ICE-Lasso: An Enhanced Form Of Lasso Selection

Hoda Dehmeshki
Computer Science and Engineering Department
York University, Toronto, Canada
Email: hoda@cse.yorku.ca

Wolfgang Stuerzlinger
Computer Science and Engineering Department
York University,Toronto, Canada
Email:http://www.cse.yorku.ca/∼ Wolfgang

***Abstract*—Lasso selection tends to be inefficient in many circumstances such as selecting spatially large clusters. ICE-Lasso is a novel technique that infers likely target clusters during an ongoing lasso gesture. It provides efficient gesture-based interaction techniques as shortcuts to select partial, complete, and multiple clusters. Additionally, it is overloaded on the traditional lasso with an *automatic* mode switching. A comparison user study show that ICE-Lasso is significantly more efficient than lasso and also well-liked by users.**

I. Introduction

Selection of a group of objects is a common task in graphical user interfaces and is required for many operations such as deletion, movement, or modification.

Lasso is a standard group selection technique in pen-based interfaces and it has been used in a variety of applications, e.g., [10], [6], [13]. It involves dragging a closed path around the target objects, while avoiding the inclusion of undesired ones. Several variations of lasso have been proposed [9], [13], [12]. Auto-complete-lasso [12] automatically closes the loop as the user performs a lasso selection, which often speeds up the selection. Some systems combine the lasso selection and action-commands into a single stroke [5], [6], [8], [1]. For example, in Scriboli [5] and InkSeine [6] a pigtail gesture at the end of a lasso activates a menu.

Lasso selection can be time-consuming when selecting a spatially large cluster. In particular, it tends to be inefficient on large vertical displays, since the area covered by targets can be large, hard to reach (e.g., too high or too low), or it can cross the bezels of the tiled displays. Using a laser pointer instead of a pen resolves some of these issues. However, hand jitter inherent to laser pointer-based interaction can make it hard to avoid nearby non-targets.

This paper presents ICE-Lasso: a new selection technique that Intelligently Completes and Extends Lasso gestures. As the user draws a lasso, the system infers the target cluster and visualizes it via a temporary overlaid loop. When the user makes a pigtail gesture, the suggested cluster is selected. Otherwise, if she continues lassoing and captures at least one object, the system automatically switches back to the normal lasso. Additionally, ICE-Lasso facilitates the selection of multiple clusters, as well as partial or complete deselection of clusters. Figure 1 illustrates compares the two techniques for a simple cluster selection task.

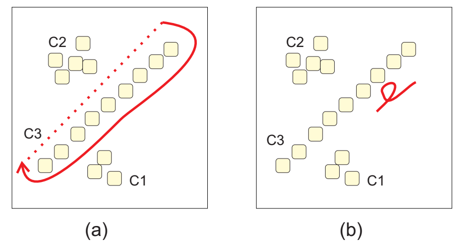

Fig. 1. In order to selecting cluster C3, a) lasso requires *precise* drawing of a long path and b) ICE-Lasso involves an initial segment of a lasso followed by a pigtail (b).

A. Related Work

Clusters, i.e., spatially neighboring graphical objects, are one of the most frequently-used structures, and they have been commonly used to organize semantically related items such as file icons digital inks [17]. Naturally, improving user interaction for cluster selection has a long history. Most relevant to the present discussion are efforts in the hypertext domain e.g. [7], [11], in the digital ink domain [3], [14], [16], [2], and general domain [4], [18]. With a few exceptions ([4], [16]), these systems apply simple hierarchical clustering algorithms, which cannot reliably detect clusters of arbitrary shapes or with non-homogeneous densities.

The interaction techniques presented in [7], [11], [4] are based on a multi-clicking approach: double-clicking on an object selects its cluster, and each successive click extends the selection with the nearest cluster(s). There are two limitations with this approach: first, performing multi-clicks with digital pens or laser pointers is hard and error-prone, second, there is no visual feedback before a selection is finalized. Also, there is no mechanism for editing a selection. Hence, if the user makes a mistake, she has to cancel the selection and start over.

Watanabe *et al.* proposed a novel approach for cluster selection named Bubble Clusters [18]. In this system, clusters are *always* visualized via surrounding bubble surfaces. The users select a cluster by clicking on its empty area ,and splits (or partially deselect) it by drawing a gesture through the cluster similar to the one in Tivoli [13]. The presence of bubbles introduces a significant visual clutter. Also, the system does not address multiple-cluster selection.

II. ICE-Lasso

ICE-Lasso consists of the following components: cluster detection, target cluster prediction, gestural interfaces, and a seamless mode switching with traditional lasso.

A. Cluster Detection

Cluster detection utilizes a perception-based scale-invariant algorithm called CODE [15]. In CODE, the influence of each object on the others is modeled by a normal distribution function. The center of this function is at the object location; the standard deviation is half the distance to the object nearest neighbor; and the peak is rescaled to one. The sum of these functions represents overall clustering strength. When the clustering strength in that region surpasses a threshold (set to one), objects on that region are clustered. Figure 2 illustrates this with four one-dimensional dots, labeled A to D. Dashed and solid curves represent the individual and the overall clustering strength, respectively. Applying the threshold results in three proximity regions, i.e., clusters $\{AB, C, D\}$.

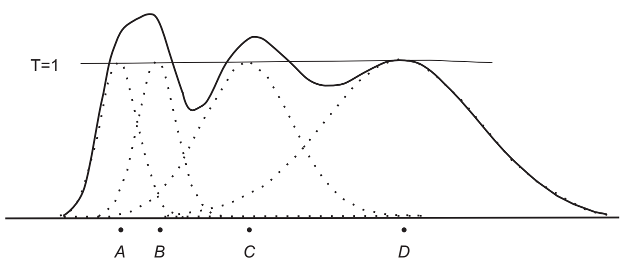

Fig. 2. Cluster Detection using CODE. Objects are labeled A to D. The dashed and solid curves represent individual and summed cluster functions, respectively. Applying the threshold T = 1 generates three cluster *AB, C, D*.

B. Predicting a Target Cluster

The prediction a target cluster is based on two observations gathered from an analysis of lasso gestures in various scenarios: first) the gestures tend to be smooth regardless of cluster shapes, and second) they often start near a target cluster and remain close to it. The search for a target cluster has two consecutive steps: direction-based and distance-based. Due to the dynamic characteristics of a lasso gesture, only the last portion (*LPG*) is considered. In the direction-based search, the algorithm searches for clusters that (a) none of their objects is currently selected, and (b) are partially or completely in "front" of the LPG. For this, the algorithm fits a line to the LPG. Then it uses the line perpendicular to the fitted lineas a separator. Clusters that are completely behind this separator are not considered in the **distance-based search**.In the distance-based search, the distances between the last point of the LPG and all remaining clusters are computed. The nearest cluster is returned as the inferred target. In case of ambiguities, where there are more than one inferred targets, the algorithm makes no prediction, and effectively waits until the user updates the LPG.

Figure 3 illustrates an example: $P1P2$ is the vector to the LPG and L is the orthogonal line. In the direction-search step $C3$ is discarded, as it is behind L. In the distance-based search, between clusters $C1$ and $C2$, $C1$ has the minimum distance to $P2$. Hence, the algorithm chooses it as the target cluster.

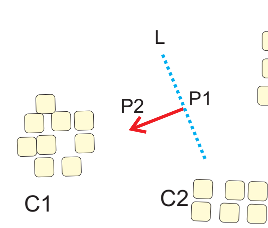

Fig. 3. $P1P2$ is the vector fitted to the LPG, and L is orthogonal to it. $C1$ and $C2$ are candidates, as they are in front of L. $C1$ is the predicted target, as it is closer to $P2$.

C. Gestural Interaction

1) Single Cluster Selection: While the user is drawing a lasso, the system infers the target cluster and visualizes it via a temporarily overlaid loop (called an ICE-Path), see Fig. 4-a and Fig. 4-b. If the user draws a pigtail gesture, the suggested cluster is selected, and the ICE-Path becomes a permanent solid loop, called ICE-Loop, see Fig. 4-c. Otherwise, if the user continues lassoing and captures at least one object, the loop disappears and the system switches back to lasso.

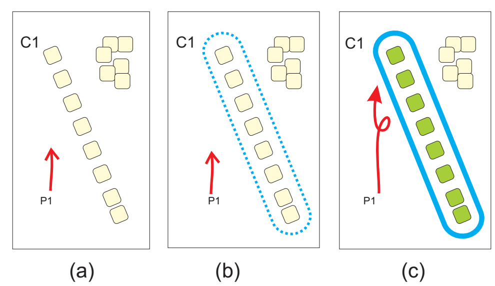

Fig. 4. Single cluster selection: a) A lasso is initiated at P1. b) Cluster C1 is predicted and an ICE-path (dotted oval) appears. c) Performing a pigtail selects C1 and an ICE-Loop (slid oval) is visualized.

2) Multiple Clusters Selection: Multiple clusters can be selected through either a continuous single gesture, or multiple disjointed gestures. Selection through continuous gesturing is an extension of a single cluster selection: after a cluster is selected, if the user continues lassoing, the system predicts the next cluster and visualizes it through an ICE-Path. Making a pigtail adds the next cluster to the selection, See Fig. 5.

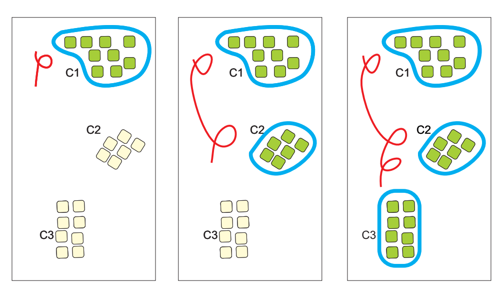

Fig. 5. Multiple cluster selection using continuous gesturing.

Continuous gesturing may involve a traversal of a long path. One common scenario is selecting a second cluster that is “behind” other objects, e.g., see Fig 6-a. To address this, ICE-Lasso offers a new disjointed selection technique which does not require a modifier key (such as the “shift” key): after a selection is finalized, i.e., when the pen is lifted and before any command is invoked, the user can extend the selection to another cluster by starting a lasso from the inside of an ICE-Loop. Figure 6 shows an example: in Fig 6-a, the user has selected cluster $C1$. Selecting $C2$ through continuous gesturing would require drawing a long path, see the dashed curve. Instead, in Fig 6-b, the user initiates a disjointed selection by starting a new lasso within the ICE-loop of $C1$. Similar to the single cluster selection, the system predicts cluster $C2$ as the target and the user confirms the prediction by drawing a pigtail.

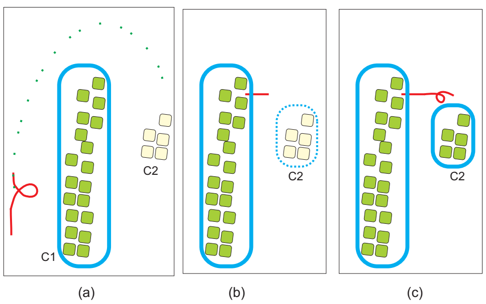

Fig. 6. Multiple cluster selection: a) Selecting $C2$ after $C1$ by continuous gesturing requires drawing a long path. b) Instead, the user starts a lasso gesture within the ICE-Loop of $C1$ and c) selects $C2$.

3) Partial or Complete Cluster Deselection: Similar to a *Bite* gesture introduced in Tivoli [13], curved gestures are used for partial deselection of a cluster. If the user makes a curved gesture (called a *cut*) within an ICE-loop, all objects that are on the concave side of the gesture are deselected, see Fig. 7.

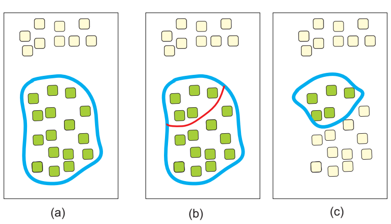

Fig. 7. Partial cluster deselection a) shows a selected cluster and its ICE-Loop. b) The user draws an inward curve. c) Objects outside on the concave part are deselected.

If all of the objects are on the concave side, the whole cluster is deselected, see Fig. 8. This feature enables users to deselect a cluster at any time during a selection without canceling the whole selection and without using a modifier key.

Unlike in Tivoli, the cut gesture does not have to start and end at the loop: if the user draws a small part of a cut gesture and pauses for a third of a second, the system extends it by a line through the gesture end-points. This feature considerably improves the deselection proposed in Tivoli when the path is narrow, (e.g., in dense clusters) or is long (e.g., on a large display).

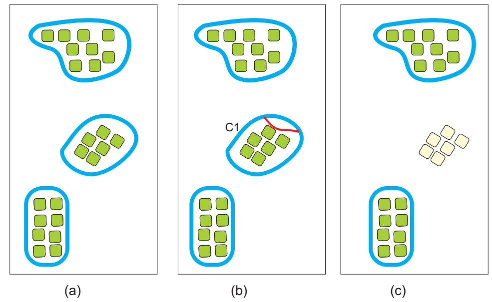

Fig. 8. Cluster deselection: Performing a “cut” gesture with no objects on the convex side deselects $C1$.

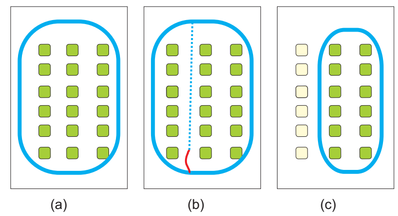

Fig. 9. Partial selection with automatic gesture completion. a) shows a selected cluster and its ICE-Loop. b) Drawing a curved gesture and a pause extends the gesture. c) Lifting the pen deselects the objects on the concave.

D. *Implicit Mode Switching Between ICE-Lasso and Lasso*

The system provides a seamless and automatic transition between the ICE-Lasso and the traditional lasso selection and eliminates the effort and potential errors of mode-switching.

Selection starts in lasso mode. When a lasso gesture is near a cluster and has not selected any of the cluster’s objects, the prediction algorithm visualizes the inferred cluster through an ICE-path. If the user performs a pigtail, the system switches to ICE-lasso mode, selects the cluster, and then switches back to lasso mode. Otherwise, if the user continues lassoing she stays in lasso mode. Figure 10 illustrates this.

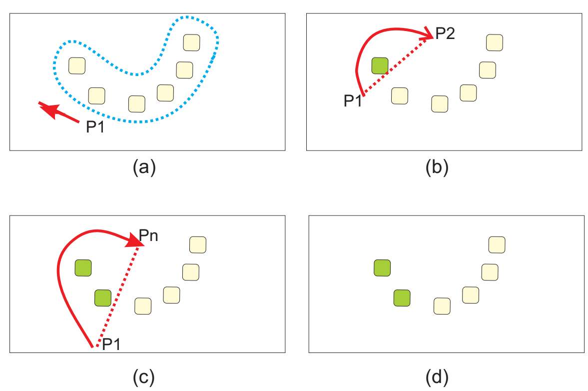

Fig. 10. ICE-Lasso can act as an auto-complete lasso. a) The user starts a lasso gesture; an ICE-Path is visualized. b) She continues gesturing towards $P2$ and an object is captured; the system switches to lasso and the ICE-Path disappears. C) Objects are now selected by lasso. D) Lifting the pen finishes the selection.

If a pigtail is performed when the lasso gesture has already enclosed any object, the corresponding objects are selected first, and then the LPG before the pigtail is used to determine the cluster to select. After any pigtail gesture has ended, a new lasso gesture is started. This (new) lasso gesture and all the subsequent ones have no effect on clusters already selected when in ICE mode. Figure 11 illustrates this. In Fig. 11-a, the user starts a lasso gesture from $P1$ to $P2$ and selects two objects in $C1$. At $P2$, the pen is close enough to $C2$, so an ICE-path for $C2$ appears. In Fig. 11-b, the user performs a pigtail and cluster $C2$ is selected. After the pigtail (i.e., at $P3$ Fig. 11-c) a a new lasso starts and the user moves toward cluster $C3$. She ignores any predictions and draws a lasso path to point $P4$ and cluster C is selected. In Fig. 11-d, the user lifts the pen and selection is completed. Alternatively and more efficiently, the user could have performed a pigtail gesture for cluster $C3$ shortly after $P3$.

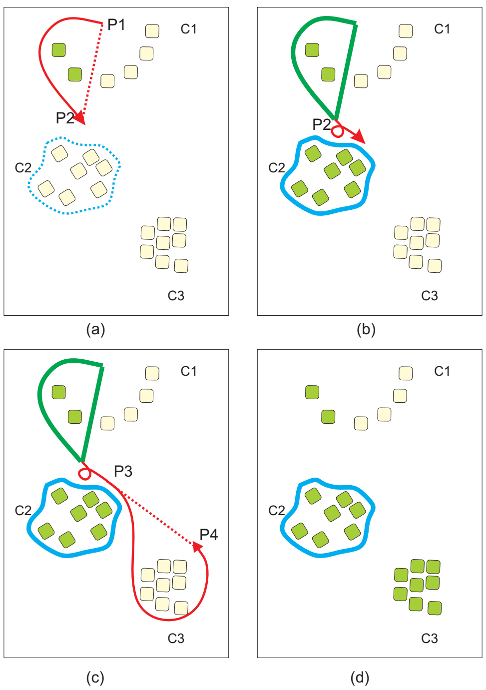

Fig. 11. Automatic mode switching. a) A lasso selects two objects from $C1$. ICE-Lasso predicts $C2$ and visualizes an ICE-path. b) A pigtail selects $C2$ and visualizes an ICE-Loop. c) A lasso starts after the pigtail (i.e. at $P3$) and the user continues lassoing (solid curve $P3P4$). $C3$ is selected. d)Lifting the pen finalized the selection.

E. Experiment

We performed an experiment to evaluate ICE-Lasso for complete selection of clusters. It was conducted in two different environments: a large display and a graphical tablet which are representative of large and small surfaces. This allowed us to investigate the effect of different input devices and movement distances on the techniques.

1) Stimuli: Each layout had four clusters of squares with 10 $times$ 10 pixels. There were one, two or three target clusters in each layout. All the clusters were similar in size and had arbitrary orientations. The size, i.e., spatial extent, of a cluster was defined by its bounding box. It was classified into three categories: small (2 cm x 2 cm), medium (4 cm x 4 cm), and large (8 cm x8 cm). This factor affected not only the circumference of clusters but also the distances between them. In summary, targets were classified as follows:

- 3 Spatial Sizes: small, medium, large;
- 3 Numbers of target clusters: one, two, three.

We designed 36 different layouts (4 layouts for each category).

2) Apparatus: The experiment was conducted on a Wacom PL-400 digitizing pen tablet with an embedded display, and a 60”x45” interactive wall (75” diagonal). The resolutions of both displays were 1024 $\times$ 768. The tablet was connected to a 2 GHz PC. The large display was connected to a 3 GHz computer. The software was written in Python and Tkinter.

3) Participants: 12 students from our university campus were recruited. None of them had used ICE-Lasso, and most of them did not have any experience with laser pointers on large display surfaces.

4) Procedure: For each task, one of the 36 above-mentioned patterns was shown to the participants. They were instructed to select the highlighted targets using the auto-complete lasso or the ICE-Lasso, as quickly and accurately as possible. Targets and distracters were displayed in green and black, respectively. If only targets were selected, a brief sound was played and the experiment advanced to the next task. Selection Time was measured from the first pen tap/laser pointer click after a layout was displayed to the time when only targets were selected. The number of cancelations, i.e, clicks on an empty area, was also recorded. During the lasso selection, the users were able to perform a disjointed selection using the Shift key. Also, they could deselect objects or clusters using a Ctrl key. These keys were not available for ICE-Lasso.

The experiment consisted of two consecutive phases, one on each platform. Both phases involved the same tasks and experimental procedure. For the first phase, the graphical tablet with a digital pen was used, while for the second phase the interactive wall with a laser pointer was used. The laser pointer was operated at a distance to the display. This was motivated by the observation that users close to the large display sometimes could not see all the target objects. To remove this potential confound, we asked users to select objects from a distance.

Participants were acquainted with both techniques on 10 practice layouts. In the main phase, the order of selection techniques was counterbalanced. Also, the order of layouts was randomized (differently) for each participant. Each participant repeated the main phase three times, using the same order of techniques and layouts. At the end, participants filled out a questionnaire to evaluate the ease of use and the learnability of both techniques.

5) Hypothesis: We hypothesized that ICE-Lasso would be significantly faster than lasso, since it considerably reduces dragging distances and does not require modifier keys. We also expected the participants to perform selections faster on the tablet.

6) Design: A within-subject full factorial design was used. There were three independent variables: Technique (lasso or

ICE-Lasso), Spatial Size, i.e., the size of the bounding box for each cluster (small, medium, large), and Target Number, i.e., the number of target clusters in each layout (1, 2, 3). The dependant variables were Selection Time and Cancellation Rate.

The 36 layouts discussed above were used in the main experiment. This sequence was then repeated three times for each participant. Also, the whole experiment was performed twice, once on the tablet and once on the interactive wall. Hence, each participant performed a total of $2 \times (10 \times 2 + 36 \times 2 \times 3) = 472$ selections including the training.

III. Results

A. Selection Time

The ANOVA showed a significant difference between Technique and Selection Time, $F_{1,11} = 25.10, p < 0.001$. The mean selection time for ICE-Lasso and lasso was 2.5 and 3.17, respectively, see Fig. 12. This confirmed our first hypothesis.

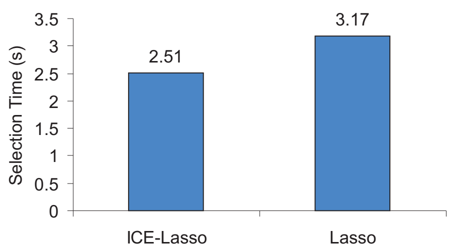

Fig. 12. Comparing mean Selection Time of Techniques.

Spatial Size There was a significant effect of Spatial Size on Selection Time, $F_{2,22} = 50.12, p < 0.0001$, and also significantly interacted with Technique $F_{2,22} = 41.18, p < 0.001$. For ICE-Lasso, there was no significant effect for Spatial Size. For lasso, selection of large clusters was considerably slower than selection of small and medium ones, see also Fig. 13.

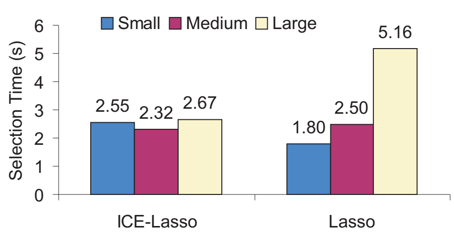

Fig. 13. The effect of Spatial Size on Selection Time.

There was a significant effect of Number on Selection Time $F_{2,22} = 102.82, p < 0.0001$. However, it did not interact with Technique, $F_{2,22} = 1.73, n.s;$, see also Fig. 14.

Input Device had a significant effect on Selection Time, $F_{1,11} = 13.48, p \ll 0.01$. There was also a marginal interaction between Input Device and Technique, $F_{1,11} = 2.09, p < 0.1$, see Fig 15. Our second hypothesis was confirmed.

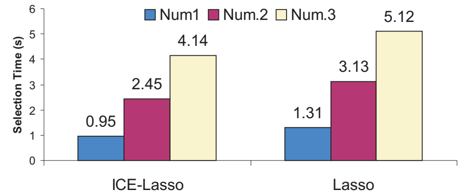

Fig. 14. The effect of Target Number on Selection Time.

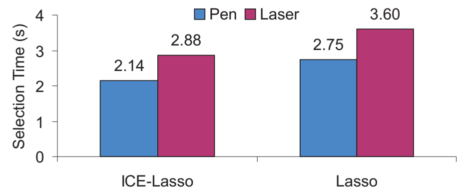

Fig. 15. The effect of Input Device on Selection Time.

Learning: Repetition had a strong main effect on Selection Time, $F_{2,22} = 8.93, p < 0.001$. There was a marginal interaction between Repetition and Technique, $F_{4,44} = 2.55, p < 0.1$. While there was no significant change on performance of ICE-Lasso, lasso significantly improved in later repetitions, see Fig. 16 and Fig. 17.

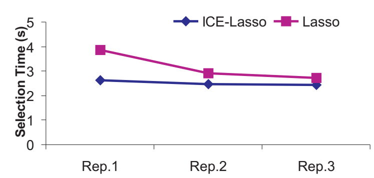

Fig. 16. The effect of Repetition on Selection Time.

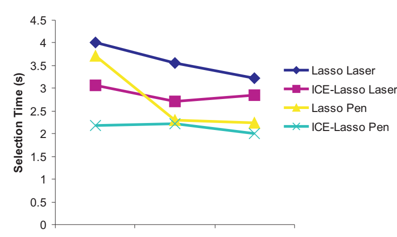

Fig. 17. The effect of Repetition and Input Device on Selection Time.

Cancelation Rate: The cancelation rate was very low for each of the technique. It had no interaction with Technique, $F_{2,20} = 0.31, n.s.$

Overall Preferences

At the end of the experiment,the participants were given a questionnaire to rank the techniques on their learnability, ease-of-use, memorability, and perceived efficiency. Ratings were on a 5-point Likert scale. Table I illustrates the averages of the subjective preferences.

	ICE-Lasso	Lasso
Easy to learn	4.68	4.25
Easy to use	4.25	3.50
Easy to memorize	4.50	4.00
Efficiency	3.95	3.41

TABLE I
AVERAGE RANKING OF THE TECHNIQUES, HIGHER IS BETTER.

Participants were also asked about the positive and negative aspects of each technique. A majority of participants, 10 out of 12, preferred ICE-Lasso over lasso. In general, ICE-Lasso was commented on as being easy to learn and use, while lasso required more effort. The lack of modifier keys was also a positive aspect of ICE-Lasso. One problem identified in ICE-Lasso was that pigtail gesture detection sometimes failed, which required a cancelation and a redraw.

IV. DISCUSSION

ICE-Lasso was significantly faster than the traditional lasso technique, because it considerably reduced dragging distances and did not require a modifier key. As it was expected, both techniques were slower on the large display, because they required longer dragging distances; and for ICE-Lasso, drawing a pigtail gesture using a laser pointer was harder.

For both techniques, increasing the number of target clusters increased the selection time. For lasso, most users performed multiple disjointed selection using the shift key. Some of them drew a single loop around the whole targets. Both alternatives involved longer dragging distances and more effort to avoid the non-targets. For ICE-Lasso, the user required to draw multiple pigtail gestures.

Spatial Size had no noticeable effect on ICE-Lasso while it significantly affected lasso selection. In addition to the increased circumference of clusters, Spatial Size also affected “tunnel width”, i.e., the distance between the targets and non-targets. This was true in our experiment, as the number of clusters was constant in all layouts. For selecting small clusters, lasso required a small dragging distance though a *wide* tunnel, which was easier with both pens and laser pointers. As Spatial Size increases, lasso selection becomes slower as it requires drawing a longer path through narrower tunnels.

While Repetition had no effect on ICE-Lasso on any of the platforms, it affected lasso. For lasso when the participants used a laser pointer, the selection times dropped linearly with repetition. when they used a pen, there was a sudden drop at the second repetition. This suggests that learning the lasso technique with a pen was easier than with a laser pointer.

V. CONCLUSION AND FUTURE WORK

This paper presented ICE-Lasso, a new intelligent selection technique that significantly reduces pen movement distances. Also, it offers the selection of multiple clusters and the partial deselection without the need for modifier keys. Additionally it can be overloaded onto the lasso technique through a seamless and automatic mode switching. Our user study indicates that it is significantly more efficient than lasso when selecting complete clusters.

There are a two limitations that we plan to address in the near future. First, we will evaluate the efficiency and learnability of ICE-Lasso for partial cluster selection. Second, we will investigate more advanced strategies for predicting user intention. Although the prediction algorithm is promising for selecting nearby clusters, it cannot reliably detect distant target clusters.

REFERENCES

[1] A. Agarawala and R. Balakrishnan. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *CHI*, pages 1283–1292, 2006.

[2] X. Ao, J. Li, X. Wang, and G. Dai. Structuralizing digital ink for efficient selection. In *IUI*, pages 148–154. ACM Press, 2006.

[3] P. Chiu and L. Wilcox. A dynamic grouping technique for ink and audio notes. In *UIST*, pages 195–202, 1998.

[4] H. Dehmeshki and W. Stuerzlinger. Intelligent mouse-based object group selection. In *Smart Graphics*, 2008.

[5] K. Hinckley, P. Baudisch, G. Ramos, and F. Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *CHI*, pages 451–460, 2005.

[6] K. Hinckley, S. Zhao, R. Sarin, P. Baudisch, E. Cutrell, M. Shilman, and D. Tan. Inkseine: In situ search for active note taking. In *CHI*, pages 251–260, 2007.

[7] T. Igarashi, S. Matsuoka, and T. Masui. Adaptive recognition of implicit structures in human-organized layouts. *Visual Languages*, pages 258–266, 1995.

[8] G. Kurtenbach and W. Buxton. Issues in combining marking and direct manipulation techniques. In *UIST*, pages 137–144, 1991.

[9] E. Lank and E. Saund. Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers & Graphics*, 29(4):183–192, 2005.

[10] J. LaViola and R. C. Zeleznik. Mathpad2: a system for the creation and exploration of mathematical sketches. In *ACM Trans. on Graphics*, volume 23, pages 432–440, 2004.

[11] C. Marshall, F. Shipman, and J. Coombs. Viki: spatial hypertext supporting emergent structure. In *ECHT*, pages 13–23, 1994.

[12] S. Mizobuchi and M. Yasumura. Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors. In *CHI*, pages 607–614, 2004.

[13] T. Moran, P. Chiu, and W. Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *UIST*, pages 45–54, 1997.

[14] T. Moran, P. Chiu, W. Melle, and G. Kurtenbach. Implicit structure for pen-based systems within a freeform interaction paradigm. In *CHI*, pages 487–494, 1995.

[15] M. Oeffelen and P. Vos. An algorithm for pattern description on the level of relative proximity. *Pattern Recognition*, 16(3):341–348, 1983.

[16] E. Saund and T. Moran. A perceptually-supported sketch editor. In *UIST*, pages 175–184, 1994.

[17] F. Shipman, C. Marshall, and T. P. Moran. Finding and using implicit structure in human-organized spatial layouts of information. In *CHI*, pages 346–353, 1995.

[18] N. Watanabe, M. Washida, and T. Igarashi. Bubble clusters: an interface for manipulating spatial aggregation of graphical objects. In *UIST*, pages 173–182, 2007.