

GPSel: A Gestural Perceptual-based Path Selection Technique

Hoda Dehmeshki* and Wolfgang Stuerzlinger**

Department of Computer Science and Engineering
York University, Toronto, Canada

Abstract. This paper introduces a gestural perceptual-based approach to select objects, i.e., nodes and/or edges along paths. Based on known results from perception research, we propose a model to detect perceptually salient paths formed by the Gestalt principles of good continuity and closure. Then we introduce gestural interaction techniques that enable users to select single or multiple perceptual paths, as well as resolving ambiguities in selection. The result of a user study shows that our system outperforms current techniques for path selection.

1 Introduction

Node-link diagrams are fundamental tools to visualize relational data. Objects and their relationships are depicted as nodes and edges, respectively. Such diagrams are used in a large variety of application domains such as software engineering, social network analysis, and data modeling.

Many algorithms for drawing comprehensive graphs by minimizing certain aesthetic criteria such as edge crossing and bending [4] have been developed. However, it is not always computationally feasible to find an optimum solution that satisfies all criteria. Hence users often direct the algorithm by relocating groups of objects (i.e., nodes and/or nodes) or imposing constraints [8, 25, 19, 23]. Both alternatives require manual selection of object groups. Group selection is also required prior to many standard operations such as deletion, translation, annotation, changing properties, rotation, navigation [16] and zooming [6].

In graphical user interfaces, standard group selection techniques are shift-select, rectangle selection, and lasso. Shift-select, i.e., clicking on objects while holding the Shift key down, is impractical if the number of targets is large. In rectangle selection, the user performs a drag operation along the diagonal of the selection region. This technique works poorly in the graph domain, because the frequent presence of nearby non-targets often necessitates subsequent de-selection(s) or disjoint selection steps. Also, layouts are often non-axis aligned. Lasso involves dragging a closed path around the targets, while avoiding inclusion of none-targets. Auto-complete lasso [22] automatically closes the path as the user performs a lasso, which makes it more efficient. As explained by the Steering

* e-mail: hoda@cse.yorku.ca

** www.cse.yorku.ca/~wolfgang

law [1], lasso tends to be inefficient for path selection, especially for long paths, since nearby distracters make the traversal “tunnel” small, which necessitates a reduction of movement speed.

This paper introduces a new gestural path selection technique called *GPSEL*. Based on established models from perception research, we present an algorithm to detect salient perceptual paths formed by the Gestalt principles of good continuity and closure. Then, we introduce gesture-based interaction techniques which allow users to select (partial or complete) good continuity paths, multiple paths, and resolving ambiguities. Finally we present a user study that shows GPSEL outperforms current techniques for visually salient groups.

2 Related Work

2.1 Perceptual Grouping in Graphs

Perceptual grouping is defined as the human ability to detect structure(s) among visual items. The most fundamental approach for perceptual grouping is Gestalt psychology, which is described via a set of principles [18]. This paper focuses on two prominent principles: good continuity and closure. Good continuity states that visual items which are along smooth curves are perceived as a group. Closure states that the humans visual system tends to seek closed areas.

The good continuity principle has been used in graph drawing algorithm as it significantly improves comprehension of graphs [21, 24] and recognition of shortest paths [29]. It also has been used to make certain paths more apparent, see Fig. 1. Two important aesthetic criteria that are frequently employed by graph drawing algorithms are to minimize the number of bends in polyline edges [28] and to maximize the *angular resolution* [15]. Angular resolution refers to the minimum angle between two edges sharing a node. These criteria try to keep paths as straight as possible. Hence, they create perceptual groups of objects, i.e., nodes or edges, which conform to the Gestalt principle of good continuity.

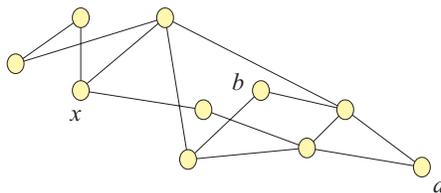


Fig. 1. The effect of good continuity on graph comprehension. Perceiving the shortest path between nodes x and a is significantly easier than that between nodes x and b .

2.2 Good Continuity Group Detection

Extracting linear and curvilinear structures from edges or dots has been intensively studied both in computer vision and perception research. Studies in

computer vision have suggested that grouping edges into (curvi-)linear structures is based on local rather than global straightness and smoothness [3, 5, 14]. This idea has been supported by several experimental studies in perception research [12, 9, 11, 10]. Field *et al.* conducted several experiments to determine the rules that govern the perception of good continuity groups [12]. The results show that the ability to detect a path is significantly reduced when the relative orientation between successive elements differs by more than 30° . Also, “end-to-end” alignment of the elements significantly increases the observer’s ability to perceive them along a path. Feldman presented a mathematical model for subjective judgment of curvilinearity among three and four dots [9, 11]. The model is a gaussian function of angles between successive dots. Experimental results in [10] suggests that perceived curvilinear configuration can be accurately modeled using a *local* 4-dot window moving along the chain of dots.

2.3 Interaction Techniques for Path Selection

The standard techniques for selection in graphical user interfaces are rectangle selection and lasso. However, these two techniques are not very well suited to path selection. Several researchers designed techniques to target this problem.

Accot and Zhai introduced a novel technique where the user drags a continuous gesture across all desired objects [2]. Their technique is time consuming and error-prone when there are too many targets, the size of each object is small, or the targets are spread over a large distance. Saund and Moran [27] developed a technique for the selection of contour groups in freehand sketch editors, called path tracing. In this technique, users can select a group by drawing a gesture that traces an approximate path over the target group. This technique can be applied in the graph domain. However, in the presence of nearby distracters the gesture has to be similar in shape and size to the target path. Hence, similar to Accot’s work [2], it is inefficient when selecting a long path.

A number of advanced graph editors provide semi-automatic path selection techniques that leverage internal information in graphs. One commonly-used approach is text querying where the user types in specific information about the desired path and runs a query. This techniques requires both the user and the system to have a deep understanding of the content visualized in the graph.

Google Maps takes a more interactive approach in its Route Planning system: the user types in a source and a destination address; the system computes and suggests the fastest route between them. The user can then iteratively modify the route by clicking on it and dragging the mouse/pen over a point (called a waypoint) on the desired route. After each modification, the system re-computes and suggests a new route composed of shortest paths between waypoints (including the source and destination). This technique is application-dependant and also limited to selection of shortest paths. If the user has other criteria in mind (e.g., a path going through parks and green areas for cycling or the path with the least turns) the technique will often require too many waypoints.

Dehmeshki and Stuerzlinger developed a system that introduces perceptual-based object group selection techniques in a traditional desktop system, i.e. with

a mouse as input device [7]. It utilizes a nearest neighboring graph to detect (curvi-)linear groups, and proposes a set of click-based interaction techniques to select among the detected group(s). Three key elements distinguish this system from the new work presented here. First, the mouse-based system is based on a model that depends also on inter-object distances, to be able to deal with object clusters. This is not really appropriate for the graph domain. Second, the system relies heavily on double- and triple-clicks, which is not appropriate for pen-based systems. Lastly, the user interface of the mouse-based system selects all groups that an object belongs to, but does not consider the direction of the grouping. This increases the need to disambiguate the selection, which results in more user effort. In the present system, the use of gestures with their *inherent* directionality greatly reduces the effort to specify the group the user intends to select.

3 GPSel: Gestural Interaction

3.1 Good Continuity Path Selection

To select a *complete* good continuity path, the user performs a straight gesture crossing one of the end nodes, called *anchor*, in the same direction as the target path. The technique uses a method to detect good continuity paths starting from the anchor, which will be described later in this document.

To select a *partial* good continuity path, the user first selects the complete path, then limits the selection by performing a second straight gesture across the last target node and in the opposite direction of the first gesture. All objects between the first and the second gestures remain selected, while objects beyond the second gesture are de-selected. Figure 2 shows an example: in Fig. 2-a, the user makes a flick gesture over node $N1$ to select the diagonal path. In Fig. 2-b, the path corresponding to the selection is highlighted. In Fig. 2-c, the user performs a second gesture starting on node $N2$. As shown in Fig. 2-d, nodes between $N1$ and $N2$ remain selected, while nodes beyond $N2$ are de-selected.

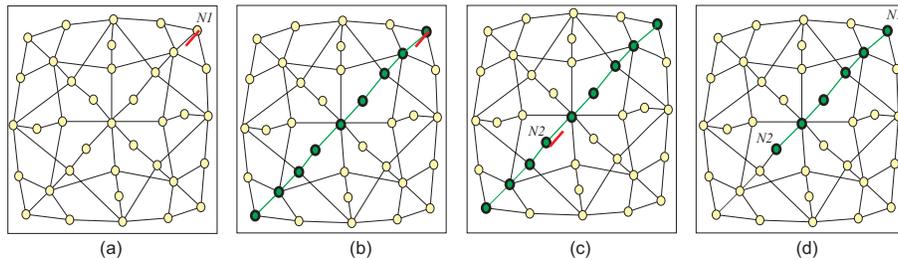


Fig. 2. Good continuity path selection. a) The user performs a straight gesture crossing node $N1$ along the diagonal path. b) The path is selected. c) Performing a second gesture starting on object $N2$ deselects objects beyond it, shown in d).

3.2 Resolving Ambiguity in Path Selection

When a gesture corresponds to multiple good continuity paths, GPSel selects all of them, and allows the user to disambiguate by performing another gesture over a node that belongs to the target path. Figure 3 illustrates an example: performing a gesture over node $N1$ selects both good continuity paths sharing $N1$; performing another gesture over node $N2$ deselects the undesired branch.

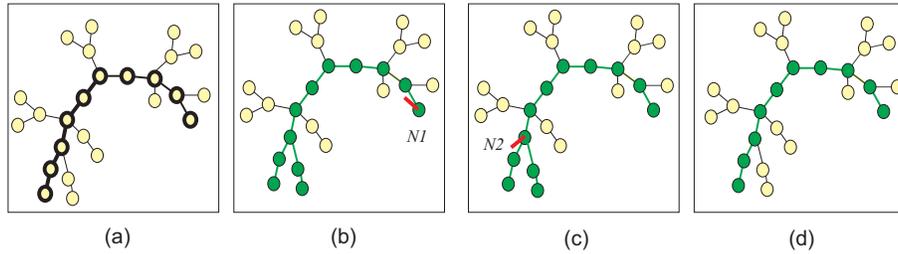


Fig. 3. Resolving ambiguity. a) Targets are shown with thick borders. b) Performing a gesture on $N1$ selects its two corresponding perceptual paths. c) A second gesture over node $N2$ deselects the undesired sub-path, as shown in (d). Graph from [17].

3.3 Compact Cyclic Path Selection

Our interaction technique also enables the user to select compact cyclic paths. In its simplest form, a circular gesture in the interior region of a closed path selects that path. If the gesture is performed *around* a node or *over* an edge, all objects along the closed paths shared by that object are selected. This interaction technique works only for planar areas, i.e., no crossing edges. This is not a significant limitation as planarity is by far the most important aesthetic criterion [24] and hence one of the main design criteria for graph drawing algorithms. Figure 4 shows three examples of cyclic path selection.

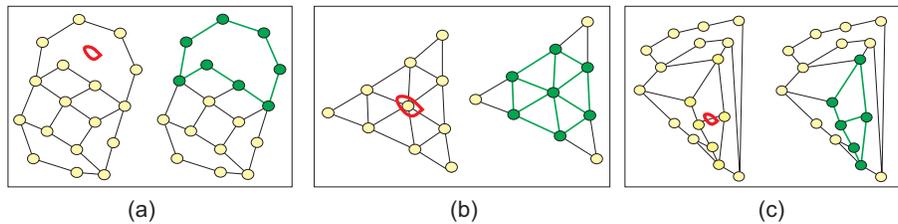


Fig. 4. Examples of Cyclic Path Selection. A circular gesture (a) in an interior area, (b) around a node or, (c) on an edge selects the corresponding cyclic paths.

3.4 Edges vs. Nodes Group Selection

The above interaction techniques select both the corresponding nodes *and* edges by default. In order to select only nodes or edges, GPSel utilizes a marking menu [20]. If the user holds the pen still for a short interval of time (approximately one third of a second) a marking menu with three items ‘select edges’, ‘select nodes’, and ‘select both’ appears under the pen’s tip. The user then selects a menu option by making a straight gesture towards the desired item. Expert users who know the location of an item can select it by immediately moving toward the item before the menu appears.

4 GPSel: Detecting Groups

4.1 Good Continuity Group Detection

The algorithm computes a linear coefficient (LC) for each set of four nodes which are along any paths that starts from the anchor node and is aligned with the direction of the gesture. This coefficient indicates how strongly the nodes are perceived as a straight line. LC is defined by:

$$LC = \exp\left(-\frac{(a1^2 + a2^2 - 2ra1a2)}{2s^2(1 - r^2)}\right),$$

where $a1$ and $a2$ are angles between successive edges (see Fig. 5), r and s are constants. This is based on Feldman’s model for linear grouping of four consecutive dots [11]. For groups of only 3 nodes a simpler formula is used [9].

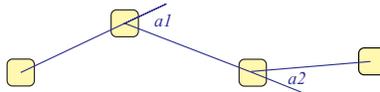


Fig. 5. Illustration of parameters used in the definition of collinearity.

We extended this idea to model arc groupings. In the above equation, we substitute every inter-line angle α_i by $(\alpha_i - \alpha_{Avg})$ where α_{Avg} is the average of all line angles α_i ’s. Hence, a uniform curvilinear path gets a higher grouping coefficient than a sinuate path.

In an extra step, the initial collinear and curvilinear sets are repetitively merged to form longer groups.

4.2 Compact Cyclic Path Group Detection

We employ an effective algorithm for detecting compact cyclic paths [26]. When the user performs a circular gesture, the algorithm finds the closest node and its closest edge to the gesture. Then it starts tracing a path from the detected node and the edge. If there are multiple paths originating from a node, it selects the edge that has maximum turning toward the gesture.

5 Experiment

We conducted a within subject study to assess the efficiency of GPSel for selecting *good continuity* paths in comparison to rectangle selection and lasso. We considered rectangle selection and lasso as comparison points, since these are the standard techniques supported by almost all graph editors.

Generating Test Graphs

Since we were interested in evaluating the performance of our technique in a practical application context we focused on automatically drawn graphs based on real data sets. The graphs were undirected and abstract. On average, they had 41 nodes and 46 edges. Figure 6 shows several examples. While randomly generated graphs seem to be a feasible alternative, they are typically not representative of real data.

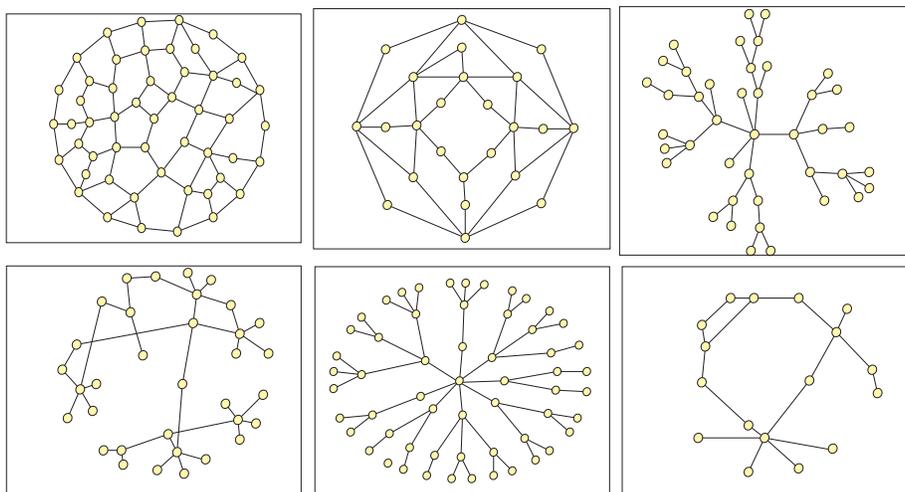


Fig. 6. Example graphs used in the experiment.

Tasks

In each task, the participants were shown a graph and asked to select targets (a good continuity path or a closed loop) using auto-complete lasso, rectangle selection, or GPSel. Targets and distracters were displayed in green and black, respectively. If only targets were selected, a brief sound was played and the experiment moved on to the next task. Selection time was measured from the first pen down event after a graph was displayed to the time when the targets were selected.

Hypothesis

We hypothesized that GPSel is significantly faster than lasso and rectangle selection. The reason is that it requires only a flick gesture, whereas the other two require tracing the whole path in different ways. Moreover, the presence of nearby distracters increases the difficulty of rectangle selection and lasso.

Experimental Design

We used a fully crossed within-participant factorial design. There were two independent variables: Selection Technique (GPSel, auto-complete lasso, and rectangle selection) and Target Length (small ≈ 250 , medium ≈ 400 , and large ≈ 750 pixels). The dependant variable was Selection Time. We used a 3x3 Latin Square to counterbalance the order of selection techniques. Participants were first trained for 10-20 minutes on all three techniques with 7 practice layouts. The main experiment used 12 different graphs which were shown for each technique. This sequence was then repeated 3 times. Hence, each participant performed a total of $12 \times 3 \times 3 = 108$ selections during the main experiment.

Apparatus

The experiments were conducted on a Tablet-PC with a Pentium M 1.6 Ghz processor and 1 GB memory. Screen resolution was set to 1024x768 and a pen was used as input device. The software was written in Python and Tkinter.

Participants

Twelve students from a local university campus were recruited to participate in the experiment. None of them had used our technique before. Most of them were unfamiliar with auto-complete lasso.

Results

The repeated measures ANOVA reveals that there is a significant difference between techniques, $F_{2,22} = 4.32, p \ll 0.001$. The mean selection time for GPSel is 0.305 seconds, whereas the means for lasso and rectangle selection are 3.242 and 3.445 seconds, respectively, see Fig. 7-Left. According to a Tukey-Kramer test, only the difference between GPSel and the other two is significant.

There is a strong interaction between Selection Time and Target Length $F_{4,44} = 12.90, p \ll 0.001$. While Target Length had no effect on GPSel, it dramatically increased lasso and rectangle selection time, see Fig. 7-Right.

6 Discussion

The most likely explanation for the results is that GPSel requires considerably shorter pen movement and hence significantly reduces the user's effort. For rectangle selection, the user has to traverse at least the diagonal of the area covered by the target group, while for lasso selection, the user has to traverse at least a major portion of the circumference of the area.

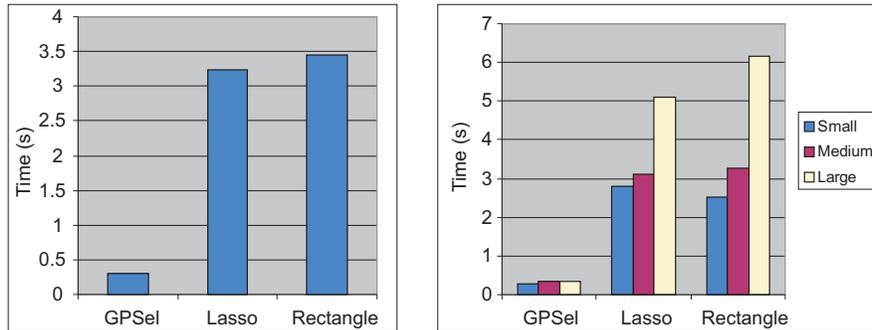


Fig. 7. Left: Comparing average Selection Time among Techniques. Right: The effect of Targets Length on Selection Time for each Technique.

The trial graphs ranged from small to average size. We believe that GPSel can be easily applied to reasonably sized graphs, i.e, graphs that fit on a single screen with readable labels. This is often defined to mean up to approximately 100 nodes [13]. While the computation in GPSel is affected with increasing degrees for nodes, our pilot study shows that the delay introduced by the computation is small enough to be unnoticeable.

7 Conclusion and Future Work

This paper introduces GPSel, a novel perceptual-based approach for path selection in graphs. This technique enables users to select perceptually salient paths by just drawing gestures over nodes and edges. Our user study showed that it outperforms current techniques for selecting complete salient paths.

As future work, we plan to extend the current system to facilitate selection of groups with arbitrary configurations. Additionally, we will do a complete user study to assess GPSel in other scenarios, such as partial and arbitrary paths selection. Finally, we plan to investigate extensions to cover large graphs, directed graphs, and graphs with different vertex types.

References

1. J. Accot and S. Zhai. Beyond fitts' law: models for trajectory-based hci tasks. In *CHI*, pages 295–302, 1997.
2. J. Accot and S. Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *CHI*, pages 73–80, 2002.
3. T. D. Alter and R. Basri. Extracting salient curves from images: An analysis of the saliency network. *Int. J. Comput. Vision*, 27(1):51–69, 1998.
4. G. Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
5. M. Boldt, R. Weiss, and E. M. Riseman. Token-based extraction of straight lines. *IEEE Trans. Systems, Man and Cybernetics*, 19(6):1581–1594, 1989.

6. J. Carriere and R. Kazman. Interacting with huge hierarchies: Beyond cone trees. In *InfoVis*, pages 74–81, 1995.
7. H. Dehmeshki and W. Stuerzlinger. Intelligent mouse-based object group selection. In *Smart Graphics*, pages 33–44, 2008.
8. E. Dengler, M. Friedell, and J. Marks. Constraint-driven diagram layout. In *Visual Languages*, 1993.
9. J. Feldman. Perceptual models of small dot clusters. *Partitioning Data Sets: DIMACS Series in Discrete Math. and Theoretical Computer*, (63):1171–1182, 1995.
10. J. Feldman. Bayesian contour integration. *Perception and Psychophysics*, (63):1171–1182, 2001.
11. J. Feldman. Curvilinearity, covariance, and regularity in perceptual groups. *Vision Research*, 37(20):2835–2848, 97.
12. D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: Evidence for a local association field. *Vision Res.*, 33(2):173–193, 1993.
13. E. Gansner and S. North. An open graph visualization system and its applications to software engineering. *Software Practice and Experience*, 30(11):1203–1233, 2000.
14. Z. Gigus and J. Malik. Detecting curvilinear structure in images. Technical report, EECS Department, University of California, Berkeley, 1991.
15. C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. *Lecture Notes in Computer Science*, 1547:167–182, 1998.
16. T. Henry and S. Hudson. Interactive graph layout. In *UIST*, pages 55–64, 1991.
17. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989.
18. K. Koffka. *Principles of Gestalt Psychology*. Routledge and Kegan Paul, 1935.
19. C. Kosak, J. Marks, and S. Shieber. Automating the layout of network diagrams with specified visual organization. *IEEE Trans. Systems, Man and Cybernetics*, 24(3):440–454, 1994.
20. G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *CHI*, pages 482–487, 1993.
21. K. Lemon, E. Allen, J. Carver1, and G. Bradshaw. An empirical study of the effects of gestalt principles on diagram understandability. In *ESEM*, pages 156–165, 2007.
22. 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.
23. H. Nascimento and P. Eades. User hints for directed graph drawing. In *GD*, pages 205–219, 2002.
24. H. Purchase. Which aesthetic has the greatest effect on human understanding. In *GD*, pages 248–261, 1997.
25. K. Ryall, J. Marks, and S. Shieber. An interactive constraint-based system for drawing graphs. In *UIST*, pages 97–104, 1997.
26. E. Saund. Finding perceptually closed paths in sketches and drawings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(4), 2003.
27. E. Saund and T. Moran. A perceptually-supported sketch editor. In *UIST*, pages 175–184, 1994.
28. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
29. C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.