I-WALKWAYS

an Exploration in Knowledge Visualization

DANIEL C. GLASER
Department of Architecture
University of California, Berkeley

Abstract. This paper describes a prototype which extends a logic system into a useful design tool to aid in designing pedestrian walkways. A highly interactive program, I-Walkways demonstrates how a logic system can meaningfully aid with design. This technique will allow the designer and the logic system to work harmoniously together to reach a good design solution.

1. Introduction

Encoding non-trivial design knowledge into computerized tools has been a difficult task for researchers in environmental design. One approach for building software to aid the designer is with expert or logic systems. These systems are built by encoding design knowledge into first-order logic expressions. In order for a design to be accepted by a particular logic system it must follow each rule in the system. Thus, it is important that the rules that we specify are compatible with our intentions. In addition, our intentions may change dynamically during the design process. Further, it is improbable to describe all of a person's design intentions. Therefore a logic system must also be flexible enough to allow exploration outside of its rules. I-Walkways demonstrates a technique which provides the instruction of a logic system as well as freedom for the designer to explore his or her design world. This work was inspired by a series of papers by Galle and Kovacs (Galle 1992; Galle 1993; Kovacs 1994).

I-Walkways is a tool written in Java to help facilitate good pedestrian walkway design (Glaser 1997). This tool demonstrates how a designer can harmoniously interact with a logic system to reach a good solution. The system visually encodes two customizable evaluators for pedestrian walkway design. One of the evaluators helps the designer minimize the cost of building their walkway network. The other evaluates how straight a path is to get between two points of the walkway network. Although the usefulness of such evaluations may be debatable, I-Walkways attempts to make these guidelines as transparent to the user as possible. Hence a good visualization system will ensure that the rules that we specify are appropriate. Ultimately the designer has to decide how to incorporate this information into their work. As Jones states, "If a decision maker cannot understand a model nor the solution produced by an algorithm, then the model and algorithm are useless" (Jones 1996).

2. The I-Walkways Program

I-Walkways represents a pedestrian walkway network as a graph on a two-dimensional landscape (Figure la). The edges of the graph represent the walkways to be built. A walkway segment has no properties other than where its endpoints are. The endpoints of a walkway segment can either be entry nodes or intermediate nodes. Entry points are places where people can enter the walkway network such as a building or street entrance. They are represented by a square. The designer can add relationships between entry nodes to mark where he or she thinks there will be the most traffic (Figure 1b).

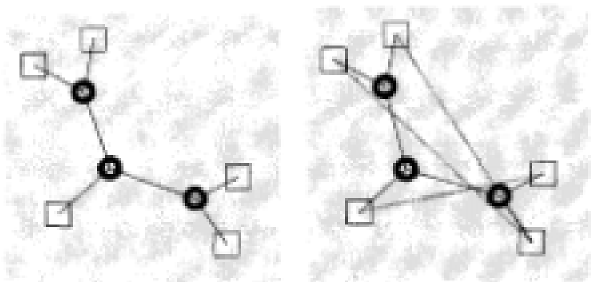

Figure 1. (a) I-Walkways representation of a pedestrian walkway network with five entry nodes and three intermediate nodes (b) after adding predictive main walking lines to the network-

2. 1. THE COST EVALUATOR

A few cost evaluators are encoded to help the designer determine how expensive their proposed walkway network is. They represent a few of many potential ways of describing how much a walkway will cost.

At a local level, I-Walkways can help guide a user to place a node such that its neighboring segments have minimal total length. Although there is no analytical solution for this', I-Walkways guesses this value through the use of adaptive quad-trees. To convince the designer of its accuracy, there is an option to visualize how I-Walkways arrives at this solution. The system then colors the node according to either its relative (in percentage) or absolute (in pixels) difference from the determined minimal cost point (Figure 2).

[1]To find the minimal amount of material to connect an arbitrary number of nodes is NP-Hard. This problem can be reduced to the NP-Hard traveling salesman problem by substituting the cities for the nodes. Hence, at best, we can only find an approximate solution for a reasonable sized problem.

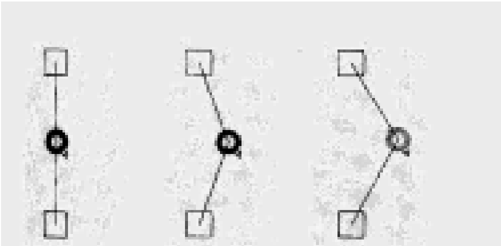

Figure 2. An example of the local cost heuristic being applied to a node being dragged. In the left-most image, the center node starts at a near optimal black state with respect to its absolute walkway cost. In the right-most image it turns cyan to signify that it is at least 25% above its optimal cost.

The system calculates the performance p_{value} of node i by the following formula:

$$p_{value} = \begin{cases} 100 * \dfrac{(C_i - C_o)}{C_o} & \text{if relative performance} \\ C_i - C_o & \text{if absolute performance} \end{cases} \quad (1)$$

where C_i is the current cost and C_0 is the optimal cost of node i. The case where C_0, is 0 is avoided since a node with only one neighbor is defined to have an infinitely high cost. The current cost of the node, C_i, is computed as the summation of the lengths of all of its neighboring walkways:

$$C_i = \sum_{j=0}^{n-1} \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}$$

where n is the number of neighbors of a node x_j is the x-coordinate and y_j is the y-coordinate y_j of neigboring vertex j. The color vector pc_{olor} is thus:

$$p_{color} = \begin{cases} a_{color} & \text{if } p_{value} \le a_{value} \\ a_{color} + \dfrac{p_{value} * \left(b_{color} - a_{color}\right)}{\left(b_{value} - a_{value}\right)} & \text{if } a_{value} < p_{value} < b_{value} \\ b_{color} & \text{if } p_{value} \ge b_{value} \end{cases} \quad (2)$$

where p_{value} is the cost derived from (1), a_{value} and b_{value} is the user definable cost sensitivity range (Figure 3), and a_{color} and b_{color} are two base colors which signify the minimal and maximal values respectively.

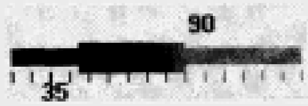

Figure 3. The dynamic cost range bar. Here a_{value}=35, b_{value}=90, a_{color} is black, and b_{color} is cyan. Values 35 and below are black, 90 and above are cyan, and intermediate values are linearly interpolated.

In addition to being a local cost advisor, I-Walkways can help evaluate the cost of larger parts of the network. An intermediate heuristic is used in I-Walkways to assess how costly a path is. In Figure 4, the path connecting the end-nodes which have main walking lines connecting them is colored cyan to signify that it is expensive to build. The coloring of a path is determined similarly as (2) with the exception that the shortest path is used for C1 and the size of the main walking line is used for C0.

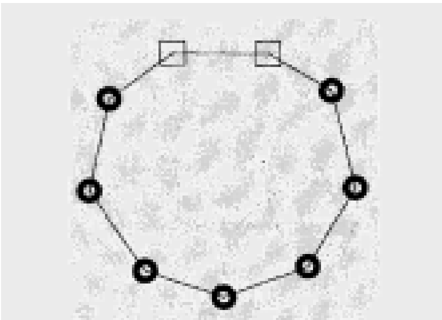

Figure 4. The cost evaluator finds a circumventive path which connects the end nodes of a predictive main walking line. This path is highlighted cyan to indicate that it is an expensive path connecting the main walking lines.

Lastly, I-Walkways reports the global cost for the walkway network. Unlike the previous two modules, this number is not relative to any optimal value. Instead the designer has to rely on comparisons among working and saved solutions to gauge its fitness.

2.2. THE STRAIGHTNESS EVALUATOR

The straightness evaluator helps predict if pedestrians will use the proposed walkways or will take a route off of the designed paths. When modifying a particular node, I-Walkways evaluates how straight the path or paths are which cross a node when modifying it. Although the correlation between straightness and pedestrian behavior "is not without a certain intuitive plausibility, and its unscientific status is hardly atypical of design knowledge" (Kovacs 1993).

Straightness of a node is defined with respect to its local neighbors. I-Walkways draws a set of guide arcs between each pair of neighboring nodes based on two user defined angles, α and ß specified by a range slider similar to Figure 3 (Figures 5a and 5b). This information gets more complex if a node has more than two neighbors (Figure 6). If a node only has one neighbor, a pair of guide lines are drawn to show where the node can be placed and still maintain the straightness quality (Figures 5c and 5d). If the single neighboring node is connected to many other nodes, then I-Walkways draws all the potential straight paths that could be constructed and, if possible, also indicates where the user can place a node to satisfy all the straightness criteria (Figure 6). A coloring scheme similar to (2) is used which compares a to the current node's placement.

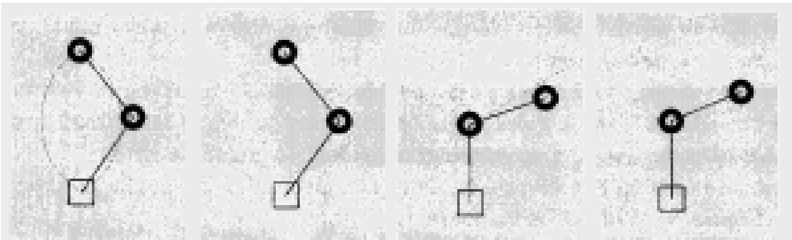

Figure 5. The straightness evaluator for (a) a center node with α=120° (b) α=100° (c) an end-node withα=120° (d) an end-node with α=100°

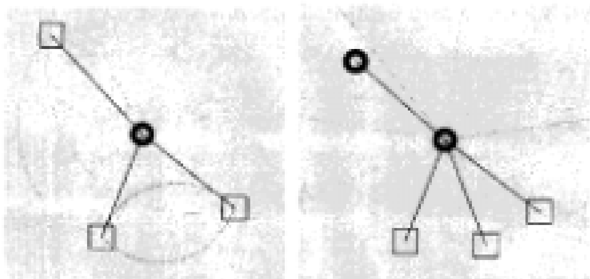

Figure 6. The straightness evaluator (a) advising a node with multiple neighbors (b) advising a node with one neighbor but connecting to many paths. Note the double arc lines in (b) which indicate where all criteria is met.

I-Walkways also checks to see if there is a path among neighboring points to avoid indicating a potential detour where an acceptably straight path already exists (Figure 7).

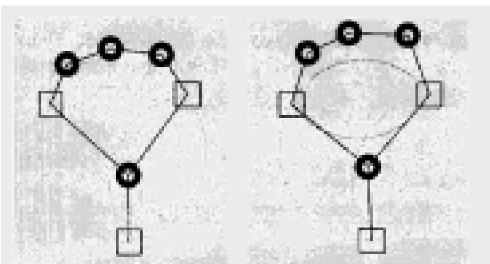

Figure 7. The straightness evaluator applied to the center node (a) with an existing acceptable path and (b) where the existing path between neighbors is worse then the path though the center node.

3. Conclusions

There are a number of practical and conceptual limitations for this work. First, the local cost heuristic may help find a local minimum, but it may be detrimental to the overall cost performance. In addition, the underlying model makes many simplifications. For example, the traffic pattern model is very crude, the landscape homogenous, and the walkways do not have a width or other geometric and material properties. Lastly, other methods can be used to solve this problem. For example, the walkways could be created naturally and then paved after some pedestrian use.

While building and using I-Walkways I have concluded that it will be impossible for a generative tool to adequately solve a design problem similar to this. I have found that I was quite capable of understanding the often graphically rich guidance offered by the software and hence, do not feel that design problems can grow too complex for a human to comprehend (Figure 8). In addition, there were things that these computerized evaluators could never be able to help me with. For example, for understanding the implications of a lake nearby. If I was ambitious enough to build a lake evaluator, I could always think of something else which needs consideration for the design. Good CAAD tools should be useful yet at the same time be quite limited in scope.

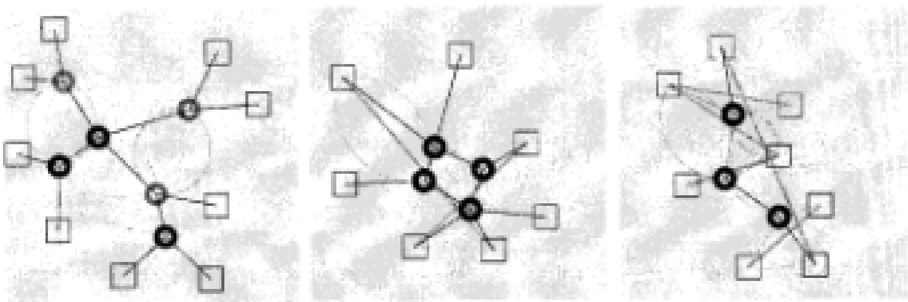

figure 8. Complex walkway networks.

Acknowledgments

I would like to thank Dr. Bruce Land at Cornell for his guidance in the visualization aspects of this project. At Berkeley, Dr. Yehuda Kalay has given me invaluable criticism and support for this work and future directions for it. In addition I would like to thank Dr. Nezar AlSayyad and the CAAD group at Berkeley for their helpful comments.

References

Galle, P., Kovacs, L. (1992) The logic of worms: a study in architectural knowledge representation, *Environment and Planning B,* 19 5-31.

Glaser, D. (1997) I-Walkways, http://www.ced.berkeley.edu/~dglaser/research/iwalkways/, (10 April 1997). Jones, C. (1996) *Visualization and Optimization,* Kluwer Academic, Norwell, MA.

Kovacs, L., Galle, P. (1993) The logic of walking: representing design knowledge on pedestrian traffic nets, *Environment and Planning B,* 20 105-118.

Kovacs, L., Galle, P. (1994) The logic of plaza space: representing design knowledge on shape and function, *Environment and Planning B,* 21 159-177.