



Efficient Selection of Multiple Objects on a Large Scale

Stenholt, Rasmus

Published in:

Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology

DOI (link to publication from Publisher):

[10.1145/2407336.2407357](https://doi.org/10.1145/2407336.2407357)

Publication date:

2012

Document Version

Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Stenholt, R. (2012). Efficient Selection of Multiple Objects on a Large Scale. In M. Green, W. Stürzlinger, M. E. Latoschik, & B. Kapralos (Eds.), *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology* (Vol. 2012, pp. 105-112). Association for Computing Machinery (ACM).
<https://doi.org/10.1145/2407336.2407357>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Efficient Selection of Multiple Objects on a Large Scale*

Rasmus Stenholt
Aalborg University
Aalborg
Denmark
rs@create.aau.dk

ABSTRACT

The task of multiple object selection (MOS) in immersive virtual environments is important and still largely unexplored. The difficulty of efficient MOS increases with the number of objects to be selected. E.g. in small-scale MOS, only a few objects need to be simultaneously selected. This may be accomplished by serializing existing single-object selection techniques. In this paper, we explore various MOS tools for large-scale MOS. That is, when the number of objects to be selected are counted in hundreds, or even thousands. This makes serialization of single-object techniques prohibitively time consuming. Instead, we have implemented and tested two of the existing approaches to 3-D MOS, a brush and a lasso, as well as a new technique, a magic wand, which automatically selects objects based on local proximity to other objects. In a formal user evaluation, we have studied how the performance of the MOS tools are affected by the geometric configuration of the objects to be selected. Our studies demonstrate that the performance of MOS techniques is very significantly affected by the geometric scenario facing the user. Furthermore, we demonstrate that a good match between MOS tool shape and the geometric configuration is not always preferable, if the applied tool is complex to use.

Categories and Subject Descriptors

H.5.2 [Information Systems]: Information Interfaces and Presentation>User Interfaces; I.3.6 [Computing Methodologies]: Computer GraphicsMethodology and Techniques

Keywords

Multiple object selection; virtual reality; HMD; user evaluation

1. INTRODUCTION

Selection of multiple objects (MOS) is a frequent goal of user interactions in desktop environments. The prime example of this is the ubiquitous selection box used for picking multiple icons located

*Presented at the VRST '12 in Toronto.

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

VRST'12, December 10–12, 2012, Toronto, Ontario, Canada.
Copyright 2012 ACM 978-1-4503-1469-5/12/12 ...\$15.00.

in a contiguous region on a computer desktop. Many image processing applications, such as Adobe Photoshop or GIMP, include tools for selecting a large number of pixels to be the target of further processing. Furthermore, MOS is very often used in computer games, where the player has to efficiently assign the same commands to several characters or units. All of the mentioned tasks, and the techniques used to accomplish them, can collectively be referred to as 2-D MOS, since they are carried out in a 2-D context.

Once MOS moves beyond the 2-D desktop and into 3-D, the case becomes more complex, and fewer studies exist. The added complexity comes from several sources: More degrees-of-freedom (DoF) to control, a lack of standardized MOS tools, and the possibility of occlusion. The lack of studies on 3-D MOS cannot be explained by a lack of potentially useful applications of 3-D MOS. To illustrate this, consider the following example: In data visualizations, there are often thousands of glyphs or voxels floating around in 3-D space [11], each representing a sample of the visualized database. In many cases, it is useful for a user exploring such a visualization to be able to highlight groups of glyphs or voxels that are of particular interest [4], or to be able to add annotations [6] to specific parts of the visualization. The prerequisite operation for this is a MOS task.

The number of selection targets is an important parameter in this context. We shall refer to this as the *scale* of the MOS task. In some applications, serialization of single-object selection techniques may be adequate and efficient, since the number of target objects is often manageable. We henceforth refer to small-scale MOS as SS-MOS. However, in the case of data mining applications, the scale of the MOS increases drastically, since it is not uncommon to work with databases consisting of thousands of records. Thus, this application is an example of large-scale MOS (LS-MOS). Since serialization of single-object techniques unavoidably becomes more and more impractical to use as the task scale increases - e.g. imagine pointing out 1,000 objects one-by-one - the techniques applicable to LS-MOS tasks are potentially quite different from those applicable to SS-MOS. This makes the technical challenge of designing good LS-MOS techniques interesting. Furthermore, efficient LS-MOS techniques in 3-D are currently relatively unexplored. For these reasons, this paper only deals with MOS tasks, where the scale of the task is at least well into the hundreds.

Thus, the main point of this paper is to test and evaluate the usefulness and performance of different 3-D LS-MOS tools. Furthermore, we seek to evaluate how the geometric layout of the objects to be selected affects the efficiency, precision and ease-of-use of the tools. The results gained through this study are therefore useful to future designers of 3-D LS-MOS toolboxes, since they provide information about the trade-offs made when using the tools. That

is, recommendations of when to use specific tools, and, just as importantly, when to avoid them.

2. RELATED WORK AND MOS THEORY

The tasks that users perform while immersed inside virtual environments are traditionally split into four categories: Selection, manipulation, navigation, and system control. This distinction of categories is e.g. presented by Bowman et al in [2], where a comprehensive design space for 3-D selection techniques is presented. However, the work only mentions single-object selection techniques. The presented design space for selection tasks includes a category for automatic selection, but does not present any examples of such techniques. Although MOS tasks clearly belong within the well-established selection category, 3-D MOS, and in particular 3-D LS-MOS, does not appear to have been the subject of much study. A few exceptions are presented in [8, 16, 17]. For this reason, much of the related work concerns traditional single-object selection.

2.1 Single-Object Selection

For several reasons, single-object selection (SOS) techniques are of interest in the design of 3-D MOS techniques. First of all, any SOS technique has the potential to be used as a MOS technique, if used serially. However, this becomes prohibitively time-consuming as the scale of the MOS task grows. Secondly, many SOS techniques are inherently MOS techniques with an added disambiguation step. This step is included in order to pick just one of the candidate objects that fall within a selection volume. The inclusion of a disambiguation step modifies a MOS technique into a SOS counterpart. As such, many SOS techniques hold additional potential as MOS techniques by removing or modifying the disambiguation step.

Ray casting is one of the most well-established SOS techniques [10, 1]. In ray casting, the selection volume is a half-line or a very narrow cylinder extending from the user's hand infinitely into 3-D space like the beam of a laser pointer. In many cases, the selection ray intersects more than one object, which calls for the inclusion of a disambiguation step. The potential of ray casting in itself as a non-serialized 3-D LS-MOS technique is limited, however, since the selection volume is very small. This issue may be alleviated by using the selection ray to pick out a single object, after which all similar or nearby objects are automatically included in the selection. The similarity/proximity criterion can be arbitrarily chosen. The idea of automatically expanding the selection from a single object forms the basis of the *magic wand* technique which is introduced and evaluated in this paper (see section 3.3).

2.2 Multiple Object Selection

2.2.1 2-D MOS

In 2-D desktop contexts, MOS tasks are very frequent. As such, the techniques for solving the problem in 2-D are well-established. Wills presents a comprehensive design space for 2-D MOS techniques in [20]. In that paper, a distinction between *brush*-type techniques and *lasso*-type techniques is made. This distinction is also valid for 3-D MOS.

Brushes: With a brush, the selection is made inside a persistent object, called a brush, which the user can manipulate in various ways. One typical manipulation is to drag the brush around. It is possible to add more to the selection simply by moving the brush while making an indication to select (e.g. clicking a button). The brush-metaphor is very clear and intuitive, because dragging the virtual brush around is very similar to painting with a real brush.

It is fairly straightforward to adapt the idea of a selection brush to 3-D MOS, since all that is needed is to 1) use a 3-D shape instead of 2-D shape for the brush and 2) map the controls of the brush to a motion-tracked 3-D interaction device. In this paper, we have chosen to use a spherical 3-D brush with adjustable radius as one of the three evaluated MOS techniques.

Lassos: The lasso category of selection techniques is based on the user defining a temporary selection shape, called a lasso. All objects that lie within the created lasso are selected, after which the lasso disappears. If the user wishes to expand the selection, more lassos must be defined. In 2-D, a lasso can be made in several ways. Desktop selection rectangles are a well-known example of such a method. In other applications, lassos are made by tracing out a closed, free-form contour on the screen using a mouse.

The lasso concept can also be adapted for use in 3-D. However, the case is not as straightforward as with the brush, because the steps involved in creating the actual lasso are non-trivial to convert from 2-D. First of all, the lasso shape must be a closed 3-D volume, or at least an infinite extrusion of a 2-D shape. This means that the desired shape of a 3-D lasso is probably the primary design choice to make. It only becomes possible to design an efficient way of producing the lasso, after deciding on the class of 3-D shape. Efficient creation of 3-D shapes, specifically 3-D boxes - the natural 3-D extension of 2-D rectangles, is discussed in [17, 15]. A box-shaped lasso, created according to the 3C technique introduced in [15] is the second 3D LS-MOS technique evaluated in this paper.

Automatic techniques: The work of Wills [20] does not account for the possibility of automating part of the MOS process, although the proposed design parameters do include the possibility of modifying the selection volume based on the objects inside of it. Nonetheless, automatic 2-D MOS selection tools are very common in desktop applications, especially in image processing. Such tools use region growing algorithms to expand a selection to all similar pixels connected to an initial seed pixel indicated by the user. The similarity of pixels is typically based on their RGB values. This is the approach which we have chosen to evaluate as the third way of performing MOS tasks. In Adobe Photoshop, the automatic selection tool is named a magic wand. This has inspired us to use the same name for our automatic tool in order to give users a metaphor to relate to. However, the region growing method underlying our magic wand is somewhat different from the one found in Photoshop. See section 3.3 for more details. A study based on automatic 2-D group selection based on human perception is found in [3]. This work has also served as inspiration.

2.2.2 3-D MOS

Few studies directly concerning 3-D MOS currently exist. In one of the currently most comprehensive studies on the subject, Lucas et al [8] presents a design space for 3-D MOS techniques. This design space identifies 6 parameters to consider in 3-D MOS design. Subsequently, variations of two of these parameters, concurrency and spatial context, are used in designing and evaluating four different 3-D MOS techniques. One technique was serialized ray casting. Two of the techniques were performed through a 2-D view of the scene on a hand-held tablet, and as such, are more or less clones of well known desktop MOS techniques. The final technique featured a persistent selection box, i.e. a box-shaped brush, which could be scaled, rotated, and positioned arbitrarily through a combination of two techniques: Go-Go [13] and PORT [9]. The box selection technique is the only one of the four techniques tested by Lucas et al which is both compatible with the LS-MOS context of this paper, and with the non-see-through HMD setup used. The number of target objects was in the range from 9 to 64, thus stay-

ing reasonably within the realm of SS-MOS, where serialization of SOS techniques is viable. The objects were laid out in a non-randomized, rectangular grid pattern in all cases. Furthermore, the selection task was visualized on a projection screen. These facts differentiate the study of Lucas et al from the one in this paper.

In the work of Ulinski et al [17], the subject of study is MOS using box lassos of arbitrary position, orientation, and scale. Several techniques for creating the lassos were evaluated wrt. two binary factors, density and occlusion. The studies were carried out by users interacting with a monoscopic rendering of the data displayed on a table-top monitor. The spatial layout of the target objects were box-shaped in all cases, meaning that no test of the influence of layout, beyond the mentioned factors, was made. I.e. there were no cases where the target objects' spatial layout did not match the shape of the lasso. No quantitative information about the scale of the task is given. Thus, our work differentiates itself from Ulinski et al's by virtue of testing several different spatial layouts of target objects, by testing other tools than 3-D box lassos, and by using an immersive display such as an HMD.

The case of automatic 3-D MOS has also been considered previously. However, it seems that existing techniques rely on the existence of a relational data structure, e.g. a scene graph, behind the rendered virtual objects. This for example means that selecting a virtual table also selects all the objects resting on the table. Such techniques are e.g. presented in [12, 7, 21]. These methods seem very attractive for virtual worlds populated by recognizable objects such as architectural visualizations or computer games. However, these techniques are not applicable in the general case, where the visualized objects come from a database without natural structures such as parent-child relations. This is frequently the case in abstract data visualizations.

3. DESIGN OF 3-D MOS TECHNIQUES

In this section, we present the specific design used for the three tested MOS techniques. Two of these, the brush and the lasso, relate directly to some of the studies mentioned above. The magic wand, however, is new in a 3-D MOS context. Therefore, the algorithm used in the implementation of the automatic aspects of the wand is of particular interest. The choice of the control mappings in all techniques is heavily based on the available controllers: Two wireless presenter mice fitted with markers to make them trackable with 6 DoF. This implies that the available controls on the devices apart from the motion tracking are three buttons (left, middle, right) and a scroll wheel. The mice are shown in Figure 1. The designed techniques take advantage of having two controllers either by being bimanual techniques, or by duplicating the same technique to both hands, giving the users a free choice of which hand to use. Furthermore, the two primary concerns addressed as design criteria are that the techniques must be usable in an HMD context, and that they must be applicable as 3-D LS-MOS techniques.

3.1 Brush

The two primary decisions to make when designing a 3-D brush is the shape of the brush and its control mappings.

In this study, we have chosen to use a spherical brush. The main motivations for this choice are twofold. First of all, a sphere is a well-known shape which should be easy to control for the average user. A spherical brush has a maximum of 4 DoF, i.e. a 3-D position and a radius. This fact is important to this study, where the time available for familiarizing oneself with the tool is fairly limited. Secondly, a spherical brush with adjustable radius should be applicable and relatively precise in many contexts.



Figure 1: The mice used in the experiment. Each mouse is fitted with reflective markers to make it detectable by an infrared camera system. The mice are equipped with the three usual mouse buttons as well as a scroll wheel.

The main disadvantage is expected to be that it will be difficult to select targets located at the edges of a non-spherical volume, e.g. a box shape, where clutter is present close to the targets. E.g. it is difficult to use a sphere to select target objects in the corner of a box surrounded by undesired objects (clutter), without also selecting some of the clutter.

The choice of a spherical shape also has direct implications on the control scheme used. Since the sphere is a closed volume in space, unlike a ray or a cone, the actual brushing process becomes local and direct. I.e. the users must place the brush at the desired targets before doing the brushing. We decided to go with a sphere and the resulting control scheme, because a cone is impractical in scenarios where clutter is obstructing a clear path of pointing. This is the case in several of the scenarios we have tested.

For the same reasons, we have also chosen to go with the same control scheme for the other MOS techniques. This not only has the mentioned benefit of working well in cluttered scenarios, it also reduces the amount of different interaction schemes that the test subjects will have to learn. Furthermore, it puts the tested techniques on even ground in the evaluation, meaning that the control scheme is not a factor of the study.

Therefore, the controls are a direct mapping of the positions of the user's two hands to the positions of two virtual brush spheres, based on the information provided by a motion tracking system. The brush spheres are made semi-transparent in order to allow better perception of the objects inside the brush. This design choice is in line with previous work by Zhai et al [22]. The adjustable radius is controlled by scrolling the wheel on the controller. Selecting the objects currently inside the brush is mapped to the left buttons. That is, clicking the left button once selects all objects inside the brush, while holding the left button allows the user to keep selecting objects while sweeping the brush through the volume of objects. The brushes in action are shown in Figure 2.A and Figure 2.E.

In order to allow the user to correct mistakes, clicking the right button toggles from the basic selection mode to a deselection mode, where the brushing removes objects from the selection set. The current mode of operation is indicated by the colour of the brushes. This mode toggling feature is also available for the lasso, whereas deselection works differently with the magic wand.

The user can select or deselect objects by repeating the process as many times as desired. Thus, the designed techniques are examples of parallel 3-D MOS tools in the terminology of [8], which can be used in a serial fashion to refine or expand a selection.

3.2 Lasso

For the lasso metaphor, we chose to use a 3-D version of the well-known 2-D selection box lasso from the desktop. Since an arbitrary 3-D box has 9 DoF, the construction technique for the lasso has to comply with this fact. Control of all 9 DoF is needed

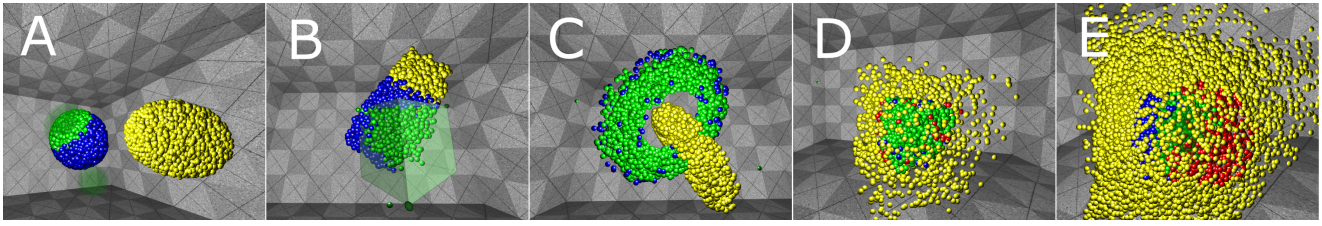


Figure 2: Screenshots of various experimental conditions and techniques. Blue glyphs are targets, yellow ones are clutter. Correctly selected glyphs are highlighted in green, incorrect ones in red. (A) The separated clouds scenario with the 3-D spherical brush shown as faint, semi-transparent green spheres. (B) The adjacent clouds scenario with the 3-D box lasso technique in use. (C) The entangled clouds with the magic wand technique in use. The magic wand has been used once to select most of the blue cloud without adjusting its proximity threshold. (D) The embedded nucleus scenario with the magic wand as selection tool. As in part C, the wand has been used once here to make a selection. A few false positives are showing as red glyphs. (E) The uniform embedding scenario with a large 3-D brush in use. This scenario is difficult due to the amount of visual occlusion, and the proximity of the clutter and target glyphs.

to allow for full control of the flexibility offered by the box shape. This makes the technique somewhat more difficult to use than the 4 DoF spherical brush. However, it also has the potential to be a better match for the presented cloud of target objects, if the cloud has angular corners or planar surfaces. Controlling 9 DoF in an efficient manner is non-trivial. For this reason, we have chosen to implement the existing 3C technique, which provides the greatest degree of precision according to [15].

The 3C technique requires the user to indicate the positions of 3 corners on the desired box in a specific pattern. We have chosen to go with the variant of 3C named 3+3+3 DoF 3C in [15]. This suits the direct interaction pattern chosen for the brush technique well, since it is simple to point out specific points in space, in this case box corners, in a direct way. I.e. the user has to place a hand at a desired corner position, after which a click places the corner there. The hand positions are indicated by small spheres. After making three clicks, the lasso is completed, and all objects inside the lasso are selected. The user can interactively see the lasso currently resulting from the positions of the hands as soon as the first lasso corner has been placed. As was the case with the brush, the lassos are rendered as semi-transparent boxes. An example of a box lasso in use is shown in Figure 2.B.

As was the case with the brush, the user has the option to right-click in order to toggle to a deselection mode, where the boxes remove objects from the selection set. Similarly, the user is allowed to make as many selection/deselection boxes as desired to reach the end result.

3.3 Magic Wand

The magic wand is based on its namesake technique used in 2-D image processing applications. The main idea is that the user indicates a single object, the seed, which is representative of the objects in the desired selection. An algorithm then takes care of expanding the selection from the seed to all objects which are similar enough to the seed according to some criterion. In general terms, any selection technique, SOS or MOS - automatic or manual, can be viewed as a binary classification task, where the objects of the scene are split into two clusters, the target and the clutter. In the special case of SOS, the target cluster only contains one object. Thus, the type of algorithm needed for the magic wand belongs to the category of clustering algorithms. Many clustering algorithms are based on knowing the expected distribution of the objects to be clustered. However, in the general case of MOS, nothing is known a priori about such distributions. I.e. the spatial layout of the objects varies much from case to case. Therefore, we have aimed to

design a magic wand algorithm which does not make any assumptions about the overall shape of the clouds to be selected.

Instead, the assumption that we make is based on human perception. People tend to think of densely packed groups of objects as a whole instead of individual constituents. This is a well-established fact in the gestalt laws of human visual perception [19]. The gestalt law of proximity is of special interest here, since it provides a proven theoretical background to base the automatic techniques on. As such, our magic wand technique is designed to select all objects that feature high proximity to other objects in the cluster, and ultimately to the seed. Furthermore, to be applicable as an interaction technique, the clustering algorithm should not be so computationally expensive that it prevents the clustering from happening in real-time with the scales of MOS tasks used in our study.

Such a procedure is e.g. presented as the initial grouping step of the clustering algorithm introduced in [5]. In our case, where we want to base the algorithm on the gestalt law of proximity, the closeness of cluster members is based on Euclidean distance in 3-D. More dimensions can also be taken into account to make the clustering sensitive to other cues than spatial proximity (e.g. colour or shape), corresponding to the gestalt law of similarity.

It is worth noting that the use of Euclidean distance as proximity criterion does not necessarily lead to spherical clusters: When viewed on a global level, it is not the Euclidean distance from the seed to the other members of the cluster that matters. Instead, all members have in common that there is at least one other member located somewhere within local proximity. I.e. clusters can have members very far from the seed, as long as there exists a path of sufficiently small jumps through the cluster from the seed to the distant cluster members. This principle maintains the idea that the selected objects should be in a region of similar density/proximity between the objects. This is illustrated in Figure 3.

Mapping the above to user controls has the following implications: 1) The user must be able to select the seed object. 2) The user must be able to adjust the proximity threshold used when deciding which objects belong to the selection set. Pointing out the seed is done directly using a small, spherical cursor - just like the indication of the corners of the 3-D box lasso. Furthermore, adjustment of the maximum allowed distance between cluster members and their closest neighbour in the cluster is mapped to the scroll wheel. This means that the magic wand is a 4 DoF tool. Increasing the threshold value has the effect of lowering the requirements of cluster membership, i.e. expanding the selection. The opposite is true when decreasing the threshold value. At the minimum value of 0, only the seed is admitted into the cluster, effectively making

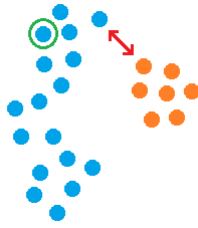


Figure 3: Example of two perceptually distinct clusters that can be individually selected using the proposed clustering method. Selecting a seed (green circle) in one cluster will not cause a selection of objects in the other cluster because the internal distances in the cluster containing the seed are smaller than the minimum distance separating the two clusters (red arrow). The other cluster is also selectable in the same way for the same reasons.

the magic wand a simple SOS tool. Figure 2.C and 2.D show the the magic wand being used.

Thus, selection and deselection with the magic wand happens when the scroll wheel is used. As with the brush and the lasso, the user is allowed to add to the selection by adding new seeds. This implies that in some difficult cases, the user has the option to keep the proximity threshold at a low level, in order to serially add new seeds with small clusters to the selection set without much risk of accidentally selecting any clutter.

There are many possible modifications to the basic magic wand technique. Some examples are: Including an upper limit on the size of the selection set, limiting the number of allowed jumps from the seed, increasing the linkage requirement from a single neighbour to more neighbours, etc. However, in this study we have chosen to go with the basic scheme as outlined above.

4. EXPERIMENT

With the three MOS techniques outlined and motivated, the performed experiment can be explained.

4.1 Geometric Scenarios

The main purpose of the experiment is to test how the three chosen MOS techniques fare in scenarios of different geometric layout. In choosing scenarios, we have aimed to include some that represent a broad range of all possible scenarios, and to design the scenarios such that they potentially highlight the strengths and weaknesses of each technique. As such, scenarios encountered in real applications should conceptually match a combination of those tested in this study. A geometric scenario consists of two randomized clouds of glyphs: A target cloud rendered as blue glyphs, and clutter rendered in yellow. The glyphs temporarily change colours to green (target) or red (clutter) upon entering a selection volume, and permanently so upon being committed to the selection set. Examples of all 5 chosen scenarios are shown in Figure 2.

The randomization is performed such that the overall shape of the clouds adheres to the design of the scenario. Both clouds are typically created with an average density of 50,000 glyphs/m³, which makes it straightforward to perceive the overall shape of the clouds from the visualized glyphs. The chosen density implied that the mean number of target glyphs was approx. 1700, while the mean amount of clutter was app. 3300. The glyphs were rendered as Phong-shaded, spherical point sprites in the experiment. The goal of the user task was to select all target glyphs while avoid-

ing selection of clutter. The best possible result, i.e. one where all targets are selected without any of the clutter being selected, is referred to as a perfect selection.

4.1.1 Separated Clouds (SC)

The separated clouds scenario is the baseline best-case scenario, which is expected to be straightforward for all techniques. It consists of two spherical clouds, which are spatially well-separated. As such it should be simple to make a perfect selection no matter which technique is used.

4.1.2 Adjacent Clouds (AC)

In the adjacent clouds scenario, two box-shaped clouds are placed adjacent to each other with only a very small separation distance. This layout poses considerable difficulty for the magic wand, because the local proximity criterion can easily cause the clustering algorithm to bridge the gap between the two clouds. Furthermore, the spherical brush can be difficult to use in the region close to the boundary between the two clouds without accidentally selecting some of the clutter. The box lasso shape is a perfect fit for this scenario.

4.1.3 Entangled Clouds (EC)

In this scenario, the clouds are shaped as two tori. The tori are oriented such that the plane of one is perpendicular to that of the other. Furthermore, they are offset from each other such that one runs through the centre of the other. The two tori are well separated everywhere. This is a perfect case for the magic wand, since the local proximity criterion allows the clustering to walk all around one torus without ever jumping to the other one. Using the brush is also expected to be somewhat straightforward. The entangled clouds are problematic with box shaped lassos. It is impossible to achieve perfect selection using a single box, since any volume completely containing one of the tori also contains part of the other torus.

4.1.4 Embedded Nucleus (EN)

In the target nucleus scenario, the target is presented as a dense nucleus completely embedded in a more sparse cloud of clutter. There is no clutter inside the volume spanned by the target nucleus. Both the targets and the clutter are box-shaped. This scenario is expected to be straightforward with the magic wand, since local proximity essentially is the same as density. This allows the automatic selection to reach good results with only a few false positives within a short amount of time. The brush will probably have problems avoiding false positives, and will likely begin to suffer from occlusion problems, where the clutter gets in the way of properly controlling the brush. The box lasso perfectly matches the task, however, visual occlusion problems may degrade its precision.

4.1.5 Uniform Embedding (UE)

The UE scenario is almost identical to the EN scenario. The only difference is that the clutter and the target are equally dense. The UE scenario is a worst-case scenario. The only possible worse situation would be the case, where the target and clutter volumes are overlapping in space. None of the techniques are expected to perform well here, however, a uniform embedding is a particularly challenging scenario for the magic wand and the brush. The box lasso has the potential to make a perfect selection in just one selection operation, however, this requires that the user is not hindered too much by the amount of visual occlusion present.

4.2 Design Matrix

Having 5 geometric scenarios and 3 different MOS techniques, produces a total of 15 different conditions to make up the 2-factorial randomized complete block design, which we have chosen to use. All conditions are outlined in Table 1.

Table 1: All combinations of the two factors of the experiment. The numbers in the table will be used to refer to the specific combinations of the two factors in the analysis of the results. The (+), (.) and (-) labels indicate if the technique is expected to perform well (+), average (.), or badly (-) in the given scenario.

Technique	Scenario				
	SC	AC	EC	EN	UE
Lasso	0 (+)	1 (+)	2 (-)	3 (+)	4 (-)
Brush	5 (+)	6 (.)	7 (+)	8 (.)	9 (-)
Wand	10 (+)	11 (-)	12 (+)	13 (+)	14 (-)

4.3 Hypotheses

Based on the preceding analysis of MOS techniques, and the presentation of the scenarios above, the hypotheses of the experiment are:

- H1 The combination of MOS tool and geometric scenario significantly affects the selection performance.
- H2 The box lasso is better than the other techniques in the adjacent clouds (AC) scenario.
- H3 The magic wand is better than the other techniques in the entangled clouds (EC) and the embedded nucleus (EN) scenarios.
- H4 Overall, the magic wand is faster to use than the other techniques.
- H5 Overall, the brush is easier to use than the other techniques.

4.4 Response Variables

The notion of performance can be measured in several ways for a MOS task. There are three main categories of response variables, which are relevant to the hypotheses: Completion time, selection quality, and ease-of-use.

The most basic approach to measuring MOS performance is to count the number of true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP). The goal of any MOS task is to maximize the TP and TN counts while minimizing the FP and FN counts. Instead of using the raw counts, we have chosen to use the scale independent quantities of sensitivity and specificity. The sensitivity is the amount of targets selected out of the total number of targets. Conversely, the specificity expresses how much of the clutter has been correctly avoided relative to the total amount clutter. As such, the perfect solution to a MOS task reaches 100% sensitivity while maintaining 100% specificity.

The concept of ease-of-use is a subjective assessment. This assessment can be given in a multitude of ways, e.g. through post-test questionnaires, structured interviews, informal discussions, etc. In order to facilitate statistical analysis along with the other response variables, we have chosen a quantitative approach, where participants are asked to subjectively quantify their perception of task difficulty on a discrete 1 (trivial) to 10 (impossible) scale after completing the trials of each test condition. The specific question asked was “How difficult do you think it was to solve the task well?”. To

supplement the subjective measurement, we also decided to count the number of operations used in each trial to get an objective measurement of the ease-of-use. Operations were counted through the number of selection/deselection indications (i.e. lasso completions, brush strokes, or magic wand seed selections), and adjustments made on the scroll wheels.

4.5 Experimental Procedure & Equipment

The experiment was run on an Intel Core i7-2600 3.4 GHz PC with 8 GB of memory and an nVidia GeForce GTX 590 graphics card. The experimental software was a custom made OpenGL renderer running under 64-bit Microsoft Windows 7. For head and hand tracking, a 24 camera OptiTrack system was used, which allowed unrestricted user motion inside a 2.25 m radius. The HMD was an nVisor SX111 featuring a $102 \times 64^\circ$ total field-of-view at a resolution of 1280×1024 pixels per eye. Two wireless presenter mice fitted with trackable markers were used as interaction devices (see Figure 1).

In the experiment, the participants first received an introduction to the equipment. Then a few basic, demographic questions (age, gender, 3DUI experience) followed. All participants were instructed that they could stop the experiment at any time, and that they could request breaks. A total of 18 people, 16 males and 2 females, participated in the study. All of the participants were recruited among local university staff and students. The mean age was 27 years ($\sigma = 6.55$ years), and the median self-reported 3DUI experience level on a 1 (novice) to 5 (expert) scale was 3. No payment was offered, apart from some light refreshments during the experiment.

After donning the HMD, the participants were presented with a few practice scenarios to familiarize themselves with all of the techniques and controls. These practice scenarios were all of the SC and AC types. Once the subjects were comfortable with all three techniques, the experiment commenced. The participants were instructed to select all of the blue glyphs, and to avoid selecting the yellow glyphs *as well as possible* in all scenarios. Not requiring perfect selections was a necessity, since far from all combinations of techniques and scenarios would be perfectly solvable, at least within reasonable time. Furthermore, this approach means that the quality parameters of sensitivity and specificity become meaningful quantities to measure, since they are not always at 100%. The subjects were instructed to let the experimenter know as soon as they felt that they were done with a trial. The experimenter would then press a button, the completion time would be logged, and the experiment proceeded to the next trial.

The sequence of the test conditions was randomized for each subject to counterbalance any effects caused by the sequence. Furthermore, all test conditions were repeated 3 times for all subjects. Thus, each subject went through a total of 45 trials during the experiment, which was doable for most subjects in less than an hour. At the end of the experiment, an informal debriefing was made.

5. RESULTS

5.1 Analysis of Hypotheses

All analyses have been made using the statistical software package R [14] using a significance level of $\alpha = 0.05$. Before doing a pre-analysis of the data, three extra response variables were computed from ratios of the directly measured responses. The three ratios were: 1) the sensitivity/operation count ratio, 2) the operation count/completion time ratio, and 3) the sensitivity/completion time ratio. These ratios provide new insights, i.e. 1) the amount of sensitivity gained per operation performed, 2) the speed of each opera-

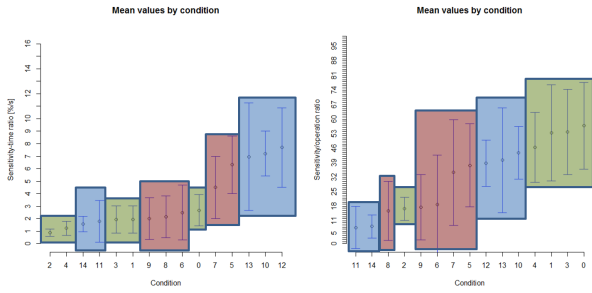


Figure 4: Illustration of mean values and standard deviations for each experimental condition. The coloured boxes show which technique each condition belongs to. (Left) The sensitivity gained per second. Here, the magic wand (blue) performs well, except in the AC and UE scenarios. Overall, the brush (red) is slower than the wand, while the box lasso (green) is slowest. (Right) The sensitivity gained per operation. The box lasso performs well, while the magic wand shows a bimodal tendency. The brush technique requires the most operations, implying that many brush strokes are needed to get the desired results.

tion, and 3) the speed of sensitivity gain. As such, these ratios represent the extent to which selection quality, ease-of-use, and speed are traded off. We have included these variables, because they highlight several important differences between the techniques.

A pre-analysis of the data was carried out to check if it conformed to the requirements of analysis of variance (ANOVA). The assumption of independence is satisfied by the fact that independent subjects were used. The normality of the residuals was checked using quantile-quantile (Q-Q) plots. No big problems were detected, however, the sensitivity, specificity, and operation count responses all have some outlier values in their Q-Q plots. The homoscedasticity requirement was not fulfilled for any of the response variables, and attempts to correct this through the use of variance stabilizing transformations failed. We did, however, observe that a logarithm transformation on all of the response variables involving completion time, made the histograms conform much better to a normal distribution bell curve. Therefore, we have used log transformations on all completion time-dependent variables. To be sure that we are not reaching the wrong conclusions, we verified that a non-parametric Friedman test reaches the same conclusions as the performed ANOVA. For this reason, we chose to use ANOVA in spite of the violation of a basic ANOVA assumption. All pairwise post-hoc tests were carried out using the Tukey’s honest significant difference method. Some of the results are illustrated in Figure 4.

5.1.1 Hypothesis 1

The hypothesis H1 stated that there would be a significant effect of the combination of MOS tool and geometric scenario. This hypothesis is unambiguously supported no matter which response variable is considered. The p -values are all $\ll 0.001$. This means that the choice of MOS tool makes a big difference depending on the geometric scenario facing the user, both in terms of the quality of selection, completion time, and the subjective judgment of ease-of-use.

5.1.2 Hypothesis 2

H2 hypothesized that the box lasso was the best tool to use in the AC scenario. This means that test conditions 1 (lasso), 6 (brush), and 11 (wand) must be compared. The performance of the box

lasso in the AC scenario relative to the other techniques depends on the response variable chosen. Wrt. subjective difficulty, the lasso is deemed significantly easier to use than the magic wand ($p < 0.001$), but there is no significant difference between the brush and the lasso. With respect to completion time, there are no significant differences among the techniques. In the case of sensitivity, the lasso does outperform the magic wand ($p < 0.001$), but not the brush. The same result is the case wrt. specificity. If the tradeoff ratios are considered, no significant difference is found wrt. the sensitivity achieved per time unit. However, if the sensitivity achieved per operation is considered, the lasso is superior to both the brush and the magic wand (both $p < 0.001$). I.e. the lasso is very good in the adjacent scenario, if judged by the quality it achieves relative to the number of lassos that you have to use. The fact that the lasso is not performing any better is surprising, or at least contrary to current speculation on the topic, e.g. on pp. 20-21 of [18] where it is stated that it is desirable to use a flexible MOS shape that fits the shape of the targets. It is likely, however, that extensive training in the usage of the complex tool modifies this result. The main impact of this result is that users may actually *not* subjectively or objectively prefer to use a tool which is perfectly shaped for the job, *if* the tool is too complex to use compared to using a simpler tool multiple times.

5.1.3 Hypothesis 3

This hypothesis deals with the performance of the magic wand in the EC and EN scenarios. The test conditions of interest are 12 and 13 (wand) compared to 2 and 3 (lasso) and 7 and 8 (brush). Specifically, H3 hypothesizes that the magic wand will outperform the other techniques in those scenarios. With respect to subjective difficulty, the magic wand is better than the lasso ($p < 0.001$ and $p = 0.013$). However, there is no significant difference between the wand and the brush. The response variable that really sets the wand apart from the other two techniques is completion time, where the wand is significantly faster to use than any of the others (p -values in the range from 0.023 to < 0.001). There is not much difference wrt. sensitivity and specificity, the only significant result being that the wand reaches significantly higher sensitivity in the EN scenario. If the sensitivity gained per time unit is considered, the wand is significantly better than the other two techniques ($p < 0.001$), except when compared to the brush in the EC scenario. The overall conclusion wrt. H3 is therefore that in EC and EN scenarios, the wand mainly outperforms the other techniques wrt. speed, but in terms of selection quality, all techniques achieve similar results.

5.1.4 Hypothesis 4

H4 states that the magic wand is faster than the other techniques in general. The results of H3 already supports H4. Inspecting the completion time response variable across all geometric scenarios, reveals that H4 is supported. The magic wand is significantly faster than the brush ($p < 0.001$), which in turn is significantly faster than the lasso ($p < 0.001$). This is also true, if the amount of sensitivity gained per second using the wand is considered. Here, the wand is significantly better than the brush ($p = 0.020$) and the lasso ($p < 0.001$).

5.1.5 Hypothesis 5

In the final hypothesis, we conjectured that the brush would be judged to be easier to use than the other techniques, viewed across all tested scenarios. The motivation of H5 being true was that the brush featured fewer DoF than the lasso (4 vs. 9), which provides for easier control. At the same time, some of the scenarios were designed to be nearly impossible to do well using the magic wand.

Thus, the brush has potential to be a jack-of-all-trades tool, which users would find easy to use. H5 is supported by the data, based on the subjective ratings of difficulty. Thus, the brush is evaluated to be easier to use than the other two tested techniques, $p = 0.026$ (wand) and $p < 0.001$ (lasso). However, if we evaluate ease-of-use in terms of the number of operations needed to accomplish the tasks, then the picture very different. In terms of number of operations, the box lasso uses significantly fewer operations than any of the other two techniques, $p = 0.0026$ (brush) and $p < 0.001$ (wand). This provides even more evidence that users prefer to use a simple tool many times rather than using a better fitting, complex tool a few times.

6. CONCLUSION & PERSPECTIVES

In this paper, we have presented the following contributions in relation to 3-D MOS tasks. We have presented a thorough analysis of the field of 3-D MOS, including the distinction between the requirements of small and large-scale MOS tasks. We have presented and evaluated a new technique, the magic wand, for partially automating 3-D LS-MOS tasks. Furthermore, we have made a rigorous experiment demonstrating that:

1. Tool efficiency is very geometry dependent. The best 3-D MOS approach in future applications must be to include an ensemble of complementary MOS tools.
2. The use of a 3-D magic wand is a very fast technique, but also very sensitive to the geometric scenario, making it either very easy or completely impractical to use.
3. The natural 3-D extension of the 2-D rectangular lasso is not preferred by participants over simpler techniques, even when the simpler options are less suitable for the geometric scenario.
4. The 3-D spherical brush is a good candidate for a simple, general 3-D MOS tool applicable to many scenarios.

Overall, we believe that the results of this experiment should be of interest to any future investigators of 3-D MOS tasks, especially those performed on a large scale, i.e. with too many objects to make serial single-object selection practical.

7. REFERENCES

- [1] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 35–ff., New York, NY, USA, 1997. ACM.
- [2] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley, first edition, 2005.
- [3] H. Dehmeshki and W. Stuerzlinger. Design and evaluation of a perceptual-based object group selection technique. In *Proceedings of the 24th BCS Interaction Specialist Group Conference*, BCS '10, pages 365–373, Swinton, UK, UK, 2010. British Computer Society.
- [4] R. Fuchs, V. Welker, and J. Hornegger. Non-convex polyhedral volume of interest selection. *Computerized Medical Imaging and Graphics*, 34(2):105 – 113, 2010.
- [5] E. Gokcay and J. C. Principe. Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:158–171, 2002.
- [6] R. Harmon, W. Patterson, W. Ribarsky, and J. Bolter. The virtual annotation system. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 239–245, 270, mar-3 apr 1996.
- [7] J. Jang and J. R. Rossignac. Multiple object selection in pattern hierarchies. Technical report, Georgia Institute of Technology, 2007.
- [8] J. F. Lucas. Design and evaluation of 3d multiple object selection techniques. Master's thesis, Virginia Polytechnic Institute and State University, 2005.
- [9] J. F. Lucas, J.-S. Kim, and D. A. Bowman. Resizing beyond widgets: object resizing techniques for immersive virtual environments. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, pages 1601–1604, New York, NY, USA, 2005. ACM.
- [10] M. Mine. Virtual environment interaction techniques. Technical report, UNC Chapel Hill CS Dept, 1995.
- [11] H. R. Nagel. *Exploratory Visual Data Mining in Spatio-Temporal Virtual Reality*. PhD thesis, Faculty of Engineering and Science, Aalborg University, 2005.
- [12] J.-Y. Oh, W. Stuerzlinger, and D. Dadgari. Group selection techniques for efficient 3d modeling. *3D User Interfaces*, 0:95–102, 2006.
- [13] 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, pages 79–80, New York, NY, USA, 1996. ACM.
- [14] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
- [15] R. Stenholt and C. B. Madsen. Shaping 3-d boxes: A full 9-degree-of-freedom docking experiment. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 103–110, March 2011.
- [16] W. Stuerzlinger and G. Smith. Efficient manipulation of object groups in virtual environments. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 251–258, 2002.
- [17] A. Ulinski, C. Zambaka, Z. Wartell, P. Goolkasian, and L. Hodges. Two handed selection techniques for volumetric data. *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, mar. 2007.
- [18] A. C. Ulinski. *Taxonomy and experimental evaluation of two-handed selection techniques for volumetric data*. PhD thesis, University of North Carolina at Charlotte, Charlotte, NC, USA, 2008. Adviser-Hodges, Larry F.
- [19] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, second edition, 2004.
- [20] G. Wills. Selection: 524,288 ways to say "this is interesting". In *Information Visualization '96, Proceedings IEEE Symposium on*, pages 54–60, 120, oct 1996.
- [21] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.
- [22] S. Zhai, W. Buxton, and P. Milgram. The "silk cursor": investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 459–464, New York, NY, USA, 1994. ACM.