PST Documentation

PST is developed by KTH School of Architecture, Chalmers School of Architecture (SMoG) and Spacescape AB.

The latest released version of PST at the time of writing is 3.2.4.

The application is available via: https://github.com/SMoG-Chalmers/PST/releases/tag/v3.2.4
Download: pstqgis_3.2.4_2023-01-10.zip

Documentation, data and tutorials are available via KTH and Chalmers:
https://www.smog.chalmers.se/pst
https://www.arch.kth.se/forskning/urban-design/software

PST Documentation

1. Overview

PST is an open source tool for performing spatial analyses. It combines the space syntax description of the urban environment with conventional descriptions of attraction into a combined accessibility analysis tool. It is currently available as a plugin for QGIS, an open source GIS software. Earlier PST versions are also available as a plugin for the MapInfo Professional GIS software.

PST is developed by KTH School of Architecture, Chalmers School of Architecture (SMoG) and Spacescape AB. Alexander Ståhle, Lars Marcus, Daniel Koch, Martin Fitger, Ann Legeby, Gianna Stavroulaki, Meta Berghauser Pont, Anders Karlström, Pablo Miranda Carranza, Tobias Nordström.

The software was first introduced in a paper by Ståhle, A., Marcus, L. and Karlström, A., (2005), "Place Syntax: Geographic accessibility with axial lines in GIS", published in the proceedings of the 5th Space Syntax Symposium held in Delft, the Netherlands.

1.1 Technical details

The latest released version of PST at the time of writing is 3.2.4.

PST for QGIS consists of two parts. The first part is written in Python and implements the user interface and all communication with QGIS. The second part is written in C++ and implements the analyses, algorithms and heavy calculations done by PST.

1.2 File formats

PST runs with Mapinfo TAB (*.tab) and Shapefiles (*.shp).

2. The PST Graph

PST uses three geometric features: axial/segment lines, unlinks and attractions.

- Axial/segment lines are the representation of the street network used for the analyses. PST only supports line objects as network input (to create a line-only map, see section 5.1, 5.2).Both an Axial map and a Line-segment map can be used as input. A Line-segment map can be created directly in PST, either from an Axial map or a Road-centre-line map (see sections 5.1, 5.2). The *Axial map* is colloquially defined as the least amount of straight lines that cover all accessible urban space shaped by built form, where each straight line (axial line) represents an urban space that is possible to visually overlook and directly access. The Road-centre-line map, is a GIS-representation of the street network, where the geometric features are polylines representing street segments which span between street junctions.
- Unlinks are point features that specify the intersection points between axial/segment lines that should be ignored during the analyses (i.e. tunnels and bridges, overpasses and underpasses).
- Attractions specify points or areas of interest such as addresses, buildings or properties. They can be used as either origins or destinations in the Attraction analyses (see sections 5.6, 5.7 and 5.8).

All analyses start by creating a graph of the network using the following steps:

1. PST creates a network graph from the axial/segment lines. In that graph, every axial/segment line is a node and every intersection is an edge. When axial lines are used, intersections are defined as points where two or more lines intersect. When segment lines are used, intersections are defined as points where two or more lines meet (i.e. endpoints).[1] Figure 1

2. All Unlink points are traversed. Each Unlink point is compared against all intersection points in the axial/segment line network and the closest intersection found is removed from the graph (effectively disconnecting the intersecting lines).[2] Note, that in case of a Segment map the lines should cross and not meet at the points of the Unlinks, in order for them to be effectively disconnected. Figure 2

1. The implementation uses a space partitioning tree to find intersections between lines efficiently.
2. Since unlinks are usually not that many, this step is currently implemented using a brute force algorithm of complexity O(m*n).

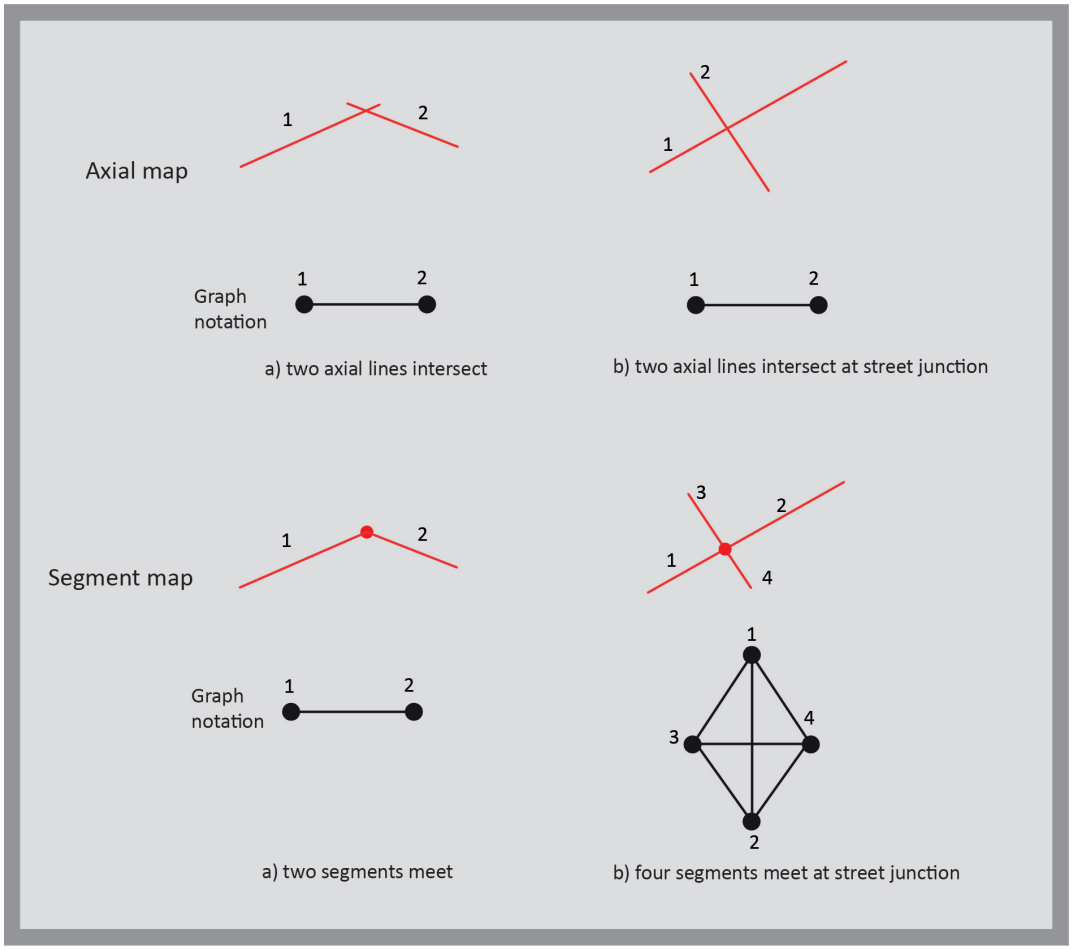

Fig 1. Graph notation of Axial and Line-segment map

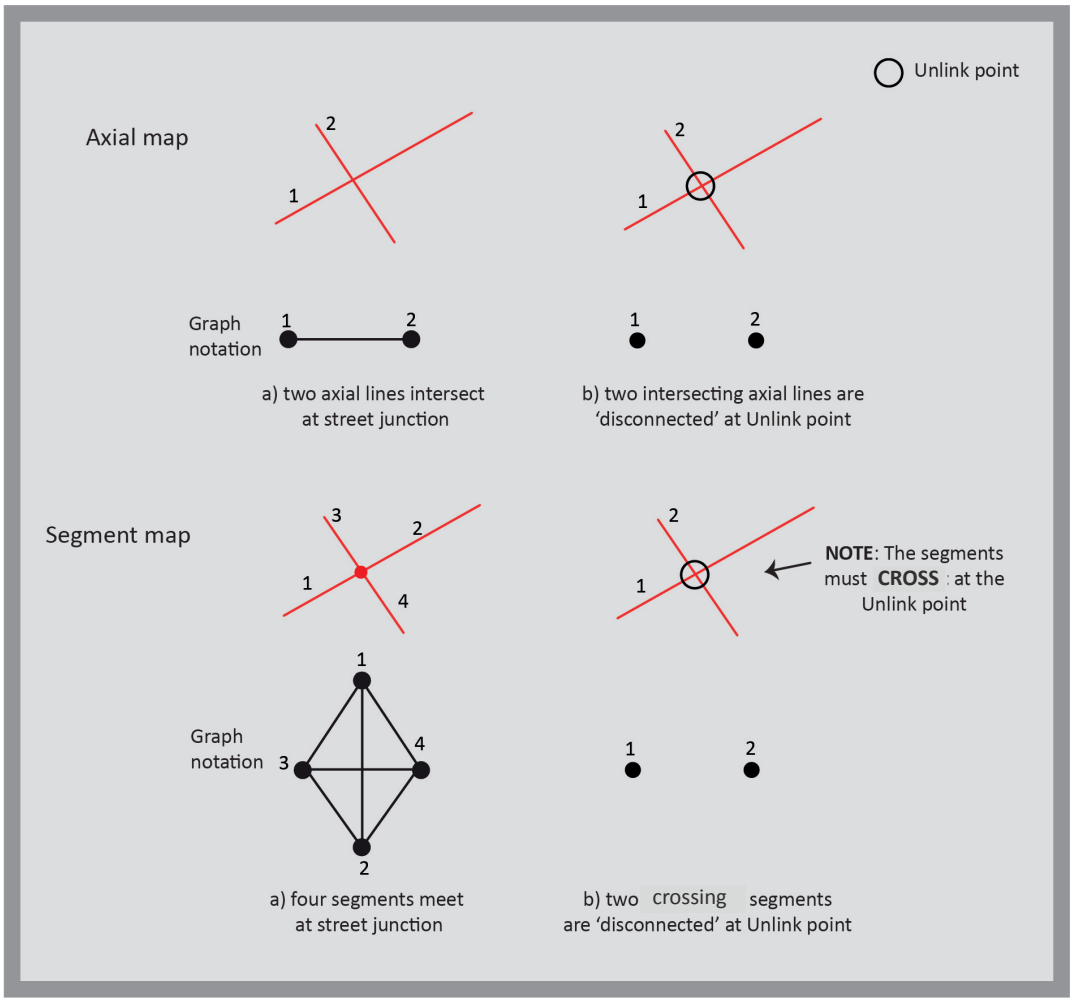

Fig 2. Representation of 'Unlinks' in Axial and Line-segment map

3. The final step (which is only applicable for Reach and the Attraction analysis, see sections 5.1, 5.6, 5.7 and 5.8) traverses all attraction points. Every attraction point is connected to the closest axial/segment line[3] and the distance of these connections will be incorporated in the analyses. PST also allows polygons or lines to be used as attractions. In the case of polygons, the user has to choose between either using the centroid of each polygon or to generate points along the edges of the polygon. The points are then connected in the same way as regular attraction points. When generating points along polygon edges the user can decide if the data associated with the polygon should be 'copied' to every generated point or 'divided' equally among them. For instance, in the first case, if two out of ten generated points of the polygon are reached form an origin, the result will be one reached destination. In the secont case if two out of ten generated points are reached, the result will be 0,2 reached destination. When lines are used as origin or destination the distance is measured from their mid-points when distance is measured as straight line distance or walking distance (see section 3).

3. Distance mode

PST allows the user to choose among different ways of measuring distance in its analyses. The supported ways are:

- Straight line distance: Metric distance of the straight line connecting two points, as if there were no obstacles between them (i.e. Euclidean distance).
- Walking distance: Metric distance of the shortest 'walking' path connecting two points through the network, as defined in section 2 (following the axial or segment lines). Figure 3
- Axial/Segment steps: Topological distance or Depth between two nodes in the network. First the shortest topological path between two nodes is selected, that is the least set of edge-edge connections needed to get from node A to node B. Then the number of Steps in the shortest path between node A and node B is counted. It can also be defined as the number of Axial or Segment lines one has to pass to get from line A to line B, following the shortest path. Figure 4
- Angular distance: Accumulated angular turns needed to get from point A to point B in the network, as defined in section 2. First, the angular distance of all possible paths between A and B are calculated and the one with the least Angular distance is selected as the Shortest path. Angular distance between A and B is then calculated based in the Shortest path. Angular distance is measured in degrees and then divided by 90 (Hillier and Iida, 2005). Note that Angular distance should only be used with Segment maps. Figure 5
- Axialmeter: An experimental measurement that aims at combining axial/segment steps and walking distance, calculated as the product of the two (steps*m).

Note that in all cases, the distance between point A and point B is measured following the Shortest path that connects them. First, the distance of all possible paths which connect A and B are calculated, using the specific Distance mode. The path with the minimum calculated distance is selected as the Shortest path, and all analyses are based on that.

4. Radius

PST allows the user to choose among different ways of setting a radius for limiting a search in all analyses. The supported ways of setting radius in PST are similar to distance mode: Straight line distance, Walking distance, Axial/Segment steps, Angular distance, Axialmeter (for description on how they are measured see section 3).

The user can choose different options for Distance mode and Radius. For example, one can choose Angular distance in the Distance mode options, but set the Radius for the analysis based on a walking distance.

The user can use different radii at the same calculation, in which case the search will stop when the first limit is reached. For example, one can choose Walking distance of 1km and Topological distance (Axial/Segment steps) of 3 steps. The search will stop, for example, when it reaches 3 steps even if the Walking distance is less than 1km.

3. The closest line is defined either by measuring the perpendicular distance of the attraction point to a line or by measuring the distance of the attraction point to the closest end-point of a line, depending on whether the perpendicular projection of the attraction point falls on the line or not.

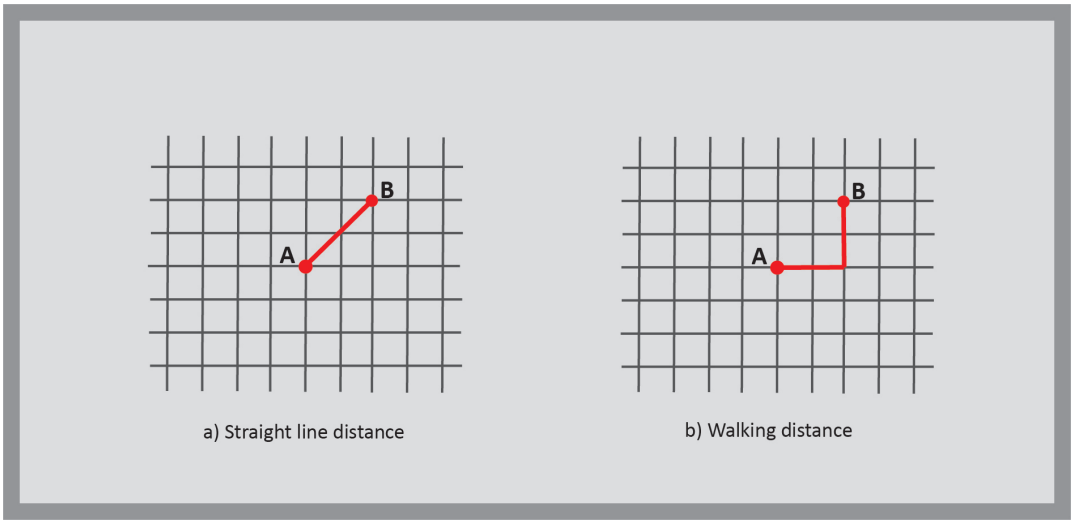

Fig 3. Straight line distance and Walking distance

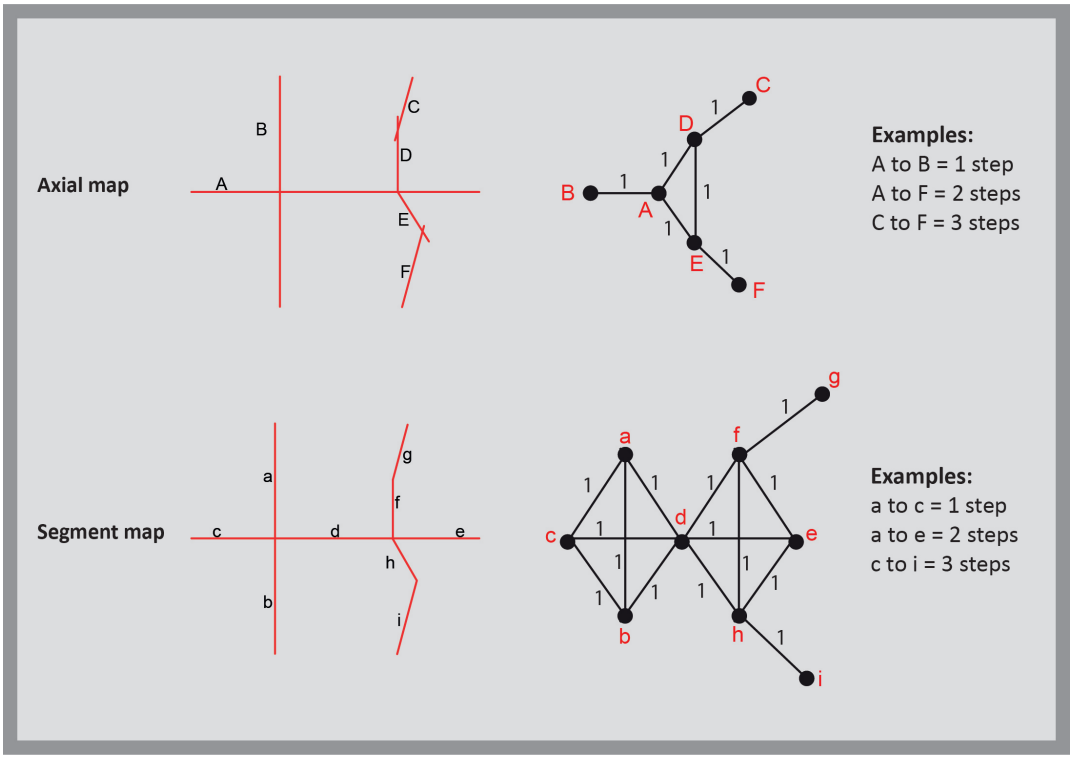

Fig 4. Distance in Axial and Segment steps

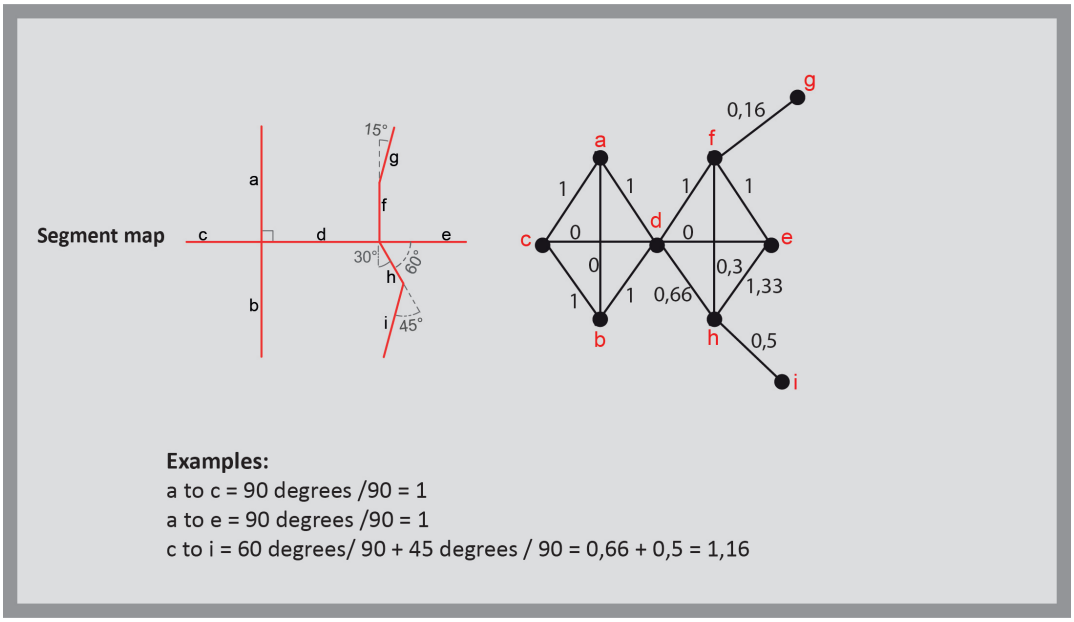

Fig 5. Angular distance in Segment maps

5. Tools

5.1 Split Polylines

This tool splits each polyline into line segments, so that it can be used in PST. The process is straightforward – all objects in the specified input table are traversed and for every segment of every encountered polyline a new row with a line object is added to the new table. If a regular line object is encountered in the input table, it is simply copied to a new row in the output table. The tool has also an option for copying one of the columns of the input table to the output table. This makes it possible to associate the new lines with their original polylines, by for example choosing to copy the ID column. The new line-only table will be created with the same name as the original, but with suffix "_lines" added.

The "Split polylines" tool is useful when Road-centre-line maps are used that consist of both polylines and lines.

5.2 Create Segment Map

This tool converts a map that consists of lines and/or polylines into a line-segment map. The input can be an Axial map, a Segment map or a Road-centre-line map. The user selects between two options: Axial/Segment map or Road Segment Line map. The main difference is that when the input is a Road Centre Line map, the tool creates both the line-segment map and a map of Unlink points, identifying the points of grade separation (i.e. bridges, tunnels). When the input is Axial/Segment map then the user must provide the Unlink point layer.

The conversion procedure is the following:

FOR INPUT: AXIAL / SEGMENT MAP
1.Find and create a list of all intersections between all pairs of lines and polylines.
2.If a table of unlink points is supplied, then for each unlink point find the closest intersection and remove it from the list.
3.For each remaining intersection in the list, split its lines at the intersection point.

FOR INPUT: ROAD CENTRE LINES
1. Find points in the RCL where lines cross and not meet (i.e. points of grade separation).
2. Output a 'point' layer of the points found in step 1 and name the layer 'unlinks'.

The following editing procedures are applied for all input maps:
1.Snap together all line end-points that are closer than a specified threshold. The threshold is defined by the user.
2.Remove duplicate lines (lines sharing same end-points).
3.Remove lines of zero length (snapping might produce lines of zero length which need to be removed).
4.Remove Tail lines using a specified threshold defined by the user. Tail lines are defined as all lines on the path from a dead-end to the closest junction (a point connecting three or more lines). Lines are removed one-by-one, starting with the dead-end line, until the total accumulated length of the removed lines has reached the specified threshold. This function is especially useful when converting an Axial map into a Line-segment map, a procedure that creates a lot of tail segments.
5.Merge two connected lines A and B into one line if the collinear[4] deviation of the three points that make up the two lines is lower than a specified threshold defined by the user. Merging is not allowed if lines A and B originate from separate polylines, or if they connect at a junction. This function is especially useful when converting a Road-Centre-line map into a Line-segment map, a procedure that creates a lot of small almost collinear segments. Figure 6 and 7

The tool has an option for copying one of the columns of the original table to the output table. This makes it possible to associate the lines of the segment map to their original lines/polylines, by for example choosing to copy the ID column.

4. Three or more points are said to be collinear if they lie on a single straight line. The "straightness" is defined by the threshold.

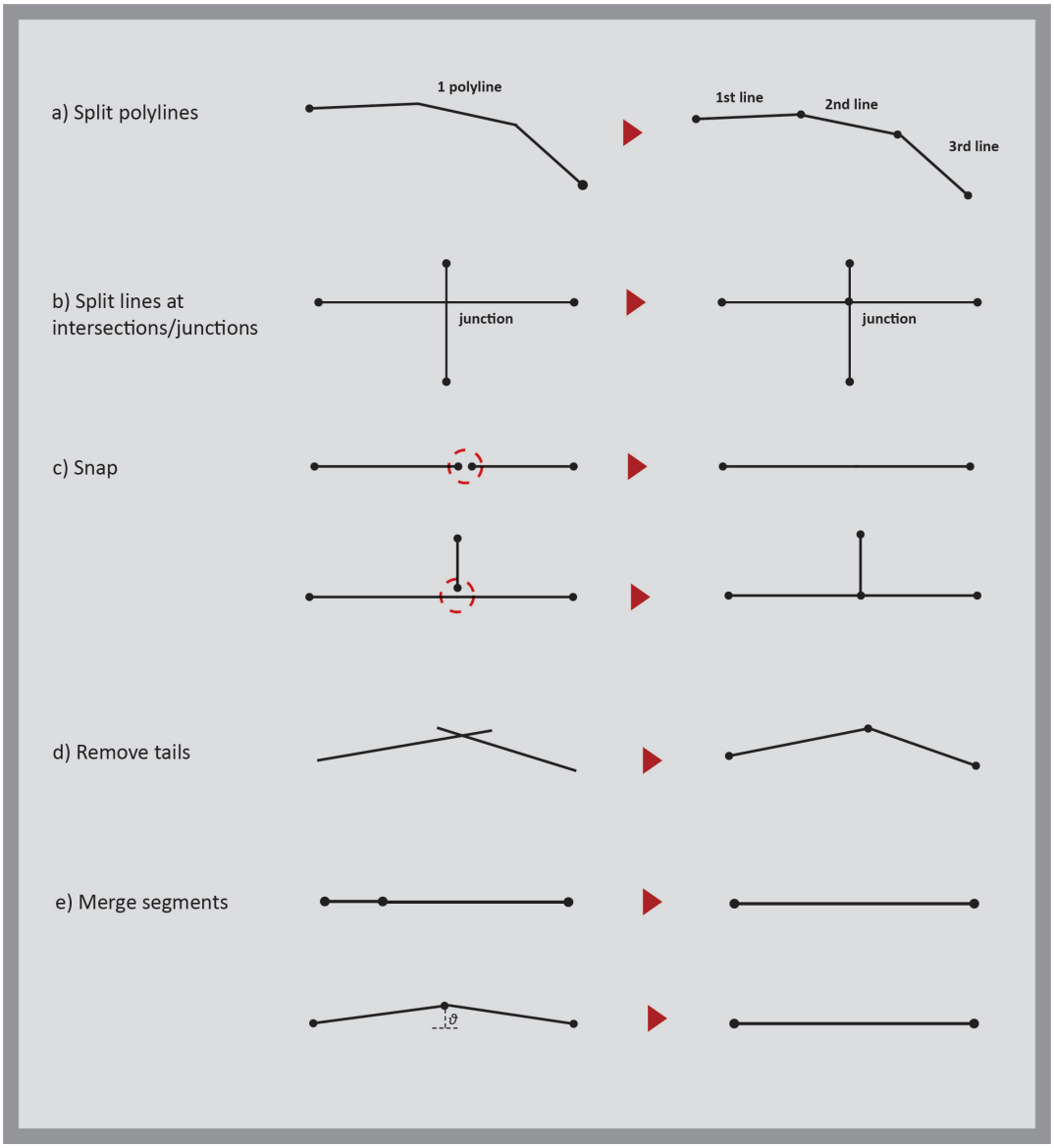

Fig 6. Examples of Editing operations in 'Create Segment map' tool

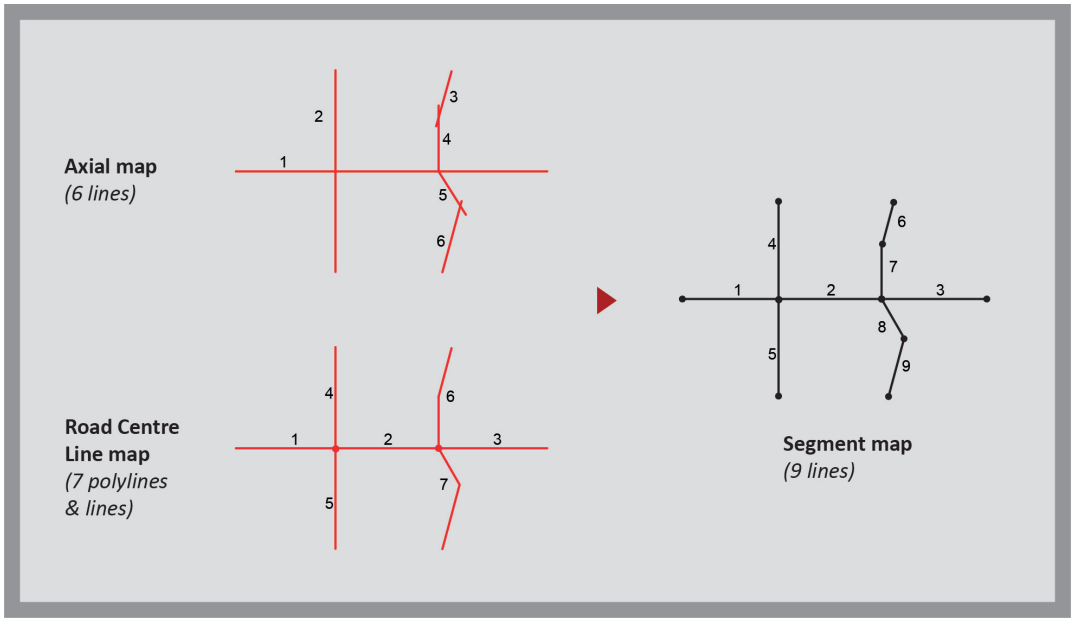

Fig 7. Example of Segment map created from Axial map and Road Centre Line map using 'Create Segment map' tool

5.3 Create Junctions

This tool intersects lines from one or two tables and outputs a new table with all intersection points/junctions. If a table with the Unlink points is also uploaded, then the points that overlap with unlink points will be excluded from the output table.

When one input table is used, a junction is defined as a point where three or more lines meet (in case of a Segment map), or where two lines intersect (in case of an Axial map). Note that the intersection point should be far enough from each line's end-points. The threshold used for this is 2% of the line's length.
When two input tables are used, a junction is defined at the point where lines from the different tables touch or intersect. Touching or intersecting lines from the same table will in this case not generate a junction.

6. Analyses

6.1 Reach

Reach is defined as the part of the network that is reachable from each line of the network within a given radius. Depending on the settings, the analysis can output the number of reachable lines, the total length of all reachable lines[6], or the reachable area. In the last case, the area is calculated as the area of the convex hull, defined by the end-points of all reachable lines. Figure 9 and 10

Supported Radius are straight line distance, walking distance, axial/segment steps, angular and axialmeter (see section 3 for more info).

The user can also use a different table of Origins, from which the reachable part of the network is calculated. In case the Origins are the lines of the network, these are also included in the calculation. In case of another table of Origins, the distance to the closest line is included in the calculation (see section 2, point 3). A line is classified as being reachable if its mid-point can be reached within the given radius.
In the first step of the analysis a graph is built from the in-data (see section 2). Then, a breadth-first search is performed from every node in the graph to find all reachable nodes within the defined radius.

Parallel calculations of Reach are available, meaning that the user can calculate the Number of reachable lines, the Total length of all reachable lines and the Reachable area, using the same Radius, at the same time.

6.2 Network Integration

Network Integration calculates how many turns have to be made from each line to reach all other lines in the network, using shortest paths. This measure thus shows, for each and every line, how many steps (measured topologically) this line is away from all other lines in the network. The measure was first introduced in 'Social logic of Space' (Hillier and Hanson, 1984).

Distance is measured in axial/segment steps depending on which network (axial map or segment map) is used (see section 3). It is recommended to use the Axial map for calculating Network Integration. When a segment map is used, it is recommended to use Angular Integration (see section 6.3). The supported radii are straight line distance, walking distance, axial/segment steps, angular and axialmeter (see section 3 for more details).

In the first step of the analysis a graph is built from the in-data (see section 2). Then for each line in the graph:

1. A breadth-first search limited by the given radius is performed. During the search the number of reached lines N (including origin line) is calculated, as well as the total sum of shortest distances to all the lines (TD).

5. The algorithm runs at a time complexity of O(n log n) where n is number of lines in the source table.

6. This measure is related to Metric Reach as introduced by Peponis et al., 2008.

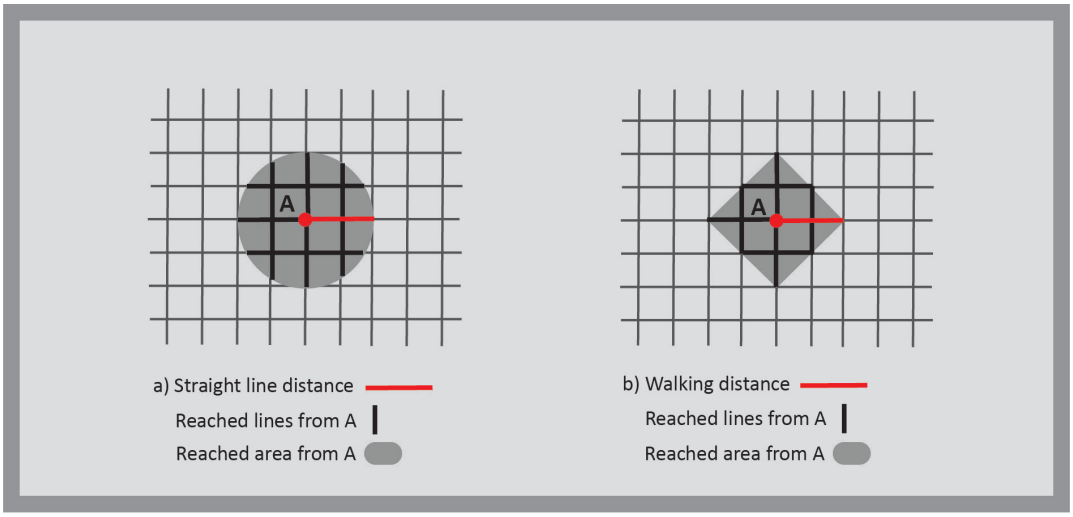

Fig 9. Example of 'Reach' using a) straight line distance and b) walking distance

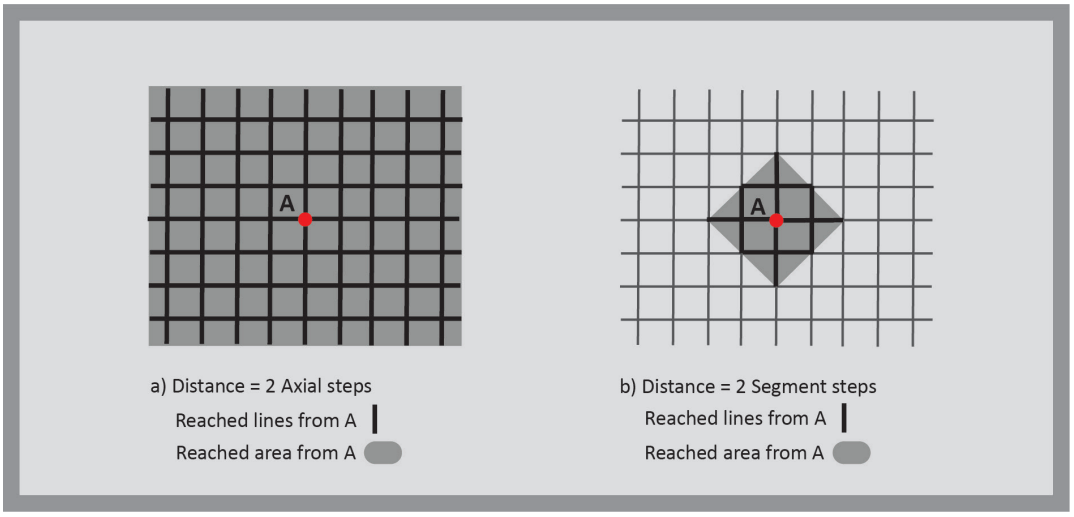

Fig 10. Example of 'Reach' using a) axial steps and c) segment steps

2. Mean depth is calculated as:
   $MD = TD / (N - 1)$ (1)
   where
   TD = total depth or sum of shortest distances to reached lines
   N = node count or number of reached nodes (including origin node)
3. Relative asymmetry is calculated as:
   $RA = 2(MD - 1) / (N - 2)$ (2)
4. "D-Value" is calculated as:
   $D = 2((\log2((N + 2) / 3) - 1) * N + 1) / ((N - 1) * (N- 2))$ (3)
5. Real Relative Asymmetry is calculated as:
   $RRA = RA / D$ (4)
6. Finally, the integration value for the line is calculated as:
   $I = 1 / RRA$ (5)

The user can choose to output also Mean depth (MD), Node count (N) and Total depth (TD) in the result table.

The user can also choose to 'Store average scores at junctions', including Network Integration, Mean depth (MD), Node count (N) and Total depth (TD).

6.3 Angular Integration

Angular Integration calculates the Angular distance between one line and every other line in the network, using shortest paths. Angular distance is defined as the accumulated angular turns needed to get from line A to line B in the network, through the shortest path (see section 3 for more details). Angular distance is measured in degrees and then divided by 90 (Iida and Hillier, 2005). This analysis is only applicable to Segment maps.

When calculating the distance along the path of three segments A, B and C, the distance from A to C is calculated as the angular deviation between A and B plus the angular deviation between B and C.

When calculating the angular deviation between two pairs of segments the angle is rounded to the closest multiple of specified angle precision (specified in the user interface). Another option is available called angle threshold. All angles lower than this threshold value will be truncated to zero. In other words, if the angle threshold is set at 10 degrees, all angular turns below 10 degrees will not be included in the shortest path calculation.

The supported radius measuring modes are straight line distance, walking distance, axial/segment steps and angular (see section 4 for more details).

The first step of the analysis builds a graph from the in-data (see step 1 & 2 in section 2). Then for each line in the graph:

1. A breadth-first search limited by the given radius is performed. During the search the number of reached nodes N (including origin node) is calculated, as well as the angular depth D (measured in degrees turned divided by 90) of all reached lines.
2. Depending on selected normalization mode the line gets the integration value for:

   Normalization (Turner 2007)[7]:

$$AI(x) = \frac{N-1}{1+\sum_{i \neq x} D(x,i)} \qquad (6)$$

   Syntax normalization (NAIN):

$$AI_A(x) = \frac{N^{1.2}}{1+\sum_{i \neq x} D(x,i)} \qquad (7)$$

   Normalization (Hillier):

$$AI_H(x) = \frac{N^2}{1+\sum_{i \neq x} D(x,i)} \qquad (8)$$

   where
   N = node count or number of reached nodes (including origin node)
   D(x, i) = angular depth of i in relation to x

7. This is same formula as in Turner (2007) page 544, but with an extra +1 in divisor to avoid division by zero.

The analysis also offers an option for weighing by length. When this option is chosen the integration values are instead calculated as follows:

Normalization (Turner 2007) :

$$AI(x) = \frac{\sum_{i \neq x} l(i)}{1 + \sum_{i \neq x} D(x,i) l(i)} \qquad (9)$$

Syntax normalization (NAIN):

$$AI_A \quad (x) = \frac{\left(\sum_{i \neq x} l(i)\right)^{1.2}}{1 + \sum_{i \neq x} D(x,i) l(i)} \qquad (10)$$

Normalization (Hillier):

$$AI_H \quad (x) = \frac{\left(\sum_{i \neq x} l(i)\right)^{2}}{1 + \sum_{i \neq x} D(x,i) l(i)} \qquad (11)$$

The user can choose to output also Mean depth (MD), Node count (N) and Total depth (TD) in the result table.

Note that Angular Integration does not require a table with the Unlink points. However, the unlinked Segments should be properly represented in the Segment map (see Section 2.2).

Parallel calculations of Angular Integration are also available, meaning that the user can choose different Normalization formulas, at the same time.

6.4 Network Betweenness

Network Betweenness calculates how often a line falls on the shortest path between all pairs of lines in a network, or how many shortest paths pass through it. In other words, lines (axial lines or segments) which control and mediate movement and connections between many other lines in the system have a high betweenness value.

Betweenness B of a node x is calculated as follows:

$$B(x) = \sum_{s \neq x \neq t} \frac{\sigma_{st}(x)}{\sigma_{st}} \qquad (12)$$

where s and t are nodes in the network different from x, σst denotes the number of shortest paths from s to t, and σst(x) is the number of shortest paths from s to t that pass through x. A shortest path between two nodes s and t is only counted once (i.e. undirected graph), that is, a node is not scored twice for being on the shortest path from both s to t and t to s. End nodes of a shortest path are not given a score (hence all tail segments in a network will get a score of zero). Figure 11

Distance can be measured in different ways. Available options are walking distance, axial/segment steps, angular and axialmeter. The supported radii are straight line distance, walking distance, axial/segment steps, angular and axialmeter (see section 3 and 4 for more details).

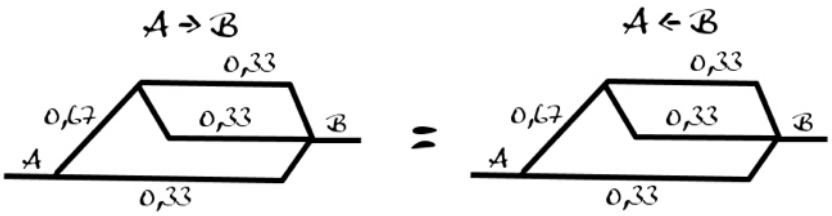

Fig 11. Network Betweenness

The analysis has an option for weighing the scores by line length or other segment data. When By segment length is chosen every segment on the shortest path between two pairs of segments A and B will be given the score Length of A * Length of B, instead of 1 point as in the non-weighed mode. When By segment data is chosen, instead of segment length the user can select any other data column X from the segment table. Every segment on the shortest path between two pairs of segments A and B will then be given the score X of A * X of B.

The analysis offers several normalization modes. The formulas used in the different modes are:

No normalization:
Values will be calculated according to formula (12).

Normalization (Turner 2007):
Normalization for a node x is done by dividing by number of node pairs excluding x:

$$B_T(x) = \frac{B(x)}{(N-1)(N-2)/2} \qquad (13)$$

where N = Node count or number of reached nodes (including origin node).

Standard normalization (0-1):
Values will be rescaled to 0-1 range:

$$B_S(x) = \frac{B(x) - \min(B)}{\max(B) - \min(B)} \qquad (14)$$

In the exception where min(B) = max(B) nodes will get score 1.

The user can choose to output also Mean depth (MD), Node count (N) and Total depth (TD) in the result table.

Parallel calculations of Network Betweenness are available, meaning that the user can choose different Normalization and Weighting formulas, at the same time.

6.5 Angular Betweenness

Angular Betweenness is Network Betweenness, with the difference that the distance mode used to define the shortest paths is Angular, meaning the accumulated angular turns needed to get from point A to point B (see Section 3).

For Angular Betweenness one more Normalisation option is added to the previous ones (12,13,14):

Syntax normalization (NACH)[8]:
Values are calculated according to formula:

$$B_C \quad (x) = \frac{\log(B(x)+1)}{\log(2+\sum_{i \neq x} D(x,i))} \tag{15}$$

where D(x,i) = depth of i in relation to x.

6.5 Angular Choice

Angular Choice is very similar to Angular Betweenness. The two analyses share the same base implementation in PST, but with some differences. The scoring is done the same way as in Angular Betweenness, but every path is counted twice, one for every direction (i.e. directed graph).

The two analyses also differ when it comes to distributing scores when several equally long shortest distances are found between two segments. In Angular Betweenness, for a given pair of segments A and B the score 1/σAB (where σAB is number of shortest paths between A and B) is added to each segment along each of the shortest paths. In Angular Choice on the other hand, a traversal of the paths from A to B is done, initially with a score of 1, and for every fork in the path the score is divided equally among the forking paths (Hillier et al, 2012; Turner, 2007). When forking paths later join up, their respective scores are summed together as the traversal continues towards B. During the traversal every segment that is passed is given the current score of that current forking path, makes the scoring direction dependant. Figure 12

The formulas in the different normalization modes available are as follows:

No normalization:
The scores from the algorithm described above will be outputted without modification.

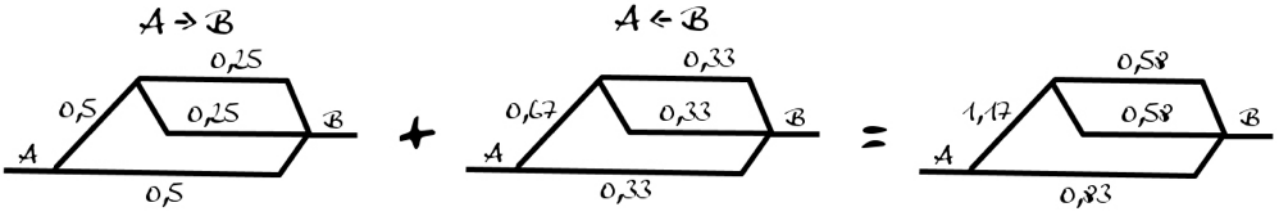

Fig 12. Angular Choice

Normalization (Turner 2007):
Normalization for a node x is done by dividing by number of node pairs excluding x:

$$CH_T\,(x) = \frac{CH(x)}{(\mathrm{N}-1)(\mathrm{N}-2)} \tag{16}$$

where N = Node count or number of reached nodes (including x).

Standard normalization (0-1):
Values will be rescaled to 0-1 range:

$$CH_S(x) = \frac{CH(x)-\min(CH)}{\max(CH)-\min(CH)} \tag{17}$$

In the exception where min(B) = max(B) nodes will get score 1.

8. This is the same formula as in Hillier et al (2012, p 191) developed for Angular choice, but with +2 instead of +3 in divisor since PST doesn't have to account for possible -1 values like Depthmap does.

Syntax normalization (NACH)[9]*:*
Values are calculated according to formula:

$$NAChoice(x) = \frac{\log(CH(x)+1)}{\log(2+\sum_{i \neq x} D(x,i))} \qquad (18)$$

where D(x,i) = depth of i in relation to x.

The user can choose to output also Mean depth (MD), Node count (N) and Total depth (TD) in the result table.

This analysis is only applicable to Segment maps.

Note that Angular Choice does not require a table with the Unlink points. However, the unlinked Segments should be properly represented in the Segment map (see Section 2.2)

Parallel calculations of Network Betweenness are available, meaning that the user can choose different Normalization and Weighting formulas, at the same time.

6.6 Attraction Distance

Attraction Distance calculates the minimum distances from each object of a set of origins, through a network, to the closest object of a set of destinations. The available Distance modes and Radii are: straight line distance (m), walking distance (m), axial/segment lines (steps), angle (degrees) and axialmeter (steps*m) (see section 3 and 4 for more details).

Attraction distance AD for a given origin o is calculated as:

$$AD(o) = \min_{a \in A}(D(o,a)) \qquad (19)$$

where
A = the set of reachable attractions within given radius
D(o,a) = shortest distance from origin o to attraction a.

There are three different types of origins that can be used: points/polygons, axial/segment lines and junctions. For the first alternative, points/polygons, the user specifies a separate table of origin objects. These objects will be linked to their closest line in the network, including the distance between object and line. The calculated minimum distances from these objects to the destinations will be written in a new column in the origin table. The second option, axial/segment lines, will calculate minimum distance from the lines of the network itself and write the calculated distances to a new column in the network table (see section 2, point 3, for more details). Finally, the third option, junctions, will generate a new point table with a point at every junction, and will calculate distances from these points, as well as write the calculated distances to a column of this new table.

Minimum distance can be measured for any destinations element, but destinations can also be weighed by data using one of the columns in the destination table. The user can choose more than one weight (i.e. column) per destination which will result in an equal amount of output columns.

The first step of the analysis builds a graph from the in-data (see step 1-3 in section 2). Then for each origin object in the analysis:

1. Depending on origin type (see above) the initial link distance to the network is added as an initial distance.
2. A breadth-first search limited by the given radius is then performed in the network graph. During the

9. This is the same formula as in Hillier et al (2012, page 191), but with +2 instead of +3 in divisor since PST doesn't have to account for possible-1 values like Depthmap does.

search the minimum distance travelled to each line of the network is calculated.
3. For each reached line in the network, its list of connected attraction points is traversed, and minimum distance to attractions is updated if a shorter distance than previously encountered is found.
4. If no attraction was found within the given radius, a score of-1 will be assigned to the origin object.

Parallel calculations of Attraction Reach are available for different Weighting of Destinations at the same time. Parallel calculations are also possible for different Distance modes.

6.7 Attraction Reach

Attraction Reach calculates, for each object in a set of origin objects, the sum of all attractions from a set of attraction objects that can be reached via the network within a by the user defined radius (see section 3 for distance modes and section 4 for radii available).

The implementation of the analysis is similar to the Attraction Distance analysis. In-data is handled in the same way, and the same algorithm is used (see section 6.6 for more details), but for every attraction that is reached its score is added to the score of the origin object.

Attraction reach AR for a given origin o is calculated as:

$$AR(o) = \sum_{a \in A}(f(a)w(D(o,a))) \qquad (20)$$

where
A = the set of reachable attractions within given radius
f(a) = attraction value associated with attraction a, or 1 if no attraction value is used
D(o,a) = shortest distance from origin o to attraction a
w(x) = attenuation function

Besides the option that all destinations are counted equally, the destinations can also be weighed by data using one of the columns in the destination table. The user can choose more than one weight (i.e. column) per destination which will result in an equal amount of output columns.

It is further possible to specify an exponential function (i.e. Attenuation functions) of the path length by choosing one of the available functions or defining one manually. This will be applied to every attraction score before adding the result to the origin object.

Parallel calculations of Attraction Reach are available for different Weighting of Destinations at the same time.

6.8 Attraction Betweenness

Attraction Betweenness re-uses the functionality of the Network Betweenness analysis (see section 6.4), but adds functionality for assigning weights to the lines from a table with attractions.

Each attraction point (or if polygons or lines are used, their centroid resp. midpoint) is assigned to the closest line (see section 2, point 3, for more details) in the network, and selected data is transferred from the point to the line. The collected scores on each line are then used as weight in the same way as length is used as weight in Network Betweenness (length weight). The main difference is that another table can be used to weigh the lines.

The formulas in the different normalization modes available are No normalization and Standard normalization (see 5.4 for more details).

Parallel calculations of Attraction Betweenness are available for different Weighting of Destinations at the same time. Parallel calculations are also possible for different Distance modes and Normalizations.

7. Segment group analysis

7. 1 Segment grouping

This experimental function is inspired by different explorations regarding angular distance (Stavroulaki et al 2017, Peponis eta al 2008, Figueiredo 2015, Jang and Claramunt 2004) and the angular deviation which is considered to be perceptually significant for the moving person.

The tool uses a Line-segment map and modifies the graph on the fly, based on two different options:

1. The user first chooses an 'Angular threshold' under which all changes of direction are ignored and are not considered as graph edges. For example, if a 30 degrees threshold is used, then an angular distance (deviation) of less than 30 degrees between line-segments will be disregarded, and the respective segments will be grouped on the fly and analysed as collinear.

2. Then, the user decides whether or not to 'Split at junctions', meaning whether or not include street junctions as graph edges, even when they lead to a direction change that is zero or below the specified threshold; meaning that street junctions are considered significant, irrespectively of the actual change of direction.

The tool offers the possibility to use the same line-segment map and create different graphs on the fly, based on these different settings.

Both graph-edges and graph-nodes change in the background of the analysis. Line segments are grouped on the fly, to form segment-groups depending on the parameters chosen; these groups are the graph nodes. Figure 13

The user has the option to visualise the different segment-groups formed by the chosen settings in a new QGIS layer, where they are given different colours, by choosing:
'Generating minimal disjunct colours' and 'Apply generated colours to map'.

This background operation doesn't change the integrity of the original data and the original Line-segment map; the line-segments remain unmodified. Imagine, for example, three segments which, according to the parameters of the analysis, are considered continuous and are 'compressed' to form one graph node; in the output table, they remain separate features and get the same Integration value (see Section 7.2).

For more information on the background and motivation of the tool see Stavroulaki et al 2017.

7.2 Segment group integration

The experimental tool, so far, calculates Network Integration at different radii (steps or walking distance), see Section 4. The difference from the typical Network Integration (Section 6.2) is the Node definition. Here, each node is a segment group, instead of one segment or one axial line. The formula is:

$$I_x = \frac{d(N-2)}{2\left(\frac{\sum_{i \neq x} D_{(x,i)}}{N-1} - 1\right)} \qquad (21)$$

Where,
"d-value" is calculated as:
d = 2((log2((N + 2) / 3) – 1) * N + 1) / ((N – 1) * (N- 2))
D(x, i) = topological steps of the shortest path between i and x
N = node count or number of the formed segment-groups.

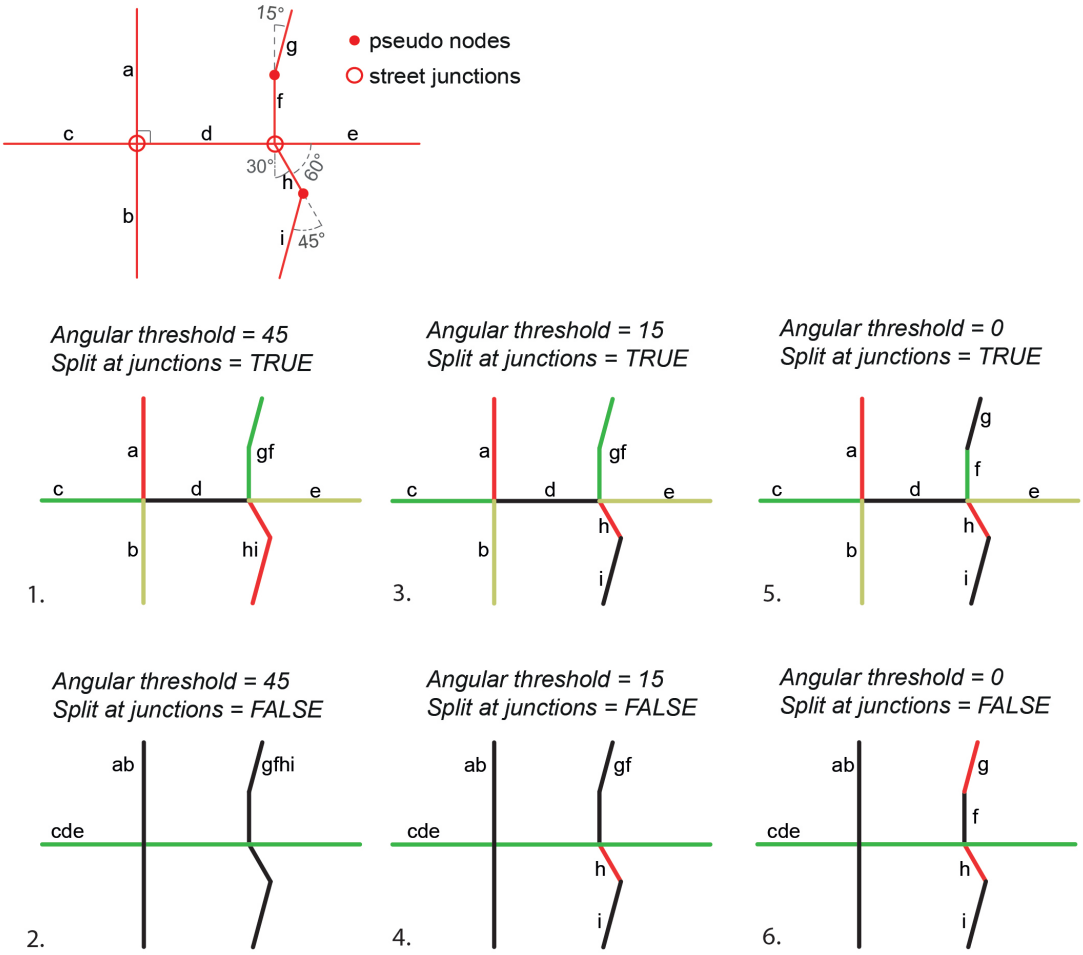

Fig 13. Variations of a Line-segment map; segment groups created on the fly using different parameters. Segment groups are separated by colour.

Also, Node Count, Total Depth and Mean Depth are outputted in the results. One can run different analyses using a different set of parameters and the results of each calculation are written in the same table, making them easily comparable.

7 References

Brandes, U., 2001, "A faster Algorithm for Betweenness Centrality", in Journal of Mathematical Sociology, 5(2), p. 163-177.

Figueiredo L. (2015), "A unified graph model for line and segment maps". In: Karimi,K., Vaughan,L., Sailer,K., Palaiologou,G. and Bolton,T. (eds.), Proceedings of the 10th International Space Syntax Symposium, London: University College London, p.146.1-146.11.

Hillier, B., Yang, T., Turner, A., 2012, "Normalising least angle choice in Depthmap and how it opens up new perspectives on the global and local analysis of city spoace", in The Journal of Space Syntax, vol. 3, issue 2, p. 155-193.

Hillier, B., Iida, S., 2005, "Network effects and psychological effects: A theory of urban movement", in: van Ness, A. (ed.) Proceedings of the Fifth International Space Syntax Symposium, Delft: University of Technology, Vol. 1, p.553-564.

Hillier, B., Hanson, J., 1984, The social logic of space. Cambridge: Cambridge University Press.

Peponis, J., Bafna, S., Zhang, Z., 2008, “The connectivity of streets: reach and directional distance,”, in Environment and Planning B: Planning and Design, Vol.35, no 5, p. 881-901.

Jiang, B. and Claramunt, C. (2004), “Topological analysis of urban street networks”. In Environment and Planning B, Vol.31, p.151-162.

Peponis, J., Bafna, S. and Zhang, Z. (2008), “The connectivity of streets: reach and directional distance,”. In Environment and Planning B: Planning and Design, Vol.35, no 5, p. 881-901.

Ståhle, A., Marcus, L., Karlström, A., 2005, ”Place Syntax: Geographic accessibility with axial lines in GIS”, in Proceedings of the 5th Space Syntax Symposium, Delft.

Stavroulaki, G., Marcus, L., Berghauser Pont, M., Nilsson, L., 2017, ‘Representations of street networks in space syntax – towards flexible maps and multiple graphs, in: (eds. Heitor T, Serra M, Silva J P, Bacharel M, da Silva L C) Proceedings of 11th International Space Syntax Symposium, University of Lisbon, , Instituto Superior Technico, Departamento de Engenharia Civil, Arquitetura e Georrecursos, Portugal, ISBN: 978-972-98994-4-7

Turner, A., 2007, “From axial to road-centre lines: a new representation for space syntax and a new model of route choice for transport network analysis”, in Environment and Planning B: Planning and Design, Vol 34, p 539-555.

Analysis	abbreviation	Distance mode (B)	abbreviation	Radius (R, NI, AI, B, AB, AC)	abbreviation	unit (not expressed in name)	abbreviation	Weight mode (AI, B, AB, AC)	abbreviation	Normalization AI, B, AB, AC	abbreviation
Reach	R	walking distance	w	straight line distance	l	meter (i)	-	no weight	-	no normalization	-
Netwok Integration	I	axial/segment steps	s	walking distance	w	meter (i)	-	weight by length	wl	NAIN	A
Angular Integration	AI	angular	a	axial/segment steps	s	n	-	by segment data	user defined (ii)	NACH	C
Network Betweenness	B	axialmeter	am	angular	a	degrees/90	-			Turner	T
Angular Choice	AC			axialmeter	am		-			Hillier	H
Angular Betweeness	AB									standard (0-1)	S

(i) change 1000 in 1k etc.

(ii) maximum amount of characters 2

example

Bww1kwIC (analysis: Network Betweenness, distance mode: walking distance, radius: walking distance 1000 m, weighted by length, normalization: NACH)

Extra output

	abbreviation		abbreviation	unit	abbreviation
Reach	R	Number of reached lines	N	amount	-
		Total length of reached lines	L	meter	-
		Area of convex hull	a	square meter	m
				square kilometer	k
				hectare	ha

example

R_w_w100_acv_m (Reach, distance mode: walking distance, radius: walking distance 100 meter, area of convex, in square meter)

	abbreviation		abbreviation
Netwok Integration	I	node count	N
Angular Integration	AI	total depth	T
Network Betweenness	B	mean depth	M
Angular Betweenness	AB		
Angular Choice	AC		

example

Analysis	abbreviation	Distance mode (D, AT)	abbreviation	Radius (D, AR, AT)	abbreviation	unit (not expressed in name)	abbreviation	Radius distance decay (AR)	abbreviation	Destination weight (D, AR, AT)	abbreviation	Normalization (AT)	abbreviation
Attraction Distance	D	straight line distance	l	straight line distance	l	meter (i)	-	no decay	-	equally	table name (iii)	no normalization	-
Attraction Reach	AR	walking distance	w	walking distance	w	meter (i)	-	by function (ii)	f	by data	column name iv)	standard (0-1)	S
Attraction Betweenness	AT	axial/segment steps	s	axial/segment steps	s	n	-						
		angular	a	angular	a	degrees/90	-						
		axialmeter	am	axialmeter	am		-						

(i) change 1000 in 1k etc.

(ii) if more functions are chosen, they will overwrite one another (if all the other choices are similar)

(iv) first two characters of table name OR user defined

(v) first two characters of the column name OR user defined

example