Darien Nurse

COMS W4172

The IEEE Symposium on 3D User Interfaces focuses on the topic of 3D User Interfaces (3D UI). It provides a unique opportunity for industrial and academic researchers to exchange about the state-of-the-art 3DUI research. The symposium also hosts a 3DUI contest that is open to anyone interested in 3D user interfaces, from researchers to students, enthusiasts, and professionals. Participants can use any software or hardware for the contest. The purpose of the contest is to stimulate innovative and creative solutions to challenging 3D user interface problems.

This year, 2014, five teams competed in the competition: Slice-n-swipe, Bi-Manual Gesture Interaction for 3D Cloud Point Selection and Annotation using COTS, The Point Walker Multi-label Approach, Touching the Cloud, and Go’Then’Tag. Each team was faced with the challenge of building a system that allows users to annotate 3D point clouds obtained from 3D scanners. The system must support the accurate labeling of sets of points, not just apply to a single point. It must also support overlapping hierarchies of annotations at varying scales. Lastly, contestants must create and a 90-120 second video demoing their solution along with a 2 page abstract. This paper will compare and contrast three of these entries: Slice-n-swipe, Touching the Cloud, and Go’Then’Tag. My analysis will be based on how well each team adheres to Jakob Neisen’s 10 Usability Heuristics for user Interface Design. Special emphasis will be placed on why I think each team used the technique that they chose for a given task (i.e. camera/viewpoint manipulation, selection, annotation), and whether that technique was a good choice. Also, if I think that a team should have selected a different technique for any of their tasks, then I will explain why, taking into account the other techniques used in that team's solution. Other concept I will consider are; how efficient are the chosen technique, how easy they are to use, and how enjoyable they are to use. Before discussing my analysis, it is important to first give some background information of these three teams.

Slice-n-swipe is a technique for 3D point cloud annotation based on free-hand gesture input (Bacim, Nabiyouni, & Bowman, 2014). The technique is based on kitchen knife, bubble, and lasso metaphors and is used to select and manipulate objects. This team also uses a process called progressive refinement, which to allow the user to specify the points of interest and recursively edit and label subgroups of those points. Slice-n-Swipe uses Leap Motion Controller for free-hand gesture input for the user’s dominant hand, a 3DConnexion SpacePilot Pro 3D mouse for virtual camera control for the user’s non-dominant hand, and a keyboard to annotate groups of points [image].

Touching the Cloud is a bi-manual user interface for the interaction, selection and annotation of immersive point cloud data (Lubos, Beimler, Lammers, & Steinicke, 2014). This team sets itself apart from other team in the competition by developing a solution that does not require the user to physically interact with hardware, meaning that all actions are performed through speech or hand gestures. The setup involves three main components; an Oculus Rift head-mounted display (HMD), a PrimeSense Carmine 1.09 Sensor, and a laptop with an Intel Core i5 2.3 GHz processor. The Oculus allows users to immerse themselves in an environment optimized for natural interaction with point cloud data. The PrimeSense facilitates hand and ﬁnger tracking, providing an interaction volume of 60cm width and 50cm depth with the sensor at a height of 65cm. The laptop is used to process and display the scene on an Oculus Rift HMD and the attached microphone is used for annotation [image].

Go’Then’Tag is a tool-set that helps users to edit complex 3-D data sets, and tag them at different levels (Veit & Capobianco, 2014). All interaction is handled through a tracked multi-touch device that combines 2-D and 3-D interaction techniques on a single device. Like Slice-n-Swipe, Go’Then’Tag also uses progressive refinement to reduce the complexity of the selection process. The environment is composed of one 3-D stereoscopic display with an optical tracking system capturing the head and device positions. The multi-touch device used for this demo is a Galaxy Note 2 running Android 4.3[image].

According to Nielsen’s first heuristic for interface design, the state of the system should always be visible to the user. Meaning that when a user interacts with the system, he or she requires some sort of audio or visual feedback indicating that the system is working or that what he has done is either correct or incorrect. Slice-n-swipe and Touching the Cloud focus on hand gestures for manipulation [image], thus the initial state for both project seems to be scenes in which the user can visibly see their tracked hand movements. It is assumed that the user can perform selection and manipulation in this state. They allows the user to switch between tools, using an on screen GUI in Slice-n-Swipe or voice commands using Touching the Cloud, which will change selection behavior. Go’Then’Tag has a similar initial state except that the scene shows multi-touch device’s position and orientation to the user. At any time during the interaction with the data set, multimodal feedback (including visual, audio and vibrotactile information) are given to the user (Veit & Capobianco, 2014). The only other significant state that any of the projects go into is the annotation state. Annotation will be discussed in greater detail later.

Nielsen’s second heuristic states that there should be a match between the system and the real world, meaning that the system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Judging from each team’s demo videos, none of them seem to use any words at all, other than typical modeling terms like selection, translation, rotation, etc. It is worth mentioning that teams Touching the Cloud and Go’Then’Tag mentioned in the evaluation section of their write-up that they tested their interface on novice users. The users did not issues regarding this heuristic.

Nielsen’s third heuristic states that systems must grant user control and freedom, meaning that users should be able to easily undo/redo mistakes. In Slice-n-Swipe when action, like a slice for instance, is performed, the user can undo the slice by waiting a couple of seconds without performing any gestures. This seems like a poor way to implement undo because it slows down the user’s progress. An accelerator like a keyboard shortcut or undo gesture would be more efficient. Touching the Cloud handles undo through voice commands. Like Slice-n-Swipe, implementing a gesture accelerator would add speed and greater functionality to the interface. Go’Then’Tag does not mention error correction in its documentation but it would make sense that it would be handled through its touchscreen GUI.

Nielsen’s fourth heuristic deals with consistency and standards. Users should not have to wonder whether different words, situations, or actions mean the same thing. The system should follow platform conventions. Since modeling software typically has a steep learning curve, consistency allows users to learn commands and techniques faster and interact with the system more efficiently. Using natural user interface technologies to manipulate and label point cloud data seems to be a relatively new problem so there are not many standards associated with this concept. For now, each project only needs to implement feature commonly found in most modeling software, like camera and object manipulation. Each project performs tool switching and annotation in consistent, unambiguous manor.

Nielsen’s fifth heuristic deals with error prevention. In addition to providing the user with helpful error messages, the designer should try to avoid these errors in the first place. It is the designer’s job to either eliminate error-prone conditions or check for them and present the user with a confirmation option before he commits to the action. Should an error occur, Nielsen’s ninth heuristic states that the system should help users recognize, diagnose, and recover from error. Each team’s demo videos do not intentionally show errors within their interface, nor do they discuss error in their write-ups. Since I do not have access to these interfaces myself, I can speculate on errors that might occur when using the interface. How does the system notify users when hand/device tracking is malfunctioning? Does this system restrict illegal annotations (Duplicate names, invalid symbols, etc.)? Does the system prevent the annotation of an empty set of points? Does the system prevent the user from applying more than one annotation to a point/set of points?

Nielsen’s sixth heuristic states that the system should emphasize user’s recognition over the user’s recollection. Designers should minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. In other words, users should not have to remember sequences of numbers, words, or long lists of items in a task. Instead, the interface should use easily recognizable pictures and symbols to communicate with the user. Slice-n-Swipe and especially Touching the Cloud will face difficulty with this heuristic because, gestures and voice commands must memorized in order to use the interface efficiently. Forcing the user to remember these commands, sharply raises the learning curve for the interface. Go’Then’Tag circumvents these issues by making objects, actions, and options visible on the multi-touch display

Nielsen’s seventh heuristic deals with flexibility and efficient of use. Designers equip their system with accelerator which can often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. In more advanced systems that are capable of learning user patterns, as novice users get acquainted with the system and perform certain tasks frequently, the system will be able to let those frequently performed tasks be accessible more efficiently. Again, each group’s write-ups do not discuss accelerators directly but I can discuss accelerator that would fit well in each project. Slice-n-Swipe would benefit from implementing keyboard shortcuts since the keyboard is already part of the interface. Slice-n-Swipe and Touching the Cloud could use additional gesture to accomplish more complex tasks that novice users would not need to perform. Lastly, Go’Then’Tag could uses touch screen accelerators.

Nielsen’s eight heuristic deals with creating aesthetic pleasing systems. The system should not contain information which is irrelevant or rarely needed. Every extra unit of information in the system competes with the relevant units of information and diminishes their relative visibility. Of the three teams, Slice-and-Swipe has the most minimalist design. Usually, only the point cloud is present on the screen and no words, symbols, or buttons. Like Slice-n-Swipe, Touching the Cloud usually only shows the point cloud to the user. However, the immersion aspect of the interface inherently creates an additional layer of complexity. Also, Touching the Cloud uses a three color scheme for selection; red for marked points, green for selected points, and pink for marked and selected points. In comparison to the other two team which simple fade unselected points, Touching the Cloud’s colors seem distracting. Go’Then’Tag has the most complex interface because it uses two screens: one for point cloud and the multi-touch device. This does not mean that Go’Then’Tag has a bad interface, on the contrary two screens would probably make annotation simpler and more organized.

Nielsen’s tenth heuristic states that the system should provide the user with a wide array of help and documentation. According to the 3DUI contest rules, team is only required to submit a 60-120 second demo video and a short 2-page paper, with a description of the solution, details about the hardware and software used, a brief description of the closest related work and how the team’s system is novel, and a description of how the team iterated upon the design. All other details are unavailable.

Lastly, let focus on how each team handles the three key interaction tasks: camera/viewpoint manipulation to obtain a reasonable view of the region of interest, text entry for the annotation, selection of the set of points to be annotated. In Slice-n-Swipe, camera control is handled by the 3D mouse. This mouse is intended to be used by the user non-dominant hand, so to enable the user to perform the more precise task of selection with the dominant hand. The mouse is type of isometric input device, meaning that it mains stationary during use (Bowman, pp. 94-95). It provides six-degree-of-freedom input, which enables camera translation and rotation in the 3D environment. Translational force causes the virtual camera to translate in the scene, while rotational force causes the camera to rotate about the center point of the 3D dataset (Bacim, Nabiyouni, & Bowman, 2014). The environment generated by the camera is presented to the user as exocentric view, meaning that user’s body is external to the environment (Bowman, p. 232).

Instead of an exocentric view, Touching the Cloud’s use of the Oculus Rift HMD allows the user to experience the scene from an egocentric perspective. An egocentric view means that the user experiences the scene as if he or she was actually inside the scene (Bowman, p. 232). This immersive experience results in an interface that is fundamentally different from Slice-n-Swipe and Go’Then’Tag. For instance, the scene camera is now attached to the position of the user’s head which means the user will not need to use his hands to manipulate the camera. The user now has the ability to manipulate the point cloud through gestures. Those unfamiliar with immersive technology may find it disorienting. If disorientation is common problem for Touching the Cloud users, then implementing wayfinding techniques could mitigate confusion.

Go’Then’Tag shifts back to the exocentric view. When a point or set of points is selected, the position and scale of the selected points are automatically calculated such that the camera can provide an adapted point of view to the current selection. The user can then ﬁnely adjust the orientation by rotating the selection around the x and y axis through sliding gestures on the touch screen.

For annotating the point cloud, each team used the method of progressive refinement. However, each team used a different method for text input and each method had benefits and drawbacks. With Slice-n-Swipe, user provides an annotation by typing on the keyboard. The annotation appears in the 3D environment near the set of points to which it refers, and stays near those points as the user manipulates the virtual camera. The system allows as many annotations as desired, and points can have multiple annotations. Out of all the text input methods, a keyboard is probably fast form of text input. However, Slice-n-Swipe’s keyboard requires that the user remove his hands from the 3D mouse and Leap Motion’s tracking area. Repeatedly switching between the keyboard and the other peripherals could become tedious.

Since Touching the Cloud’s HMD obstructs the user’s view of the real world, a keyboard is not a viable option for text input. So stay in line with the theme of natural interaction, Touching the Cloud’s speech recognition feature allows the user to simply say the annotation after the predefined keyword ‘annotate’. The user can also use enumerated annotations. Speech recognition free up the user’s hands, but the software may need to be trained to the user’s speech patterns and requires a quite environment.

What sets Go’Then’Tag apart from its competitors is that its interface in centered on providing the most efficient annotation possible. Even though manipulation and selection must be done in 3-D, this team recognized that annotating data sets, manipulating tags and editing text are more easily achieved using a 2-D interface. The touch screen implements GUI that helps the user manage the hierarchy generated from possessive refinement. The only drawback to this method is that text input efficiently will vary with the screen size of the touch screen device.

Of three key interaction tasks, selection is the most difficult and complex. The selection technique used by Slice-n-Swipe based on free-hand gestures detected by a Leap Motion device. The interface’s main tool, the knife, allows users to iteratively cut the dataset (slice) and remove all unwanted points (swipe) until the desired selection is achieved. Each selection tool; knife, lasso, and bubble, utilizes a concept called hand ray casting. Hand ray casting is a unimanual technique in which the user emits a virtual ray from their finger into the environment, usually for object selection or manipulation (Bowman, p. 151). These selection tools form the foundation for Slice-n-Swipe because: 1) they do not require extreme precision; 2) they are easy to understand; 3) they avoid the specification of parameters like shape and size; and 4) they do not require tracking of the orientation of the the hand or the position of multiple fingers.

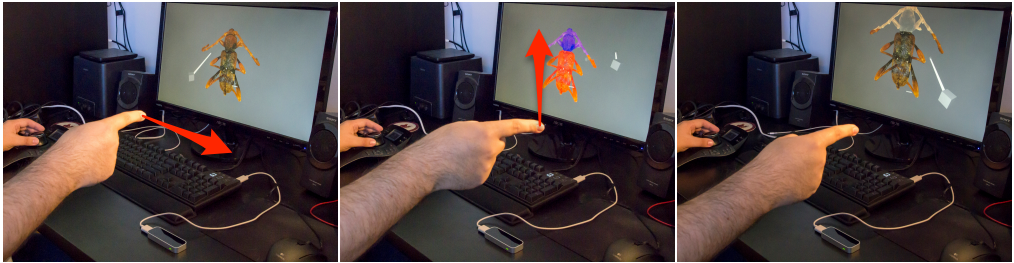


Figure 1: Three stages of the Slice-n-Swipe technique: the user prepares to slice the dataset (left); the user slices the dataset, resulting in two

Touching the Cloud improves upon the hand gesture selection approach by using virtual hand techniques. A virtual hand is a hand-shaped 3D cursor that maps to the position and orientation of the user’s hand (Bowman, p. 160). This allows to interface to implement a wide array of symmetric and asymmetric bimanual tasks. For instance, point cloud rotation, scaling, and translation all implement symmetric bimanual pinch gestures.

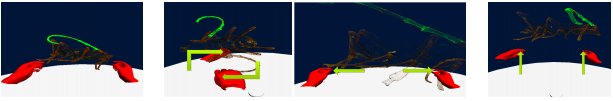


Figure 2: Supported transformations: (a) initial position, (b) rotation, (c), scaling, and (d) translation.

The user can also mark points by touching them with his fingertip given a customizable tolerance radius. Marked points can be set as selected or, should the user accidentally select too many points, they can be easily unselected, as well. Once the selection is ﬁnished, the user can ﬁnalize it and annotate it.

Unlike the other two teams, Go’Then’Tag does not have the ability to track hand gestures. Instead, all manipulation and selection is handled by the isotonic trackers attached to the multi-touch device. Isotonic devices move with the user’s body in order to measure the user’s position and orientation (Bowman, p. 94). Selection is performed through a 3-D selection tool: a sphere attached to a ray controlled by the tracked device. Radius of the selection tool can be modified at will. The original point cloud is automatically divided using an octree. The user can navigate in this hierarchy. When the desired subpart is reached, only the points contained in the current node can be edited or annotated.

The final results of the contest show that Go’Then’Tag was the best interface followed by Slice-n-Swipe. Prior to starting this analysis, I was disappointed to see that Go’Then’Tag had won, mostly because it does not use any natural user interface technology.

Hand tracking is still difficult. Lack of tactile feedback.

Body positioning

Head position

Use of hands

Go'Then'Tag’s attached trackers seem cumbersome

# Bibliography

Bacim, F., Nabiyouni, M., & Bowman, D. A. (2014). *Slice-n-Swipe: A Free-Hand Gesture User Interface.* Minneapolis: IEEE Symposium on 3D User Interfaces 2014.

Lubos, P., Beimler, R., Lammers, M., & Steinicke, F. (2014). *Touching the Cloud: Bimanual Annotation of Immersive Point Clouds.* Minneapolis: IEEE Symposium on 3D User Interfaces 2014.

Nielsen, J. (1995, January 1). *10 Usability Heuristics for User Interface Design.* Retrieved from Nngroup: http://www.nngroup.com/articles/ten-usability-heuristics/

Veit, M., & Capobianco, A. (2014). *Go’Then’Tag: A 3-D point cloud annotation technique.* Minneapolis: IEEE Symposium on 3D User Interfaces 2014.