

# A Gesture-Based Interface for the Exploration and Classification of Protein Binding Cavities

Zachary A. Daniels  
Computer Science and  
Engineering  
Lehigh University  
27 Memorial Dr. West  
Bethlehem, Pennsylvania  
18015  
zad309@lehigh.edu

Shenchi Tian  
Computer Science and  
Engineering  
Lehigh University  
27 Memorial Dr. West  
Bethlehem, Pennsylvania  
18015  
sht315@lehigh.edu

Evan Mulbry  
Computer Science and  
Engineering  
Lehigh University  
27 Memorial Dr. West  
Bethlehem, Pennsylvania  
18015  
ecm216@lehigh.edu

Steven R. Stinson  
Computer Science and  
Engineering  
Lehigh University  
27 Memorial Dr. West  
Bethlehem, Pennsylvania  
18015  
srs514@lehigh.edu

Brian Y. Chen<sup>1</sup>  
Computer Science and  
Engineering  
Lehigh University  
27 Memorial Dr. West  
Bethlehem, Pennsylvania  
18015  
chen@cse.lehigh.edu

## ABSTRACT

Molecular biologists seek to explain why similar proteins bind different molecular partners. The visual examination and comparison of binding cavities in protein structures can reveal information about the molecular partners that bind to a protein. Similarities can reveal regions that accommodate similar molecular fragments, while differences in binding preferences can arise from regions where binding sites differ. By comparing the binding cavities of multiple proteins, further information about the specificity of each protein can be discovered. But the visual examination of protein structure is a difficult cognitive task that requires persistence and quantitative precision. Software supports these efforts, but software for analyzing structure is difficult to use for investigators without computational backgrounds. By enabling non-computational users to better use analytical software, we hope to support progress in structural biology. Below, we present LeapRenderer, a 3-dimensional gesture-driven interface for visualizing protein surface models controlled primarily by the Leap Motion Controller. LeapRenderer serves exploration and classification functions. It allows researchers to explore the protein structures by manipulating 3-D renderings via rotation and scaling. It also aids researchers in categorizing similar proteins into groups by providing a simple interface for comparing and sorting protein surfaces. These capabilities thereby support the discovery and classification of protein binding sites.

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. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
MARS'14, June 16, 2014, Bretton Woods, New Hampshire, USA.  
Copyright 2014 ACM 978-1-4503-2823-4/14/06 ...\$15.00.  
<http://dx.doi.org/10.1145/2609829.2609838>.

## Categories and Subject Descriptors

[**Human-centered computing**]: Scientific visualization;  
[**Applied computing**]: Computational biology

## General Terms

Human-Centered Computing, Computational Biology

## Keywords

protein structure analysis, gesture-recognition, human computer interaction

## 1. INTRODUCTION

Examining protein structures is useful when trying to examine the biochemical mechanisms of proteins. Well-used programs exist for rendering protein structures, such as PyMol [2], GRASP [6], and Chimera [7], but the interfaces for interacting with these visualization tools can be encumbered with complex rendering options and analyses. To examine the possibility of simplifying these interfaces, this paper considers gesture controls to provide a more intuitive user experience.

The visual manipulation of three dimensional objects rendered on a screen through gesture-based interfaces has a long history. Recent applications of gesture-based technologies focus on interactions with virtual objects, such as radiology images [10] or volumetric objects [3]. Other applications focus on interaction realism via two handed interaction [8] or proprioception [5]. To recognize user gestures, a range of fundamental tools have been developed for camera based recognition [4], using boundary extraction [1], depth images [9], and principal components [13].

This paper uses consumer gesture-recognition technology in our software, *LeapRenderer*, to assess how effectively a

<sup>1</sup>Corresponding Author

gesture-based system can simplify two specific cognitive challenges:

- **Exploration:** The visual examination of multiple protein binding cavities, both individually and superposed.
- **Classification:** LeapRenderer supports user efforts to classify binding cavities into groups based on visual similarity.

Classification is a novel capability not supported in existing visualization tools that is made possible with the LeapRenderer interface. Because cavities exhibit complex three dimensional shapes that can be similar or vary in subtle ways, this function will ultimately support studies in structural biology that may reveal how protein function is achieved.

## 2. SOFTWARE DESIGN

LeapRenderer is built on top of three technologies:

- **surfRenderer:** surfRenderer is an application for rendering 3-dimensional models of protein surfaces. LeapRenderer uses surfRenderer to load and draw protein surface models and manipulate some aspects of the surfaces' appearances such as their levels of transparency. surfRenderer functions by parsing ".SURF" files which contain sets of points that define triangles to be drawn by OpenGL as well as some additional information about the appearance of protein surface models (e.g. color).
- **Leap Motion Controller:** The Leap Motion Controller [11] is an inexpensive device for tracking hands, fingers, and pen-like objects in 3-dimensional space. LeapRenderer uses the Leap Motion Controller to capture gestures which are then used to manipulate the protein surface models. The Leap Motion Controller provides three types of spatial information: the location of fingers, hands, and pen-like objects in euclidean space, motion vectors for individual fingers and pen-like objects, and spherical representations of hand curvature.
- **\$1 Unistroke Recognizer [12]:** The \$1 Unistroke Recognizer is an algorithm for recognizing 2-dimensional single-stroke gestures. Once 3-dimensional information about finger positions is captured by the Leap Motion Controller, LeapRenderer projects it into two dimensions and passes it to the \$1 Unistroke Recognizer. Based on the path of the collected points, the recognizer makes a decision about whether or not a known gesture has been mimed and if so, makes a prediction about the identity of the gesture. The \$1 Unistroke Recognizer provides a set of twelve built-in gestures, including triangles, check-marks, zig-zags, arrows, and more.

## 3. IMPLEMENTATION

A screenshot of the initial version of LeapRenderer can be seen in Fig. 1. LeapRenderer takes a "two-surface" approach to comparing proteins. If one wants to compare an unknown protein with a set of known, related proteins, there are multiple approaches to accomplishing this. One way is to load the unknown protein and all proteins in the known set to get a general idea of whether or not the unknown protein belongs in the set. However, this presents two major problems. First, it is difficult to know how to compare an

individual protein surface to many other protein surfaces at the same time; there's simply too much spatial information being presented. Second, 3-D rendering can be computationally expensive. The more surfaces displayed, the greater the computational burden. In order to avoid these two issues, we only load the unknown surface and one of the surfaces in the known surface set at a time. We then provide simple methods for switching which surface from the set of known proteins is displayed. This design allows the unknown protein surface to be compared to every protein surface in the set of known proteins without exposing the user to too much information at once, while limiting rendering time.

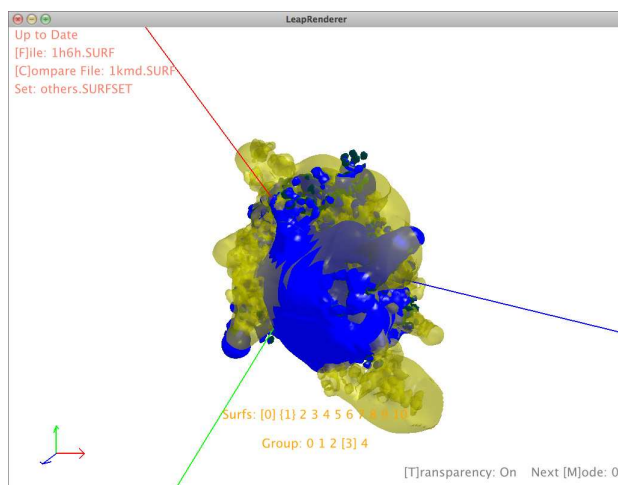


Figure 1: The LeapRenderer GUI

The second major design choice we faced was deciding how many and which gestures to recognize. The prototype version of LeapRenderer makes use of eight general classes of gestures. These include gestures able to be directly recognized by the Leap Motion Controller and gestures recognized by the \$1 Unistroke Recognizer. The Leap Motion Controller is able to detect translation, swiping, and circling with a finger. The gestures recognized by the \$1 Unistroke Recognizer include arrow, loop, checkmark, rectangle, and caret gestures. Every gesture also has an equivalent mouse or keyboard action in case the Leap Motion Controller is not present or unstable.

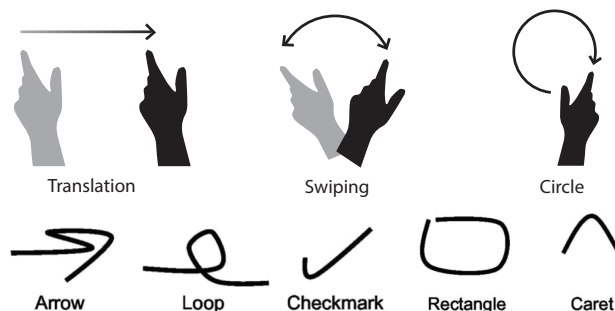


Figure 2: Gestures recognized by LeapRenderer

### 3.1 Exploration

The user loads cavities through keyboard interactions with LeapRenderer. LeapRenderer has three modes of operation: classification mode, rotation mode, and scaling mode. The default mode is classification mode. When a protein surface is loaded, a user can draw an arrow to switch modes. In rotation mode, horizontal gestures over the controller rotate the cavity, as if rolling a ball on a flat surface. In translation mode, vertical motions cause the binding cavity to zoom in and out. Lastly, LeapRenderer has a gesture for making surfaces transparent, which permits differences between two surfaces to be better examined.

### 3.2 Classification

A novel capability of LeapRenderer is to permit the user to manually categorize protein binding cavities into groups. Cavities loaded by the user are first deposited into a single group, and through inspection, the user can recategorize the cavities into different groups as needed. Group contents can then be saved and later loaded into special “.SURFSET” files, which simply record the contents of each group for later work. A structural biologist using such a tool might be examining a series of protein structure models, and deciding which models have an inactivating obstruction in the binding cavity. Moving cavities between categories during comparison enables the user to ensure that their visual comparisons result in correct classifications.

## 4. RESULTS

We focused on addressing four performance requirements:

- Gesture recognition should be stable, i.e. the software should have an acceptable accuracy rate for recognizing gestures.
- Gesture recognition should be real time, i.e. there should be minimal delay between when a gesture is made and when the software recognizes the gesture.
- The interface should be intuitive.
- Rendering of protein surface models should be fast.

In order to test gesture stability, we ran experiments where we performed each gesture thirty times in a controlled environment and for each gesture, we made note of the gesture being performed and the recognition result. The accuracy rates for each gesture are in Table 3. The average accuracy was fairly good at 82.9%. However, some gestures were better than others. The gestures built into the Leap Motion Controller had higher accuracy rates than those recognized by the \$1 Unistroke Recognizer. The recognition rates for the loop and caret gestures were lower than the others with accuracy rates in the seventies.

Some gestures were confused often with others as seen in the confusion matrix shown in Fig. 4. Swiping was sometimes confused with checkmarks. Carets were occasionally confused with arrows and checkmarks. Loops tended to be confused with checkmarks and rectangles. Lastly, Rectangles were sometimes confused with loops.

As we were programming gestures, we ensured that they were recognized in real-time or near real-time. We did not test the intuitiveness of the interface, but have planned experiments to test this in the near future.

We also needed to ensure that the protein surfaces were capable of being rendered in near real-time. We conducted

Gesture	Accuracy
Swipe	0.833
Circle: 1 Pointer	0.933
Circle: 2 Pointers	0.933
Arrow	0.833
Caret	0.733
Checkmark	0.833
Loop	0.700
Rectangle	0.833
Average	0.829

Figure 3: Accuracy rates for gestures

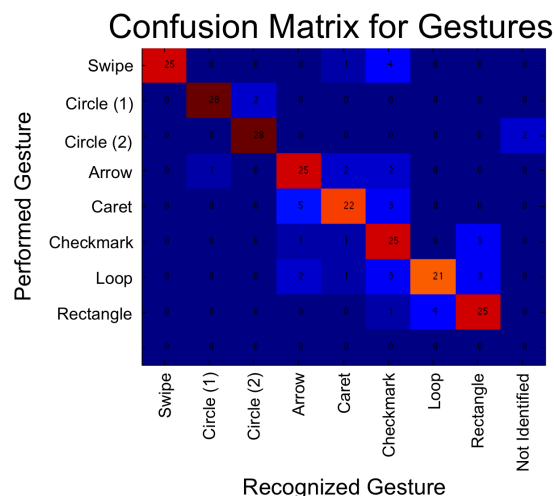


Figure 4: Some gestures were occasionally confused with others, but acceptable recognition was achieved overall.

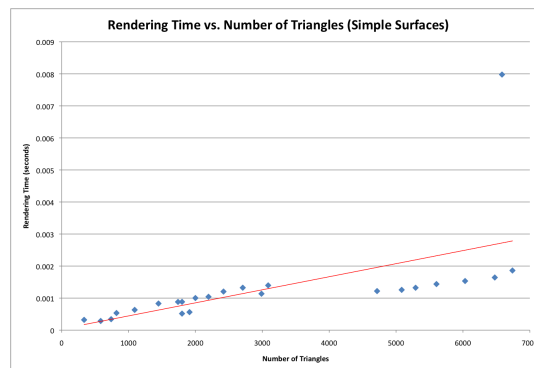


Figure 5: Rendering time for simple surfaces

an experiment to test this. We timed how long it took to render individual surfaces models for 35 proteins. Ten of these surfaces were complex, having over 100,000 component triangles. Twenty five surfaces were simple, having less than 10,000 component triangles. The results of the experiment are shown in Figs. 5 and 6. In every instance, the rendering took fractions of a second, and the rendering time appeared to grow linearly with the number of triangles. This means the rendering of protein surface models scales fairly nicely.

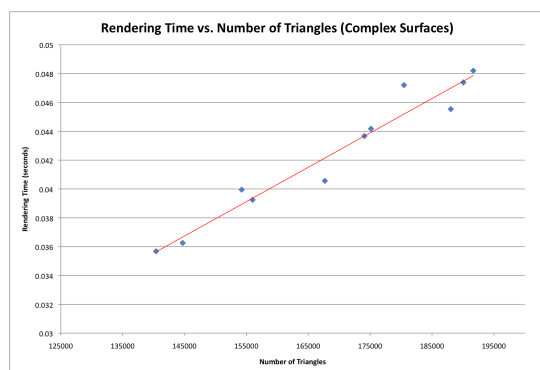


Figure 6: Rendering time for complex surfaces

## 5. DISCUSSION AND FUTURE WORK

We showed that it is possible to create a gesture-driven interface for exploring and classifying protein surface models. We showed that such a system can run in real-time while being able to recognize gestures fairly stably. This shows that the Leap Motion Controller can be used for serious applications. Next, we will investigate how to make the application more intuitive, and eventually, we hope to prove that our gesture-driven interface is intuitive. If we are capable of achieving this goal, we hope to release the software to the scientific community and aid researchers in identifying the structure and functionality of proteins.

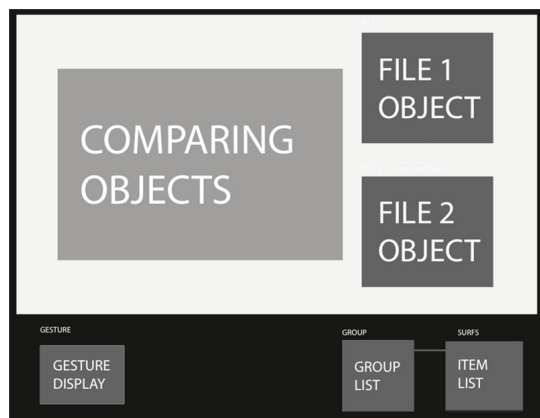


Figure 7: Interface layout based on initial mind mapping studies.

To improve the current user interface and experience, we are applying a psychological and a graphic approach known as a mind map. A mind map is a diagrammatic representation of information that has become widely used for the simplification of software interface designs. After a first round of individual interviews with 10 students, a draft mind map has been created. The interface layout, sketched in Figure 7, reflects content sought by users that is presented in the order in which they seek this information. The top area is designed to provide visual information to the user and the bottom section is used to give feedback on the user's gesture and result of the comparison. Larger scale studies are planned for future work.

## 6. ACKNOWLEDGMENTS

This work was supported in part by NSF CCF 1320137. The authors are grateful to Dr. Jian Chen for advice on gesture-based interfaces.

## 7. REFERENCES

- [1] W. H. Chen, C. T. Hsieh, and T. T. Liu. A real time hand gesture recognition system based on dft and svm. *Applied Mechanics and Materials*, 284:3004–3009, 2013.
- [2] W. L. DeLano. The PyMOL Molecular Graphics System, 2002.
- [3] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 61–70. ACM, 2004.
- [4] L. H. Li and J. F. Du. Visual based hand gesture recognition systems. *Applied Mechanics and Materials*, 263:2422–2425, 2013.
- [5] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26. ACM Press/Addison-Wesley Publishing Co., 1997.
- [6] D. Petrey and B. Honig. GRASP2: visualization, surface properties, and electrostatics of macromolecular structures and sequences. *Method Enzymol*, 374(1991):492–509, Jan. 2003.
- [7] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera—a visualization system for exploratory research and analysis. *J Comput Chem*, 25(13):1605–12, Oct. 2004.
- [8] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, pages 1–54, 2012.
- [9] L. Song, R. M. Hu, H. Zhang, Y. L. Xiao, and L. Y. Gong. Real-time 3d hand gesture detection from depth images. *Advanced Materials Research*, 756:4138–4142, 2013.
- [10] J. P. Wachs, H. I. Stern, Y. Edan, M. Gillam, J. Handler, C. Feied, and M. Smith. A gesture-based tool for sterile browsing of radiology images. *Journal of the American Medical Informatics Association*, 15(3):321–323, 2008.
- [11] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Basel, Switzerland)*, 13(5):6380, 2013.
- [12] 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. ACM, 2007.
- [13] Z. Q. Zhang and W. Zhu. Robust hand gesture detection based on feature classifier. *Advanced Materials Research*, 823:626–630, 2013.