

Classifying Urban Geometry Impact on Solar Radiation

Anas M.Hosney Lila^{1,2}, Simon Lannon²

¹Department of Architectural Engineering, Faculty of Engineering, Tanta University, Tanta, Egypt

²Welsh School of Architecture, Cardiff University, Cardiff, UK

Abstract

This research introduces a novel way of classifying the built environment based on its geometrical features and its surroundings and link the classified buildings to its environmental performance. These features consist of urban block orientation within the configuration, urban blocks exposure to main street, building orientation within the urban block, building area and number of edges, building exposure to urban void and the height of buildings compared to its surroundings with its orientation related to its height relation

This categorization is utilized to save simulation time by reducing the number of simulated buildings in the urban configuration. The process starts with running a solar radiation simulation for some configurations fully. In these runs, each building is simulated in isolation but considering the rest of the configuration as context. These results act as a repository of saved data linked to the buildings' categorization. Then, these saved results are compared to newly generated configurations and only unique buildings with unique features are simulated. The simulation results are compared to the normally ran iterations to figure out the accuracy of this process.

Introduction

Quantifying the built environment and translating its geometry to what is known now as computer modelling is mostly based on the principles on alexander's theory of extracting the patterns of built environment (Alexander, 1979; Alexander, Ishikawa, & Silverstein, 1977) along with the work of geometrical mathematics of March (March & Steadman, 1971) this became into urban scale by the introduction of "urban grid as generator" of different urban morphologies done by (Martin & March, 1972). With the continuous development of these principles, urban geometry complexity has been addressed in various studies. Urban modelling has been trying to dismantle this complexity through creating different methods to have a clearer of urban geometry like "space syntax" (Hillier, 2007), applications of "shape grammar" principles (Duarte & Beirão, 2011; Kepczynska-Walczak & Pietrzak, 2017; Shekhawat & Duarte, 2017; Tsamis, 2017). This continuous breaking down of urban modelling into its patterns especially with the existence of parametric modelling approach and its tools, goes hand in hand with the steady progress of allowing artificial intelligence solvers in these parametric modelling platforms (Cichocka, Migalska, Browne, & Rodriguez, 2017; Harding, 2017; Rutten, 2013; University of Applied Arts Vienna Bollinger+Grohmann

Engineers., 2014; Wortmann, 2017) has led to make the ambition of having a "cognitive urban design"(König, Schmitt, Standfest, Chirkin, & Klein, 2017) closer to be approachable specially with the capability of combining urban modelling with environmental performance analysis and simulation.

The link between geometrical features and environmental performance have been studied from different perspectives (Bassett, Lannon, Waldron, & Jones, 2012; Chatzivasileiadi, M. Hosney Lila, Lannon, & Jabi, 2018; Hosney Lila & Lannon, 2017; M.Hosney Lila, Lannon, & Jabi, 2017; Robinson, 2006; Robinson et al., 2007; Vartholomaios, 2017). This paper, as a part of an ongoing research project, focuses on solar radiation as the first stage of multi-stage simulation sequence for solar radiation, energy and lighting availability performance analysis in an urban simulation framework. This study highlights the limitations and potential of urban design generation, analysis state of the art tools, i.e. ladybug tools and decoding spaces, and the integration of parametric urban generation with environmental analysis on an urban scale. This process provides a new approach to link geometrical features to performance on urban scale. It seeks for reducing the time consumed by regular simulation methods to ease the integration of simulation-based decision making to be done in the early stage of design. Time consumption for urban scale performance simulation is considered one of the major issues, this process will also allow designer interaction to decide the optimal solution by balancing designer and computer agency over the design process. The classification of buildings based on its physical features and performance is a novel way to cluster buildings and enhance the urban scale simulation time without sacrificing the accuracy of the results.

Computational solvers applications vary between different software platforms with different processes and features. because of the time consumed in multi-objective building performance analysis and the scarcity of a comprehensive urban form finding frameworks, there is still a need to provide an efficient urban modelling and building performance comprehensive analysis technique that can generate new climate responsive urban forms in the early stage of design. This research is trying to fill that gap by using Ladybug tools (Sadeghipour & Pak, 2013) for building performance analysis and applying a multi-fitness interactive solver on it within the shape grammar created in Grasshopper (Mcneel, 2014). This workflow will offer framework for designers that can generate neighbourhood urban forms for both new and existing

urban neighborhoods with the respect of climate conditions integrated and presented in a commonly used interface like Grasshopper. This will provide the designers in early stage of design with a tool that enables environmentally guided decision in the early design stages of urban design.

Methods

The process presented in this paper for simulating the urban environments is undertaken through several stage (Figure 1) is merely focused on new urban neighbourhoods with the plan to apply it on existing neighbourhoods in future stages of this project. The first stage of running is to generate the road network and the buildable areas within the given boundaries introduced by the user. To do this, the framework depends on a plug-in developed by a group of European universities. It is called “Decoding Spaces” (Koenig, Miao, Knecht, Buš, & Mei-Chih, 2017). This tool allows the generation of urban networks within a certain boundary with a certain amount of inputs and controls. The major drawback is the limitation it imposes on the design process by constraining some geometrical aspects, this research framework has overcome these limitations. Although the capabilities of this tool allow the generation of a whole urban geometry including buildings, for this research it is just used for generating the buildable areas which will be used as a start point for the needed urban geometry generation in this framework. For the preparation of the clustering phase the framework had to create a parallel

from the tool set of components. As this tool is still under development not all the number inputs are effective regarding the expected outputs. For example, the minimum block size is not changing the output of the urban configuration. The logics of creating the network is published by the research group that illustrates the logics of creating street pattern (Koenig, Treyer, Schmitt, & Zurich, 2013).

For the following stages of the framework require the creation of general urban controls not specific controls to the buildings. For example, it gives Floor Area Ratio (FAR) to control density instead of controlling the heights of buildings. The following step is to order the geometry by the block’s orientation. The orientation is defined by creating a vector from the boundary centre point to the centre point of each block. Then the angle difference from vector in the direction X is measured. These angles fall

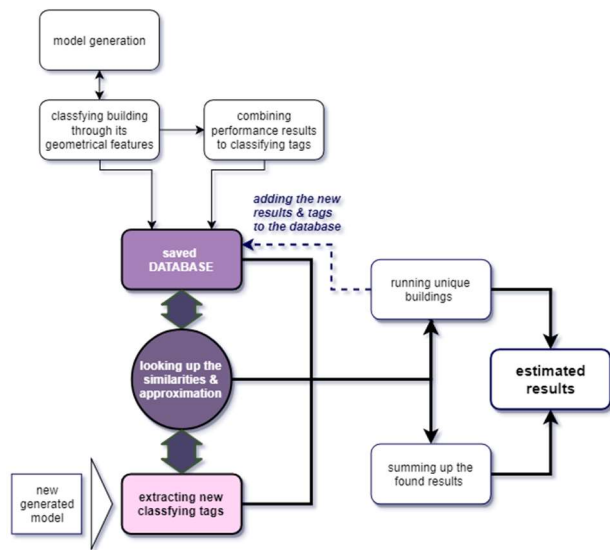


Figure 2 framework flow chart

text tagging system for each building in order to identify it based on its geometrical features.

The framework starts with boundary options to be set by the user as an input. In this paper this boundary is set to be a typical rectangular boundary created by rectangle component in grasshopper. Once the street network has been generated these are transferred to the next component “street blocks” which creates the blocks for the urban configuration then to “parcels” and divide it to the needed “buildable areas”. This sequence is inherited

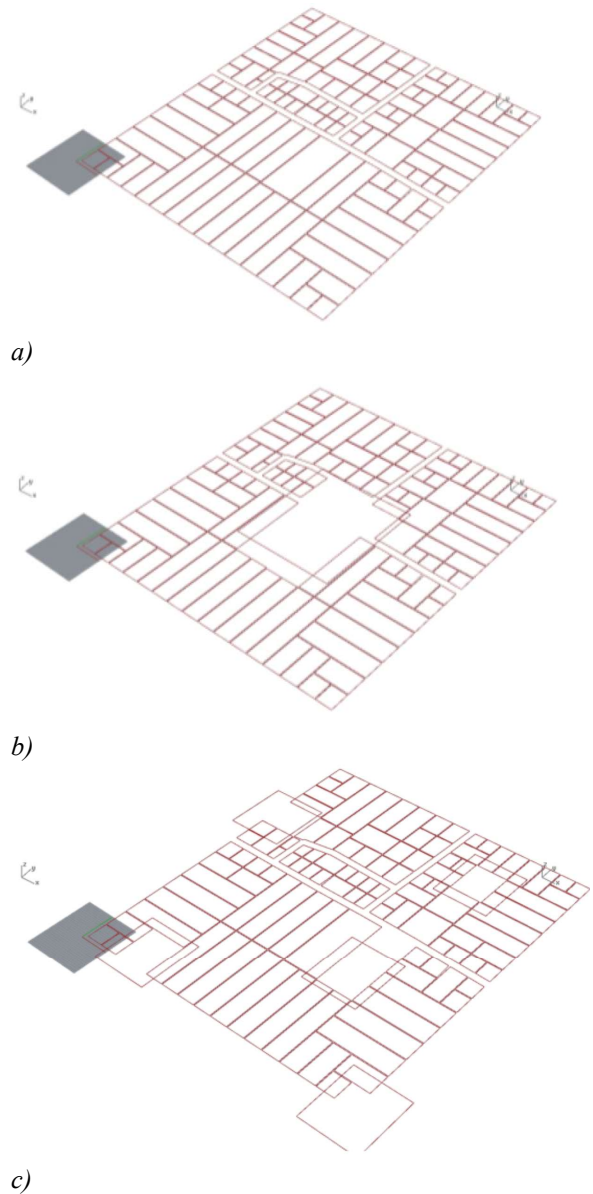


Figure 1 urban voids examples showcase. a) urban configuration with no void, b) one central urban void, c) multiple randomly generated urban voids

into either of the 8 azimuth orientations, these orientation

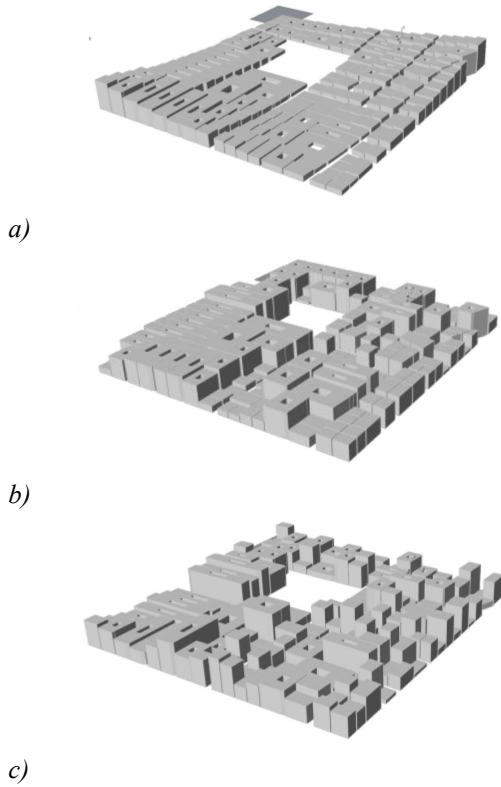


Figure 5 a) heights approximately distributed based on selected attractors, b) heights distributed based on areas (larger areas get higher extrusion), c) randomly distributed heights

by then can be tagged to each block. Afterwards, blocks are sorted by its exposure to the main street. This creates two groups one with exposure to the boundary main street and another with no exposure to the main street. This sorting is done by scaling down the boundary curve and test if the blocks intersects with it then it is exposed otherwise it will an inner block with no exposure to outer main streets. Following this an orientation sorting is done but this time for each buildable area. The process starts with creating a vector for each buildable area. These vectors were created based on the nearest distance from each buildable area's centre point to the generated street network. Then a comparison study for the angle of each of these selected vectors from vector in global X direction which starts at the centre point of each block. By this comparison, buildable area could be categorized based on its orientation within each of its blocks.

Then, it is the stage of generating urban voids (Figure 3) the framework allows 12 cases of urban voids to be created based on the user controls of urban void percentage of the total area of the original site boundary and the location of desired urban voids and the in case of multiple urban voids the user can control the percentage of each to the total targeted area of voids.

Urban voids are created by selecting the buildable areas' centre points that is located within the allocated urban void curve. Then these selected areas are removed from the list of buildable areas for the next stages. The twelve

cases consist of eight cases include the eight azimuth directions and one case for centric urban void and another for no urban void at all to let the generation be as it is and the last two cases for multiple urban voids one is random, and the other is set by user's agency.

The next phase is to differentiate between the buildable areas that are exposed to the newly created urban voids and those which does not have direct or near access to it. To do this the framework used "Isovist", a grasshopper (Mcneel, 2014) component, that is capable of testing a set of points visible from the urban void centre point in space

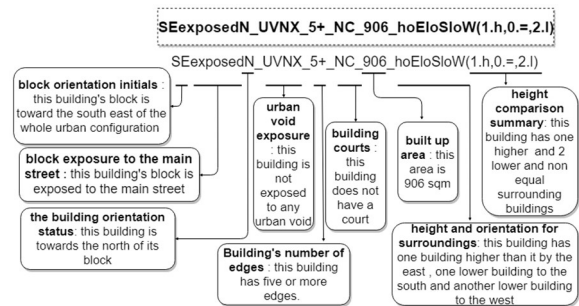


Figure 4: a show case for one building text tag explained

and with respect to the rest of the buildable areas acting as obstacles due to street openings. Within the configuration the framework had to add boundary for the "Isovist" to stop at it to prevent it from including non-exposed curves just because of the street orientation did not have any obstacles. The next stage identifies the buildable areas by its number of edges. With the enormous variation of number of edges, the area identification summed up into two groups the first is areas with 4 edges and less and the other is 5 edges and more. Another detail to the building typology which is building courts added through the next phase of building the model (Figure 3). Building courts is decided based upon the area

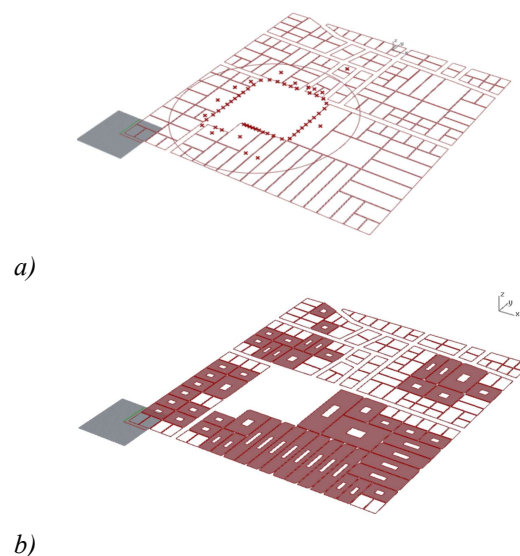


Figure 3 a): urban void exposure show case, to the b): creating central courts in buildable areas larger than the threshold

of each buildable area and its threshold can be controlled by the user. For this model the area threshold is set to 2500 sqm which means any buildable area larger than this then it will have a centric court in it with a scale of 30% of its area and this ratio is controllable too. The next stage of modelling was to introduce heights to this generated 2D configuration (Figure 4). Heights had 3 options to be decided by the user. First is to create a list of desired heights in the configuration. The number of heights in this list and the ratio of each one in the total number of heights is determined and controlled by the user with a maximum number of 10 heights. Then the order of these heights introduced in just a random way by grasshopper's random component with the exact same number of buildable areas. The second option is to order the buildable areas in a list based on heights and give it the heights from lower to higher or vice versa. The third option is to assign custom attractors and the height of each building will be assigned based on its distances from these points.

Table 1 Geometrical variables number of iterations

Geometrical category	Number of cases
Urban Void Cases	12
Building Courts	2
Maximum Arms	3
Random Angles	4
Height Distribution	3
Street Width	3
Road Setback	3
Side & Back Setback	3
Total Iterations	61,398

The last phase of identifying the model is to tag each building based on its surrounding heights. The first step is to group every building with its surrounding buildings along with its heights attached in order to be able to do heights comparison. Then a comparison is made to sort out the heights to three groups of higher, lower and equal surrounding heights. This process starts with grouping each building to its surrounding ones. This happens by isolating each building in a separate list then an offset of the centred building is used to detect the surrounding building by intersection. Then to determine its orientation to the centred building a line is created from the area centre of that building to the centre of its surrounding and the same orientation method, done before to determine blocks orientation, is repeated again for these buildings. The same way of sorting the lists of the buildable areas is repeated for the list of building heights to do a numerical comparison and include in the tag. Then an orientation comparison done for each building is tagged based on its location to the middle building.

The tag ends up telling the number of buildings higher, lower and equal to the tagged middle building and the orientation of each surrounding building. These inputs, along with initial inputs inherited from decoding spaces tool, resulted a generation of more than 61,000 urban

configurations. Solar radiation results of this pool of urban configurations were collected using ladybug tools along with time consumed to run each configuration. For this analysis, the urban configuration model was inputted as a whole single model at each run. A summary of these variables is shown in Table 1.

The last phase of identifying the model is to tag each building based on its surrounding heights. The first step is to group every building with its surrounding buildings along with its heights attached in order to be able to do heights comparison. Then a comparison is made to sort out the heights to three groups of higher, lower and equal surrounding heights. Then an orientation comparison done for each building is tagged based on its location to the middle building. The tag ends up telling the number of buildings higher, lower and equal to the tagged middle building and the orientation of each surrounding building.

The tag starts with initials for the block orientation followed by its exposure to the main street status (Figure 5). Then, the building orientation initial followed by the urban void exposure status. After this, comes the number of edges for the buildable area next to it whether the building have a court or not. Built area comes after this in numbers. The tag ends with the height comparison which starts with the height status of each surrounding building and its orientation and then between brackets it gives a summary of the height comparison status how many buildings are higher, equal or lower.

After this process of classification is still under testing to verify its accuracy and how helpful would it be to use it to save time consumed for environmental performance optimization goal in early stage of design. The testing started with analysing the different portions of the tag of buildings to know more about its accuracy. These clustering tests were done for solar radiation simulation for urban configuration as it is one of the least time

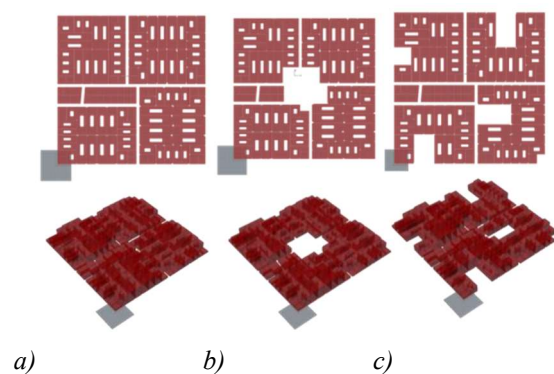


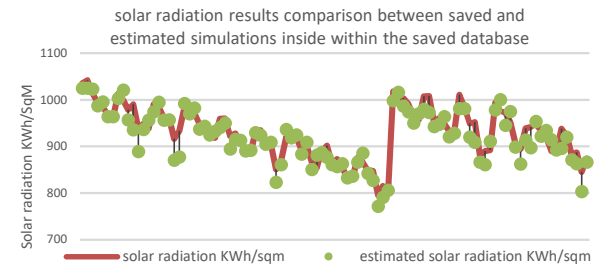
Figure 6 three examples of tested configurations for the first phase testing for case 01a. a) top & perspective views with no voids, b) top & perspective views with one central void, c) top & perspective views with multiple randomly generated voids

demanding analysis in the environmental aspects. Six iteration were assigned to do this clustering to test the tags accuracy. Those six iterations were generated from two configuration families with the change of key urban void

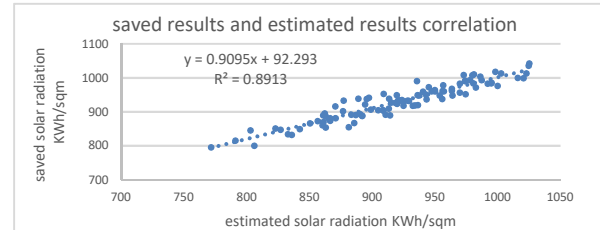
status. Then a detailed run for each building were done in all the six iteration. Then a Grasshopper definition was developed to locate the distinct buildings with the distinct tags and by calculating the average of those distinct tag based on is number of repetitions each a speculation can be done summing these averages to represent the whole configuration solar radiation. For each case a multi stage clustering have been made. First stage was with tags including all orientation aspects for buildings and blocks added to the building court status and number of edges along with urban void exposure. Second trial added heights only to the clustering tags. The following trail added built up area for each building. Finally, the surrounding heights comparison were added for the final and optimal accurate trial. Consequently, a larger sample of saved results were done and saved to have a better understanding of this identification process and its accuracy, time saving and potentials. A database of 270 urban configuration were run and saved with an average of 150 buildings in each of these configurations resulted in generating of 40,000 building tag saved with its solar radiation results. The idea is to test the accuracy of using this database of saved tags and results to estimate the performance of newly generated urban configuration from the same pool and some other configurations were tested with some variations of the initial inputs of the main generation framework.

Heights were then added to the tag described in figure 5 for the next testing stage. The second stage of testing were done by running a detection test on these saved 40,000 building tags. This was done in two phases. First phase was by selecting 100 random configurations (Figure 6) within the previously run configurations. While the second stage were to test these tags detection against another 100 configurations selected from the initial larger pool of configuration which did not has any saved tags but, each configuration has its whole solar radiation saved without dissolving it into its buildings. The framework was set to look up similar tags in the saved data base and get its solar radiation then the total result of the urban configuration is calculated by summing these looked up saved results.it is important to note that this initial detection process was set with some tolerance of the tag parameters. For example, it was looking for the same number of surrounding building with the same height comparison relationship but not necessarily the same area or height if it was not available. If building or block orientation were north west and not found it can be replaced with either north or west with the same urban void and boundary street exposure conditions. It's important to note these simulations were conducted on regular computational facilities. This simulations number of computers dedicated for the simulation varied from 4 to 6 computers over the time of conducting these runs. Each computer contained an intel i7 (8 cores,3.4 GHz) processors with 32 GB. Operating system was windows 7 64 bit. All solar radiation analysis used the weather file of Aswan city in southern Egypt (24.0889° N, 32.8998° E).

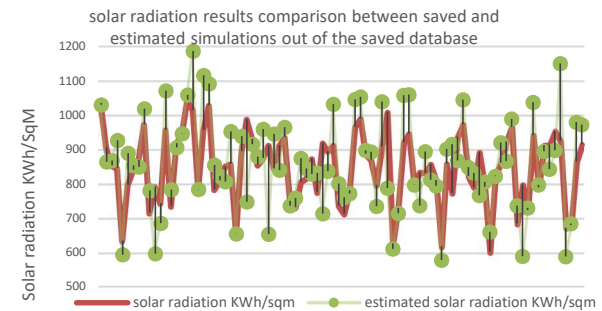
Results



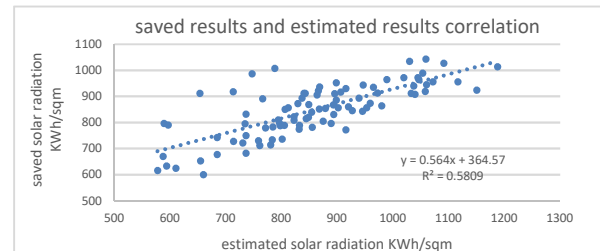
a)



b)



c)



d)

Figure 7 a) the graph shows the results comparison between the saved and estimated results, b) shows the correlation between saved and estimated results for the first phase of this analysis c) the graph shows the results comparison between the saved and estimated results. d) shows the correlations between saved and estimated results for the second phase of this analysis

The first stage of testing has shown an accuracy of almost 100% in the highest achieving self-detecting estimations for the full classifying tag with expected lower results for prior stages of the tagging process. On the other hand, as it is shown in the Table 2 and the number of distinct iterations is lower.

Table 2 first phase results for the six tested configurations

First Family results					
Case 01a		Case 02a		Case 03a	
Accuracy %	Unique tags of 123	Accuracy %	Unique tags of 114	Accuracy %	Unique tags of 132
36	46/123	33	45/114	22	36/132
53	65/123	48	60/114	35	51/132
59	72/123	55	69/114	41	60/132
99	120/123	99	111/114	94	124/132
Second Family results					
Case 01b		Case 02b		Case 03b	
Accuracy %	Unique tags of 148	Accuracy %	Unique tags of 142	Accuracy %	Unique tags of 157
34	47/148	31	41/142	21	34/157
52	74/148	42	60/142	35	59/157
66	104/148	54	84/142	49	92/157
100	148/148	95	133/142	95	151/157

For the second stage, the first phase which was within the saved data base of tags got an average 98% accuracy with the lowest accuracy of 93%. the time consumed was approximate average from 5 to 6 minutes for each configuration to get its results against double this time when it comes to the same way of running each building on its own then summing the results to get the total performance calculation. It is important to highlight that the initial runs of the total configuration as whole did not take more than 30 seconds for each. But the importance of this is discussed in the following section of this paper. While the same time were consumed during the second phase, the accuracy dropped to be around 86% with a low of 71% of the original model saved solar radiation results. Also, this phase failed to look up 8 out of the aimed 100 configurations. The comparison between predicted and estimated results is shown in Figure 7.

Discussion

During testing the study was faced with multiple challenges to be conducted. Some of these were inherited by the limitations of the used tools and other was caused by the limited capabilities of the used computational facilities to handle such a large amount of data and environmental performance simulation. This limited the inputs of the shown case study in this paper to the numbers afore mentioned which still compelling when compared to common practices of such analysis. The framework discussed in this paper shown a way of generating urban models along with parallel tagging each building in these models based on its geometrical features such as orientation, location, height relations with surrounding neighbouring buildings, exposure to urban voids, etc. these text tags is utilised in creating a database for classifying each building solar radiation performance within the urban configuration. Although this case study is rectangular boundary site, the framework has been tested with different boundary outlines. This shows the need to optimize the used algorithm from just depending on Visual Programming Language (VPL) hosted by Grasshopper along with some modelling and iterative

functions with python programming language (Python Software Foundation., 2001) within grasshopper interface. The algorithm optimization with python coded components included timing components, the component responsible of iterating the different model's inputs and some other functions needed to handle the model and its data. The first stage of this classification testing was to examine the basic function of estimating performance for urban configuration based on clustered saved analysis results by classifying its geometrical features. Also, it aimed to have a better understanding of the extent of variables needed to achieve an acceptable rate of accuracy for this estimation technique. After that, the next phase used a loose identification way to enhance the time consumed in the process and to assess this approximation effect on the accuracy ratio. Testing this classification technique for solar radiation is a preliminary stage for a sequencing method that works on clustering between dependent performance aspect. The aim from this study was to investigate the potentiality of this classification technique to reduce the needed time to estimate the urban configuration performance with reasonable performance. Although the framework so far has shown it does not reduce the time consumed to analyse the urban configuration as one entity, this technique proved a significant time reduction when it came to estimate the urban solar radiation performance on individual buildings' classification and similarity detection. The results accuracy, on the other hand, did achieve an acceptable ratio with the estimation within the retrieved database, yet it still needs more development of to enhance the results for the case of estimating new urban geometries not saved in the retrieved database.

Table 3 the selected showcase text tags and its solar radiation results. Generated tags (A1 and B1) and the detected nearest tags (A2 and B2).

Case number	Text Tag detail	Solar radiation kWh/sqm
A1	SE_notexposed_SE_UVNX_5+_NC_2146_7 5.0_LoNWLoLoNE_(0*H,0*-,3*L)	9.55
A2	SE_notexposed_SE_UVNX_5+_C_2996_60. 0_LoWLoNWLoN_(0*H,0*-,3*L)	11.97
B1	SE_notexposed_W_UVX_- =4_NC_1173_7.5_HoEHoSEHoSHoSHoN_(5*H,0*-,0*L)	1.44
B2	SE_notexposed_W_UVX_- =4_NC_2030_7.5_HoEHoSHoSWHoWHoN_(5*H,0*-,0*L)	2.40

Such development seems feasible with the current time reduction achieved in both cases. It's also imperative to highlight that the testing iterations were selected randomly. From Figure 7, the estimation does follow the pattern of the saved results while it still does the same for the second phase but with larger distances on some iterations which affirms the fact that there is a need to revisit the detection algorithm. The detection development is ongoing to differentiate each geometrical variable and give each a relative importance weight based on its role in predicting the accurate result. The detection

technique is aimed to have a numerical based system for all the geometrical variable used in the tags which will reduce the confusion between different types of data (letters and numbers) used now in the tags. an example of that is shown in Table 3. It shows the low effect of numbers used to tag areas and heights against letters used to different features of the tagging text. Figure 8 shows a sample show case for the detection process results. The selected buildings and tags (A1 and B1) are from the optimal achieving examples in this urban configuration. Solar radiation simulation was done specifically to extract the data shown in Table 3 for the new generated buildings results and to show the deviation between the actual results in comparison with the estimated results but in individual building scale. This comparison will take place in the detection development process. This brief showcase shows how the surrounding heights and its orientation to

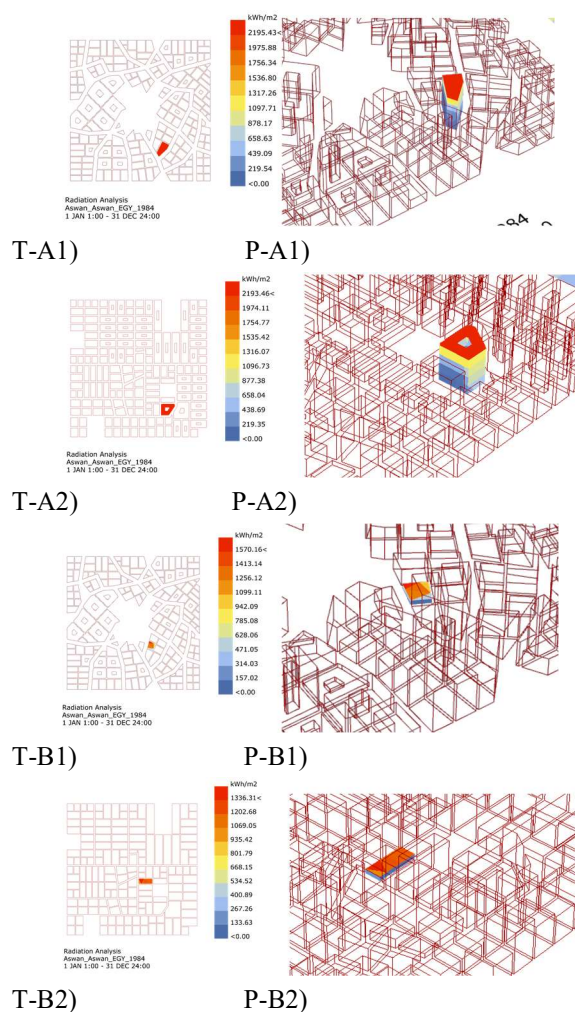


Figure 8 shows a selected showcase for the detection process results. Two buildings were selected from the same configuration (A1 and B1) and the detected equivalent buildings that has the nearest tags is shown as A2 and B2. Each case has a top view (T) and perspective view (P)

the tested building has a high impact in the detection process. This will be preserved and enhanced in future versions.

Conclusion

The literature review showed the continuous development of addressing urban complexity through computational modelling. Environmental performance analysis for urban scale has been under examination with some various approaches too. This paper shown a novel way of breaking up urban complexity through addressing geometrical features of the consisting buildings within the urban configuration though a parametric classifying model ready for multiple uses with multiple inputs. This framework provides a technique to link urban geometry to its performance and use the classification in estimating its solar radiation performance. This was conducted for newly designed neighbourhoods was a plan for application on existing neighbourhoods. This was done by breaking down urban configurations into its individual building's geometrical features to find an efficient way to work urban scale performance simulation. The first phase of testing this technique shown the potentiality of reaching for an acceptable accuracy with a minimal time reduction. The next phase took time into consideration by using less tightened way of finding out similarities to use for the performance estimation. This technique managed to achieve a considerable time reduction with acceptable accuracy levels which is being enhanced to reach for a consistent reliable level.

Future work

Currently the detection process is getting enhanced to allow more specific control on what to prioritise of the tag components. Also, the new process included re-running the unique tags and include it in the saved database, so it can be used to enhance the upcoming runs. Another test for broken text tags detection is ongoing to detect the relative importance of each factor. The next stage of this research is to add phase of artificial intelligence application section to this framework. This AI application well help to reach an optimal solution for the desired testing performance aspect. The scope of the research also includes having a holistic analysis for urban performance which means this process will be repeated in sequence for lighting availability to have a better insight of the optimal solution performance in different aspects.

References

- Alexander, C. (1979). *The timeless way of building* (Vol. 1). New York: Oxford University Press.
- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: towns, buildings, construction* (Vol. 2). Oxford University Press.
- Bassett, T., Lannon, S. C., Waldron, D., & Jones, P. J. (2012). Calculating the solar potential of the urban fabric with SketchUp and HTB2. *Solar Building Skins*, . Bressanone, Italy.
- Chatzivasileiadi, aikaterini, M. Hosney Lila, A., Lannon, S., & Jabi, W. (2018). The Effect of Reducing Geometry Complexity on Energy Simulation Results. *36th Annual Education and Research in Computer Aided Architectural Design in Europe (ECAADe)*. Retrieved from

file:///C:/Users/anas_/Downloads/AS6751401740902451537977369923_content_1.pdf

- Cichocka, J. M., Migalska, A., Browne, W. N., & Rodriguez, E. (2017). *SILVEREYE – The Implementation of Particle Swarm Optimization Algorithm in a Design Optimization Tool*. https://doi.org/10.1007/978-981-10-5197-5_9
- Duarte, J. P., & Beirão, J. (2011). Towards a Methodology for Flexible Urban Design: Designing with Urban Patterns and Shape Grammars. *Environment and Planning B: Planning and Design*, 38(5), 879–902. <https://doi.org/10.1068/b37026>
- Harding, J. (2017). Biomorpher. Retrieved from <https://github.com/johnharding/Biomorpher/blob/master/README.md>
- Hillier, B. (2007). *Space is the machine: a configurational theory of architecture*. Space Syntax.
- Hosney Lila, A. M., & Lannon, S. (2017). A parametric sensitivity analysis of the impact of built environment geometrical variables on building energy consumption. *PLEA*. Retrieved from <https://orca.cf.ac.uk/102918/1/Lila PLEA 2017.pdf>
- Kepczynska-Walczak, A., & Pietrzak, A. (2017). An Experimental Methodology for Urban Morphology Analysis. *International Conference on Computer-Aided Architectural Design Futures CAADFutures 2017: Computer-Aided Architectural Design. Future Trajectories*, 3–22. https://doi.org/10.1007/978-981-10-5197-5_1
- Koenig, R., Miao, Y., Knecht, K., Buš, P., & Mei-Chih, C. (2017). *Interactive Urban Synthesis*. https://doi.org/10.1007/978-981-10-5197-5_2
- Koenig, R., Treyer, L., Schmitt, G., & Zurich, E. (2013). Graphical Smalltalk with My Optimization System for Urban Planning Tasks. *Generation, Exploration and Optimisation - Volume 2 - Computation and Performance - ECAADe 31*, 195–203. Retrieved from <http://www.ia.arch.ethz.ch/>
- König, R., Schmitt, G., Standfest, M., Chirkin, A., & Klein, B. (2017). Cognitive computing for urban planning. In *The virtual and the real in planning and urban design: perspectives, practices and applications* (pp. 93–111). Retrieved from <https://www.research-collection.ethz.ch/handle/20.500.11850/230314>
- M.Hosney Lila, A., Lannon, S., & Jabi, W. (2017). Holistic sensitivity analysis on urban geometry and its effect on building performance in hot arid zones. In J. ElSharkawy, Heba, Zahiri, Sahar and Clough (Ed.), *International Conference for Sustainable Design of the Built Environment (SDBE)* (pp. 193–204). Retrieved from <http://orca.cf.ac.uk/107789/>
- March, L., & Steadman, P. (1971). *The geometry of environment: an introduction to spatial organization in design*. Retrieved from <http://library.wur.nl/WebQuery/clc/355952>
- Martin, L., & March, L. (1972). *Urban Space and Structures. Cambridge Urban and Architectural Studies, No. 1*. Cambridge University Press.
- Mcneel, R. (2014). Grasshopper 3D. Retrieved from <http://www.grasshopper3d.com/>
- Python Software Foundation. (2001). Python. Retrieved January 19, 2019, from <https://www.python.org/about/legal/>
- Robinson, D. (2006). Urban morphology and indicators of radiation availability. *Solar Energy*, 80(12), 1643–1648.
- Robinson, D., Campbell, N., Gaiser, W., Kabel, K., Le-Mouel, A., Morel, N., ... Stone, A. (2007). SUNtool—A new modelling paradigm for simulating and optimising urban sustainability. *Solar Energy*, 81(9), 1196–1211.
- Rutten, D. (2013, March). Galapagos: On the Logic and Limitations of Generic Solvers. *Architectural Design*, 83(2), 132–135. <https://doi.org/10.1002/ad.1568>
- Sadeghipour, M. r, & Pak, M. (2013). Ladybug: a Parametric Environmental Plugin for Grasshopper To Help Designers Create an Environmentally-Conscious Design. *13th Conference of International Building Performance Simulation Association*, 3129–3135. Retrieved from http://www.ibpsa.org/proceedings/bs2013/p_2499.pdf
- Shekhawat, K., & Duarte, J. P. (2017). *Rectilinear Floor Plans*. https://doi.org/10.1007/978-981-10-5197-5_22
- Tsamis, A. (2017). *The Marching Shape*. https://doi.org/10.1007/978-981-10-5197-5_20
- University of Applied Arts Vienna Bollinger+Grohmann Engineers. (2014). octopus | Food4Rhino. Retrieved June 14, 2017, from <http://www.food4rhino.com/app/octopus>
- Vartholomaios, A. (2017). A parametric sensitivity analysis of the influence of urban form on domestic energy consumption for heating and cooling in a Mediterranean city. *Sustainable Cities and Society*, 28, 135–145. <https://doi.org/10.1016/J.SCS.2016.09.006>
- Wortmann, T. (2017). OPOSSUM Introducing and Evaluating a Model-based Optimization Tool for Grasshopper. In P. Janssen, P. Loh, A. Raonic, & M. A. Schnabel (Eds.), *Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA)* (pp. 283–293). Retrieved from www.food4rhino.com