

148

The space syntax toolkit: Integrating depthmapX and exploratory spatial analysis workflows in QGIS

Jorge Gil

Space Syntax Laboratory, The Bartlett School of Architecture, UCL
jorge.gil@ucl.ac.uk

Tasos Varoudis

Space Syntax Laboratory, The Bartlett School of Architecture, UCL
t.varoudis@ucl.ac.uk

Kayvan Karimi

Space Syntax Laboratory, The Bartlett School of Architecture, UCL
k.karimi@ucl.ac.uk

Alan Penn

Space Syntax Laboratory, The Bartlett School of Architecture, UCL
a.penn@ucl.ac.uk

Abstract

The “Space Syntax Toolkit” is a QGIS plug-in for spatial network and statistical analysis. It provides a front-end for the depthmapX software within QGIS, offering user-friendly space syntax analysis workflows in a GIS environment. It is primarily aimed at supporting the space syntax methodology, and enhancing it with GIS data, analysis and visualisation features. Nevertheless, its functionality is of general benefit to QGIS users by introducing tools for exploratory spatial data analysis. In this article we present the “Space Syntax Toolkit”, describing its principles and implementation, the features of its two initial modules – “Graph Analysis” and “Attribute Explorer” – and give indication of future developments.

Keywords

GIS, QGIS, space syntax, network analysis, urban design education.

1. Introduction

Urban design practice, as a discipline closely related to architecture, has traditionally relied on the use of Computer Aided Design (CAD) platforms for computational tasks. However, urban design processes in evidence-based design, comprehensive planning and sustainable urban development have been evolving to require increasingly complex decision making in many fields, e.g. social, economic, environmental, transportation. These advanced design processes rely on a range of support tools, including spatial and statistical analysis, of which space syntax is one of the frequently used theories and methods. Over the years, a range of software has been developed to perform space syntax analysis¹, as standalone tools (e.g. Axman and WebMap by Nick Dalton, Depthmap (Turner, 2007), Mindwalk by Lucas Figueiredo), as extensions to CAD (e.g. Spatialist from Georgia Tech, sDNA (Chiaradia et al., 2012), UNA Toolbox (Sevtsuk and Mekonnen, 2012)) or to Geographic Information Systems (GIS) (e.g. Axwoman by Bin Jiang, UNA Toolbox, sDNA, Confeego and iVALUL from Space Syntax Ltd). GIS seems to offer a natural platform for such methods (Jiang and Claramunt, 2002; Gil et al., 2007; Chiaradia et al., 2008), with its ability to handle geographic and geometric data associated with attribute information, to easily perform spatial, mathematical and statistical calculations, and to visualise the results. In order to improve space syntax's contribution to urban design research, education and practice, it is important to develop support tools that integrate its complete workflows in a single platform, are intuitive for beginners, flexible for researchers, operational for practitioners, and accessible to all users. With this aim we have created an open source software project to develop a GIS toolkit that integrates the depthmapX multi-platform spatial network analysis software (Varoudis, 2012) with the QGIS software platform (QGIS Development Team, 2014). The "Space Syntax Toolkit" (SST) (Gil, 2014) provides a frontend for depthmapX within QGIS. It enhances space syntax workflows with GIS data, analysis and visualisation features, integrates space syntax network analysis in QGIS, and expands QGIS with new exploratory data analysis functionality. In this article we present the SST, describe its implementation requirements and principles, the features of its two initial modules – "Graph Analysis" and "Attribute Explorer" – and indicate future developments.

2. Requirements for and development of a toolkit

Integrated workflows for QGIS

The SST's aim is to provide an entry into the use of GIS for advanced spatial network analysis methods, primarily in post-graduate education and research. The key challenges are related with the engagement of users that have a visual and architectural background in the use of GIS, avoiding as much as possible the frustrations inherent to complex analysis workflows, with complicated and lengthy intermediate steps that require specific input/output formats, and instead offering swift and intuitive data exploration. For space syntax beginners, the SST should offer an interface in a single platform to support complete analysis workflows, with a clear user interface using space syntax specific terminology and offering simple options. For more experienced researchers and practitioners, the SST should allow advanced analytic options and support greater exploratory depth of the results. Along with these user experience requirements, the tool should be flexible in dealing with a variety of data sets, supporting large network models and streamlining the workflows for greater speed, eliminating as much as possible error prone and cumbersome intermediate steps. Figure 1 illustrates (a) the current space syntax analysis workflow in depthmapX (Varoudis, 2012) for CAD and GIS users, versus (b) the desired workflow integrated in a single and expandable GIS platform. The most obvious advantage is the elimination of several data export steps with an accumulated benefit, as each data export entails the selection of file formats, versions, projections and other settings for both the axial map and the unlinks, which set wrongly will cause problems in the analysis or the interpretation of results. Furthermore, these steps have to be repeated every time the user needs to make changes to the axial model, for example to correct a mistake in the axial model. In the desired workflow, the user should be able to carry out what could be considered standard space syntax analysis practice without the use of additional software applications.

¹ https://en.wikipedia.org/wiki/Spatial_network_analysis_software

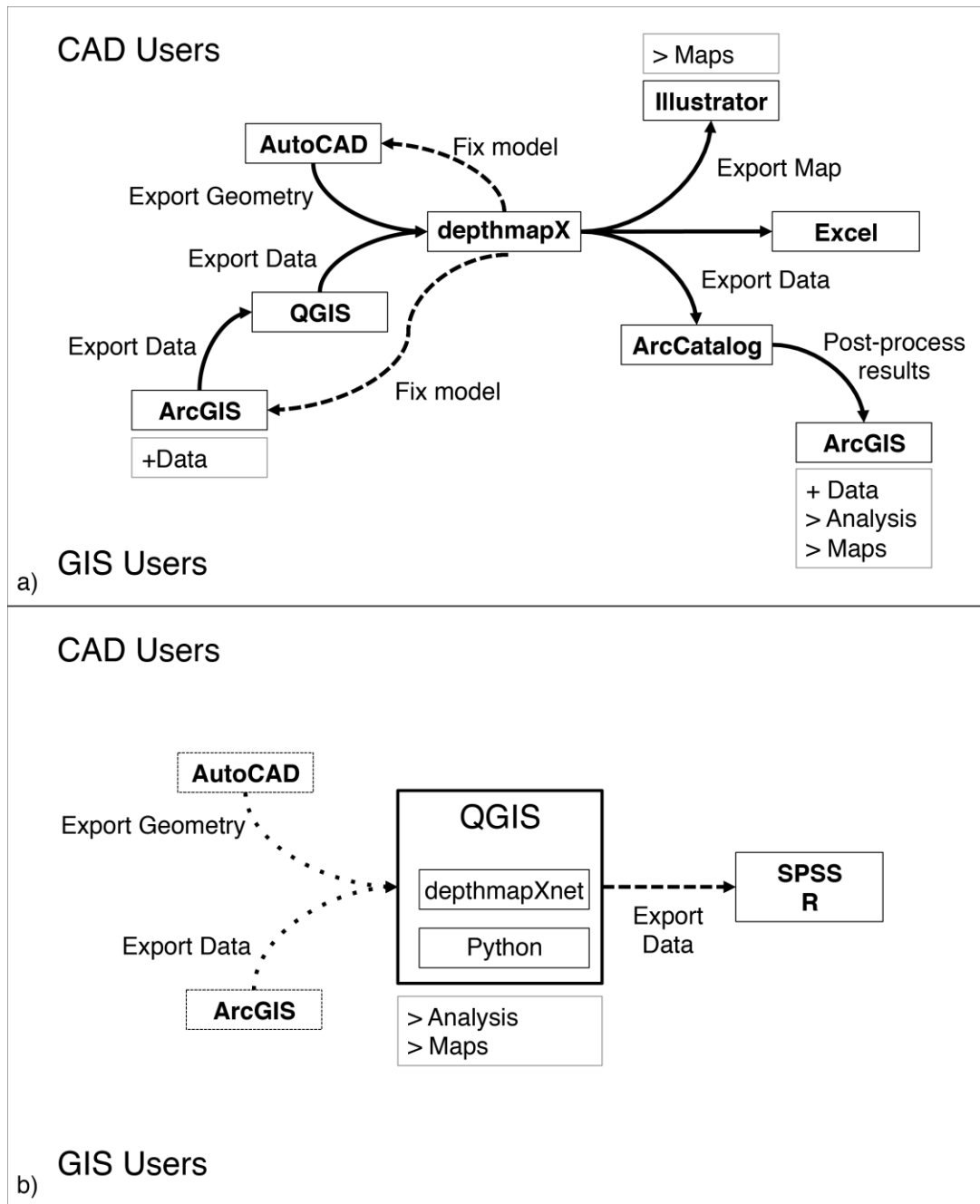


Figure 1: Diagrams of the current (a) and desired (b) workflow of space syntax analysis using the Depthmap or depthmapX softwares.

The typical space syntax research workflow goes through a series of stages that define the user requirements for a comprehensive SST. Table 1 offers a list of such stages and operations adequate for urban research, including data acquisition, geo-processing of spatial data sources, network model preparation and analysis, statistical exploration and analysis, and production of maps and charts for reporting. This list is necessarily incomplete, as much depends on the specific research topic, questions and methodology.

Table 1: Stages and operations of a typical space syntax urban research workflow.

Stage	Operations
Data acquisition	Collect base map data Convert data formats Georeference vector and raster data User behaviour data entry Location survey data entry
Geo-processing	Filter, aggregate and join data based on attributes and spatial relations Find nearest feature Transform geometries Attributes update and calculation
Network model preparation	Draw map Extract map from larger data set Prepare plans Generalise road network Map verification Manage unlinks and links
Network analysis	Analyse network models (space syntax) Post process analysis results (attribute naming, selection, new calculations) Calculate shortest route and catchment areas Street and block morphology
Exploratory spatial data analysis	Interactive visual exploration of results Inquiry of individual feature values Identify the core of the network Spatial analysis (interpolation, hotspots, clustering) Terrain analysis
Statistical analysis	Descriptive statistics Transform columns Interactive charts (histogram, scatter plot) Linear regression Multivariate regression Statistical clustering (PCA, k-means, hierarchical) Export data to statistics packages
Reporting and visualisation	Create base map compositions for print Set output format and resolution Standard visualisation (colour ranges) for screen and print Store named views Produce map series based on attributes

Implementation in QGIS

The SST is implemented in a Python plug-in for the QGIS platform, which offers various advantages as a project and software to the space syntax community. The project is an open source project member of the OSGeo Foundation, meaning that it is compliant with open standards, incorporates many OSGeo analysis packages, and is free to use for any purpose. Furthermore, it has a growing support and user community with an active development cycle and a growing number of plug-ins developed by this community. The software itself is cross platform running on Windows, Linux, Mac OSX and Android, with a user friendly interface emulating in many respects that of ArcGIS, the leading commercial GIS platform, and is documented in English and other languages. It supports most input and output data formats (i.e. raster, vector, databases, web services, tabular data), offers managed plug-in installation, supports automation with macros and scripts, and more advanced customisation with its Python API, that gives access to useful packages developed by the wider Python community.

QGIS can perform, either natively or using existing plug-ins, many of the user requirements outlined in Table 1 that are of a general GIS nature. Nevertheless, to meet those requirements, QGIS still presents several challenges when compared to other platforms currently used by space syntax researchers. On the one hand, it lacks some advanced spatial analysis and statistics available in ArcGIS, and when these are available their use is not intuitive, can be slow and unreliable, and invariably the results are in new tables that require additional (elaborate) operations to merge. On the other hand, it lacks support for exploratory analysis available in depthmapX, such as interaction between statistics and the map canvas, and a fast preview for the long lists of numeric attributes that result from the analysis. In addition, it does not currently have plugins to support space syntax specific workflows, such as axial map production, unlinks management, survey data entry, or network centrality analysis. The aim of the SST is to cover some of these gaps, in particular those related to the integration of depthmapX analysis and the implementation of its user interface features.

The toolkit's architecture

The architecture of the SST consists of a single plug-in for QGIS containing separate modules developed similar to independent plug-ins to allow easier expansion of the toolkit, and a set of shared utility functions to increase consistency between modules and facilitate development of new features. Each module of the SST is accessible individually from the menu and toolbar developed as a side panel (modeless window) allowing the continued use of QGIS's other features. Rather than performing single shot operations from multiple dialog boxes, these modules aim to function alongside and in support of the users' standard workflow, integrating the specific features and tools required. Currently there are two modules, 'Graph Analysis' and 'Attributes Explorer' that are presented in detail in the next section of the paper.

The SST has dependencies on other packages that are not present in QGIS by default. Of special importance is the integration of depthmapX in the 'Graph Analysis' module to perform the standard space syntax network analysis calculations. This integration uses, a newly developed, special version of depthmapX called depthmapX[net] that was developed by Tasos Varoudis in order to deliver all internal analysis capabilities through simple network calls to external applications or scripting environments (e.g Grasshoper plug-in; JAVA and Python based depthmapX client by Tasos Varoudis). The only requirement is that depthmapX[net] is installed and running when performing a space syntax analysis, and the 'Graph Analysis' module works as a remote control, sending the data and analysis parameters and retrieving the results when the calculation is completed. Additional Python packages are required to use the full features of the SST, namely *igraph* (Csardi and Nepusz, 2006) to do alternative network analysis, and *pyqtgraph* (www.pyqtgraph.org) for displaying interactive charts. If these packages are missing from the Python installation, the user will still be able to use the SST, albeit with some limitations, as the alternative features are not completely equivalent.

The SST has been developed to work with any vector layer loaded in a QGIS map canvas, and currently supports output to the native data formats of QGIS: shape files, spatialite personal geodatabases, PostGIS geodatabases and memory layers. Each of these data formats has unique characteristics and the SST takes advantage of format specific features, such as the spatialite and PostGIS SQL query engines to optimise performance in many operations, particularly noticeable with larger data sets. This is a new approach in QGIS plugins, which by default use the shape file format. For this reason users should become familiar with the pros and cons of each data format, and choose the most suitable to their needs.

3. The 'Space Syntax Toolkit' plug-in

The SST plug-in is available for installation and update via the 'Plugins Manager' tool of QGIS, making it easily accessible and maintainable to any QGIS user. Currently, the SST plug-in consists of two modules, "Graph analysis" and "Attributes explorer", whose features are described next.

'Graph Analysis' module features

The 'Graph analysis' module supports the verification and analysis of the space syntax network model, representing the public open space network. It works with the axial line and segment models, including a unlinks layer to indicate bridges and tunnels where there is no level crossing between lines. The layers that make up the model are selected in the respective tabs. The module has a verification tool to check the geometric and topological integrity of each layer, helping to correct problems before running the analysis. The axial and segment analysis is performed in depthmapX[net] with the remote link from QGIS.

Layer verification

The layer verification tools in the 'Verify layer' tab check if the model layers are correct and have the characteristics required for analysis (Figure 2). The axial map verification checks for geometry and topology mistakes that can easily occur when digitising the axial lines over digital cartography, or when merging maps from different authors. The mistakes checked include:

- Small line – lines with a length below the minimum (default: > 1 m)
- Polyline – lines made up of more than two nodes
- Coinciding points – lines with two points coinciding
- Duplicate geometry – two lines with identical geometry
- Short line – line whose endpoint is close to another line without intersecting it (default: > 1 m)
- Orphan – line that is not intersecting any other line
- Island – group of lines that are disconnected from the main map (only verified if the igrph package is present)

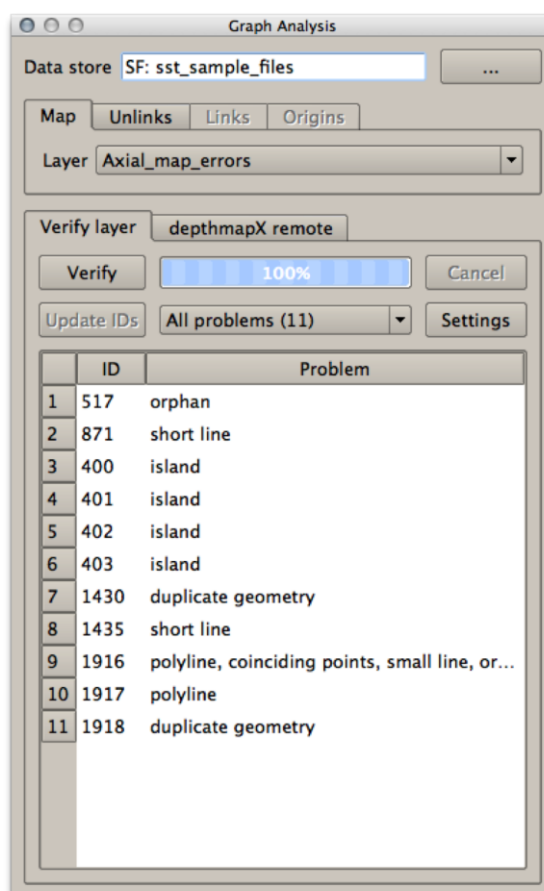


Figure 2: 'Graph Analysis' panel with axial map verification report.

The distance thresholds used in the verification process can be adjusted if required via a separate settings window, to keep the user interface as simple as possible. When the verification process concludes, any problems found are listed in the report window and the total number of occurrences of each problem type is indicated in a menu. The verification tool supports the user to manually correct any problems identified by filtering the list of problems by type, and by automatically selecting and zooming into the line or group of lines corresponding to an entry in the report. The user can then use the QGIS layer editing features to modify or remove the line(s).

When verifying the axial map, if the 'Unlinks' layer is selected, these are taken into account and can reveal additional topological problems due to unlink disconnections. However, this unlinks layer must also be verified and corrected for the axial map verification results to be accurate. An unlink can be any map feature that identifies an intersection between two axial lines to be disconnected, i.e. it can be a point near the intersection, or a line or polygon intersecting the two relevant axial lines. The unlinks layer verification checks for topological and id problems in relation to the axial lines layer:

- Multiple lines – unlinks intersecting more than two axial lines
- Single line – unlinks intersecting only one axial line
- No lines – isolated unlinks not intersecting any lines
- Same line id – unlinks with the same line id attributes
- Unmatched line id – unlinks with the line id attributes not matching the currently intersected axial lines

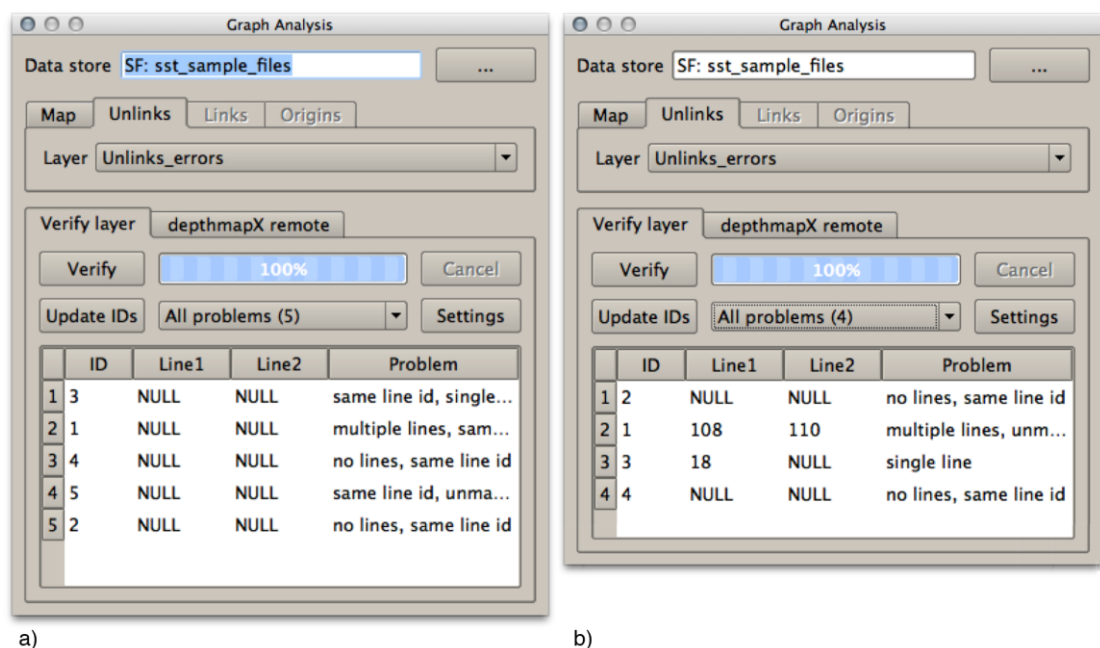


Figure 3: Unlinks verification reports, before (a) and after (b) updating the line ids.

The first time the unlinks verification tool runs, it checks if two attributes exist (i.e. 'line1' and 'line2') that identify the lines to be unlinked. If these attributes are missing, the problems report includes all unlinks (Figure 3a) and the user must run the "Update IDs" tool assigns the id of the lines at the intersection identified by each unlink. This update needs to be triggered by the user to prevent (incorrect) automatic id changes after axial map edits. In such cases, any unlinks with unmatched line id will be signalled and can be confirmed by the user, only then updating the line ids. The unlinks' problem selection, filtering and editing process is identical to that for axial maps.

depthmapX remote

The 'depthmapX remote' tool controls the graph analysis of the selected map and unlinks layers, with the same settings that one would find in depthmapX but in a more compact layout (Figure 4), with advanced settings available from a separate settings window. One can choose between axial (topological) and segment (angular) analysis, set the radius distance, and select an attribute to use for weighted analysis, and select the output layer name.

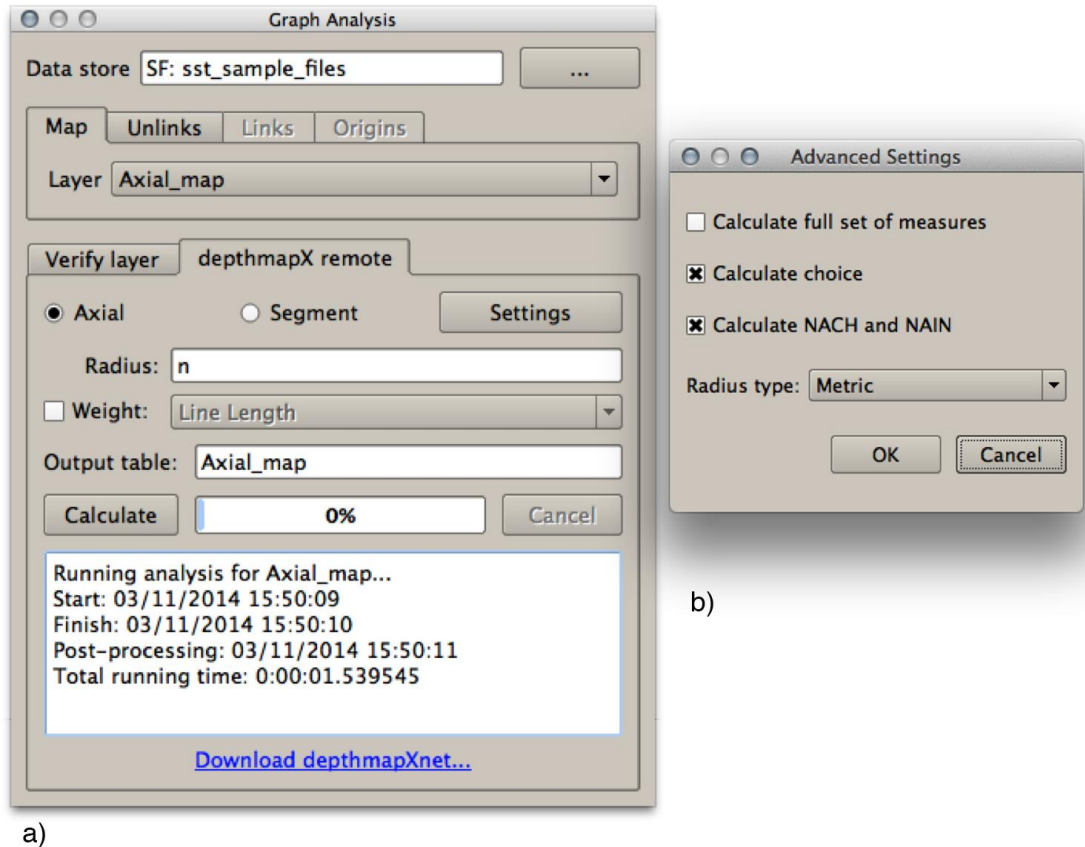


Figure 4: depthmapX remote tab (a) for setting up and running axial and segment analyses remotely, with 'Advanced settings' (b) for changing some of the default analysis parameters.

The layers' data and analysis parameters are sent to depthmapX[net], and both QGIS and depthmapX[net] must be kept running throughout the process, although one can continue working on both while the analysis runs in the background. Once the analysis is completed, the post-processing stage takes place on the QGIS side, where the results are filtered to select a relevant set of measures, new standard measures are calculated, and the attributes are renamed to be compatible with the output format, e.g. shape files only support attribute names with up to 10 characters. The results layer is then added to the current map canvas and the new attributes can be seen in the attributes table. However, the best way to analyse the results is to use the 'Attributes Explorer' module.

'Attributes Explorer' module features

The 'Attributes explorer' module supports the visual and statistical exploration of network analysis results, with the aim to emulate the interactive nature of depthmapX. It offers the quick analysis of individual quantitative attributes of a selected layer, displaying the values on the map using simplified symbology, providing descriptive statistics, and plotting basic interactive charts (Figure 5).

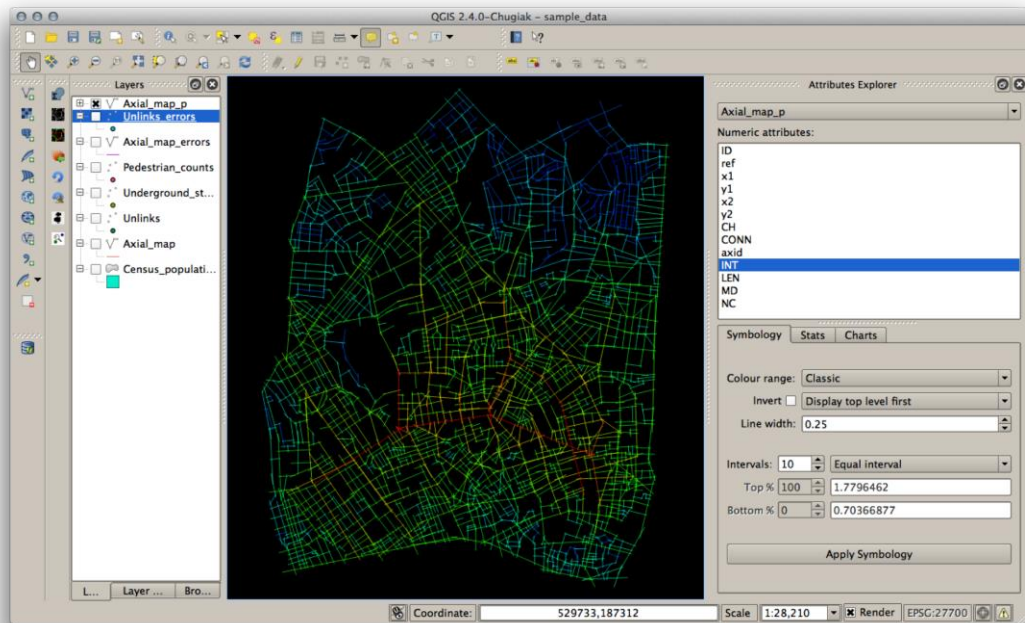


Figure 5: QGIS window with the “Attributes Explorer” panel docked on the right.

Attributes symbology

The Symbology tool (Figure 6) supports the display and production of maps that are standard within the space syntax community and suitable for academic publication. The default style is applied as soon as an attribute is selected in the ‘Numeric attributes’ list, but the user can change the colour and data range settings to create a custom symbology that is more appropriate to the specific attribute. The symbology settings of the different attributes are stored in the QGIS project and become the style associated with those attributes when using this module, allowing the alternate visualisation of different attributes based on their individual styles.

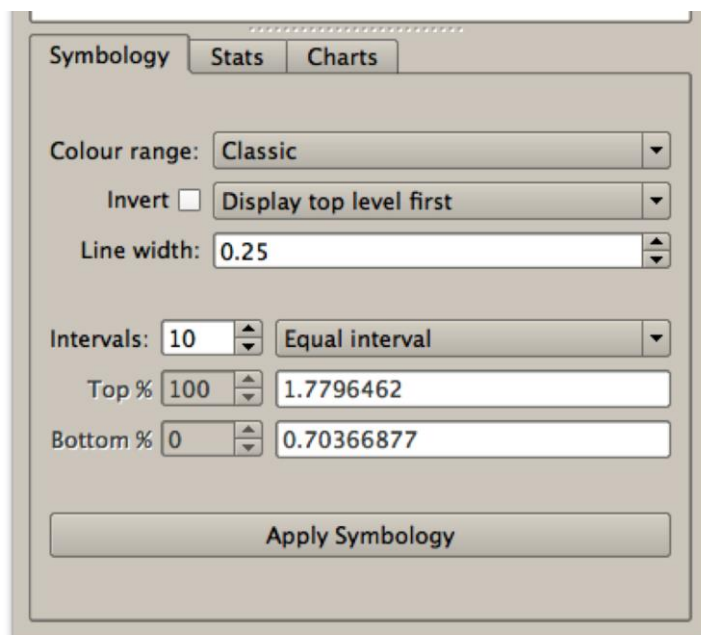


Figure 6: Attribute symbology settings tab.

The symbology settings cover the display style and data range. The display style sets the colour scale (i.e. space syntax classic, red-blue, grey scale and monochrome, shown in Figure 7 a through c), with options to invert the scale and sort the colours, ensuring that the top or bottom values are displayed first on the map, and set the line width. The grey scale and monochrome colour ranges use black or white lines depending on the background's colour (Figure 7d). The data range sets the number and type of colour intervals, e.g. equal range, quantiles, natural breaks or custom. The custom interval is by default the maximum (top) and minimum (bottom) values of the attribute, but the user can change the size of the top and bottom range, using a percentage or the absolute value. The other intervals in between are automatically calculated using equal ranges distribution. This setting is equivalent to the slider controls in the 'Colour Range' settings of depthmapX.

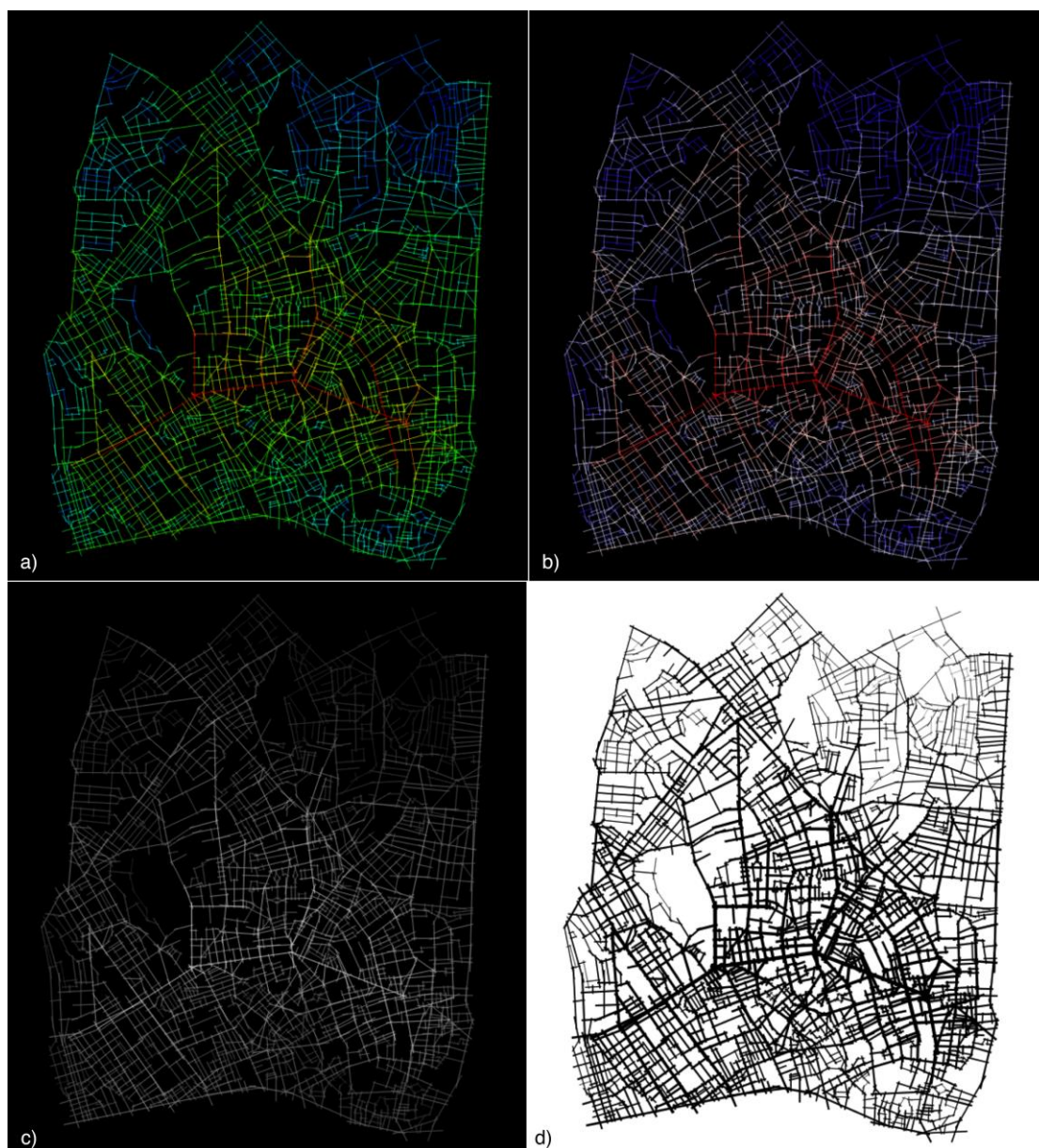


Figure 7: The different colour range settings: classic (a), red-blue (b), greyscale (c) and monochrome (d) with varying line width. The greyscale and monochrome colour ranges automatically use black or white lines depending on the background colour.

Attributes statistics and charts

The Statistics tool (Figure 8) displays a table with descriptive statistics of the selected numeric attribute, and of the currently selected features from the same layer. These statistics give an

overview of the overall distribution and range of values, and are helpful to understanding the results, identifying anomalies, and decide on the most appropriate symbology settings. The Charts tool (Figure 9) shows simple interactive charts of the selected attribute. The histogram and scatter plot can be zoomed in and panned freely by the user, and the currently selected features are displayed in red. For the scatter plot (Figure 9b), the Y axis (dependent variable) remains constant allowing the quick comparison of different independent variables selected in the attributes list. These charts are only available if the pyqtgraph Python package is installed.

Statistic	Value	Selection
Mean	1.14595585342	1.37035642571
Std Dev	0.170433853558	0.140481728418
Median	1.1428182	1.36595425
Minimum	0.70366877	1.071714
Maximum	1.7796462	1.7796462
Range	1.07597743	1.07597743
1st Quart	1.0310065	1.250248925
3rd Quart	1.25660885	1.4731474
IQR	0.22560235	0.222898475
Gini	0.0841057505602	0.0580875112952

Figure 8: Descriptive statistics of the selected attribute values, with the statistics for the data of the selection on the map.

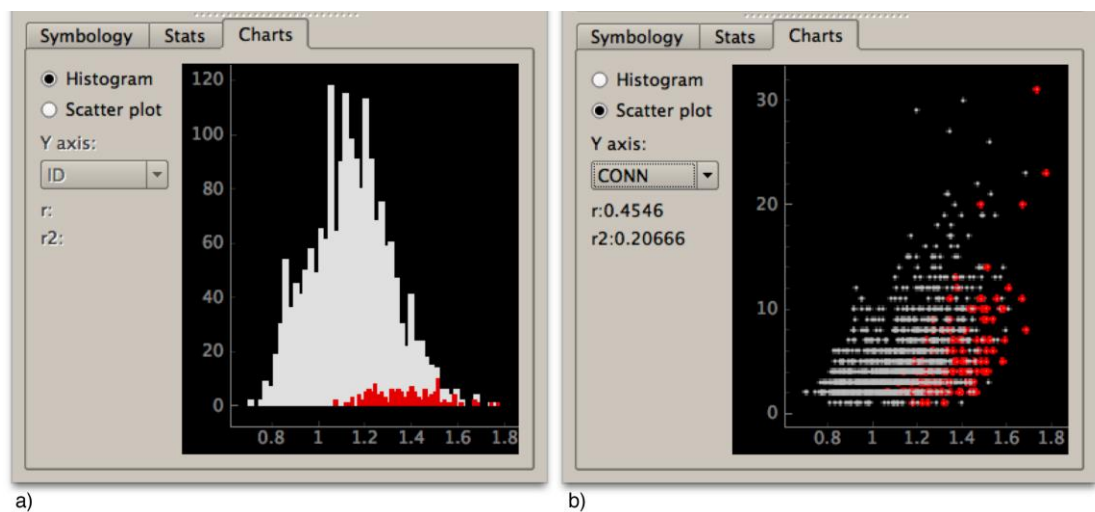


Figure 9: Histogram (a) and scatter plot (b) of the selected attribute values. In red are the data points corresponding to the selected features on the map.

4. Future development

The SST is a long-term open source project currently managed by the Space Syntax Laboratory at the Bartlett School of Architecture, UCL. The current release only covers a fraction of the stages and operations of space syntax research workflows listed in Table 1. While some of the more general functionality will probably be added to the fast evolving QGIS project, the SST should continue to grow, with fixes and enhancements to existing functionality, new functionality added to the current modules, and the development of new modules. Some improvements planned for future versions of the SST, namely: adding two way interaction between the map and the attributes explorer charts; a catchment area analysis module; or an urban morphology module for street and block metrics.

As an open source project, open to the wider communities of space syntax researchers and QGIS users, the direction of development should be informed by the needs of that community, and open to contributions in the form of new code, fixes, testing or documentation.

5. Conclusion

In this article we introduced the ‘Space Syntax Toolkit’, a Python plug-in for QGIS that integrates depthmapX[net]. The use of GIS with space syntax analysis is of growing importance to address a wide range of empirical studies. However, the space syntax research process has a set of specific stages and operations that can benefit from integrated workflows in GIS, making it more accessible but also efficient to a range of different users. The SST implements with its initial modules the basic workflow of axial and segment map analysis, from map production to publication of the results, emulating in part the exploratory spatial analysis features of depthmapX. This should offer a smoother entry into QGIS to space syntax beginners but also to existing depthmapX users who do not use GIS. This open source project will continue to evolve, to offer a broader set of features to, and with the support of, the space syntax research and practice community.

Acknowledgements

This work has been funded by Platform Grant Extension: Space, Time and Interaction – EPSRC GR/S64561/01.

We would like to thank the UCL researchers that offered insight into their practices to help define the user requirements. In particular, we would like to thank Stephen Law and colleagues at Space Syntax Ltd for their detailed analysis of the workflows and stages of space syntax projects. Finally, we would like to thank the early adopters of the SST for their feedback and questions.

References

- Chiaradia, A., Schwander, C., Gil, J., Friedrich, E., Gosset, A., (2008). ‘Mapping the intangible value of urban layout (i-VALUL): Developing a tool kit for the socio-economic valuation of urban areas, for designers and decision makers’, In: Timmermans, H., DeVries, B. (Eds.), *9th International Conference on Design & Decision Support Systems in Architecture and Urban Planning*. Eindhoven University of Technology, Eindhoven, The Netherlands.
- Chiaradia, A., Webster, C., Cooper, C., (2012). ‘sDNA - A software for spatial design network analysis’. Cardiff University, Cardiff, UK.
- Csardi, G., Nepusz, T., (2006). ‘The igraph Software Package for Complex Network Research’. In *InterJournal Complex Systems*, p.1695.
- Gil, J., (2014), ‘Space Syntax Toolkit for QGIS’, Version 0.1.0, Open Source, Available at: <https://github.com/SpaceGroupUCL/qgisSpaceSyntaxToolkit>
- Gil, J., Stutz, C., Chiaradia, A., (2007). ‘Confeego: Tool Set for Spatial Configuration Studies’, In: Turner, A. (Ed.), *New Developments in Space Syntax Software*. Presented at the 6th International Space Syntax Symposium, Istanbul Technical University, Istanbul, p.15–22.
- Jiang, B., Claramunt, C., (2002). ‘Integration of Space Syntax into GIS: New Perspectives for Urban Morphology’. In *Transactions in GIS* 6, p.295–309. doi:10.1111/1467-9671.00112
- QGIS Development Team, (2014). ‘QGIS Geographic Information System’. Open Source Geospatial Foundation Project. Available at: <http://qgis.osgeo.org/en/site/>
- Sevtsuk, A., Mekonnen, M., (2012). ‘Urban network analysis. A new toolbox for ArcGIS’. In *Revue Internationale de Géomatique* 22, p.287–305.
- Turner, A., (2007). ‘UCL Depthmap 7: From Isovist Analysis to Generic Spatial Network Analysis’, In: Turner, A. (Ed.), *New Developments in Space Syntax Software*. Presented at the 6th International Space Syntax Symposium, Istanbul Technical University, Istanbul, p.43–51.
- Varoudis T., (2012), ‘depthmapX Multi-Platform Spatial Network Analysis Software’, Version 0.30 Open Source. Available at: <http://varoudis.github.io/depthmapX/>