POPULATION BASED PROCEDURAL ARTIFICIAL CITY GENERATION USING BETA DISTRIBUTION

Baha ŞEN, Abdullah ÇAVUŞOĞLU, Haldun GÖKTAŞ and Nesrin AYDIN

Karabük University, Engineering Faculty, Computer Engineering Department 78050, Karabük TURKEY.
Yıldırım Beyazıt University, Engineering Faculty, Ulus Altındağ / Ankara TURKEY.
baha.sen@karabuk.edu.tr, abduallah.cavusoglu@gmail.com, haldungoktas@gmail.com, nesrinaydin@karabuk.edu.tr

Abstract- Artificial city generation on computer graphics platforms introduce several problems from the point of view of the application programmer. Especially in the cases where the product is aimed for virtual reality applications, this becomes more important since the target is achieving city layouts akin to the real cities. The same is valid for the civil engineers where the layouts of the blocks/cities are determined in advance of the construction. An important parameter for artificial cities is the determination of the population distribution over the cities which in turn affect the overall appearance of the city or the blocks forming it. In this study, the Beta distribution has been used to disperse artificial city populations over the city blocks to generate cities that do not look to regular. The system uses HTF based maps and the produced 3D cities are quite realistic when compared to the similar products.

Keywords: Beta distribution, Procedural modeling cities, Artificial city generation, Genetic algorithm.

1. INTRODUCTION

Producing large city databases on the artificial platforms requires a series of complex operations. This is because, the city formations contain several different visual objects and also, placement of these objects over the ground (i.e. blocks) hierarchically poses a series of complex functions and procedures. In addition to make an artificial city more realistic, not only we need to consider a natural looking overview but also other considerations are required; such as the placement of schools, hospitals, parking areas, parks etc. which are the factors closely related to the city population. Also, other problems such as incorporating these objects into the varying geographical conditions arise.

The overview of the current works shows that the simulation techniques are often employed for modeling the natural phenomenon. Some systems are focusing on the modeling of the erosion [1], wood particles [2], cloud modeling etc. [3]. In addition, rule based production mechanisms are also employed (esp. L-systems) to model the plant geometry [4]. L-systems are among the systems containing powerful and sophisticated tools to model the ecosystems [5, 6 ,7]. While some of the modeling platforms focus on the modeling of artificial cities. Especially determining the building objects within the city by means of the satellite images [8], and using them for the road layout formations for the artificial city and finally modeling the rest of the city by

B. Şen, A. Çavuşoğlu, H. Göktaş and N. Aydın

employing the L-systems [9]. One other problem –apart from the artificial city production– is the timing issue. Since most of the platforms mentioned above uses some rules and parameters to produce cities and following employs separate modeling software to obtain screenshots. There are however real time systems which allow the user to navigate within the artificial city (i.e. in 3D) with a selection of certain view point [10]. In some of the studies, the geographical maps are subdivided into smaller areas to be able to model artificial cities [11]. Various graphical displaying techniques are employed for visualizations [12, 13].

The main distinction between this study an the rest of the studies above is that, our system is totally artificial, in other words, none of the constituent part of our artificial cities are based on existing cities and therefore we do not need anything (i.e. road layouts, block/building placements etc.) from them. Secondly, this approach is unique since it is the only approach which employs population parameter to determine the size, appearance of the buildings (e.g. by classifying them in terms of possible resident populations) which in turn affects the overall appearance of the city. And the result is the city layouts which are not so regular/mechanical. The system uses the New York city model, works on the Height Tile Format (HTF), and the population parameter is dispersed over the city blocks using a certain set of pre-classified buildings in terms of the population (i.e. by taking into consideration the civil engineering rules). For the population dispersion the Beta distribution function, which helps us to non-uniform city appearance by determining the number for each type of city blocks, is employed – resulting either a much densely populated city or a much green city with parks etc. – and the buildings are placed on the blocks using genetic algorithms [14,15].

2. ARTIFICIAL CITY GENERATION USING THE BETA DISTRIBUTION

In this study the aim has been to produce artificial cities based on predefined production rules by employing procedural production techniques over the ever changing geographical conditions. Naturally this will lead to different city layouts every time. The main parameters for the system are namely: the total population size of the city, and the block size. The rest of the rules are embedded into the system –such as the determination of land size for the various establishments taken from the civil engineering rules– for the sake of simplicity. As mentioned above the New York City model which is based on rectangular square blocks is employed in the current system. The system tries to distribute the total city population over the city blocks. Therefore, for each block, according to its population, a set of buildings are selected among the pre-classified buildings. The area classes and their respective tags used for the determination of block requirements for each establishment are shown in Table 1. The beta distribution is used to determine the size of the population to be distributed over the blocks. In other words, the number for each type of blocks is determined by this function. Following, the relevant objects are formed and placed by the genetic algorithm for each block [14].

Table 1. Area class types and their tags

Area Class Types	Class Tag
Green Areas (Parks)	6
Health Establishments	5
Cultural Establishments	4
Social Establishments	3
Education Cites	2
Technical Cites	1
Administration (or management)	0

As the Figure 1 illustrates the Beta distribution is in fact a statistical distribution mechanism linked to gamma distribution. It has two parameters namely: α and β, and used as $\beta' = \beta - 1$ and $\alpha' = \alpha - 1$ [16]. The Beta distribution is generally employed at the Bayesian Analysis [17] to determine prior distributions in binominal sections.
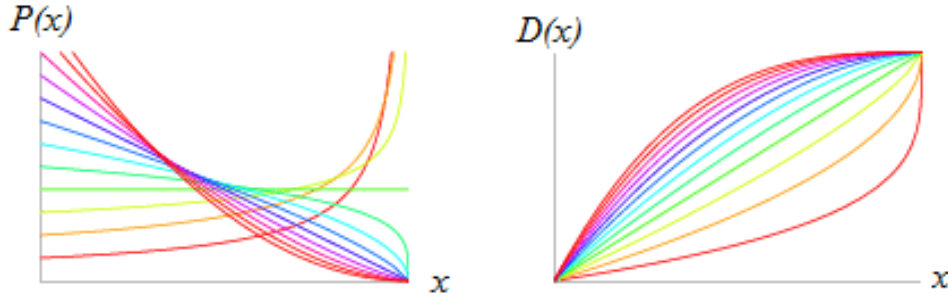
Figure 1. Distribution graphics for $\alpha = 1$ and β varying between 0.25 and 3.00 (17)

Here, the formed domain is between the [0, 1] and $P(x)$ represents the probability function while $D(x)$ represents the distribution graphics. The probability and distribution functions are represented by;

$$P(x) = \frac{(1-x)^{\beta-1} x^{\alpha-1}}{B(\alpha, \beta)} \tag{1}$$

$$= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} (1-x)^{B-1} x^{\alpha-1} \tag{2}$$

$$D(x) = \mathrm{I}(x; a, b) \tag{3}$$

Where $B(a,b)$ is the beta function, $\mathrm{I}(x; a, b)$ is the regularized beta function and $\alpha, \beta > 0$. To normalize this distribution,

B. Şen, A. Çavuşoğlu, H. Göktaş and N. Aydın

$$\int_0^{-1} P(x)dx = 1 \tag{4}$$

The characteristic function is represented by,

$$\phi(t) = \int \frac{X^{a-1}(1-x)^{b-1}}{\beta(a,b)} e^{-2\pi i x t} dx =_1 F_1(a; a+b; it) \tag{5}$$

Where the raw moments are calculated by,

$$\mu_\gamma' = \int_0^1 P(x)x^\gamma dx = \frac{\Gamma(\alpha+\beta)\Gamma(\alpha+r)}{\Gamma(\alpha+\beta+r)\Gamma(\alpha)} \tag{6}$$

While the central moments are calculated by,

$$\mu_\gamma = \left(-\frac{\alpha}{\alpha+\beta}\right)^\gamma {}_2F_1\left(-\gamma, \alpha; \alpha+\beta; \frac{\alpha+\beta}{\alpha}\right) \tag{7}$$

Where the term ${}_2F_1(a,b;c,x)$ is a hypergeometric function and the arithmetical mean, variance, skewness and kurtosis are expressed as:

$$\mu = \frac{\alpha}{\alpha+\beta} \tag{8}$$

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{9}$$

$$\gamma_1 = \frac{2(\beta-\alpha)\sqrt{1+\alpha+\beta}}{\sqrt{\alpha\beta}(2+\alpha+\beta} \tag{10}$$

$$\gamma_2 = \frac{6\left[\alpha^3 + \alpha^2(1-2\beta) + \beta^2(1+\beta) - 2\alpha\beta(2+\beta)\right]}{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)} \tag{11}$$

Therefore, the mode of variant distributed $\beta(\alpha,\beta)$ can be expressed as [18]:

$$\hat{\chi} = \frac{\alpha-1}{\alpha+\beta-2} \tag{12}$$

After the brief revisiting of the Beta distribution, we may proceed onto its application over the city population. The total amount of land required for the city establishment is calculated by using the block size and local civil engineering rules –which is used to determine what type of establishment requires how much land for an ideal city– [19]. The block size is represented by:

$$b_a = b_h * b_w \tag{13}$$

Where b_a is the block area (size), b_h represents the block height and b_w is used for the block width. The total area required for the city is calculated by:

$$\sum city\_area = \sum_{i=0}^{i=6} area_i \tag{14}$$

Where the $area_i$ represents the relevant area illustrated in Table 1 that is needed to be allocated for the required establishments. For each class of establishments the number of blocks that are needed to be allocated is determined by:

$$\sum blk\_Count_{prk} = \frac{\sum prk}{b_a}, ..., \sum blk\_Count_{adm} = \frac{\sum adm}{b_a} \tag{15}$$

Where *prk* stands for parks and *adm* stands for administration. Using these values, the total number of blocks is calculated by:

$$\sum blk\_Count = \sum blk\_Count_{prk} + ... + \sum blk\_Count_{adm} \tag{16}$$

Using the geographical information, we try to fit the city into a rectangular area (where appropriate). Therefore, the number of blocks to be fit into the rectangular diameters (i.e. the width, height of the whole land to be occupied by the city) is calculated by:

from $per_{xy} = \dfrac{b_w}{b_h}$ we get;

$$nobx = \sqrt{\frac{\sum blk\_Count}{per_{xy}}} , \quad noby = \frac{\sum blk\_Count}{nobx} \tag{17}$$

The calculated values of nob_x and nob_y which represent the number of blocks on the x axis and y axis of the block matrix respectively must be integer numbers and if needed they are rounded. In other words, we make sure that the total area where the whole city is placed is rectangular. The calculated total number of blocks (eq.15) and the number of blocks needed to be distributed among the field types (eq.16) are used to build a block list represented by a 2D matrix. In other words, the population size is used to determine approximate city land size in terms of blocks. By parsing on this list, the tag of each block will be marked. However at this point, the total number of blocks for each type (i.e. tag) needs to be determined. Following the calculation of total number of blocks, this is done by the $B(a,b)$ function. This also means that the distribution of various blocks type over the GIS map. To obtain a non-uniform appearance within the city blocks, Genetic Algorithm is employed.

Table 2. The distribution results of the $B(a,b)$ function for a=1,200 and b=2,700

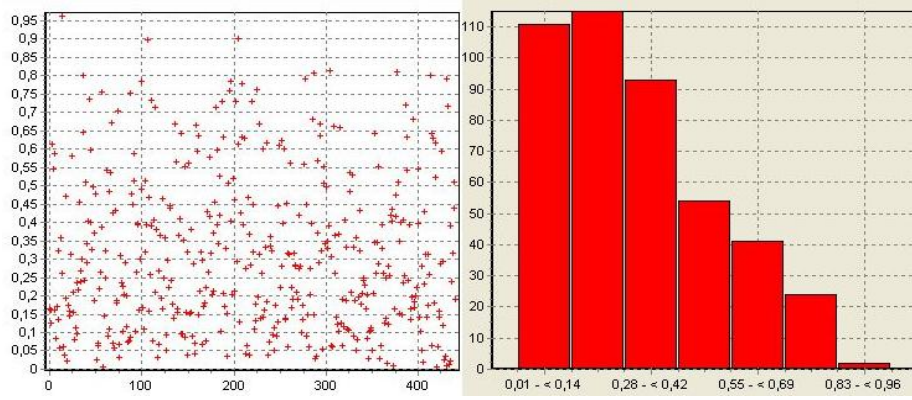| Block Types	T_{Green}	T_{Medic}	$T_{Cultural}$	T_{Social}	$T_{Educational}$	$T_{Technical}$	$T_{Management}$
Number of Blocks	3	24	41	54	93	115	111
Number of Values : 441 Number of Classes : 7							
a= 1,200 b= 2,700 γ_1=0,7097 μ=0,2640 σ=0,0443 γ_2=-0,3040							

B. Şen, A. Çavuşoğlu, H. Göktaş and N. Aydın

Figure 2. The graphical representation of the distribution illustrated in Table 2

As Figure 2 illustrates, due to the difference between the parameters (i.e. *a* and *b*), the difference between the first and last class is big. This means that, the population of the city center is dense when compared to the city outskirts. This is shown in Figure 3. Playing with these two parameters, (i.e. reducing the difference between them) the $B(a,b)$ function gives the results as in Table 3. In this graph, the distribution is more homogeneous when compared to the previous one. As Figure 5 shows, there are not many differences between the block types. This will mean that the city population will also be homogeneously distributed over the districts.
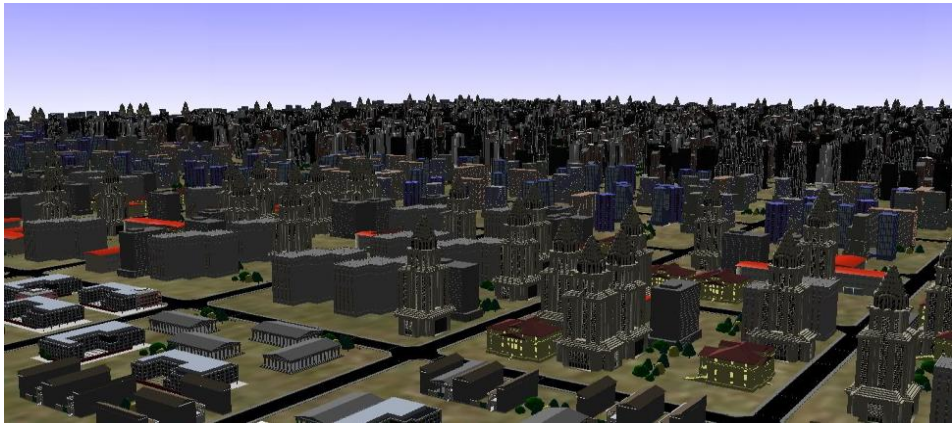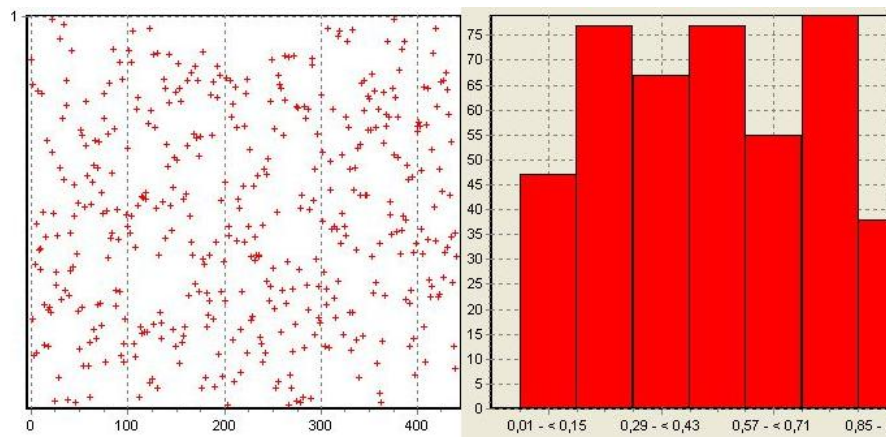

Figure 3. 3D View of the city produced in Table 2

Table 3. The distribution results of the $B(a,b)$ function for a=1,200 and b=1,250

Block Types	T_{Green}	T_{Medic}	$T_{Cultural}$	T_{Social}	$T_{Educational}$	$T_{Technical}$	$T_{Management}$
Number of Blocks	39	79	55	77	67	77	47
Number of Values : 441			Number of Classes : 7				
a= 1,200 b= 1,250			γ_1 =0,0260				
μ =0,4816 σ =0,0702			γ_2 =-1,1366				

Figure 4. The graphical representation of the distribution illustrated in Table 3

The corresponding 3D model of the city represented in Figure 3 is shown in Figure 5. Comparing Figure 5 with the Figure 3 clearly shows that the city center is not as denser and a more homogeneous population distribution is accomplished. By changing a and b parameters within the $B(a,b)$ function, various city layouts for different population densities may be obtained. This provides an automatic mechanism for the artificial city generation. The objects placed on the city blocks are predetermined and selected randomly from a pre-formed object class. The placement problem of the objects on the blocks is regarded as an optimization problem and solved by genetic algorithm.

Figure 5. 3D View of the city produced in Table 3

3. CONCLUSIONS

In this study, artificial cities have been produced, –using the population and block size of the city– by taking into consideration the population and block size parameters by means of applying the Beta distribution. The main target has been fast artificial city production for virtual environments using parameterized variables as well as establishing non-uniform appearance for the produced cities. At this point, the Beta distribution has been employed. It helps to produce varying cities –in terms of appearance and building placements– by using the number of blocks and relevant parameters (i.e. a and b). Since the block types are taken into consideration, the algorithm also contributes towards a meaningful distribution of the population over the city. This naturally results in cities of various densities in terms of population and building placements.

The distribution mechanism employed in the study, is not deterministic and therefore does not result in mechanical or regular appearance (i.e. which is the main disadvantage of such systems). Another advantage of the system is that the cities are produced on real geographical maps. Therefore, the mechanism may be employed in non-artificial applications with some modifications.

REFERENCES

1. F. K. Musgrave, C. E. Kolb, R. S. Mace, The Synthesis and Rendering of Eroded Fractal Terrains, SIGGRAPH 89 Proceedings,41-50,1990.

2. W. T. Reeves, R. Blau, Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems, Computer Graphics (SIGGRAPH 85 Proceedings), 19(3), 313-322, 1985.

3. K. Perlin, An Image Synthesizer, Computer Graphics (SIGGRAPH 85 Proceedings), 19(3), 287-296,1985.

4. P. Prusinkiewicz, A. Lindenmayer, The algorithmic beauty of plants, Springer-Verlag, New York,1990.

5. R. Mech, P. Prusinkiewicz , Visual Models of Plants Interacting with Their Environment, SIGGRAPH 96 Conference Proceedings, 397-410, 1996.

6. P. Prusinkiewicz, M. James, R. Mech , Synthetic Topiary, SIGGRAPH 94 Conference Proceedings, 351-358, 1994.

7. O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr, P. Prusinkiewicz, Realistic Modeling and Rendering of Plant Ecosystems, SIGGRAPH 98 Conference Proceedings, 275-286, 1998.

8. A. Gruen, E. Baltsavias, O. Henricsson , Automatic extraction of man-made objects from aerial and space images(ii), Proceedings of the Monte Verita Workshop,1997

9. Y. Parish, P. Muller, Procedural Modeling of Cities, ACM Proceedings of SIGGRAPH 2001, 301-308, 2001.

10. M. Wimmer, P. Wonka, F. Sillion, Point-based impostors for real-time visualization, EuroGraphics Workshop on Rendering, 2001.

11. J. Bittner, P. Wonka , M. Wimmer, Visibility preprocessing for urban scenes using line space subdivision, 9th Pacific Conference on Computer Graphics and Applications, Tokyo, Japan ,276–284 , 16-18 October, 2001.

12.N. Greene, M. Kass, G. Miller, Hierarchical z-buffer visibility, Proceedings of ACM SIGGRAPH 93, 231–240, 1993.

13.M. Levoy, P. M. Hanrahan , Light field rendering, Proceedings of ACM SIGGRAPH 96, Computer Graphics Proceedings,31–42,1996.

14.H. H. Göktaş, A. Çavuşoğlu, Şen B., Algorithmic 3D Virtual City Production Using Genetic Algorithms, The 5th International Symposium On Intelligent Manufacturing Systems (IMS2006), Sakarya, Turkey, May 29-31,2006.

15.A. Cavusoglu, H.H. Goktas , B. Sen , AUTOCITY : A System for Generating 3D Virtual Cities for Simulation Systems on GIS Map, AutoSoft - Intelligent Automation and Soft Computing, 15(1), USA, 29-39,2009.

16.W. H. Beyer, CRC Standard Mathematical Tables, FL: CRC Press, Boca Raton, 1987.

17.M. Evans, N. Hastings, Peacock B 2000 Beta Distribution, Wiley, New York, 2000.

18.Beta Distribution 2006, http://mathworld.wolfram.com/BetaDistribution.html, 2006.

19.İmar Kanunu Mevzuatı, Seçkin Yayınevi, Ankara,2004.