Developing Space Syntax Tools for Free and Open Source Software for GIS

Burak Beyhan
Department of City and Regional Planning
Faculty of Architecture, Mersin University
Mersin, Turkey
burakbeyhan@mersin.edu.tr

***Abstract* — The aim of this paper is to reveal the potential of Free and Open Source Software for GIS (FOSS4GIS) in Space Syntax (SS) analysis. For this purpose, firstly the existing software exclusively developed for SS are reviewed in order to unveil both their modalities to produce the key measures of SS and illustrate the lack of their cooperation with GIS and Social Network Analysis (SNA) software. After discussing basic SS parameters, several FOSS4GIS are examined in order to divulge their potential for the extraction of adjusted graphs and calculation of basic SS parameters by presenting the core algorithm behind the plugins created employing the scripting facilities in the respective software programs. This paper exposes that there are rich avenues for future research in the field of analysis of social networks and spatial configurations if the potential of FOSS4GIS and the graph theoretical background of SNA and SS are properly utilized.**

Keywords-space syntax analysis; GIS; free and open source software (FOSS); social network analysis (SNA)

I. Introduction

Both Space Syntax (SS) studies and the availability of Free and Open Source Software (FOSS) for Geographic Information Systems (GIS) have increased tremendously in recent years. Yet, there is no significant interest among the scientific community studying SS for the employment of FOSS for GIS (FOSS4GIS) in their spatial analyses in spite of the plugin based and flexible architecture of the latter. This paper argues that a close collaboration is extremely necessary between not only SS studies and FOSS4GIS, but also SS studies and Social Network Analysis (SNA). Indeed, although SS parameters were first developed at the University College London (UCL) during the late 1970s and the early 1980s in order to analyze the spatial configurations [1], some of the parameters (such as connectivity, total depth and mean depth) employed in SS have already been used by those social scientists developing or employing SNA (formerly known as socio-metrics) in their studies (see [2] for a short review of the historical development of SNA), albeit under different names (see especially [3, 4] for an elaboration of the concept of centrality in the respective studies).

It should be emphasized that these two fields of study heavily draw on the graph theory borrowed from mathematics. In this respect, graph theory, the essence of the method of analysis employed in SNA and SS is also actively employed in other disciplines within the realm of natural and applied sciences (see, among many others, [5, 6] for chemistry, and see [7] for bioinformatics and medicine). In similar ways, SS analysis based on the graph theory is also enthusiastically employed by a range of disciplines including not only city and regional planning (see, among many others, [8-10]) and architecture (see [11-13]), but also especially archeology (see [14, 15]) and research on both behavioral and spatial sciences (see [16, 17]), and real estate (see [18]). In spite of this last group of experiments which seem to be products of an interdisciplinary focus, today, application of SNA in social sciences is more widespread than SS.

In order to unveil the necessity for the close cooperation between FOSS4GIS, SS and SNA, firstly existing software programs exclusively developed for SS analysis will be elaborated according to their capacity to cover different modalities of SS. In this elaboration, particular attention will be paid to the capacity of the available SS software programs to perform line-network analysis and their capability to read and write in different file formats that can be easily handled by a standard GIS software package. Subsequently, after presenting the basic SS parameters, most popular FOSS4GIS including OpenJUMP, OrbisGIS, gvSIG, MapWindow, SAGA, Thuban, Quantum GIS (QGIS) and OpenEV will be reviewed together with R-Project in order to reveal the basic characteristics and potential of the respective software programs for the extraction of the adjusted graphs and the calculation of basic and widely used SS measures (such as connectivity, integration and depth) by explaining the core algorithm behind the plugins created using the scripting facilities in the respective programs.

Additionally, it will be discussed that the provision of some extra output options in SS plugins or scripts is important for those willing to employ the algorithms available in the software programs specifically developed for SNA. What is evident from this study is that there are rich avenues for future research in the field of analysis of spatial configurations and social networks if those involved in SS studies properly utilize the potential of both SNA and especially FOSS4GIS. The plugins developed by employing FOSS4GIS programs can be easily integrated into other FOSS programs without any restriction and further modified to include other functionalities thanks to GNU GPL (General Public License). Extentionability offered by the general architecture of FOSS is crucial to the development of not only new functionalities but also special packages designed for specific communities of users.

II. Existing Space Syntax Software Programs

As noted above, space syntax is one of the analysis method employed in a wide range of disciplines including city and regional planning, architecture, archeology, geography and behavioral sciences. Yet, the majority of the software programs available for the employment of space syntax algorithms are not open source even though some of them can be used freely without any payment provided that they are used for academic, not commercial purposes. Thus, prospective users of these software programs can not modify the algorithm embedded in the original software programs if they are willing to produce new parameters based on their own contribution to the field.

Even some of the software programs designed for space syntax analysis is only available upon the formal request of the prospective user who are sometimes required to fill a form approved by the director or head of his or her institution, which is quite discouraging for a potential user if he or she conceives this restricting his or her freedom of conducting research. The developer of software programs may also require some further information from the potential user. Even it is sometimes observed that the respective software programs are allowed to be used for academic purposes provided that "[a]t the reasonable request of [the developer], [users] will supply copies of any spatial models created for incorporation within [their] Spatial Form Database" [19]. Thus, it is clear that the respective developers not only strictly monitor the community of users but also request the spatial models created by the users for their specific research, which actually establishes and sustains an unequal exchange between the developer and users.

Within this context, in this paper, firstly the existing software programs which are mostly not open source but available for the space syntax analysis are reviewed in order to illustrate the need for the development of new plugins operational on a series of FOSS4GIS. In this respect, it is important to note that there are already some studies conducted to review the software programs available for space syntax analysis (see for example Dong et al. [20] for a comparison of Syntax2D, a FOSS developed for SS analysis, with Depthmap, OmniVista, Confeego and Spatialist, also the volume edited by Turner [21] for the information about Spatial Positioning Tool (SPOT) [22], WebmapAtHome [23], Confeego [24], Syntax2D [25], Segmen [26], Place Syntax Tool [27] and UCL Depthmap 7 [28]). In order to compare the available space syntax software programs, a simple table (see Table I) is constructed to show the capacities of the respective programs according to some basic modalities and input-output formats involved in the calculation of space syntax parameters.

Accordingly, although the majority of the plugins created for this study can perform only line network analysis, in Table I available SS software programs are checked whether or not they provide users with the raster based analysis. The visibility analysis based on raster data which is also part of space syntax studies is out of the scope of this paper and it is designated by the author as a separate package for which again a series of scripts will be produced as plugins operational in a series of FOSS4GIS that are most appropriate for such kind of a project due to the availability of functions required for raster analysis. In a similar vein, available SS programs are also checked whether or not they provide users with convex spaces analysis. Although, parallel to line networks, convex spaces are stored in vector file format, they should be further processed before creating adjusted graphs from the available spatial databases. Some of the plugins created for this study by employing the built-in functions capable of detecting whether or not geographic features intersect each other can potentially calculate basic space syntax parameter also for convex spaces. Nevertheless, production of convex spaces from the original spatial database requires some extra treatment.

TABLE I. Charateristics of Existing SS Programs.

SS Software	**Modalities**			***Available Input-Output Formats***	
	Line Network Analysis	***Convex Spaces Analysis***	***Raster-Grid Analysis***	***input***	***output***
Syntax2D [25]	✓	-	✓	dxf	ss2, csv
Depthmap [28]	✓	✓	✓	dxf, mif, gml, cat, rtf, ntf	txt, mif, graph
OmniVista [29]	-	-	✓	pict	pict
Confeego[a] [24]	✓	-	-	tab[d]	tab[d]
Spatialist[b] [30]	✓	✓	-	dgn[d]	dgn[d]
SPOT [22]	✓	-	-	dxf	
WebmapAtHome [23]	✓	-	-	dxf	svg, csv
Segmen [26]	✓	-	-	mif[d]	mif[d]
Place Syntax Tool[a] [27]	✓	-	-	tab[d]	tab[d]
Axwoman[c] [31]	✓	✓	-	shp[d]	shp[d]
GRASS-plugin [33]	✓	-	-	shp[d]	shp[d]
Mindwalk [32]	✓	-	-	dxf, txt	dxf, txt

a. A plugin dependent upon MapInfo.

b. A plugin dependent upon MicroStation.

c. A plugin dependent upon ArcInfo.

d. These software programs (scripts) are actually plugins that can be run via different GIS software programs that are capable of opening a wide range of vector file formats. For this reason only the native vector format of the respective GIS program is listed in the table.

Among the software programs listed Table I only Syntax2D and GRASS are known to be open source software programs or scripts that can run via an open source software program. As it is seen from the table, there is also some space syntax plugins that can be run in GIS programs. Nevertheless, except for the script described by Wang and Liao ([33]) and run in GRASS that is a complex program for standard users to employ, all the plugins listed in Table I run in a GIS software program that is not free and open source, albeit the respective plugins themselves can be freely available. Fortunately, FOSS which is increasingly being available and spreading all over the world creates a potential in order to overcome this obstacle. FOSS4GIS have particularly experienced a rapid development in a short time period thanks to the public bodies initiating and supporting the projects paving the way for the development of the respective software programs, and also a group of volunteers all over the world sustaining the projects with their remarkable contribution (see [34, 35] for comparisons of different FOSS4GIS).

III. BASIC SPACE SYNTAX PARAMETERS

In SS the urban space is described by using an adjusted graph in which lines (such as streets) are represented as nodes, and the interconnections between them are shown as edges linking nodes. It is important to note that as they employ the same mathematical infrastructure, graph theory, most of the parameters developed for SNA and SS actually overlap with each other. For example, SNA's degree centrality which is based on the simple counting of the ties incoming or outgoing from a node in a network is called connectivity (1) in SS studies. Since it is a local measure, degree centrality is not a prominent measure for large areas [36]. Although other forms of centrality measures used in SNA are not formally available in standard space syntax software programs provided by various research centers, one can easily discover that 'closeness centrality' of a node in SNA can be defined in SS as the reciprocal of 'mean depth' (4) which is the average length of the shortest paths between the respective node and every other node. In this respect, 'closeness centrality' can be "interpreted as the ability to access information through the 'grapevine' of network members" [37] because nodes characterized with high closeness centrality have low average length of the path to all the other nodes in the network. The formulas for some of the graph theoretic parameters that can be calculated by employing the plugins produced for this study can be given as below [1];

$$c_i = k \tag{1}$$

where c_i is the connectivity value for basic spatial unit (BSU) i, k is the number of other BSUs connected to BSU i;

$$ctr_i = \sum_{j=1}^{m} \frac{1}{c_j} \tag{2}$$

where ctr_i is the control value for BSU i, m is the number of neighbors of BSU i, and c_j is the connectivity value of BSU j;

$$td_i = \sum_{j=1}^{n} d_{ij} \tag{3}$$

where td_i is the total depth value for BSU i, d_{ij} is the geodesic distance between BSU i and BSU j, and n is the total number of BSUs;

$$md_i = \frac{td_i}{n-1} \tag{4}$$

where md_i is the mean depth value for BSU i, td_i is the total depth value for BSU i, n is the total number of BSUs;

$$g_i = \left(\frac{\dfrac{2(md_i - 1)}{k-2}}{\dfrac{2(n(\log_2((n+2)/3)-1)+1))}{(n-1)(n-2)}} \right)^{-1} \tag{5}$$

where g_i is the global integration value for BSU i, md_i is the mean depth value for BSU i.

Although SS parameters such as connectivity (1), total depth (3), and mean depth (4) directly correspond to some of the parameters already available in SNA, this is not the case for all the parameters developed for SS studies. Nevertheless, some other parameters developed for SNA provide us with approximate measures for the respective parameters used in SS studies. For example, it is known that 'closeness centrality' correlates with the global integration (5) values. Theoretically, BSUs associated with high closeness centrality and global integration values reveal the core of the network.

In SS, two other important parameters are 'local depth' and 'local integration' values calculated for each BSU by taking into account the network within a range of generally three closest BSUs. Local integration is basically employed to unveil the variation of integration across the network within a given radius. Actually, both 'local depth' and 'local integration' values reflect the average length of pedestrian walks, which has been proven empirically by many SS studies (see [38, 39]). It is noticeable that there is no direct corresponding parameter for local integration value in SNA. Nevertheless, it is also important to note that some of the critical parameters (such as 'betweenness centrality') available in SNA are not available in a standard SS software program. It is especially for this reason that the employment of the outcome of plugins as inputs in SNA software is very important.

Based on this elaboration of the basic parameters of SS, in the next section the algorithms employed for the production of the plugins will be presented. Nevertheless, before proceeding into the next section, it is important to emphasize that all the plugins described in this paper have been basically designed to handle line-networks (such as street networks), albeit some of them can also analyze convex spaces owing to the built-in functions available in FOSS4GIS programs employed to develop the respective plugins. Although the 'geodesic distance' upon which majority of the basic parameters of SS are built does not indicate the metric distance between BSUs, as an expression of fewest turns it may assert much more influence on the route finding behavior of the people [40]. Nevertheless, a graph theoretic conceptualization for the spatial configurations of line-networks can be meaningful only if the semantic and ontological qualities of BSUs involved in the network are acknowledged as it is done by Beyhan (see [41]).

IV. ALGORITHMS UTILIZED IN FOSS4GIS PLUGINS

In constructing the algorithms designed for the creation of plugins, firstly scripting facilities and software libraries available in each FOSS4GIS have been checked whether or not they are capable of testing intersections between the geographic features by employing available built-in functions (see Fig. 1). If there is no built-in function available for this task, a series of scripts has been produced in order to calculate the bounding box (BB) of each geographic feature in the spatial database (see Fig. 2). BB is actually very important in the determination of whether or not geographic features intersect each other. Instead of checking intersection of each segment of a feature with any segment of another feature, it would be wise first to check whether or not their respective bounding boxes overlap each other. If BBs do not overlap with each other, it will be superfluous to conduct a segment vise checking for the intersection of two geographic features. Thus, BBs help us save time in the determination of whether or not two features intersect with each other.

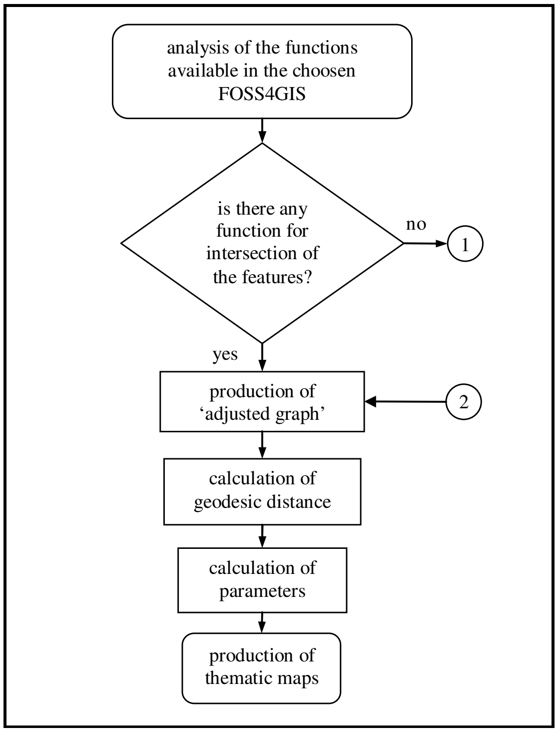

Figure 1. Algorithm employed for the production of SS plugins (1).

Some FOSS4GIS have built-in functions for the calculation of BBs (see Table II). Yet, the calculation of BBs is not a difficult procedure once all the coordinates of a geographic feature in the database are known. Calculation of maximum and minimum values for x and y coordinates actually provides us with the corner coordinates of BB of the respective features. Once it is known that BBs of two objects overlap with each other, each segment of the respective geographic features can be checked against each other in order to determine whether or not they intersect. The typical formula for the calculation of the intersection between two line segments is given as below;

Considering the fact that the equation for any line segment can be written as $y = a + bx$ where *b* is the slope of the line and *a* is a constant, if the coordinates of two points placed along the line are known, then $b = (y_2 - y_1)/(x_2 - x_1)$. Subsequently, if $y = a + bx$ is solved for *a* for one of the points (let it be the first point), then $a = y_1 - bx_1$. Based on these results, the intersection point for two lines can be easily calculated by solving the following equation first for *x*, and then for *y*; $a_1 + b_1x = a_2 + b_2x$ where subscripts *1* and *2* placed after *a* and *b* stand for, respectively, the first and the second segments, and which can also be rewritten as following; $x = -(a_1-a_2)/(b_1-b_2)$. If $y = a + bx$ is solved for *y* for one of the segments (let it be the first segment), then $y = a_1 + b_1x$. As the intersection point is calculated for the equations representing continuous lines, it may not lie along the individual segments. Thus, the respective point should be checked whether or not it lies along the actual segments. In other words, $((x_1-x)*(x-x_2)>=0$ and $(x_3-x_i)*(x-x_4)>=0$ and $(y_1-y)*(y-y_2)>=0$ and $(y_3-y)*(y-y_4)>=0)$ where subscripts *1-2* and *3-4* stand for, respectively, the end points of the first and the second segments should be satisfied. Yet, it should be noted that special treatments are required for the cases where any of the two segments lies along a horizontal line for which the calculation of *b* gives a "division by zero" error.

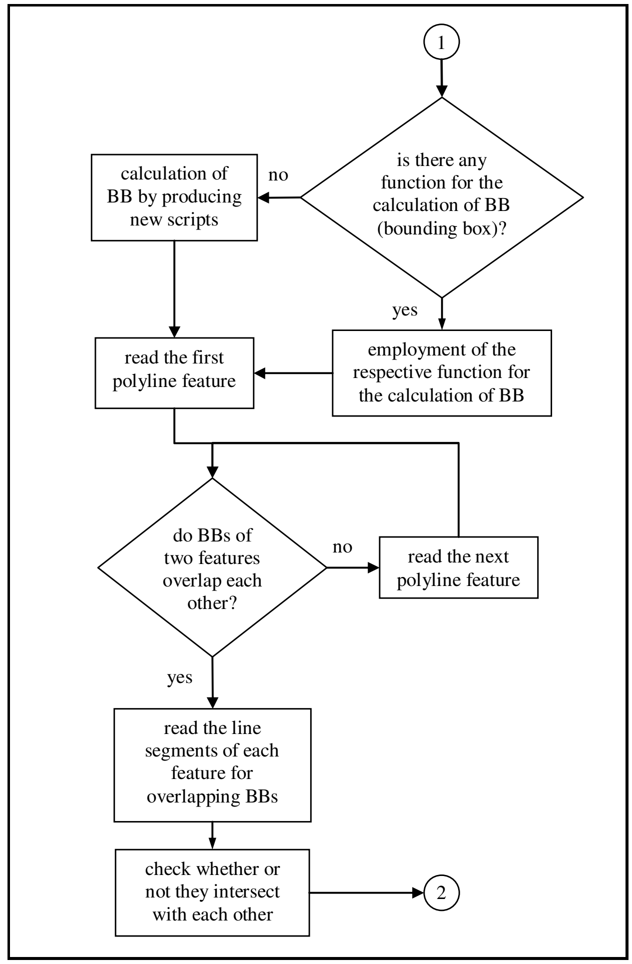

Figure 2. Algorithm employed for the production of SS plugins (2).

V. COMPARSION OF THE PERFORMANCE OF FOSS4GIS

A series of plugins has been produced for different FOSS4GIS programs including OpenJUMP (BeanShell, Python and Java), OrbisGIS (BeanShell), MapWindow (Visual Basic), SAGA (C++), QGIS (Python), Thuban (Python), OpenEV (Python) and gvSIG (Jython), and also R, actually a GNU based statistical software project which is similar to the S language and environment, by translating the formulas and algorithms described in the previous sections into the software codes operational in Java (Bean-Shell), Visual Basic, C++, Python, Jython and R. The respective plugins will be available over internet by the end of 2011 at the following website: http://mekandizim.mersin.edu.tr/.

Although some experiments were also conducted with ILWIS (C++), uDig (Java), TerraView (C++) and GRASS (ANSI-C) for which there is already a script produced by Wang and Liao (see [33]), these attempts were not successful because of the unresolved issues that the author experienced. Nevertheless, it is known that respective software programs can be used to develop similar kinds of plugins. It should be emphasized that a standard GIS software program can not only perform spatial queries that are critical for the determination of whether or not geographic features intersect but can also read and write a wide range of vector and raster file formats [35].

During the creation of plugins, a simple table (Table II) is constructed in order to compare the performance of FOSS4GIS according to the functionalities that can used to build adjusted graphs and write the results of the analysis to the attribute table of the original GIS file or to an external text file. What is evident from Table II is that FOSS4GIS programs developed by employing Java are more successful in the determination of intersection of geographic features owing to the built-in functions available in the respective programs (OpenJUMP, OrbisGIS and gvSIG) for this specific task that is critical in the construction of adjusted graphs required for the calculation of space syntax parameters. This success of FOSS4GIS based on Java actually stems from JTS (Java Topology Suit) on which the respective software programs draw.

TABLE II. AVAILABLE FUNCTIONS IN FOSS4GIS FOR SS.

FOSS4GIS	**Functionalities**			
	Checking Intersection of Features	***Calculation of Bounding Box***	***Writing Parameters to Table***[a]	***Writing Parameters to text-dbf***
OpenJUMP	✓	✓	✓	✓
OrbisGIS	✓	✓	✓	✓
gvSIG	✓	✓	–	✓
MapWindow	–	✓	✓	✓
SAGA	–	–	✓	✓
QGIS	✓	✓	✓	✓
Thuban	–	✓	– ✓	✓
OpenEV	–	–	✓	✓
R-project	–	✓	– ✓	✓

a. Table here stands for the attribute table and "– ✓" denotes plugins that can not write to the original attribute table though they can easily write the results of the analysis to a new GIS file.

QGIS has also built-in functions that can be used to check whether or not geographic features intersect each other. As either there was no similar kinds of function or the author could not explore the respective functions, intersection of geographic features in other software programs is determined by either employing other available built-in functions (such as BB) or developing new scripts as discussed in the previous section. Among FOSS4GIS in which intersection of features could not be checked by using a single function, BB functionality has been found to be available for MapWindow, Thuban and interestingly R Project. Only for SAGA and OpenEV, the author has written new scripts in order to check whether or not geographic features intersect with each other.

After handling construction of adjusted graphs according to the intersections between geographic features, all the scripts required for the calculation of both geodesic distance and subsequently basic parameters of space syntax have been produced by the author according to the sub-algorithms developed for this purpose. Due to the space available to us, respective algorithms could not be given here.

Although the most successful FOSS4GIS platform for the development of space syntax plugins seems to be the one based on Java, it is observed that, as usual, the scripts produced in C++ (SAGA) and Visual Basic (MapWindow) work faster compared with the ones written in Java and Python (QGIS, Thuban and OpenEv). Nevertheless, it is important to emphasize that among the other FOSS4GIS the most capable one for the development of plugins is OpenJUMP in which one can develop plugins by employing BeanShell, directly Java or Python. Especially console applications (BeanShell console in OpenJUMP and OrbisGIS, Python (or Jython) console (or Shell) in again OpenJUMP, OpenEV and gvSIG) help developers explore and test the built-in functions available in the respective software platforms.

VI. CONCLUDING REMARKS

The plugins described in this paper can successfully build adjusted graphs from a given line-network and subsequently calculate the geodesic distance and the basic space syntax parameters. Some of the plugins can also build adjusted graphs for convex spaces owing to the built-in functions available in FOSS4GIS employed for the development of the respective plugins. It is very evident from this paper that FOSS4GIS have a great potential for the employment and exploration of graph theoretical analysis thanks to their flexible and extensible architecture.

It is expected that the plugins developed in this study will be employed in both design and planning studies in order to develop some objective criteria for the decisions related to socio-spatial settings. Thus, space syntax parameters should be considered not only in relation to their potential to discover the spatial configuration and social network embedded in a place but also with respect to the fact that they can be actively employed as a scientific tool developed to intervene into the process of design and planning. As the plugins developed by employing the formulas and algorithms presented in this study will be available to the end users as a FOSS, it is believed that it will trigger the development of further research in the analysis of socio-spatial settings.

Moreover, since some of the outcomes of the plugins are designed to be used as inputs for use in the software programs specifically designed for SNA, compared with the existing SS software packages the end users will be able to draw on a wide range of parameters (such as betweenness, authority, eigen-vector and hub centralities) in their graph theoretic studies. There is no doubt that the differentiation of intervention policy according to some objective criteria defined with reference to the outcome of the space syntax analysis helps us consolidate different aspects of socio-spatial processes.

ACKNOWLEDGMENT

The author is grateful to Mersin University for the research fund used to conduct the study presented in this paper and TÜBİTAK for the financial support to participate in the 19th International Conference on GeoInformatics.

REFERENCES

[1] B. Hillier and J. Hanson, The social logic of space, Cambridge, Cambridge University Press, 2003.

[2] B. Beyhan. Toplumsal-bilişsel yakınlık ve işgücü hareketliliği: hemşehrilik bağları üzerinden bir irdeleme (Social-cognitive proximity and labor mobility: An examination through the fellowmen ties). Anadolu Üniversitesi Sosyal Bilimler Dergisi (Anadolu University Journal of Social Sciences), 2011, vol. 11(3), (in press).

[3] L.C. Freeman. A set of measures of centrality based on betweenness. Sociometry, 1977, vol. 40(1), pp. 35-41.

[4] L.C. Freeman. Centrality in social networks: Conceptual clarification. Social Networks, 1978, vol. 1, pp. 215-239.

[5] A.T. Balaban (ed). Chemical applications of graph theory. London: Academic Press, 1976.

[6] A.C. Puiu, O. Sinanoğlu. The graph theoretical chemistry of Moebius vs. Hueckle type organic molecules. Chimica Acta Turcica., 1978, vol. 6, pp. 1-16.

[7] P.F. Jonsson, T.Cavanna, D. Zicha, P.A. Bates. Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. BMC Bioinformatics, 2006, vol. 7(2), pp. 1-13.

[8] Y. Asami, A.S. Kubat, and C. Istek. Characterization of the street networks in the traditional Turkish urban form. Environment and Planning B: Planning and Design, 2001, vol. 28, pp. 777-795.

[9] A.S. Choia, Y.O. Kima, E.S. Oha, Y.S. Kimb. Application of the space syntax theory to quantitative street lighting design. Building and Environment, 2006, vol. 41(3), pp. 355–366.

[10] K.P. Baran, A.D. Rodríguez, J.A. Khattak. Space syntax and walking in a new urbanist and suburban neighbourhoods. Journal of Urban Design, 2008, vol. 13(1), pp. 5-28.

[11] T. Bellal. Understanding home cultures through syntactic analysis: The case of Berber housing. Housing, Theory and Society, 2004, vol. 21(3), pp. 111-127.

[12] R. Aboukhater. Analysis of spatial structure and social significance of a sample of hammams in Mediterranean cities. International Journal of Architectural Research, 2008, vol. 2(3), pp. 109-121.

[13] S. Malhis The new upper-middle class residential experience: A case study of apartment flats in Jordan using the logics of Burden, Hillier and Hanson. Architectural Science Review, 2008, vol. 51(1), pp. 71-79.

[14] M. Cutting. The use of spatial analysis to study prehistoric settlement architecture. Oxford Journal of Archaeology, 2003, vol. 22(1), pp. 1-21.

[15] T.J. Ferguson. Historic Zuni architecture and society: An archaeological application of space syntax. Tucson: University of Arizona Press, 1996.

[16] J. Peponis, S. Bafna, R. Bajaj, J. Bromberg, C. Congdon, M. Rashid. Designing space to support knowledge work. Environment and Behavior, 2007, vol. 39(6), pp. 815-840.

[17] J.D. Wineman, F.W. Kabo, G.F. Davis. Spatial and social Networks in organizational innovation. Environment and Behavior, 2009, vol. 41(3), pp. 427-442..

[18] R. Enström, O. Netzell. Can space syntax help us in understanding the intraurban office rent pattern? Accessibility and rents in downtown Stockholm. The Journal of Real Estate Finance and Economics, 2008, vol. 36, pp. 289–305.

[19] Space Syntax Laboratory, UCL: Terms and conditions of the licensing agreement. Available online at: http://217.155.65.93:81/terms.htm

[20] T. Dong, M. Hanratty, A. Torres, L. Xu. Comparative evaluation of Syntax2D. [2009-03-28]. Available online at: http://www.adamatorres.com/docs/Portfolio/Projects/syntax20/Syntax2.0_Comparative_Evaluation.pdf.

[21] A. Turner, Ed. New developments in space syntax software. İstanbul Technical University, Faculty of Architecture, İstanbul, 2007. [2010-06-15]. Also availanle on line at: http://eprints.ucl.ac.uk/4109/1/4109.pdf.

[22] H. Markhede, P. Miranda Carranza. Spatial Positioning Tool (SPOT). In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 1-6.

[23] N.S. Dalton. WebmapAtHome. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 7-13.

[24] J. Gil, C. Stutz, A. Chiaradia. Confeego: Tool set for spatial configuration studies. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 15-22.

[25] J. Wineman, J. Turner, S. Psarra, S.K. Jung, N. Senske. Syntax2D: An open source software platform for space syntax analysis. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 23–26.

[26] S. Iida. Segmen: A programmable application environment for processing axial maps. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 27-34.

[27] A. Ståhley, L. Marcusy, A. Karlströmz. Place Syntax Tool – GIS software for analysing geographic accessibility with axial lines. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 35–42.

[28] A. Turner. UCL Depthmap 7: From isovist analysis to generic spatial network analysis. In A. Turner, Ed. New developments in space syntax software, İstanbul Technical University, 2007, pp. 43-51.

[29] R.C. Dalton, N. Dalton. OmniVista: An application for isovist field and path analysis. Paper presented at the 3rd International Space Syntax Symposium, Atlanta. [2010-08-15]. Availanle on line at: http://undertow.arch.gatech.edu/homepages/3sss/papers_pdf/25_conroy&dalton.pdf.

[30] E. Edgü. A syntactic approach to space. Paper presented at the 1st International Symposium of Interactive Media Design, 2003.

[31] B. Jiang. Axwoman: An ArcView extension for urban morphological analysis. Proceedings of Geoinformatics'99, 15-19 June, Beijing, 1998, pp. 394-401.

[32] L. Figueiredo. Mindwalk: A Java based software for spatial analysis. (2005) [2009-10-15]. Availanle on line at: http://www.mindwalk.com.br/papers/Figueiredo_2005_Mindwalk.pdf.

[33] W.C. Wang, H.J. Liao. Implementing space syntax in an open source GIS: GRASS GIS approach. Paper presented at the 6th International Space Syntax Symposium, İstanbul. (2007) [2009-06-15]. Availanle on line at: http://www.spacesyntaxistanbul.itu.edu.tr/papers\longpapers\100 - Wang Liao.pdf.

[34] S. Steiniger, E. Bocher. An overview on current free and open source desktop GIS developments. International Journal of Geographical Information Science, 2009, vol. 23(10), pp. 1345-1370.

[35] B. Beyhan, B. Belge, F. Zorlu. Özgür ve açık kaynak kodlu masaüstü CBS yazılımları üzerine: Karşılaştırmalı ve sistemli bir değerlendirme (Free and open source desktop GIS software programs: A comparative and systematic evaluation). Harita Dergisi, 2010, vol. (143), pp. 45-61.

[36] M. Tomko, S. Winter, C. Claramunt. Experiential hierarchies of streets. Computers, Environment and Urban Systems, 2008, vol. 32, pp. 41–52.

[37] D.V. Kalamaras. Social Networks Visualizer (SocNetV). [2010-06-28]. Available online at: http://socnetv.sourceforge.net/.

[38] C. Liu. Spatial configuration and vehicle flow: Topologically analyzing and modeling the Hong Kong street network. Unpublished Master Thesis, Department of Technology and Built Environment, Programme in Geomatics - Land Surveying, 2007.

[39] B. Hillier, A. Penn, J. Hanson, T. Grajewski, J. Xu. Natural movement: Or, configuration and attraction in urban pedestrian movement. Environment and Planning B, 1993, vol. 20(1), pp. 29-66.

[40] B. Hillier, S. Iida. Network and psychological effects in urban movement. In Proceedings of the International Conference on Spatial Information Theory (COSIT 2005), 2005, pp. 475-490.

[41] B. Beyhan. Socio-spatial characteristics of labor mobility and innovation inside an industrial cluster: Some reflections from Siteler in Ankara. Paper presented at the 50th Anniversary European Congress of the Regional Science Association International, 19-23 August 2010, Jönköping, Sweden.