

# Yea Big, Yea High: A 3D User Interface for Surface Selection by Progressive Refinement in Virtual Environments

Bret Jackson\*

Brighten Jelke†

Gabriel Brown‡

Department of Mathematics, Statistics, & Computer Science  
Macalester College

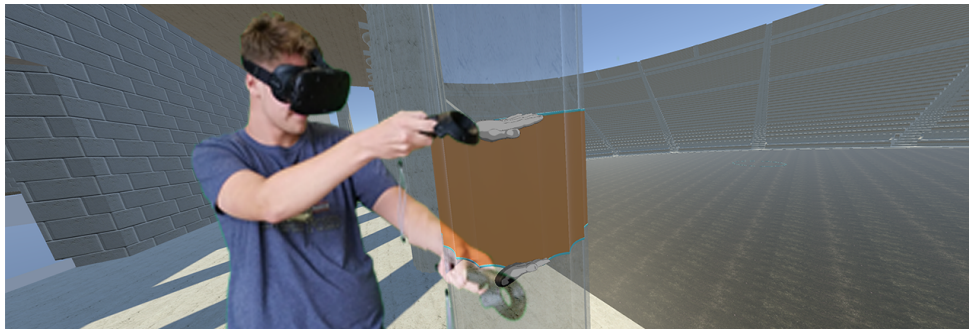


Figure 1: A user selects part of a column in a reconstruction of a cultural heritage site. The selection is constrained between the user’s hands. After setting an initial selection, the selected surface patch can be constrained further through progressive refinement.

## ABSTRACT

We present *Yea Big, Yea High* — a 3D user interface for surface selection in virtual environments. The interface extends previous selection interfaces that support exploratory visualization and 3D modeling. While these systems primarily focus on selecting single objects, *Yea Big, Yea High* allows users to select part of a surface mesh, a common task for data analysis, model editing, or annotation. The selection can be progressively refined by physically indicating a region of interest between a user’s hands. We describe the design of the interface and key challenges we encountered. We present findings from a case study exploring design choices and use of the system.

**Index Terms:** Human-centered computing—Human computer interaction—Interaction techniques—Gestural input; Human-centered computing—Human computer interaction—Interaction paradigms—Virtual reality

## 1 INTRODUCTION

Selection is one of the most important low-level interaction tasks in user interfaces. It is the primary way that users determine how their interactions will affect the system and environment in which they are working. For creative applications like three-dimensional (3D) modeling, selecting subsets of vertices or triangles is fundamental to being able to modify the shape of geometry. Similarly, for visualization applications, users must be able to select subsets of the data to query or filter for more analysis [29]. Because of its importance in these types of applications, significant effort has been made towards developing appropriate selection metaphors. In desktop-based systems, techniques like picking, lassoing, and brushing are used for selecting multiple parts of a model or subsets of a dataset; however,

these techniques are less adequate for interacting in virtual environments with 3D user interaction. In these types of environments, selection is more difficult because the added dimension of depth adds additional ambiguity, and 3D user interfaces commonly lack the control provided by interacting with a 2D mouse or keyboard.

Given the recent availability and wider adoption of low-cost virtual reality (VR) displays, developing effective three-dimensional user interfaces (3DUIs) for selection is increasingly critical. Predominately, existing 3DUI selection interfaces rely on selecting entire objects. For example, it is now quite common to directly select a complete object by ray-casting [15] or using a virtual hand [22] metaphor. Unfortunately, these types of interfaces are limited for applications in 3D modeling and scientific visualization, where a user might want to select part of an object (e.g. a set of triangles or vertices in a model) or a volume of interest (e.g. a set of streamlines running through a specific spatial area). Although interfaces exist that allow users to pick multiple points through brushing or multiple picking in 3D, they can be time consuming and fatiguing when a user wants to select large numbers of points [2]. Existing techniques that do allow for selection of multiple points or a volume of interest frequently rely on predefined shapes, such as a cube [23], to indicate the selection volume. These predefined shapes do not adapt well to complex selection regions that do not match the given shape [32].

This necessitates novel interfaces that support complex volumetric selections in an intuitive way. In this paper, we present *Yea Big, Yea High*, a 3DUI for selection in virtual environments. Named after a phrase that became popular in American slang in the 1950s and 1960s, meaning “this big, this high”, the phrase is usually accompanied by a spreading of the hands to indicate the size. Following a similar metaphor, our interface allows a user to indicate a selection by positioning their hands apart to indicate a region of interest between them. This action can be repeated multiple times with the hands held in different orientations in order to progressively refine the selection by indicating the boundary of the 3D selection volume.

Although the interface generalizes to a full volumetric selection, we have chosen to focus the implementation discussion around selecting a subset of a 3D surface mesh. The case study described in Section 5 details how this type of interface might be used to support annotations of a virtual 3D model. By discussing the iterative

\*e-mail: bjackson@macalester.edu

†e-mail: bjelke@macalester.edu

‡e-mail: gbrown3@macalester.edu

development, lessons learned, and limitations of this application of the interface, we hope others in the 3DUI community developing selection interfaces will benefit in the future.

The primary contributions of *Yea Big*, *Yea High* are:

- A bimanual 3D user interface for selecting portions of surface meshes using a novel metaphor.
- Use of progressive refinement within the interface to allow for complex selections
- A case study detailing how the selection interface could be used as an annotation tool within virtual reconstructions of cultural heritage sites.

In the following sections, the paper begins with a discussion of related work on 3D selection interfaces. We then describe the interface in detail, followed by a case study showing how it might be applied. Finally, we close with a discussion of limitations and future work.

## 2 RELATED WORK

There is a long history of developing 3D user interfaces for selecting objects in virtual environments. Early work in this area involves ray-cast selection [15], where a virtual ray extends from a user's hand to intersect with objects that are then selected, or direct selection using a virtual hand metaphor [16]. Both of these techniques have been extended to work at long distances and at different scales (e.g. through nonlinear mapping of the interaction space [19] or through interacting with a world-in-miniature [18, 22]).

Unfortunately, both techniques suffer from issues of control, particularly when selecting among a dense set of objects. In ray-casting, small muscular jitters at the hand are amplified along the ray making selection difficult. Occlusion from the user's current viewpoint also increases the difficulty of selection in dense object sets [21].

Many variations of the ray-casting technique have been proposed to address these issues. For example, Liang and Green [12] proposed the spotlight technique that uses a projection cone rather than a ray to require less precision in the user's hand movements. Forsberg et al. [6] extended the technique using an aperture-based approach where the user could control both the cone direction and aperture size. Others have looked at disambiguating the selection for closely spaced objects by snapping the ray [30] or a 3D bubble cursor [25] to the closest object or by bending the selection ray towards the object that has stayed inside the selection cone the longest [5].

Another common approach uses the concept of progressive refinement of the selection to enable increased precision. The SQUAD interface [11] allows a user to make an initial selection using a sphere encompassing multiple objects, and then to refine the selection by choosing from that smaller subset displayed in different quadrants of the screen. Cashion et al. [4] build on this technique, proposing the Expand interface. Expand addresses the problem of users losing the spatial context for items when they are displayed in different quadrants on the screen in the progressive refinement step. It allows a user to interactively move the objects from their original positions in the environment onto a grid to preserve context. Our own interface uses progressive refinement with similar motivation to gradually reduce the size of the selection until it just encompasses the user's intended subset.

While these techniques work well for selecting single objects, they are limited for selecting multiple objects. Lucas et al. [13] introduced a multiple object selection technique that uses a tablet to draw 2D lassos on a camera view of the 3D environment. Keefe et al. [9] introduced Dynamic Dragging as a technique for controlled drawing of 3D lassos around multiple objects that need to be selected. Zhou et al. [33] extended this technique using progressive refinement and haptics to select multiple streamlines in a flow visualization. These

techniques work well for selecting objects like streamlines, where the lasso is drawn perpendicularly to the direction of the line and clearly indicates which bundle should be selected; however, it is less well suited for selecting part of an object such as a subset of vertices in a 3D mesh or data within a volume rendering, which is the primary goal of our interface.

Ullinski et al. [23] introduced a volumetric selection approach to address this challenge using a six-sided box that the user controls with both hands. Our own approach is similar to this technique in that objects or portions of an object between the user's hands are selected. Indeed, we use a similar box as an activation cube to filter possible candidates for selection as defined in Section 3.1. However, unlike our approach which allows the user to refine the selection to arbitrary shapes, the box selection technique is limited to data that falls within a cube.

Serenio et al. [20] overcome this limitation by allowing a user to draw a shape on the surface of a tracked tablet. The shape is extended into a volume by moving the tablet through space. This approach is somewhat of an inverse to our technique — relying on growing a selection region rather than progressively trimming away unwanted data.

Perhaps most similar to our technique is the slice-n-swipe technique by Bacim et al. [1]. This technique allows a user to filter an unstructured point cloud dataset by making a slicing gesture with their finger. Using the metaphor of a chef's knife, this slicing action cuts the point cloud in two and a swipe gesture discards one half. Although the *Yea Big*, *Yea High* technique also uses cutting planes to segment a selection subset, the metaphor is different. Also, unlike slice-n-swipe, its bimanual approach allows a user to make two cuts at once while implicitly defining which side of the cutting plane to keep, reducing the number of progressive refinement steps needed to make a selection.

*Yea Big*, *Yea High* also draws similarity to the Large Scale Cut Plane technique of Mossel and Koessler [17], that allows a user to define a large cutting plane through a dense 3D model, like the ones used in our case study. However, the goal of Mossel and Koessler's technique is not to define the selection, but rather to remove occlusions so that the remaining objects are easier to select. Additionally, the scale of the cutting plane affects the entire model rather than just the portion that is between the user's hands, as in our approach.

Selecting only a part of an existing object can also be improved through context-aware techniques like CloudLasso or TeddySelection [31], which use an initial 2D drawn lasso and then infer a selection volume by analyzing particle density within the lassooed region. Newer approaches like SpaceCast, TraceCast, and PointCast [32] better attempt to take the shape and position of the lasso into account when deriving the underlying selection. These types of context-aware techniques are not limited to point clouds. Force Brushes [8] allows a user to define an initial selection along a streamline and then to extend the selection based on similar data values. Similarly, VolSelectAware [28] allows a user to define a selection point in volumetric data and grow the selection to similar regions. While these systems allow for more precise selection with fewer refinement steps, they predominately are domain specific, requiring different growing/inflating metrics depending on the underlying data that is selected. In order to generalize to a wider range of virtual environments, our technique focuses on directly defining data within a selection volume. However, the *Yea Big*, *Yea High* interaction could easily be combined with a context-aware approach, using it to define an initial region of interest that is then refined through domain specific methods.

## 3 YEA BIG, YEA HIGH SELECTION

*Yea Big*, *Yea High* is designed to be used in an immersive VR environment. The interface depends on tracking either the user's hands

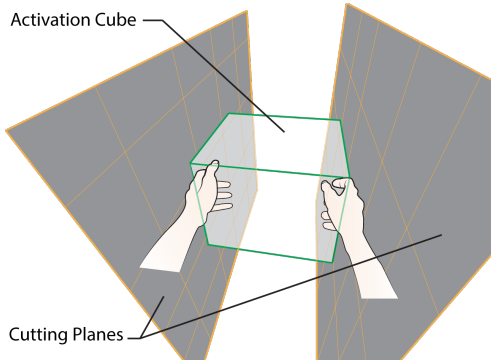


Figure 2: Objects that intersect with an invisible activation cube, visualized here between the user’s hands are considered as candidate objects for possible selection actions. The final selection consists of the portions of the candidate objects that fall between infinite cutting planes (visualized as finite planes for clarity) passing through the user’s hands.

or tracked devices held in either hand. As shown in Figure 1, our implementation uses the HTC VIVE, a commonly available head-mounted display, and its two tracked wands; however, the interaction technique is also applicable in CAVE<sup>TM</sup> environments or with other types of VR displays.

### 3.1 Interaction Design

Many times in our lab, we have discussed a 3D model or scientific dataset in an immersive environment with a group of collaborators. Invariably, the need arises to direct the group’s attention to a specific region of space to indicate a subset of the data or a portion of the model’s 3D surface for discussion. Most often, we have seen users indicate this region by moving both hands through the air to loosely define the boundaries of the region, and letting our minds fill in the rest.

Based on these experiences, there are several requirements that motivate the technique’s design rationale:

1. The design should support selecting arbitrary surface patches from within large, dense, 3D environments. Patches might represent an entire object within the model or just a smaller portion of an object.
2. Users should be able to define patches with irregular shape boundaries to make complex selections, while only precisely selecting points of interest.
3. The technique must be easy to understand in order to foster ease of use.
4. It must be quick to perform to allow for many repeated selections with limited fatigue, and it must run at interactive rates.

The interaction design attempts to address these requirements by allowing a user to indicate the boundaries of the selection region using a bimanual approach. The selection boundaries are determined using two infinite cutting planes (Shown in gray in Figure 2), each attached to one of the user’s hands. Prior work has found that bimanual selection techniques using symmetric-synchronous actions, with both hands performing the same action at the same time, result in faster task completion times [24]. As a result, only the position of the user’s hands, with three degrees-of-freedom (3DOF) each, is used to set the position of the cutting planes. The orientation of the planes is set so that the normal of the right-hand plane is always

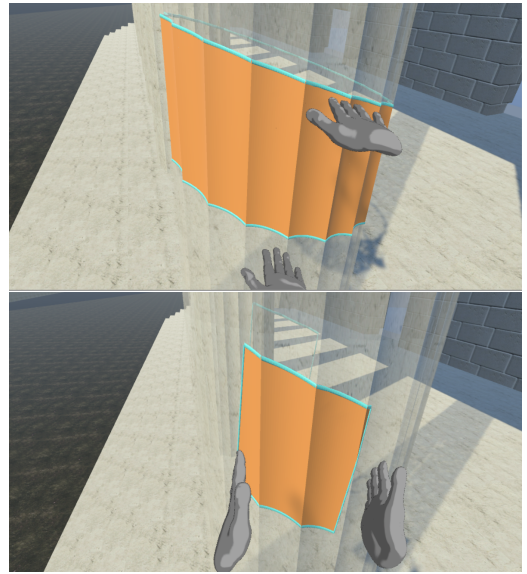


Figure 3: Top: The cutting planes attached to each hand segment the model. Bottom: A progressive refinement step constrains the selection.

pointing directly at the position of the left hand and vice versa. This causes the planes to always be parallel and facing each other.

To begin a selection, the user approaches the object or objects that they would like to select, holding out both hands. Shown in Figure 1, our implementation, which focuses on selecting surface patches from a 3D mesh, visually indicates the selected portion that is between the user’s hands by highlighting the surface in orange. Because the cutting planes extend infinitely, the model is made slightly transparent to facilitate placement of the planes. Further visual feedback is given in the form of 3D tubes outlining the intersection of the planes and the 3D model.

If the user chooses to move away from the object at this stage, the surface texture will be reset to its original state. Otherwise, they can continue to move both hands closer or farther apart until the desired region is highlighted. To confirm the selection, the user clicks a button on the tracked controller. If the user is satisfied with the selection, they can now move away and the selected region will remain highlighted.

Although the design could select all geometry between the infinite cutting planes, in practice, we limit the selection to objects between the user’s hands. This makes intuitive sense (placing your hands around an object is a clear indication that it is the focus of your attention), and it also helps meet the design requirement that the interaction run at interactive rates.

To determine which objects are candidates for selection, an invisible activation cube (Shown in green in Figure 2) is anchored midway between the user’s hands. This is similar to the selection cube used by Ulinski et al. [23], but rather than being used for the final selection, it is used to reduce the amount of model geometry that needs to be tested for intersection with the cutting planes. The size of the activation cube scales proportionally to the distance between the user’s hands, so that spreading the hands apart allows a user to include more candidate objects for possible selection. In addition, it is offset forward slightly, so that the user can indicate areas of interest on the ground.

The technique allows for more complex and precise selections by repeating the cutting process as many times as desired to progressively refine the regions boundaries. In each case, the highlighting will appear only on regions contained both within the user’s hands



and the previously defined selection (Figure 3 bottom).

The *Yea Big, Yea High* metaphor comes from this progressive refinement. It is based on the English phrase and common gestures used to indicate the size of something by first showing the width dimension with one’s hands and saying “Yea big”, followed by the height dimension and saying “Yea high”. Note, however, that the interface does not require the user to progressively refine by indicating orthogonal angles. Indeed, although somewhat tedious, a user could indicate a cylindrical selection by making many progressive refinements while rotating the hands.

#### 4 IMPLEMENTATION

Our implementation of the interface uses the Unity3D game engine. When running, the user exists in one of two states: Navigation or Selection. In the Navigation state, the user can move around the environment, and they will see virtual models of the HTC VIVE controllers at the location of their hands. When the user enters the Selection state, the controller models are replaced with hand models to indicate the transition. The current state is determined by checking for object intersections with the activation cube anchored between the two controllers. If any model objects collide with the activation cube, then the user enters or stays in the Selection state; likewise, if nothing is between the user’s hands then they enter or stay in the Navigation state.

In the Selection state, the interface splits each potential selection object into two submeshes: one containing the selected portion and one containing the unselected portion. The algorithm for determining these selection meshes is as follows:

---

**Algorithm 1:** Selection algorithm implemented using Unity3D

---

```

for each object colliding with the activation cube do
  for each triangle in the object’s mesh do
    if left hand or right hand plane intersects the triangle
      then
        split triangle at intersection and re-tessellate;
        add new triangles to the Selected and Unselected
        submeshes;
    else if the triangle is in the normal half-space of both
      planes then
        add it to Selected submesh;
    else
      add it to the Unselected submesh;

```

---

Once each potential selection object is divided into selected and unselected submeshes, new materials are applied, highlighting the selection and making the rest of the object transparent. If the user has made a previous selection within an object, the algorithm operates only on the previously selected submesh, enabling progressive refinement.

#### 5 CASE STUDY: SELECTING SURFACES IN CULTURAL HERITAGE RECONSTRUCTIONS

In this section, we present a case study exploring how the *Yea Big, Yea High* interface can be applied in a virtual environment. We hope that the design decisions, limitations, and lessons-learned discussed here will inform future researchers designing selection interfaces in the 3DUI community.

The environment for this case study comes from a long-term project analyzing structures from the late Archaic, Classical, and Hellenistic periods (ca. 500-100 BCE) that staged performances of political and legal oratory. Together with a scholar of ancient Greek rhetoric and oratory, we endeavor to use immersive virtual reality to

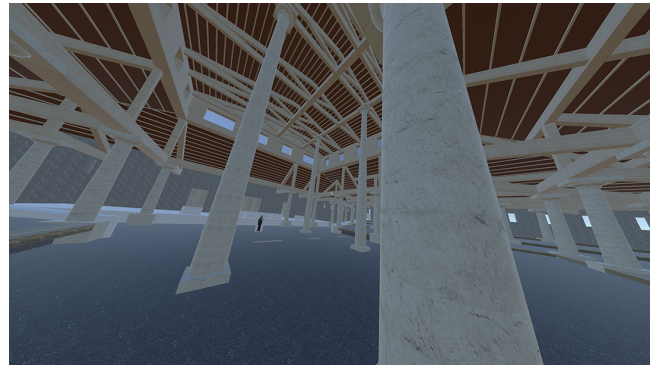


Figure 4: The Thersillion — A reconstruction of a cultural heritage site in Megalopolis. The need to select surface patches to annotate the uncertainty and provenance of inferences related to specific architectural features in the reconstruction motivates the use of *Yea Big, Yea High* selection by our collaborating domain expert.

determine each structure’s suitability as a performance venue and the impact the physical settings had on speakers and audience members.

Unlike other cultural heritage sites previously studied using immersive VR (e.g. [10, 26, 27]), the majority of enclosed spaces in our study were made from wood and largely exist only as ruins today. Using archaeological reports (e.g. [3, 7]) detailing existing architectural features, such as column bases and wall foundations, verified by personal observations, we have generated precise reconstructions of the sites. Despite basing the reconstructions on published reports and making informed inferences, many of the details remain in dispute and have some level of uncertainty. As a result, it is important to be able to annotate pieces of the virtual models to explain the provenance of design decisions.

In this case study, we use *Yea Big, Yea High* as a technique to select specific portions of the virtual reconstructions as a first step in a workflow to annotate specific architectural features. The particular model used is “The Thersillion”, a large, roofed structure nearly the size of an acre in the city of Megalopolis in the Peloponnese. Built in 369 BCE, it was designed to house meetings of the independent city-states in the region.

There are several architectural features to annotate within the model. First, although the columns supporting the roof are very prominent in the reconstruction (See Figure 4), only the rectangular bases and one four foot section of a pillar remain at the site today. Our collaborating domain expert would like to annotate which portion still exists, along with additional notes about its current condition. Selecting a column section is perhaps the easiest interaction with *Yea Big, Yea High* because it only requires one selection action with the hands held horizontally to indicate the top and bottom extents of the selection volume. Note that the design of the interface does allow for several variations on this selection. For example, if a user wanted to select the entire column rather than just a portion, they just need to keep their hands (and therefore the selection planes) on either side of the column, but outside of its boundaries. Because the object being selected is determined through intersections with the activation cube between the hands, the column will be selected, and because it is contained entirely within the selection planes it will be selected in its entirety.

Although the selection planes were designed to be parallel to each other to make the selection a bimanual symmetric technique [23], it is possible to select volumes where the edges are not parallel using the progressive refinement technique. For example, if a user wanted to select a section of a column and have one end of the selection angled to show how it was broken off in present times, they could first select a slightly larger portion with both hands held horizontally,

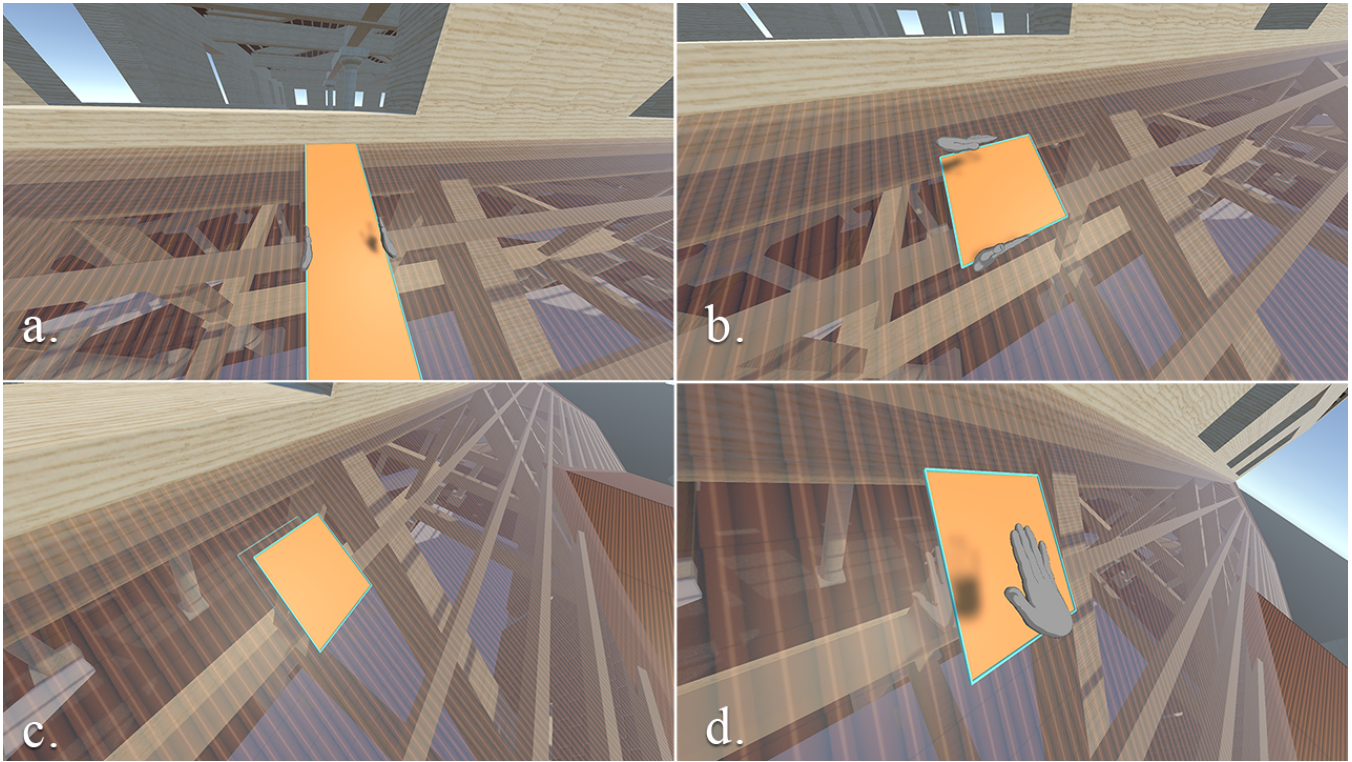


Figure 5: Selection workflow to annotate that a roof tile was found at the archaeological site. (a.) The user starts by selecting a slice through the entire roof. (b.) The original slice is progressively refined to a small square. (c.) Both the top and bottom surface of the roof geometry are selected. (d.) A final refinement selects only the top surface of the tile that is then ready for an annotation.

Then keeping one hand far enough outside of the initial selection that the plane does not constrain it further, they can use the other hand to set the angle. In this way, a user can produce arbitrary selection shapes that don't conform to a cube or even a volume with parallel sides.

In addition to columns, a user might want to annotate information about the architectural elements of the roof. Little remains of the structure's roof today, but based on the spans between columns, the site excavators conjecture that it was likely made of wood covered in tiles [7]. Indeed, a layer of roof tiles was found below the present surface level of the ground. In order to annotate this information, a user might want to select a portion of the virtual roof geometry that is textured with a tile image. As seen in Figure 5, this can easily be accomplished using two progressive refinements. First the user selects a band along the surface of the roof (Figure 5a), followed by reorienting his hands perpendicularly to the first selection to refine the selection to the shape of a single tile (Figure 5b).

This interaction raises an important design decision for the interface; should the selection range take into account depth (i.e. selecting a section all the way through the roof) or just the closest surface to the user constrained by the bounding planes (i.e. just the upper surface of the section). In our implementation, we have chosen to take into account depth because many of the interactions, like selecting a column segment described above, require an entire section through a model, and this case is more general. For selecting objects like a roof tile it makes more sense to only select the closest surface; however, this can be accomplished by just using a third progressive refinement to place one hand above the surface and the other just below it while still inside the volume of the roof depth (Figure 5d). Note, however, that placing ones' hands in the position where one is closer to the body is not as ergonomic and makes it more difficult to judge depth. This is one limitation of the technique that could be

addressed in future work.

Another design decision relates to how to treat objects that are grouped together or stored in a hierarchical scene graph. For the column selection described previously, the column cylinder and the box representing its base are modeled as separate objects, but they are grouped together within the model. Our implementation treats them separately, only selecting from both if the activation cube intersects both objects. But, alternatively, the system could treat all grouped objects as one. If the activation cube intersects any one of the grouped objects, then the cutting planes would be used to intersect with all triangles in each of the group's meshes. In this situation, the resulting selection mesh might also want to group separate selected sub-meshes to keep track of the individual parts.

For models using a scene graph, the idea of grouping could be extended depending on the intended application. One possibility is when the activation cube intersects with a mesh, then the system might choose to operate the selection on all of its child meshes further down in the scene graph. This makes conceptual sense because a child mesh is likely related in some way to the parent mesh. For specific applications, meta-data stored within the scene graph could be used to help refine which objects should be included for a potential selection.

Finally, beyond the design decisions raised by the column and roof selection, the floor is also important within our case study model. The slope and composition of the floor is critical to analysis of audience sight-lines to the speaker. It is likely that the floor was made from wood and sloped towards the center of the structure [7]. In this case, a user might select a thin slice starting at the center of the structure and extending towards the walls to use as a cross section to annotate the slope. This is possible with the interface, but raises the important question of scale, one of the primary limitations of our technique.

## 6 LIMITATIONS AND FURTHER IMPROVEMENTS

The current interface is unable to select a cross section that is wider than the user is able to spread their arms to extend the selection planes. For instance, this would make it impossible to select the floor or an entire side of the building. Selecting out-of-reach objects remains a challenge for most 3DUI selection interfaces that do not involve ray-casting [14]. Nevertheless, it must be addressed before a single selection interface can be used in all virtual environments.

We see two immediate possibilities to overcome this limitation. First, the constraint that the cutting planes pass through the user's hands could be relaxed. In this mode, the distance between the hands could be mapped nonlinearly. As a user opens their hands wider, the selection planes would move further apart at a greater rate. Using this mapping, it would be possible to have arbitrarily large selections, although the individual precision for specifying the boundaries would be less and visual feedback for the planes' positions would become more important. Our initial designs explored this possibility, but we found it difficult to control, and it lacks the advantage of proprioception.

Alternatively, our preferred approach would be to allow a user to adjust the scale of the model, either directly or through a world-in-miniature (WIM) approach [22]. Users could choose between selecting at the life-size scale versus selecting from the WIM for out-of-reach selections. In future work, we would like to explore which of these options is more effective for users making large selections.

A similar problem exists for small scale selections, for example if a user wanted to select only a small subset of triangles from a highly detailed mesh. In this situation, a WIM would enable a user to maintain spatial context while adjusting the scale of the model to zoom in to the selection area.

The decision to make the cutting planes infinite also relates to scale. Because the planes are infinite, a user can select a complete slice through an object. However, this also limits the possible selection volumes to convex shapes. Our future work will explore whether allowing the user to adjust the size of the plane can enable more interesting and complex selections, including ones with concave topology.

Finally, the dependence on flat cutting planes makes it challenging to select subregions within an object where the boundary has an organic or curved shape. These types of selections are possible with the interface, but require many progressive refinements to approximate the shape as a series of small line segments clipping the volume. One way of improving this interaction would be to allow for continuous progressive refinement after an initial selection rather than requiring the user to indicate discrete refinements through button clicks. Using this approach, a user could make a cylindrical selection by initially selecting a slice through a model, and then simply sweeping their hand around in an arc. In the future, we would like to explore these options using real-world scenarios and compare the technique to others by performing user studies.

## 7 EXTENSIONS TO OTHER TYPES OF DATA

Although we have chosen to focus our discussion of the technique on selecting surface meshes, it is generally applicable to different types of data. In this section, we discuss the benefits and limitations of the technique when applied to point clouds and volumetric data.

Point cloud data requires slight modifications to the algorithm described in Section 4. Unlike surface meshes which use the activation cube as an optimization to determine which mesh should be tested for intersection with the cutting planes, these types of data do not need the activation cube. Instead, all points in the point cloud can be tested to determine whether they fall between the cutting planes. Additional data structures would also be needed to store the set of selected and unselected points. Although each point would need to be tested, the process could easily be parallelized on modern graphics hardware because each point is independent of the others.

Similarly, *Yea Big*, *Yea High* can be extended to volumetric data. Assuming the data values are stored in a 3D grid, there are three possible cases: (1) a grid point and its immediate grid neighbors are entirely between the cutting planes, (2) a grid point and its neighbors are entirely outside of the cutting planes, and (3) a grid point is inside the cutting planes but one or more of its neighbors is outside. For case one, the grid point should be added to the selection volume. For case two, the grid point should remain unselected. For case three, the intersection points along the edges of the grid must be calculated in a similar way to how intersections along triangle edges are calculated for surface meshes. These new points and their interpolated data values should be added to the selection volume.

One limitation for both point clouds and volumetric data compared to surface meshes is that areas a user might want to select in these data are unlikely to follow geometric shapes, such as flat planes or rectilinear objects. For instance, imagine selecting the volume of a tumor from medical imaging data. As mentioned previously, the current implementation does not allow a user to select concave shapes, and selecting organic shapes requires many selection refinements which becomes fatiguing and tedious. Although users can relax their arms at their sides between progressive refinements, we have found that the system works best for shapes that can be selected in one to five refinements. As a result, for these types of data the technique is less well suited compared to selecting architectural surface meshes. However, we still see a benefit when combined with context-aware selection approaches. In this situation, the technique can be used to select an initial region of interest. Note, that this region need not be a cube, as in prior work. Using the interface, the user still has the ability to make several progressive refinements to define a more complex initial volume. The selection within that region could then be further refined using context-aware techniques like VolSelectAware [28] or SpaceCast [32]

## 8 CONCLUSION

With the *Yea Big*, *Yea High* technique, we introduce an interface for defining a selection based on a region of interest within a virtual environment. Although we have not yet performed a formal user evaluation, initial users report that the metaphor is intuitive and easy to use. The case study shows that it is already effective for supporting immersive visualizations of cultural heritage models. Our method addresses the critical need in immersive 3D modeling and data analysis to select portions of virtual objects, as well as entire objects without needing to repeatedly select individual points. In this way, we hope that users adopting the interface will more effectively and efficiently be able to work in virtual environments.

## ACKNOWLEDGMENTS

The authors would like to thank Richard Graff for his feedback and domain expertise motivating the cultural heritage application in our case study. Thanks to Azidah Rabbani for her help in developing the architectural models. This research is supported through grants from the Clare Boothe Luce Foundation and a grant from the Macalester College Anderson-Grosshuesch Interdisciplinary Summer Research Fund.

## REFERENCES

- [1] F. Bacim, M. Nabiyouni, and D. A. Bowman. Slice-n-swipe: A free-hand gesture user interface for 3D point cloud annotation. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 185–186, 2014.
- [2] A. Banic. Selection classification for interaction with immersive volumetric visualizations. In *Human Interface and the Management of Information: Information and Knowledge Design and Evaluation*, volume 8521 of *LNCs*, pp. 10–21. Springer, 2014.
- [3] E. Benson. The Thersilion at Megalopolis. *Journal of Hellenic Studies*, 13:319–326, 1892–93.



- [4] J. Cashion, C. Wingrave, and J. J. LaViola Jr. Dense and dynamic 3D selection for game-based virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):634–642, 2012.
- [5] G. de Haan, M. Koutek, and F. H. Post. Intenselect: Using dynamic object rating for assisting 3D object selection. In *Proceedings of the 11th Eurographics Conference on Virtual Environments*, EGVE'05, pp. 201–209. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2005.
- [6] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 95–96. ACM, New York, NY, USA, 1996.
- [7] E. Gardner, W. Loring, G. Richards, and W. Woodhouse. Excavations at Megalopolis 1890-91. *Journal of Hellenic Studies*, Supp. 1, 1892.
- [8] B. Jackson, D. Coffey, and D. F. Keefe. Force Brushes: Progressive Data-Driven Haptic Selection and Filtering for Multi-Variate Flow Visualizations. In *Proceedings of EuroVis 2012*, pp. 7–11. Eurographics Association, Vienna, Austria, 2012.
- [9] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Tech-note: Dynamic dragging for input of 3D trajectories. In *Proceedings of IEEE Symposium on 3D User Interfaces 2008*, pp. 51–54, 2008.
- [10] K. Kim, B. Jackson, I. Karamouzas, M. Adeagbo, S. Guy, R. Graff, and D. Keefe. Bema: A multimodal interface for expert experiential analysis of political assemblies at the pnyx in ancient greece. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 19–26, 2015.
- [11] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3D selection by progressive refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 67–74, 2011.
- [12] J. Liang and M. Green. Jdcad: A highly interactive 3d modeling system. *Computers and Graphics*, 18(4):499–506, 1994.
- [13] J. F. Lucas. Design and evaluation of 3D multiple object selection techniques. Master's thesis, Virginia Polytechnic Institute and State University, 2005.
- [14] D. Mendes, D. Medeiros, M. Sousa, E. Cordeiro, A. Ferreira, and J. A. Jorge. Design and evaluation of a novel out-of-reach selection technique for VR using iterative refinement. *Computers & Graphics*, 67:95–102, 2017.
- [15] M. R. Mine. Virtual environment interaction techniques. Technical report, Chapel Hill, NC, USA, 1995.
- [16] 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*, SIGGRAPH '97, pp. 19–26. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997.
- [17] A. Mossel and C. Koessler. Large scale cut plane: An occlusion management technique for immersive dense 3D reconstructions. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pp. 201–210. ACM, New York, NY, USA, 2016.
- [18] J. S. Pierce, B. C. Stearns, and R. Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pp. 141–145. ACM, 1999.
- [19] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 79–80. ACM, New York, NY, USA, 1996.
- [20] M. Sereno, M. Ammi, T. Isenberg, and L. Besançon. Tangible brush: Performing 3D selection with portable and position-aware devices.
- [21] A. Steed and C. Parker. 3D selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th International Immersive Projection Technology Workshop*, pp. 13–14, 2004.
- [22] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–272. ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, United States, 1995. doi: 10.1145/223904.223938
- [23] A. Ulinski, C. Zambaka, Z. Wartell, P. Goolkasian, and L. F. Hodges. Two handed selection techniques for volumetric data. In *IEEE Symposium on 3D User Interfaces*, pp. 107–114.
- [24] A. C. Ulinski, Z. Wartell, P. Goolkasian, E. A. Suma, and L. F. Hodges. Selection performance based on classes of bimanual actions. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 51–58. IEEE, 2009.
- [25] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *IEEE Symposium on 3D User Interfaces*, pp. 115–122, 2007.
- [26] E. Vote, D. Feliz, D. Laidlaw, and M. Joukowsky. Discovering Petra: archaeological analysis in VR. *Computer Graphics and Applications*, IEEE, 22(5):38 – 50, 2002.
- [27] J. Wall. Virtual paul's cross project: A digital re-creation of john donne's gunpowder day sermon. <http://vpcp.chass.ncsu.edu>, 2013. Accessed: 2018-01-21.
- [28] D. Wilches and A. Banic. Volselectaware: Visual-cognition coupled non-visual data-driven volume selection for immersive scientific visualizations. In *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction*, VINCI '16, pp. 148–149. ACM, New York, NY, USA, 2016.
- [29] G. J. Wills. Selection: 524,288 ways to say “this is interesting”. In *Proceedings IEEE Symposium on Information Visualization*, pp. 54–60, 1996.
- [30] C. Wingrave. 3DUI flavors beyond vanilla. In *Proceedings of the Software Engineering and Architectures for Realtime Interactive Systems Workshop*, SEARIS'09, 2009.
- [31] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2245–2254, Dec. 2012. doi: 10.1109/TVCG.2012.217
- [32] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds. *IEEE transactions on visualization and computer graphics*, 22(1):886–895, 2016.
- [33] W. Zhou, S. Correia, and D. H. Laidlaw. Haptics-Assisted 3D lasso drawing for tracts-of-interest selection in DTI visualization. In *IEEE Visualization Poster Compendium*, 2008.