Figure 3. Sketch-VRML. Floor plan sketches (left) generate a corresponding VRML 3D scene (right).

Speech recognition

Except for occasional cursing, interaction with CAD programs is a silent enterprise. Although demonstrations of limited vocabulary speech recognition have been made regularly over the past fifteen years it has largely remained a novelty due to high error rates and stringent training requirements. However, recent advances in speech recognition have made it possible to embed this technology in architectural design tools. One could enhance distance collaboration with voice annotation and record and store the audio along with online critiques. In meeting support (whether co-located or geographically distributed), the machine could record the discussion and link the recorded speech with the graphical and video data. Recognition software would transcribe the recorded speech, enabling document reviewers to search the transcript textually and find and play back selected portions of the audio, video, or drawing track of the meeting or critique. Some initial work done in this area (Abowd, Atkeson et al. 1996; Moran, Palen et al. 1997) promises significant rewards for the application in architectural design systems.

Human-Human Interaction: Collaboration and coordination (CSCW)

Following a decade of research in computer supported collaborative work (CSCW) a rapidly growing market in Internet based communication and collaboration tools has emerged. Commercial products run the gamut from simple videoconferencing to Web brokering of construction drawings. Most of today's off-the-shelf collaboration products

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

are not tailored for design but for general-purpose meeting support. They provide whiteboard and videoconferencing tools at such a general and unstructured level that they are of little use in architectural design. At the other end of the spectrum CAD companies are developing Web-centric versions of their personal computer CAD programs that allow geographically distributed users to interact synchronously or asynchronously with a design database. (For example, Bentley Systems' MicroStation/J, AutoDesk's Architectural Desktop). Like conventional (single user) CAD, these programs are most useful at the level of design development and working drawings, less so for supporting conceptual design discussions.

NetDraw - simultaneous collaboration

The NetDraw prototype(Qian and M. D. Gross 1999), built in Java, is a real-time shared drawing program that goes beyond simple pixel based paint-like environments yet stops short of being a fully featured CAD program. A NetDraw server allows multiple users to share a drawing surface and draw and edit a design. The program includes extensible drawing objects whose appearance and behavior can be programmed in Java, the ability to trace over a background image, end-user programmable grids, simple geometric constraints, and ephemeral gestures used to point at and refer to the drawing. A chat window enables textual interaction among drawing participants.



**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

Figure 4. NetDraw

Compadres - group awareness

One of us (Johnson) has participated in several Virtual Design Studio experiments over the past few years (Kolarevic 1998). One of the salient features of a physical design studio that virtual settings lack is this: In the physical studio space one immediately perceives the presence of other designers. One overhears bits of conversations; one keeps an eye on what one's peers are working on. This informal communication forms an important yet often neglected channel in design collaboration.

The Compadres system addresses this need by providing presence awareness in a distributed virtual design studio environment. Compadres appears as a sidebar in the designer's web-browser window. The top part is an automatically refreshed "presence monitor" which lists the members (compadres) of the logged-on user's group (cadre) with color cues to indicate status and links to access member information. When accessed, a compadre's personal information is presented in the lower part of the window. It includes a variety of information: recent usage, email address, phone number, a "door sign" that they can use to post quick notes to the world, a link to "recent work" files, and a one-line message form that the visitor can use to post a (signed) note to the compadre. If desktop webcam software is available, the data page can display live thumbnails of user's physical work area and a screen dump of their desktop. Accessing one's own information gives one the opportunity to edit the "door sign", adjust links, and upload files. Other links in

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

the sidebar interface connect to a shared bulletin board, allowing for conventional VDS exchange and joint review of drawing and model files.

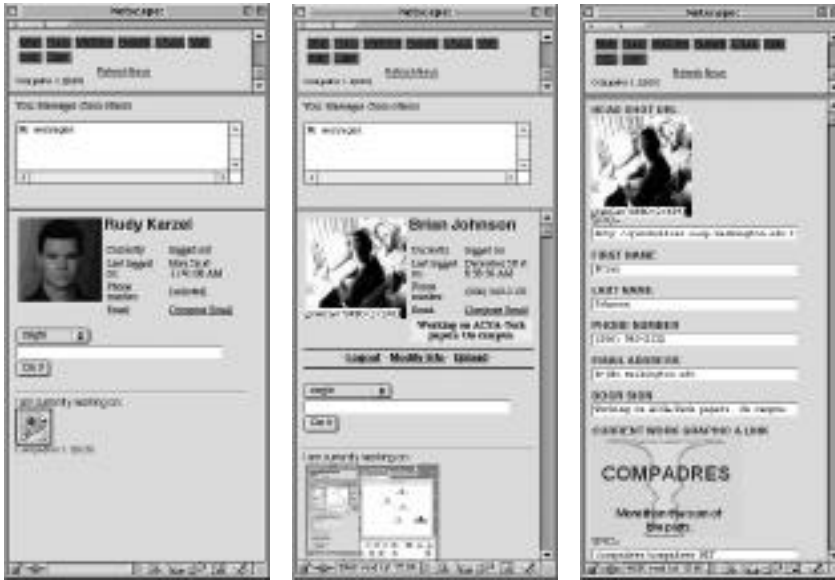


Figure 5. Compadres promotes presence awareness of design team members.

RedLiner - asynchronous collaboration using desktop VR

Whereas NetDraw and Compadres focus on synchronous (same time) collaborative design, the Immersive Redliner project (Jung, Do et al. 1999) explores a dimension of asynchronous collaboration. The system, built as a composite of VRML scripts and Java, enables participants in a design process to view a three dimensional model online and add annotations to the model, much as one might leave post-it notes stuck to the walls in a physical building. The collection of annotations is stored on a central server and can be reviewed by anyone browsing the 3D model. The designer can provide alternative designs for comment as well as enable reviewers to explore some parametric variations.

Beyond the low-hanging fruit: Information Technology in Architectural Design past, present, and future



Figure 6. Immersive Redliner (adding annotations-left, and design alternative-right)

Virtual reality in architecture

Immersive virtual reality hardware such as the head mounted display and data glove or the CAVE (Cruz-Neira, D. J. Sandin et al. 1992) have captured the imagination of architects and students who see it as a medium for creating form by body gesture and for directly understanding the three dimensional spatial implications of proposed designs. However, virtual reality hardware has been slow to penetrate the commercial market. Desktop virtual reality, which involves panoramic images and interactive three dimensional models displayed on a CRT or LCD screen, enables us to explore the applications of VR technology, albeit without the immersive qualities.

Apart from simply enabling a designer to convey and experience—visually, interactively, and in three dimensions—an architectural space, virtual reality offers a natural way to organize design information spatially. This spatial integration of knowledge and artifact can promote more informed (and therefore better) design. The two examples here are about organizing information about historical places—in one case an archaeological

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

reconstruction, in the other the documentation of an historic building. The same techniques, however, may be applied to structuring information about yet unbuilt places, either to store general design guidelines or to explore design alternatives for a specific building project.

In the Ceren virtual archaeology project (Lewin, Ehrhardt et al. 1997), we reconstructed 3D models of structures at an archaeological site in El Salvador, working from field notes, logs, and drawings. The models include artifacts found at the site—pots, knives, metate grinding stones, and even chili peppers. The models are presented on a Web site using panoramic pictures with linked hot spots to detail images and field notes about specific items stored in a database. Any reconstruction can represent informed opinions but despite the best efforts to conform model making with archaeological record, it is not necessarily an accurate picture. Experts may differ about the correct interpretation of the data and alternative interpretations may result in different reconstructions. For example, in one of the Ceren structures, expert members of our team disagreed about the form of the roof—one argued for a hip roof and the other for a gable. When the dispute is difficult to resolve, rather than select a "correct" interpretation it may be valuable to make alternative interpretations available for discussion and argument. Thus we came to the idea that along with each alternative (resulting from a modeling decision) we could record the underlying rationale. That is, we make available alternative visualizations of the structures each annotated with the arguments for that particular configuration.

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**



Figure 7. The Ceren Virtual Archaeology Site

We took a similar approach in another virtual architecture project, the Hagia Sophia Web Resource. Here, unlike Ceren, the building (the Hagia Sophia church/mosque in Istanbul) exists, so the project focused on making three-dimensional imagery of the building more widely available via the Web and linking this imagery with interpretive information. We used the project as an occasion to build a more general tool kit for constructing "place based Web resources" in which information (text, photos, audio, and links to other Web sites) is embedded into the three dimensional representation of the building. The toolkit we built facilitates the construction of place based Web resources by providing the tools to assemble and integrate panoramic images, text, photos, and other multimedia data with a minimum of technical know-how.

Beyond the low-hanging fruit: Information Technology in Architectural Design past, present, and future



Figure 8. The Hagia Sophia Web Resource

Both these projects (Ceren and Hagia Sophia) use desktop VR to deliver information about an historic site. However, the same technology can be applied to organize information about future designs. One interesting application is to construct online databases of design information to advise architects of relevant concerns for specific building materials and systems, building types, and programs. Efforts have been made to gather case libraries about specific building types and deliver this information for architects to use (Domeshek and Kolodner 1992). However, case libraries have not found wide use in practice, arguably because although they offer information relevant to the task at hand it is structured in ways that make it unattractive for designers to access. Embedding the same information in a three dimensional model offers a potentially more effective delivery vehicle for knowledge designers need.

Discussion

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

We have argued that research in several areas of information technology can be fruitfully applied to architectural design. In particular research in Human Computer Interaction, Computer Supported Collaborative Work, and Virtual Reality has brought these fields to the point where they have the potential to significantly enhance architectural design.

However, the application of information technologies to architectural design must take into account the needs of designers. In each of these areas, some obvious applications have already been explored with relatively unsatisfying results. For example, providing a highly realistic digital and pen-driven simulation of paper media (e.g., the Painter application) misses a key opportunity of integrating drawing with computational power. Limiting collaboration tools to general purpose videoconferencing and a shared whiteboard ignores the more specialized activities of architectural design. Seeing virtual reality only in terms of creating realistic immersive images of designs fails to take advantage of the "information" in information technology. Therefore we argue that successful applications of innovative technologies such as, but not limited to, these three examples, must be based on an understanding of design process.

We recognize that schools of architecture cannot compete with industry and corporations in software research and product development. On the other hand, the software industry cannot focus on long term research efforts, nor is architectural design a large enough market (in comparison, for example, with mechanical and electrical engineering) to command significant development resources. These two factors suggest a role for university based research in information technology and design. Schools of architecture must capitalize on what they know and do best, and that is architectural design. By exploring new information technologies with an eye to how they can best be integrated

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

into the architectural design process, and by rethinking architectural design with an eye to how information technology might improve it, schools of architecture can find the fulcrum in this balance between research, development, and design.

Acknowledgements

This research was supported in part by the National Science Foundation under Grant No. IIS-96-19856. The views contained in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Abowd, G. D., C. G. Atkeson, et al. (1996). Teaching and Learning as Multimedia Authoring: The Classroom 2000 Project. ACM Multimedia '96. Boston, ACM: 187-198.
- Cruz-Neira, C., D. J. Sandin, et al. (1992). "The CAVE: Audio Visual Experience Automatic Virtual Environment." *Communications of the ACM* 35(6): 65-72.
- Do, E. Y.-L. (1998). *The Right Tool at the Right Time: Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools*, Georgia Institute of Technology.
- Do, E. Y.-L. and M. D. Gross (1995). *Drawing Analogies - Supporting Creative Architectural Design with Visual References*. Computational Models of Creative Design, Heron Island, Australia.
- Domeshek, E. and J. Kolodner (1992). *A case-based design aid for architecture*. Artificial Intelligence in Design. J. S. Gero. Netherlands, Kluwer.
- Gross, M. D. (1995). Indexing visual databases of designs with diagrams. *Visual Databases in Architecture*. A. Koutamanis, H. Timmermans and I. Vermeulen. Aldershot, UK, Avebury: 1-14.
- Gross, M. D. (1996). "The Electronic Cocktail Napkin - working with diagrams." *Design Studies* 17(1): 53-70.

**Beyond the low-hanging fruit:
Information Technology in Architectural Design past, present, and future**

Gross, M. D. and E. Y.-L. Do (1996). Ambiguous Intentions. Proceedings, ACM Symposium on User Interface Software and Technology (UIST '96). Seattle, WA, ACM SIGGRAPH and SIGCHI: 183-192.

Igarashi, T., R. Kadobayashi, et al. (1998). Path Drawing for 3D Walkthrough. UIST 98. San Francisco, ACM: 173-174.

Jung, T., E. Y.-L. Do, et al. (1999). Immersive redlining and annotation of 3D design models on the Web. Proceedings of Computer Aided Architectural Design Futures '99. G. Augenbroe and C. Eastman, Kluwer.

Kolarevic, B., G. Schmitt, U. Hirshberg, D. Kurmann, B. Johnson. (1998). An Experiment in Design Collaboration. ACADIA'98, Qubec City, Canada, Universal Printing.

Lewin, J., M. Ehrhardt, et al. (1997). Not Just Another Pretty Face: Image and argument in an archaeological web site. Computer Assisted Architectural Design Futures '97. R. Junge. Dordrecht, Kluwer Academic Publishers: 635-654.

Moran, T. P., L. Palen, et al. (1997). I'll get that off the audio": A case study of salvaging multimedia meeting records. CHI'97 Conference on Human Factors in Computer Systems., Atlanta, GA, ACM.

Qian, D. and M. D. Gross, eds. (1999). Collaborative Design with NetDraw. Proceedings of Computer Aided Architectural Design Futures '99. G. A. a. C. Eastman, Kluwer.

Schweikardt, E., and Gross, M. D (1998). "Digital Clay: Deriving Digital Models from Freehand Sketches.". Digital Design Studios: Do Computers Make A Difference? ACADIA 98. T. Seebohm and S. V. Wyk. Quebec City, Canada., ACADIA: Association for Computer-Aided Design in Architecture: 202-211.

Tolba, O., J. Dorsey, et al. (1999). Sketching with Projective 2D Strokes. UIST '99, Asheville, ACM.

Zelevnik, R., K. P. Herndon, et al. (1996). SKETCH: An Interface for Sketching 3D Scenes. SIGGraph '96 Conference Proceedings: 163-170.