Hexagonal Prism: A Geometric Primitive for Architecture Design

Jorge Madera-Valdez[#1], Francisco A. Madera[#2]

[#1]*Universidad Marista, Escuela de Arquitectura*
Periférico Norte Tablaje 13941, Mérida, Yucatán, México
[#2]*Universidad Autónoma de Yucatán, Facultad de Matemáticas*
Periférico Norte Tablaje 13615, Mérida, Yucatán, México

Abstract: *We study the geometric features of the hexagonal prism and show some designs to be used in architectural geometry. Motivated by the architecture model built in [1], we propose several geometrical designs using the hexagonal prism as the basic primitive. We focus on the geometrical patterns of the hexagonal prism,we pursuit to identify those patterns that space dictates and describe how these patterns are manifested in mathematics and in computer graphics.*

Keywords —*Hexagonal Prism, Architecture Design, Computational Geometry.*

I. Introduction

The patterns we find in our buildings, in nature, and in our world are dictated by space and are identifiable by people. People have taken the patterns evident in nature and space and they have used them to create homes, offices and parks, among others.

In times gone by, single buildings tended to be one of three shapes: square, rectangular or round. The ground plan still generally follows one of those three standards but there are a number that break the conventions and go for something a little more striking like a hexagon. Noteworthy examples of hexagonal buildings can be found in [2]: Fort Jefferson Florida, Berlin Tegel Airport Berlin, Langport Workhouse England, Camberra Civic Centre Australia, Greensville Correctional Center Virginia, New York Supreme Court Building.

Our approach can be described as a guided exploration of the hexagonal prism to propose some designs to be used in architecture and industrial design. Computationally, we represent the hexagonal prism of a polygonal mesh and display a set of them in specific arrangements that we call pattern designs. Our proposal includes the hexagonal prism construction by using a circumference filled with radii.

We were inspired by the architecture model built in [1] which utilizes hexagonal prisms to represent a set of student departments (Figure 1). This model was designed in a space of 5895 m^2 for a little town called Xcunya in México. Departments are hexagonal prisms and they are arranged in 3 buildings of 10 hexagonal prisms each. We chose this primitive due to the Mayan culture [15], which considers the snake's skin (hexagons) as a base for constructing temples.

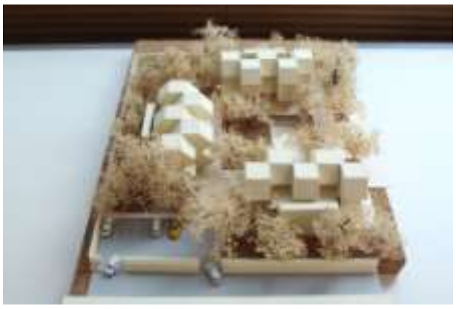


Fig. 1. The architecture model of student departments.

Shape modelling systems often provide the user with little support to satisfy constraints implied by function and fabrication of the designed product. Geometrically, complex architecture is one of the areas where an integrated design approach is in high demand. Most of the previous work in this field deals with the combination of form and fabrication; it has already entered a variety of real projects [3].

The combination of structural analysis and shape modelling is probably best studied for self-supporting masonry. In particular we here refer to the thrust network method [4,5], which may be seen as a non-conforming finite element discretization of the continuous theory of stresses in membranes [6], and which is the basis of recent work on the interactive computational design of self-supporting masonry.

We define hexagonal prism patterns and implement the computational program to generate the design, which consists of a set of hexagonal prisms. We utilize algorithms to create the geometrical patterns to represent a design. We start creating a hexagonal prism with radius*r*, and height *h* placed in the origin. We analyse the geometry of hexagonal prism and discuss their flexibility to form aesthetic shapes.

II. Related Work

Advances in architectural geometry have made it possible for many buildings to be shaped as freeform surfaces. There are several challenges in designing polyhedral patterns or prism patterns. The usage of hexagonal prism in architecture and industrial design has been done for years, for instance Glass panels and multilayer metal sheets for roofing structures [7,9].

In other work [8] explicit constructions of polyhedral patterns that approximate surfaces with varying Gaussian curvature are studied. They introduce a theoretical study of polyhedral patterns that explains their choice of regularizes, which leads to an objective function that is neither over- nor under-constrained. Our work presents an analysis of 3D geometric primitive to construct designs according to certain patterns. The surface of a prism can be decomposed in triangles for meshing construction. In the following, we provide an analysis of feasible prisms putting together in different shapes.

C. Jiang et al. [10] create free form shapes composed of hexagonal cells whose faces meet at angles close to 120 degrees. They show how to compute and design honeycomb structures, discuss applications and their limitations.

III. The Hexagonal Prism

The hexagonal prism (hexa) is a prism with hexagonal base, *8* faces, *18* edges, and *12* vertices. The volume is found by taking the area of the base, with a side length of *a*, and multiplying it by the height *h*:

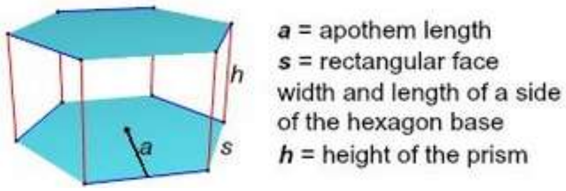
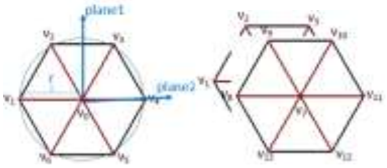

h
a
s
a = apothem length
s = rectangular face width and length of a side of the hexagon base
h = height of the prism

Fig. 2. The Hexagonal Prism and its parameters *a*, *s*, *h* [14].

There are hexagonal prisms to be observed in the world around us. Some examples could include a pencil (before it is sharpened, of course), a nut, or a stone, among many others. The surface area of a regular hexagonal prism is two times the area of the hexagonal base plus 6 times the area of the rectangle face: *6s(a + h)*, where *s*is the side length of the base, *a*is the apothem length, and *h*is the height of the prism.

We propose to build a hexa by using a circumference whose radii (or rays) can serve as vertices. A circumference is employed to obtain the vertices of the hexa. A circumference with centre in $v_0=(0,0,0)$ and radius *r* is defined. We need to create the vertices starting with $v_1=(-r, 0, 0)$. Rotating the radius in clock wise order to$\frac{360}{6}=60$ degrees we obtain v_2, and the same procedure is performed for v_3, v_4, v_5, v_6 (Figure 3). The plane *XY is*formed with the lines from the origin to *plane1=(1,0,0)* and from the origin to *plane2=(0,1,0)*. The vertices can be calculated with the next formula, from *k=0* increasing $\frac{360}{6(180)}$steps, six times:

$$plane1.xr\cos(k\pi)+plane2.xr\sin(k\pi),$$
$$plane1.yr\cos(k\pi)+plane2.yr\sin(k\pi),$$
$$plane1.zr\cos(k\pi)+plane2.zr\sin(k\pi)$$



(a) (b)

Fig. 3.Circumference's radii are in red. (a) The first seven vertices and the (b) last seven vertices.

As the height is indicated by the *z*value (the depth), we require six more vertices. We only extrude the hexa in the plane *XY* to the *Z* axis to have seven more vertices: $v_{0\to}v_7$, $v_{1\to}v_8$, …, $v_{6\to}v_{13}$.

Therefore, we create the hexa mesh. A mesh *M=(V, F, E)* is formed with $V=\{v_0, v_1, \ldots, v_{13}\}$.The first *6* triangles are formed with the centrev_0: $f_0=<v_0, v_1, v_2>$, $f_1=<v_0, v_2, v_3>$, $f_2=<v_0, v_3, v_4>$, $f_3=<v_0, v_4, v_5>$, $f_4=<v_0, v_5, v_6>$, $f_5=<v_0, v_6, v_1>$. The other six polygons are formed with v_7: $f_6=<v_7, v_9, v_8>$, … We need to create the polygons of the height of the hexa by using the vertex's indices: $f_{12}=<9,1,8>$, $f_{13}=<9,2,1>$, $f_{14}=<10,2,9>$, …

The polygons are labelled in counter clock wise order. This is important to achieve the right normal vectors. A normal vector is formed in each polygon using the cross product. Thus the normal vector for polygon $f_0=<v_0, v_1, v_2>$ isv_0v_1Gv_0v_2, which points out to allow the reflections of the lights defined in the environment.

IV. The Pattern Designs

Contrary to [11] and [12] where operations with hexas are commonly constructed by using surfaces, we arrange hexas employing affine transformations. A vertex *P=(PxPyPz 1)* can be mapped to another point *Q=(QxQyQz 1)*. Transformation operates in *P* and produces *Q* according to the function *T()*: *Q=(QxQyQz1)=T (PxPyPz 1)*.

Designs with translation only

Affine transformations are common in computer graphics and allow rotation, scaling, shearing, and translation. The *Q* coordinates are lineal combinations of *P* coordinates.
$Q=(QxQyQz\ 1) = (m_{11}Px+m_{12}Py+m_{13}Pz+m_{14}$ $m_{21}Px+m_{22}Py+m_{23}Pz+m_{24})$for some constants m_{11}, m_{12}, …, m_{23}, m_{24}.
Then we have

$$Q = (Qx\ Qy\ Qz\ 1)\ = \begin{pmatrix} m_{11}m_{12}m_{13}m_{14} \\ m_{21}m_{22}m_{24}m_{24} \\ m_{31}m_{32}m_{33}m_{34} \\ 0\quad 0\quad 0\quad 1 \end{pmatrix}$$

The translation to be done is indicated by the fourth column of the matrix:

$$(Qx\ Qy\ Qz\ 1)\ = \begin{pmatrix} 1\ 0\ 0m_{14} \\ 0\ 1\ 0m_{24} \\ 0\ 01m_{34} \\ 000\ \ 1 \end{pmatrix}(P_xP_yP_z\ 1)$$

Or simply

$$(Qx\ Qy\ Qz\ 1)\ = \begin{pmatrix} Px+m_{14} \\ Py+m_{24} \\ Pz+\ m_{34} \\ 1 \end{pmatrix}$$

Algebraically, *Q=P+d*, where the displacement vector *d* equals *(*m_{14} m_{24} m_{34}*)*.

The first design (Figure 4) consists of the hexas placed in columns from left to right, and rows from bottom to top.
Display_Hexa(2(i+1)v_4*.x, 2 j* v_2*.y, 0), i,j=0,1,…, n*

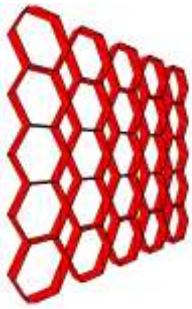


Fig. 4. Design 1 is formed with rows of hexas arranged vertically.

A variation of Design *1* can be obtained translating the hexas in *X*-axis (Figure 5, Design 2)
Display_Hexa(2(i+1)v_4*.x +j*v_4*.x, 2 j* v_2*.y, 0)*

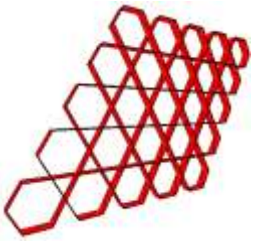


Fig. 5. Design 2 performs the shearing operation tohexas of Design 1.

Another variation (Figure 6, Design 3) is achieved by using an *X*-axis translation:
Display_Hexa(iv_4*.x, 2 j* v_2*.y, 0).*

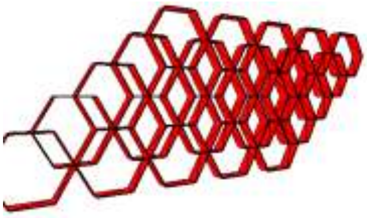


Fig. 6. Design 3 translates hexas of Design 1to the *X*-axis.

Note in these first three designs, the vertical value is the same: *2 j* v_2*.y*, and thez value equals zero. Another variation by using an *X*-axis translation is achieved (Figure 7, Design 4). We employ the circumference values instead to define the hexas location. This is similar to the previous Design 3.
Display_Hexa(2(i+1)rayo$_3$*.x + 3 (j+1) rayo*$_2$*.x, j rayo*$_2$*.y, 0)*

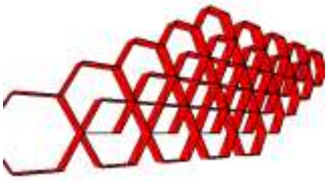


Fig. 7. Design 4 applies another translation to Design 1, but using the rays of the circumference to indicate the hexa's location.

Combining different strips and symmetries leads to a large variety of aesthetic results from a single basic pattern. We could perform some operations between the hexas such as translation in *Y*-axis, change of displaying rays, change of displaying facets, coloring, lights, change *r* and *h*, smooth the object, etc. Observe there is no intersection among hexas in Design 1, but in Designs 2,3,4.

The next design is well known and consists on placing hexas without intersection, that is, edges are adjacent (Figure 8, Design 5). We could use a recursive routine to construct such design.

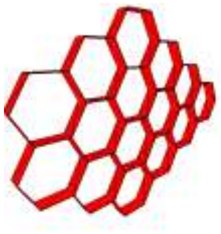


Fig. 8. Design 5, hexas without intersection simulating a honeycomb structure.

Starting from an hexa in the origin, the two edges of the right are considered to continue the construction. Then we call the construction of the hexas from the upper edge and the construction of the hexas from the lower edge. The routineis recursive and require a stop condition, depending on the size of the structure.

```
Make_Hexa(centroX, centroY)
Display_Hexa(centroX+$v_3$.x +$v_4$.x, centroY+ $v_2$.y , 0)
Display_Hexa(centroX+$v_3$.x +$v_4$.x, centroY- $v_2$.y , 0)
Make_Hexa(centroX+$v_3$.x +$v_4$.x, centroY+ $v_2$.y)
```

Designs with translation and rotation

Rotations can be performed in any axis or line in the space. Fortunately our designs perform rotations in *X*,*Y*, or *Z* axis. In 2D, when *T()*performs a rotation in the origin, the displacement vector *d* equals zero, and *Q=T(P)* is as follows:

(A1) $Qx=Pxcos\theta- Py\ sin\theta$

(A2) $Qy=Px\ sin\theta- Pycos\theta$

This results in positive values of θ and the angle follows a counter clockwise motion. In 2D we can represent a rotation in the origin as follows:

$$\begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

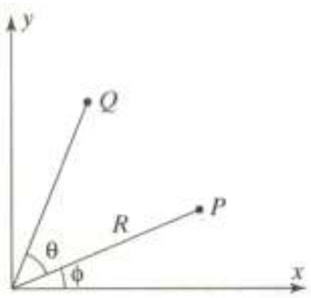


Fig. 9. Rotation from *P* to *Q*[13].

Figure 9depicts how to obtain the coordinates from a point *Q* which results to rotate *P* in the origin with angle θ. If *P* is located to a distance *R* from the origin with angle ϕ, then $P=(R\ cos\theta,\ R\ sin\theta\ )$. *Q* must be at the same distance from *P* and the angle $\theta+\phi$. Using trigonometry, we find *Q*:

$Qx=Pxcos\theta-Pysin(\theta+\phi)$
$Qy=Pxsin\theta-Pycos(\theta+\phi)$

Employingtrigonometryidentities:
$cos(\theta+\phi)=cos\theta cos\phi- sin\theta sin\phi$
$sin(\theta+\phi)=sin\theta cos\phi-cos\theta sin\phi$

and utilizing $Px=R\ cos\theta$, $Py=R\ sin\theta$, we obtain (A1) and (A2). In matricial form, we have:

$$M = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finally, the matrix multiplication is calculated: *Q=MP*.

Design 6 (Figure 10) employs two hexas for each location, the first one remains as original and the second one is rotated *90* degrees around the*X*-axis.

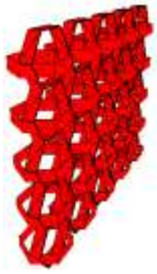


Fig. 10. Design 6 adds rotated hexas.

Making horizontal displacements we obtain another similar design (Figure 11, Design 7).

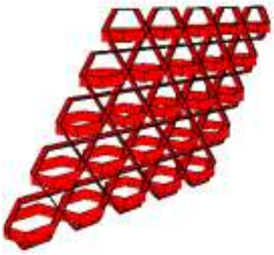


Fig. 11. Design 7 with translation and rotated hexas.

By putting two rotated hexas instead we could modify Design 7 (Figure 12, Design 8).

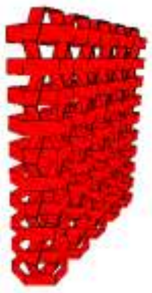


Fig. 12. Design 8 by adding rotated hexas.

Changing the rotation angle to *Y*-axis we achieved Design 9 (Figure 13).

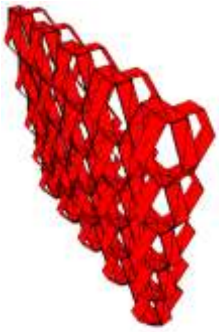


Fig. 13. Design 9, where the rotation is performed in the Y-axis.

V. Computational Issues

An hexa can be represented with *14* vertices and *24* polygons. Each vertex stores *3* floats of *4* bytes each giving *12* bytes per vertex.Thus, *14* vertices x *12* bytes = *168* bytes. Polygons store *3* integers each, making reference to the vertex indices, so that we can use unsigned short int of *2* bytes, giving *6* bytes per polygon. Then, *6*bytes x *24* polygons = *144* bytes. Therefore, an hexa requires *144* + *168* bytes = *312* bytes. As we need *nxn* hexas to represent our design, the space complexity equals $312O(n^2)$.Temporal complexity refers to the double loop performed to display the hexas: $O(n^2)$.

We construct and display hexas in an arranged manner pursuing aesthetic models,without making additional operations. The following step would be to convert our line structure to a curve or to a surface structure. Computational program was built in C++ with OpenGL, requiring minimum sources in comparison with the usage of a 3D modelling software.

VI. Conclusions

We have shown some designs for architectural geometry using a basic primitive, the hexagonal prism. By joining the hexas using affine transformations we can generate a variety of designs which can be utilized in architectural modeling or industrial design.

Our operations use basic arithmetically operationsso that we could arrange the hexas as required. A mathematical and computational description was made by analyzing the geometrical patterns. We demonstrate the temporal and space complexity are useful to consider in computational processing, in particular for larger projects.

Although our research is primarily inspired by architecture, the problem is of relevance in the broader context of geometric modeling. As a future work we plan to study free form shapes with other kind of prisms for support in architectural applications, folding patterns, and timevarying polyhedral patterns for shading systems.

References

[1] Universidad Marista, Mérida, México. http://www.marista.edu.mx/

[2] Famous Hexagonal Buildings. http://infomory.com/famous/famous-hexagonal-buildings/

[3] C. Tang, X. Sun, A. Gomes, W. Johannes, H. Pottmann.*Form-finding with polyhedral meshes made simple*, *ACM Trans. On Graphics*. Vol. 33, No. 4, pp. 701-709, 2014.

[4] P. Block, J. Ochsendorf, *Trust network analysis: A new methodology for three dimensional equilibrium*, *Journal International Assoc. Shel and Spatial Structures*. Vol. 48, No. 3, pp. 167-173, 2007.

[5] P. Block. *Trust network analysis: Exploring Three dimensional equilibrium*. PhD Thesis MIT, 2007.

[6] F. Fraternali.*A Trust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions*, *Mechanics Re. Comm*. vol. 37, no. 2, pp. 198-204, 2010.

[7] H. Pottmann, Y. Liu, J. Wallner, A. Bobenko, W. Wang. *Geometry of multilayer freeform structures for architecture*, *ACM Trans. On Graphics*. Vol. 26, No. 3, 2007.

[8] C. Jiang, C. Tang, A. Vaxman, P. Wonka, H. Pottmann. *Polyhedral patterns.ACM Trans. On Graphics*. Vol. 34, No. 6, 2015.

[9] M. Eigensatz, M. Kilian, A. Schiftner, N. J. Mitra, H. Pottmann, M. Pauly. *Paneling architectural freeform surfaces.ACM Trans. On Graphics*. Vol. 29, No. 4, 2010.

[10] C. Jiang, C. Tang, J. Wang, J. Wang, H. Pottmann. Freeform honeycomb structures and lobel frames.In SIGGRAPH ACM *Poster*,2015.

[11] H. Pottmann, C. Jiang, M. Höbinger, J. Wang, P. Bompas, J. Wallner. *Cell packing structures. Elsevier Computer Aided Design*. Vol. 60, pp. 70-83, 2015.

[12] C. Jiang, C. Tang, M. Tomiči, J. Wallner, H. Pottmann. *Interactive modeling of architectural freeform structures: Combining Geometry with Fabrication and Statics*. Advances in Architectural Geometry, 2014.

[13] P. Schneider, D. H. Eberly. *Geometric Tools for Computer Graphics*. Morgan Kaufmann, first edition, 2002.

[14] Hexagonal Prism: Properties, Formula & Examples. http://study.com/academy/lesson/hexagonal-prism-properties-formula-examples.html#lesson

[15] J. Díaz-Bolio. *The Geometry of the Maya and the Rattlesnake Art*. Area Maya, 1987.