DESIGNING FRACTAL BUILDINGS USING ITERATIVE FUNCTION SYSTEMS

Berdiev Golib Rashidovich
PhD Student at Tashkent University of
Information Technologies named after Muhammad al-Khwarizmi
ORCID ID: 0000-0003-0075-5414

Djuraeva Madina Muxiddin qizi
Master Student at Tashkent University of
Information Technologies named after Muhammad al-Khwarizmi

Abstract

This article presents an approach to rapid analytical geometry and computer graphics used to create building projects using fractal forms. The system is fully integrated into the graphics processing unit and the results can be viewed in real time. Because the buildings are depicted as remote areas, high-quality modeling and image display or 3D computer graphics can be used to effectively see the results. Using the visual complexity of a class of fractals known as iterative function systems (IFS), it is possible to geometrically model complex fractal structures reminiscent of ornamental architectural styles such as Gothic and Baroque, with simpler rules than grammatical methods.

Keywords: Fractal, fractal geometry, design, building, architectural form, geometry, model, computer, modeling, IFS, generative systems.

Introduction

Many fractal images and surfaces feature complex structures that repeat on multiple scales. You can create and explore these structures using methods such as fractal-form construction formulas and iterative function systems (IFS). Familiar fractals, such as serpentine napkins or Menger sponges, have structures that can be easily deduced from the geometric rules of architecture. The Menger sponge looks like the characteristic cube shown in Figure 1. Gupka's IFS configuration requires recursive use of affinity substitution for point sets. However, subtle changes to the creation rules can lead to a variety of complex shapes. It is not difficult to see how the recursive nature of this deformed Menger sponge can generate many structures.

For most 3D fractals, you can estimate the distance from a specific point to the surface. This allows the separation of accelerated beam tracking algorithms and remote computations. Using fractals to model arbitrary structures requires solving difficult inverse problems. The collage theorem describes a general approach to constructing an inverse problem by specifying general conditions for the existence of an inverse problem, for example finding iterative processes leading to a given shape [1].

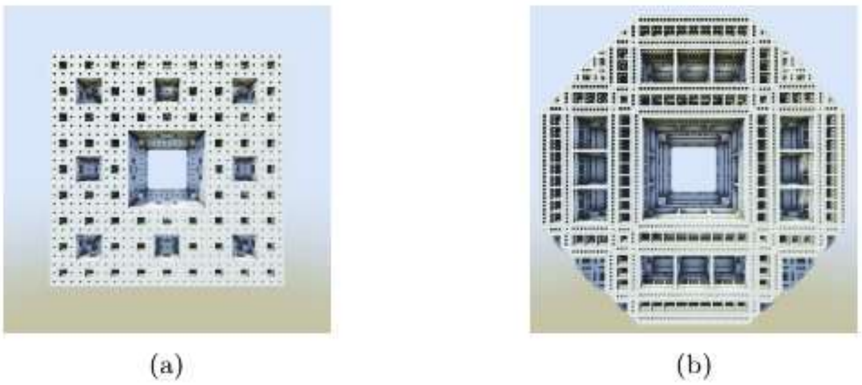

Figure 1. (a) with Menger sponge and (b) with IFS c = (0.93, 0.93, 0.33)

In 2D this led to fractal image compression algorithms, but in 3D it was difficult to find common solutions. The use of fractals in the modeling of 3D buildings This approach is to convert the shape of the selected building to another shape bounded by the IFS fractal shape. In this study, periodic mapping with local symmetry is represented by a mod function. The resulting shape can be seen by observing the rays or by rasterizing a triangular grid formed by obtaining an iso-surface. This approach allows the user to select the building parameters and immediately see the resulting structure.

Main Part

These modeling techniques allow you to create graphical content (such as grids or textures) using automated or interactive algorithms. The scope of construction methods includes cities [2], the area of individual buildings, and the interior layout of buildings [3].

Most previous studies on buildings have focused on grammar-based approaches [4,5]. What they have in common is the output of trace shapes to a volume mass model that defines the overall shape of the building. An alternate set of rules then defines how the model is divided into floors and how each floor is divided to create the facade of the building. The renovation process ends with simple basic elements such as bricks, windows and doors. In comparison, this method uses simple arithmetic equations to isolate the front surface, and the final basis is visual fractal complexity using regions. The modeling method for this structure is similar to the shape modeling process described by McGraw and Herring, with several important differences [6]. This document uses a specific class of fractal IFS that is very suitable for creating building designs. The periodic, symmetrical structures of many buildings, along with simple boundary dimensions that can be represented by combinations of geometric circles, simplify the process of placing fractals on surfaces. In contrast, McGraw and Herring require the user to interactively determine the spline-based deflection from the fractal field to the net by placing individual peaks. An iterative function system is to obtain a set of entry points and apply an affine transform to them to form a fractal from the set of points [7]. The first use of IFS in computer graphics suggests that relatively small sets of variations can access natural objects such as leaves and ferns, as well as reproduce classic fractals such as Cantor sets and Serpin Napkin [8].

The IFS fractal described in Algorithm 1 was used by Knighty to develop a distance estimate for the 3D fractal Serpin tetrahedron and the Menger gupus [9]. View the processes as a repetitive sequence of the same scaling operations around the center point given by the sum (rows 5-10), the cycle (rows 3, 11), and x_c, y_c, z_c (rows 12-14). possible. This algorithm generates the Menger sponge for $x_c=y_c=z_c=1$, $s=3$ and $R_1=R_2=I$ For other parameter values, a rich set of properties of organic and synthetic forms appears, depending on the parameter value. As shown in Figure 2, the surface becomes sparse and breaks at s> 3.

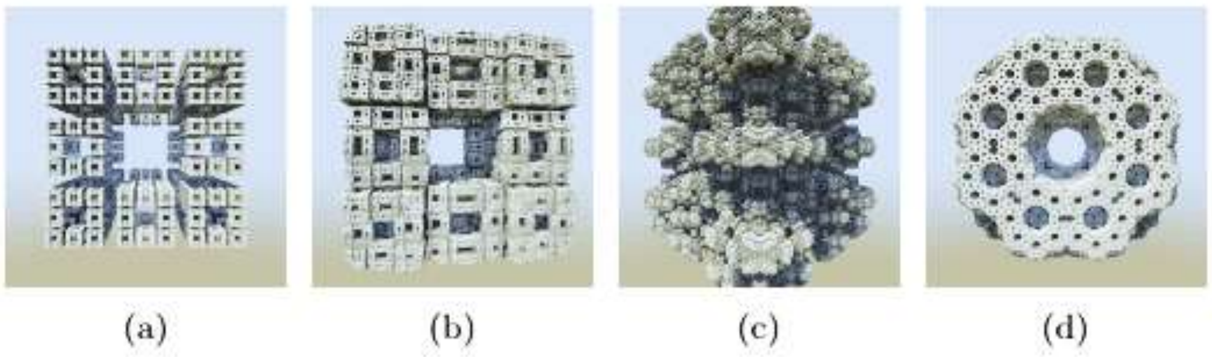

Figure 1. IFS (a) $s = 3.5$, (b) $R_1 = R_y(\pi/25)$, (c) $R_1 = R_y(\pi/4)$ and (d) $R_2 = R_y(\pi/8)$

For R_1, the surface is less regular at small angles of rotation and resembles an old brittle structure. R_2 matrices can transform linear structures into structures with properties such as polygons and stars. As with most fractal systems, a good way to understand shape boundaries is to test and study their parametric fields. This is supported by a fast graphics processor and a user interface where parameters can be defined. Hart et al. introduced the idea of setting distance limits on fractal surfaces to accelerate ray observation [10]. Knowing that the distance to the surface is greater than or equal to d, it is safe to observe the light at distance d when repeatedly finding the intersection of the beam and the surface. Stop when d falls below a certain limit.

The process of displaying images in real time using this modeling technique was introduced by Inigo Quilez [11].

Algorithm 1. Algorithm for estimating the distance from a point *(x, y, z)* to an IFS fractal. The scale center parameters x_c, y_c, z_c and scale factor s are scalar values, and R_1 and R_2 are rotation matrices. In our experiments, we use conditional limits of maximum iterations $M = 6$ and $b = 1.5$.

```
function $d_{IFS}(x, y, z, x_c, y_c, z_c, s)$
for i = 0 to $M$ do
$[x\ y\ z]^T = R_1[x\ y\ z]^T$
$x = |x|, y = |y|, z = |z|$
if $x - y < 0$ then swap$(x,y)$
end if
if $x - z < 0$ then swap$(x,z)$
end if
if $y - z < 0$ then swap$(y,z)$
end if
$[x\ y\ z]^T = R_2[x\ y\ z]^T$
$x=s(x-x_c) + x_c$ , $y = s(y - y_c) + y_c$ , $z = sz$
if $z < z_c(s - 1)/2$ then $z = z - z_c(s - 1)$
end if
$r = x^2 + y^2 + z^2$
if $r > b$ then break
end if
end for
return $(r^{1/2} - 2)s^{-i}$
end function
```

In addition, the distance function expression makes it easy to perform constructive rigid geometry (CRG) operations on figures.

Methods

Our building creation system is integrated into real-time raycasting rendering and buildings are represented by size as a function of distance. The light is observed in the glsl fragment shader until it crosses the building, then the light is calculated. Project construction models are made from simple shapes, such as 3D boxes. The main box has a distance function

$$\max(|x| - h_x, |y| - h_y, |z| - h_z) \qquad (1)$$

where *x, y, z* are the coordinates of the point, and h_x, h_y, h_z are the half-widths of the box along the *x, y, z* axes. We allow users to choose from several construction models created from constructive rigid geometry (CRG) operations in boxes. The operations of union (∪) and intersection (∩) are given by the formula.

$$\cup(A, B) = \min(d_A, d_B) \qquad (2)$$

$$\cap(A, B) = \max(d_A, d_B) \qquad (3)$$

Here *A, B* are the shapes and d_A, d_B are the distances to *A* and *B*. The mass models used in the proposed system consist of an inner and an outer shell. The outer shell is the outermost part of the building. Once the IFS has identified the infinite plates of the fractal, the calculation of the intersection with this outer shell limits the building to a limited area. The inner shell represents the windows and the outer walls. Calculating the fractal and inner shell joints prevents the user from seeing the inside of the fractal (Figure 3).

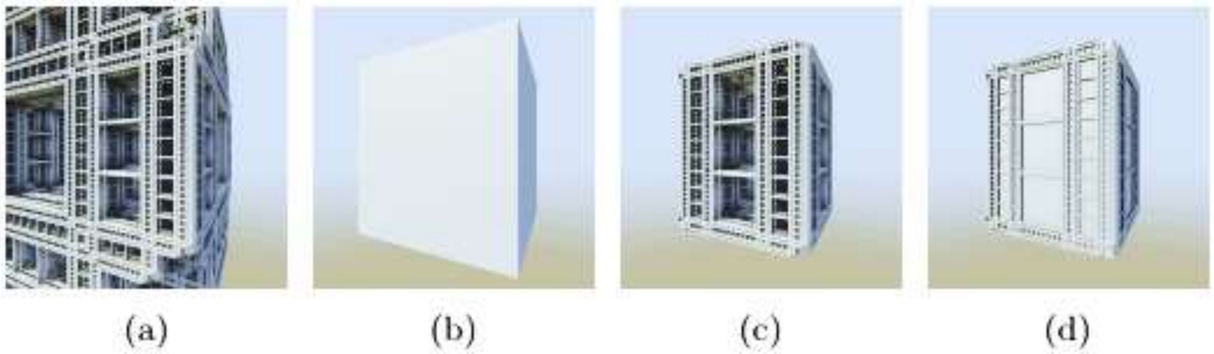

Figure 3. IFS detail (a), project model outer shell (b), IFS and outer shell intersection ∩ (IFS, Outer) (c), previous result merging with inner shell ∪ (Internal, ∩ (IFS, External))) (d)

Building technology is based on changing the distance and coordinates of the IFS system described in Algorithm 1. If the distance to the fractal surface is given as $d_{IFS}(x, y, z)$, then the distance to the simulated building is $g\ (d_{IFS}(f_1\ (x, z), f_2\ (y), f_3\ (x,)z)))$, where *y* is the vertical axis of the structure. Distance conversion *g ()* is a series of constructive rigid geometry (CRG) operations that define an architectural mass model. The functions $f_1\ (x, z)$, $f_2\ (y)$, and $f_3\ (x, z)$ are coordinate changes that align the IFS areas to the building surface.
Given by the change in vertical coordinates

$$f_2(y) = s_y \bmod (y - y_o, h)/h \qquad (4)$$

where mod operation results in periodic repetition of the fractal part that forms the floors or surfaces of the building. The parameters y_o, *h* define the vertical slab in IFS and how s_y defines the slab in the building.

WEB OF SCIENTIST: INTERNATIONAL SCIENTIFIC RESEARCH JOURNAL
ISSN: 2776-0979, Volume 3, Issue 5, May., 2022

The area of the building at each level is also periodic, but should allow for symmetry and repetition of areas. These possibilities are realized by a more complex function using the mod-functions given in Algorithm 2.

Algorithm 2. Algorithm for calculating the modified coordinates of a building, $x_f=f_1(x, z)$, $z_f=f_3(x, y)$ and facade identifiers id_x, id_z. The coordinates being changed are x, z; w_x, w_z is the period of pattern repetition on the x and z faces of the building, and $\mathbf{r_x}=(r_{x,1}, r_{x,2}, r_{x,3})$ checks the width identifiers inside the pattern when constructing x-surfaces, $\mathbf{r_z}$ works the same way.

```
function $f_{xz}(x, z, w_x, w_z, \mathbf{r_x}, \mathbf{r_z})$
$x_f = |2 \bmod (x, w_x)/w_x - 1/2|$
$z_f = |2 \bmod (z, w_z)/w_z - 1/2|$
$id_x = 0, id_z = 0$
for $i = 1$ to $3$ do
$x_f = |x_f - r_{x,i}| - r_{x,i}$
$z_f = |z_f - r_{z,i}| - r_{z,i}$
if $x_f > 0$ then $id_x = id_x + 1$
end if
if $z_f > 0$ then $id_z = id_z + 1$
end if
$x_f = |x_f|, z_f = |z_f|$
end for
return $x_f, z_f, id_x, id_z$
end function
```

The arbitrary change of x, z coordinates supported by the system is to transform the curve into polar coordinates to shape the building. Transformations that produce cylindrical and curved structures are:

$$x' = s_\theta(\arctan(z, x) + c_\theta) \qquad (5)$$

$$z' = s_r\left(\sqrt{x^2 + (z - z_0)^2} + c_r\right), \qquad (6)$$

where c_θ is the angle of rotation, s_θ determines how much the arc affects the building, and c_r, s_r controls the inside and outside radii of the building. Examples of curved building models and façade identifiers are shown in Figure 6.

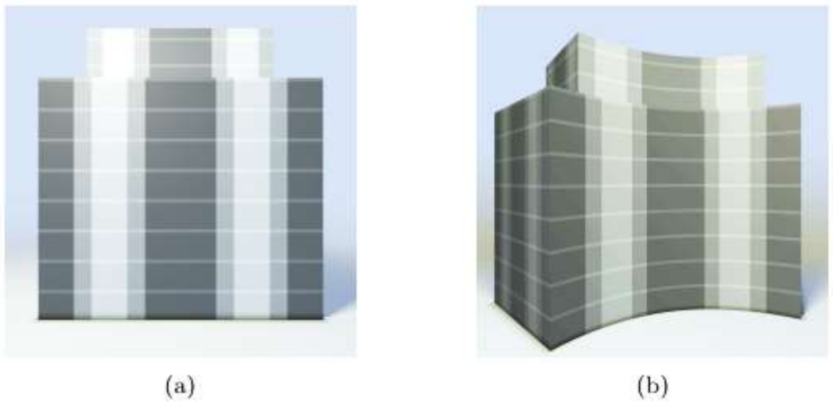

Figure 6. For $r_x = r_z = (0.25, 0.125, 0.4)$ Identifiers and layers of colored building area according to (a) straight and (b) curved mass models

The x_f, z_f coordinates are further scaled based on the building's facade identifier, moved from one state to another, and then evaluated as a function of distance to IFS. The transformations associated with this identifier allow each facade area to have a different shape by selecting from different areas of the IFS fractal.

Results

Our IFS creation system was introduced in C ++ and OpenGL on a Dell Optiplex workstation with a 3.4 GHz Intel Core i7-3770 processor and 8 GB of RAM, a 640 shader core Nvidia GeForce GTX 750 Ti, and 2 GB of GDDR5 dedicated video RAM.
The set of parameters that determine the appearance of the building and the values used by our system are as follows:
- Select a project model from the list (3 options).
- Enable / disable curved coordinates.
• If enabled, select s_θ, c_θ, s_r, c_r.
- Select w_x, w_z, r_x, r_z to divide the facade.
• Simplify by setting $w_x = w_z$ and $r_x = r_z$, which make faces x and z the same.
- Select for each identifier (s_x, t_x, s_z, t_z) to change the facade.
• We simplify by accepting $s_x = s_z = a_1 id + a_2$, $t_x = t_z = b_1 id + b_2$.
- Select KIFS parameters (x_c, y_c, z_c, s, R_1, R_2).
• Clear the values of $s = 3$, $R_1 = I$, $R_2 = Ry(\theta)$.
The result is 13 parameters for a straight line building and 4 parameters for a curved building.
IFS Structures formed in linear and curved coordinates are shown in Figures 7,8,9.
The results were generated and displayed in a 640 × 640 viewport for 24 ms to 133 ms (approximately 8 to 42 fps) for each frame.

The algorithm we present has several limitations. Roofs and other elements are not automatically recycled. Moving blocks allow the creation of physically impossible structures. However, this system can be extended to manage common construction methods, such as first floors, which do not fit the shape of other floors, but require more user input.

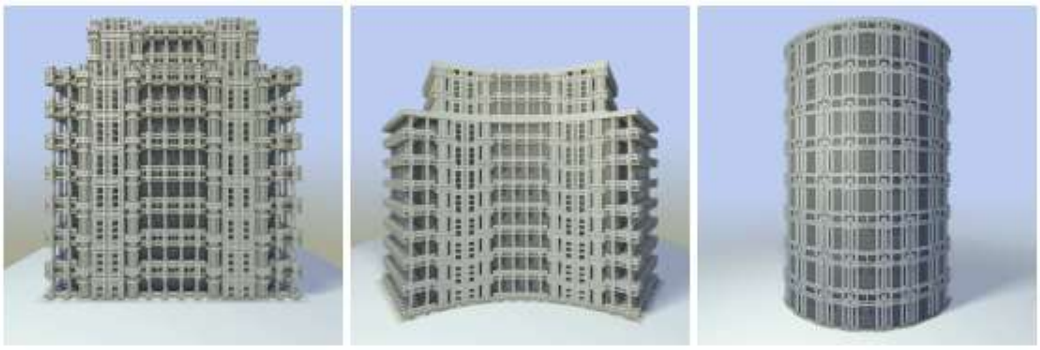
Figure 7. Straight (left) and curved (center, right) IFS buildings

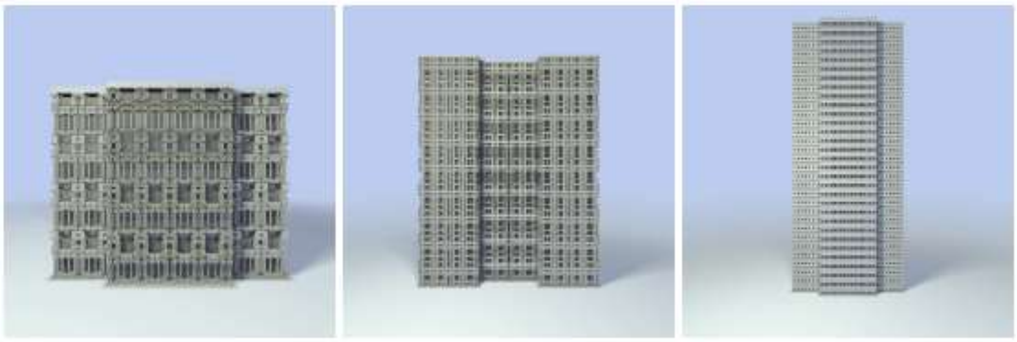
Figure 8. IFS buildings with T-shaped (left, right) and H-shaped contours of the project model

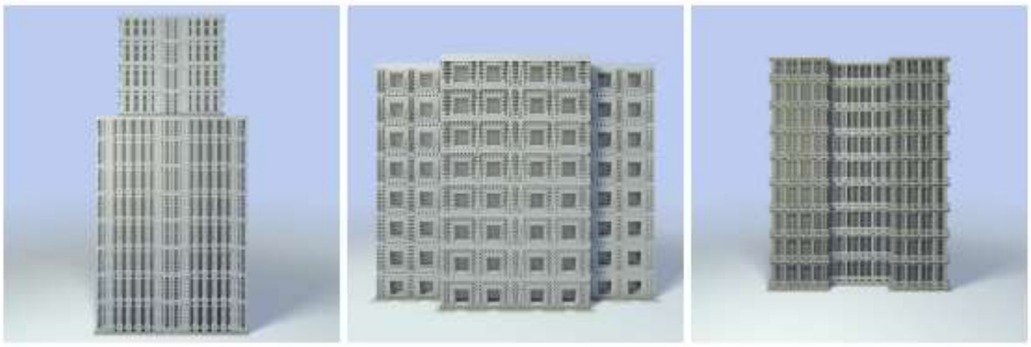
Figure 9: Additional results of IFS construction

Conclusion

This article presents a method for the procedural creation of complex buildings using decorative architectural patterns created by duplicate functional systems (IFS). For the convenience of designers, a field-changing approach has been developed that allows them to define and control periodic patterns and symmetry. This system can create and raycast the structures created on the GPU in a fully interactive way. Methods can be implemented in a few hundred lines of fragment shader code. Preliminary results show that this method can effectively build reliable buildings. Future work will provide additional tools to support the creation of buildings; improve support for non-standard elements such as entrances and roofs; includes user research and interface development to assess system usability.

References

1. Barnsley, M.F., Ervin, V., Hardin, D., Lancaster, J.: Solution of an inverse problem for fractals and other sets. Proceedings of the National Academy of Sciences of the United States of America 83 (1986) 1975–1977.
2. Demir, I., Aliaga, D.G., Benes, B.: Proceduralization of buildings at city scale. In: 2014 2nd International Conference on 3D Vision (3DV). Volume 1., IEEE (2014) 456–463.
3. M¨uller P., Zeng G., Wonka P., Van Gool L.: Image-based procedural modeling of facades. In: ACM Transactions on Graphics. Volume 26., ACM (2007) 85.
4. Aliaga, D.G., Rosen, P., Bekins, D.R.: Style grammars for interactive visualization of architecture. IEEE Transactions on Visualization and Computer Graphics 13 (2007) 786–797.
5. M¨uller P., Wonka P., Haegler S., Ulmer A., Van Gool L.: Procedural modeling of buildings. ACM Transactions On Graphics 25 (2006) 614–623.
6. McGraw T., Herring D.: Shape modeling with fractals. In: Advances in Visual Computing. Springer (2014) 540–549.
7. Barnsley M.F., Demko S.: Iterated function systems and the global construction of fractals. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. Volume 399., The Royal Society (1985) 243–275.
8. Demko, S., Hodges, L., Naylor, B.: Construction of fractal objects with iterated function systems. In: ACM SIGGRAPH Computer Graphics. Volume 19., ACM (1985) 271–278.
9. Knighty: Kaleidoscopic (escape time) IFS. http://www.fractalforums.com/ (2010)
10. Hart, J.C., Sandin, D.J., Kauffman, L.H.: Ray tracing deterministic 3-d fractals. ACM SIGGRAPH Computer Graphics 23 (1989) 289–296.

11. Quilez, I.: Modeling with distance functions. http://www.iquilezles.org (2008)
12. Sadullayeva Sh.A., Berdiyev G‘. R. “L-TIZIMLAR USULIDA FRAKTAL TUZILISHDAGI OBYEKTLARNI 3D MODELLASHTIRISH”, Рақамли технологиялар: соҳаларда амалий жорий этишнинг ечимлари ва муаммолари Республика илмий-техник анжумани. Тошкент 28-29 апрел 2021 йил. 39-43б.
13. S.Anarova, Z.Ibrokhimova, and G.Berdiev, “An Algorithm for Constructing Equations of Geometry Fractals Based on Theories of R-functions,” 2020, doi: 10.1109/ISMSIT50672.2020.9254635.
14. S.Anarova, S.Sadullaeva, G. Berdiev. “Calculation of building dimensions in the method of composition fractal analysis”, 2021 International Conference on Information Science and Communications Technologies (ICISCT). 3-5 Nov. 2021. DOI:10.1109/ICISCT52966.2021.9670390.