# Lodestones and Leylines: Gracefully Constraining Movement in Multiscale

**Susanne Jul**
University of Michigan
sjul@acm.org

## ABSTRACT
This paper introduces a technique to help users in a graphic multiscale environment move between locations of interest faster and prevent them from getting lost. The technique constrains movement to direct paths between salient objects or views of objects. Based on an indication of approximate direction from the user, the system predicts the desired destination, and then uses this prediction to compute a path and guide movement. Using this technique, users perform fewer physical actions and experience less disorientation.

## KEYWORD
Navigation, Locomotion, Movement, Multiscale, Zooming User Interface, Jazz, Lodestone, Leyline.

## INTRODUCTION
Many cross-country ski resorts proudly advertise the number of miles of groomed trails they maintain. A groomed trail consists of two parallel grooves—such as the track a skier might leave—carved into the snow, each about the width of a ski, about a foot apart. When track skiing, skiers allow the grooves to guide their skis. Groomed trails not only lead skiers to interesting places, they also simplify the mechanics of controlling movement. Grooming trails in the physical world requires considerable effort and the resulting trails do not change, either in response to skier s needs or environmental changes.

This paper reports on a technique that dynamically grooms trails in a graphic multiscale environment in response to user input. This technique constrains movement to paths that lead directly from the current location to objects or views of objects. Based on an indication of approximate direction from the user, the system predicts the desired destination, and then uses this prediction to compute a path and guide movement. Results from a small pilot study indicate that users do less physical work and are more confident when using this technique for movement.

## RELATED WORK
Both prediction of destination location and automatically computed paths have been used elsewhere. History mechanisms [7] and query relevance metrics [2] use past behavior and explicit user input, respectively, to predict probable destinations. They then prioritize the accessibility of the predicted locations, for instance, by organizing them in a special menu or other list. These techniques do not guide actual movement. Point of interest movement [6] and path drawing [3] guide movement along automatically computed paths. Paths are based on the user s indication of the desired destination or an approximate path, respectively. The present work combines these approaches by predicting the destination and using this prediction to compute a path.

Jazz [1, 4], an application framework for designing and building multiscale electronic worlds, is the basic interaction environment used in this work. Like its predecessor, Pad++, Jazz employs an interaction metaphor of a conceptually infinite two-dimensional surface that can be viewed at an infinite range of magnifications. Objects have position and extent on the surface, and can appear differently, even becoming invisible, depending on the magnification (scale) of the view. Traditionally, movement is by panning (moving across the surface) and zooming (changing the scale of the view). Not knowing where next to go to reach an object (often *any* object) is a common problem in Jazz worlds [5]. Because both space and scale dimensions are conceptually infinite, random movement is unlikely to be fruitful. Systematic movement is nearly impossible using conventional pan and zoom movement.

## LODESTONES AND LEYLINES
The lodestones and leylines prototype permits movement only along a *leyline*. A leyline is a direct path from the present location to a *lodestone.* A lodestone is any entity that defines a location and is salient to the user. This can be an object, a set of objects, a view, etc. In the prototype, zoom-in lodestones are individual objects and the single zoom-out lodestone is a (special) view of a certain set of objects. A fundamental assumption of the design is that the user s goal is *always* to reach a lodestone.

In order to initiate movement, the user indicates the desired direction of zoom (in or out) by pressing the appropriate mouse-button. If zooming in, the system uses the mouse location to select the nearest lodestone in the view (regardless of whether it is visible) as the predicted destination and begins to zoom along the leyline that centers the lodestone (Figure 1 A-C) in the view. The user can change the targeted lodestone at any time during zoom-in by moving the mouse to be closer to a different lodestone (Figure 1 $B_1$, B). The system will immediately determine the new destination and switch to its leyline. This allows effortless error correction and scanning of potential targets. If no lodestones can be reached by zooming, zoom-in is not permitted.

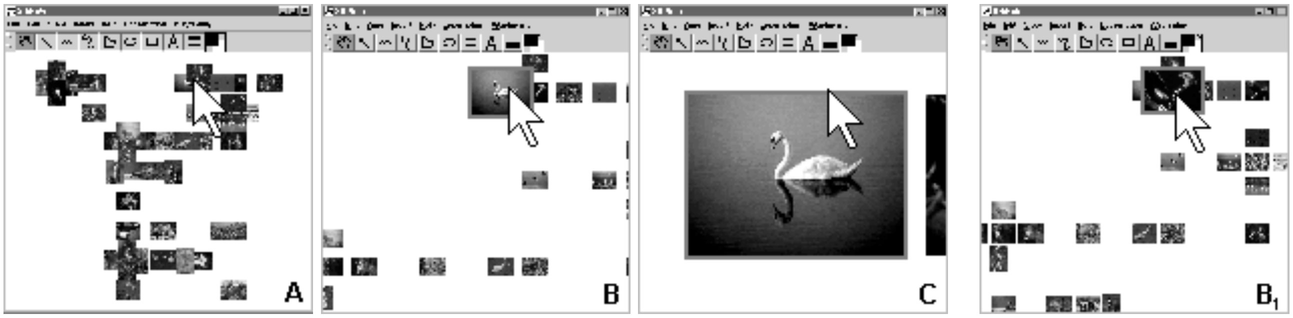


**Figure 1** *Zoom-in:* The user clicks in the vicinity of the object that they want to go to (A). The system selects the nearest lodestone, highlights and zooms toward it (B, C). *Changing target:* If zoom-in target prediction is incorrect ($B_1$), the user corrects it by moving the mouse (B). *Zoom-out:* The Top of the World view (A) is always the lodestone targeted during zoom-out. Clicking to zoom out anywhere in B, C or $B_1$ follows a leyline to A, where zoom-out stops.

If zooming out, the target destination is assumed to be the *Top of the [Lodestone] World* view (Figure 1 A). This is the most magnified view that contains all lodestones in the world. Since no new lodestones can be brought into view by further zooming, zoom-out is not permitted past this view.

Thus, to get from one lodestone object to another, the user either does a single zoom-in or a two-step zoom-out-zoom-in. If the desired lodestone is in the current view, the user need only click somewhere in its vicinity to follow the leyline that leads to it. If the desired lodestone is not in the current view, the user clicks anywhere to zoom out until it is in view, then clicks to zoom in. In either case, zooming ceases when no more lodestones can be brought into view.

This is like a skier pointing their skis in the general direction they want to go, and a track being groomed just ahead of them leading to the nearest interesting place in that direction. If they change their mind or the nearest destination is not the one they want, they lean a bit (while skiing) and the track shifts smoothly toward a different location. If there are no interesting places in the indicated direction, no track is groomed. If a destination cannot be reached by going forwards, the skier points backwards and a track leads them to a place from which all locations can be reached.

The focus of the work was on movement control, but it proved necessary to experiment with visual feedback as well. In spite of knowing that movement always leads to something, users were uncomfortable not knowing where it led—much like track skiers in a blizzard. The most successful of the feedback mechanisms implemented was to provide thumbnail views of lodestones not otherwise visible (Figure 1 A, B, $B_1$), and highlighting the targeted lodestone (or its thumbnail) during zoom-in (Figure 1 B, $B_1$).

Results from a small pilot study are quite promising. When using lodestones and leylines, subjects visibly performed fewer physical actions, including much less stop and go movement. Many subjects spontaneously reported feeling less lost and more confident about their actions. One subject, upon being asked to return to traditional (unconstrained) movement, responded, Do I have to?

## FUTURE WORK

There are three areas of further development. The first area expands destination prediction by generally considering groups of lodestones—like the special Top of the World group—as single lodestones. The second area explores relaxing movement constraints selectively, e.g., allowing unconstrained movement locally (in the vicinity of lodestones). The third area relates zoom speed to the number of lodestones in the view, proportionally adjusting the time available for making decisions during movement.

## SUMMARY

This paper has introduced a technique for helping users find and follow their way by constraining what movement is possible. Because constraints are based on the user s immediate needs and change dynamically, they are supportive rather than restrictive to users. This technique differs from most efforts to support movement in reducing, rather than increasing, freedom of movement. The approach used illustrates the utility of considering the purposes of movement when designing the mechanics, and the potential for addressing high-level needs through low-level mechanisms.

## ACKNOWLEDGMENTS

Thanks to George Furnas, Rudy Darken, Barry Peterson and the Naval Postgraduate School MOVES department.

## REFERENCES

1. Bederson B., Meyer J., Good L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *Proceedings of ACM UIST 2000*, ACM Press.
2. Frants, V. J., Shapiro, J., and Voiskunskii, V. G. (1997). *Automated Information Retrieval: Theory and Methods.* San Diego: Academic Press.
3. Igarashi, T., Kadobayashi, R., Mase, K., Tanaka, H. (1998). Path Drawing for 3D Walkthrough. *Proceedings of ACM UIST 98*, ACM Press, 173-174.
4. http://www.cs.umd.edu/hcil/jazz
5. Jul, S., Furnas, G. W. (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. *Proceedings of ACM UIST 98*, ACM Press, 97-107.
6. Mackinlay, J. D., Card, S. K., Robertson, G. G. (1990). Rapid Controlled Movement Through a Virtual 3D Workspace. *SIGGRAPH 90 Conference Proceedings*, in *Computer Graphics* 24 (4), 171-176.
7. Nielsen, J. (1990). The Art of Navigating Through Hypertext. *Comm. ACM* 33, 3 (Mar.), 296-310.