

A Freehand Sketching Interface for Progressive Construction and Analysis of 3D Objects

M. Masry, D. Kang, I. Susilo, and H. Lipson *

Sibely School of Mechanical and Aerospace Engineering
Cornell University, Ithaca, USA, 14853
{mark.masry, hod.lipson}@cornell.edu

Abstract

The possibility of using freehand sketching as the language for interactive design is a longstanding goal. The ability to sketch a 3D object, predict its performance, and redesign it interactively using physics-based feedback would bring the power of state-of-the-art analysis tools into the critical, early design phase. The enormous potential of sketch-based interfaces is widely recognized, and has been broadly pursued. The practical use of such attempts has remained limited because these interfaces have been primarily 2D, losing much of the benefit of mainstream 3D analysis potential. In order to become truly 3D, the spatial geometry must be automatically – and quickly – reconstructed from a single 2D sketch in near real-time. Once reconstructed, it can be converted into a model for simulation, and the simulation results interpreted back into the sketch. This paper presents a system that performs that reconstructs a 3D object from a freehand sketch, and uses the reconstructed object as the basis for a physical simulation. The system represents a first step towards fully interactive physics-based 3D design.

Introduction

Visual methods of communication are often the simplest and most efficient way of conveying information about the shape, composition and relationships of an object's components. Furthermore, visual information often transcends the limitations imposed by spoken or written languages, directly addressing a part of the brain capable of entirely different modes of thought. *Freehand sketching*, the informal drawing of shapes using freeform lines and curves, is one of the most ubiquitous forms of visual communication. Sketches can quickly and easily be created in order to convey shape information.

As pointed out by Ferguson (1992), it is revealing to watch how a designer, when given a design problem, instinctively reaches for a pencil and paper. Visual thinking is necessary in engineering: A major portion of engineering information is conceived, recorded and transmitted in a visual language. Many of the qualities that an engineer thinks are dealt with by a visual, non-verbal process. The informal,

written form of this means of expression is sketching. The significance of sketching to the design process is captured by comments made by Malo at an informal architectural design discussion forum (2002):

"...[Sketching] is not just a matter of "rendering," or of producing slick presentations, but rather about architectural thinking. It's "hand-eye coordination"—when we're talking not just about the perceptual eye, but the conceptual eye—the "mind's eye." ... If you can visualize it, you can draw it—without computer programs to do the visualization. One can only draw what one understands. If one can't draw it, one doesn't understand it. Period. As a design critic in the studio, this is the surest indication of a student's real understanding of what he or she is doing. Drawing is basic to architectural thinking and practice ... Drawing, as said, is about three-dimensional understanding—not merely about presentation technique."

Despite the abundance of computerized 3D graphics and CAD systems, plain pencil-and-paper freehand sketching has remained one of the most powerful and intuitive tools at the conceptual design stage. Conventional CAD user interfaces that deal with spatial construction are typically cumbersome to use and hamper creative flow. In a survey of adequacy of CAD tools for conceptual design, an industrial designer is quoted saying "The interface is just not for us. I can do thirty sketches on paper by the time it takes me to do two on the computer" (Puttre 1993). Freehand sketching, on the other hand, still provides one of the most fluent methods for conveying 3D information among designers, despite its reliance on an inherently 2D medium. Humans seem to be able to understand and discern 3D spatial concepts even when they are depicted on 2D medium in the form of sparse and inaccurate line drawings.

Paper-sketching also has many drawbacks: The viewpoint is fixed and cannot be changed in mid drawing; the sketch is passive and cannot be directly simulated or analyzed using computational engineering tools (e.g. structural analysis or kinematic simulation); the sketch is tentative and if a final, accurate model is desired, it must be recreated from scratch. The combination of freehand sketching with 3D reconstruction and physical simulation opens the door to a new world of design possibilities. Our understanding of spatial reconstruction and refinement, the necessary computational power for interactive-time reconstruction and simulation, and the

*This research has been supported in part by a Microsoft University Relations research grant for Tablet Computing
Copyright © 2004, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

availability of digital sketching hardware have all matured to the point where this new type of tool is within practical reach.

This paper presents an intuitive, pen-based sketching tool that has two goals:

1. to reconstruct a 3D object from a single, flat, freehand sketch, and
2. to simulate 3D kinematic behavior of the object in interactive time.

The proposed system uses two optimization-based reconstruction algorithms to achieve the first goal. Additional features can be added to the object post-reconstruction. Whenever possible, depth information for the strokes added post-reconstruction is inferred from the reconstructed objects planes and lines, rather than through subsequent optimization. This approach eliminates much of the difficulty involved in attempting to sketch a single projection of a complex 3D object that captures all of its projected elements, and reduces the computational complexity of the reconstruction involved in earlier approaches (Lipson & Shpitlani 1996). The same interface used to sketch and reconstruct the 3D object is used to perform 3D kinematic simulations. The governing kinematic equations are solved using a relaxation solver (Lipson 2004). Users with little or no training were very quickly able to make sketches of 3D objects and perform rigid-body experiments with the kinematic simulator.

Previous work

There have been several attempts to construct systems with sketch-based input. Stahovich, Davis, & Schrobe (1998) demonstrated a system that could interpret the causal functionalities of a two dimensional mechanism depicted in a sketch, and generate alternative designs. Davis (2002) recently showed a system that can simulate rigid-body dynamics of a sketched two-dimensional mechanism. These systems are mostly two dimensional, and the few that are 3D require additional steps that break the flow of sketching, and do not tie in physical analysis.

The problem of reconstructing 3D shapes from 2D sketches has also been the focus of much research. The reconstruction process is summarized in Figure 1, in which any arbitrary set of depths $\{Z\}$ that are re-assigned to the vertices in the sketch constitutes a 3D configuration whose projection will match the given sketch exactly. In principle, each such assignment yields a valid candidate 3D reconstruction. Optimization-based reconstruction methods determine missing depth values as the optimizing solution to a compliance function. Systems of linear equalities and inequalities are employed by Sugihara(1986) and Grimstead & Martin (1995) to characterize the 2D sketch with respect to an underlying 3D object. A generalized approach based on the 3D geometrical relationships of the strokes in a sketch was proposed by Lipson & Shpitlani (1996). These authors also presented statistical approaches to optimization-based reconstruction (2000; 2002). The correlation between the 2D angles formed by lines in the sketch plane and the angle between these lines in 3D space are learned from a large

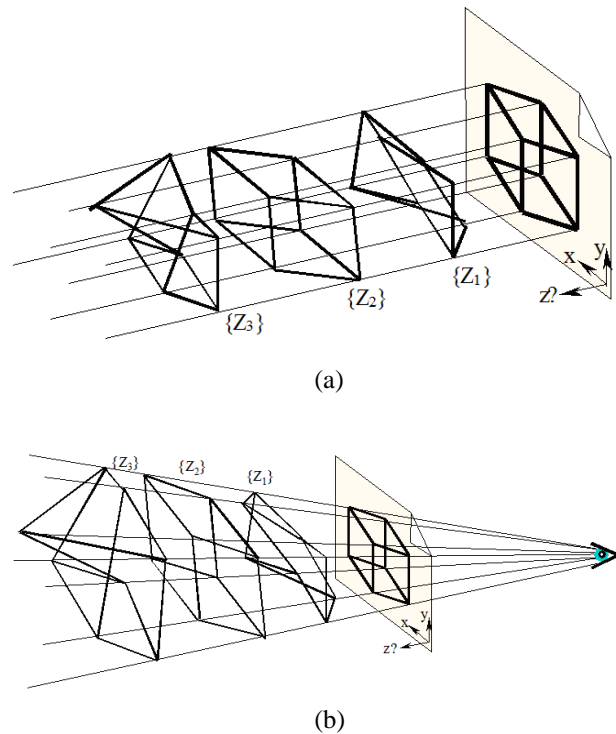


Figure 1: A sketch provides only two of the coordinates (x, y) of object vertices. A 3D reconstruction must recover the unknown depth coordinate z . (a) In parallel projections, this degree of freedom is perpendicular to the sketch plane; (b) in a perspective projection, it runs along lines that meet at the viewpoint. In either case, there are an infinite number of candidate objects – the problem is indeterminate. Each candidate object is represented by a unique set of Z coordinates, e.g. sets $\{Z_1\}$, $\{Z_2\}$ and $\{Z_3\}$

number of computer-generated 3D shapes and the corresponding projections of these shapes onto a viewing plane and used to determine the most likely 3D shape.

Approaches not based on optimization include those using line labeling (Huffman 1971; Clowes 1971), methods that analyze the relationship between the slopes of sketch lines and gradients of 3D faces (Mackworth 1973; Wei 1987), interactive methods in which 3D objects are incrementally constructed by attaching facets sketched by the user in 2D(Lamb & Bandopdhay 1990; Fukui 1998), interactive gesture-based systems (Zelevnik, Herndon, & Hughes 1996; Igarashi, Matsuoka, & Tanaka 1999), or systems based that assume that the scene is composed entirely of a limited set of known 3D primitives (Wang & Grinstein 1989). Though these methods each have advantages optimization-based methods are more general than those described above, and can be used to reconstruct 3D objects of varying complexity.

Sketching System

The interface paradigm of the sketching system is pencil and paper: The user requires minimal interaction with the program beyond what can be done with the pen itself. Users create a sketch by drawing a series of loosely connected strokes with the digitizer pen and can also erase strokes using the flip side of the pen, either before or after reconstruction of the 3D shape. An iterative algorithm is used to merge nearby vertices and transform the sketch into a connectivity graph with edges specified by the stroke lines. Reconstruction and 3D-spinning of the geometry are triggered when the user attempts to "drag" a vertex of the shape using a button on the barrel of the pen.

Following reconstruction, connected circuits of nearly coplanar strokes are identified and used as the basis for shading and hidden line removal. Vertices in the reconstructed shape may also be anchored (set as force sinks) for a real-time kinematic simulation which is activated when a vertex is dragged using the pen to apply a force. The kinematic equations are solved using an iterative relaxation method.

Reconstruction

A sketch of connected strokes can be interpreted as a connectivity graph with edges given by the strokes and vertices occurring at connection points. Given that the vertex (x, y) positions in the sketch plane are specified by the sketch, the reconstruction problem consists of assigning values along the z axis to each vertex in such a way that the spatial characteristics of the reconstructed shape optimize a set of reconstruction criteria. Two algorithms are used to reconstruct the 3D shape from the 2D sketch:

1. The sketch is first tested for the presence of one or more underlying axis systems using a histogram of the angles of the strokes in the sketch. The underlying axis systems are then reconstructed, and used to determine the depth of the sketch vertices. This process is described in more detail in these proceedings (Kang, Masry, & Lipson 2004). While this approach can reconstruct complex shapes very quickly, it cannot be used with sketches that do not conform to an underlying axis system, or that have disconnected components.
2. A more general but more computationally intensive reconstruction algorithm is used to reconstruct sketch elements to which the first algorithm cannot be applied, as well as all non-interpolated strokes added to the sketch following the initial reconstruction. This algorithm is based on the work by Lipson & Shpitlani (1996). The optimization cost function is the sum of separate cost functions given by the parallelism, isometry and orthogonality of the resulting 3D shape:

$$f(\mathbf{v}_m, \mathbf{v}_n) = (1 - |\langle \mathbf{v}_m, \mathbf{v}_n \rangle|) + \left(1 - \frac{\min(|\mathbf{v}_m|, |\mathbf{v}_n|)}{\max(|\mathbf{v}_m|, |\mathbf{v}_n|)}\right) + (|\langle \mathbf{v}_m, \mathbf{v}_n \rangle|) \quad (1)$$

where the vector $\mathbf{v}_n = (x_n, y_n, z_n)$ is the vector given by the difference between the endpoints of stroke n ,

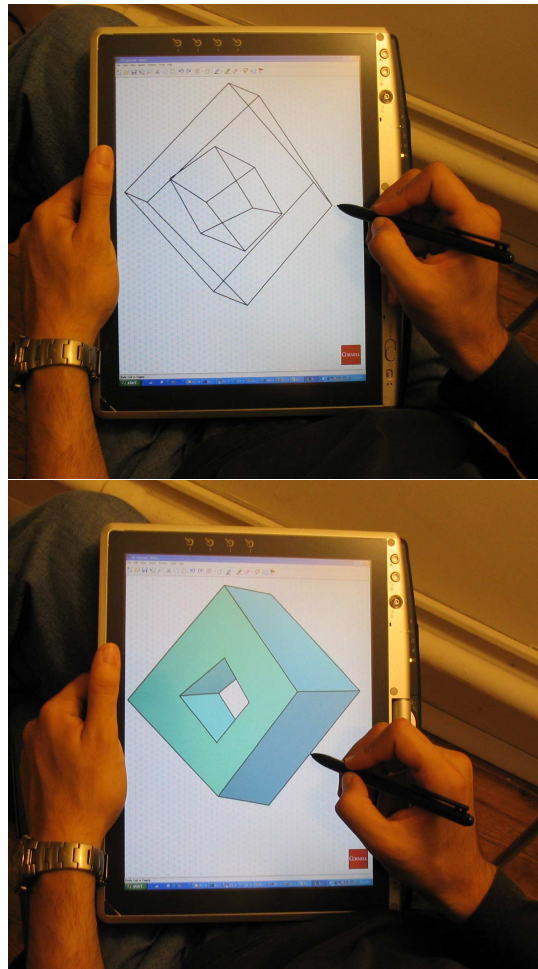


Figure 2: A user creating, rendering and rotating a cube with a see through hole using the proposed system on a Tablet PC

$\langle \mathbf{v}_m, \mathbf{v}_n \rangle$ is the normalized inner product of \mathbf{v}_m and \mathbf{v}_n , and $|\mathbf{v}_n|$ is the vector magnitude. The optimization cost for a sketch consisting of N strokes is given by $\sum_{m=1}^N \sum_{n=m}^N f(\mathbf{v}_m, \mathbf{v}_n)$. A hill-climbing optimization (Press *et al.* 2003) is used to minimize the total cost.

These algorithms run in interactive time. On a Pentium 4 Mobile PC this system can reconstruct sketches of 30 or more strokes in less than 0.1 seconds using algorithm 1, and less than 8-10 seconds using Algorithm 2. Once reconstructed, the 3D object can be rotated around any axis, and new strokes can be freely added.

Face Identification and Refinement

The reconstruction step outlined above generates only a 3D wireframe object. In order to complete the transition into a true solid, it is still necessary to identify which of the edge circuits constitute faces of the object, and what is the material side of each face. Several works (e.g. (Shpitlani & Lipson)), have defined methods of identifying faces in 2D

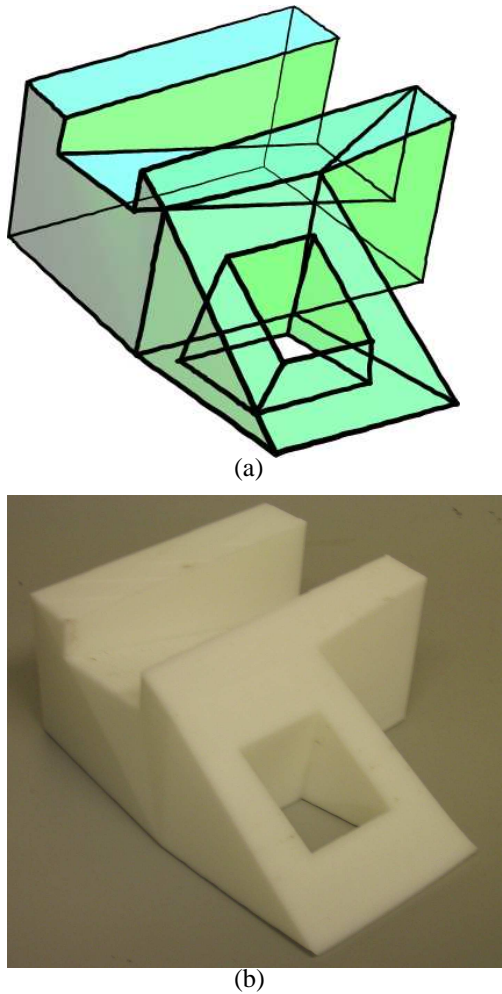


Figure 3: (a) A 3D shape with hidden lines visible created with the proposed system (b) A physical model of the shape extruded and printed on a 3D printer

sketches. These algorithms are computationally intensive and relatively complex to implement, however.

In this work, faces are identified by recursively searching the connectivity graph of the reconstructed object for roughly coplanar cycles in the reconstructed 3D object. The search begins with an initial stroke and randomly chooses another connected stroke to be the current stroke. An initial normal vector is given by the normalized cross product of the two strokes. The connected, unvisited stroke whose cross product with the current stroke has the highest projection onto the initial normal is chosen as the next stroke. The search recurses until a cycle is completed or until there is no connected stroke that is sufficiently coplanar with the current one. This search is performed beginning with all possible sets of two connected strokes until these have all been assigned to a planar face, or have been determined not to be sufficiently coplanar with any connected stroke.

Once faces have been identified, the endpoints of all new strokes that are added to the sketch following the initial re-

construction are classified as one of the following:

1. Overlapping an existing reconstructed vertex, in which case this endpoint is linked to the vertex
2. Lying along a stroke in the 3D object, in which case depth information for the endpoint is determined by interpolating along the 3D stroke
3. Embedded within one or more candidate faces, in which case the candidate face closest to the sketch plane is chosen and depth information from the endpoint is determined using the planar equation for that face. A cycle of vertices all of which are embedded into the same face are said to constitute descendant face. The algorithm maintains faces as a hierarchy that considers the descendants of a face at the top level of the hierarchy to be holes.
4. None of the above, in which case the depth information for the endpoint is determined by optimization using reconstruction Algorithm 2, holding fixed the optimization results for all reconstructed vertices.

The sketch is rendered by triangulating each of the faces, assuming that each descendant face is a hole, using a Delauney triangulator (Shewchuk 1996). A hidden line removal algorithm is used to determine which of the sketch strokes are occluded by faces, in order to facilitate 3D interaction by the user.

Kinematics

One of the most interesting applications for sketch reconstruction is in the domain of conceptual analysis. The 3D shapes reconstructed using the methods outlined above were used as the basis for kinematic simulations that will allow designers to sketch out mechanisms and simulate them, then remove, edit and add links by sketching additional strokes. The kinematics specified the distance traveled by each of the vertices connected by the elastic links. Each vertex was subjected to a gravitational force, and the user defined force applied by the rubber band.

Users began by “anchoring” one or more sketch vertices. Anchored vertices served as force skins for the kinematic simulation. The simulation was activated when users selected one of the sketch vertices and dragged it using the pen. The force applied to the vertex was proportional to the distance between the vertex and the pen point. This allows forces to be specified in two dimensions; an additional degree of freedom was required to specify a 3D force. The tablet hardware used in this work captures pressure information, which was used to specify the third dimension.

The kinematic equations governing the displacement of the sketch vertices in three dimensions were solved using an iterative relaxation technique that propagated forces through the connectivity graph specified by vertices and strokes in reconstructed 3D object (Lipson 2004). The vertex positions and the stroke vectors are thus functions of time. Each stroke was modeled as a spring with a stiffness constant K . The directional spring force exerted by stroke n on each of the two attached vertices as a result of its displacement from rest length at time t is given by

$$\mathbf{s}_n(t) = K (L_n(t) - L_n(0)) \mathbf{v}_n(t) \quad (2)$$

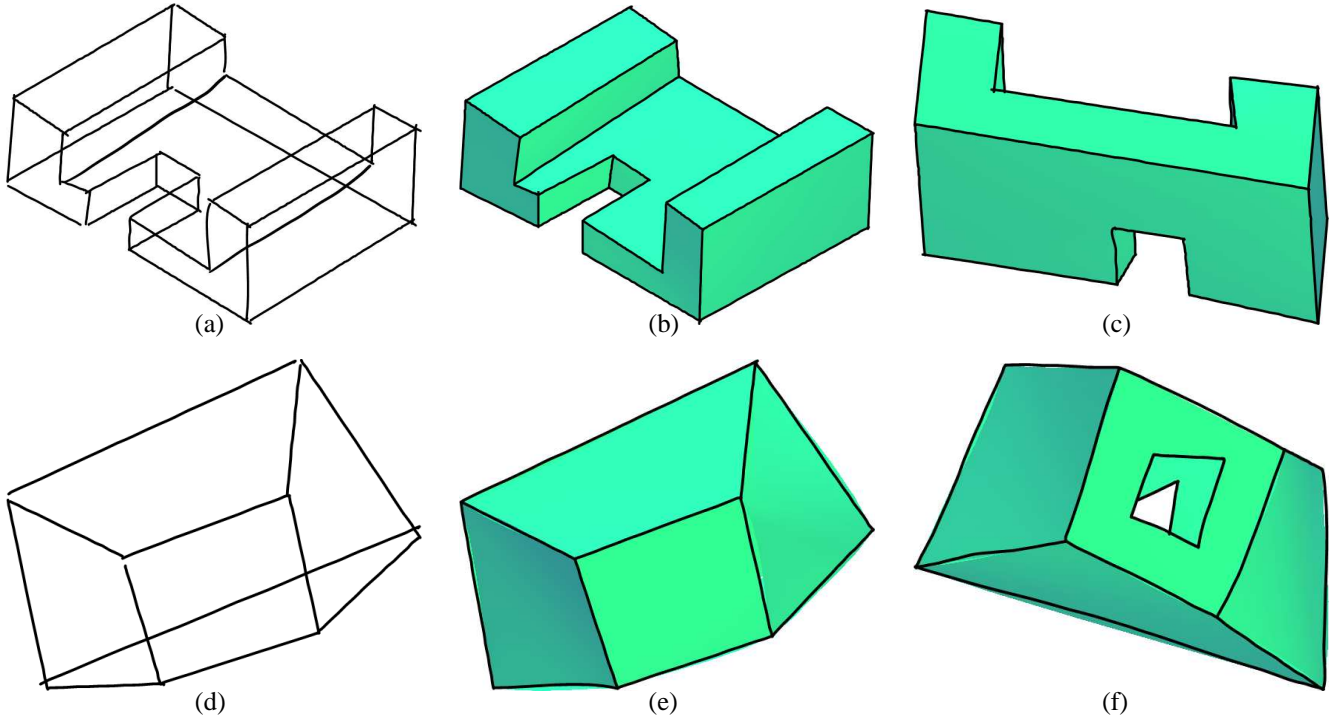


Figure 4: (a) A symmetrical unreconstructed 2D sketch (b) The sketch reconstructed using Algorithm 1 after shading and removal of hidden lines. Reconstruction time was approximately 0.1 seconds (c) An alternate viewpoint (d) An asymmetrical, unreconstructed 2D sketch (e) The sketch reconstructed using Algorithm 2 after shading and removal of hidden lines. Reconstruction time was approximately 4 seconds (c) An alternate viewpoint with holes added by sketching onto the object's faces

where $v_n(t)$ is the stroke vector at time t , $L_n(t) = |\mathbf{v}_n(t)|$, and $L_n(0)$ is the rest length. The forces \mathbf{F}_a exerted on vertex a at time t are determined by summing the directional spring force exerted by all of the attached strokes, gravitational force and any forces resulting from user input. The vertex displacement at time t as a result of the summed forces is given by

$$\mathbf{d}_a = \frac{\mathbf{F}_a}{stiffness_a} \quad (3)$$

where $stiffness_a$ is a function of the the number and orientation of all strokes attached to vertex a . This process was iterated at about a rate of 100 iterations per second to give users a sense of real time feedback.

An example session is shown in Figure 5. Nave users were able to construct multi-link multi-loop mechanisms in minutes and explore them in real-time, whereas several days of training and hours of construction would be necessary to perform similar activity on a conventional kinematic modeler like Pro/Mechanica or ADAMS. The kinematic interface may easily be used to perform other computations with existing solid modeling libraries (e.g. Parasolid or ACIS), or structural analysis software (e.g. publicly available Finite Elements codes) to provide realistic interactive-time prediction of physical behavior of the geometry (e.g. stress concentrations). Models of reasonable complexity can be

meshed and solved in less than a second with 1000 elements using commercial FEM libraries. This will allow the proposed system to be compatible with many standard CAE tools and other software, and be sufficiently responsive for interactive time performance.

Conclusions

This paper has presented the development of a system for progressively creating and analyzing 3D shapes using hand drawn sketches. Three-dimensional objects can be sketched as projections onto a sketch plane; depth values are assigned to each sketch vertex using one of two reconstruction algorithms. Further details can then be assigned to the resulting 3D object. The object can be used in interactive kinematic simulations. The proposed system has many applications both from the points of view of teaching and design. Further work will center on developing more sophisticated physics models for kinematic simulations, refining the human-computer interaction mechanisms and reconstructing objects containing curves and severe irregularities. A sample version of this program can be downloaded at <http://www.mae.cornell.edu/ccsl/>

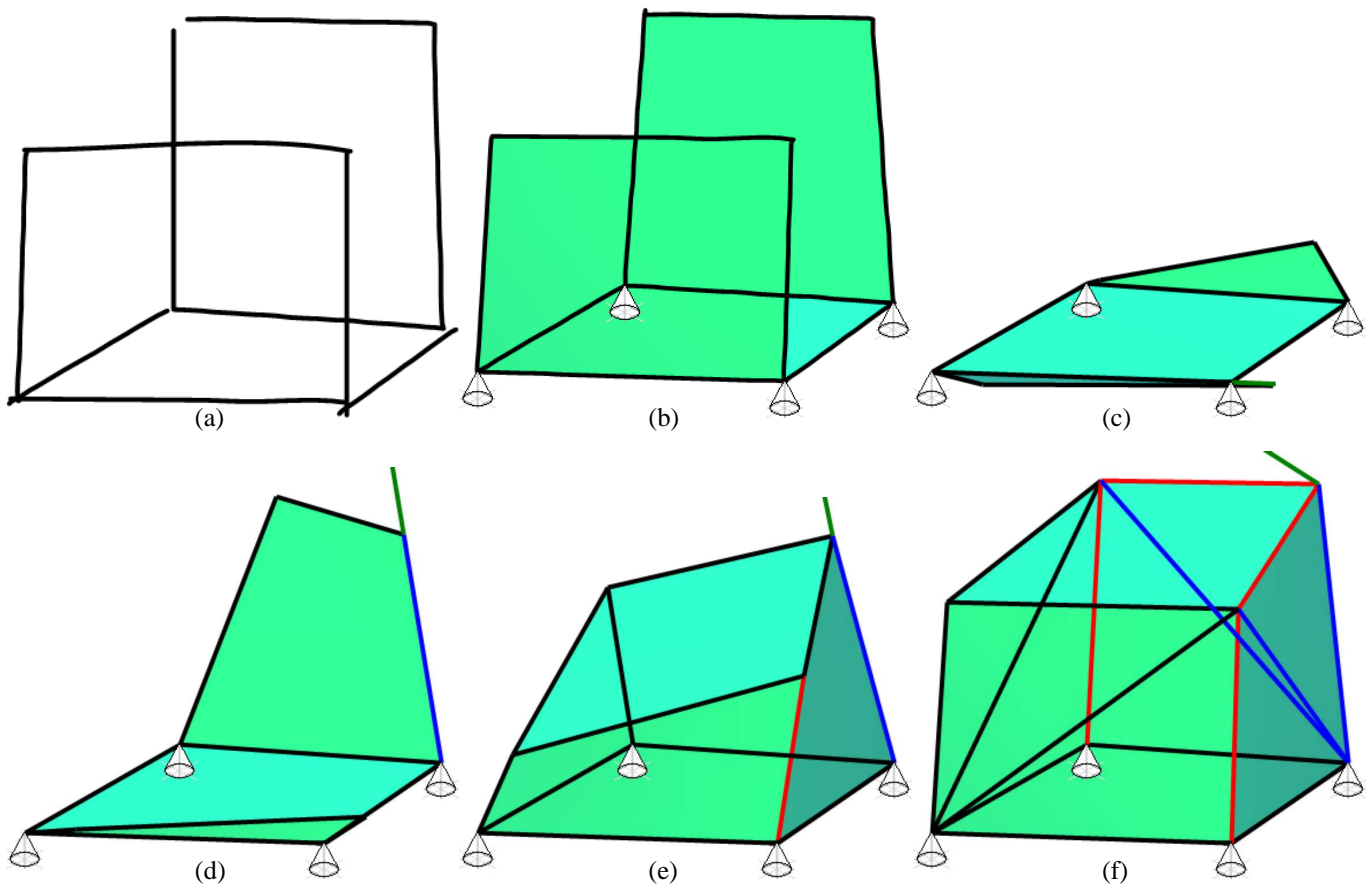


Figure 5: (a) An unreconstructed 2D sketch of a box missing 3 sides (b) The reconstructed shape displayed with hidden lines visible, anchored at 4 corners (c) The shape after several iterations of kinematic simulation with a gravitational force towards the bottom of the page (d) After several iterations of simulation with a user specified force at vertex (elements undergoing tension shown in red, elements undergoing compression shown in blue, force shown by green line, force direction is towards the top-left of the figure) (e) Same simulation but with missing sides added (f) Same simulation with bracing elements added to side faces

References

- Clowes, M. B. 1971. On seeing things. *Artificial Intelligence* 2(1):79–112.
- Davis, R. 2002. Position statement and overview: Sketch recognition at MIT. In *AAAI Sketch Understanding Symposium*.
- Ferguson, E. S. 1992. *Engineering and the mind's eye*. MIT Press.
- Fukui, Y. 1998. Input method of boundary solid by sketching. *Computer aided design* 20(8):434–440.
- Grimstead, I. J., and Martin, R. R. 1995. Creating solid models from single 2D sketches. In *Solid Modeling '95*, 323–337.
- Huffman, D. A. 1971. Impossible objects as nonsense sentences. In Meltzer, B., and Mitchie, D., eds., *Machine Intelligence*. Edinburgh: Edinburgh University Press.
- Igarashi, T.; Matsuoka, S.; and Tanaka, H. 1999. Teddy: sketching interface for 3d freeform design. In *Proceedings of SIGGRAPH*, 409–416.
- Kang, D. J.; Masry, M.; and Lipson, H. 2004. Reconstruction of a 3d object from main axis system. In *AAAI Fall Symposium Series: Making Pen-Based Interaction Intelligent and Natural*. To Appear.
- Lamb, D., and Bandopdhay, A. 1990. Interpreting a 3D object from a rough 2D line drawing. In *Proceedings of Visualization '90*, 59–66.
- Lipson, H., and Shpitlani, M. 1996. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Journal of Computer Aided Design* 28(8):651–663.
- Lipson, H., and Shpitlani, M. 2000. Conceptual design and analysis by sketching. *Journal of AI in Design and Manufacturing (AIEDEM)* 14:391–401.
- Lipson, H., and Shpitlani, M. 2002. Correlation-based reconstruction of a 3D object from a single freehand sketch.

In *AAAI Spring Symposium on Sketch Understanding*, 99–104.

Lipson, H. 2004. A computational relaxation method for simulating compound nonlinear mechanisms. *ASME Journal of Mechanical Design*. In Review.

Mackworth, A. K. 1973. Interpreting pictures of polyhedral scenes. *Artificial Intelligence* 4:121–137.

Malo, P. 2002. Computer vs. freehand drawing. In *Design Community Architecture Scrapbook*, message #1983.

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; and Flannery, B. P. 2003. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, second edition.

Puttre, M. 1993. Gearing up for conceptual design. In *Mechanical Engineering*, volume 93, 46–50.

Shewchuk, J. R. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Lin, M. C., and Manocha, D., eds., *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*. Springer-Verlag. 203–222. From the First ACM Workshop on Applied Computational Geometry.

Shpitlani, M., and Lipson, H. Identification of faces in a 2D line drawing projection of a wireframe object. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(10):1000–1012.

Stahovich, T. F.; Davis, R.; and Schrobe, J. 1998. Generating multiple new designs from a sketch. *Artificial Intelligence* 104:211–264.

Sugihara, K. 1986. *The interpretation of line drawings*. MIT Press.

Wang, W., and Grinstein, G. 1989. A polyhedral object's CSG-Rep reconstruction from a single 2D line drawing. In *Proceedings of 1989 SPIE Intelligent Robots and Computer Vision III: Algorithms and Techniques*, volume 1192, 230–238.

Wei, X. 1987. *Computer Vision Method for 3D Quantitative Reconstruction from a Single Line Drawing*. Ph.D. Dissertation, Department of Mathematics, Beijing University, China. (in Chinese; for a review in English see Wang and Grinstein, 1993).

Zelevnik, R.; Herndon, K.; and Hughes, J. 1996. SKETCH: An interface for sketching 3d scenes. In *Proceedings of SIGGRAPH*, 163–170.