SEARCHING GENERATIVE METHODS BASED ON BUILDING ENVIRONMENTS

BIAO LI¹ and RONG LI²
1. South East University, Nanjing, China, jz_stduio@126.com
2. Generative Architectural Design Co., Ltd, Jiangsu, China, lr19790114@126.com

Abstract. Architectural contexts present both resources and restrictions from local spatial environment, such as factors of sunlight, landscape, line of sight, terrain and etc. They play important roles in architectural design. Following the traditional methods, it takes the architects lots of time to analyse the relative relationships. This paper takes terrain and sunlight into consideration to illustrate how to employ generative methods at the early stage of architectural design.

Keywords. Terrain context; sunlight; multi-agent; genetic algorithm, evolve.

Applying tremendous amount of architectural elements to the set of agents and materializing their characteristics to dynamic motion, multi-agent generative models employed in architecture require architects to survey kinds of architectural rules and elements as mutually cooperative agent systems. Moreover, the approach should be translated from angle of architects-oriented to angle of the architecture-oriented. By means of theoretical investigation and case studies of architectural environmental contexts for harmonizing future environments of new architectural schemes, any prototypes and parameters for computer programming will be extracted, synthesized and evolved via Simple Genetic Algorithms to both enhancing efficiency of architectural design and promoting people’s environmental awareness. Any tools developed in our toolkits are specific to the techniques for architectural design during the early stages, and the two factors, terrain and solar, are selected as examples in this paper to demonstrate the process.
1. Generating terrain based on a picture

All terrain information, such as properties of altitude, geometry, water, and spatial dimensions of lands, must be set as initial conditions for evolving of computer program. Many tools developed could be supportive in an alternative way, for example, GIS can support the terrain generation. However, this kind of software developed by other computer languages, such as C++, can’t be packaged into our Java based developing generative tools. More importantly, the files input from GIS contain too much information than we need, and can’t be employed to loop the evolutionary process of Genetic Algorithm. A simple but efficient new algorithm should be applied for generating different kinds of site terrain.

Some computer languages and file formats, such as Extensible Markup Language (XML) and file type of “*.dxf”, are competent for implementing datum of a predefined site as simple as in figure 1-left for example, all dots’ coordinates of the “key turn” in the site can be input one by one or predefined by XML. However, while dealing with the site of the future real estate projects, it is previously unpredictable whether the site will be as simple as figure 1-left or complicated as that of figure 1-right, and it is a tedious job for users to input datum of a complicated site even it’s only a description of 2-dimensions, let alone any other 3-dimensional information of altitude (coordinates of z added). One possibility based on pixel analysis which presented as follows may solve the problem of spatial data input and editing, meanwhile, the output keeps different kinds of properties, such as, land, grass, river or mountains and etc.

![Figure 1. Simple and complicated datum for sites and their environments](image-url)
The first step of extracting the datum of a site is to translate the surrounding spatial context to a picture filled with series of pure colours presented in figure 2 as an example. Colours can be parameterized according to their content of Red, Green and Blue (RGB). On one hand, each part of architectural environmental context maintains characteristic of “pure colour”; but on the other hand, it keeps any other specific information, for example, the green colour denotes spaces of green land, which whose “G” parameter is 255, but meanwhile, the parameter of “R” and “B” denote its altitude. As presented parameters of “G” in figure 2 on top-right, values of a, b, c and d are 255, but their “R” and “B” are all different, which the values of “R” and “B” are mapped for their altitudinal datum.

Employing the technique of pixel-based analysis, the whole datum of the 3D terrain in figure 3 will be generated from the above picture. Different resolution of meshes can be generated according to defined parameters. One noteworthy point is that the generated achievement of the site is divided according to predefined spatial properties, such as blue for lake, green for grass and etc. Based on the datum of architectural terrain, computational geometry and genetic algorithms, the optimized spatial locations of buildings will be evalu-
ated according to the environmental factors such as best scenic spots or region of water. The genetic processes include space data coding, individual adaptive evaluation, selection and alternative steps.

2. Evolution based on environments’ consideration

2.1. TERRAIN CONSIDERATION

![Figure 3. One generated terrain](image)

![Figure 4. Different ratios of priority for the grey boxes](image)
Benefiting from the generated terrain presented before, different volume of 3D space with different height of z-dimension can be defined above the building site (the above volume of red colour in Figure 3). One result is showed in figure 4, all the small grey boxes can be sorted by parameter named priorities’ ratios (PRI) where the positions were iterated and calculated by the amount of “water pixels” the grey boxes can “see”. (See figure 4 above: 5% of the best positions occupied by grey boxes, and 30% of that presents in below case).

Taking three red boxes which will be set as the main structure of the building later as an example, they are generated at random positions in figure 4, and then the three red boxes as agents will move or rotate automatically to try to intersect more grid grey boxes to get higher “scores” of better views, meanwhile, all the red boxes cannot be intersected by each other (see figure 5). The process makes frameworks of which the approximate locations for buildings in 3D spaces and all the buildings have got resources for best views from the local terrain context. Any other restrictions of environmental factors, such as viewpoints of 3D objects, can be set as goals for the process of program evolution. The dynamic evolving process applies all red boxes with biotic characteristic, and it takes several seconds for the three boxes to get balanceable locations where every box can get their best fitness for goals predefined previously.

![Figure 5. Red boxes search for best locations](image)

Originating from a real estate project, the tool added any other building restrictions of environmental contexts, such as simple calculation of sunlight (Java solar package presented in section of 2.2), systematizing package for architectural functions of topological relationship developed before and also tested in diversiform sites. Moreover, two other small tools were developed for the real project. The first one employing an algorithm of 3D convex was for enveloping the three boxes (see figure 6); and the second one was for the styles of the main structure of red boxes, any patterns such as Voronoi Diagram, Delaunay
Triangulation were tried, but finally an algorithm for plane split was adopted. The detailed introductions for the two programs omit here, the achievement of the real project presents in figure 7.

![Figure 6. Architectural envelope of convex.](image)

![Figure 7. Perspective for the real estate project](image)

2.2. SOLAR CONSIDERATION

Solar system involving in any knowledge of astronomy needs to be considered as a climatic condition of architectural environmental context, and it plays an important role in architectural design. Some software can be found for solar considerations in Architecture, such as Ecotect or even AutoCAD. However, these kinds of software are normally developed for verifying an accomplished layout of architecture, and they can’t tell architectural agents how to evolve for getting a higher genetic fitness. During the evolution of Genetic Algorithm, computer program needs to get tens of achievements per second for processes of selection, crossover, mutation and etc., which Ecotect and AutoCAD are
also incompetent. Simpler but more efficient java package was developed into our group.

In the design of apartment buildings, some restrictions need to be considered, such as insolation duration, height of buildings and design values of cantilever of balcony, the developed solar package combines sunshine calculating under various conditions and integrating it into a digital tool of Java program. During process of Genetic Algorithm, it is competent for verifying and contrasting fitness of next generation with evolution goal function predefined. The achievement which shows different colours in various 3D positions is presented in figure 8.

The Java solar package involving complex geometrical algorithms can be employed in different locations of the whole world. All the parameters can be changed and calculated easily during the running of a program, such as duration of sunlight needed for local houses in one day. All the current algorithms are perfect for verifying results of design. However, if applying the technique to Genetic Algorithm, which need to set solar parameters for condition of individual seeding selection, goal functions for evolutions and iteration system for more than ten generations per second, the current technology seems incompetent to its ambition.
Besides improving the efficiency of current algorithms, our Java package set *Adjacent-Searching-System* for detecting neighbours need to be considered during solar calculation of one building. *Voronoi Diagram* set a good algorithm for both defining and finding adjacent units. Let’s take a look at the figure 9 and imagine that a building site locates somewhere in the Northern Hemisphere (sunlight comes from south). Take the agent (or buildings) in the tessellations as considering objects, and focus on finding the agents which “r” will cast shadow on them. All tessellations marked “o” those who locate to the south of “r” needn’t to be considered.

![Figure 9. Adjacent-searching methods](image)

Two rules are set for finding tessellations named “g” that “r” which will cast shadow on: one is that we may set the depth of program should search, for example, if the deep set 1, and then 3 of “g1” should be considered; if the deep is 3, and then three of “g1”, 5 of “g2” and 9 of “g3” will be taken into account. All the “g” will be detected to confirm whether “r” will cast shadow, and all the “g” covered by shadow from “r” will be stored in a list. The second rule: if “r” cast shadow on one “g3”, and then all the adjacent tessellations of “g3” will be taken into account for detecting whether or not they get shadow from “r”, and the operation will iterate within tessellations named “b”. By such means, a mapping between each tessellation and others who may cast shadow to it will be established. Similarly, a mapping of all tessellations which receive shadow from others can be established.

The algorithms presented above show simple principle employed in
developed Java package, and the whole agents are just represented by points. For the real architectural projects, the points must be transformed or meshed with real 3D geometries (see Figure. 8). Taking residents or apartment planning as an example, many countries set times of duration for each apartment that should get solar light during worst weather in winter days. Employing the java solar package and Genetic Algorithm, many achievements of layout will be obtained within less than one minute. Two achievements predefined with different numbers of building floor and floor area ratios generated are presented in figure 10.

![Image](image.jpg)

*Figure 10. Solar package for apartment design.*

2.3 COMBINATION OF PACKAGES FOR ARCHITECTURE

Architectural generative design, which integrates multidisciplinary collaborative thinking in unconventional ways, still lacks both basic theoretical guidance and systematic methodologies. Our Java packages based on contexts of terrain and solar are powerful for early architectural design, any other Java package involving architectural environmental contexts are being developed
to implement the solution for architectural design. These Java packages include not only the geometry algorithms classes for implementing the basic shapes of architectural types, but also the utility classes based on the models for real estate projects. In real application, there is no rigid way for employing these packages, and the generative approaches for architectural design are extendable and flexible, which can be implemented as effective models for dynamic applications that run on real time requirements. All the requirements on performance and availability of detail applications have pushed the research on extracting prototypes in Architecture. The widespread availability of the digital techniques brings about great flexibility and facility to architects, however, it also brings about a series of new problems and challenges to the traditional architectural methodology.

3 Summary

Building environments contain many factors that need to be derived from real projects, and each prototype must be developed from basic architectural principles. Generative design based on architectural rules and controlled by programming logic embodies an operable and flexible methodology for all architects. However, traditional architectural methodology in architecture is increasingly threatened by increasing globalization of digital techniques, such as buildings showing strange shape are attractive to clients and etc. rather than improving quality of architectural spaces and consideration of engineering budget.

Our generative tool provides a design system for the early stages in architectural design. Employed the techniques embodying humanistic characteristics simultaneity, concept of architectural design methodology can be redefined. Digital techniques supply simulative tools to create unparalleled output, which is the ‘generator’ for new environmental contexts, In accordance to its development prosperity, generative methods applied in the artificial world will endow architecture with the aesthetics jocundity through the approach of Artificial Intelligence and Artificial Life System.