

A Suggestive Interface for Image Guided 3D Sketching

Steve Tsang, Ravin Balakrishnan, Karan Singh, Abhishek Ranjan

Department of Computer Science

University of Toronto

www.dgp.toronto.edu

stsang | | ravin | karan | aranjean @dgp.toronto.edu

ABSTRACT

We present an image guided pen-based suggestive interface for sketching 3D wireframe models. Rather than starting from a blank canvas, existing 2D images of similar objects serve as a guide to the user. Image based filters enable attraction, smoothing, and resampling of input curves, and allows for their selective application using pinning and gluing techniques. New input strokes also invoke suggestions of relevant geometry that can be used, reducing the need to explicitly draw all parts of the new model. All suggestions appear in-place with the model being built, in the user's focal attention space. A curve matching algorithm seamlessly augments basic suggestions with more complex ones from a database populated with previously used geometry. The interface also incorporates gestural command input, and interaction techniques for camera controls that enable smooth transitions between orthographic and perspective views.

Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques – *Interaction Techniques*; H.5.2 [Information Interfaces and Presentation]: User Interfaces – *Interaction styles*.

General Terms: Design, Algorithms

Keywords: Image based interaction, sketching interfaces.

INTRODUCTION

Current digital 3D modeling software enable the creation and manipulation of extremely sophisticated geometric objects, and have been widely adopted by practitioners in industry. The user interface for these 3D modelers, however, are typically quite complex, relying on the standard WIMP interaction metaphor to perform all tasks, typically using a mouse and keyboard as the primary input devices. The skills designers, artists, and architects have in working with various rich physical media is simply not leveraged; instead, a whole new skillset has to be developed for working with the digital tools. Given that the fidelity of current digital tools often do not approach the physical

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2004, April 24–29, 2004, Vienna, Austria.

Copyright 2004 ACM 1-58113-702-8/04/0004...\$5.00.

ones, many traditional methods continue to be used in the industry. In particular, in the early stages of designing a 3D model, paper and pencil is typically used to quickly create and iterate on concept sketches. However, there is currently no mechanism for easily integrating these concept sketches into the digital 3D modeling pipeline. At present, these pencil and paper sketches are not used for anything more than an external visual reference when the designs are transformed to 3D models by artists who painstakingly build the model from low level graphical primitives.

Notable recent research [12, 23] has sought to improve this situation by creating prototype 3D modelers with a gesture and sketching interface that allow users to quickly and directly sketch their model, rather than working with low level primitives. A suggestive interface [9] can also assist the user by giving useful suggestions for subsequent actions based on the current context. While these efforts are a huge step forward, these interfaces still do not adequately leverage the still invaluable pencil and paper sketches.

This paper presents a new style of interface for 3D modeling that extends gestural and suggestive interfaces by integrating scanned images of concept sketches into the modeling process (Figure 1). These images serve as a guide to the user when sketching the 3D model. Image based filters can guide the user's input curves by attracting them towards curves in the image. User control is achieved via pinning and gluing techniques. The system also suggests relevant new geometry by approximately matching the user's input strokes against a database populated with standard and previously used geometry, resulting in a system that improves with use. Gestural commands, along with interaction techniques for camera control and switching between image planes, result in a fluid interface.

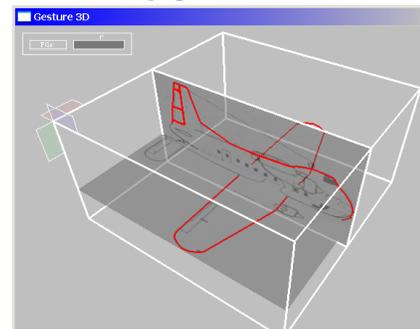


Figure 1. Screen shot showing geometry in-place with images

BACKGROUND

We draw on several related areas in our work, including gesture based interfaces, predictive and suggestive interfaces, recognition systems, image understanding, constraint based systems, and curve creation techniques.

Gestural interfaces depart from the WIMP metaphor by inferring actions from user input strokes, rather than relying on explicit manipulation of onscreen widgets. This interaction style has been explored in 2D pen-based applications [5, 14, 18]. The SKETCH system [23] was the first gesture-based interface for creating 3D scenes. The Teddy system [12] for sketching freeform 3D models improved upon the usability of SKETCH [23] by using a smaller but easily understood gesture set, at the expense of being able to create only relatively simple objects. To reduce the complexities inherent in large gesture sets, our system uses a small gesture set for command input, and a single technique for drawing curves and lines.

SKETCH [23] and Teddy [12] are similar in that users start from a blank canvas and explicitly create every single piece of geometry. Igarashi and Hughes [9] extended ideas from predictive interfaces [3, 17] by providing suggestions of geometry that could be used next. The suggestions in [9] were inferred based on hints provided by the user at the immediate past timestep. The Pegasus system [10, 11] interactively beautified input strokes by inferring desirable properties from surrounding context. Our system builds upon these ideas by using a dynamically updated database of possible candidates from which suggestions are extracted based on an approximate curve matching algorithm.

Recognition-based systems for handwriting, speech, and text input techniques also face the problem of resolving ambiguity amongst multiple candidates [15]. Multiple candidates are also generated when searching for good possibilities within large parameter spaces in other graphics applications [16]. Our interface also generates multiple candidate suggestions, and seamlessly presents the top few options spatially in-place with the existing geometry, rather than in a separate space as in [9, 11], reducing the need for users to context switch between different screen locales. Our system, like [9], presents the top candidates from a discrete space of possibilities, unlike [16] which presents representative examples of a large continuous space.

Constraint based systems impose limits on subsequent actions by inferring user's past operations [8]. Examples include drawing systems that impose graphical constraints [4, 13], and layout systems [8] that provide options based on user provided examples. While the inferred options in these cases are dynamic rules, our system, like [9], infers static options that are interactively utilized as needed.

We use curves as the primary drawing primitive in our system. Our curve drawing technique draws upon previous work in two handed curve drawing [1, 7], and interfaces for sketching 3D curves [2]. Curves are also important to image processing and computer vision [21], where feature curves

extracted from images are used for image matching, segmentation, and filtering. Active contour models or snakes [22] provide a framework whereby curves can evolve on images to track features based on energy minimization. Intelligent scissors [19] builds upon active contour models to interactively segment images. Saund et al. [20] explored using image processing techniques to interactively manipulate hand drawn sketches, but their focus was not on creating 3D geometry. Another difference from prior art is that our images are not a finished product to be tracked precisely but rather an evolving approximate guide towards a final model. We combine the energy minimization concept of snakes with interactively sketched constraints to allow images to selectively influence curves as an integrated part of our sketching workflow.

SYSTEM HARDWARE

The primary input device is a pen on a Wacom digitizing tablet. The pen has a barrel button with two click actions – forward, backward; and an active eraser end. On a 1GHz laptop with an NVidia GeForce2Go 16Mb graphics card, we achieve 30Hz update rates. Our software will also run on a TabletPC, however, current TabletPCs do not have sufficient graphics acceleration to achieve interactive rates.

INTERACTION TECHNIQUES

Spatially Integrated Construction Planes

3D models in our system are created by drawing the relevant 2D profile curves on orthographic construction planes from the three primary viewpoints – top, side, front. Rather than viewing these construction planes in separate 2D views as is typically done in 3D modeling software, in a manner similar to [6] our construction planes are spatially integrated into a 3D cuboid working volume, and can be moved within that volume to enable curve drawing at different locations (Figure 2a). This integration provides the user with information on the correspondence between the different viewpoints without switching visual context. In addition to 2D planar curves on the primary construction planes, the system also supports drawing 3D non-planar curves by projecting 2D curves onto previously created 3D surfaces in the scene (Figure 2b), similar to [7].

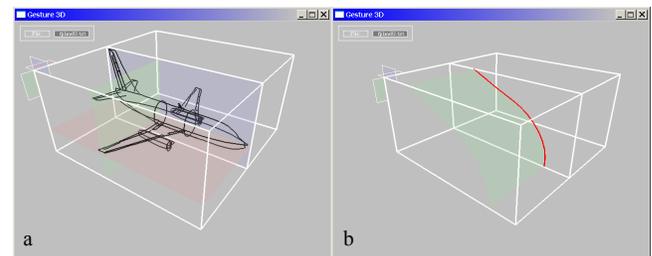


Figure 2. (a) Three primary construction planes displayed in pink, green and blue. Clicking on any of the planes, or the relevant tabs on the top left corner, makes it the active drawing plane. (b) Arbitrary surfaces can also be defined and used as construction planes.

Camera Controls

In perspective view, camera tumbling and zooming is performed by using the keyboard arrow keys, or by holding down a keyboard modifier key while moving the pen. Clicking the backward pen button with cursor on a construction plane transitions to an orthographic view of that plane, and vice versa. All transitions are animated to maintain continuity of context [6, 7].

Curve and Line Creation

We use a one-handed version of the two-handed tape drawing technique used in [6, 7] for creating 2D curves on the construction planes (Figure 3). This previous work used the non-dominant hand to change the length of the drawing vector, and consequently the variation of the resulting curve. We fix the drawing vector's length to allow for one-handed operation. By default, all input is treated as a curve, but pressing the stylus' forward button constrains drawing to straight lines. 3D curves are created using the techniques described in [7], and we refer the reader to that paper for the details. In curve drawing, all points of intersection with other curves in the scene are highlighted to enable users to align new curves to existing ones.

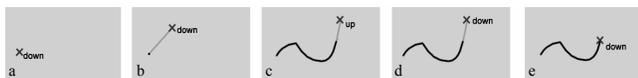


Figure 3. Curve drawing. (a) Pen down. (b) Pen moves to define the drawing vector. (c) Dragging drawing vector creates curve with smoothness defined by length of drawing vector. (d-e) Pen up ends curve at start point of drawing vector.

Gestures

By default, all stylus stroke input to the system is treated as curve and line drawings. However, when the user presses hard on the stylus tip (i.e., pressure exceeds a threshold of 800 out of 1024 units), it is assumed that gestural commands are being input. In order to maintain usability and keep ambiguity to a minimum, we use a very small set of three gestures: stroke, cut, and delete.

Stroke

This is a simple straight line gesture (Figure 4a), used for selecting/deselecting curves and construction planes. In perspective view, a stroke that intersects any curve in the scene makes that curve and its associated construction plane active. Once active, the curve can be edited, or moved by moving its plane. A stroke that intersects a suggested curve (to be described) confirms the inclusion of that curve into the scene. A stroke that intersects one of the edges of the bounding cuboid will generate a new active construction plane in the appropriate dimension depending on the intersected edge. A stroke that does not intersect any edge or curve will deselect the currently selected curve, if any. In an orthographic view (i.e., when drawing), strokes are only recognized when they intersect curves that lie on the currently active plane. This is to avoid inadvertent switching to another construction plane.

Cut

The cut gesture (Figure 4b) is a v-shaped two-segment gesture. When the cut gesture intersects a scene curve, the curve is cut into two segments at the intersection point. In perspective view, cut gestures can be applied to any scene curve, whereas in orthographic view cutting is limited to curves on the currently active construction plane. Cut gestures cannot be applied to suggested curves that have yet to become a permanent part of the scene.

Delete

The delete gesture (Figure 4c) is a N-shaped three-segment gesture. In perspective mode, a delete gesture that intersects any curve, including suggested curves, deletes that curve from the scene. In orthographic view, only curves in the active construction plane can be deleted.

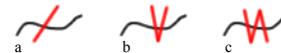


Figure 4. Gestures. (a) Stroke. (b) Cut. (c) Delete.

Image Guided Drawing

One of the primary goals of our system was to integrate scans of concept sketches into the 3D modeling workflow, and to use those sketches as guides for creation of new geometry. Currently, our system allows for images of sketches to be imported and placed on any of the three primary construction planes. Once loaded, these images can be used to influence the shape and position of drawn curves, using three interaction techniques: snapping, pinning, and gluing. At present, we do not yet support loading images onto arbitrarily shaped construction surfaces that can be defined by the user for non-planar 3D curve creation. Loading images onto an arbitrary surface is synonymous with texture mapping and thus a straightforward extension to our current implementation.

Snapping

When an image is loaded into a construction plane, image maps are created for various image attributes. Since our images are often concept line sketches with lines built from multiple strokes, we filter the images to remove noise in sketch stroke and intensity. We then precompute continuous intensity maps by letting strokes bleed intensity value into neighboring regions using reaction-diffusion (Figure 5).

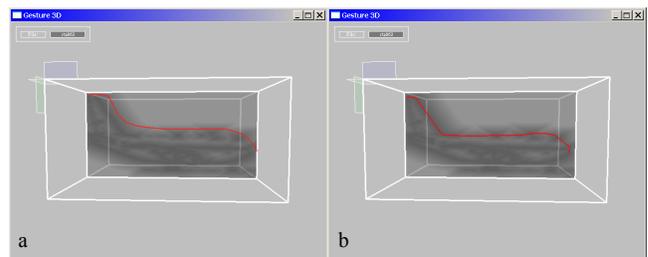


Figure 5. Intensity bleeding of line sketch. (a) Before snapping. (b) After snapping.

We then iteratively move and resample points on the sketched curves to minimize their overall energy. The active contour model [22] defines the energy of a snake as a combination of internal curve energy and external energy resulting from image guides and user constraints. The internal energy is a measure of bending and stretching of points on the curve, and can be reduced by Laplacian smoothing and arc-length minimization. The image energy at a point on the curve is a measure of various image attributes in a localized region of the image around the point. Attributes such as image intensity attracts curves to light or dark areas, image gradient attracts a curve to sharp edges. The individual energy terms in our system are under user control via gluing and pinning techniques. Roughly sketched curves can thus be quickly smoothed, resampled, and attracted to underlying image features (Figure 6).

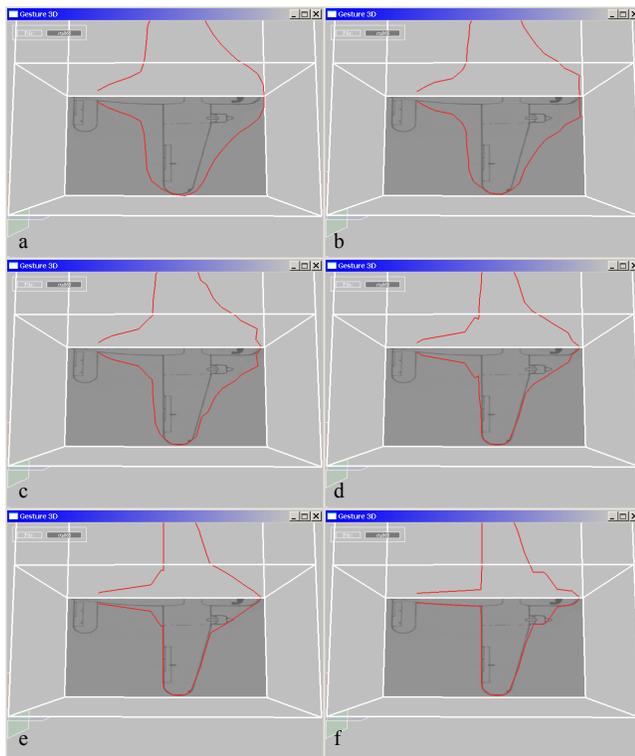


Figure 6. Snapping. (a) User draws curves roughly based on lines of underlying image. (b-f) Successive timesteps of snapping algorithm attracts the curves to the image lines.

Gluing

It is often desirable to constrain parts of a drawn curve from being affected by the image snapping algorithm. We support this via a “gluing” operation, which is performed by drawing along the curve using the eraser end of the stylus (i.e., flip the stylus). Parts of the curve that lie within the glued area remain invariant while the rest of the curve is attracted to the plane’s underlying image (Figure 7). Although we designed this technique to support image guided drawing, we have it to be useful when editing curves directly, without running the image snapping algorithm.

Pinning

We have found it useful to be able to specify a point on the image that the drawn curve should pass through, and designed a pinning technique to support this functionality. A drawn curve is first selected, and a pin point is specified by pressing the pen down on the construction plane. Dragging the tip away from the pin point adjusts the area of influence of the subsequent attractor algorithm. This area of influence is a circle centered at the pin point with radius equal to the distance from the cursor to the pin point. Parts of the curve that lie within this influence area behave like a rubber band being pulled to the pin point (Figure 8). Pinning can also be used independently of image snapping.

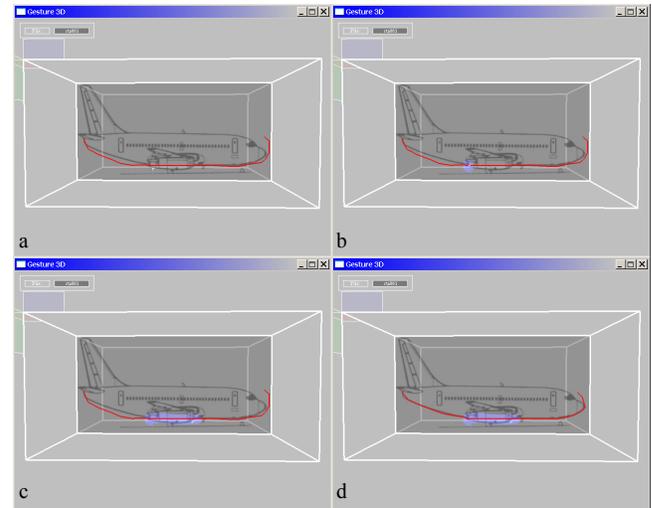


Figure 7. Gluing. (a) User drawn curve does not match underlying image exactly. (b-c) Blue swath indicates that part of the curve is glued. (d) Running the snapping algorithm attracts the unglued parts of the curve to the underlying image lines, while leaving the glued parts unchanged.

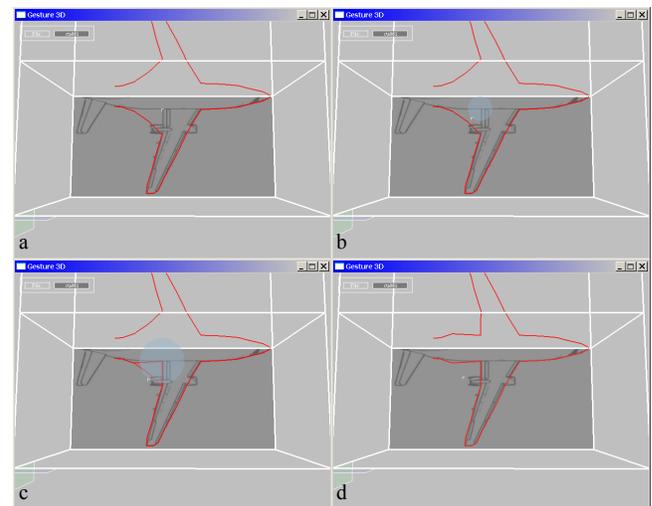


Figure 8. Pinning. (a) Curve is selected, and pin point located. (b) Dragging pen away from pin point increases area of influence. (c) Parts of curve that intersect the pinning area of influence behave like a rubber band attached to the pin point. (d) Lifting the pen confirms the pinning

Suggestions

When the user is drawing curves, our system provides ongoing suggestions of possible geometry that could be quickly used without the need to draw that geometry. The suggestions are inferred from characteristics of the currently drawn curve. Our work is motivated by Igarashi and Hughes [9], but differs in two ways. First, we present the top three candidate suggestions in the same viewing space as the model being created, unlike [9] where suggestions are displayed in a separate location on screen. This reduces the need for users to switch their focus of attention between viewing suggestions and drawing curves. Second, we provide two types of suggestions:

- algorithmically generated suggestions that infer simple subsequent geometry possibilities. For simplicity, we consider only two possibilities, closure and extrusion.
- database suggestions of previously used geometry extracted from a database by approximate matching to signature characteristics of user selected scene curves.

Closure Suggestions

When a curve has been drawn beyond a certain length and is not relatively straight, a curve is suggested that closes the newly drawn curve (Figure 9). Straightness is determined by iterating through the points of the newly drawn curve and summing the absolute angles between successive points. If the resulting sum is greater than a threshold (70°), then a closing curve is suggested. The suggested curve can be accepted into or deleted from the scene using the stroke and delete gestures respectively. Unused suggestions fade away after a 10 second time interval. This avoids burdening the user with explicitly dealing with unwanted suggestions as they proceed with their work. This accept/delete/ignore interaction is used for all suggestions in our system.

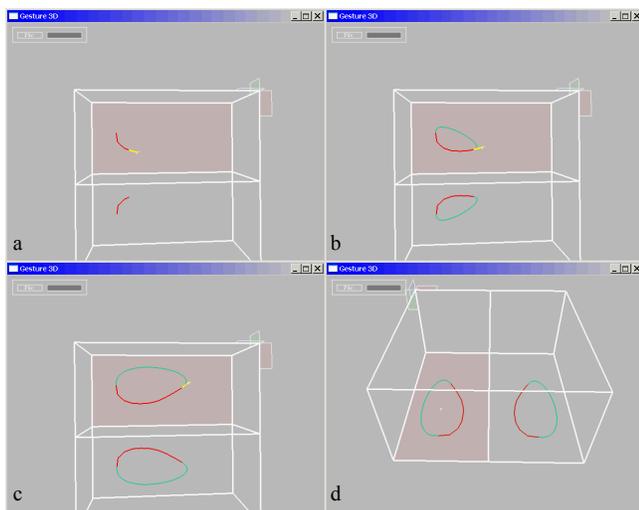


Figure 9. Closure suggestions. (a) A new curve is begun. **(b)** When length and curvature threshold is exceeded, a closed curve is suggested (shown in green). **(c)** User continues with drawing, suggestion accommodate changes. **(d)** User finishes drawing, and can select or ignore the suggested closed curve.

Extrude Suggestions

When a newly drawn curve intersects with another existing curve on a different axial construction plane, then an extrusion is suggested. The suggested extrusion curves represent the intersected curve translated to the two endpoints of the recently drawn curve. If necessary, the suggested curves are also cropped to keep them within the cuboid working volume.

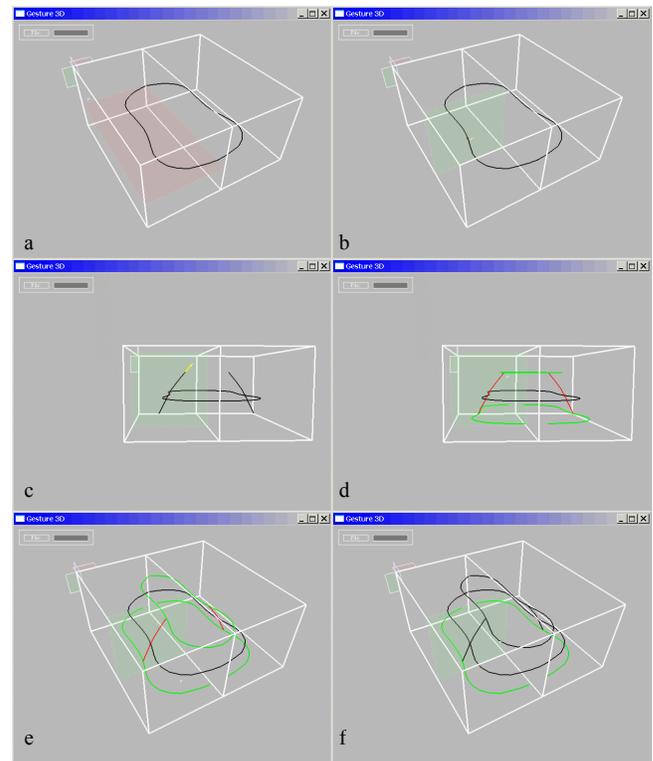


Figure 10. Extrude suggestions. (a) First set of curves created. **(b)** Perpendicular construction plane selected. **(c)** New curve drawn. Note mirroring of geometry. **(d)** Two extrusion curves are suggested (shown in green). **(e)** Perspective view highlights the extruded suggestions. **(f)** Top extruded suggestion selected.

Database Suggestions

Our system maintains a database of previously created geometry that has been explicitly saved by the user to provide suggestions for future drawings. Geometry in the database is processed to create a set of *curve-signatures*. Curve signatures are created by deconstructing the model curves into a network of tangent continuous curve segments. Half the fuselage curve in Figure 11a, for example, is segmented into 5 pieces with three relatively straight curves defining the wing. The lengths of these curve segments are normalized with respect to the overall length of the model curves and stored along with their mean curvature and relative orientation to each other. The *curve-signatures* thus provide a scale invariant representation of geometry in the database.

When new curves are drawn, their *curve-signature* is matched against those in the suggestion database. We first compute a correspondence between the drawn curve

segments and those in the database by graph matching. An overall match distance is then computed by the difference in length, orientation, and curvature of corresponding curve segments. Match distances less than a specified threshold are displayed as suggestions. Up to three top ranking matches are presented as suggestions to the user, within the working cuboid. In addition, additional lower-ranking matches can also be browsed using a widget that appears at the bottom of the screen. Should the user decide to increase the rank of one of the lower ranking suggestions, they simply click on that suggestion in the widget and it will appear within the working cuboid. Figure 11 illustrates.

These database suggestions effectively allow users to seamlessly leverage off previously created geometry, without having to explicitly browse or import them from the file system in a traditional manner.

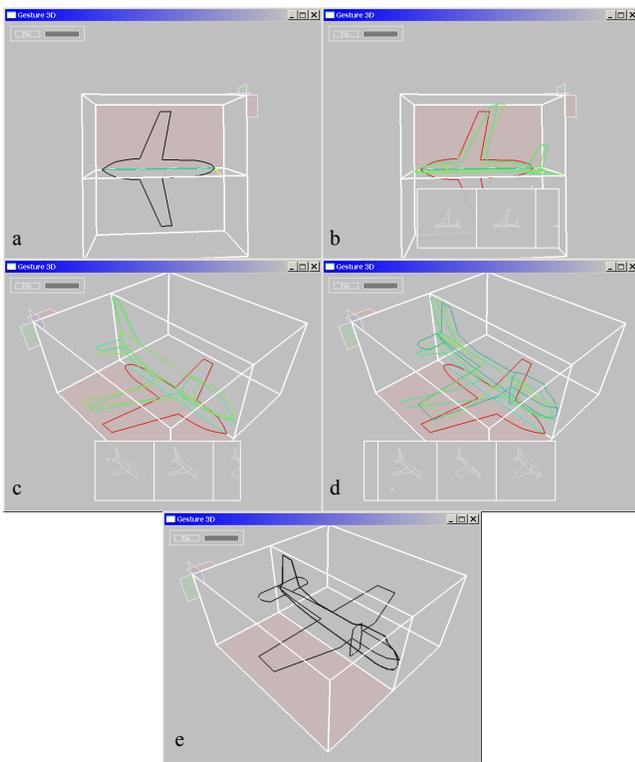


Figure 11. Database suggestions. (a) A set of curves is drawn as a template. (b) Database match results in two high ranking suggestions (shown in green) displayed within working cuboid. Widget at bottom of screen displays additional possibilities. (c) Perspective view. Note that the views in the widget change as well. (d) For illustration purposes, user selects a lower ranking alternative from the widget, and it is displayed in the cuboid. (e) Suggestion is accepted, and user drawn template deleted.

Curve Editing

Our system supports several different ways to edit curves that preserve the fluidity of the interface. To edit a curve, it must first be selected using a stroke gesture. Once a curve is selected (Figure 12a), touching the pen on the selected curve's construction plane without intersecting the curve itself invokes the pinning editing operation (Figure 12b-d).

Note that the pinning operation does not require that an image be previously loaded. A gluing operation can also be performed using the eraser end of the pen.

If the tip of the pen intersects the selected curve, a new curve or straight line can be drawn using the standard curve/line drawing technique. When the new curve has been drawn, it will be merged into the selected curve in one of two ways, depending on where the starting point of the new curve intersects the originally selected curve:

- If the starting point of the new curve is on one of the ends of the originally selected curve, it is assumed that the user wishes to extend the selected curve. The two curves are thus merged (Figure 12e-g). Note that the act of drawing the new curve could trigger suggestions as shown in Figure 12f.
- If the starting point of the new curve intersects some other part of the originally selected curve, it is assumed that the user wishes to edit a midsection of the curve, beginning at the intersection point (Figure 12h). The closest point on the original curve to the final endpoint of the new curve is determined, via the suggested curve (Figure 12i), and the new curve is thus smoothly incorporated into the original curve, displacing the unwanted segment.

Cut and delete gestures can also be used for editing, as shown in (Figure 12j-l).

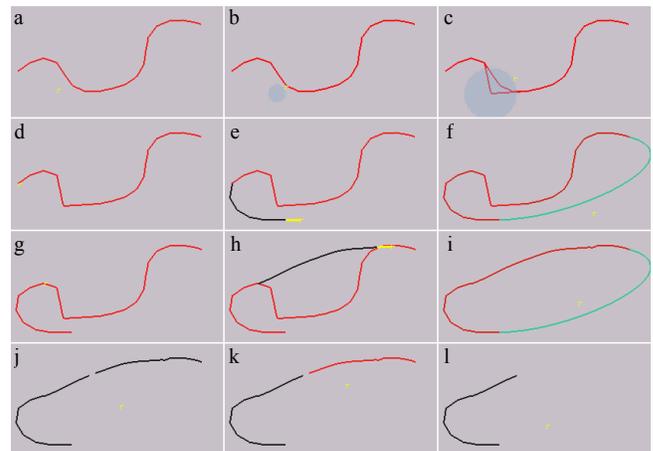


Figure 12. Curve editing. (a) Curve is selected. (b) Pin point specified outside the curve. (c) Pin influence region increased, curve rubberbands to pin point. (d) Pinning operation completed, rubberband deformation is accepted. (e) A new curve that starts at the endpoint of selected curve extends it. (f) New curve can trigger suggestions (shown in green). (g) Suggestion is ignored and new curve completed, resulting in extended original curve. (h) A new curve is drawn that intersects some middle point on selected curve. (i) New curve replaces unwanted segment of original curve. Suggestion appears (shown in green). (j) Suggestion is ignored and disappears, a cut gesture divides the curve into two sections. (k) One segment is selected (shown in red). (l) Delete gesture removes selected curve.

DISCUSSION

The motivation for this research came out of frustration expressed by designers at the inability to use their pen on paper concept sketches meaningfully in the construction of a 3D model. It was evident from looking at the evolving state of these sketches that the problem statement was to design an image guided sketching system rather than a system that attempted to accurately reproduce the feature curves of the input image. The second design requirement was simplicity of use that inspired our largely modelless workflow using gestures to invoke various actions. Finally, virtually all free-form industrial design must obey certain geometric and style constraints and involves the reuse of various geometric constructs. Rather than hard-code an increasing set of design constraints and primitive shapes, we introduced the notion of a user defined suggestion database. These geometric databases, comprising models built to specifications of style and design constraints, are dynamically searched for potential matches during the sketching process. This would allow a user to quickly incorporate previous work either as a starting point for a new model, or to reuse geometry that had been previously certified in parts of a new model.

While our system is still in the research prototype stage, we thought that it was important to get some early feedback from potential users. An equal amount of time was spent by two users on both the breadth of the interface and on building models resulting in the following feedback:

- The use of a 3D cuboid working volume as the virtual sketchpad – within which the image planes, interactive sketching, and suggestions are visually integrated – is quite effective. This enables the user to maintain an overall spatial awareness of the task at hand
- The fixed drawing vector length simplifies the curve drawing paradigm but tends to create curves of limited maximum curvature. In practice this is not a problem since high curvature regions can be drawn as two tangent discontinuous segments and then smoothed locally by Laplacian smoothing.
- Pinning and gluing techniques, originally designed for use with image planes were found to be a useful general way of precisely constraining curves.
- We found that keeping the gesture set small made it robust and easy to understand and remember.
- In general, precise control over pressure while drawing on a tablet was hard to master. For this reason we only used one pressure threshold to distinguish gestures from curve drawing. Different users naturally apply different amounts of average pressure to the pen and the threshold that distinguishes drawing from gestures should be calibrated for each user independently.
- While the system in its present state is tailored to early conceptual design that did not require high precision, users

were able to establish precision when needed by incorporating suggestions from the database.

- One user mentioned that *curve-signatures* would be a great way to search and index into arbitrary visual databases. Indeed in our approach, when saving a model to the suggestion database, the images and other related information can be linked with the model and recalled by sketching its *curve-signature*.

- At present, suggestion database matches are shown in place as an affine transformation of their original construction. It was mentioned that it would be useful to indicate as additional suggestions, versions of the geometry that was left unchanged as well as one that was locally deformed to better conform to the sketched curve.

To test the versatility of our system we used it to construct 2D line sketches of various facial expressions from images. The system was then used to index into this database and distinguish various facial expressions with just a few sketch strokes (Figure 13). For our current database sizes of around 10 models the curve matching algorithm was able to perform robustly at interactive rates. A study into the scalability of suggestion databases is subject to future work as well as exhaustive user testing of our system.

A direction for future work would be to extend our system to construct 3D curves while still working with multiple 2D image planes. We hope to do this by compositing the many 2D image maps, such as that seen in Figure 5 to a single 3D volume map that can attract the curves in 3D instead of on a 2D manifold.

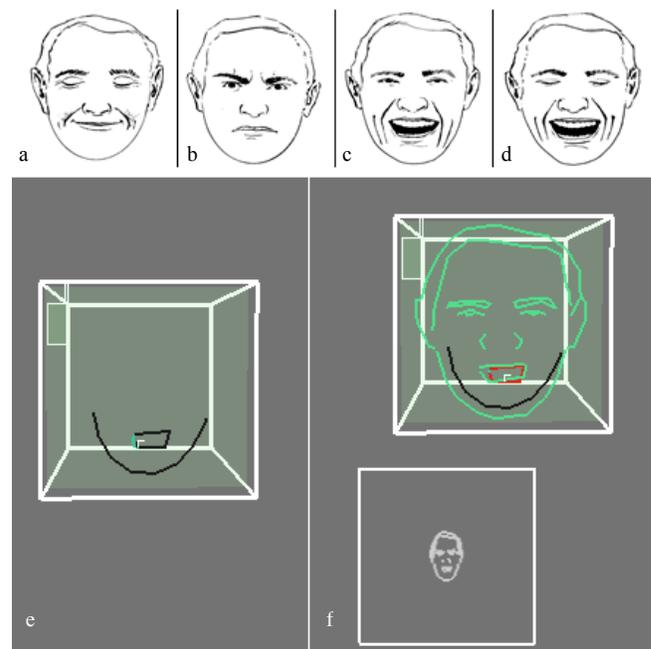


Figure 13. (a-d) Example face geometry in suggestion database. (e) Sketch of initial face. (f) Suggested face

CONCLUSION

Overall, our system has demonstrated how a simple sketching interface when augmented appropriately with gestures, concept images, and suggestions can become a compelling and effective design tool. The airplane model in Figure 2a was built in under 5 minutes using the system described in this paper by one of the authors of this paper.

ACKNOWLEDGEMENTS

We thank members of the Dynamic Graphics Project lab at the University of Toronto for assistance throughout this project, and MITACS for financial support.

REFERENCES

- Balakrishnan, R., Fitzmaurice, G., Kurtenbach, G., & Buxton, W. (1999). Digital tape drawing. *ACM UIST 1999 Symposium on User Interface Software and Technology*. p. 161-169.
- Cohen, J., Markosian, L., Zeleznik, R., Hughes, J., & Barzel, R. (1999). An interface for sketching 3D curves. *ACM I3DG 1999 Symposium on Interactive 3D Graphics*. p. 17-21.
- Cypher, A. (1991). Eager: programming repetitive tasks by example. *ACM CHI Conference on Human Factors in Computing Systems*. p. 33-39.
- Gleicher, M., & Witkin, A. (1994). Drawing with constraints. *The Visual Computer*. 11(1). p. 39-51.
- Gross, M.D., & Do, E.Y.-L. (1996). Ambiguous intentions: a paper-like interface for creative design. *ACM UIST Symposium on User Interface Software and Technology*. p. 183-192.
- Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., & Buxton, B. (2001). Interaction techniques for 3D modeling on large displays. *ACM I3DG 1999 Symposium on Interactive 3D Graphics*. p. 17-23.
- Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., & Buxton, B. (2002). Creating principal 3D curves with digital tape drawing. *ACM CHI 2002 Conference on Human Factors in Computing Systems*. p. 121-128.
- Hudson, S., & Hsi, C. (1993). A synergistic approach to specifying simple number independent layouts by example. *ACM CHI Conference on Human Factors in Computing Systems*. p. 285-292.
- Igarashi, T., & Hughes, J. (2001). A suggestive interface for 3D drawing. *ACM UIST Symposium on User Interface Software and Technology*. p. 173-181.
- Igarashi, T., Matsuoka, S., Kawachiya, S., & Tanaka, H. (1997). Interactive beautification: a technique for rapid geometric design. *ACM UIST Symposium on User Interface Software and Technology*. p. 105-114.
- Igarashi, T., Matsuoka, S., Kawachiya, S., & Tanaka, H. (1998). Pegasus: A drawing system for rapid geometric design. *Extended Abstracts of ACM CHI Conference on Human Factors in Computing Systems*. p. 24-25.
- Igarashi, T., Matsuoka, S., & Tanaka, H. (1999). Teddy: a sketching interface for 3D freeform design. *ACM SIGGRAPH*. p. 409-416.
- Karsenty, S., Landay, J., & Weikart, C. (1992). Inferring graphical constraints with Rockit. *HCI Conference*. p. 137-153.
- Landay, J., & Myers, B. (1995). Interactive sketching for the early stages of design. *ACM CHI Conference on Human Factors in Computing Systems*. p. 43-50.
- Mankoff, J., Hudson, S., & Abowd, G. (2000). Interaction techniques for ambiguity resolution in recognition based interfaces. *ACM UIST Symposium on User Interface Software and Technology*. p. 11-20.
- Marks, J., Andalman, B., Beardsley, P., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., & Shieber, S. (1997). Design galleries: A general approach to setting parameters for computer graphics and animation. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. p. 389-400.
- Maulsby, D., Witten, I., & Kittlitz, K. (1989). Specifying graphical procedures by example. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. p. 127-136.
- Moran, T.P., Chiu, P., Melle, W.v., & Kurtenbach, G. (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. *ACM UIST Symposium on User Interface Software and Technology*. p. 127-136.
- Mortensen, E., & Barrett, W. (1995). Intelligent scissors for image composition. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*.
- Saund, E., Fleet, D., Lerner, D., & Mahoney, J. (2003). Perceptually-supported image editing of text and graphics. *ACM UIST Symposium on User Interface Software and Technology*. p. 183-192.
- Sonka, M., Hlavac, V., & Boyle, R. (1999). Image processing, analysis and machine vision. PWS Publishing. .
- Witkin, A., Kass, M., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*. 1(4). p. 321-331.
- Zeleznik, R.C., Herndon, K., & Hughes, J. (1996). SKETCH: An interface for sketching 3D scenes. *ACM SIGGRAPH 1996 Conference on Computer Graphics and Interactive Techniques*. p. 163-170.