Heuristics for placing non-orthogonal axial lines to cross the adjacencies between orthogonal rectangles

Ian Sanders and Leigh-Ann Kenny
School of Computer Science
University of the Witwatersrand
Johannesburg.
E-mail: ian@cs.wits.ac.za

1 Introduction

This research extends on previous work which has its origin in a method, *Space Syntax Analysis* [2], to describe and analyse patterns of architectural space – both at the building and urban level. In the process of applying space syntax analysis [3] the space (roads, parks, etc.) in the town is partitioned into the smallest number of convex polygons to cover the whole town. Then the axial lines (the fewest number of longest line segments) are placed to cross all of the adjacencies (shared edges) between the convex spaces. An axial map consists of the minimum number of "axial lines" necessary to cross the adjacencies between the convex polygons which make up the convex map. In this research any adjacency *must* be crossed at least once but may be crossed more than once. An additional requirement is that any axial line should cross as many adjacencies as possible.

Sanders *et al.* [6] considered a restriction of the general problem – the problem of placing the fewest orthogonal axial lines which cross all of the adjacencies in a configuration of nonoverlapping orthogonal rectangles (***ALP-OLOR***) – and showed this variation to be NP-Complete by a transformation from planar vertex cover. Sanders [4] then considered the problem of placing the fewest non-orthogonal axial lines which cross all of the adjacencies in a configuration of nonoverlapping orthogonal rectangles (***ALP-NLOR***). ***ALP-NLOR*** was also shown to be NP-Complete using a similar transformation to that discussed in Sanders *et al.* [6] for orthogonal axial lines.

In this paper the emphasis is on heuristics to find axial lines to cross the adjacencies between orthogonal rectangles – finding heuristic solutions to ***ALP-NLOR***. A problem which arises whether an exact solution is being calculated or a heuristic is being applied to find an approximate solution is to be able to efficiently decide whether an axial line can be placed to cross a given set of adjacencies. The next section (Section 2) addresses this problem before some heuristics which are assumed to use the idea proposed here are presented (Section 3). In Section 4 some results of testing one of the heuristics are briefly discussed. Some ideas for further research are discussed in Section 5.

2 Placing axial lines in chains of adjacent rectangles

When placing orthogonal lines to cross the adjacencies between orthogonal rectangles it is very easy to determine whether an axial line crosses a number of adjacencies. If the adjacent rectangles which give rise to the adjacencies are treated as a chain of rectangles (i.e. the rectangles are adjacent and each rectangle has at most one left neighbour and one right neighbour) and if there is an overlap in the y-values of the adjacencies then an axial line can be placed to cross those adjacencies. Note that, an axial line is actually a representative of a set of lines all of which fall within the same range of y-values and cross the same adjacencies between rectangles.

In the case of nonorthogonal lines and orthogonal rectangles any axial line could cross both vertical and horizontal adjacencies and these thus have to be considered at the same time. This makes determining whether an axial line can be placed to cross all of the adjacencies in a "chain" of adjacent rectangles more complicated. The approach taken in this paper is to convert the problem of determining whether an axial line can be placed to cross the adjacencies in a given chain of rectangles to the problem of determining partial visibility between two edges of a particular polygon. This can be done by converting the chain of rectangles into an "adjacency polygon" and then making use of an approach suggested by Avis *et al* [1] for generating the "Inner Convex Hull" of the adjacency polygon. This is done by finding the inner convex hulls of the top chain and the bottom chain respectively and using these to create the "reduced" top and bottom chains (Figure 1). If it were possible for an axial line to be placed to cross all of the adjacencies in the chain of rectangles then there

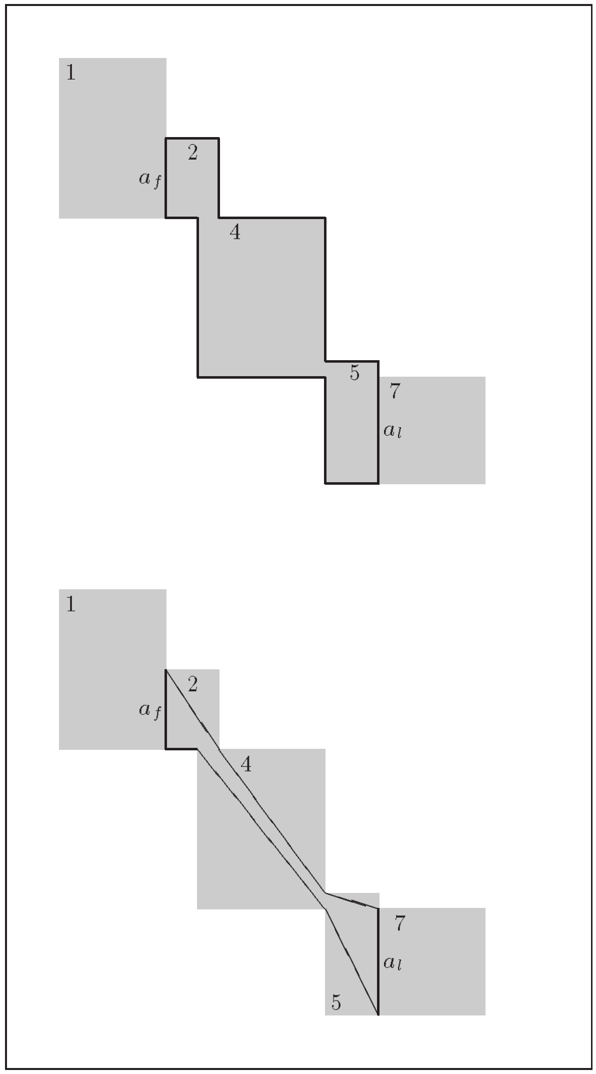

Figure 1: Converting an adjacency polygon into a reduced adjacency polygon

will be partial edge visibility between the two edges of the reduced adjacency polygon which were the first and last adjacencies in the chain (edges a_f and a_l in Figure 1).

3 Heuristics for *ALP-NLOR*

ALP-NLOR is NP-Complete ([4]) and it is thus necessary to consider heuristics to find approximate solutions in reasonable time. In this section some ideas for heuristics are presented.

3.1 Extending lines into all neighbours

In Sanders *et al.* [5] an $O(n^2)$ algorithm to return a non-redundant set of maximal orthogonal axial lines for the problem of placing orthogonal axial lines to cross the adjacencies between orthogonal rectangles ***ALP-OLOR*** was presented. This algorithm has four phases (after determining which rectangles are adjacent). It generates all the possible orthogonal axial lines which cross the adjacencies between rectangles; it determines which axial lines are essential (i.e. are the only lines which cross a particular adjacency); it removes any lines which only cross adjacencies crossed by the essential lines (redundant lines); and then it resolves the choice conflict. The resolving of the choice is done by repeatedly choosing the choice line which crosses the highest number of previously uncrossed adjacencies. One possible approach to finding an approximation to ***ALP-NLOR*** is to adopt a similar approach to that used before. The main difference would be that the restriction on the axial lines being orthogonal would be removed – lines with arbitrary orientation would be allowed.

3.2 Separating top-bottom and left-right lines

Another heuristic could be to calculate a solution in two passes. First working from the leftmost rectangle to the rightmost rectangle and only extending lines which enter the rectangle being considered from the left into its right neighbours (if this can be done). This is analogous to the solution for ***ALP-OLOR*** except here the lines need not be orthogonal. Next a pass is made from the highest rectangle to the lowest rectangle. This second pass extends lines which enter the rectangle being considered from the top into its lower neighbours (if this can be done). Each of these passes would (as before) generate all possible lines (while obeying the left-right or top-bottom constraint); determine essential lines; remove redundant lines; and resolve any choice. These two passes then generate two sets of lines – a set which crosses the vertical adjacencies and a set which crosses the horizontal adjacencies. Combining the final two sets of lines would give an approximation to ***ALP-NLOR***. This approach would have the advantage of being efficient – for any rectangle it would only be necessary to determine if it was possible to extend any lines passing through it into neighbours on one side. The approach is, however, likely to give a poor approximation as lines in an optimal set of lines are likely to cross both horizontal and vertical adjacencies.

3.3 Longest Chains

A different approach for finding an approximate solution for ***ALP-NLOR*** could be to attempt to find the longest possible axial line at any stage of the algorithm. Clearly to do this would require first generating all possible lines which is inefficient. An approach which goes some way towards using this idea is to identify "extreme" rectangles, generate the chains which include these rectangles and then successively choose the longest of such chains until all of the adjacencies have been crossed. The motivation for doing this is that lines which cross the adjacencies of these extreme rectangles could potentially cross a number of other adjacencies as well and so might be a good "greedy" choice.

3.4 Crossing one adjacency at a time

Another idea is to consider crossing each as yet uncrossed adjacency in turn with a line which is "as long as possible". The approach here works by starting with the two rectangles which define the leftmost as yet uncrossed adjacency and successively adds rectangles to this chain until adding another rectangle would mean that no straight line could cross all of the adjacencies in the chain because the rectangle being considered is on the same side of the last rectangle in the chain as the penultimate rectangle. These last two adjacencies would be colinear. If a line can be placed to cross the adjacencies in the chain then this line is added to the final set of lines. If a line cannot be placed to cross the all of the adjacencies in the chain then a different chain is generated starting with the same adjacency and following the same constraints. If a line cannot be placed through *any* chain starting at that adjacency then make a line which crosses just that adjacency. The resulting set of lines is an approximation to the solution for ***ALP-NLOR***

3.5 Extending forwards and then backwards

This approach is similar to the approach suggested above. Again for each adjacency that is still uncrossed a new chain working from left to right is constructed. When the chain "wraps around" the forward phase is complete and it is necessary to determine whether a straight line can be placed in the chain. If a line can be placed then the next stage of the process – extending the chain to the left – is done. However, if a line cannot be placed then one rectangle at a time will be removed from the right hand side of the chain until a line can be placed or until there are only two rectangles in the chain. These two remaining rectangles are adjacent to each other and clearly a line can be placed to cross the adjacency between them.

This stage ensures that the lines extend as far as possible to the right. However, it is possible that they don't extend as far to the left as possible. The second stage of the heuristic addresses this. The chain is now extended backwards (to the left) as far as possible and then a similar process of testing for line placement and removing rectangles from the chain is followed.

The resulting line is "as long as possible" and is thus included in the final set of lines.

4 Empirical testing

The heuristic in Section 3.5 has been implemented and tested on some configurations of adjacent rectangles. The heuristic seems to offer a reasonable approximation to the exact result in the cases tested. There are, however, some cases where the heuristic produces redundant lines – lines which only cross adjacnecies which are crossed by other lines generated later in the process. The removal of redundant lines could be achieved by a post processing phase – identifying "essential lines" (axial lines that cross some adjacencies that are not crossed by any other axial lines) and then removing the redundant lines. A more significant problem with this heuristic is that it is biased in favour of crossing vertical adjacencies (the first rectangle considered to extend a chain of rectangles is a right neighbour) and this can affect the accuracy of the approximation. Some ideas for improving the heuristic's performance are being investigated.

5 Future Research

Further work in this area can be focussed on

- developing more, and hopefully better, heuristics to solve the problem,
- testing of the heuristics presented here and any new heuristics which are developed,
- determining bounds on the approxomations given by the heuristics presented above,
- considering the use of genetic algorithms to give approximate solutions to the problem,
- and looking at special cases for which polynomial time solutions can be found.

6 Conclusion

This paper has suggested some heuristics which could be used to produce approximate solutions to ***ALP-NLOR***. Some preliminary testing has been done on one of these heuristics. Other ideas for heuristics are currently being investigated.

References

[1] David Avis, Teren Gum, and Godfried Toussaint. Visibility between two edges of a simple polygon. *The Visual Computer*, (2):342–357, 1986.

[2] B. Hillier, J. Hanson, J. Peponis, J. Hudson, and R. Burdett. Space syntax, a different urban perspective. *Architects' Journal*, 178:47–63, 1983.

[3] G. Mills. The spatial structure of ideology in informal settlements: A case study in southern africa. *Building and Environment*, 27(1):13–21, 1992.

[4] I. D. Sanders. Non-orthogonal ray guarding. In *Abstracts for the Eleventh Canadian Conference on Computational Geometry*, pages 80–83. University of British Columbia, Vancouver, August 1999. The full paper is available in the electronic proceedings – `http://www.cs.ubc.ca/conferences/CCCG/elec_proc/elecproc.html`.

[5] I. D. Sanders, D. J. Lubinsky, and M. Sears. Ray guarding configurations of adjacent rectangles. In *Proceedings of the 25th Annual Southern African Computer Lecturers' Association Conference*, pages 104–116. Rhodes University, July 1995.

[6] I. D. Sanders, D. J. Lubinsky, M. Sears, and D. Kourie. Orthogonal ray guarding of adjacencies between orthogonal rectangles. *South African Computer Journal*, (23):18–29, 1999.