

Sketching with Projective 2D Strokes

Osama Tolba Julie Dorsey Leonard McMillan

Computer Graphics Group
Laboratory for Computer Science
Massachusetts Institute of Technology

E-mail: {tolba, dorsey, mcmillan}@graphics.lcs.mit.edu

ABSTRACT

Freehand sketching has long had appeal as an artistic medium for conceptual design because of its immediacy in capturing and communicating design intent and visual experience. We present a sketching paradigm that supports the early stages of design by preserving the fluidity of traditional freehand drawings. In addition, it attempts to fill the gap between 2D drawing programs, which have fixed views, and 3D modeling programs that allow arbitrary views. We implement our application as a two-dimensional drawing program that utilizes a projective representation of points — i.e. points that lie on the surface of a unit sphere centered at the view-point. This representation facilitates the production of novel re-projections generated from an initial perspective sketch and gives the user the impression of being immersed in the drawing or space. We describe a method for aligning a sketch drawn outside the system using its vanishing points, allowing the integration of computer sketching and freehand sketching on paper in an iterative manner. The user interface provides a virtual camera, projective grids to guide in the construction of proportionate scenes, and the ability to underlay sketches with other drawings or photographic panoramas.

KEYWORDS: perspective, panoramas, illustration, vanishing points, grids, view alignment

INTRODUCTION

Despite the progress and sophistication of three-dimensional (3D) modeling in computer-aided design (CAD) systems, designers continue to use traditional media for rapid exploration of early design ideas [5]. This early stage of design exploration relies mainly on visual examination rather than precise quantitative specification [16, 20]. While CAD systems are suitable in the later stages of design for generating renderings and construction and manufacturing drawings, they divert attention from conceptualizing the design by forcing

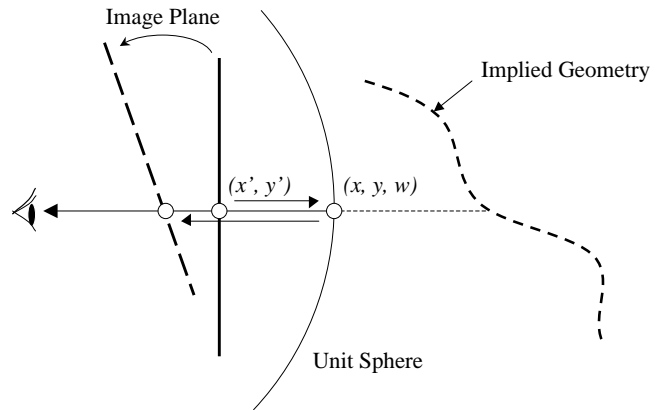


Figure 1: A drawing consists of a collection of strokes, each of which is a series of *projective* 2D points. As the user draws on the image plane, stroke points (x', y') are back-projected onto the unit sphere surrounding the viewer. Later, when the user changes the viewing direction and zoom level, these points are re-projected onto the new image plane forming the desired view.

the proposed model into precise dimensions. Most designers see CAD tools as too rigid, lacking the fluidity of the pencil or charcoal sketch [19]. Consequently, the use of computers in the early stage of design is often limited to two-dimensional (2D) sketching and image editing programs [25]. These programs, while possessing the desired ease-of-use, are restricted to depicting static views.

We propose a sketching approach that preserves many of the advantages of traditional freehand drawings, while simultaneously providing new capabilities that are beneficial for the explorative process. It utilizes strokes as the main drawing primitive, thereby allowing for fluid interaction. This is in contrast to 3D modeling programs, which struggle with interfaces for specifying 3D geometry and dimensions. Our approach improves on traditional 2D systems by providing capabilities that are currently thought to be strictly within the domain of 3D modeling. By re-projecting the strokes into new views, we simulate camera motion (panning, tilting, and zooming), thus providing the visual experience of being immersed in the designed space. At the same time, we

allow the user to draw complex geometric features such as trees, or annotate the drawing with comments and symbols. Such features and annotation are difficult to achieve using 3D geometry and remain a major shortcoming of conventional computer generated scenes.

Our approach also integrates paper sketching and computer drawing in a fashion that permits the selection of the desired mode at any point during the process. Paper drawings can be superior to ones made with computers because of their fluidity, immediacy, and portability. On the other hand, the computer facilitates re-projection, editing and refinement. Hence, this dual-mode approach combines the best of both worlds.

This paper is organized as follows: First, we review selected related research followed by an overview of the new representation and the implemented drawing system. Then we show preliminary results and examples of different types of drawings generated using our system. Finally, we discuss ideas for future work.

RELATED WORK

Many of today’s commercial illustration and image editing programs embrace fluid interfaces and emphasize ease-of-use as one of their design goals. They allow user input via freehand strokes as well as loosely specified geometry. Instead of coordinate entry they use drawing regulators such as grids and other guides. Toolbars, reminiscent of painters’ palettes, are used for switching between various interaction modes. We use many of these techniques in our implemented program.

Some commercial drawing programs incorporate rendering tools that emulate traditional media for the production of art with a hand-drawn look. Researchers have also investigated the production of this look, for example using textured strokes [13, 21]. (For a partial review of such rendering techniques see [15].)

The art of perspective drawing, formalized during the Renaissance, still constitutes a fundamental part of art and design education [6, 14]. However, the advent of 3D computer graphics has begun to overshadow it, and 2D computer graphics has failed to recognize its potential. We draw upon basic elements of this art, such as the use of vanishing points and projective grids.

Panoramic image-based rendering (IBR) systems use viewer-centric representations similar to the one we employ. For example, “QuickTime VR” represents environments with cylindrical panoramas and synthesizes perspective views through panning, tilting, and zooming [2]. Other IBR systems use 3D geometry in conjunction with images in order to facilitate viewpoint translations. For example, the “Tour into the picture” approach models a scene from a single photograph or drawing using a collection of planes and vanishing points [12]. IBR systems typically facilitate rapid navigation within a scene. In our approach, we provide controls for editing and

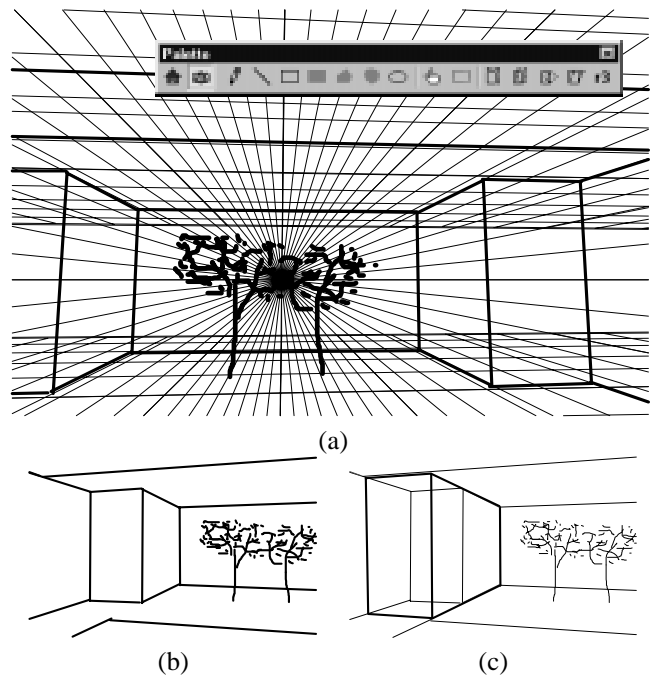


Figure 2: The user interface displays vanishing lines and projective grids, while a toolbar presents the user with camera control and drawing tools (a). Drawings can be layered in order to facilitate visual comparison of different design conditions. For example, the drawing in (b) is displayed in subdued color when viewed as an underlay to the design revision shown in (c).

modifying the scene’s underlying representation.

PROJECTIVE 2D STROKES

In traditional drawing programs, primitives are specified via a collection of points. Generally, these points are described by two coordinates, which can be imagined to lie on a two-dimensional plane. The coordinates specify the position of a point relative to a specified origin and two perpendicular basis vectors. In mathematical parlance, such points are considered two-dimensional Euclidean points.

This Euclidean representation of points is practically universal in all 2D drawing systems. There are, however, alternative representations of 2D points, which are not only more powerful than Euclidean points, but also contain them as a subset. In particular, the set of projective two-dimensional points can be represented using three coordinates in conjunction with the following rules: the origin is excluded, and all points of the form (a, b, c) and $\lambda(a, b, c)$, where λ is non-zero, are equivalent. The subset of projective points for which a value of λ can be chosen, such that $\lambda(a, b, c) = (\lambda a, \lambda b, 1)$, is the Euclidean subset.

There are several possible mental models for projective 2D points, which are comparable to the plane of the Euclidean points. We adopt a model in which all projective points lie on a unit sphere. Thus, the preferred representation of the point (a, b, c) is the one with λ chosen such that $a^2 + b^2 + c^2 = 1$.

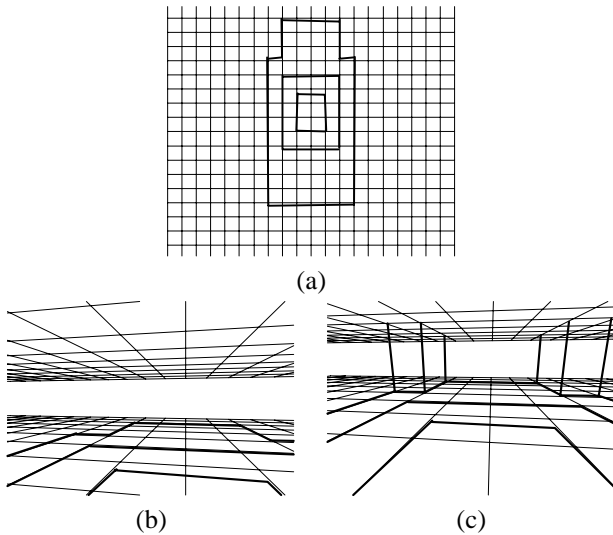


Figure 3: A floor plan can be drawn while looking down (a), then re-projected into perspective views for vertical line extrusion (b, c).

We will further restrict all values of λ to be non-negative. This additional restriction results in a special set of projective points called the *oriented* projective set [24].

The advantage of projective 2D points is the ease with which they can be manipulated. Unlike Euclidean points, translations of projective points can be described by matrix products, thus allowing them to be composed with other matrix products, such as scaling and rotation. However, projective points also permit re-projection to be described as a simple matrix product. This property of projective point representations gives unique capabilities to our two-dimensional drawing system.

Each stroke in our system is stored as a list of such projective points obtained by back-projecting drawn image points to lie on the surface of a unit sphere. The stroke also supports auxiliary attributes such as pen color and thickness. A drawing is a collection of such strokes. Our projective representation allows us to generate novel re-projections of the drawing (Figure 1). These re-projections can be interpreted as rotations and zooming about a single point in a three-dimensional space, similar to “QuickTime VR.” Re-projections of 2D projective points do not permit the changes in viewing positions that result in parallax changes. A technique for deriving the re-projection matrix associated with a particular three-dimensional camera motion (rotation and/or zoom) is described in the appendix. A conventional 3D graphics system could also be used to accomplish these re-projections.

SYSTEM OVERVIEW

In this section, we give an overview of our drawing system. First, we describe aspects of the user interface that make this a novel 2D drawing system. Then we discuss in detail how we integrate paper sketches with computer drawings.

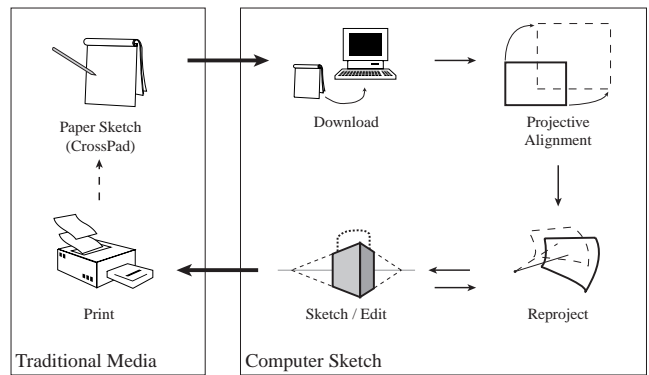


Figure 4: Integration of traditional media with computer sketching in a closed loop.

User Interface

The system has a simple user interface that allows a designer to input and manipulate freehand perspective sketches. Drawing tools are limited to freehand (pencil tool) or straight lines, the color and thickness of which are configurable (Figure 2.a). In the future we may add different brushes and stroke styles.

Virtual Camera. The camera’s rotation is controlled by *dragging* the drawing left/right and up/down, and the zoom level is changed while dragging via keyboard modifiers (shift and control). This direct way of interacting with the drawing allows the user to implicitly specify the camera’s rotation and field of view.

Projective Grids. Most 2D drawing programs display background Euclidean grids that help the user in the visual composition of a drawing. In contrast, we provide projective ground plane grids that can be interpreted as floor and ceiling (or any other horizontal surface). Although not implemented in the current system, projective grids can be configured to align with any surface, thus allowing the user to construct proportionate drawings on vertical and inclined surfaces. In addition to grids, our interface displays vanishing points and vanishing lines that aid as approximate directions.

Backdrops. Our system facilitates two types of backdrops: stroke-based and image-based. The stroke-based backdrop is analogous to the traditional underlay of trace paper, and is drawn in any color the user desires (Figure 2.b,c). An underlay is typically used for either refining design ideas or for drawing on top of drawings of pre-existing site conditions. The other type is the image-based backdrop, which can be a single photograph or a panorama that is generated from site photographs using any standard image stitching program (Figure 9). Image backdrops may also be synthesized by traditional modeling and rendering programs, in which case our system can be used for quick design reviews and annotation.

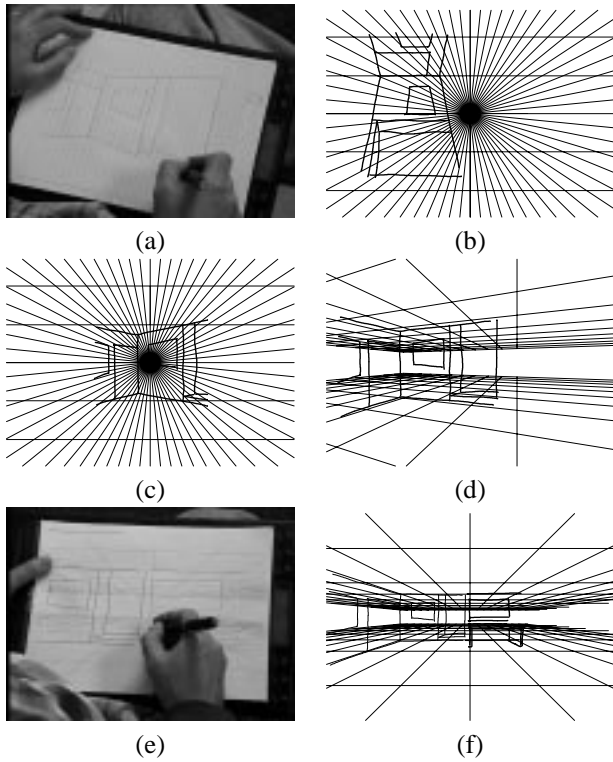


Figure 5: Aligning an imported sketch: A person starts sketching with the digital notepad (a), then imports the drawing into the system (b) and translates and rotates it to fit the horizon (c). Manual or automatic tools are used to align the vanishing points (d). The sketch can be printed with a different view and new strokes added on paper (e), then the import/align procedure is repeated (f).

Limited Orthographic Views

There are many ways drawings can be constructed in the system. The technique described in this section may be used when the initial design is expressed in orthographic views, such as plan or elevation, and a perspective visualization is required. The projective representation of this system provides the ability to draw orthographic views (provided they depict coplanar features) and perspective views in the same interface. For example, if the view points down then all input strokes can be envisioned to lie on a single horizontal plane, such as the floor plan. Later, when the drawing is viewed in perspective, vertical lines can then be extruded from the plan (Figure 3).

Paper Sketches

As mentioned above, the designer needs to sketch freely, both while using the computer and away from it. We have used the CrossPad portable digital notepad [3] that records strokes while the pen deposits ink on paper, but we have expanded the system to include perspective views drawn with the pad (Figure 4). Since the camera rotation and field of view are unknown for the imported drawing, we provide both interactive and semi-automatic tools for defining these attributes.

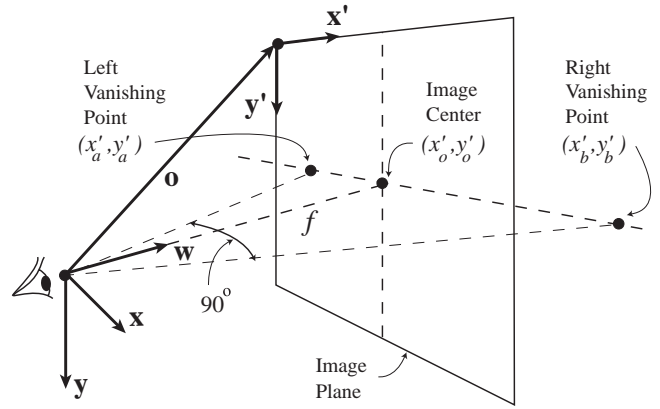


Figure 6: Viewing geometry for aligning two-point perspective drawings.

A typical scenario for aligning the imported drawing interactively starts with the user translating and rotating it to fit the horizon, followed by panning and zooming to align the vanishing points. The drawing may later be printed with a different view and new strokes added on paper, and then the import/align procedure may be repeated (Figure 5). In this way, the user may opt to use the pad at any point during the design stage.

The system also provides a semi-automatic tool for aligning perspective drawings containing two vanishing points. This tool requires that the user centers the drawing and specifies its two vanishing points. The system then computes the camera rotation and focal length using the viewing geometry depicted in Figure 6. Since the vanishing points correspond to two mutually orthogonal directions lying in the $x-w$ plane, the rays from the eye point to the vanishing points are orthogonal (their dot product is equal to zero). This gives us the following equation that we use to compute the focal distance f :

$$x_a x_b + f^2 = 0,$$

where x_a and x_b are computed directly from the image coordinates of the vanishing points and the image center as follows:

$$x_a = x'_a - x'_o, x_b = x'_b - x'_o$$

RESULTS AND APPLICATIONS

We present four examples showing results that are difficult to attain with other systems. They illustrate different techniques of drawing and system capabilities. In addition, they provide examples of the various uses for this system in a variety of disciplines, such as landscape architecture and urban design, classical archaeology, and freehand drawing of interior and exterior spaces. More generally, the examples show that drawing in our system can be either speculative or documentary.

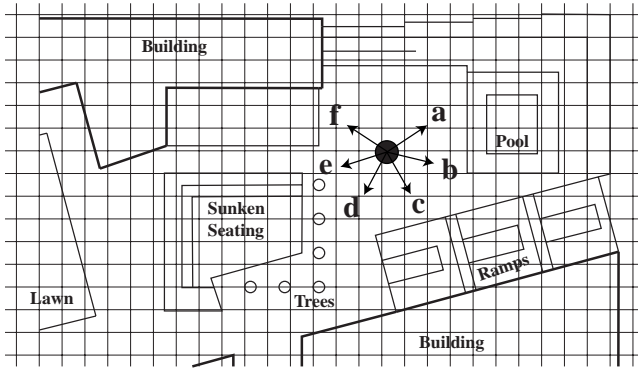


Figure 7: Partial plan of an outdoor plaza from which the buildings were extruded. A panning sequence of this space is shown in Figure 8.

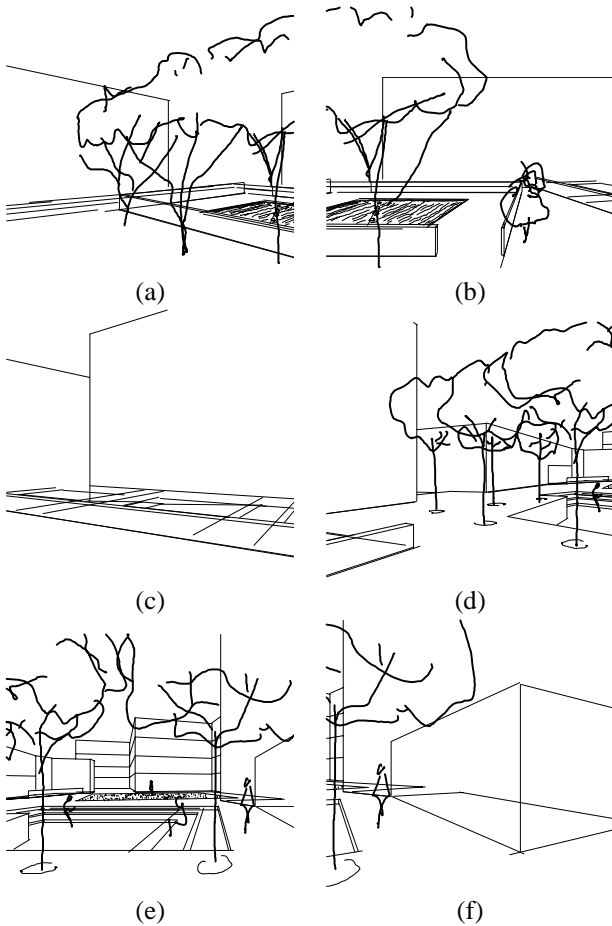


Figure 8: Visualization of the architectural design of an outdoor plaza via a panning sequence generated with our system (see Figure 7 for gaze directions).

Outdoor Plaza. This drawing was created using the floor plan technique applied to a conceptual landscape architectural design of an outdoor plaza. First, the plan was drawn using the grids in two rotations (Figure 7). Then the buildings were extruded and the abstract plants and human figures added quickly. Figure 8 shows a complete panning sequence of this space. Such a sequence is commonly desired in design studies and is often generated by hand.

Peirene Fountain. The second example shows the use of panoramic image backdrops and a sketch underlay. These techniques are applied to the study of a Greco-Roman fountain building in Corinth, Greece. The objective was to visualize the restored elevation of the Early Roman Period as depicted in the Corinth Series [23]. Tracing over a cylindrical panorama created a drawing of the existing conditions. The restored elevation was then imported into the system via a digitizer tablet and aligned with the previous drawing as an underlay (Figures 9, 10). In a real application, the archaeologist would use the system to study the restoration in conjunction with views of the existing conditions and resolve any conflicts that might arise.

Library Interior. This example shows a panoramic sketch created entirely from freehand sketches originally drawn on paper. The panorama was assembled from sketches pointing at four different directions by estimating the fields of view visually (Figures 11, 12).

Campus Court. The final example shows a panoramic sketch by a different artist. It is drawn more methodically with attention to detail, thereby exhibiting a slightly different quality than the previous example. The artist used a photographic panorama as a backdrop while constructing this drawing (Figures 13, 14).

Applications

This system can be useful to designers in various disciplines and situations. Interior designers, architects, landscape architects, and urban designers may use it for recording site conditions and sketching new design ideas and alternatives. They may also use it in design reviews, where the image backdrop is generated from a CAD model, and the system is used for design critique and annotation. Archaeologists and cultural resource managers may use it for recording and annotating existing site conditions as well as exploring and communicating historical reconstruction ideas. It is also conceivable that students wishing to learn perspective drawing will use such a system to enhance their understanding of the dynamics of the perspective projection. Other applications include animation, set design for the theater and cinema, and illustrative panoramas for public exhibitions [18].

DISCUSSION AND FUTURE WORK

In the short term, our system may evolve into a sort of digital sketchbook with multiple pages resembling different design concepts or the refinement of a single concept. Such a sketch-



(a)



(b)

Figure 9: Restored elevation of the Peirene Fountain (darker lines) aligned with a drawing of existing site conditions (a) and a photographic backdrop (b).

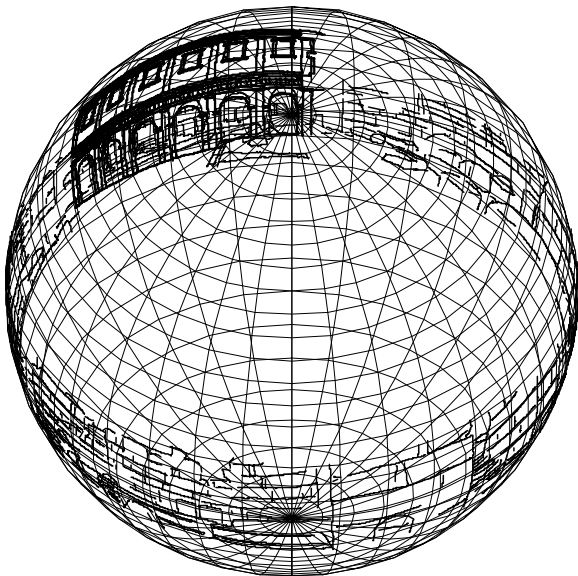


Figure 10: Sketch of existing conditions of the Peirene Fountain Building and restored elevation (darker lines) shown as points on the unit sphere.

book should allow tracing over previous pages and transferring strokes between pages. Our short-term goals also include enhancing the set of pens and brushes to allow more expressive drawings. In addition, sketches often include broad tone or colored regions that are better represented as filled or textured polygons.

We see the existing technology for input devices as limiting the creative process and may extend our research into building hardware for sketching. Today's digitizer tablets are often too small and divorce the hand from the display feedback. The portable digital notepad we used, although useful for on-site drawings, lacks the ability to quickly display new views. A better device would incorporate input into a flat panel display that is significantly large and oriented like a drafting board, in a manner that allows drawing freely at an arm's length.

Our work addresses the general need for better design tools to bridge the gap between the designer and computer. We approached the problem by creating a new drawing paradigm, based on projective 2D points, and incorporating one form of traditional design media. New input devices and other traditional media, such as wood models, also need to be seamlessly integrated with computer-aided design.

Other researchers have approached this problem by attempting to infer 3D models from sketches. Most systems generate 3D models by interpreting perspective and axonometric drawings using various techniques [1, 11], or through gestural interfaces [7, 26]. However, some sketching systems rely on the user's specification of 3D coordinates directly, for example, by drawing on pre-defined planes [9, 17] or using immersive environments (VR) with 3D manipulators [4]. In addition, while the technique of painting on 3D shapes does not address the creation of the shapes themselves, it offers a quick and intuitive way of modifying their appearance [10]. The proposed approach has the same directness of these methods without the internal 3D representation. We do not, however, preclude the possibility of using our system as a starting point for producing 3D models. Finally, while non-photorealistic rendering of 3D models can simulate the look of traditional design drawings through various techniques [8, 15], it does not address the creation of the computer models themselves.

ACKNOWLEDGMENTS

We would like to thank Duke Reiter and Max Chen for helpful discussions. Thanks to Stephen Duck for the drawing in Figure 14. Photographs of the Peirene Fountain were provided by the Corinth Computer Project of the University of Pennsylvania Museum, David Romano, director. This work was supported by an Alfred P. Sloan Research Fellowship (BR-3659), NSF CAREER awards (CCR-9624172 and CCR-9875859), and an NSF CISE Research Infrastructure award (EIA-9802220).

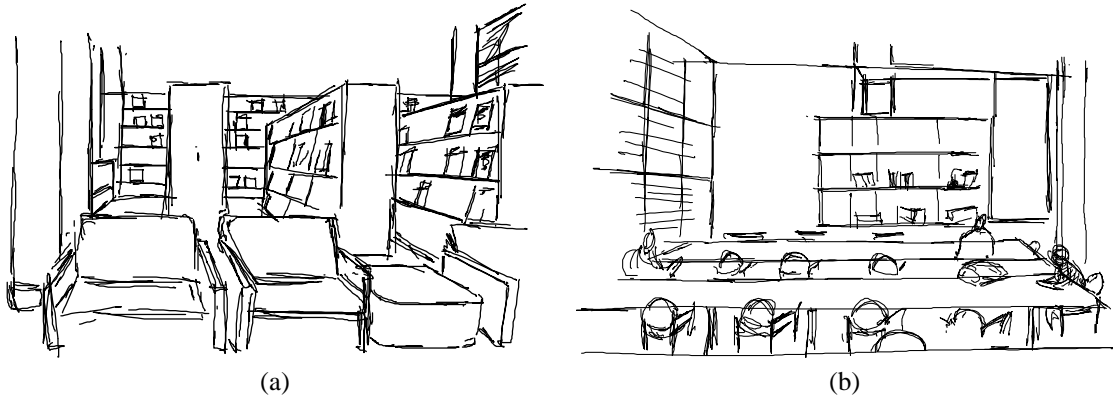


Figure 11: Two of the four input sketches used to generate the library interior panorama depicted in Figure 12.



Figure 12: Panorama of library interior shown as an unrolled cylinder.



Figure 13: Photographic panorama of Killian Court at the M.I.T. campus.

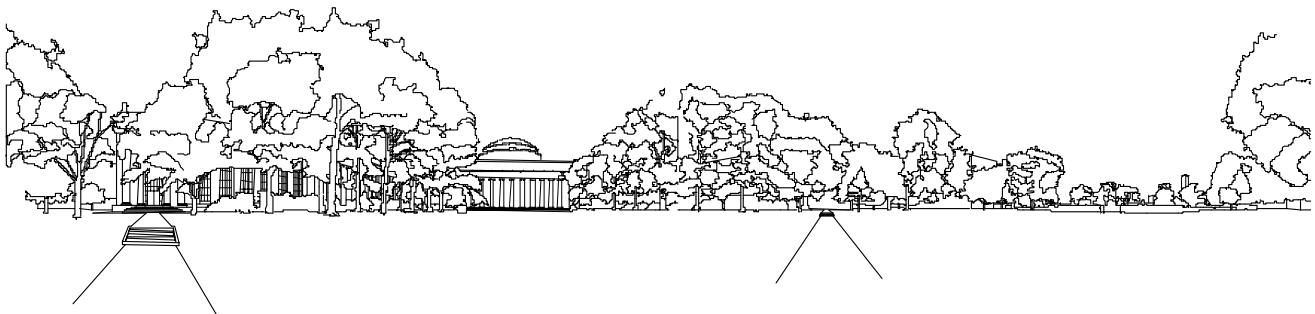


Figure 14: Artist's rendering of the panorama in Figure 13.

REFERENCES

1. Branco, V., A. Costa and F. Nunes Ferreira. Sketching 3D Models with 2D Interaction Devices. *Computer Graphics Forum*, Eurographics '94 Conference issue, 13(3): 489-502, 1994.
2. Chen, S. Eric. Quicktime VR - An Image-Based Approach to Virtual Environment Navigation. In *SIGGRAPH '95 Conference Proceedings*, pages 29-38, August 1995.
3. *CrossPad*. Portable digital notepad. Cross Pen Computing Group.
4. Deering, Michael F. HoloSketch: A Virtual Reality Sketching / Animation Tool. *ACM Transactions on Computer-Human Interaction*, 2(3): 220-238, September 1995.
5. Dorsey, Julie and Leonard McMillan. Computer Graphics and Architecture: State of the Art and Outlook for the Future. *Computer Graphics*, 32(1): 45-48, February 1998.
6. Dürer, Albrecht (1471-1528). *The Painter's Manual*. Walter L. Strauss (trans.), Abaris Books, New York, 1977.
7. Eggli, Lynn, Ching-yao Hsu, Beat D Bruederlin, and Gershon Elber. Inferring 3D Models from Free-hand Sketches and Constraints. *Computer-Aided Design*, 29(2): 101-112, 1997.
8. Gooch, Amy, Bruce Gooch, Peter Shirley and Elaine Cohen. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In *SIGGRAPH '98 Conference Proceedings*, pages 447-452, July 1998.
9. Hale, Brent J., Robert P. Burton, Dan R. Olsen, and W. Douglas Stout. A Three-dimensional Sketching Environment Using Two-dimensional Perspective Input. *Journal of Imaging Science and Technology*, 36(2): 188-195, March/April 1992.
10. Hanrahan, Pat and Paul Haeberli. Direct WYSIWYG Painting and Texturing on 3D Shapes. *Computer Graphics*, 24(4), August 1990.
11. Herot, Christopher F. Graphical Input through Machine Recognition of Sketches. In *SIGGRAPH '76 Conference Proceedings*, pages 97-102, July 1976.
12. Horry, Youichi, Ken-ichi Anjyo and Kiyoshi Arai. Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image. In *SIGGRAPH '97 Conference Proceedings*, pages 225-232, August 1997.
13. Hsu, Siu Chi and Irene H. H. Lee. Drawing and Animation Using Skeletal Strokes. In *SIGGRAPH '94 Conference Proceedings*, pages 109-118, July 1994.
14. Ivins, William Mills. *On the Rationalization of Sight, with an Examination of Three Renaissance Texts on Perspective*. Da Capo Press, New York, 1973.
15. Lansdown, John and Simon Schofield. Expressive Rendering: A Review of Nonphotorealistic Techniques. *IEEE Computer Graphics and Applications*, 15(3): 29-37, May 1995.
16. Lawson, Bryan. *How Designers Think*. Butterworth Architecture, London, 1990.
17. Marshall, Tony B. The Computer as a Graphic Medium in Conceptual Design. *ACADIA '92*, pages 39-47, 1992.
18. Oettermann, Stephan. *The Panorama: History of a Mass Medium*. Deborah Lucas Schneider (trans.), Zone Books, New York, 1997.
19. Radford, Antony and Richard Coyne. Architectural Computing: CAD versus 2B Pencil for Sketch Design. *Architecture Australia*, volume 78, pages 28-29, May 1989.
20. Robins, Edward. *Why Architects Draw*. MIT Press, Cambridge, Massachusetts, 1994.
21. Salisbury, Michael P., Michael T. Wong, John F. Hughes and David H. Salesin. Orientable Textures for Image-Based Pen-and-Ink Illustration. In *SIGGRAPH '97 Conference Proceedings*, pages 401-406, August 1997.
22. Semple, J. G. and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, London, 1952.
23. Stillwell, Richard. *Architecture*. Corinth Series, 1(2), Harvard University Press, pub. for the American School of Classical Studies at Athens, Cambridge, Mass, 1941.
24. Stolfi, Jorge. *Oriented Projective Geometry: a Framework for Geometric Computations*. Academic Press, Boston, 1991.
25. Wolf, Kathleen L. Digital Sketching. *Landscape Architecture*, 82(5): 75-77, May 1992.
26. Zeleznik, Robert C., Kenneth P. Herndon and John F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. In *SIGGRAPH '96 Conference Proceedings*, pages 163-170, August 1996.

APPENDIX

In this appendix we describe a method for determining the 2D mapping associated with a particular 3D camera motion. An equivalent mapping can be derived by eliminating a row and column from the 4 by 4 matrix describing the 3D camera's motion and projection directly. This technique, however, incurs a greater computation and representation overhead than the approach described here. Furthermore, this

method is comparable in terms of the intuition that it provides.

As stated earlier, our system uses projective 2D image points that lie on the surface of a unit sphere:

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ w \end{pmatrix}, \|\mathbf{p}\| = 1.$$

We desire to specify a particular projection of these projective points according to a specific class of view changes. All such mappings can be specified by a 3 by 3 matrix called a planar homography, \mathbf{H} [22].

$$\begin{pmatrix} w' x' \\ w' y' \\ w' \end{pmatrix} = \mathbf{H}\mathbf{p},$$

where the displayed point is (x', y') . Thus, we desire to specify the nine elements of \mathbf{H} according to our desired camera's view. Furthermore, since the projective points that vary by a positive scale factor are considered equivalent, so too will the homographies that vary by such scale factors.

The matrix \mathbf{H}^{-1} can be decomposed, via \mathbf{QR} decomposition, into a rotation matrix, \mathbf{R}_1 , and an upper triangular matrix, \mathbf{U}_1 : $\mathbf{H}^{-1} = \mathbf{R}_1\mathbf{U}_1$. From this decomposition, we can derive an alternate decomposition of \mathbf{H} into an upper triangular matrix, \mathbf{U}_2 , and a rotation matrix \mathbf{R}_2 .

$$\mathbf{H} = (\mathbf{H}^{-1})^{-1} = \mathbf{U}_1^{-1}\mathbf{R}_1^{-1} = \mathbf{U}_2\mathbf{R}_2$$

The operation of this homography can be easily understood in terms of these two matrices. The rotation matrix, \mathbf{R}_2 , is analogous to a three-dimensional rotation of the unit sphere, described in our conceptual model of the projective 2D space used in our representation. The upper triangular matrix, \mathbf{U}_2 , specifies the projection of those points onto a plane. The non-zero elements of this matrix specify the viewing frustum of the desired projection.

$$\mathbf{U} = \begin{pmatrix} f & \sigma & x'_0 \\ 0 & \rho f & y'_0 \\ 0 & 0 & 1 \end{pmatrix},$$

where σ is skew, ρ is the aspect ratio, and the remaining terms are illustrated in Figure 6. Typical projections will have a skew of 0 and an aspect ratio of 1. The resulting homography specification gives an intuitive method for specifying the re-projection of the 2D projective points used in our representation.

$$\mathbf{H} = \begin{pmatrix} f & 0 & x'_0 \\ 0 & f & y'_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}_{\theta\phi\psi}$$

This process can be reversed to map drawn coordinates to their corresponding projective representation.

$$\tilde{\mathbf{p}} = \mathbf{H}^{-1} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix},$$

and the point on the unit sphere is given as $\frac{\tilde{\mathbf{p}}}{\|\tilde{\mathbf{p}}\|}$.