Digital Clay:
Deriving Digital Models from Freehand Sketches.

Schweikardt, E., and Gross, M. D.

Digital Design Studios: Do Computers Make A Difference?
ACADIA 98

T. Seebohm and S. V. Wyk, eds.,

ACADIA: Association for Computer-Aided Design in
Architecture, Quebec City, Canada, 202-211

1998

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

Eric Schweikardt
University of Colorado-Boulder
schweika@colorado.edu

Mark D. Gross
University of Colorado-Boulder
mdg@spot.colorado.edu

Digital Clay: Deriving Digital Models from Freehand Sketches

During the initial stages of design, it is not uncommon to find an architect scribbling furiously with a thick pencil. Later in the design process, however, one might not be surprised to encounter the same individual in front of a computer monitor, manipulating three dimensional models in a series of activities that seem completely divorced from their previous efforts.

Armed with evidence that sketching is an effective design method for creative individuals, we also recognize that modeling and rendering applications are invaluable design development and presentation tools, and we naturally seek a connection between these methodologies. We therefore present Digital Clay, a working prototype of a sketch recognition program that interprets gestural and abstracted projection drawings and constructs appropriate three dimensional digital models.

Argile digitale: la dérivation de modèles digitaux à partir d'esquisses main-libre

Durant les phases initiales de la conception, on trouve souvent l'architecte en train de barbouiller furieusement avec un crayon épais. Plus tard dans le processus de design, cependant, on ne s'étonnera pas de rencontrer le même individu devant le moniteur d,un ordinateur, en train de manipuler des modèles trois dimensionnels lors d'une série d'activités semblant complètement séparées de ses efforts précédents.

Ayant des preuves que le dessin est une méthode effective de conception pour des individus créateurs, nous reconnaissons aussi que les outils électroniques servant à faire de la modélisation et des dessins de synthèse ('rendering') sont aussi utiles lors du développement et la présentation du design. Naturellement, nous cherchons une connection entre ces méthodologies. Nous présentons donc Argile Digital, un prototype fonctionnel d'un programme qui reconnaît les esquisses et qui interprète les dessins de projection abstraits, et construit des modèles trois-dimensionnels appropriés.

introduction

Few studies have looked closely at the effectiveness of extant computer-aided design applications; nevertheless their acceptance in professional environments is increasing rapidly. The capability of current software packages to quickly produce elegant renderings and detailed models is at the root of their usefulness as presentation tools, yet the debate continues about their role in initial design.

Many architects choose to begin projects using paper and pencil only to transfer that material to a computational environment for final polishing and presentation. It seems probable that the popularity of this rather redundant design method stems from a certain frustration with the initial design exploration tools provided by current CAD packages like AutoCAD and Form•Z. This phenomenon can be credited to a prevalent misconception about the inherent nature of computer graphics: that digital tools inherently demand a high level of precision and a lack of ambiguity. The acronym CAD evokes a common imagery of grid snaps, tight tolerances, and exactitude, and these inflexible constraints can severely inhibit the freedom and abstraction necessary for exploratory design. Many designers are content with the transition from pen and paper to mouse and monitor, but for some, this rift in media is inherently distracting, and the translation process often necessitates a copious amount of time spent redrawing and digitizing existing work. The motivation behind the development of Digital Clay is a desire to bridge the gap between initial physical sketching exercises like Figure 1 and subsequent development in a digital design environment, like Figure 2.

The community of practicing architects is not the only group affected by the CAD industry's ignorance of its users' needs. At the university level, students are engaged not only in learning to practice design, but are also expected to embrace new technologies as vehicles for their exploration. First year students with little or no design experience often find that CAD packages have a steep learning curve, and their unintuitive interfaces simply become another obscure convention to master.

Digital Clay, on the other hand, operates from the familiar convention of sketching so that proficiency with the application is almost immediate. By interpreting hand-drawn projection sketches as three-dimensional digital models, the application allows designers to combine their effective exporatory design techniques with advanced visualization software. We have constructed Digital Clay as a functional prototype to address the key concept of interaction between the sketch and the digital model. Consequently, we will not only explain the existing system but also the motivation behind its evolution, and outline a framework for further development of these ideas.

sketches and 3D digital models as tools

The act of sketching has long been embraced by architects as a versatile tool for exploratory design. The representation of mental images on paper often adds clarity to a design. Michael Graves notes that "the tension of lines on paper or cardboard in space has an insistence of its own that describes possibilities which perhaps could not be imagined in thought alone" (Graves 1981). According to Daniel Herbert, sketches are "the designer's principal means of thinking" (Herbert 1993). It follows that effective design software should address sketching as a possible method of input. Most current CAD packages include a basic uninterpreted two-dimensional sketch mode, but surprisingly, little work has been done to enable designers to use sketching as an interface to communicate three-dimensional forms to modeling software.

In a short and remarkable introductory essay to his examination of three Renaissance texts on perspective by Alberti, Dürer, and Viator, William M. Ivins Jr. (Ivins 1938) reviews the development of perspective drawing and descriptive geometry. He contends that perspective, or more generally, "the rationalization of sight," was the single greatest contribution of the Renaissance to modern society, and this bold assertion is borne out through the multiple techniques of art, science, and technology that have developed on the basis of perspective drawing and descriptive geometry. The representation of reality in the form of digital models can be thought of as an extension of the ideas of Renaissance perspective.

The acceptance of digital models as visualization tools by designers like Gehry, Mayne, and Eisenman is evidence of their usefulness, and most design schools now include the digital model in their repertoire of standard media. The recent explosion in the number of available three-dimensional digital modeling applications reflects not only their popularity, but also that the professional design community has begun to embrace these technologies as tools for design visualization.

Three-dimensional computer graphic models are more effective representations than two-dimensional drawings for comprehending physical space. Unlike a physical model, the revision of a computer graphic model does not require a time consuming material reconstruction. (It can be argued that a three-dimensional digital model is no more an effective visualization tool than a perspective drawing because the model is also presented in a two-dimensional projection. This makes sense intuitively; yet even a 3D physical object is sensed as a two-dimensional representation on the viewer's retina.) Construction and editing of a physical model is often time consuming, yet a three-dimensional digital browser, in contrast, provides for real time adjustment of not only a model's position and rotation, but also of its scale, color, camera lens, and lighting attributes. The availability of walk-through and virtual reality technologies enhances the versatility of digital models by providing a completely immersive environment, a visualization technique that is infeasible with conventional scale models. User-defined pre-scripted animations, natural extensions of a digital model, can further aid in not only spatial, but temporal interpretation of a three-dimensional object or scene.

In general, the presentation of a digitally modeled object comprises not only the model itself, but also images rendered from the model. Perhaps the greatest advantage of digital over physical models is that they can be easily transformed into photorealistic images. Design processes centered around the manipulation of physical models often rely on a series of abstracted prototypes, with construction of a detailed model only to take place when the design phase is completed. Digital models, on the other hand, can be rendered for evaluation at any point during design, and readily avail-

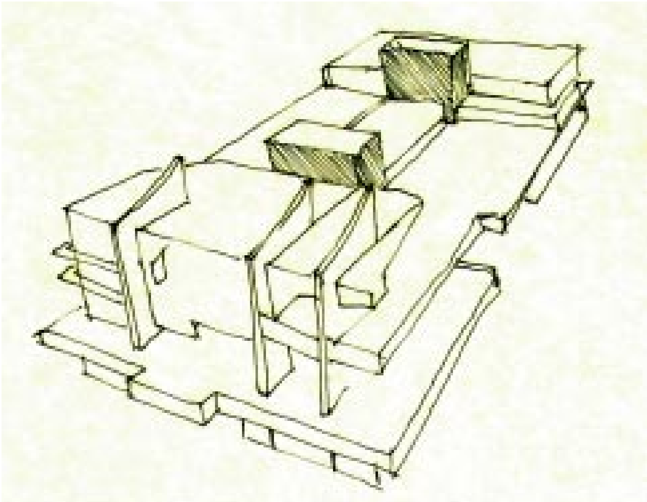

Figure 1. A typical architect's working sketch.

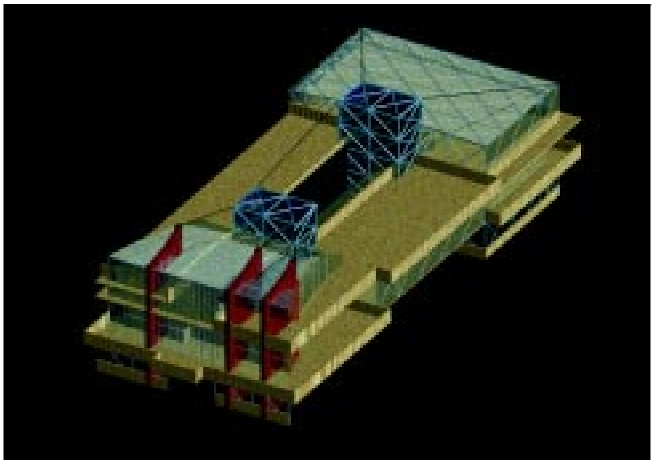

Figure 2. A digital model based on the same scene as Figure 1.

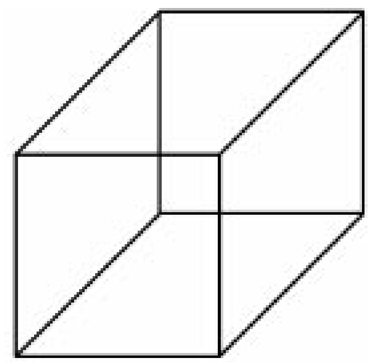

Figure 3. The Necker Cube. Note the ambiguity of the direction of projection.

able imaging applications can render scenes with quality and detail far superior to that of a physical scale model.

related work

Work at MIT in the 1970s began to address the problem of computer based sketch recognition. Johnson's Sketchpad III (Johnson 1963) added three-dimensional modeling to Sutherland's light-pen driven, constraint based Sketchpad program. Taggart's programs HUNCH and STRAIN (Taggart 1975) processed freehand drawing data from a light pen, filtered the data, latched endpoints of lines, and identified corners from changes in line direction. Neither Johnson nor Taggart, however, addressed the problem of interpreting sketches as representations of three-dimensional forms.

Zeleznik et al. implemented SKETCH, a gesture based interface to a three-dimensional modeler. (Zeleznik 1996) To instantiate a rectangular solid, the user draws a three-line gesture indicating one corner of the solid and the dimensions of the solid. Similar gestures are used to generate other solids and to initiate conventional editing commands such as move, rotate, and copy. SKETCH uses heuristics to resolve properties of the model that are ambiguous, for example continuing lines that are partially hidden by a surface closer to the viewer. SKETCH is a compelling argument for a gestural, as opposed to a menu based, interface to geometric modeling, but it addresses a different goal than Digital Clay. SKETCH, like the stylized unistroke Graffiti alphabet used on popular portable digital assistant (PDA) devices, demands that the user learn and use a new gestural code to indicate three-dimensional form. Digital Clay, on the other hand, interprets the traditional conventions of isometric and perspective sketching.

Do's 3D sketch tool (Do 1997) enables a designer to construct a three-dimensional form by drawing freehand on top, side, and front views as well as an axonometric. It restricts the designer to drawing in a two-dimensional plane, however, and then locates the 2D drawing in a 3D space. The Electronic Cocktail Napkin program (Gross 1996) supports freehand drawing input with a pen, but for the making and recognition of diagrams: graphical symbols and spatial relationships. Digital Clay, on the other hand, supports freehand input to describe three-dimensional form.

Sivaloganathan (1991) describes a system that extracts models from isometric drawings, but information beyond that contained in the sketch is required. Construction lines, center lines, and hidden lines must be specified through a conventional menu-based interface, and the model is constructed from these explicit drafting instructions. Digital Clay foregoes this type of supplemental annotation and derives models exclusively from the implicit information contained in a conventional sketch.

Recently, digital photogrammetry techniques have been developed that can construct a 3D model from two or more photographs of a building. However these techniques typically require a skilled operator to oversee the model construction process.

Although people are generally quite adept at interpreting pictorial representations of 3D space, construction of accurate 3D models from 2D scenes has remained a difficult problem in machine vision for the past twenty years. Marr (Marr 1982) hypothesized that humans possess a 2-1 / 2D sketch, or visible surface representation, which comprises a combination of the two-dimensional image on the retina and various mental annotations depicting shape and surface information. Depth is allowed only half a dimension because its measure is assumed from the two-dimensional image rather than shown explicitly like height and width.

interpreting a 2D sketch as 3D form

To interpret a two-dimensional projection of a three-dimensional scene, it is necessary to decipher which edges and vertices in the drawing project toward the viewer and which recede away. People are good at perceiving whether edges are convex or concave, but machines need to be explicitly programmed to perform this task. Figure 3 illustrates Necker's famous cube drawing with its two mutually exclusive interpretations. This image reminds us that projections of three-dimensional objects can be ambiguous, as do the familiar woodcuts of impossible buildings by M.C. Escher and other 2D / 3D curiosities such as those found in Gregory's *The Intelligent Eye*. (Gregory 1970).

We are exploring two methods for generating three-dimensional coordinates from a sketch. The first method makes use of an old machine vision algorithm, the Huffman-Clowes labeling scheme (Clowes 1971, Huffman 1971). Figure 4 depicts an annotated sketch in which the application considers the line drawing as a graph of edges. The algorithm labels the edges and vertices, for example by identifying a corner that projects toward the viewer as "+ + +' and one that recedes away from the viewer as "– – –". Since most drawings are unambiguous, it is possible to arrive at a labeling of the vertices that can provide a description of the three-dimensional shape of the vertices and edges.

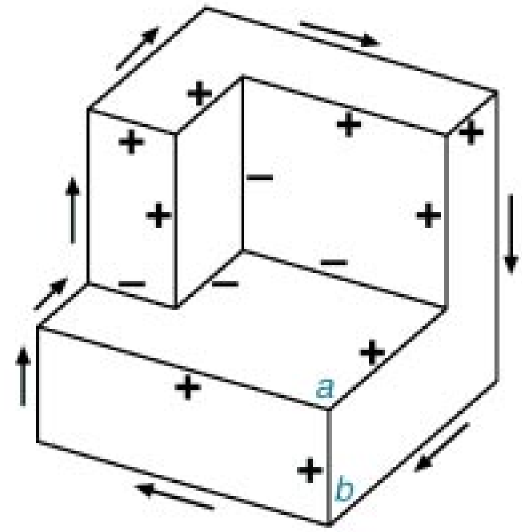

Figure 4. The Huffman-Clowes labeling scheme. Arrows indicate occluding edges, while + and - respectively indicate convex and concave edges.

The Huffman-Clowes labeling scheme requires that the line drawing meet certain well formedness requirements. Specifically, every line must connect to at least one other line at each of its endpoints. Digital Clay requires that the three-dimensional form consists only of solids; no planar surfaces without depth are permitted. Ambiguous and impossible shapes also cannot be interpreted as

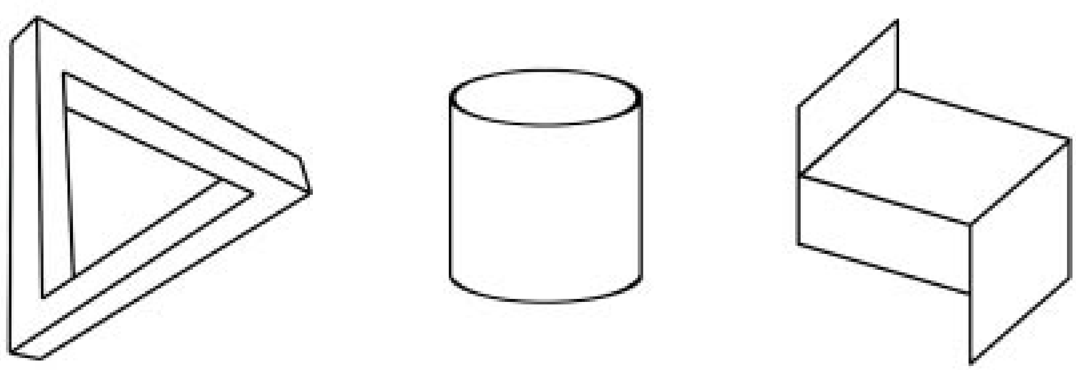

Figure 5. The Huffman-Clowes scheme prevents interpretation of objects such as these.

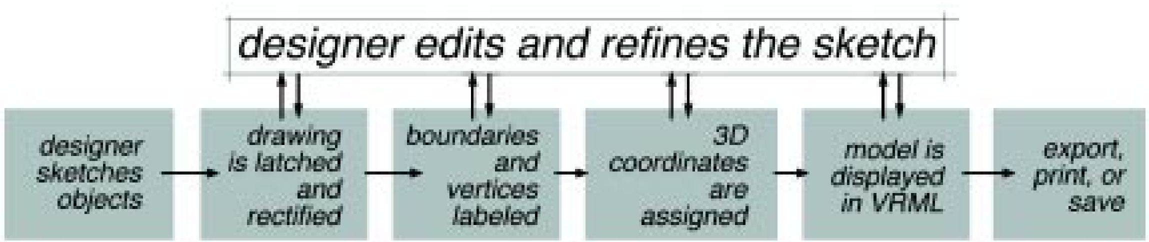

Figure 6. Diagram of the Digital Clay interpretation process.

feasible three-dimensional forms. Some of the limitations of the Huffman-Clowes labeling scheme are illustrated by the unparsable objects in Figure 5.

The second method we are using to infer three-dimensional coordinates from a sketch is based on the inherent rules that govern each type of drawing projection. The axes of isometric drawings, for example, are separated by equal angles, and perspective drawings feature the familiar foreshortening of lines which are closer to the viewer. By arbitrarily labeling one vertex as the origin, it is possible to move outward along each axis and label the coordinates of adjacent vertices. To use this algorithm for assigning coordinates, the type of drawing projection must be explicitly specified.

Figure 6 is a diagram of the current Digital Clay process. First the designer draws a freehand sketch. It is then latched and straightened. Next, the boundary edges are identified and the vertices are labeled indicating the three-dimensional orientation and coordinates on points in the drawing. Finally, the labeled drawing is rendered as a 3D model in Virtual Reality Modeling Language (VRML). The designer can then edit the object by drawing over the model to make changes.

At the outset, the designer is presented with a rather barren desktop environment. The *sketch window* fills the screen, and conventional toolbars and palettes are noticeably absent. The inherent simplicity of a physical sketchbook is certainly a component of its universal acceptance, and this metaphor is extended to serve as the basis for much of Digital Clay's interface design. Using a tablet and stylus, the designer draws conventionally in the sketch window until a satisfactory freehand representation of a three-dimensional object is arrived at (Figure 7).

Upon completing an acceptable first sketch, the user executes the *model* command, initiating a series of procedures that successively build upon the previous representations. Initially, through elimination of duplicates and the substitution of straight lines for linear pen strokes, edges are extracted from the image. These lines are subsequently connected to each other's endpoints, *latched*, through a rectification process that is similar to the *join* and *trim* commands available in most CAD packages. At this point, the sketch window contains a *latched sketch*, a hard-line approximation overlaid on the original image (Figure 8). By sketching over the original drawing or dragging lines and vertices, the designer can modify the program's interpretation to more closely resemble their initial intention.

Before Digital Clay derives a three-dimensional coordinate system from the two-dimensional projection, the constituent lines of the latched sketch must be rectified in order to accurately represent a correct projection of the intended object. The same rules which govern the assignment of coordinates to each vertex are now used to correct the user's imprecise angles. The rectification process simply applies these standards to the latched sketch in order to construct a representation that is not only spatially accurate, but also technically correct.

Once a feasible two-dimensional interpretation of the original sketch is derived, Digital Clay can begin to translate the image into a three-dimensional model. Before the Huffman-Clowes labeling scheme can be applied, certain preliminary information is required; therefore data objects are created that annotate edges and vertices with coordinate and boundary condition details. At this point, the only explicit information about each line is its location and whether it happens to be an occluding edge or not. In most cases, this data is sufficient to begin execution of the vertex labeling algorithm.

Digital Clay implements the Huffman-Clowes labeling scheme through a process of *constraint propagation*. At the outset, each vertex could be labeled with a relatively large number of possibilities. After boundary conditions are established, the number of constraints on the possibilities of a vertex's labeling increases, and the number of possible interpretations diminishes. As the number of possibilities for a vertex's labeling decreases, however, its neighboring vertices are affected too. For example in Figure 4, if boundary conditions are established and corner "a" is labeled "+ + +" indicating that it projects toward the viewer, then corner "b" must be labeled "+ ->->", indicating its orientation. In this manner, a series of constraints is propagated in a continuous loop from vertex to vertex until one interpretation is arrived at for each point. When this end condition is reached, the application possesses a representation of the spatial orientation of the object that can be displayed (Figure 9) as an annotated drawing reminiscent of Marr's 2-1 / 2 dimensional sketch.

At this point, Digital Clay possesses both a spatial interpretation of the sketch and a map of each vertex's coordinates in three-dimensional space. No matter what view was originally sketched, there is always ambiguity about the hidden faces; Digital Clay addresses this problem in two ways. By constructing a model with faces extracted from the known coordinates, a model can be rendered which contains only the visible faces and no assumptions about the hidden sides. A benefit of the Huffman-Clowes scheme, however, is the implicit assumption about the hidden portions of an object. By modeling with solids instead of surfaces, a series of cubes projected from convex vertices will naturally create an orthogonal face on each unspecified side. Where the attributes of a hidden face are unknown, the algorithm assumes the most simple case. The designer can override these assumptions if they are incorrect.

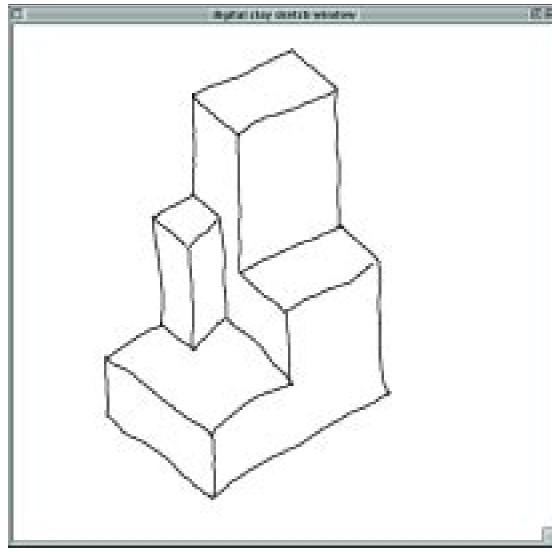

Figure 7. A raw sketch in Digital Clay's sketch window.

The three-dimensional coordinates are parsed into VRML code, and the object appears in a browser window for analysis and editing (Figure 10).

discussion and future direction

Digital Clay applies a well known technique from early artificial intelligence work in machine vision, an algorithm for converting a 2D representation of edges into a 3D model. This, clearly, is not the contribution of the current work. While well known as a machine vision technique, it appears that its application to sketch recognition was overlooked. Our contribution is to apply this algorithm as a means to support freehand drawing interfaces for 3D design. By working with freehand sketches, we provide an alternative to the structured and complex command language used by most CAD packages.

It also may be argued that a parsing approach like Digital Clay will never be able to deal with the richness and ambiguity of real life 3D sketches made by designers. We have been working with methods to resolve ambiguity through the designer's direct input, and we have also actively been experimenting with different drawing methods and emphases. For example, edge detection filters have proven effective in extracting dominant lines from noisy images and the latching process can be used

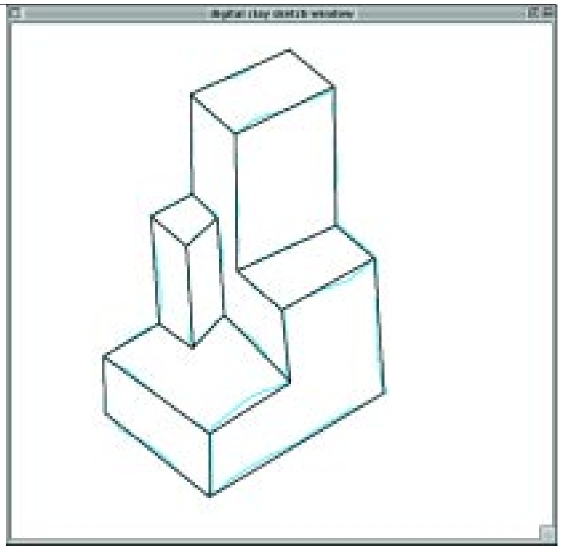

Figure 8. A latched version of the original sketch.

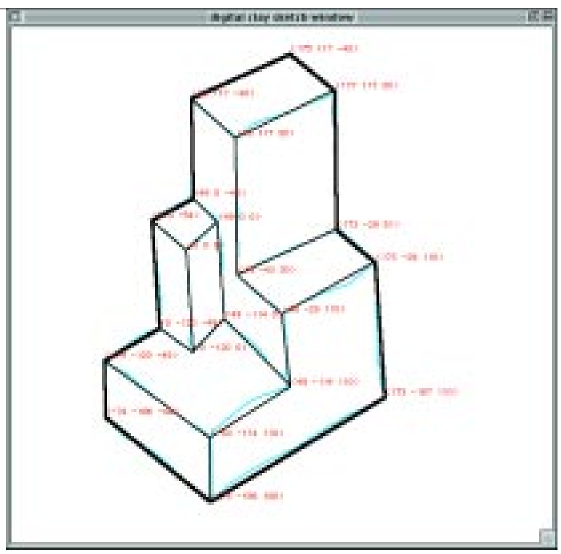

Figure 9. Projection with occluding edges darkened and coordinates displayed.

to resolve stylized drawings with numerous lines. Currently, Digital Clay is limited to line sketches, but many architects employ other techniques in their initial designs such as the use of symbols and text, and other sketching methods like the use of hatching to portray light and shadow. Our future work will address many of these issues.

We have experimented with Digital Clay using various input modalities: mouse, digitizing tablet, and a digitizing tablet with LCD display. We also plan to explore using PDA devices to input line drawings. Designers would use these as electronic sketchbooks to draw in the field and later convert their sketches to three-dimensional representations. Interface issues are not confined to hardware, though, and Digital Clay's software "look and feel" is not a trivial issue. We favor an entirely drawing-based interface: instead of traditional menus and dialogs, commands are carried out by sketching small symbols. This approach not only simplifies the screen layout and the learning curve, but also extends the metaphor of the analog sketchbook and allows the designer to be fully immersed in one medium, the pen-based interface. Another paradigm of interest is direct model editing. Dragging vertices to re-size rectangles is a standard procedure in computer-aided design applications, but to carry the principle of sketching further, we plan to implement the process of sketching directly on top of a VRML browser for editing and refinement.

An obvious area for future work lies in parsing the types of objects that are currently unacceptable. Unclear or sloppy sketches with overlapping or ambiguous lines have proven difficult to work with, but heuristics could be implemented to interpret a designer's intent, thereby automatically simplifying and editing a poor quality drawing. Objects featuring curved surfaces, which have proven elusive to most machine vision applications, are another category being addressed in our current research.

acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-96-19856 and the University of Colorado Undergraduate Research Opportunities Program.

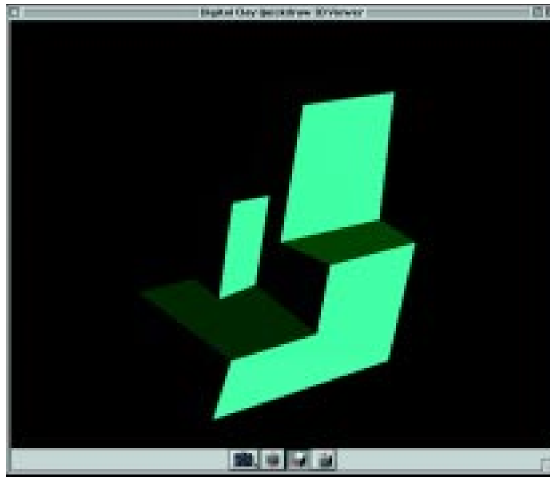

Figure 10. A model of Digital Clay's interpretation of the original sketch.

references

Arnheim, Rudolf, 1977. *The Dynamics of Architectural Form*. Berkeley, CA: University of California Press.

Ching, Francis D. K., 1995, 1997. *A Visual Dictionary of Architecture*. New York, NY: Van Nostrand Reinhold.

Clowes, M. B., 1971. "On Seeing Things." *Artificial Intelligence* vol. 2, p. 79-116.

Do, Ellen, 1995-1998. Personal communication.

Gregory, R.L., 1970. *The Intelligent Eye*. McGraw Hill.

Graves, Michael, 1981. "Le Corbusier's Drawn References". Introduction to *Le Corbusier's Selected Drawings*. New York: Rizzoli.

Gross, M. D., 1996. "The Electronic Cocktail Napkin - working with diagrams." *Design Studies* 17(1). pp. 53-70.

Herbert, Daniel M., 1993. *Architectural Study Drawings*. New York, NY: Van Nostrand Reinhold.

Huffman, D. A., 1971. "Impossible Objects as Nonsense Sentences." *Machine Intelligence* 6, B. Meltzer and D. Michie (eds.), Edinburgh University Press. pp. 295-323.

Huffman, D. A., 1977. "Realizible Configurations of Lines in Pictures of Polyhedra." *Machine Intelligence* 8, E.W. Elcock and D. Michie (eds.), Ellis Horwood, England, p. 493-509.

Ivins, William M. Jr., 1938. *On the Rationalization of Sight*. Da Capo Edition 1973.

Johnson, Timothy, 1963. "Sketchpad III: Three-dimensional Graphical Communication with a Digital Computer". *Proceedings of AFIPS Spring Joint Computer Conference* 23. pp. 347-353.

Marr, David, 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY: W.H. Freeman and Company.

Marr, David and H. K. Nishihara, 1978. "Representation and Recognition of the Spatial Organization of Three-dimensional Shapes." *Proceedings of the Royal Society of London, Series B*, volume 200, p. 269-294.

Sivaloganathan, S., 1991. "Sketching Input for Computer Aided Engineering." Doctoral Thesis, Department of Mechanical Engineering and Aeronautics, City University, London.

Sugihara, Kokichi, 1986. *Machine Interpretation of Line Drawings*. Cambridge, MA: MIT Press.

Taggart, James, 1975. "Sketching, an Informal Dialogue between Designer and Computer." *Reflections on Computer Aids to Design and Architecture*, N. Negroponte (ed), New York, NY: Petrocelli Charter, p. 147-162.

Waltz, David L., 1972. "Generating Semantic Descriptions From Drawings of Scenes With Shadows," Technical Report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Number AITR-271, p. 349.

Zeleznik, Robert C., K. P. Herndon, and J. F. Hughes, 1996. "SKETCH: An Interface for Sketching 3D Scenes." *SIGGRAPH'96 Conference Proceedings*, p. 163-170.