Sketching Over Props: Understanding and Interpreting 3D Sketch Input Relative to Rapid Prototype Props

Bret Jackson
University of Minnesota
Department of Computer Science and Engineering
bjackson@cs.umn.edu

Daniel F. Keefe
University of Minnesota
Department of Computer Science and Engineering
keefe@cs.umn.edu

ABSTRACT
We present a discussion of our recent work to understand and interpret 3D sketch input made relative to rapid prototype props. 3D printing technology has now matured to the point where it is readily available for use in creating rapid prototypes of scientific and other datasets to support physical visualization of complex 3D geometries. We believe that the utility of these physical printouts can be dramatically increased if we can better understand how to use them as interactive tools rather than simply static physical displays. To this end, we have been exploring the potential of combining 3D sketch-based interfaces with physical rapid prototypes for accomplishing tasks such as linking the physical printout with complementary stereoscopic visualizations of data *inside* the bounding surface of the 3D geometry. This research trajectory raises several interesting discussion points related to understanding how best to bring sketch recognition to this new 3D application. In this paper, we describe the research context and initial insights that we have obtained through a formative design critique of our current sketching interface. We conclude by identifying four specific research challenges that we believe are critical for better understanding how sketch-based input can be used to turn rapid prototype props into highly-interactive visualization tools.

Author Keywords
Gesture recognition, pen-based interfaces, sketch-based interfaces, tangible user interfaces, rapid prototypes

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces—*Input devices and strategies, Interaction styles*

INTRODUCTION
Recent technological advances allow scientists to collect large and complex 3D datasets that exceed the capacity for easy analysis. To facilitate analysis of these spatially complex data, we have been exploring the potential of combining

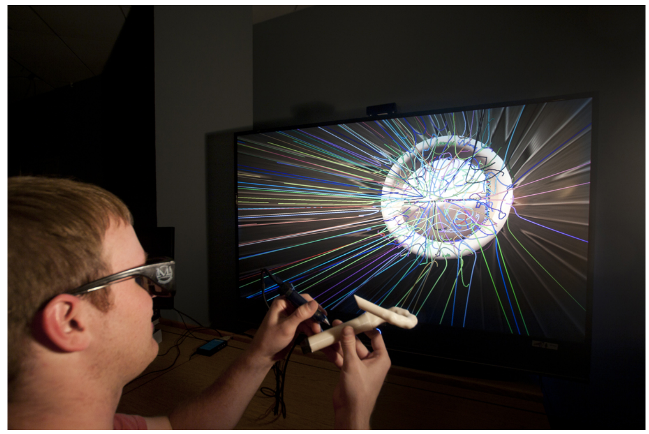

Figure 1. Our recent work has explored combining the immediate spatial understanding provided by 3D rapid prototype props (physical printouts) of 3D data with complementary VR visualizations, using 3D sketches made relative to the 3D prop to connect the physical display with the virtual. We are interested in how sketch recognition and sketch-based interfaces change in this type of physical 3D context.

physical models of scientific datasets, fabricated with rapid prototyping machines, with 3D sketch-based input to support visualization tasks. Our goal is to extend work in tangible interfaces to support not just viewing the outside of the model, but also to support a new style of intuitive navigation and visualization of internal data within the 3D space bounded by a tangible prop. To this end, our recent work [12] has included creating the sketch-based 3D interface shown in Figure 1, which combines a tracked 3D printout of a cardiovascular flow dataset generated by our collaborators studying computational fluid dynamics [22] with a hand-held pen device and a virtual reality (VR) data visualization.

Informed by this recent work, the focus of this paper is on better understanding how sketch recognition changes in non-traditional interfaces, which involve 3D sketches and gestures made relative to a 3D physical prop. We are motivated in this investigation by the intuition we have gained from our experience and by the work of other groups. In short, we have found that sketching directly on top of a rapid prototype (with the pen in contact with the prop) is very difficult to control because of the realistic organic shape variations that today's 3D printers are able to capture. On the other hand, completely freehand drawings made in the air are typically difficult to control. Informally, we see some evidence that a middle ground may exist in sketching just above (e.g. a centimeter above) a 3D prop, as it provides some context to anchor the sketching motions, but since the pen is free to move in the air, the problem of sketching on top of a bumpy surface is avoided. In the right context, others have also found that 3D sketching shows great promise [20] and can even be very controllable if the right approach is taken in designing the user interface [10, 11]. Thus, we are interested in better understanding how sketching on and near the context provided by a physical rapid prototype printout impacts the accuracy of 3D sketches and how such sketches might be incorporated into new 3D user interfaces. Our primary contribution in this paper is a discussion of these and other issues tied to sketching in relation to 3D physical props. Specifically, we identify four research challenges that need to be addressed in order to achieve accurate sketch recognition and fluid interfaces within this context. Our discussion is anchored by a small design critique of our current interface.

As background for understanding the discussion and critique presented in the remainder of the paper, we briefly describe the interface pictured in Figure 1 (presented in more detail in [12]), which serves as an example application for sketching over props. This visualization system consists of a desktop-scale "fishtank" VR environment that includes a head-tracked stereoscopic display, which we have coupled with a 3D rapid prototype of a scientific dataset. In this application, the virtual reality visualization depicts data from a cutting-edge high-performance simulation of blood flow through a 3D aorta model derived from medical imaging data [22]. Thus, the prop employed in the sketch-based interface is generated from the bounding surface of the 3D aorta model. (We use the same prop for the design studies reported here.) Both the prop and the pen are tracked with a 6-DOF Polhemus Fastrak magnetic tracking system that allows us to update the display as the user, prop, and pen move through space. To maintain a consistent frame of reference, the interface enforces useful constraints, for example, the virtual model is always constrained to rotate in such a way that its orientation relative the user is consistent with the orientation of the physical prop relative to the user. The advantage of the interface comes as the user begins exploring the data, using sketch-based gestures as well as "geometric gestures" (e.g. holding the pen as a pointing or slicing device) to adjust parameters of the visualization. To date, we have implemented a small gesture set that includes support for navigation and bookmarking. To keep the user's focus on the physical prop and facilitate gesture recognition, we begin each gesture with a tap of the pen on the prop. Our initial implementation uses a very simple recognition strategy. To move beyond this and support more sophisticated operations, we believe that new approaches are needed. Hence, our interest in improving 3D sketch recognition in this context.

The remainder of the paper begins with a discussion of related work. Then, we present details and results of the design critique of the current tool. Finally, we end with a discussion of insights that came to light during the critique and we formalize these findings as four important future research challenges.

RELATED WORK

3D Gesture Recognition

Much 3D gesture recognition research has focused on recognizing hand gestures for interacting with virtual environments [16, 18, 25]. These systems use machine learning and statistical techniques for recognition. More recently, the creation of low cost spatial input devices such as the Nintendo Wii Remote (Wiimote) has stimulated research in 3D gesture recognition for gameplay. These systems take advantage of the 3D acceleration data that the Wiimote provides, and recent results show that they can be accurate [7]. However, both hand gestures and Wiimote gestures tend towards larger full body motions such as a golf swing or hand wave.

In our context, when sketching over a small hand-held prop, the gestures require more precise fine motor control. As a first step toward recognizing these gestures, we have adapted traditional 2D sketch-recognition techniques for use in 3D by following the strategy recently employed for VR games [9]: the trajectory of the stylus is first projected onto the best fit plane of the stroke sample points, and then a lightweight 2D recognizer [24] is used to identify the gesture.

Gestures Without Visual Feedback

Because we aim to sketch gestures over a physical prop, barring a more sophisticated augmented reality display, we expect that users will be performing these gestures without seeing some visual feedback of the path of the stroke as it is drawn. Previous work by Ni and Baudisch [19] explored the use of 3D hand gestures over a scanning interface on the user's wrist. This interface, which does not provide visual feedback, highlighted the spatial challenges of connecting strokes in complex Graffiti characters. For instance, D glyphs were often mis-recognized as P's, because the user was unable to properly close the gesture.

Additional studies have tried to evaluate the extent to which our visuospatial memory can capture gesture paths for interacting with imaginary interfaces in 3D [5]. Their results show that we are able to replace visual feedback with memory for gestures, such as Graffiti characters, which contain a small number of strokes, and in contrast with Ni and Baudisch users do not seem to have difficulty closing shapes for single stroke gestures. Given the results currently available in the literature, we believe more work is needed to better characterize user performance when sketching without visual feedback, especially in new contexts, such as 3D sketching and sketching relative to 3D props.

Combining Pen-based Gestures with a 3D Prop

Interfaces that combine pens with tablets have been studied extensively (e.g. [21]), including in virtual environments [2, 6]. However, sketching on or near a complex organic form, such as a 3D printout from volumetric medical data, presents new challenges. We are interested in understanding how sketch recognition is affected in this context and how it can be extended to provide richer input for navigating complex internal datasets.

Perhaps most closely related to our sketch-based prop interaction technique is work by Song et al. [23]. Their ModelCraft interface allows freehand annotations on physical 3D paper models. However, their approach differs in several ways from ours. For tracking of gestures, they print a dot pattern on paper that is then folded up to form the 3D prop shape. A camera on their pen uses the pattern to register itself with the prop. This limits the gesture recognition to the surface of the prop. Additionally, because their props are created from folded paper, they are limited to basic shapes. Their system would be unable to handle the complex organic forms, such as the ones found in biomedical datasets.

Another closely related work is that by Kruszyski et al. [14], in which a smartly calibrated pen interface was used to interact with high-resolution 3D rapid prototypes of corals. This work demonstrated the feasibility and potential impact on scientific workflows of designing pen-based interfaces for interacting with rapid prototypes. We believe much more is possible when pens are used not only as a pointing device, but also as a sketching device.

SKETCHING OVER A PROP

To evaluate the requirements of sketch recognition over a physical prop, we present a small formative design critique of the sketch interaction techniques used in our current interface. The participants were two male doctoral students working in the area of computer graphics and interactive visualization and one female graduate student in architecture. Sitting at a desk in front of a stereoscopic TV, each participant was asked to sketch four different gestures (a circle, triangle, star, and left bracket) five times. This was done first using a tracked pencil on a sketchbook, then in the air, and finally over a prop. For each sketching context, the task was repeated twice, once without visual feedback (no pencil lead and a blank screen), and once with visual feedback shown on the vertical display. The feedback presented was a stereoscopic visualization of the prop and stylus with a real-time trace of the gesture path. The participants were allowed to practice before the critique until they felt comfortable with the interface.

Calibration of the Polhemus Fastrak magnetic trackers attached to the pen and prop was completed in a two step process. The first step calculates the offset from the pen tracker to the pen tip using a least-squares optimization [8, 17]. The second step calibrates the the prop by manually adjusting the attachment position of the tracker on the prop until the virtual and physical models closely match. Although it was not implemented at the time of our design critique, we have subsequently implemented an interative closest points algorithm [4] to match the virtual model to sample points drawn on the surface of the physical prop, which further refines the calibration.

Discussion

One of the biggest difficulties with any 3D gesture system is determining gesture segmentation, i.e. given a stream of continuous tracker data, how are the start and end points of the gestures determined. For one-stroke gestures on 2D tablet input systems, this is easily solved by assuming that a gesture is being drawn whenever the pen is touching the input surface. However, on 3D printed rapid prototypes the surfaces are frequently not smooth enough to maintain contact while drawing an accurate gesture.

Recent work has addressed 3D gesture segmentation, also called gesture spotting [15], using supervised learning techniques such as Hidden Markov Models [3] or Adaptive Boosting [13] to determine thresholds in motion parameters such as velocity, acceleration or curvature. Other methods try to identify temporal segmentation of gestures by finding intervals in the input data that give good recognition scores [1].

Because of the complexity in implementing these segmentation algorithms, we have begun by using a simple approach to determine segmentation. We start recording gesture sample points when the user touches the tip of the pen to the 3D prop. (Aside from the benefit of simplifying gesture segmentation, we think this is a good design decision because it repeatedly directs the user's focus to the 3D prop and the context that it provides.) We then record tracking samples for a full second after the initial tap and parse these samples to recognize a gesture. Our critique showed us that, despite our initial successes with this simple scheme, there are wide variations in how people draw and what feels comfortable, especially when the drawing is done in 3D. All three participants had trouble adjusting their drawing speed to the complexity of the gesture, as more complex gestures must be drawn faster to fit within the one second time-frame. For instance, all expressed frustration at not being able to complete the star gesture. Subsequently, we have implemented a technique that identifies the end of a gesture based on a pause in the motion of the user. (Users are asked to pause briefly after each gesture.) This also simplifies gesture segmentation, but is less constraining for users. Starting from the last sample in the gesture identified via this strategy, the algorithm iteratively increases the number of samples to consider for the gesture, working backwards toward the initial tap, and returns the gesture with the highest recognition score. This strategy produces workable results for initial studies, but additional refinement will be necessary to support more sophisticated applications.

We also discovered that there was large variation in how close to the prop each participant drew after the initial tap to begin each gesture. If the user pulls too far away from the prop before drawing the gesture, the best-fit plane that the gesture is projected onto can be aligned more with the path moving way from the prop than the actual gesture plane, decreasing the accuracy of the recognition engine. This is seen in Figure 2. The technique described above (iteratively increasing the segment length by working backwards toward the initial tap point) addresses this problem.

We also noticed that when drawing in the air all the users placed their elbow on the table to provide support. One reviewer commended that "Drawing on air is not very intuitive. I am used to having my arm rest on something." We also noticed at least one reviewer resting several fingers of

Figure 2. Proper gesture segmentation is required when determining the best-fit plane to project gesture points onto for recognition. The image on the right shows how the user has pulled the pen way from the prop surface after indicating the start of a gesture, causing the best fit plane to be almost perpendicular to the plane that contains the "real" gesture. The image on the left shows the sketch input projected onto the best fit plane and scaled to fit within a square.

the hand holding the pen on the prop while drawing gestures relative to the prop. This stabilization seemed to help their gesture accuracy.

The upper half of Figure 3 demonstrates that for sketching in air, displaying visual feedback on the screen did not help users close and align their gesture segments. This also seemed to be the case when sketching relative to a prop (lower half of figure). We noticed that two of our reviewers continued to look at the prop when drawing even when visual feedback was displayed on the screen in front of them. Perhaps a system that projects visual feedback onto the prop would work better and allow for more accurate gestures, albeit at the cost of a more sophisticated display system.

Perhaps the most interesting result of our critique was that the gestures our reviewers made in the context of the prop were smaller than those that they drew either in the air or on a flat surface. Most reviewers either only drew inside the silhouette of the prop or started that way and then gradually started drawing bigger. This led to more noise in the best-fit plane optimization and the gesture path itself (note the elongation of circular prop gestures in Figure 3), lowering the recognition accuracy significantly. One reviewer commented, "I felt constrained by the width of the prop." This indicates that the size of the prop is particularly important for intuitive sketching. Perhaps scaling the prop to a larger size would allow people to immediately feel like they could draw larger and more accurate gestures.

IMPLICATIONS FOR DESIGN AND FUTURE WORK

The results of our design critique suggest several research challenges that must be addressed when doing sketch recognition over a prop:

- To date, most 3D gesture recognition uses hands or input devices to make large motion gestures, such as a golf swing or hand wave. When sketching over a prop, users tend to limit their sketching motions to the size of the prop. More work needs to understand the extent to which large-scale 3D gesture recognition algorithms and other strategies can be employed for use with smaller fine-motor control gestures made using 3D pen input.

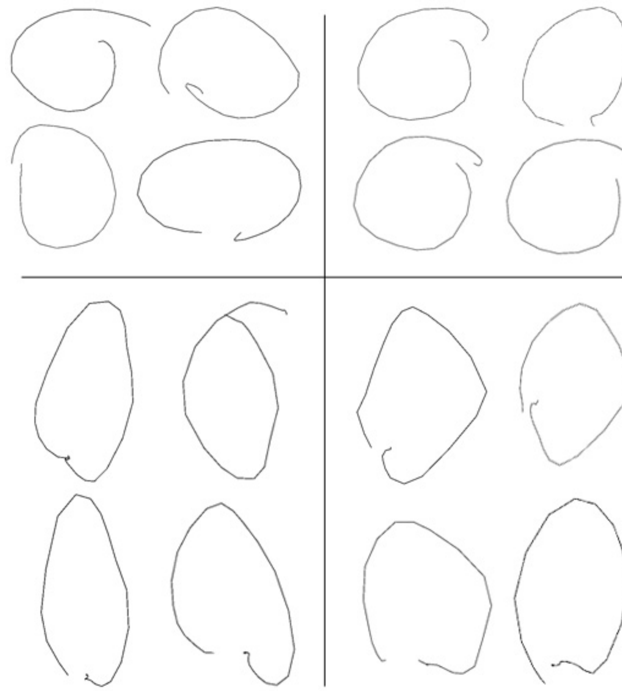

Figure 3. Examples of circle gestures (all scaled to a consistent size). In the sketching over prop conditions (bottom row), the elongated shapes demonstrate how participants naturally tried to stay within the prop silhouette as they drew. Upper Left: Drawing in the air with visual feedback on the screen. Upper Right: In the air without visual feedback. Lower Left: Sketching over a prop with visual feedback on the screen. Lower Right: Over a prop without visual feedback.

- Tracking technology and calibration are vitally important to providing accurate visual feedback for the gesture path. One reviewer reported that our magnetic trackers seemed to have a little lag when updating the display which made it difficult to accurately draw a gesture where segments needed to meet at specific points. More work needs to be done to see whether optical or inertial tracking would provide more accuracy and intuitive input.
- More work is needed to understand the perceptual and cognitive implications of holding a prop while sketching, and user interface design guidelines based on perceptual and cognitive principles need to be developed.
- Holding a physical prop and sketching above it gives us a surface onto which we can project more information, potentially including visual feedback for sketched input. This aspect of sketching above props needs to be explored in more detail, and the cost-benefit tradeoffs implied by a more sophisticated display system need to be explored.

Further study in these areas will extend the possibilities of combining 3D printed rapid prototypes with an intuitive and rich sketch-based interface to support navigation and visualization of complex datasets. We see a clear need to improve 3D gesture recognition and segmentation algorithms to support this type of 3D sketch-based interface.

CONCLUSION

We believe that sketch-based interaction on or near a prop is an exciting area of research that can push the boundaries of current work in sketch recognition, ultimately leading to new algorithms and interfaces that may feed back into more mainstream applications of sketch recognition. In this paper, we have attempted to formalize our thinking about sketch recognition in this context so as to present it to the community as an important challenge area. As such, our main contribution is clearly identifying several research challenges in this area. We believe that continued discussion of challenges related to sketching over props will be particularly valuable.

ACKNOWLEDGEMENTS

Vamsi Konchada was instrumental in developing the original interface that enabled the design study reported in this paper. Our motivation and the specific prop geometry used in this study come from a scientific visualization of a cardiovascular simulation that we have developed together with our collaborators in computational fluid dynamics: Fotis Sotiropoulos, Iman Borazjani, Trung Le, and colleagues at the St. Anthony Falls Laboratory. We thank Will Durfee for the support of his laboratory and staff in printing the prop. We also acknowledge Art Erdman, H. Birali Runesha and their staffs at the Medical Devices Center and Minnesota Supercomputing Institute for continued discussion and support in this research.

REFERENCES

1. J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1685–1699, 2009.
2. D. A. Bowman. Interaction techniques for immersive virtual environments: Design, evaluation, and application. *Journal of Visual Languages and Computing*, 10:37–53, 1998.
3. J. W. Deng and H. T. Tsui. An hmm-based approach for gesture segmentation and recognition. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, page 3683, 2000.
4. D. Eggert, A. Lorusso, and R. Fisher. Estimating 3D rigid body transformations: A comparison of four major algorithms. *Machine Vision and Applications*, 9:272–290, 1997.
5. S. Gustafson, D. Bierwirth, and P. Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, pages 3–12, 2010.
6. M. Hachet and P. Guitton. The interaction table: A new input device designed for interaction in immersive large display environments. In *Proceedings of the Workshop on Virtual Environments 2002*, pages 189–196, 2002.
7. M. Hoffman, P. Varcholik, and J. LaViola. Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices. *IEEE Virtual Reality*, pages 59 –66, 2010.
8. M. Ikits, C. D. Hansen, and C. R. Johnson. A comprehensive calibration and registration procedure for the visual haptic workbench. In *Proceedings of the Workshop on Virtual environments 2003*, pages 247–254, 2003.
9. M. Katzourin, D. Ignatoff, L. Quirk, J. LaViola, and O. Jenkins. Swordplay: Innovating game development through VR. *IEEE Computer Graphics and Applications*, 26(6):15 –19, 2006.
10. D. Keefe, R. Zeleznik, and D. Laidlaw. Tech-note: Dynamic dragging for input of 3D trajectories. In *Proceedings of IEEE Symposium on 3D User Interfaces*, pages 51–54, 2008.
11. D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Drawing on air: Input techniques for controlled 3D line illustration. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1067–1081, 2007.
12. V. Konchada, B. Jackson, T. Le, I. Borazjani, F. Sotiropoulos, and D. F. Keefe. Supporting internal visualization of biomedical datasets via 3d rapid prototypes and sketch-based gestures. In *Poster Proceedings of ACM Symposium on Interactive 3D Graphics and Games*, 2011.
13. N. Krishnan, P. Lade, and S. Panchanathan. Activity gesture spotting using a threshold model based on adaptive boosting. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 155 –160, 2010.
14. K. Kruszyski and R. van Liere. Tangible props for scientific visualization: Concept, requirements, application. *IEEE Virtual Reality*, 13:235–244, 2009.
15. H.-K. Lee and J. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 21(10):961 –973, 1999.
16. M. Moni and A. Ali. Hmm based hand gesture recognition: A review on techniques and approaches. In *Proceedings of IEEE International Conference on Computer Science and Information Technology*, pages 433–437, 2009.
17. S. Mooslechner. Stylus calibration and prop registration. *Bachelor Thesis: Institute for Computer Graphics and Vision, Graz University of Technology, Austria*, 2008.
18. G. Murthy and R. Jadon. Hand gesture recognition using neural networks. In *Proceedings of IEEE 2nd Annual International Advanced Computing Conference*, pages 134 –138, 2010.

19. T. Ni and P. Baudisch. Disappearing mobile devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, pages 101–110, 2009.

20. J. M. Peschel, B. Paulson, and T. Hammond. A surfaceless pen-based interface. In *Proceeding of the Seventh ACM Conference on Creativity and Cognition*, pages 433–434, 2009.

21. E. Sachs, A. Roberts, and D. Stoops. 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, 11(6):18 –26, 1991.

22. H. Simon, L. Ge, I. Borazjani, F. Sotiropoulos, and A. Yoganathan. Simulation of the three-dimensional hinge flow fields of a bileaflet mechanical heart valve under aortic conditions. *Annals of Biomedical Engineering*, 38:841–853, 2010.

23. H. Song, F. Guimbretière, C. Hu, and H. Lipson. Modelcraft: Capturing freehand annotations and edits on physical 3d models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 13–22, 2006.

24. J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, pages 159–168, 2007.

25. X. Zhang, X. Chen, W.-h. Wang, J.-h. Yang, V. Lantz, and K.-q. Wang. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 401–406, 2009.