

# Metro Maps on Flexible Base Grids

## ABSTRACT

We show that a previously developed method of drawing metro-maps on octilinear grid graphs can be used to draw metro maps following arbitrary base grids. We experiment with several types of such base grids: ortho-radial grids, octilinear quad trees, octilinear Hanan grids (which we introduce in this work) and TODO. Our interest is threefold: first, we want to speed up the original approach by using a simplified base grid which still guarantees octilinearity. Second, we are interested in the capability of the original approach to render maps which are not octilinear, but for example ortho-radial. Last, we would like our approach to automatically adapt to the local node densities of the input graph. We also show how quad trees can be used to enlarge high-density areas into otherwise empty areas.

### ACM Reference Format:

. 2021. Metro Maps on Flexible Base Grids. In *Proceedings of Some conference*. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Drawing generalized metro maps is a well-studied topic. Usually, the goal is to produce octilinear maps in which all edge segments have an orientation which is a multiple of 45 degrees. In [? ], an approach was described to render such maps with an arbitrary number of edge bends by routing the edges of the input graph in a special octilinear grid graph (the base grid) in which line bends were penalized. This approach can be optimized exactly via an Integer Linear Program, or by a fast approximation method. In this work, we want to study how this approach behaves when different kinds of base grids are used. Our experiments will show the following: (1) In the context of octilinear drawings, the size of the octilinear base grid graph can be greatly reduced with only minimal impact on the solution quality. To achieve this, we introduce the concept of an octilinear Hanan grid. This is motivated by the well-known fact that a (regular) Hanan grid for a set of points  $P$  contains an optimal rectilinear Steiner Tree for  $P$ . (2) The approach is able to render ortho-radial, triangular or hexagonal metro maps with only minor modifications by using corresponding base grids. (3) If quad trees are used as base grids, the metro map density can be dynamically adjusted to local input graph densities. Quad trees can also be used to enlarge high-density into neighboring low-density areas.

We use the following notation: Our input is an undirected labeled graph  $G = (V, E, L)$ .  $V$  may be understood as stations but may also contain non-station intersections. The edges  $E$  are connections between nodes labeled with lines  $L(e) \subseteq \mathcal{L}$  traveling through them.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*Some conference, Some date, Somewhere*

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

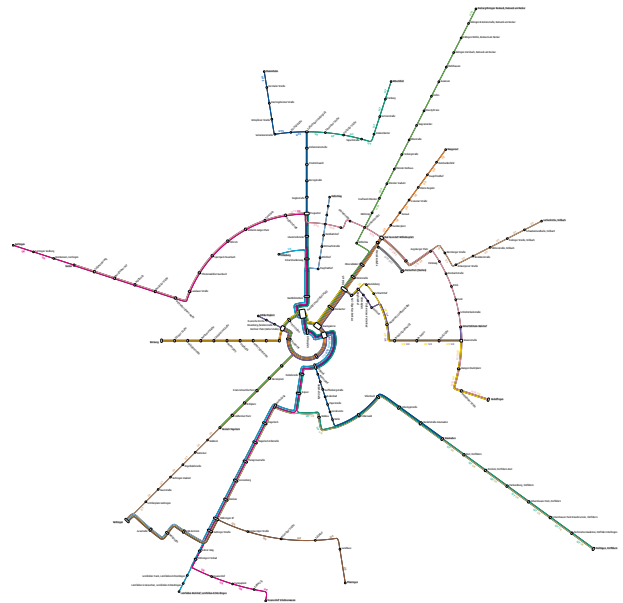


Figure 1: The Stuttgart light rail network rendered in an ortho-radial fashion by our approach.

We then search for a drawing  $\mathcal{D}(G)$  of  $G$  that resembles a metro map.

### 1.1 Related Work

[describe the original approach] [other related work, especially recent work on orthoradial metro maps, but also bezier curve based maps]

### 1.2 Metro Maps on Grid Graphs

The method described in [? ] first builds an octilinear grid graph with the size of the bounding box of the input graph  $G$ . The grid cell size can be chosen freely. In [? ], it was observed that the average distance between adjacent nodes in the input graph works well in practice. The optimal drawing is then constructed by routing all input edges in this octilinear grid graph, with the restriction that no routed edge may cross another routed edge. The assignment of input nodes to grid nodes is unrestricted, but a larger geographical distance to the original position is penalized. To be able to penalize line bends in the final drawing, grid nodes were extended by so-called port nodes, which made it possible to add penalty costs to bends. The modelling of these line bend costs is similar to the modelling of turn costs in transportation networks, but does not require a directed graph. Additionally, the circular ordering of outgoing edges in the input graph were required to remain the same in the final drawing.

As an ILP based solution to compute the globally optimal drawing proved to be too slow, an approximate approach was developed in which edges we routed explicitly through the grid graph by Dijkstras algorithm. As a polishing step, a local search was added.

[mention missing enlargement of dense areas]

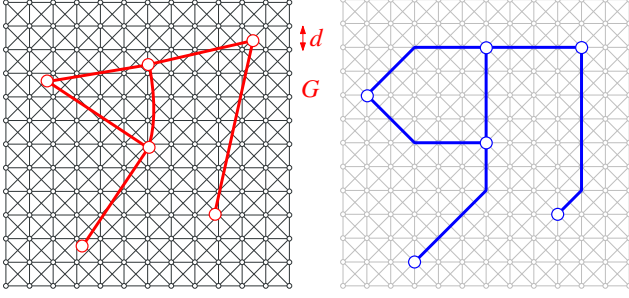


Figure 2: TODO

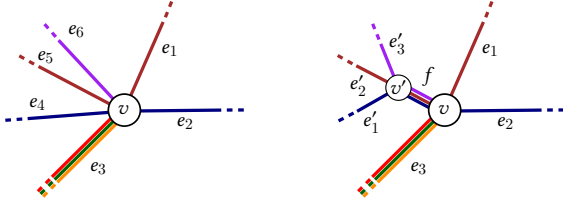


Figure 3: Left: Node  $v$  in an input line graph has a degree of 6, making it impossible to render the graph in an orthoradial fashion. Right: We keep the first (in clockwise order) 3 adjacent edges of  $v$ , combine the lines of the remaining edges  $e_4, e_5$  and  $e_6$  into a single new edge  $f$  and connect it to a new non-station node  $v'$ .

## 2 NODE SPLITTING

Naturally, octilinear metro maps can only be drawn for input graphs with a maximum node degree less than 9. Real-world transportation networks typically satisfy this requirement. However, orthoradial metro maps require a maximum node degree of 4. Triangular maps require a maximum node degree of only 3. As node degrees larger than 4 are still common in real-world transit networks, this problem has to be addressed prior to rendering such maps.

A simple approach is depicted in Figure ?? . Given a maximum node degree  $D$ , a node  $v$  with  $\deg(v) > D$  and incident edges  $e_1 \dots e_{\deg(v)}$ , we remove edges  $e_{D+1} \dots e_{\deg(v)}$  from  $v$  and add them to the new node  $v'$ . An additional edge  $f$  is added between  $v'$  and  $v$  with  $L(f) = \bigcup_{e_{D+1} \dots e_{\deg(v)}} L(e)$ . If we still have  $\deg(v') > D$ , we repeat to process for  $v'$ . Note that while Figure ?? depicts  $v'$  with a different position than  $v$ , this position will remain the same as  $v$  in the modified input graph to not distort displacement penalties during the metro map rendering.

## 3 OCTILINEAR HANAN GRID

Given a set of two-dimensional points  $P$ , the Hanan grid  $H(P)$  is a graph that can be constructed in the following way: (1) Draw vertical and horizontal lines through each  $p \in P$ . (2) Where these lines intersect, add vertices. A Hanan grid  $H(P)$  contains a minimum rectilinear Steiner tree for  $P$ . As rectilinear Steiner trees are slightly related to rectilinear Metro Maps, it seems worthwhile to evaluate their applicability to grid-based Metro Map drawing. In particular, given the relative sparsity of a Hanan grid when compared to a

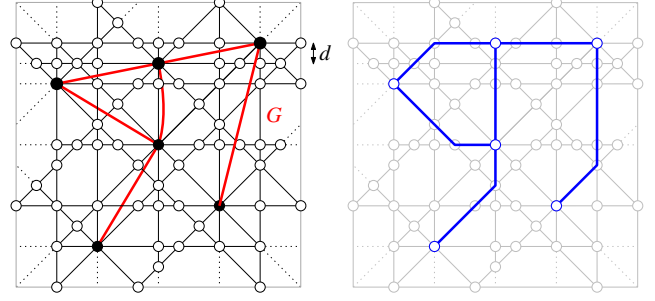


Figure 4: Left: a bounding-box restricted octilinear Hanan grid for an input graph  $G$ . All depicted nodes are part of  $O(G)$ , but only the black nodes were part of the input graph  $G$  and have been snapped to a base grid with cell size  $d$  prior to construction of  $O(G)$ . Edges of  $G$  are depicted in red. Right: an octilinear drawing of  $G$  on  $O(G)$ .

full grid, they have the potential to speed up the time needed to generate such drawings.

As rectilinear Metro Maps are of only little interest, we slightly extend the definition of a classic Hanan grid and introduce the concept of octilinear Hanan grids. Given a point set  $P$  on the plane, the construction of an octilinear Hanan grid  $O(P)$  is similar to a classic Hanan grid, but we also draw diagonal lines and consider them for possible intersection points. Obviously, all paths in  $O(P)$  will only consist of segments with an octilinear orientation.

### 3.1 Base Grid Snapping

As we would like to maintain a minimum segment length in the final maps, we do not construct the octilinear Hanan grid directly on the input graph  $G$ , but first snap its node positions to a regular base grid.

### 3.2 Hanan Iterations

An octilinear Hanan grid  $O(P)$  with nodes  $V$  can be made increasingly more fine-grained if we take the nodes  $V$  as points  $P'$  and construct an octilinear Hanan grid  $O(P')$  from them. We call such a step a Hanan iteration.

[\[Show example iterations\]](#)

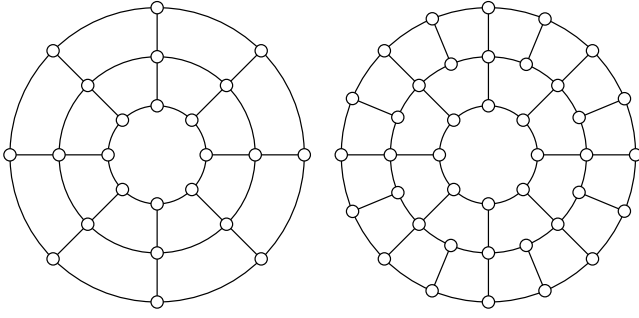
### 3.3 Path Cost Preserving Weights

## 4 QUAD-TREE

A quad tree is a grid which can be constructed for a set  $P$  of input points in the following way: start with the bounding box of  $P$  as a single-cell grid. While any cell has more points in it than a threshold value  $t$ , split this cell into 4 rectangular cells. The resulting grid has a graph-like structure, where each grid cell of size  $d$  may have 4 direct child vertices of size  $d/2$ .

## 5 ORTHO-RADIAL GRID

Recent work discussed the problem of drawing Metro Maps in an ortho-radial fashion [? ]. While octilinear Metro Maps are much more common than ortho-radial maps, the latter can still be found



**Figure 5: Two kinds of ortho-radial grid graphs. Left: Ortho-radial grid graph with  $b = 8$  and a central node. Right: Ortho-radial grid graph where  $b$  is doubled each time the radius doubles.**

and have an intriguing visual quality [\[example\]](#). In [\[?\]](#), it was observed that an ortho-radial map can be understood as a flat projection of an octilinear map which is joined at two parallel sides. Figure 5, left depicts a classic ortho-radial grid graph. An obvious problem with such a grid are the large white-space areas, as the grid cell size grows with increasing distance to the grid center. To overcome this, we use a slightly altered ortho-radial grid graph, as depicted in Figure 5, right. The innermost ring starts with a number of nodes  $b = 8$ , but this number is doubled each time the ring radius doubles. So at ring 1,  $b_1 = 8$ , at ring 2,  $b_2 = 16$ ,  $b_4 = 32$ ,  $b_8 = 64$  and so on. This will give our maps a denser look while still maintaining the visual impression or ortho-radiality.

## 6 EXPERIMENTAL EVALUATION

### 6.1 Effect of Octilinear Hanan Grid on Solution Times and Quality

### 6.2 Effect of Quad Trees on Solution Times and Quality

### 6.3 Ortho-Radial Experiments

## 7 CONCLUSION AND FUTURE WORK

[\[Mention local enlargement prior to ortho-radial drawing\]](#)