Improving drag-and-drop on wall-size displays

Maxime Collomb, Mountaz Hascoët

Improving drag-and-drop on wall-size displays

Maxime Collomb, Mountaz Hascoët
LIRMM, UMR 5506 du CNRS. Univ.Montpellier II. Montpellier, France.
collomb@lirmm.fr

With the emergence of wall-size displays, touch and pen input have regained popularity. Touch/pen input requires users to physically reach content in order to interact with it. This can become a problem when targets are out of reach, e.g., because they are located too far or on a display unit that does not support touch/pen input as explained by Baudisch et al. (2003). Some interaction techniques have been proposed to simplify drag-and-drop from and to inaccessible screen locations, across long distances, and across display unit borders.

The approaches

The proposed techniques include *pick-and-drop* by Rekimoto (1997), *push-and-throw* by Hascoët (2003) and *drag-and-pop* by Baudisch et al. (2003).

Pick-and-drop mechanism is close to traditional drag-and-drop. It does not require users to maintain contact with the screen. Instead, users make a click to pick an object and another click to drop it. The pick and drop operations can occur on different displays but have to be made with the same pen. Push-and-throw and drag-and-pop use opposite approaches.

1. *The pointer-to-target approach*

The first approach, illustrated by push-and-throw (fig. 1-left), consists in throwing objects to target instead of moving the pointer all the way to the target. As the main problem with throwing is precision, the idea behind push-and-throw is to provide adequate feedback and trajectories. The feedback provides users real-time preview of where the dragged object will come down if thrown, and trajectories are inspired by the metaphor of the pantograph. Hence, this temporarily turn the pen/touch input, inherently a direct pointing device, into an indirect pointing device in order to shorten distances faster as well as to make it possible to reach locations further away or on different screen units.

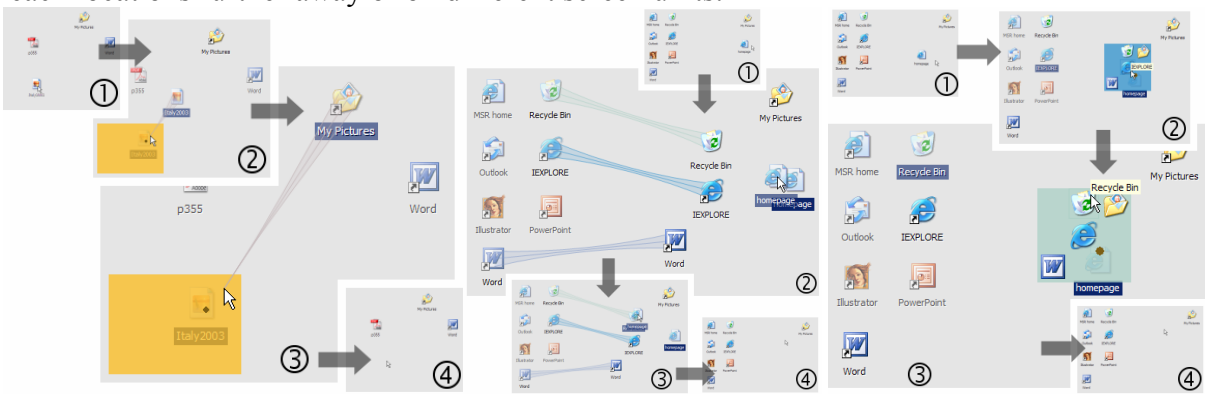

Figure 1 (L to R): push-and-throw, drag-and-pop and push-and-pop walkthrough.

2. *The target-to-pointer approach*

Drag-and-pop (fig. 1-centre) uses the opposite approach to push-and-throw. Rather than sending the dragged object to the periphery, it allows users to bring a selection of likely candidates to the user (a "tip icon" is created for each candidate). This allows users to complete drag interactions in a convenient screen location.

3. *Comparison of approaches*

There are two major differences between these approaches. The first one is the need of reorientation. Indeed, using push-and-throw, users are focused on the target space and have to constantly monitor the screen to adjust their movement. On the other hand, drag-and-pop requires users to reorient themselves only once. Rubber bands are used to minimize that impact and once users have identified the target tip icon, they can complete the interaction easily.

The second difference is the possibilities offered by each approach. The target-to-pointer approach assumes that the movement has a target which is the case when dragging an icon to the recycle bin for example. This is not the case when rearranging icons on the desktop.

The best of both approaches

Based on our analysis of push-and-throw and drag-and-pop, we created a new technique designed to combine the strengths of both techniques. We call this new technique push-and-pop, see Collomb et al. (2005). Fig 1-right shows a walkthrough in which the user is dragging a word document into the recycle bin. The idea behind push-and-throw is to use the world in miniature environment from push-and-throw while keeping the full size tip icons of drag-and-pop, allowing users keep focus on the source area.

In case users need to rearrange icons on the desktop, they can switch push-and-pop temporarily into a push-and-throw mode. Users invoke this functionality by moving the pointer back to the location of invocation. Push-and-throw has been improved with the introduction of a non-linear acceleration which addresses the lack of precision of push-and-throw and allows a one pixel pointing precision.

Studies

We made 2 experiments to compare movement times and error rates for six techniques (fig. 2): drag-and-drop, pick-and-drop, push-and-throw, drag-and-pop, push-and-pop and acc. push-and-throw. Both studies had similar results. Push-and-pop performed best. It was just a little better than drag-and-pop for times but much better for errors. Then comes acc. push-and-throw, pick-and-drop and push-and-throw. Classic drag-and-drop performed well for short distances but performed poorly when the task required user to cross bezel between screens.

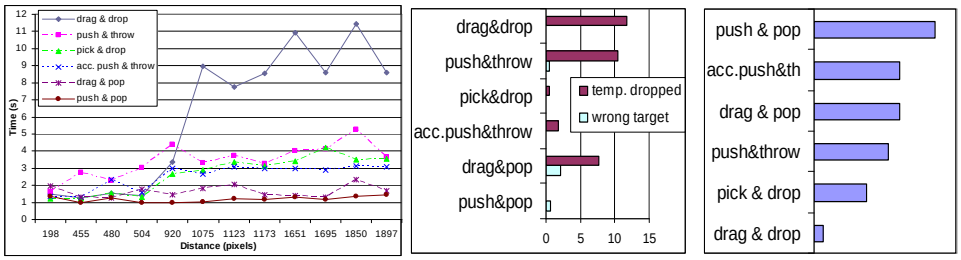

Figure 2: Results of the first study. (L to R): movement times, error rates, user preferences.

Conclusion

Confirming findings by Baudisch et al. (2003) drag-and-drop performed well as long as source and target icons were situated in the same display unit, but failed quickly when long distances and bezels were involved. In addition, we found that pick-and-drop is affected by distance in a similar way, though to a lesser extent. This is coherent with the fitt's law.

For all other evaluated techniques, target distance had comparably little impact on task performances. However, our studies seem to indicate a performance benefit of acquisition techniques that require a one-time reorientation (drag-and-pop and push-and-pop) over techniques that require continuous tracking.

Overall, the study indicates that push-and-pop is indeed a useful technique. Push-and-pop outperformed all other techniques, including its ancestors, drag-and-pop and push-and-throw. Participants' subjective preference reflected this. Push-and-pop also offered a very low error rate. Among pointer-to-target techniques, accelerated push-and-throw performed significantly better than traditional push-and-throw. Consequently, the combination of push-and-pop and accelerated push-and-throw appears as the most efficient technique in terms of accuracy, speed and reachability.

References

Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A (2003). Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch and Pen-operated Systems. Interact'03, p 57–64.

Collomb, M., Hascoet, M., Baudisch, P., and Lee, B. (2005) Improving drag-and-drop on wall-size displays. GI'05, p 25–32.

Hascoët, M. (2003). Throwing models for large displays. HCI'03, pp. 73–77.

Rekimoto, J (1997). Pick-and-Drop: A direct manipulation technique for multiple computer environments. UIST'97, p 31–39.