Appendix E
Articles Published During the Period August 2002 – May 2003

Presenting Methods for Unraveling the First Two Regular 4D Polytopes (4D Simplex and the Hypercube)

Antonio Aguilera Ramírez
aguilera@mail.udlap.mx

Ricardo Pérez Aguila
sp104378@mail.udlap.mx

Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

Abstract

This article presents methods for unraveling the hypercube and the 4D simplex and obtaining the unravelings that corresponds to the hyper-flattening of their boundaries. These regular polytopes can be raveled back using the methods in an inverse way. The transformations to apply include rotations around a plane (characteristic of the 4D space). All these processes can be viewed using a computer animation system.

Proceedings of the 2002 International Symposium Cyber Worlds: Theory and Practices CW2002, pp. 439-446. Editors: Peng, S., Savchenko, V.V. & Yukita, S. Published by the IEEE Computer Society. November 6 to 8, 2002. Ichigaya, Tokyo, Japan. ISBN: 0-7695-1862-1.

Presenting Methods for Unraveling the First Two Regular 4D Polytopes (4D Simplex and the Hypercube)

Antonio Aguilera Ramírez
aguilera@mail.udlap.mx

Ricardo Pérez Aguila
sp104378@mail.udlap.mx

Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

Abstract

This article presents methods for unraveling the hypercube and the 4D simplex and obtaining the unravelings that corresponds to the hyper-flattening of their boundaries. These regular polytopes can be raveled back using the methods in an inverse way. The transformations to apply include rotations around a plane (characteristic of the 4D space). All these processes can be viewed using a computer animation system.

1. Introduction

Recent interest has been growing in studying multidimensional polytopes (4D and beyond) for representing multidimensional phenomena in the Euclidean n-dimensional space. Some of these phenomena's features rely on the polytope's geometric and topologic relations. So, we have developed some algorithms for classifying n-D polytopes' elements as manifold or non-manifold [2]. However, [3] motivates us to think about two important questions: Is it possible to visualize a polytope to know how it looks like? And if we can't see it, how can we be sure about the proper understanding of its relations and properties? The answer is that the task of visualizing polytopes in the fourth and higher dimensions belongs to the computer graphics field [3]. Visualizing these new dimensions lead us to learn and to understand the events, relationships and properties for these phenomena.

[3], [5], [8], [10] and [12] start their introductions to the 4D space study presenting three methods for visualizing the hypercube: through its shadows (projections), its cross sections with 3D space, and its unravelings.

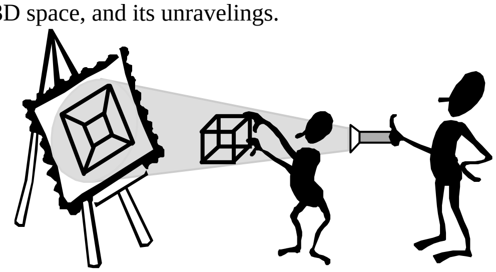


Figure 1. Projecting a cube on a plane (central projection).

If it is possible to make drawings of 3D solids when they are projected onto a plane, then it is possible to make drawings or 3D models of 4D polytopes when they are projected onto a hyperplane [5]. The shadows method is based in this principle.

Let us follow the analogy presented in "Flatland" [1]. If a 3D being wants to show a cube to a 2D being (a flatlander) then the first one must project the cube's shadow onto the plane where the flatlander lives. For this case, the projected shape could be, for example, a square inside another square (Figure 1).

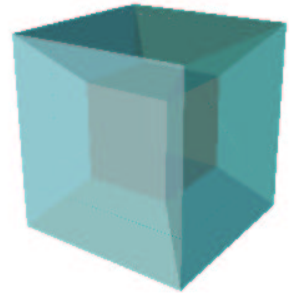


Figure 2. Hypercube's central projection onto the 3D space.

If a 4D being wants to show us a hypercube, he must project the shadow onto the 3D space where we live. The projected body could be a cube inside another cube [8] called central projection (Figure 2). We know that a projected cube onto a plane is just an approximation of the real one. Analogously, the hypercube projected onto our 3D space is also a mimic of the real one. Another useful projection is due to Claude Bragdon (see [11] for details about this projection). See Figure 3.

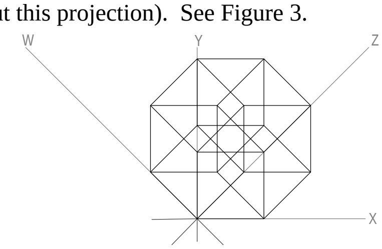


Figure 3. Claude Bragdon's hypercube projection.

A cube can be unraveled as a 2D cross. The six faces on the cube's boundary will compose the 2D cross (Figure 4). The set of unraveled faces is called the unravelings of the cube.

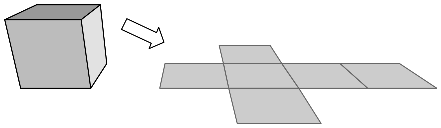


Figure 4. Unraveling the cube.

In analogous way, a hypercube also can be unraveled as a 3D cross. The 3D cross is composed by the eight cubes that forms the hypercube's boundary [8]. This 3D cross was named ***tesseract*** by C. H. Hinton (Figure 5).

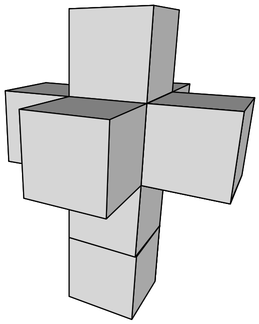


Figure 5. The unraveled hypercube (the tesseract).

A flatlander will visualize the 2D cross, but he will not be able to assembly it back as a cube (even if the specific instructions are provided). This fact is true because of the needed face-rotations in the third dimension around an axis which are physically impossible in the 2D space. However, it is possible for the flatlander to visualize the raveling process through the projection of the faces and their movements onto the 2D space where he lives.

Analogously, we can visualize the tesseract but we won't be able to assembly it back as a hypercube. We know this because it is necessary the translation of the cubes (the hypercube's boundary) in the fourth dimension and rotate them around a plane (this transformations are physically impossible in our 3D space).

Before going any further, we would like to underline that the cube's boundary faces can be grouped into three pairs of parallel faces, where their supporting planes define two 2D-spaces parallel to each other. Each pair can be obtained by ignoring all those edges parallel to each main axis (X, Y and Z), see Figure 6

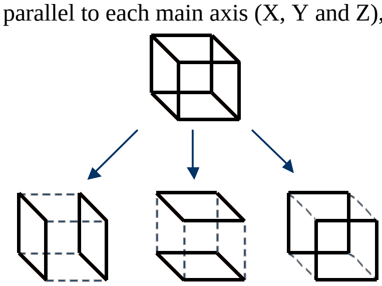


Figure 6. Viewing the cube's boundary faces.

It is interesting to analyze the hypercube using its analogy with the cube and the visualization methods above described. [6] has determined that a hypercube is composed of sixteen vertices, twenty-four faces and eight bounding cubes (also called cells or volumes). Similarly, and as shown in Figure 7, all these volumes can be grouped into four pairs of parallel cubes, furthermore, their supporting hyper-planes define two 3D-spaces parallel to each other [9]. Moreover, [4] states it is instructive for the reader to find all eight bounding cubes in the Bragdon's projection.

[5] points that each face is shared by two cubes not in the same three-dimensional space, because they form a right angle through a rotation around the shared face's supporting plane. These properties are visible through Bragdon's projection (Figure 3). The Bragdon's projection as well as the central projection will be used through the remaining of this work.

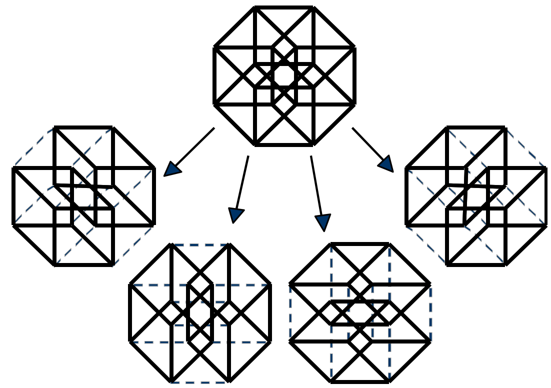


Figure 7. Viewing the hypercube's boundary volumes.

2. Problem

[3] and [8] describe with detail a representation model for the hypercube through their unravelings. They also mention the physical incapacity of a 3D being to ravel the hypercube back, because the required transformations are not possible in our 3D space (Figure 8).

[3] and [8] also describe that if we witness the raveling process, seven of eight cubes that compose the tesseract will suddenly disappear, because they have moved in the direction of the fourth dimension. However, they don't provide a methodology that indicates the transformations and their parameters to execute the raveling process. In spite of our physical incapacity, we can visualize a projection onto our 3D space of the cubes on the hypercube's boundary through the unraveling and raveling processes.

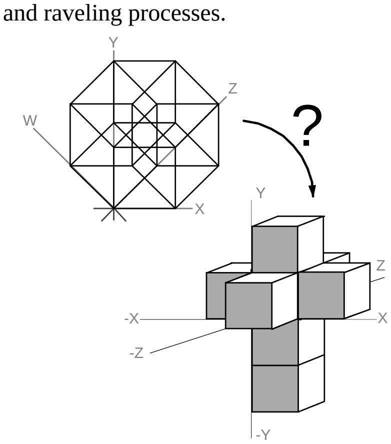
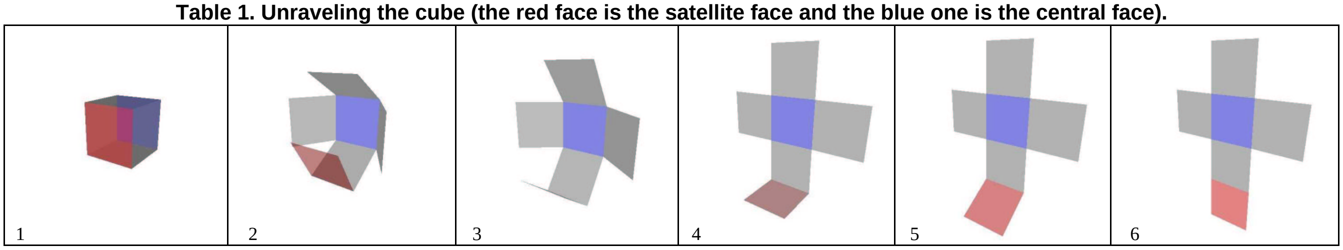


Figure 8. The hypercube's unraveling process.

This article presents a method for unraveling the hypercube and getting the 3D-cross (tesseract), and unraveling the 4D simplex and getting the *stellated tetrahedron* that corresponds to the hyper-flattening of their boundary. These polytopes can be raveled back using the same method in an inverse way. The transformations to apply include rotations around a plane (See [7] for details about the topic). All these processes can be viewed using a computer animation system.

Table 1. Unraveling the cube (the red face is the satellite face and the blue one is the central face).



3. Unraveling the 4D Hypercube

3.1. Cube's Unraveling Methodology

Although this process is absolutely trivial, it is included here to underline some key points that will be very useful when extending it to the 4D case.

The unraveling process for a cube can be resumed in the following steps:

1. Identify a face that is "naturally embedded" into the plane where all the cube's faces will be positioned. This face will be called "central face". Because the central face is "naturally embedded" in the selected final plane (for example, the XY plane), it will not require any transformation.
2. Identify those faces that share an edge with the central face. There are four of such faces and they will be called "adjacent faces".
3. After the identification of the central and adjacent faces there will be a face whose supporting plane is parallel to central face's supporting face. This face will be called "satellite face" because its movements will be around an edge that is shared with any arbitrary selected adjacent face (and the selected adjacent face will rotate around an edge that is shared with the central face).
4. The adjacent faces will rotate around those edges that share with the central face.
5. When the central, adjacent and satellite faces are identified, it must be determined the rotating angles and their directions. All four adjacent faces will rotate right angles, however two opposite adjacent faces will have opposite rotating directions; otherwise, one of them will end in the same position as the central face.

Table 1 presents some snapshots from the cube's unraveling sequence. In snapshots 1 and 2, the applied rotations are 0° and ±30° (the rotation's sign depends of the adjacent face). In snapshot 3, the applied rotation is ±53°; the satellite face looks like a straight line --an effect due to the selected 3D-2D projection. In snapshot 4 the applied rotation is ±90°; the adjacent faces have finished their movements. In snapshots 5 to 6, the satellite face moves independently and the applied rotations are +60° and +90°.

3.2. Hypercube's unraveling methodology

The process will be easier if we take the following considerations:

- Select the hypercube's position in the 4D space.
- Select the hyperplane (a 3D subspace embedded in the hyperspace) where the volumes will be directed to.
- Establish the angles which guarantee that all volumes will be totally embedded in the selected hyperplane.
- All the volumes through their movement into the selected hyperplane must maintain a face adjacent to another volume.

Table 2. Hypercube's coordinates.

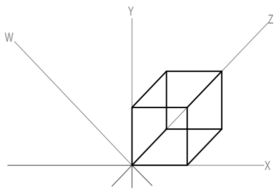
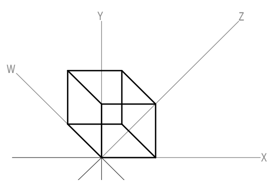
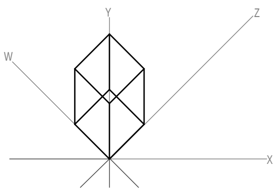
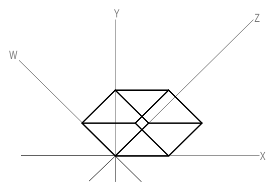
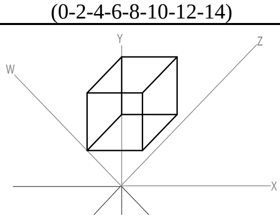
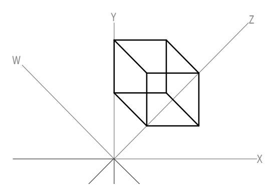
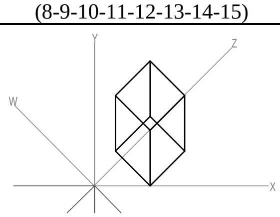
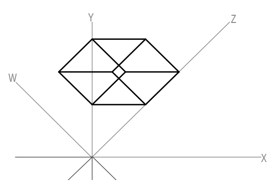
Vertex	*X*	*Y*	*Z*	*W*
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

The hypercube's position in the 4D space is essential, because it will define the rotating planes used by the volumes to be positioned onto a hyperplane. For simplicity, one vertex of the hypercube will coincide with the origin, six of its faces will coincide each one with some of XY, YZ, ZX, XW, YW and ZW planes and all the coordinates will be positive (see [3] for the methodology to get the hypercube's coordinates). The coordinates to use are presented in Table 2 (each vertex is arbitrary numbered).

We know now why the hypercube's position in the 4D space is important, since it will define the rotating planes to use. The situation is the same for the selected hyperplane, because it is where all the volumes will be finally positioned. Observing the hypercube's coordinates we can see that eight of them present their fourth coordinate value (W) equal to zero. This fact represents that one of the hypercube's volumes (formed by vertexes 0-1-2-3-4-5-6-7) has W=0 as its supporting hyperplane. Selecting the hyperplane W=0 is useful because one of the volumes is "naturally embedded" in the 3D space and it won't require any transformations.

Now, it is also useful to identify the hypercube's volumes through their vertices and to label them for future references. Until now we have one identified volume, it is formed by vertexes 0-1-2-3-4-5-6-7, and it will be called volume A. See Table 3.

Table 3. The hypercube's volumes (the numbers indicate the vertices that compose them).

 Volume A (0-1-2-3-4-5-6-7)	 Volume B (0-1-2-3-8-9-10-11)
 Volume C (0-2-4-6-8-10-12-14)	 Volume D (0-1-4-5-8-9-12-13)
 Volume E (8-9-10-11-12-13-14-15)	 Volume F (4-5-6-7-12-13-14-15)
 Volume G (1-3-5-7-9-11-13-15)	 Volume H (2-3-6-7-10-11-14-15)

We have already described volume A as "naturally embedded" in the 3D space, because it won't require any transformations. Volume A will occupy the central position in the 3D cross and it will called the "central volume".

From the remaining volumes, six of them will have face adjacency with the central volume. Due to this characteristic they can easily be rotated toward our space because their rotating plane is clearly identified. Each of these volumes will rotate around the supporting plane of its shared face with central volume. They will be called "adjacent volumes". Adjacent volumes are B, C, D, F, G and H. The remaining volume E will be called "satellite volume" and it will be discussed later on.

All of the adjacent volumes will rotate right angles. In this way we guarantee that their W coordinate will be equal to zero. As in the 3D case, it is also important to consider their rotating directions, because the volumes, after the rotations, could otherwise coincide with the central volume. The direction and rotating planes for each adjacent volume are presented in Table 4 (the central volume is also included in each image as a reference for the initial and final position of the volume being analyzed).

At this point, we have seven of the eight hypercube's volumes placed in their final positions (volumes A, B, C, D, F, G and H). Volume E will perform a rather more complex set of transformations. There are two reasons that justify this conclusion:

- The supporting hyperplane for volume E is parallel to the supporting hyperplane for the central volume. Consequently, there are no adjacencies between volume E and central volume (this is the reason for not calling "adjacent volume" to volume E).
- In the tesseract, we still have an empty position. This position corresponds to the most distant volume from the central volume (the inferior position, Figure 5). This position will be occupied by volume E. This is the reason for calling "satellite volume" to volume E.

Table 4. Applied transformations to the adjacent volumes.

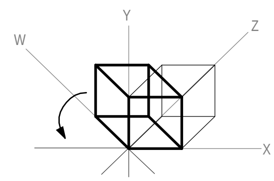
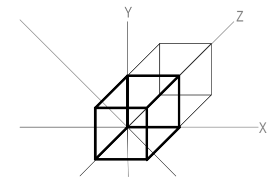
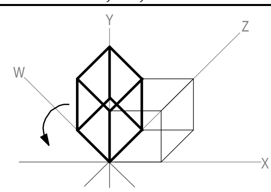
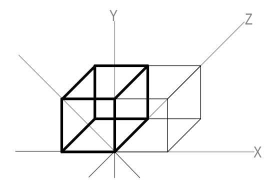
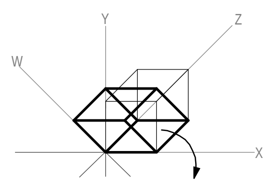
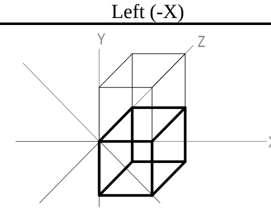
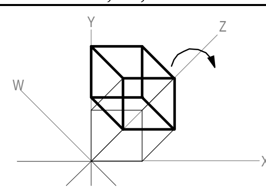
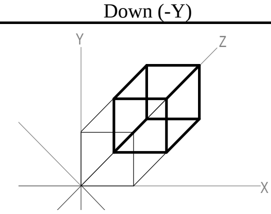
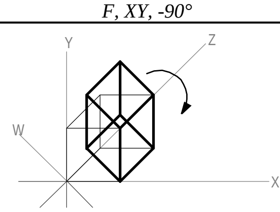
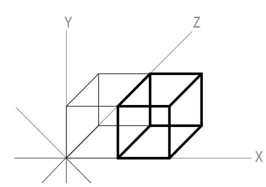
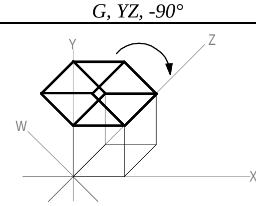
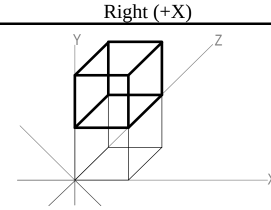
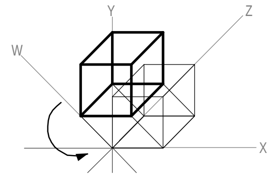
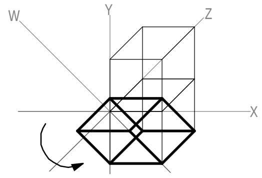
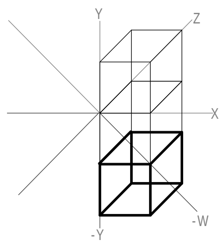
Adjacent volume (previous to rotation), rotation plane and angle	*Position in the 3D space and in the tesseract after rotation*
 *B, XY, +90°*	 Front (-Z)
 *C, YZ, -90°*	 Left (-X)
 *D, ZX, +90°*	 Down (-Y)
 *F, XY, -90°*	 Back (+Z)
 *G, YZ, -90°*	 Right (+X)
 *H, ZX, -90°*	 Up (+Y)

Table 5. Associated transformations to satellite volume.

Current position	*Transformations*
	Rotation of volumes D and satellite around the plane ZX (+90°).
	Volume D is in its final position. Rotation of satellite volume of +90° around the shared face with volume D (parallel plane to ZX).
	Satellite volume in its final position (inferior position in the 3D cross on Y axis).

At the beginning of this document its is mentioned the need for maintaining a face adjacency between all the volumes while they rotate towards the selected hyperplane. Volumes B, C, D, F, G and H share a face with central volume (remember that central volume is static during the whole unraveling process). In order to determine the needed transformations for the satellite volume, we must first select the volume which will share a face with it. Any volume, except the central one, can be selected for this. In this work, volume D will be selected to share a face with satellite volume through the hyper-flattening process.

The direction and the rotation plane for volume D was determined before (ZX plane +90°). These transformations will take it to its final position. During the beginning of the unraveling process, the same transformations will be applied to satellite volume. In this way, we ensure that volumes E and satellite will share a face.

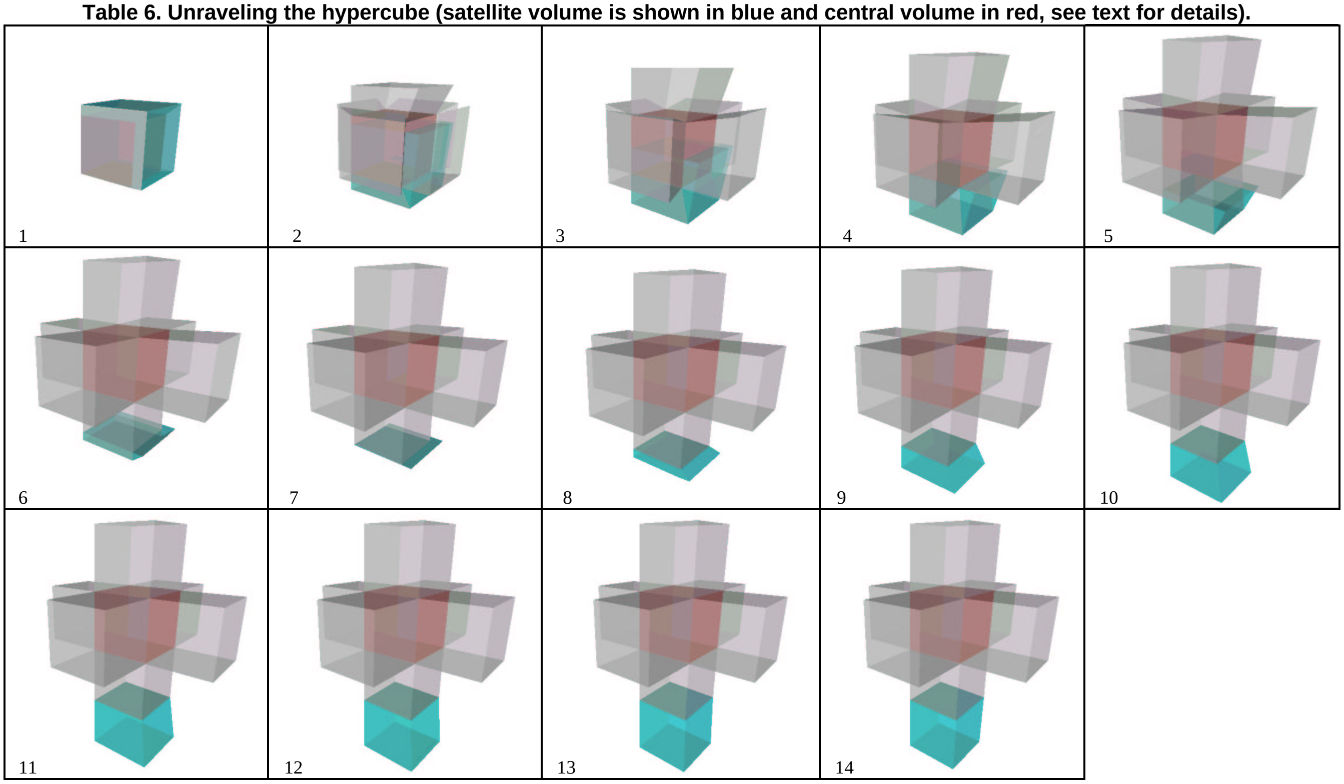
When volume D has finished its movement, it will be placed in its final position in the tesseract. At this moment, the satellite volume's supporting hyperplane will be perpendicular to the selected hyperplane and the shared face will be parallel to ZX plane. The last movement to apply to the satellite volume will be a +90° rotation around the supporting plane of the shared face with volume D.

The set of movements to be executed for the satellite volume are resumed in the Table 5 (Central volume and volume D are shown too).

3.3. Visualizing The Hypercube's Unraveling Process

Table 6 presents some snapshots from the hypercube's unraveling sequence. In snapshots 1 to 6, the applied rotations are ±0°, ±15°, ±30°, ±45°, ±60° and ±75° (the rotation's sign depends on the adjacent volume). In snapshot 7, the applied rotation is ±82°; the satellite volume looks like a plane --an effect due to the selected 4D-3D projection. In snapshot 8, the applied rotation is ±90°; the adjacent volumes finish their movements. In snapshots 9 to 14, the satellite volume moves independently and the applied rotations are +15°, +30°, +45°, +60°, +75° and +90°.

Table 6. Unraveling the hypercube (satellite volume is shown in blue and central volume in red, see text for details).



4. Unraveling the 4D Simplex

Using a method similar to that of unraveling a cube, we presented how to unravel a 4D hypercube. We have not found any references that mention any methods or results (like the tesseract as the result of the hypercube's unraveling) about the unraveling process for other 4D regular polytopes such as the 4D simplex which corresponds to the 4D equivalent of the tetrahedron (Figure 9). As the hypercube's unraveling process, we will visualize a projection onto our 3D space of the volumes (tetrahedrons) on the 4D simplex's boundary through its unraveling and raveling processes.

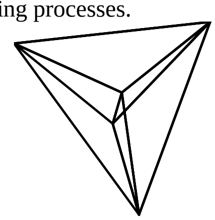


Figure 9. The 4D simplex

4.1. The Tetrahedron Unraveling Methodology

Although the tetrahedron’s (3D simplex) unraveling process is trivial, we will consider here some key points that will be extended later in the 4D simplex unraveling:

1. Identify a face that is "naturally embedded" into the plane where all the tetrahedron's faces will be positioned. This face will be called "central face". Because the central face is "naturally embedded" in the selected plane, it will not require any transformation.
2. Each of the remaining faces shares an edge with the central face. These faces will be called "adjacent faces".
3. The adjacent faces will rotate around those edges that share with the central face.
4. When the central and adjacent faces are identified, it must be determined the rotating angles and their directions. The rotating angle is the supplement of the tetrahedron's dihedral angle. Finally we obtain a ***stellated triangle***.

Table 7. Unraveling the 3D simplex (see text for details).

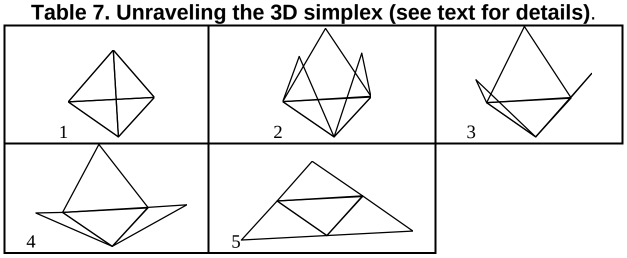


Table 7 presents some snapshots from the 3D simplex's unraveling sequence. In snapshots 1 and 2, the applied rotations are ±0 and ±27.35° (the rotation’s sign depends of the adjacent face). In snapshot 3 the applied rotation is ±54.7°; one adjacent face looks like a straight line --an effect due to the selected 3D-2D projection. In snapshots 4 and 5, the applied rotations are ±76.58° and ±109.4°.

4.2. The 4D Simplex's Unraveling Methodology

Because the 4D simplex boundary is composed by five tetrahedrons [5], we can expect, by analogy, that the unravelings of the 4D simplex will be a tetrahedron surrounded by four other tetrahedrons and sharing a face with each one (the unravelings of the tetrahedron are a triangle surrounded by other three triangles and sharing an edge with each one). We will refer to the unravelings of the 4D simplex as a ***stellated tetrahedron*** (as the unravelings of the hypercube are referred as the tesseract).

We will consider and adapt the same recommendations proposed in section 3.2 to unraveling the simplex:

- Select the simplex's position in the 4D space.
- Select the hyperplane (a 3D subspace embedded in the hyperspace) where the volumes will be directed to.
- Establish the angles which guarantee that all volumes will be totally embedded in the selected hyperplane.
- All the volumes through their movement into the selected hyperplane must maintain a face adjacent to another volume.

Table 8. The 4D simplex coordinates.

Vertex	X	Y	Z	W
0	0	0	0	0
1	1	0	0	0
2	$1/2$	$\sqrt{3}/2$	0	0
3	$1/2$	$\sqrt{3}/6$	$\sqrt{2/3}$	0
4	$1/2$	$\sqrt{3}/6$	$\sqrt{2}/4\sqrt{3}$	$\sqrt{5/8}$

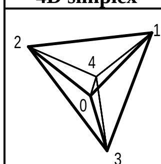
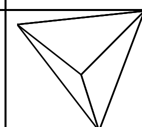
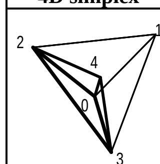
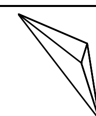
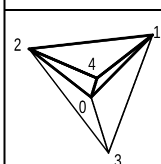
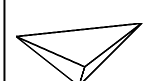
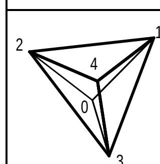
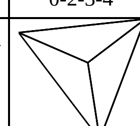
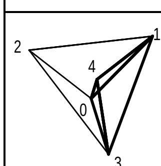
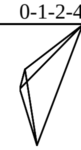
We consider that the simplex will have a position with the following characteristics:

- One vertex of the simplex will be the origin.
- An edge will coincide with X axis.
- A face will coincide with XY plane.
- All the coordinates will be positive.

The coordinates to use are presented in Table 8 (see [3] for a methodology to get the 4D simplex's coordinates).

Observing the 4D simplex's coordinates we can see that four of them present their fourth coordinate value (W) equal to zero. This fact represents that one of the simplex's volumes (formed by vertexes 0-1-2-3) has W=0 as its supporting hyperplane. Selecting the hyperplane W=0 is useful because one of the volumes is "naturally embedded" in the 3D space and it won't require any transformations.

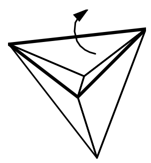
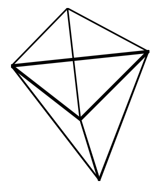
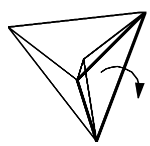
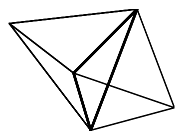
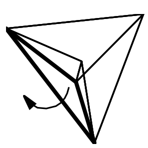
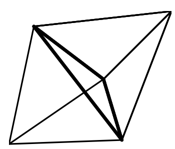
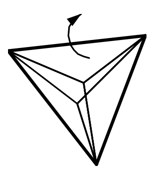
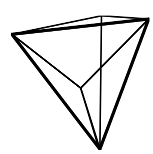
Table 9. The 4D simplex's boundary volumes.

Volume's position in the 4D simplex	Volume and Label	Volume's position in the 4D simplex	Volume and Label
	 Volume A 0-1-2-3		 Volume D 0-2-3-4
	 Volume B 0-1-2-4		 Volume E 1-2-3-4
	 Volume C 0-1-3-4		

Now, it is also useful to identify the simplex's volumes through their vertices and to label them for future references. Until now we have one identified volume, it is formed by vertexes 0-1-2-3, and it will be called volume A. See Table 9.

We have already described volume A as "naturally embedded" in the 3D space, because it won't require any transformations. Volume A will occupy the central position in the stellated tetrahedron and it will called the "central volume".

Table 10. Applied transformations to the adjacent volumes (rotation around XY plane is the same for all volumes).

Adjacent volume previous to rotation	Transformations to apply	Position in the stellated tetrahedron after the transformations
	Volume B **XY** 104° 29'	
	Volume C XW 109° 30’ **XY** 104° 29' XW -109° 30’	
	Volume D ZW -60° XW 70° 30’ **XY** -104° 29' XW -70° 30’ ZW 60°	
	Volume E T(-1,0,0,0) ZW 60° XW 70° 30’ **XY** -104° 29' XW -70° 30’ ZW -60° T(1,0,0,0)	

All of the remaining volumes will have face adjacency with the central volume. Due to this characteristic they can "easily" be rotated toward our space because their rotating plane is clearly identified. Each of these volumes will rotate around the supporting plane of its shared face with central volume. They will be called "adjacent volumes".

Although the rotating planes are clearly identified, the main difference between the hypercube and simplex's unraveling is that the rotating planes don't correspond to 4D space main planes (XY, YZ, ZX, XW, YW and ZW) in the simplex's unraveling. Due to this situation, the volume's rotations will be a composition of rotations around the 4D space main planes. The objective taken for us was to position a volume's face in the XY plane, and then rotate it 104° 29'. This angle corresponds to the supplement of the simplex's dihedral angle that is 75° 31' [4]. In this way we guarantee that their W coordinate will be equal to zero. The direction and rotating planes for each adjacent volume are presented in Table 10 (the central volume is also included in each image as a reference for the initial and final position of the volume being analyzed).

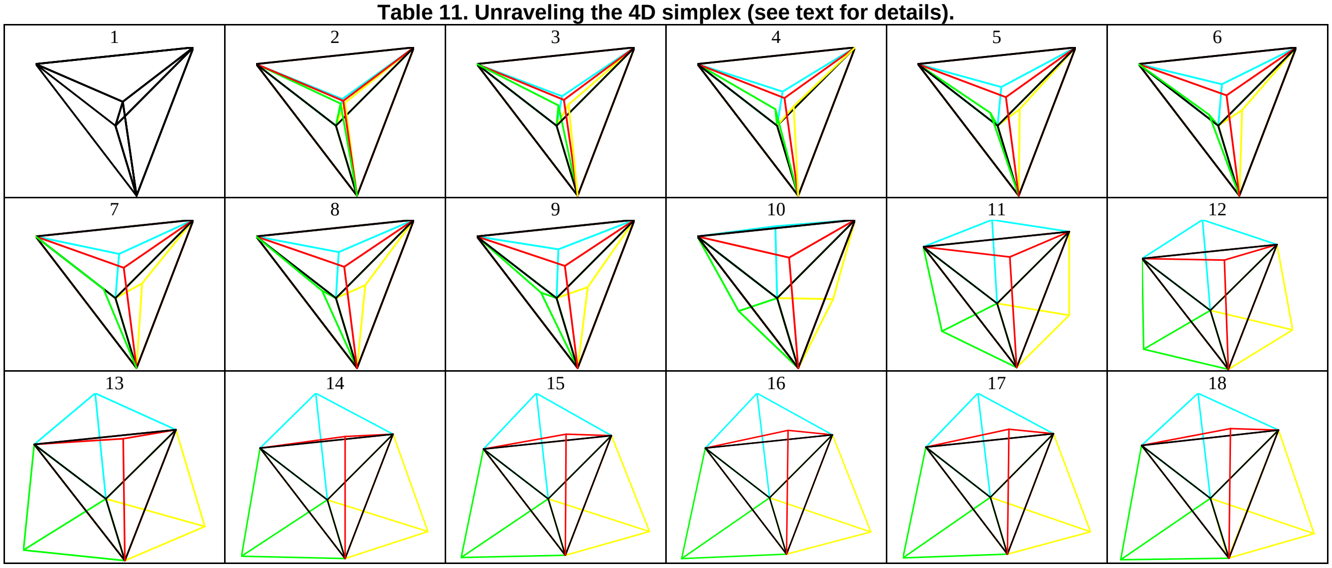
Now, all the transformations to unravel the simplex have been determined. To ravel it back, the same process must be applied in an inverse way but only the angles’ signs for rotations around XY plane must be changed, because the remaining rotations only position the volumes with a face on XY plane.

4.3. Visualizing The 4D Simplex’s Unraveling Process

Table 11 presents some snapshots from the 4D simplex's unraveling sequence. In snapshots 1 to 8, the applied rotations around XY plane are ±0, ±1.043°, ±2.086°, ±4.172°, ±6.258°, ±7.301°, ±8.344° and ±9.387° (the rotation’s sign depends of the adjacent volume). In snapshot 9, the applied rotation around XY plane is ±10.43°; the adjacent volumes look like planes (coinciding with the central volume's faces) --an effect due to the 4D-3D projection. In snapshots 10 to 18, the applied rotations around XY plane are ±20.43, ±30.43, ±40.43, ±50.43, ±60.43, ±70.43, ±80.43, ±90.43 and ±104.3.

At the present time, the results of this research are used with efficiency as didactic material in the Universidad de las Américas - Puebla, México.

Table 11. Unraveling the 4D simplex (see text for details).



5. Future Work

5.1. The n-Dimensional Hyper-Tesseract

Observing the unravelings for the square (a 2D cube), the cube and the 4D hypercube and the fact a nD parallelotopes-family share analogous properties [4] we can generalize the ***n-dimensional hyper-tesseract*** (n≥1) as the result of the (n+1)-dimensional parallelotope's unraveling with the following properties:

- The (n+1)-dimensional hypercube will have 2(n+1) n-dimensional cells on its boundary [3].
- A central cell will be static during the unraveling/raveling process.
- 2(n+1)-2 cells are adjacent to central cell. All of them will share a (n-1)-dimensional cell with central cell.
- A satellite cell won't be adjacent to central cell because their supporting hyperplanes are parallel. It will be adjacent to any of the adjacent cells (it will share a (n-1)-dimensional cell with the selected adjacent cell).
- All the adjacent cells and satellite cell during the unraveling/raveling process will rotate ±90° around the supporting hyperplane of the (n-1)-dimensional shared cells.

For example, the 4D hyper-tesseract is the result of the 5D hypercube's unraveling. The 4D hyper-tesseract will be composed by 10 hypervolumes, where one of them will be the central hypervolume (static), eight of them are adjacent to central hypervolume (they share a volume) and the last one will be the satellite hypervolume (it shares a volume with any of the adjacent hypervolumes). See Figure 10. The adjacent hypervolumes and the satellite hypervolume will rotate around a volume or a hyperplane during the unraveling/raveling process.

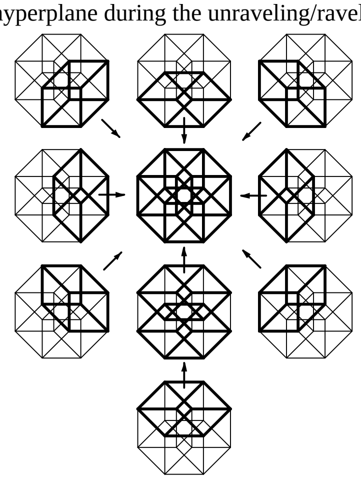


Figure 10. The possible adjacency relations between the hyper-tesseract's central hypervolume and adjacent hypervolumes.

5.2. The Stellated n-Dimensional Simplex

Analyzing the unravelings for the triangle (a 2D simplex), the tetrahedron (a 3D simplex) and the 4D simplex and the fact a nD simplexes-family share analogous properties [4], we can generalize the ***stellated n-dimensional simplex*** (n≥1) as the result of the (n+1)-dimensional simplex's unraveling with the following properties:

- The (n+1)-dimensional simplex will have (n+2) n-dimensional cells on its boundary.
- A central cell will be static during the unraveling/raveling process.
- (n+1) cells are adjacent to central cell. All of them will share a (n-1)-dimensional cell with central cell.
- All the adjacent cells during the unraveling/raveling process will rotate the supplement of the simplex's dihedral angle around the supporting hyperplane of the (n-1)-dimensional shared cells.

6. Conclusions

In this research we found methods to unravel the hypercube and the 4D simplex. Also, we have proposed a generalization to describe the properties of the n-dimensional hyper-tesseract, the result of the (n+1)-dimensional parallelotope's unraveling and the stellated n-dimensional simplex, the result of the (n+1)-dimensional simplex's unraveling. For the 5D space the rotations will be around a volume, for the 6D space they will be around a hypervolume and so forth. This is the direction to follow in our research to get the needed parameters to unravel the 5D hypercube and simplex and to obtain the 4D hyper-tesseract and the stellated 4D simplex. Also, another direction to follow will be related to rotations around arbitrary planes in the 4D space (analogously to rotations around an arbitrary axis in the 3D space). Finding the procedures to rotate around arbitrary planes, the hypercube and simplex's position may not be relevant.

Acknowledgements

We thank the support of the "Consejo Nacional de Ciencia y Tecnología" (CONACyT), México (Project W35804-A).

References

[1] Abbott, E. A. Flatland: A Romance of Many Dimensions, New American Library, 1984.

[2] Aguilera Ramírez, Antonio & Pérez Aguila, Ricardo. Classifying the n-2 dimensional elements as Manifold or Non-Manifold for n-Dimensional Orthogonal Pseudo-Polytopes. Proceedings of the 12th International Conference on Electronics, Communications and Computers CONIELECOMP 2002, pp. 59-63. February 25 to 27, 2002. Acapulco, Guerrero, México.

[3] Banchoff, Thomas F. Beyond the Third Dimension, Scientific American Library, 1996.

[4] Coxeter, H.S.M. Regular Polytopes, Dover Publications, Inc., New York, 1963.

[5] Coxeter, H.S.M. Fundamentos de Geometría, Editorial Limusa, 1984.

[6] Hilbert, D. & Cohn-Vossen, S. Geometry and the Imagination, Chelsea Publishing Company, 1952.

[7] Hollasch, S. R. Four-Space Visualization of 4D Objects, Arizona State University, 1991, Thesis for the Master of Science Degree.

[8] Kaku, M. Hyperspace: A Scientific Odyssey Through Parallel Universes, Time Warps, and the Tenth Dimension, Oxford University Press, 1994.

[9] Pérez Aguila, Ricardo. 4D Orthogonal Polytopes. B.S. Thesis. Universidad de las Américas– Puebla, 2001.

[10] Robbin, T. Fourfield: Computers, Art & The 4th Dimension, Bulfinch Press, 1992.

[11] Rucker, R.V.B. Geometry, Relativity and the Fourth Dimension, Dover Publications, Inc., New York, 1977.

[12] Rucker, R. The Fourth Dimension, Houghton Mifflin Company, Boston, 1984.

Classifying Edges and Faces as Manifold or Non-Manifold Elements in 4D Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila
sp104378@mail.udlap.mx

Antonio Aguilera Ramírez
aguilera@mail.udlap.mx

Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

Abstract

This article presents our experimental results for classifying edges and faces as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes (4D-OPP's). For faces in 4D-OPP's we propose a condition to classify them as manifold or non-manifold. For the edges' analysis in 4D-OPP's we have developed two approaches: 1) The analogy between incident (manifold and non-manifold) edges to a vertex in 3D Orthogonal Pseudo-Polyhedra (3D-OPP's) with incident (manifold and non-manifold) faces to a edge in 4D-OPP's; and 2) The extension of the concept of "cones of faces" (which is applied for classifying a vertex in 3D-OPP's as manifold or non-manifold) to "hypercones of volumes" for classifying an edge as manifold or non-manifold in 4D-OPP's. Both approaches have provided the same results, which present that there are eight types of edges in 4D-OPP's. Finally, the generalizations for classifying the n-3 and the n-2 dimensional boundary elements for n-dimensional Orthogonal Pseudo-Polytopes as manifold or non-manifold elements is also presented.

Posters Papers Proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2003. ISBN: 80-903100-2-8. February 3 to 7, 2003. Plzen, Czech Republic.

Classifying Edges and Faces as Manifold or Non-Manifold Elements in 4D Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila Antonio Aguilera Ramírez
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir. Phone: +52 (222) 229-2664
México 72820, Cholula, Puebla.
sp104378@mail.udlap.mx aguilera@mail.udlap.mx

ABSTRACT

This article presents our experimental results for classifying edges and faces as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes (4D-OPP's). For faces in 4D-OPP's we propose a condition to classify them as manifold or non-manifold. For the edges' analysis in 4D-OPP's we have developed two approaches: 1) The analogy between incident (manifold and non-manifold) edges to a vertex in 3D Orthogonal Pseudo-Polyhedra (3D-OPP's) with incident (manifold and non-manifold) faces to a edge in 4D-OPP's; and 2) The extension of the concept of "cones of faces" (which is applied for classifying a vertex in 3D-OPP's as manifold or non-manifold) to "hypercones of volumes" for classifying an edge as manifold or non-manifold in 4D-OPP's. Both approaches have provided the same results, which present that there are eight types of edges in 4D-OPP's. Finally, the generalizations for classifying the n-3 and the n-2 dimensional boundary elements for n-dimensional Orthogonal Pseudo-Polytopes as manifold or non-manifold elements is also presented.

Keywords: Computational geometry, Geometric interrogations and reasoning, Geometric and topological representations.

1. INTRODUCTION

Recent interest has been growing in studying multidimensional polytopes (4D and beyond) for representing phenomena in n-dimensional spaces. Some examples include the works described in [Fei90], [Weg97] and [Lee99]. These previous works show how some of these phenomena's features rely on the polytopes' geometric and topologic relations. However, due to the need of visualizing and analyzing these phenomena (i.e. multidimensional data), it is essential first to analyze these polytopes and their boundaries that compose them [Her98]. So, this article covers that first step, in our research, with the boundary's analysis for classifying edges and faces as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes.

2. THE 4D ORTHOGONAL POLYTOPES

[Cox63] defines an Euclidean polytope Π_n as a finite region of n-dimensional space enclosed by a finite number of (n-1)-dimensional hyperplanes. The finiteness of the region implies that the number N_{n-1} of bounding hyperplanes satisfies the inequality $N_{n-1}>n$. The part of the polytope that lies on one of these hyperplanes is called a cell. Each cell of a Π_n is an (n-1)-dimensional polytope, Π_{n-1}. The cells of a Π_{n-1} are Π_{n-2}'s, and so on; we thus obtain a descending sequence of elements Π_{n-3}, Π_{n-4},..., Π_1 (an edge), Π_0 (a vertex).

Orthogonal Polyhedra (3D-OP) are defined as polyhedra with all their edges (Π_1's) and faces (Π_2's) oriented in three orthogonal directions ([Jua88] & [Pre85]). Orthogonal Pseudo-Polyhedra (3D-OPP) will refer to regular and orthogonal polyhedra with non-manifold boundary [Agu98].

Similarly, **4D Orthogonal Polytopes (4D-OP)** are defined as 4D polytopes with all their edges (Π_1's), faces (Π_2's) and volumes (Π_3's) oriented in four orthogonal directions and **4D Orthogonal Pseudo-Polytopes (4D-OPP)** will refer to 4D regular and orthogonal polytopes with non-manifold boundary. Because the 4D-OPP's definition is an extension from the 3D-OPP's, is easy to generalize the concept to define **n-dimensional Orthogonal Polytopes (nD-OP)** as n-dimensional polytopes with all their Π_{n-1}'s, Π_{n-2}'s,..., Π_1's oriented in n orthogonal directions. Finally, **n-dimensional Orthogonal Pseudo-Polytopes (nD-OPP)** are defined as n-dimensional regular and orthogonal polytopes with non-manifold boundary.

WSCG POSTERS proceedings
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

3. THE Π_{n-2} ANALYSIS FOR 2D, 3D AND 4D-OPP'S

The Π_0 Analysis for 2D-OPP's

A set of quasi-disjoint rectangles determines a 2D-OPP whose vertices must coincide with some of the rectangles' vertices [Agu98]. Each of these rectangles' vertices can be considered as the origin of a 2D local coordinate system, and they may belong to up to four rectangles, one for each local quadrant. The two possible adjacency relations between the four possible rectangles can be of edge or vertex. There are $2^4 = 16$ possible combinations which, by applying symmetries and rotations, may be grouped into six equivalence classes, also called configurations [Sri81].

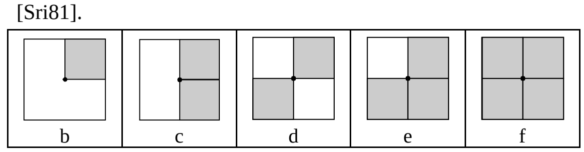


Table 1. The 2D configurations with all their rectangles incident to the origin.

Because we are interested in the vertex analysis, we will consider only those configurations where all their rectangles are incident to the origin. According to the configurations' nomenclature presented in [Agu98], the studied configurations are b, c, d, e and f (see Table 1). There are only two types of vertices in the 2D-OPP's: **the manifold vertex with two incident edges** (configurations b and e), and **the non-manifold vertex with four incident edges** (configuration d) [Agu98]. The remaining configurations represent no vertex because configuration c has only two incident and collinear edges, and in configuration f there are no incident edges.

The Π_1 Analysis for 3D-OPP's

A set of quasi-disjoint boxes determines a 3D-OPP whose vertices must coincide with some of the boxes' vertices [Agu98]. Each of these boxes' vertices can be considered as the origin of a 3D local coordinate system, and they may belong to up to eight boxes, one for each local octant. There are $2^8 = 256$ possible combinations which, by applying symmetries and rotations, may be grouped into 22 equivalence classes [Lor87], also called configurations [Sri81]. Each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class [Agu98]. Grouping complementary configurations leads to the 14 major cases [Van94]. The configurations with 5, 6, 7 and 8 surrounding boxes are complementary, and thus analogous, to combinations with 3, 2, 1 and 0 surrounding boxes, respectively [Agu98]. Finally, each configuration, with four surrounding boxes is self-complementary.

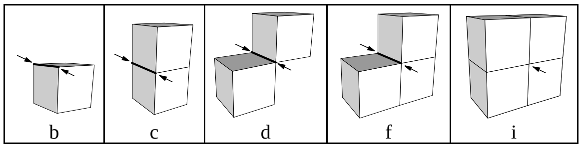


Table 2. The 3D configurations where all their boxes are incident to a same edge (the arrows show the analyzed edge).

Because we are interested in the edge analysis, we will consider only those configurations where all their boxes are incident to a same edge. According to the configurations' associated nomenclature presented in [Agu98], the studied configurations are b, c, d, f and i (see Table 2). [Agu98] concluded that there are only two types of edges in the 3D-OPP's:

- **The manifold edge with two incident faces.** This type of edges is found in configurations b and f. The edge's two incident faces in configuration b belong to one cube's boundary and they are perpendicular to each other. The edge's two incident faces in configuration f belong to two different cubes with edge adjacency and they result perpendicular to each other.
- **The non-manifold edge with four incident faces**. This type of edges is found in configuration d, where two of its faces belongs to a cube and the remaining belong to a second cube with edge adjacency.
- The remaining configurations represent no edge because in configuration c there are only two incident and coplanar faces, and in configuration i there are no incident faces.

The Π_2 Analysis For 4D-OPP's

A set of quasi-disjoint *hyper-boxes* (i.e., hypercubes, which in this paper will be represented using Claude Bragdon's projection [Ruc84]) determines a 4D-OPP whose vertices must coincide with some of the *hyper-boxes'* vertices. We will consider the *hyper-boxes'* vertices as the origin of a 4D local coordinate system, and they may belong to up to 16 *hyper-boxes*, one for each local *hyper-octant*. The 4D-OPP's vertices are determined according to the presence of absence of each of these 16 surrounding *hyper-boxes*. The four possible adjacency relations between the 16 possible *hyper-boxes* can be of volume, face, edge or vertex. There are 2^{16}=65,536 possible combinations of vertices in 4D-OPP's which can be grouped, applying symmetries and rotations, into 253 equivalence classes, also called configurations [Pér01]. Each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class. Grouping complementary configurations leads to the 145 major cases [Pér01]. The combinations with 9, 10, 11, 12, 13, 14, 15 and 16 surrounding hyper-boxes are complementary, and thus analogous, to combinations with 7, 6, 5, 4, 3, 2, 1 and 0 surrounding hyper-boxes, respectively. Finally, each configuration, with eight surrounding hyper-boxes is self-complementary [Pér01].

We will consider only those configurations whose *hyper-boxes* are incident to a same face. According to the configurations' associated nomenclature presented in [Pér01], the studied configurations are 2, 3, 4, 7 and 13 (Table 3). In [Pér01] is concluded that there are only two types of faces in the 4D-OPP's:

- **The manifold faces with two incident volumes.** The face's two incident volumes in configuration 2 belong to the boundary of only one hypercube and they are perpendicular to each other. While in configuration 7, The face's two incident volumes belong to two different hypercubes with face adjacency and they result perpendicular to each other.
- **The non-manifold faces with four incident volumes.** This type of faces is found in configuration 4, where two of its incident volumes belongs to a hypercube and the remaining two belong to a second hypercube with face adjacency.
- The remaining configurations represent no face because in configuration 3 there are only two incident and *co-hyperplanar* volumes, and in configuration 13 there are no incident volumes (analogous to 3D configurations c and i in Table 2).

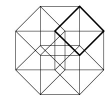
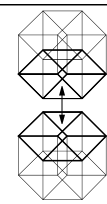
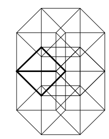
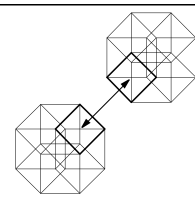
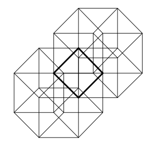
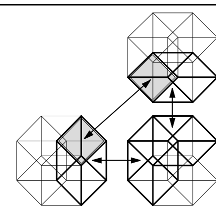
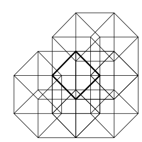
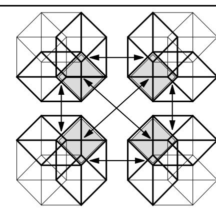
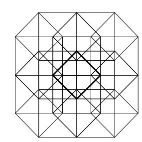
Adjacencies between hyper-boxes	**Configuration**
	 **2**
	 **3**
	 **4**
	 **7**
	 **13**

Table 3. Configurations 2, 3, 4, 7 and 13 for 4D-OPP's

Classifying the Π_{n-2}'s in nD-OPP's

Finally, the generalized conditions to classify a Π_{n-2} as manifold or non-manifold in a nD-OPP are:

- If two perpendicular Π_{n-1}'s are incident to a Π_{n-2} then it must be classified as **manifold**.
- If four Π_{n-1}'s are incident to a Π_{n-2} then it must be classified as **non-manifold**.

4. THE Π_{n-3} ANALYSIS FOR 3D AND 4D-OPP'S

The Π_0 Analysis for 3D-OPP's

There are eight types of vertices (also two non valid vertices are identified) for 3D-OPP's [Agu98]. These vertices can be classified depending on the number of two-manifold and non-manifold edges incident to them and they are referred as V3, V4, V4N1, V4N2, V5N, V6, V6N1 and V6N2 [Agu98] (Table 4). In this nomenclature "V" means vertex, the first digit shows the number of incident edges, the "N" is present if at least one non-manifold edge is incident to the vertex and the second digit is included to distinguish between two different types that otherwise could receive the same name.

Each vertex has the following properties [Agu98]:

- V3: all three incident edges are two-manifold and perpendicular to each other.

- V4: all four incident edges are two-manifold, they lie on a plane, and can be grouped in two couples of collinear edges.
- V4N1: three of its four incident edges are perpendicular to each other and also two-manifold ones, while the fourth is non-manifold and collinear to one of the other three.
- V4N2: two of its four incident edges are two-manifold and collinear, while each of its other two is non-manifold and perpendicular to the other three.
- V5N: four of its five incident edges are two-manifold and lie in a plane, while the fifth is non-manifold and perpendicular to the rest of them.
- V6: all six incident edges are two-manifold.
- V6N1: three of its six incident edges are perpendicular to each other and also two-manifold ones, while each of its remaining three edges is non-manifold and collinear to one of the first three.
- V6N2: all of its six incident edges are non-manifold.
- Non valid vertex 1: its two manifold edges are collinear.
- Non valid vertex 2: its two non-manifold edges are collinear.

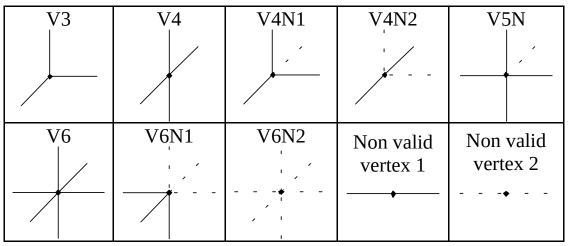


Table 4. Vertices present in 3D-OPP's (dotted lines indicate non-manifold edges and continuous lines indicate manifold edges).

The Π_1 Analysis for 4D-OPP's

Vertices can be defined in terms of the manifold or non-manifold edges that are incident to these vertices in 3D-OPP's [Agu98]. The same process will be extended to describe edges in terms of the manifold or non-manifold faces that are incident to those edges in 4D-OPP's. In this way, we have identified eight types of edges and two non valid edges. We will also extend the nomenclature used by [Agu98] to describe them. Such edges will be referred as E3, E4, E4N1, E4N2, E5N, E6, E6N1 and E6N2 (Table 5). The only difference with the nomenclature used to describe the vertices is that "E" means edge instead of "V" that means vertex. Each edge has the following properties:

- E3: all three incident faces are two-manifold and perpendicular to each other.
- E4: all four incident faces are manifold and lie on a hyperplane, and they can be grouped in two couples of coplanar faces.
- E4N1: three of its four incident faces are perpendicular to each other and also two-manifold ones, while the fourth is non-manifold and coplanar to one of the other three.
- E4N2: two of its four incident faces are two-manifold and coplanar, while each of its other two is non-manifold and perpendicular to the other three.
- E5N: four of its five incident faces are two-manifold and lie in a hyperplane, while the fifth is non-manifold and perpendicular to the rest of them.
- E6: all six incident faces are two-manifold.
- E6N1: three of its six incident manifold faces are perpendicular to each other, while each of its remaining three faces is non-manifold and coplanar to one of the first three.
- E6N2: all of its six incident faces are non-manifold.
- Non valid edge 1: its two manifold faces are coplanar.
- Non valid edge 2: its two non-manifold faces are coplanar.

It results interesting that the number, classifications and positions of the incident faces to an edge in 4D-OPP's are analogous to the way that a set of edges are incident to a vertex in 3D-OPP's.

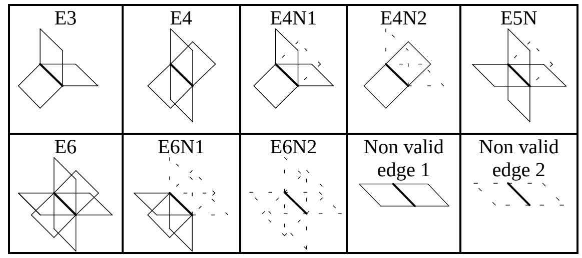


Table 5. Edges present in 4D-OPP's (dotted lines indicate non-manifold faces and continuous lines indicate manifold faces).

Classifying the Π_0's in Polyhedra Through its Cones of Faces

A **polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite set of plane polygons such that every edge of a polygon is shared by exactly one other polygon (adjacent polygons) [Pre85]. A **pseudo-polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite collection of planar faces such that every edge has at least two adjacent faces, and if any two faces meet, they meet at a common edge [Tan91]. Edges and vertices, as boundary elements for polyhedra, may be either two-manifold (or just manifold) or non-manifold elements. In the case of edges, they are (non) manifold elements when every points of it is also a (non) manifold point, except that either or both of its ending vertices might be a point of the opposite type [Agu98]. A manifold edge is adjacent to exactly two faces, and a manifold vertex is the apex (i.e., the common vertex) of only one **cone of faces**. Conversely, a non-manifold edge is adjacent to more than two faces, and a non-manifold vertex is the apex (i.e., the common vertex) of more than one **cone of faces** [Ros91].

3D vertex	Classification
V3	Manifold
V4	Manifold
V4N1	Non-manifold
V4N2	Non-manifold
V5N	Non-manifold
V6	Non-manifold or manifold according to its geometric context.
V6N1	Non-manifold
V6N2	Non-manifold

Table 6. 3D-OPP's vertices classification.

Using the concept of cones of faces it is easy to construct an algorithm to determine the classification of a vertex as manifold or non-manifold in any polyhedron or pseudo-polyhedron. Using this algorithm over the possible vertices in 3D-OPP's we have the results presented in Table 6 which coincide with those presented by [Agu98].

Classifying the Π_1's in 4D Polytopes Through its Hyper-Cones of Volumes

Due to the analogy between 3D-OPP's vertices described in terms of their incident manifold or non-manifold edges, and 4D-OPP's edges described in terms of their incident manifold or non-manifold faces, the next logical step is to extend the concept of cones of faces presented in the previous section to classify 4D polytopes' edges as manifold or non-manifold.

Faces, edges and vertices, as boundary elements for 4D polytopes, may be either manifold or non-manifold elements. [Cox63] and [Han93] have stated that a manifold face is adjacent to exactly two volumes, and now we suggest that a manifold edge is the common edge (apex) of only one **hyper-cone of volumes**. Conversely, we have suggested that a non-manifold face is adjacent to more than two volumes, and now we suggest that a non-manifold edge is the common edge (apex) of more than one **hyper-cone of volumes**.

Using the concept of hyper-cones of volumes, it is easy to extend the algorithm for obtaining the vertex classification for 3D-OPP's used for previous section, to allow us classifying an edge, as manifold or non-manifold, in any 4D polytope or 4D pseudo-polytope. The algorithm is defined with the following steps:

1 Get the set of Π_3's that are incident to edge *A* (a Π_1).
2 From the set of Π_3's select one of them.
3 The selected Π_3 has two Π_2's that are incident to *A*, get one of them and label it as *START* and *ANOTHER*.
4 Repeat
4.1 If the number of Π_3's to *ANOTHER* is more than one, then A is a non-manifold Π_1. End.
4.2 The *ANOTHER* Π_2 is common to another Π_3, find it.
4.3 The Π_3 has another Π_2 that is common to *A*, find it and label it as *ANOTHER*.
4.4 Until *START* = *ANOTHER* (it has been found a hyper-cone of volumes).
5 If there are more Π_3's to analyze then *A* is non-manifold (there are more hyper-cones of volumes). End.
6 Otherwise, *A* is manifold (*A* is the common edge of only one hyper-cone of volumes). End.

5. RESULTS

Using the algorithm presented in the previous section over the possible edges in 4D-OPP's we have that the edges' classifications are analogous to the 3D-OPP's vertices' classifications. Table 7 shows the edges' classifications given by the extended algorithm and their analogous 3D results.

4D edge	Classification through hyper-cones of volumes	3D vertex	Classification through cones of faces
E3	Manifold	V3	Manifold
E4	Manifold	V4	Manifold
E4N1	Non-manifold	V4N1	Non-manifold
E4N2	Non-manifold	V4N2	Non-manifold
E5N	Non-manifold	V5N	Non-manifold
E6	Non-manifold or manifold according to its geometric context.	V6	Non-manifold or manifold according to its geometric context.
E6N1	Non-manifold	V6N1	Non-manifold
E6N2	Non-manifold	V6N2	Non-manifold

Table 7. 4D-OPP's edges classifications and their analogy with 3D-OPP's vertices.

Classifying the Π_{n-3} in nD Polytopes Through its nD Hyper-Cones of Π_{n-1}'s

Due to the analogy found between 3D vertices and 4D edges with the extension of the concept of cones of faces, is feasible to generalize the last presented algorithm to classify the Π_{n-3} as manifold or non-manifold in nD polytopes through their nD hyper-cones of Π_{n-1}'s. The proposed general algorithm is the following:

1 Get the set of Π_{n-1}'s that are incident to Π_{n-3} *A*.
2 From the set of Π_{n-1}'s select one of them.
3 The selected Π_{n-1} has two Π_{n-2}'s that are incident to Π_{n-3} A, get one of them and label it as *START* and *ANOTHER*.
4 Repeat
4.1 If the number of incident Π_{n-1}'s to *ANOTHER* is more than one, then *A* is a non-manifold Π_{n-3}.
4.2 The *ANOTHER* Π_{n-2} is common to another Π_{n-1}, find it.
4.3 The Π_{n-1} has another Π_{n-2} that is common to *A*, find it and label it as *ANOTHER*.
4.4 Until *START* = *ANOTHER* (it has been found a nD hyper-cone of Π_{n-1}'s).
5 If there are more Π_{n-1}'s to analyze then Π_{n-3} *A* is non-manifold (there are more nD hyper-cones of Π_{n-1}'s).
6 Otherwise, Π_{n-3} *A* is manifold (A is the common Π_{n-3} of only one nD hyper-cone of Π_{n-1}'s).

The Eight Types of Π_{n-3}'s in nD Orthogonal Pseudo-Polytopes

Due to the analogy between vertices in 3D-OPP's and edges in 4D-OPP's (Table 7), we can extend their properties to propose the eight types of Π_{n-3}'s in nD-OPP's. Such Π_{n-3}'s will be referred as Π_{n-3}3, Π_{n-3}4, Π_{n-3}4N1, Π_{n-3}4N2, Π_{n-3}5N, Π_{n-3}6, Π_{n-3}6N1 and Π_{n-3}6N2. In this nomenclature 'Π_{n-3}' indicates the (n-3)-dimensional element (i.e. vertices in 3D-OPP's and edges in 4D-OPP's), the first digit shows the number of incident Π_{n-2} (i.e. edges in 3D-OPP's and faces in 4D-OPP's), the 'N' is present if at least one non-manifold Π_{n-2} is incident to the Π_{n-3} and the second digit is included to distinguish between two different types that otherwise could receive the same name.

6. FUTURE WORK

The results of this article are being used in studying the extension for the Extreme Vertices Model (EVM) [Agu98] to the four dimensional space (EVM-4D). The EVM-4D will be a representation model for 4D-OPP's that will allow queries and operations over them. However, the fact related to a model purely geometric (four geometric dimensions) is not restrictive for our research, because it will be used under geometries as the 4D spacetime. The first main application for the EVM-4D will cover the visualization and analysis for multidimensional data under the context of a Geographical Information System (GIS).

7. ACKNOWLEDGEMENTS

This work is supported by the "Consejo Nacional de Ciencia y Tecnología" (CONACyT), México (Project W35804-A).

8. REFERENCES

[Agu98] Aguilera,A. Orthogonal Polyhedra: Study and Application. PhD Thesis, Universitat Politècnica de Catalunya. 1998.

[Cox63] Coxeter,H.S.M. Regular Polytopes. Dover, 1963.

[Fei90] Feiner,S., & Beshers,C. Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds. In Proceedings ACM Symposium on User Interface Software and Technology, ACM Press / Addison-Wesley, 76-83. 1990.

[Han93] Hansen,H., & Christensen,N.J. A model for n-dimensional boundary topology. In Proceedings Second Symposium on Solid Modeling and Applications, 65-73. 1993.

[Her98] Herman,G. Geometry of digital spaces. Birkhäuser, 1998.

[Jua88] Juan-Arinyo,R. Aportació a l'Estudi de la Transcodificació dels Models de Fronteres i Arbres Estesos en Model de Geometria Constructiva de Sòlids. PhD Thesis (written in Catalan). Universitat Politècnica de Catalunya. 1988.

[Lee99] Lees,J.M. Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences. Computers & Geosciences 26, 7, 751-761. 1999.

[Lor87] Lorensen,W. E., & Cline,H. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics 21, 4, 44-50. 1987.

[Pér01] Pérez,R. 4D Orthogonal Polytopes. B.Sc. Thesis, Universidad de las Américas - Puebla. 2001.

[Pre85] Preparata F. P., and Shamos, M. I. Computational Geometry: an Introduction. Springer-Verlag. 1985.

[Ros91] Rossignac,J. & Requicha,A. Constructive Non-Regularized Geometry. Computer Aided Design 23,1,21-32. 1991.

[Ruc84] Rucker,R. The fourth dimension. Houghton Mifflin, 1984.

[Sri81] Srihari,S.N. Representation of Three-Dimensional Digital Images. ACM Computing Surveys 13, 1, 399-424. 1981.

[Tan91] Tang,K., & Woo,T. Algorithmic Aspects of Alternating Sum of Volumes. Part 2: Non-Convergence and Its Remedy. Computer-Aided Design 23, 6, 435-443. 1991.

[Van94] Van Gelder, A., Wilhelms, J. Topological Considerations in Isosurface Generation. ACM Transactions on Graphics 13, 4, 337-375. 1994.

[Weg97] Wegenkittl,R., Löffelmann,H., & Gröller,E. Visualizing the Behavior of Higher Dimensional Dynamical Systems. In Proceedings of IEEE Visualization 1997, 119-125. 1997.

Presenting the 'Test-Box' Heuristic for Determining the Configurations for the n-Dimensional Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila
sp104378@mail.udlap.mx
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.

Abstract

This article presents the "Test-Box" heuristic that gives a solution to the problem of determining the configurations that can represent the n-Dimensional Orthogonal Pseudo-Polytopes. This heuristic presents better performance that the procedure through exhaustive searching. It has as a fundament the extrusion of (n-1)-dimensional configurations to obtain n-dimensional configurations.

Proceedings of the 13th International Conference on Electronics, Communications and Computers CONIELECOMP 2003, pp. 64-69. Editors: Báez López, J.M.D. & Murphy A., R. February 24 to 26, 2003. Cholula, Puebla, México.

Presenting the 'Test-Box' Heuristic for Determining the Configurations for the n-Dimensional Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila
sp104378@mail.udlap.mx
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.

Abstract

This article presents the "Test-Box" heuristic that gives a solution to the problem of determining the configurations that can represent the n-Dimensional Orthogonal Pseudo-Polytopes. This heuristic presents better performance that the procedure through exhaustive searching. It has as a fundament the extrusion of (n-1)-dimensional configurations to obtain n-dimensional configurations.

1. Introduction

Recent interest has been growing in studying multidimensional polytopes (4D and beyond) for representing multidimensional phenomena in the Euclidean n-dimensional space. Some of these phenomena's features rely on the polytopes' geometric and topologic relations [Aguilera,03]. However, due to the need of visualizing and analyzing these phenomena (i.e. multidimensional data), it is essential first to analyze these polytopes and their boundaries that compose them [Herman,98]. Therefore, this article covers the analysis for obtaining the configurations that can represent the n-Dimensional Orthogonal Pseudo-Polytopes. Moreover, it is presented a heuristic that gives a solution for determining those configurations in 5D space and beyond.

2. The n-Dimensional Orthogonal Pseudo-Polytopes

[Coxeter,63] defines an Euclidean polytope Π_n as a finite region of n-dimensional space enclosed by a finite number of (n-1)-dimensional hyperplanes. The finiteness of the region implies that the number N_{n-1} of bounding hyperplanes satisfies the inequality $N_{n-1}>n$. The part of the polytope that lies on one of these hyperplanes is called a cell. Each cell of a Π_n is an (n-1)-dimensional polytope, Π_{n-1}. The cells of a Π_{n-1} are Π_{n-2}'s, and so on; we thus obtain a descending sequence of elements Π_{n-3}, Π_{n-4},…, Π_1 (an edge), Π_0 (a vertex).

We know that a Π_3 (a 3D Euclidean polytope) is a polyhedron. The polyhedron's cells are Π_2. A Π_2 (a 2D Euclidean polytope) is a polygon. The polygon's cells are Π_1. A Π_1 (a 1D Euclidean polytope) is a segment. Finally, the segment's cells are Π_0, a set of vertices. The cells of a Π_4 (a 4D Euclidean polytope) are Π_3 (polyhedra, also called volumes in the context of Π_4).

Orthogonal Polyhedra (3D-OP) are defined as polyhedra with all their edges and faces oriented in three orthogonal directions ([Preparata,85] & [Juan-Arinyo,88]). Orthogonal Pseudo-Polyhedra (3D-OPP) will refer to regular and orthogonal polyhedra with non-manifold boundary [Aguilera,98].

Similarly, **4D Orthogonal Polytopes (4D-OP)** are defined as 4D polytopes with all their edges, faces and volumes oriented in four orthogonal directions and **4D Orthogonal Pseudo-Polytopes (4D-OPP)** will refer to 4D regular and orthogonal polytopes with non-manifold boundary [Aguilera,02]. Because the 4D-OPP's definition is an extension from the 3D-OPP's, is easy to generalize the concept to define **n-dimensional Orthogonal Polytopes (nD-OP)** as n-dimensional polytopes with all their Π_{n-1}, Π_{n-2},…, Π_1 oriented in n orthogonal directions. Finally, **n-dimensional Orthogonal Pseudo-Polytopes (nD-OPP)** are defined as n-dimensional regular and orthogonal polytopes with non-manifold boundary [Aguilera,02].

3. Configurations for 1D, 2D, 3D and 4D-OPP's.

3.1. Configurations for Segments in 1D Space

Although it is a trivial case, we will present the three possible configurations in 1D space (table 1). They will be usefulness when proposing the "Test-Box" heuristic.

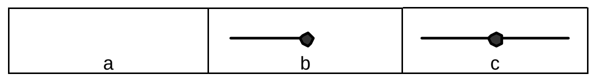


Table 1. The posible configurations (a-c) in 1D space.

We have the configuration a with 0 surrounding segments, which is complementary to configuration c with two surrounding segments. Configuration b with just one segment is autocomplementary [Aguilera,98].

3.2 Configurations for 2D-OPP's.

A set of quasi-disjoint rectangles determines a 2D-OPP whose vertices must coincide with some of the rectangles' vertices [Aguilera,98]. Each of these rectangles' vertices can be considered as the origin of a 2D local coordinate system, and they may belong to up to four rectangles, one for each local quadrant. The two possible adjacency relations between the four possible rectangles can be of edge or vertex. There are 2^4 = 16 possible combinations which, by applying rotational symmetries, may be grouped into six equivalence classes, also called configurations [Srihari,81] (table 2). Moreover, each possible combination has its complementary combination, and each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class [Aguilera,98].

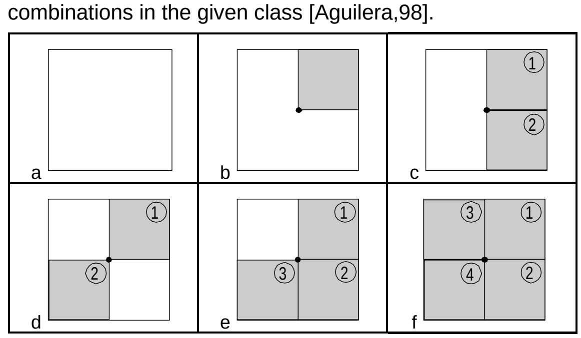


Table 2. Possible configurations (a to f) for 2D-OPP's.

These 16 possible combinations are distributed in the following way [Aguilera,98]:

$$2^4 = \sum_{k=0}^{4} C\binom{4}{k} = 1+4+6+4+1 = 16$$

And using combinatorial analysis, there are:

$C\binom{4}{0} =$ 1 combination with zero surrounding rectangles (configuration a).

$C\binom{4}{1} =$ 4 combinations with one surrounding rectangle (configuration b).

$C\binom{4}{2} =$ 6 combinations with two surrounding rectangles (configurations c and d).

$C\binom{4}{3} =$ 4 combinations with three surrounding rectangles (configuration e).

$C\binom{4}{4} =$ 1 combination with four surrounding rectangles (configuration f).

Configurations a and f, as well as configurations b and e, are complementary to each other. Configurations c and d are self-complementary [Aguilera,98].

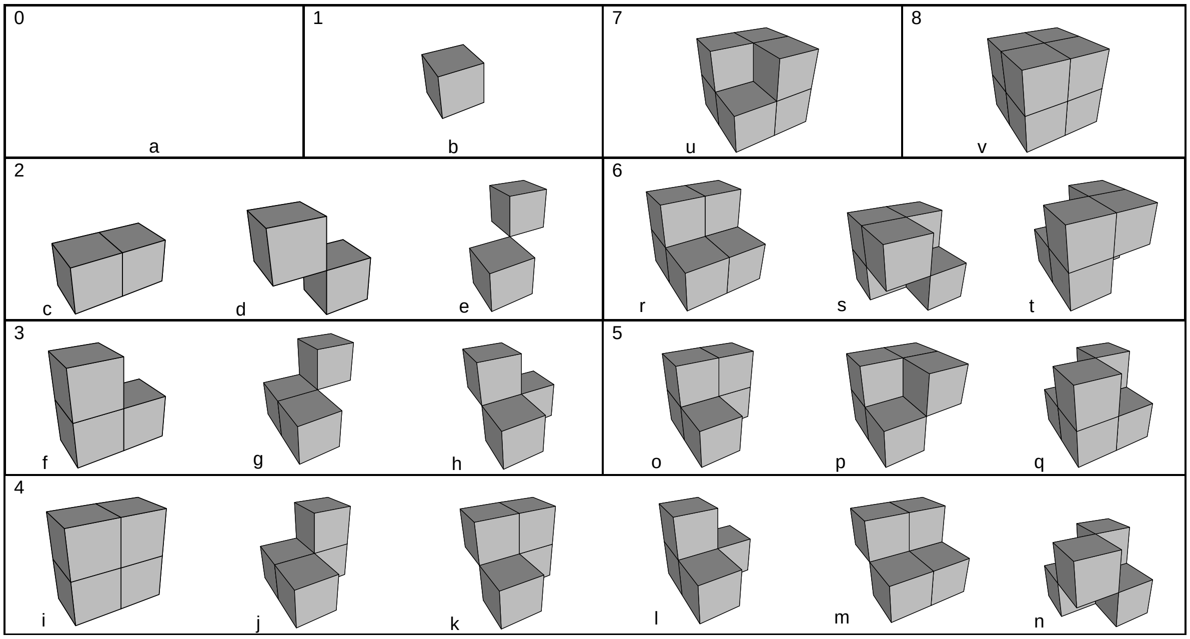


Table 3. Possible configurations (a to v) for 3D-OPP's.

3.3. Configurations for 3D-OPP’s.

A set of quasi-disjoint boxes determines a 3D-OPP whose vertices must coincide with some of the boxes' vertices [Aguilera,98]. Similarly to the 2D case, each of these boxes' vertices can be considered as the origin of a 3D local coordinate system, and they may belong to up to eight boxes, one for each local octant. The three possible adjacency relations between the eight possible boxes can be of face, edge or vertex. There are 2^8 = 256 possible combinations which, by applying rotational symmetries, may be grouped into 22 equivalence classes [Lorensen,87], also called configurations [Srihari,81] (table 3). As in the 2D case, each possible combination has its complementary combination, and each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class [Aguilera,98]. Grouping complementary configurations leads to the 14 major cases [Van Gelder,94].

Similarly to the 2D case, these 256 possible combinations are distributed in the following way [Aguilera,98]:

$$2^8 = \sum_{k=0}^{8} C\binom{8}{k} = 1+8+28+56+70+56+28+8+1 = 256$$

And using combinatorial analysis, there are:

$C\binom{8}{0} =$ 1 combination with zero surrounding boxes (configuration a).

$C\binom{8}{1} =$ 8 combinations with one surrounding box (configuration b).

$C\binom{8}{2} =$ 28 combinations with two surrounding boxes (configurations c, d and e).

$C\binom{8}{3} =$ 56 combinations with three surrounding boxes (configurations f, g and h).

$C\binom{8}{4} =$ 70 combinations with four surrounding boxes (configurations i, j, k, l, m and n).

The remaining combinations with 5, 6, 7 and 8 surrounding boxes are complementary, and thus analogous, to combinations with 3, 2, 1 and 0 surrounding boxes, respectively [Aguilera,98]. Finally, each configuration, with four surrounding boxes is self-complementary.

3.4 Configurations for 4D-OPP’s.

By analogy, it can be assumed that a set of quasi-disjoint *hyper-boxes* (i.e., hypercubes, which in this paper will be represented using Claude Bragdon’s projection [Rucker,77]) determines a 4D-OPP whose vertices must coincide with some of the *hyper-boxes’* vertices. We will consider the *hyper-boxes’* vertices as the origin of a 4D local coordinate system, and they may belong to up to 16 *hyper-boxes*, one for each local *hyper-octant*. The 4D-OPP’s vertices are determined according to the presence of absence of each of these 16 surrounding *hyper-boxes*. The four possible adjacency relations, extending by analogy, between the 16 possible *hyper-boxes* can be of volume, face, edge or vertex. There are 2^{16} = 65,536 possible combinations which can be grouped, applying rotational symmetries, into 253 equivalence classes called configurations [Pérez,01]. Each possible combination has its complementary combination, and each configuration (i.e. each class) has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class. Grouping complementary configurations leads us to the 145 major cases [Pérez,01].

The 65,536 possible combinations are distributed in the following way [Pérez,01]:

$$2^{16}=\sum_{k=0}^{16} C\binom{16}{k} = \left\{\begin{array}{c} 1+16+120+560+1{,}820+4{,}368+ \\ 8{,}008+11{,}440+12{,}870+11{,}440+8{,}008+ \\ 4{,}368+1{,}820+560+120+16+1 \end{array}\right\}$$

$$=65{,}536$$

And using combinatorial analysis, there are:

$C\binom{16}{0}=$ 1 combination with zero surrounding hyper-boxes (configuration 1).

$C\binom{16}{1}=$ 16 combinations with one surrounding hyper-box (configuration 2, shown in table 4).

$C\binom{16}{2}=$ 120 combinations with two surrounding hyper-boxes: configurations 3 (volume adjacency), 4 (face adjacency), 5 (edge adjacency) and 6 (vertex adjacency), shown in table 4.

$C\binom{16}{3}=$ 560 combinations with three surrounding hyper-boxes (configurations 7 to 12, only 7 and 8 shown in table 4).

$C\binom{16}{4}=$ 1,820 combinations with four surrounding hyper-boxes (configurations 13 to 28, only 13 shown in table 4).

$C\binom{16}{5}=$ 4,368 combination with five surrounding hyper-boxes (configurations 29 to 48).

$C\binom{16}{6}=$ 8,008 combinations with six surrounding hyper-boxes (configurations 49 to 78).

$C\binom{16}{7}=$ 11,440 combinations with seven surrounding hyper-boxes (configurations 79 to 108).

$C\binom{16}{8}=$ 12,870 combinations with eight surrounding hyper-boxes (configurations 109 to 145).

The remaining combinations with 9, 10, 11, 12, 13, 14, 15 and 16 surrounding hyper-boxes are complementary, and thus analogous, to combinations with 7, 6, 5, 4, 3, 2, 1 and 0 surrounding hyper-boxes, respectively. Finally, each configuration, with eight surrounding hyper-boxes is self-complementary [Pérez,01].

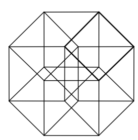
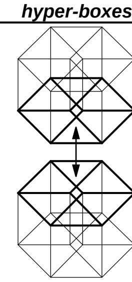
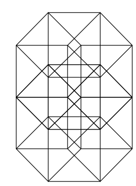
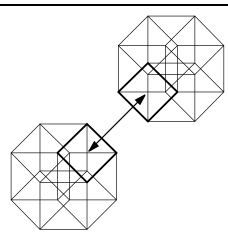
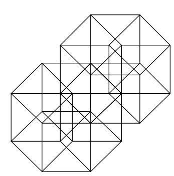
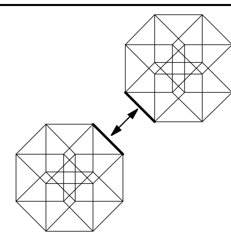
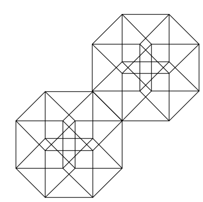
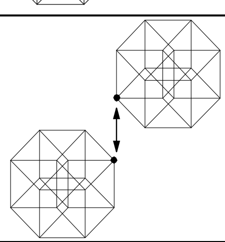
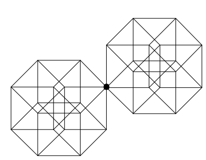
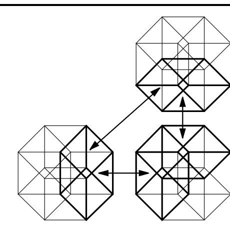
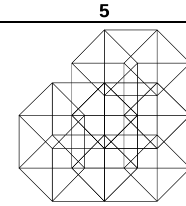
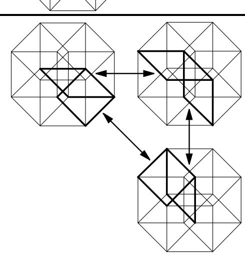
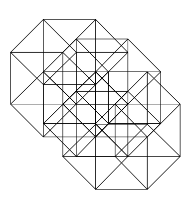
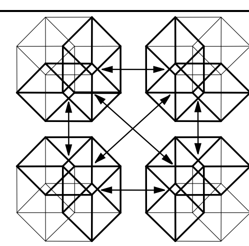
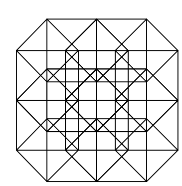
Adjacencies between *hyper-boxes*	**Configuration**	**Adjacencies between *hyper-boxes***	**Configuration**
	 **2**		 **3**
	 **4**		 **5**
	 **6**		 **7**
	 **8**		 **13**

Table 4. Configurations 2-8 and 13 for the 4D-OPP's (each hypercube is show using Bragdon's projection).

4. The Problem of Determining the Configurations for nD-OPP's (n > 4)

For the Euclidean n-Dimensional space we have 2^n possible hyper-octants (4 quadrants for 2D space, 8 octants for 3D space, and 16 hyper-octants for 4D space). This number of hyper-octants has repercurssion over the possible number of combinations of vertices described through the presence of absence of *hyper-boxes* each one in every hyper-octant. In general, we have that the total number of combinations in nD space is [Hill,98]:

$$2^{(2^n)}$$

It was before discussed that in 4D space we have 2^{16} = 65,536 combinations. [Pérez,01] determined that there are 253 configurations for 4D-OPP's through exhaustive searching. However, if we want to determine the configurations for 5D-OPP's through exhaustive searching, we would have to consider that there are 32 hyper-octants in 5D space, and for instance to analyze 2^{32} = 4,294,967,296 combinations [Hill,98].

Moreover, if the number of configurations is associated with the total number of combinations, it is evident that the first one is very less than the second one. For example, in 3D space we have 22 configurations for 256 possible combinations, this can be translated as that only the 8% of the combinations can perform the role of representatives (equivalence classes) of the others. See table 5 for the application of this comparison over the configurations in 1D, 2D and 4D spaces.

nD Space	*Combinations*	*Configurations*	*Percentage (Configurations Vs. Combinations)*
1D	4	3	75 %
2D	16	6	37.5 %
3D	256	22	8 %
4D	65,536	253	0.3 %
5D	4,294,967,296	?	<< 0.3 %

Table 5. Comparing the number of configurations with the number of combinations for the nD-OPP's.

These situations lead us to conclude that the complexity imposed by the exhaustive searching makes difficult to determine the configurations for OPP's in spaces of 5 dimensions and beyond [Hill,98]. In the following sections we will present a heuristic for obtaining the configurations in a more direct way. The heuristic has as first step to obtain a subset of the nD configurations' final set through the extrusion of (n-1)D configurations. This process will be described in the following section.

5. Extruding Configurations

The extrusion of a configuration (n-1)D to an nD space implies that each one of its boxes will be traslated in a direction that is perpendicular to the (n-1)D space in which are embedded. The traslation of each box will describe then a *hyper-box* (this process is analogous to obtaining the hypercube through the method proposed by Bragdon [Rucker,77]). It is important to consider that an nD configuration obtained through the extrusion of a (n-1)D configuration is not unique, because there are two possible traslation directions for each box. For example, in table 6 it is presented the extrusion of the 2D configuration e for obtaining 3D configurations f, g and h.

Through extruding configurations it is possible to obtain some configurations from nD space by using the configurations from (n-1)D space and so on. By this way, we obtain then a recursive process whose basic case are the configurations for 1D-OPP's (see table 1).

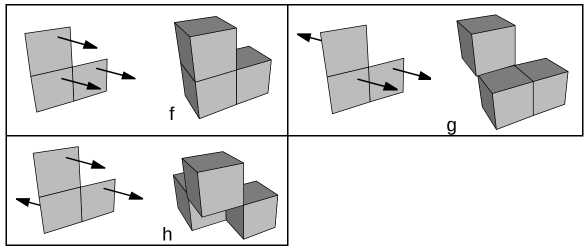


Table 6. Extrusion of 2D configuration e and the obtained 3D configurations (the arrows indicate the extrusion direction for each rectangle).

6. Obtaning the Configurations Through a "Test-Box"

The "Test-Box" heuristic starts with the following principle: to have access to (n-1)D configurations for obtaining the nD configurations. Each (n-1)D configuration is extruded just one time and in just one direction, this means that, the boxes that compose the (n-1)D configuration are extruded towards the same perpendicular direction from space in which are embedded. Once this process is applied, the (n-1)D configuration is not required again. For example, five configurations for 2D-OPP's are extruded just one time and towards the same direction for obtaining five configurations for 3D-OPP's (table 7).

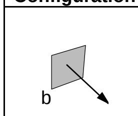
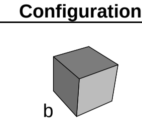
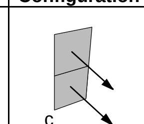
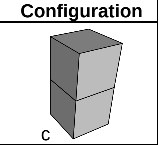
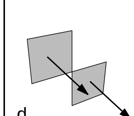
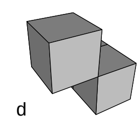
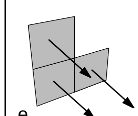
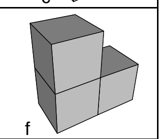
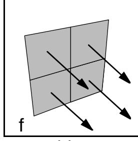
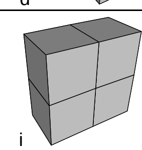
2D Configuration	**Extrusion: 3D Configuration**	**2D Configuration**	**Extrusion: 3D Configuration**
 b	 b	 c	 c
 d	 d	 e	 f
 f	 i		

Table 7. Extruding 2D configuration in the same direction and obtaining their 3D analogous.

Once the configurations from (n-1)D space have been extruded, we have now the same number of nD configurations (denominated analogous configurations [Aguilera,02]). The next step is the use of each nD configuration for obtaning the remaining configurations. We will use a "Test-Box" (a rectangle, a cube, a hypercube, etc.). For each configuration, we will add it a "Test-Box" in one of its empty hyper-octants. This adding will produce a new combination which must be analized with the set of the configurations (before combinations) yet processed. If the combination is not in the set of configurations, then we have a new configuration. This process is repeated until all the configuration's empty hyper-octants have been evaluated with a "Test-Box". In Table 8 are shown the 3D combinations obtained from the configuration f and by applying a "Test-Box" in all its empty octants.

We have now the elements to propose an algorithm applying extrusions and a "Test-Box". The algorithm is resumed with the following main procedures:

1. For a number n of dimensions we obtain the (n-1)D configurations. If n = 1 then we have the basic case

which return the configurations from table 1 (1D configurations).

2. The (n-1)D configurations are extruded in nD configurations.
3. It is added a “Test-Box” to each nD configuration in their empty hyper-octants, this operation will produce new combinations.
4. Each new produced combination will be evaluated with the set of identified configurations. If it is a new configuration then it will be added to the set of identified configurations and considered to be evaluated with a “Test-Box”, because it could produce new configurations.

We present now the proposed algorithm:

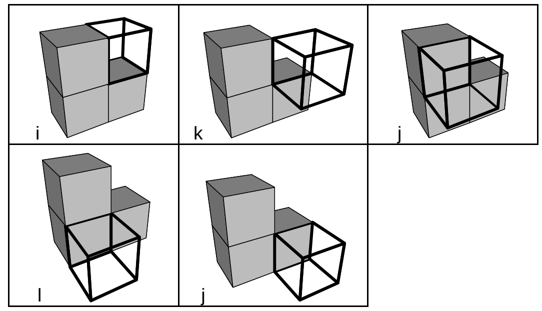


Table 8. Obtaining new configurations through 3D configuration f and a “Test-Box” (shown as wireframe model).

Input: The number of dimensions > 0 for the configurations to obtain.
Output: The set of configurations for the specified space.

```
getConfigurationsForSpaceUsingTestBox(dimensions)
{
        if(dimensions == 1)
                // Basic Case: just return the three configurations for 1D space.
                return getConfigurationsFor1DSpace( );
        else
        {
                /* Recursive call: the configurations from (n-1)D space are obtained and they are added
                   to the set ‘previousConfigurations’. */
                previousConfigurations = getConfigurationsForSpaceUsingTestBox(dimensions - 1);
                For each configuration c in the set previousConfigurations
                {
                        /* Configuration ‘c’ is (n-1)D. The configuration ‘newC’ (n-dimensional) is the
                           result of extruding configuration ‘c’. */
                        newC = extrudeConfiguration(c);
                        /* The configuration ‘newC’ is added to the set ‘configurations’ (the configurations from
                           current nD space). */
                        configurations.add(newC);
                }
                /* Starts the cicle for generating new combinations from the configurations contained in the
                   set ‘configurations’ using a “Test-Box” (rectangle, cube, hypercube,etc.) whose
                   position (hyper-octant to occupy) is indicated by variable ‘testBoxPosition’. */
                hyperOctants = 2^dimensions;
                testBoxPosition = 0;
                For each configuration c in the set configurations
                {
                        testBoxPosition = 0;
                        /* Starts the cicle for generating new combinations from configuration ‘c’ using a “Test-Box”. */
                        while(testBoxPosition < hyperOctants)
                        {
                                /* It is obtained the combination ‘newC’ from configuration ‘c’ and the “Test-Box”
                                   added in the hyper-octant specified by ‘testBoxPosition’. */
                                newC = getNewConfiguration(c, testBoxPosition);
                                /* It is verified if combination ‘newC’ was before obtained. If not, then it is added
                                   to set ‘configurations’ and for instance a new configuration has been found. */
                                if(configurations.isContained(newC) == false)
                                        configurations.add(newC);
                                testBoxPosition++;
                        }
                }
                 // All the posible configurations have been found. The set ‘configurations’ is returned as output,.
                return configurations;
        }
}
```

For determining the number of nD combinations analyzed to obtain the nD configurations through the “Test-Box” heuristic it is necessary to analyze the output’s size, i.e., the number of configurations. Due to we will know the number of configurations until the algorithm finishes, we have then an *output-sensitive* complexity analysis [deBerg,97]. Be CTB (*Configurations-by-Test-Box*) the number of configurations obtained by the algorithm and 2^n the number of hyper-octants for the nD space. Then the number of combinations to analyze is at most:

$$CTB \cdot 2^n$$

This is an upper bound because we are considering that for each configuration (with 1, 2, 3, etc. *hyper-boxes*) there are 2^n empty hyper-octants (this is possible only for configurations with 0 *hyper-boxes*). We must consider, in fact, that configurations with 1 box have 2^n-1 empty hyper-octants, configurations with 2 boxes have 2^n-2 empty hyper-octants and so on. Be CTB_i the number of those configurations with i boxes, then we have that the exact number of combinations to analyze is:

$$\sum_{i=0}^{2^n} CTB_i \cdot (2^n - i)$$

7. Results

The presented algorithm has confirmed the expected configurations for 2D, 3D [Aguilera,98] and 4D [Pérez,01] spaces. Specifically, the greatest number of combinations to analyze for obtaining the configurations in 4D space is 253 * 2^4 = 4,048. Although this is an upper bound, it is better than the obtained through exhaustive searching by [Pérez,01] (2^{16} = 65,536). Through the "Test-Box" heuristic we have found 20,983 configurations for the 5D-OPP's (see table 9 for the configuration's distribution).

5D hyper-boxes (i)	CTB_i	5D hyper-boxes (i)	CTB_i
0	1	32	1
1	1	31	1
2	5	30	5
3	10	29	10
4	38	28	38
5	66	27	66
6	164	26	164
7	236	25	236
8	454	24	454
9	570	23	570
10	887	22	887
11	989	21	989
12	1,388	20	1,388
13	1,406	19	1,406
14	1,754	18	1,754
15	1,607	17	1,607
16	1,831		

Table 9. Configuration's distribution for 5D-OPP's.

The precise number of analyzed 5D combinations is:

$$\sum_{i=0}^{2^5} CTB_i \cdot (2^5 - i) =$$

$$\left\{\begin{array}{l} 1\cdot 32 + 1\cdot 31 + 5\cdot 30 + 10\cdot 29 + 38\cdot 28 + 66\cdot 27 + 164\cdot 26 + \\ 236\cdot 25 + 454\cdot 24 + 570\cdot 23 + 887\cdot 22 + 989\cdot 21 + \\ 1{,}388\cdot 20 + 1{,}406\cdot 19 + 1{,}754\cdot 18 + 1{,}607\cdot 17 + 1{,}831\cdot 16 + \\ 1{,}607\cdot 15 + 1{,}754\cdot 14 + 1{,}406\cdot 13 + 1{,}388\cdot 12 + 989\cdot 11 + \\ 887\cdot 10 + 570\cdot 9 + 454\cdot 8 + 236\cdot 7 + 164\cdot 6 + 66\cdot 5 + \\ 38\cdot 4 + 10\cdot 3 + 5\cdot 2 + 1\cdot 1 + 1\cdot 0 \end{array}\right\}$$

$$= 335{,}728$$

This result represents a great improvement compared with the number of combinations to analyze through exhaustive searching (2^{32} = 4,294,967,296).

For obtaining the configurations for the 6D-OPP's we would have to analyze, through exhaustive searching, a total of 2^{64} = 18,446,744,073,709,551,616 combinations. Through the "Test-Box" heuristic, we found 15,440,344 configurations, which implies that the number of 6D combinations analyzed is (upper-bound):

$$15{,}440{,}344 \cdot 2^6 = 988{,}182{,}016$$

8. Conclusions

It is esential to determine the configurations for the nD-OPP's, because they represent a finite subset which can be used to determine geometric and topologic properties for these nD-OPP's. For example, [Aguilera,03] uses only the configurations for determine the properties for 4D-OPP's. Through the "Test-Box" heuristic, we have now a method faster and more direct to obtain configurations for nD-OPP's in spaces of 5 dimensions and beyond.

Acknowledgements.

Thanks to Antonio Aguilera, Pilar Gomez, and Daniel Vallejo (Computer Sciences Department, UDLAP) for their valuable comments and suggestions. This research is supported by the Consejo Nacional de Ciencia y Tecnología, CONACyT, México (project W35804-A).

References

[Aguilera,98] Aguilera, Antonio. Orthogonal Polyhedra: Study and Application. Ph.D. Thesis. Universitat Politècnica de Catalunya, 1998.

[Aguilera,02] Aguilera Ramírez Antonio & Pérez Aguila, Ricardo. Classifying the n-2 dimensional elements as Manifold or Non-Manifold for n-Dimensional Orthogonal Pseudo-Polytopes. Proceedings of the 12th International Conference on Electronics, Communications and Computers Conielecomp 2002. February 24 to 27, 2002. Acapulco, Guerrero, México.

[Aguilera,03] Aguilera Ramírez, Antonio & Pérez Aguila, Ricardo. Classifying Edges and Faces as Manifold or Non-Manifold Elements in 4D Orthogonal Pseudo-Polytopes. Submitted for Evaluation to the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2003. February 3 to 7, 2003. Plzen, Czech Republic.

[Coxeter,63] Coxeter,H.S.M: Regular Polytopes, Dover Publi-cations, Inc., New York, 1963.

[deBerg,97] de Berg, Mark; van Krevald, Marc; Overmars, Mark & Schwarzkopf, Otfried. Computational Geometry, Algorithms and Aplications. Springer Verlag, 1997.

[Herman,98] Herman, Gabor T. Geometry of Digital Spaces. Birkhäuser, Boston, 1998.

[Hill,98] Hill, S. & Roberts J.C. Generating Surface Geometry in Higher Dimensions using Local Cell Tilers. Technical Report 4-98, University of Kent at Canterbury, Computing Laboratory, University of Kent, Canterbury, Kent CT2 7NF, March 1998.

[Juan-Arinyo,88] Juan-Arinyo, R. Aportació a l'Estudi de la Transcodificació dels Models de Fronteres i Arbres Estesos en Model de Geometria Constructiva de Sòlids. Ph.D. dissertation. (written in Catalan). Escola Tècnica Superior d'Enginyers Industrials de Barcelona, Universitat Politècnica de Catalunya, 1988.

[Lorensen,87] Lorensen, W. E. & Cline, H. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, 21 (4): 44-50, 1987.

[Pérez,01] Pérez Aguila, Ricardo. 4D Orthogonal Polytopes. B.Sc. Thesis. Universidad de las Américas – Puebla, 2001.

[Preparata,85] Preparata, F.P. & Shamos, M.I. Computational Geometry: an Introduction. Springer-Verlag, 1985.

[Rucker,77] Rucker,R.V.B. Geometry, Relativity and the Fourth Dimension, Dover Publications, Inc., 1977.

[Srihari,81] Srihari, S. N. Representation of Three-Dimensional Digital Images. ACM Computing Surveys, 13 (1): 399-424. 1981.

[Van Gelder,94] Van Gelder, A. & Wilhelms, J. Topological Considerations in Isosurface Generation. ACM Transactions on Graphics, 13 (4): 337-375, 1994.

Extreme Edges: A New Characterization for 1-Dimensional Elements in 4D Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila
sp104378@mail.udlap.mx

Antonio Aguilera Ramírez
aguilera@mail.udlap.mx

Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

Abstract

This article presents our experimental study about the 1-dimensional boundary elements (edges) for 4D Orthogonal Pseudo-Polytopes (4D-OPP's). We propose a new characterization for these elements which classify them as Extreme or Non-Extreme. We show how this characterization is the result of a 3D analysis over the possible configurations for the 4D-OPP's.

Proceedings of the 13th International Conference on Electronics, Communications and Computers CONIELECOMP 2003, pp. 58-63. Editors: Báez López, J.M.D. & Murphy A., R. February 24 to 26, 2003. Cholula, Puebla, México.

Extreme Edges: A New Characterization for 1-Dimensional Elements in 4D Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila — sp104378@mail.udlap.mx
Antonio Aguilera Ramírez — aguilera@mail.udlap.mx
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

ABSTRACT

This article presents our experimental study about the 1-dimensional boundary elements (edges) for 4D Orthogonal Pseudo-Polytopes (4D-OPP's). We propose a new characterization for these elements which classify them as Extreme or Non-Extreme. We show how this characterization is the result of a 3D analysis over the possible configurations for the 4D-OPP's.

1. The 4D Orthogonal Polytopes and Their Properties

1.1. Definition

[Coxeter,63] defines an Euclidean polytope Π_n as a finite region of n-dimensional space enclosed by a finite number of (n-1)-dimensional hyperplanes. The finiteness of the region implies that the number N_{n-1} of bounding hyperplanes satisfies the inequality $N_{n-1}>n$. The part of the polytope that lies on one of these hyperplanes is called a cell. Each cell of a Π_n is an (n-1)-dimensional polytope, Π_{n-1}. The cells of a Π_{n-1} are Π_{n-2}'s, and so on; we thus obtain a descending sequence of elements Π_{n-3}, Π_{n-4},..., Π_1 (an edge), Π_0 (a vertex).

Orthogonal Polyhedra (3D-OP) are defined as polyhedra with all their edges (Π_1's) and faces (Π_2''s) oriented in three orthogonal directions ([Juan-Arinyo,88] & [Preparata,85]). Orthogonal Pseudo-Polyhedra (3D-OPP) will refer to regular and orthogonal polyhedra with non-manifold boundary [Aguilera,98].

Similarly, **4D Orthogonal Polytopes (4D-OP)** are defined as 4D polytopes with all their edges (Π_1's), faces (Π_2's) and volumes (Π_3's) oriented in four orthogonal directions and **4D Orthogonal Pseudo-Polytopes (4D-OPP)** will refer to 4D regular and orthogonal polytopes with non-manifold boundary [Aguilera,02].

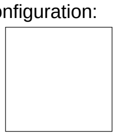
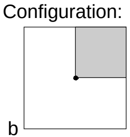
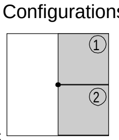
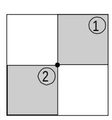
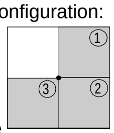
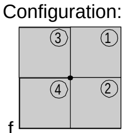
$C\binom{4}{0} =$ 1 combination with 0 surrounding rectangles. Configuration:  a	$C\binom{4}{1} =$ 4 combinations with 1 surrounding rectangle. Configuration:  b
$C\binom{4}{2} =$ 6 combinations with 2 surrounding rectangles. Configurations:  c  d	
$C\binom{4}{3} =$ 4 combinations with 3 surrounding rectangles. Configuration:  e	$C\binom{4}{4} =$ 1 combination with 4 surrounding rectangles. Configuration:  f

Table 1. Combinatorial Analysis for Configurations in the 2D-OPP's.

1.2. Configurations and Vertex Analysis for 2D-OPP's

A set of quasi-disjoint rectangles determines a 2D-OPP (2D Orthogonal Pseudo-Polygon) whose vertices must coincide with some of the rectangles' vertices [Aguilera,98]. Each of these rectangles' vertices can be considered as the origin of a 2D local coordinate system, and they may belong to up to four rectangles, one for each local quadrant. The two possible adjacency relations between the four possible rectangles can be of edge or vertex. There are $2^4 = 16$ possible combinations which, by applying symmetries and rotations, may be grouped into six equivalence classes, also called configurations [Srihari,81]. The distribution of the 16 combinations can be determined using combinatorial analysis [Aguilera,98], which is presented in table 1.

According to table 1, configurations a and f, as well as configurations b and e, are complementary to each other. Configurations c and d are self-complementary [Aguilera,98].

Considering only those configurations where all their rectangles are incident to a vertex (configurations b, c, d, e and f, see table 1) it is concluded that there are only two types of vertices in the 2D-OPP's: **the manifold vertex with two incident edges** (configurations b and e), and **the non-manifold vertex with four incident edges** (configuration d) [Aguilera,98]. The remaining configurations represent no vertex because in configuration c there are only two incident and collinear edges, and in configuration f there are no incident edges.

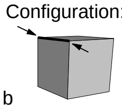
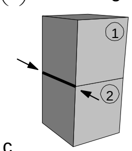
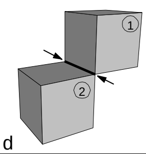
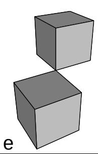
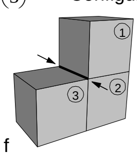
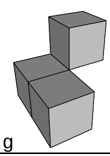
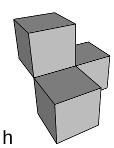
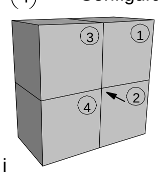
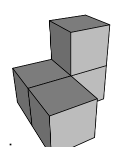
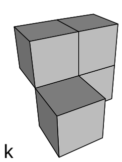
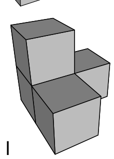
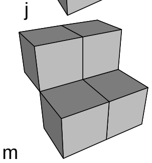
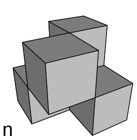
$C\binom{8}{0} =$ 1 combination with zero surrounding boxes. Configuration: a	$C\binom{8}{1} =$ 8 combinations with one surrounding box. Configuration:  b
$C\binom{8}{2} =$ 28 combinations with two surrounding boxes. Configurations:  c  d  e	
$C\binom{8}{3} =$ 56 combinations with three surrounding boxes. Configurations:  f  g  h	
$C\binom{8}{4} =$ 70 combinations with four surrounding boxes. Configurations:  i  j  k  l  m  n	

Table 2. Combinatorial Analysis for Configurations in the 3D-OPP's.

1.3. Configurations and Edge Analysis for 3D-OPP's

A set of quasi-disjoint boxes determines a 3D-OPP whose vertices must coincide with some of the boxes' vertices [Aguilera,98]. Each of these boxes' vertices can be considered as the origin of a 3D local coordinate system, and they may belong to up to eight boxes, one for each local octant. There are $2^8 = 256$ possible combinations which, by applying symmetries and rotations, may be grouped into 22 equivalence classes [Loresen,87], also called configurations [Srihari,81]. Each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class [Aguilera,98]. Grouping complementary configurations leads to the 14 major cases [Van Gelder,94]. The distribution of the 256 combinations can be determined using combinatorial analysis [Aguilera,98], which is presented in table 2.

The combinations with 5, 6, 7 and 8 surrounding boxes are complementary, and thus analogous, to combinations with 3, 2, 1 and 0 surrounding boxes (table 2), respectively [Aguilera,98]. Finally, each configuration, with four surrounding boxes is self-complementary.

Considering only those configurations where all their cubes are incident to a same edge (b, c, d, f and i; table 2), it is concluded that there are only two types of edges in the 3D-OPP's [Aguilera,98]:

- **The manifold edge with two incident faces.** This type of edges is found in configurations b and f. The edge's two incident faces in configuration b belong to one cube's boundary and they are perpendicular to each other. The edge's two incident faces in configuration f belong to two different cubes with edge adjacency and they result perpendicular to each other.
- **The non-manifold edge with four incident faces**. This type of edges is found in configuration d, where two of its faces belongs to a cube and the remaining belong to a second cube with edge adjacency.
- The remaining configurations represent no edge because in configuration c there are only two incident and coplanar faces, and in configuration i there are no incident faces.

1.4. Configurations and Face Analysis for 4D-OPP's

A set of quasi-disjoint *hyper-boxes* (i.e., hypercubes, which in this paper will be represented using Claude Bragdon's projection [Rucker,77]) determines a 4D-OPP whose vertices must coincide with some of the *hyper-boxes'* vertices. We will consider the *hyper-boxes'* vertices as the origin of a 4D local coordinate system, and they may belong to up to 16 *hyper-boxes*, one for each local *hyper-octant*. The 4D-OPP's vertices are determined according to the presence of absence of each of these 16 surrounding *hyper-boxes*. The four possible adjacency relations between the 16 possible *hyper-boxes* can be of volume, face, edge or vertex. There are $2^{16} =$ 65,536 possible combinations of vertices in 4D-OPP's which can be grouped, applying symmetries and rotations, into 253 equivalence classes, also called configurations [Pérez,01]. Each configuration has its complementary configuration which is the class that contains the complementary combinations of all the combinations in the given class. Grouping complementary configurations leads to the 145 major cases [Pérez,01].

The distribution of the 65,536 combinations can be determined using combinatorial analysis [Pérez,01]:

$C\binom{16}{0} =$ 1 combination with zero surrounding hyper-boxes: configuration 1.

$C\binom{16}{1} =$ 16 combinations with one surrounding hyper-box: configuration 2, shown in table 3.

$C\binom{16}{2} =$ 120 combinations with two surrounding hyper-boxes: configurations 3, 4 (table 3), 5 and 6.

$C\binom{16}{3} =$ 560 combinations with three surrounding hyper-boxes: configurations 7 (table 3) to 12.

$C\binom{16}{4} =$ 1,820 combinations with four surrounding hyper-boxes: configurations 13 (table 3) to 28.

$C\binom{16}{5} =$ 4,368 combinations with five surrounding hyper-boxes: configurations 29 to 48.

$C\binom{16}{6} =$ 8,008 combinations with six surrounding hyper-boxes: configurations 49 to 78.

$C\binom{16}{7} =$ 11,440 combinations with seven surrounding hyper-boxes: configurations 79 to 108.

$C\binom{16}{8} =$ 12,870 combinations with eight surrounding hyper-boxes: configurations 109 to 145.

The remaining combinations with 9, 10, 11, 12, 13, 14, 15 and 16 surrounding hyper-boxes are complementary, and thus analogous, to combinations with 7, 6, 5, 4, 3, 2, 1 and 0 surrounding hyper-boxes, respectively. Finally, each configuration, with eight surrounding hyper-boxes is self-complementary [Pérez,01].

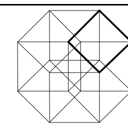
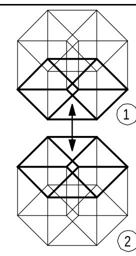
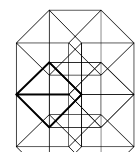
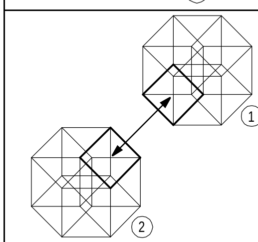
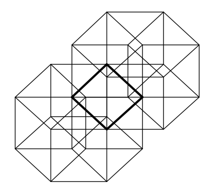
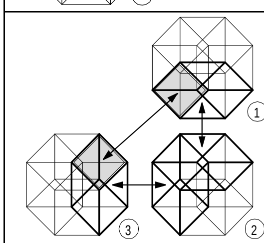
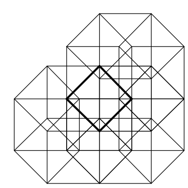
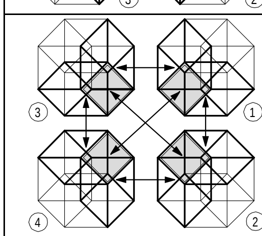
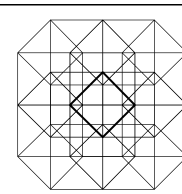
Adjacencies between hyper-boxes	Configuration
	 **2**
	 **3**
	 **4**
	 **7**
	 **13**

Table 3: Configurations 2, 3, 4, 7 and 13 for 4D-OPP's (each hypercube is shown using Bragdon's projection).

Considering only those configurations where all their *hyper-boxes* are incident to just one face (configurations 2, 3, 4, 7 and 13, see table 3), it results that there are only two types of faces in the 4D-OPP's (for a more detailed analysis see [Aguilera,02]):

- **The manifold faces with two incident volumes.** The face's two incident volumes in configuration 2 belong to the boundary of only one hypercube and they are perpendicular to each other. While in configuration 7, The face's two incident volumes belong to two different hypercubes with face adjacency and they result perpendicular to each other.

- **The non-manifold faces with four incident volumes.** This type of faces is found in configuration 4, where two of its incident volumes belongs to a hypercube and the remaining two belong to a second hypercube with face adjacency.
- The remaining configurations represent no face because in configuration 3 there are only two incident and *co-hyperplanar* volumes, and in configuration 13 there are no incident volumes (analogous to 3D configurations c and i in table 2).

1.5. The Eight Types of Vertices in the 3D-OPP's

The vertices in the 3D-OPP's can be classified depending on the number of two-manifold and non-manifold edges incident to them. They are referred as V3, V4, V4N1, V4N2, V5N, V6, V6N1 and V6N2 (there are also two non valid vertices) [Aguilera,98]. In this nomenclature "V" means vertex, the first digit shows the number of incident edges, the "N" is present if at least one non-manifold edge is incident to the vertex and the second digit is included to distinguish between two different types that otherwise could receive the same name (See [Aguilera,98] for detailed properties of these eight vertices). See table 4.

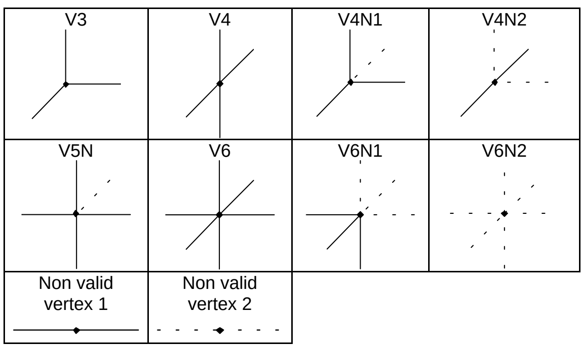


Table 4. Vertices present in 3D-OPP's (dotted lines indicate non-manifold edges and continuos lines indicate manifold edges).

1.6. The Eight Types of Edges in the 4D-OPP's

Analogously to the vertices in the 3D-OPP's, the edges in 4D-OPP's can be described in terms of the manifold or non-manifold faces that are incident to them. In this way, [Pérez,01] has identified eight types of edges and two non valid edges; and extended the nomenclature used by [Aguilera,98] to describe them. Such edges are referred as E3, E4, E4N1, E4N2, E5N, E6, E6N1 and E6N2 (See Table 5). The only difference with the nomenclature used by [Aguilera,98] is that "E" means edge instead of "V" that means vertex (See [Pérez,01] for detailed properties of these eight edges).

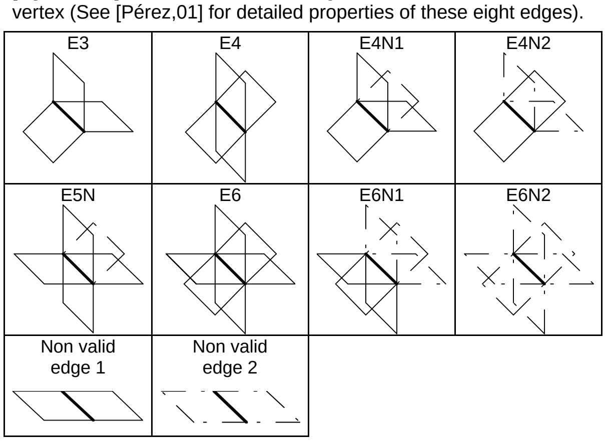


Table 5. Edges present in 4D-OPP's (dotted lines indicate non-manifold faces and continuos lines indicate manifold faces)

2. The Extreme Vertices in the 3D-OPP's

2.1. Properties

[Aguilera,98] defines a brink or extended-edge as the maximal uninterrupted segment, built out of a sequence of collinear and contiguous two-manifold edges of a 3D-OPP with the following properties:

- Non-manifold edges do not belong to brinks.
- Every two-manifold edge belongs to a brink, whereas every brink consists of *m* edges ($m \geq 1$), and contains $m+1$ vertices.
- Two of the vertices of type V3, V4N1 or V6N1 (section 1.5) are at either extreme of the brink (**Extreme Vertices**). These vertices have in common that they are the only ones that have exactly three incident two-manifold and perpendicular edges, regardless of the number of incident non-manifold edges, therefore those vertices mark the end of brinks in all three orthogonal directions.
- The m-1 vertices of type V4, V4N2, V5N or V6 are the only common point of two collinear edges of a same brink (interior vertices).
- Due to all six incident edges of a V6N2 vertex are non-manifold edges, none of them belongs to a brink, thus this vertex does not belong to any brink.
  (This work not consider brinks in 1D-OPP's and 2D-OPP's, however see [Aguilera,98] for details). See Figure 1.a for an example of a wireframe model of a 3D-OPP. Also in Figure 1.b are shown the OPP's brinks parallel to X axis. The continous lines indicate manifold edges and the dotted one a non-manifold edge (it do not belong to a brink). The points at both extremes of the brinks are Extreme Vertices.

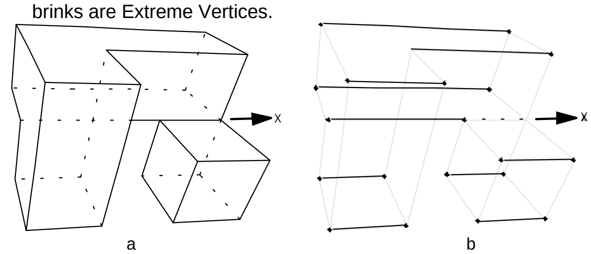


Figure 1. a) A wireframe model of a 3D-OPP. b) Their brinks parallel to X axis (See text for details).

Based in the previous properties for brinks, [Aguilera,98] presents the following properties for the **Extreme Vertices** in the 3D-OPP's:

- Every Extreme Vertex of a 3D-OPP has exactly 3 incident manifold edges perpendicular to each other. This number is even for every non-extreme vertex.
- Every Extreme Vertex has an odd number of incident faces, and every non-extreme vertex has an even number of incident faces.
- Any Extreme Vertex of a 3D-OPP when is locally described by a set of surrounding boxes, is surrounded by an odd number of such boxes. An even number of surrounding boxes either defines a non-extreme vertex, or does not define any vertex at all.

2.2. The 2D Analysis for Vertices in 3D-OPP's

In section 1.3 were presented the configurations, identified by [Aguilera,98], which determine a 3D-OPP through a set of quasi-disjoint boxes. Each of these boxes' vertices can be considered as the origin of a 3D local coordinate system. In such 3D local coordinate system can be identified three main planes: XY, YZ and XZ. If the faces that are coplanar to such main planes are grouped, ignoring those faces that are shared by two cubes (face adjacency), they compose three 2D configurations (one for each main plane). For these 2D configurations the vertex can be classified as manifold or non-manifold (section 1.2). See Table 6 for examples for 3D configurations b to k.

Applying this analysis over the 22 configurations for the 3D-OPP's [Pérez, 01], it results that for those configurations whose vertex is extreme (V3, V4N1 or V6N1) and their number of boxes is odd, the three vertex analysis for their 2D configurations classify the 2D vertex as manifold (in Table 6, configurations b and f, for example). From this pattern, we can infer if a vertex is extreme or non-extreme.

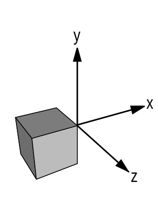
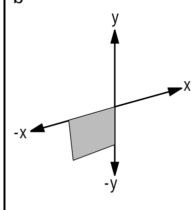
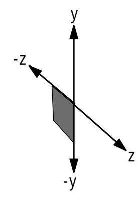
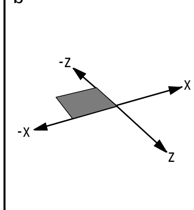
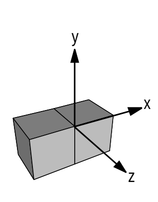
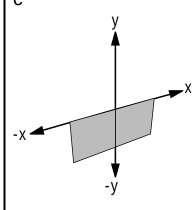
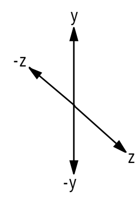
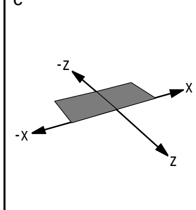
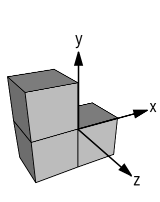
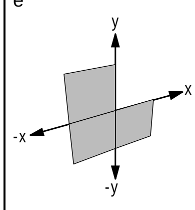
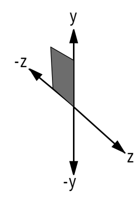
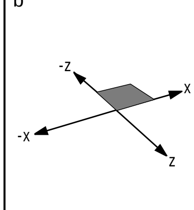
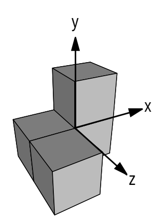
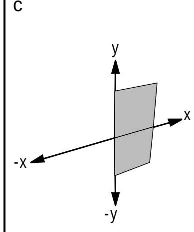
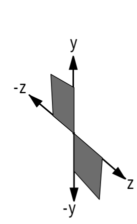
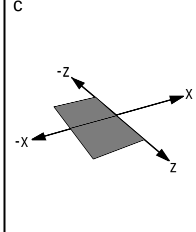
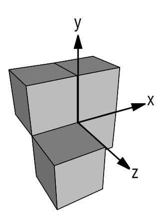
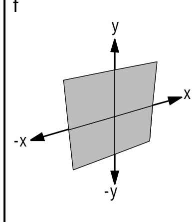
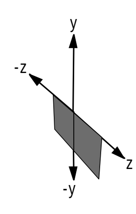
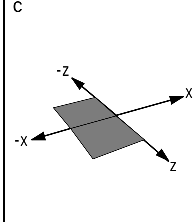
3D configuration	2D configuration on XY Plane	2D configuration on YZ Plane	2D configuration on XZ Plane	Analysis for 2D vertex
b 	b 	b 	b 	***XY: Manifold*** ***YZ: Manifold*** ***XZ: Manifold***
c 	c 	a 	c 	XY: Non vertex YZ: Non vertex XZ: Non vertex
f 	e 	b 	b 	***XY: Manifold*** ***YZ: Manifold*** ***XZ: Manifold***
j 	c 	d 	c 	XY: Non vertex YZ: Non manifold XZ: Non vertex
k 	f 	c 	c 	XY: Non vertex YZ: Non vertex XZ: Non vertex

Table 6. Vertex analysis for 2D configurations on the main planes in 3D configurations b to k.

2.3. The 3D Analysis for Edges in 4D-OPP's

The vertex analysis for 2D configurations embedded in the main planes of a 3D configuration (previous section) classify the 2D vertex as manifold or non-manifold, and through these three 2D analysis we can infer if the 3D vertex is extreme or non-extreme. For consequence, in analogous way, we can assume that the edges analysis for 3D configurations embedded in the main hyperplanes of a 4D configuration will classify to 3D edges as manifold or non-manifold, and through these 3D analysis we can infer, due to the analogy with 3D vertex, if the 4D edges are **"Extreme" or "Non-Extreme"**.

In section 1.4 were presented the 253 configurations which determine a 4D-OPP through a set of quasi-disjoint hyper-boxes (hypercubes). Each of these hyper-boxes' vertices can be considered as the origin of a 4D local coordinate system. In such 4D local coordinate system can be identified four main hyperplanes: XYZ, XYW, XZW and YZW. If the volumes that are *co-hyperplanar* to such main hyperplanes are grouped, ignoring those volumes that are shared by two hypercubes (volume adjacency), they will compose four 3D configurations (one for each main hyperplane). Table 7 presents the four 3D configurations that are present in 4D configurations 3 to 6.

For the 3D configurations that are embedded in the main hyperplanes in 4D space, it is possible to analyze their edges and classify them as manifold or non-manifold (section 1.3). In Table 8 are shown the edges analysis for the 3D configurations that are present in 4D configurations 3 to 6.

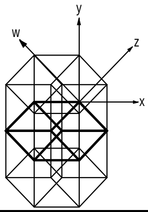
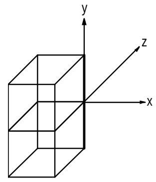
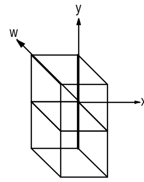
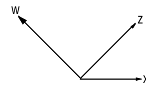
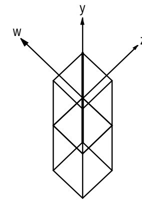
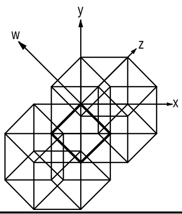
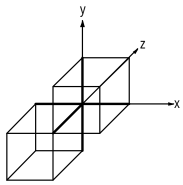
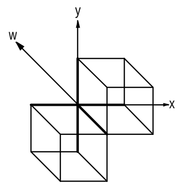
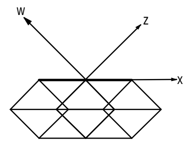
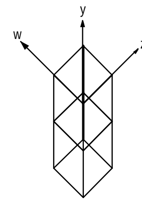
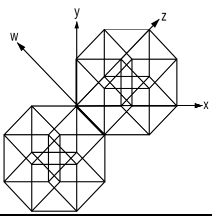
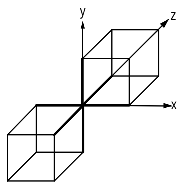
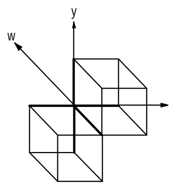
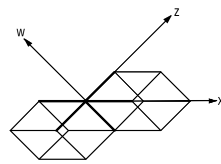
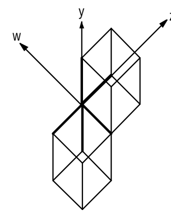
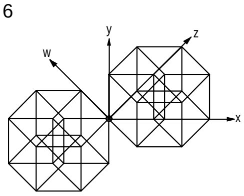
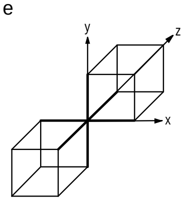
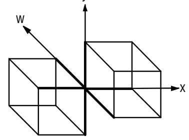
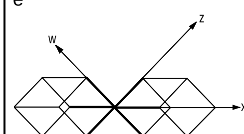
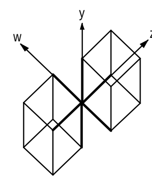
4D configuration	3D configuration on XYZ hyperplane	3D configuration on XYW hyperplane	3D configuration on XZW hyperplane	3D configuration on YZW hyperplane
3 	b 	b 	a 	b 
4 	d 	d 	b 	b 
5 	e 	d 	d 	d 
6 	e 	e 	e 	e 

Table 7. Determining the 3D configurations on the main hyperplanes in 4D configurations 3 to 6.

	3D Edges Analysis			
4D Configuration	Configuration on XYZ hyperplane	Configuration on XYW hyperplane	Configuration on XZW hyperplane	Configuration on YZW hyperplane
3	X: Non edge -X: Non edge ***Y: Manifold*** ***-Y: Manifold*** Z: Non edge -Z: Non edge	X: Non edge -X: Non edge ***Y: Manifold*** ***-Y: Manifold*** W: Non edge -W: Non edge	X: Non edge -X: Non edge Z: Non edge -Z: Non edge W: Non edge -W: Non edge	***Y: Manifold*** ***-Y: Manifold*** Z: Non edge -Z: Non edge W: Non edge -W: Non edge
4	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** Z: Non edge -Z: Manifold	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** W: Non manifold -W: Non edge	***X: Manifold*** ***-X: Manifold*** Z: Non edge -Z: Non edge W: Non edge -W: Non edge	***Y: Manifold*** ***-Y: Manifold*** Z: Non edge -Z: Non edge W: Non edge -W: Non edge
5	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** ***Z: Manifold*** ***-Z: Manifold***	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** W: Non edge -W: Non manifold	***X: Manifold*** ***-X: Manifold*** ***Z: Manifold*** ***-Z: Manifold*** W: Non edge -W: Non manifold	***Y: Manifold*** ***-Y: Manifold*** ***Z: Manifold*** ***-Z: Manifold*** W: Non edge -W: Non manifold
6	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** ***Z: Manifold*** ***-Z: Manifold***	***X: Manifold*** ***-X: Manifold*** ***Y: Manifold*** ***-Y: Manifold*** ***W: Manifold*** ***-W: Manifold***	***X: Manifold*** ***-X: Manifold*** ***Z: Manifold*** ***-Z: Manifold*** ***W: Manifold*** ***-W: Manifold***	***Y: Manifold*** ***-Y: Manifold*** ***Z: Manifold*** ***-Z: Manifold*** ***W: Manifold*** ***-W: Manifold***

Table 8. Edges analysis for 3D configurations on the main hyperplanes in 4D configurations 3 to 6.

3. Results

Through a computer program [Pérez,01], the edges analysis for the 3D configurations embedded in the main hyperplanes of a 4D configuration, was applied over the 253 configurations for the 4D-OPP's and the obtained results are:

- A edge in a 4D-OPP can be classified by three 3D analysis (a 4D edge can only be present in three of the four main hyperplanes) as:
    - 3 times as manifold and 0 times as non-manifold, or
    - 0 times as manifold and once as non-manifold, or
    - 0 times as manifold and 3 times as non-manifold, or
    - 0 times as manifold and 0 times as non-manifold.
- The above patterns can be found in any 4D configuration because it can have from 0 to 8 incident edges to the origin.
- Following the analogy with the vertex analysis for 2D configurations embedded in the main planes of a 3D configuration (section 2.2), we can propose that if a edge in a 4D-OPP has been classified in the 3D analysis three times as manifold, then it can be considered as an **Extreme edge**, and any other result will classify it as a **Non-Extreme Edge**.
- The manifold or non-manifold classification for a edge in a 4D-OPP is independent of its classification as extreme or non-extreme. Is the same situation for a vertex in a 3D-OPP, where its classification as extreme or non-extreme is independent of its classification as manifold or non-manifold (For the topic of the characterization of vertices and edges in 3D-OPP's and 4D-OPP's respectively, as manifold or non-manifold see [Aguilera,03]).
- If we analyze the incident manifold or non-manifold faces that are incident to an extreme or non-extreme edge in 4D-OPP's, we can observe that the analogy with the description of extreme or non-extreme vertices in terms of the incident manifold or non-manifold edges that are incident to those vertices is preserved, as shown in Table 9.

4D edge	Classification (Extreme or Non-Extreme)	3D vertex	Classification (Extreme or Non-Extreme)
E3	***Extreme***	V3	***Extreme***
E4	Non extreme	V4	Non extreme
E4N1	***Extreme***	V4N1	***Extreme***
E4N2	Non extreme	V4N2	Non extreme
E5N	Non extreme	V5N	Non extreme
E6	Non extreme	V6	Non extreme
E6N1	***Extreme***	V6N1	***Extreme***
E6N2	Non extreme	V6N2	Non extreme

Table 9. The 4D-OPP's edges classifications and their analogy with 3D-OPP's vertices.

Conclusions and Future Work

The characterization of edges, as Extreme or Non-Extreme, together with the classification of faces and edges as manifold or non-manifold (both discussed in [Aguilera,02] and [Aguilera,03]), provide a solid theorical base for extending the Extreme Vertices Model (EVM), presented in [Aguilera,97] and [Aguilera, 98], to the fourth dimensional space (EVM-4D). The EVM-4D will be a representation model for 4D Orthogonal Polytopes that will allow queries and operations over them. However, the fact related to a model purely geometric (four geometric dimensions) is not restrictive for our research, because it will be applied under geometries as the 4D spacetime. The first main application for the EVM-4D covers the visualization and analysis for multidimensional data and events under the context of a Geographical Information System (GIS).

Acknowledgements.

This research is supported by the Consejo Nacional de Ciencia y Tecnología, CONACyT, México (project W35804-A).

References.

[Aguilera,97] Aguilera, A. & Ayala, D. Orthogonal Polyhedra as Geometric Bounds in Constructive Solid Geometry. In C. Hoffman, and W. Bronsvort, editors, Fourth ACM Siggraph Symposium on Solid Modeling and Applications (SM' 97), pp. 56-67, Atlanta (USA), 1997.

[Aguilera,98] Aguilera, Antonio. Orthogonal Polyhedra: Study and Application. Ph.D. Thesis. Universitat Politècnica de Catalunya, 1998.

[Aguilera,02] Aguilera Ramírez, Antonio & Pérez Aguila, Ricardo. Classifying the n-2 Dimensional Elements as Manifold or Non-Manifold for n-Dimensional Orthogonal Pseudo-Polytopes. Memoria Técnica del XII Congreso Internacional de Ingeniería Electrónica, Comunicaciones y Computadoras CONIELECOMP 2002, pp. 59-63. 24 a 27 de febrero 2002. Universidad de las Américas – Puebla.

[Aguilera,03] Aguilera Ramírez, Antonio & Pérez Aguila, Ricardo. Classifying Edges and Faces as Manifold or Non-Manifold Elements in 4D Orthogonal Pseudo-Polytopes. Submitted for Evaluation to the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2003. To be held february 3 to 7, 2003. Plzen, Czech Republic.

[Coxeter,63] Coxeter, H.S.M. Regular Polytopes, Dover Publications, Inc., New York, 1963.

[Juan-Arinyo,88] Juan-Arinyo, R. Aportació a l'Estudi de la Transcodificació dels Models de Fronteres i Arbres Estesos en Model de Geometria Constructiva de Sòlids. Ph.D. dissertation. (written in Catalan). Escola Tècnica Superior d'Enginyers Industrials de Barcelona, Universitat Politècnica de Catalunya, 1988.

[Lorensen,87] Lorensen, W. E. & Cline, H. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, 21 (4): 44-50, 1987.

[Srihari,81] Srihari, S. N. Representation of Three-Dimensional Digital Images. ACM Computing Surveys, 13 (1): 399-424. 1981.

[Pérez,01] Pérez Aguila, Ricardo. 4D Orthogonal Polytopes. B.Sc. Thesis. Universidad de las Américas – Puebla, 2001.

[Preparata,85] Preparata F. P. & Shamos, M. I. Computational Geometry: an Introduction. Springer-Verlag, 1985.

[Rucker,77] Rucker,R.V.B: Geometry, Relativity and the Fourth Dimension, Dover Publications, Inc., New York, 1977.

[Van Gelder,94] Van Gelder, A. & Wilhelms, J. Topological Considerations in Isosurface Generation. ACM Transactions on Graphics, 13 (4): 337-375, 1994.

Characterizing the (n-3)-Dimensional Elements as Manifold or Non-Manifold in n-Dimensional Orthogonal Pseudo-Polytopes

Ricardo Pérez Aguila   Antonio Aguilera Ramírez
sp104378@mail.udlap.mx   aguilera@mail.udlap.mx
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir, Zip Code: 72820. Cholula, Puebla, México.
Phone: +52 (222) 229-2664

Abstract

This article presents our experimental results for classifying edges as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes (4D-OPP's). For this edges' analysis in 4D-OPP's we have developed two approaches: 1) The analogy between incident (manifold and non-manifold) edges to a vertex in 3D Orthogonal Pseudo-Polyhedra (3D-OPP's) with incident (manifold and non-manifold) faces to a edge in 4D-OPP's; and 2) The extension of the concept of "cones of faces" (which is applied for classifying a vertex in 3D-OPP's as manifold or non-manifold) to "hypercones of volumes" for classifying an edge as manifold or non-manifold in 4D-OPP's. Both approaches have provided the same results, which present that there are eight types of edges in 4D-OPP's. Finally, the generalizations for classifying the n-3 dimensional boundary elements for n-dimensional Orthogonal Pseudo-Polytopes as manifold or non-manifold elements is also presented.

Proceedings of the 13th International Conference on Electronics, Communications and Computers CONIELECOMP 2003, pp. 53-57. Editors: Báez López, J.M.D. & Murphy A., R. February 24 to 26, 2003. Cholula, Puebla, México.

Characterizing the (n-3)-Dimensional Elements as Manifold or Non-Manifold in n-Dimensional Orthogonal Pseudo-Polytopes[1]

Ricardo Pérez Aguila   Antonio Aguilera Ramírez
sp104378@mail.udlap.mx   aguilera@mail.udlap.mx
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Ex-Hacienda Santa Catarina Mártir. Phone: +52 (222) 229-2664
México 72820, Cholula, Puebla.

Abstract

This article presents our experimental results for classifying edges as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes (4D-OPP's). For this edges' analysis in 4D-OPP's we have developed two approaches: 1) The analogy between incident (manifold and non-manifold) edges to a vertex in 3D Orthogonal Pseudo-Polyhedra (3D-OPP's) with incident (manifold and non-manifold) faces to a edge in 4D-OPP's; and 2) The extension of the concept of "cones of faces" (which is applied for classifying a vertex in 3D-OPP's as manifold or non-manifold) to "hypercones of volumes" for classifying an edge as manifold or non-manifold in 4D-OPP's. Both approaches have provided the same results, which present that there are eight types of edges in 4D-OPP's. Finally, the generalizations for classifying the n-3 dimensional boundary elements for n-dimensional Orthogonal Pseudo-Polytopes as manifold or non-manifold elements is also presented.

1. Introduction

Recent interest has been growing in studying multidimensional polytopes (4D and beyond) for representing phenomena in n-dimensional spaces. Some examples include:

- In [Feiner,90] is presented the *n-Vision* system for the visualization of n-dimensional spaces. Its applications are related to the visualization and control of multidimensional financial data.
- [Wegenkittl,97] presents a visualization interactive tool for exploring and analyzing multidimensional dinamical systems. Such systems include chemical reactions and statistical models.
- [Lees,99] describes *Geotouch*, a Geographical Information System (GIS) which includes the time as a fourth dimension with the objective of visualizing earthquake hypocenters, volcanic eruptions or other time sequences of events.
- In [Weeks,02] a set of tools for visualizing and understanding 2 and 3-manifolds are referred, whose main objective is to analyze the possible topologies of our universe.

Those examples show how some of these phenomena's features rely on the polytopes' geometric and topologic relations. However, due to the need of visualizing and analyzing these phenomena (i.e. multidimensional data), it is essential first to analyze these polytopes and their boundaries that compose them [Herman,98]. So, this article covers that first step, in our research, with the boundary's analysis for classifying edges as manifold or non-manifold elements in 4D Orthogonal Pseudo-Polytopes.

2. The 4D Orthogonal Polytopes

2.1 Definition

[Coxeter,63] defines an Euclidean polytope Π_n as a finite region of n-dimensional space enclosed by a finite number of (n-1)-dimensional hyperplanes. The finiteness of the region implies that the number N_{n-1} of bounding hyperplanes satisfies the inequality $N_{n-1}>n$. The part of the polytope that lies on one of these hyperplanes is called a cell. Each cell of a Π_n is an (n-1)-dimensional polytope, Π_{n-1}. The cells of a Π_{n-1} are Π_{n-2}'s, and so on; we thus obtain a descending sequence of elements Π_{n-3}, Π_{n-4},..., Π_1 (an edge), Π_0 (a vertex).

We know that a Π_3 (a 3D Euclidean polytope) is a polyhedron. The polyhedron's cells are Π_2. A Π_2 (a 2D Euclidean polytope) is a polygon. The polygon's cells are Π_1. A Π_1 (a 1D Euclidean polytope) is a segment. Finally, the segment's cells are Π_0, a set of vertices. The cells of a Π_4 (a 4D Euclidean polytope) are Π_3 (polyhedra, also called volumes in the context of Π_4).

Orthogonal Polyhedra (3D-OP) are defined as polyhedra with all their edges and faces oriented in three orthogonal directions ([Joan-Arinyo,88] & [Preparata,85]). Orthogonal Pseudo-Polyhedra (3D-OPP) will refer to regular and orthogonal polyhedra with non-manifold boundary [Aguilera,98]. Similarly, **4D Orthogonal Polytopes (4D-OP)** are defined as 4D polytopes with all their edges, faces and volumes oriented in four orthogonal directions and **4D Orthogonal Pseudo-Polytopes (4D-OPP)** will refer to 4D regular and orthogonal polytopes with non-manifold boundary. Because the 4D-OPP's definition is an extension from the 3D-OPP's, is easy to generalize the concept to define **n-dimensional Orthogonal Polytopes (nD-OP)** as n-dimensional polytopes with all their Π_{n-1}, Π_{n-2},..., Π_1 oriented in n orthogonal directions. Finally, **n-dimensional Ortho-**

[1] In this article we have omitted some background information because it is contained in [Pérez,03a] and [Pérez,03b]. Both references also have been submitted to this CONIELECOMP Conference. In the following sections will be indicated which of the references should be consulted to obtain the proper background information.

gonal Pseudo-Polytopes (nD-OPP) are defined as n-dimensional regular and orthogonal polytopes with non-manifold boundary.

3. The Π_{n-3} Analysis for 3D and 4D-OPP's

3.1 The Π_0 Analysis for 3D-OPP's

There are eight types of vertices (also two non valid vertices are identified) for 3D-OPP's [Aguilera,98]. These vertices can be classified depending on the number of two-manifold and non-manifold edges[2] incident to them and they are referred as V3, V4, V4N1, V4N2, V5N, V6, V6N1 and V6N2 [Aguilera,98] (Table 1). In this nomenclature "V" means vertex, the first digit shows the number of incident edges, the "N" is present if at least one non-manifold edge is incident to the vertex and the second digit is included to distinguish between two different types that otherwise could receive the same name.

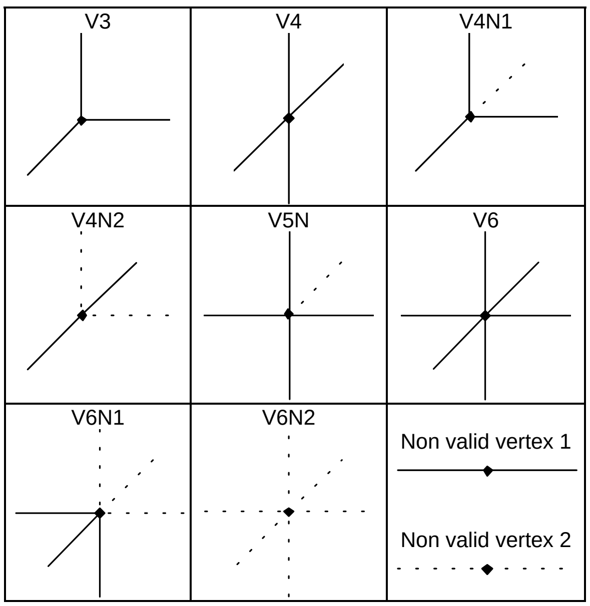


Table 1. Vertices present in 3D-OPP's (dotted lines indicate non-manifold edges and continuous lines indicate manifold edges).

Each vertex has the following properties [Aguilera,98]:

- V3: all three incident edges are two-manifold and perpendicular to each other. It is present in 3D configurations[3] b, f, o and u.
- V4: all four incident edges are two-manifold, they lie on a plane, and can be grouped in two couples of collinear edges. It is present in configuration j.
- V4N1: three of its four incident edges are perpendicular to each other and also two-manifold ones, while the fourth is non-manifold and collinear to one of the other three. It is present in configurations g and p.
- V4N2: two of its four incident edges are two-manifold and collinear, while each of its other two is non-manifold and perpendicular to the other three. It is present in configuration k.
- V5N: four of its five incident edges are two-manifold and lie in a plane, while the fifth is non-manifold and perpendicular to the rest of them. It is present in configurations d and s.
- V6: all six incident edges are two-manifold. It is present in configurations e, l and t.
- V6N1: three of its six incident edges are perpendicular to each other and also two-manifold ones, while each of its remaining three edges is non-manifold and collinear to one of the first three. It is present in configurations h and q.
- V6N2: all of its six incident edges are non-manifold. It is present in configuration n.
- Non valid vertex 1: its two manifold edges are collinear. It is present in configurations c and r.
- Non valid vertex 2: its two non-manifold edges are collinear. It is present in configuration m.

3.2 The Π_1 Analysis for 4D-OPP's

Vertices can be defined in terms of the manifold or non-manifold edges that are incident to these vertices in 3D-OPP's [Aguilera,98]. The same process will be extended to describe edges in terms of the manifold or non-manifold faces[2] that are incident to those edges in 4D-OPP's. In this way, we have identified eight types of edges and two non valid edges. We will also extend the nomenclature used by [Aguilera,98] to describe them. Such edges will be referred as E3, E4, E4N1, E4N2, E5N, E6, E6N1 and E6N2 (Table 2). The only difference with the nomenclature used to describe the vertices is that "E" means edge instead of "V" that means vertex.

Each edge has the following properties:

- E3: all three incident faces are two-manifold and perpendicular to each other.
- E4: all four incident faces are manifold and lie on a hyperplane, and they can be grouped in two couples of coplanar faces.
- E4N1: three of its four incident faces are perpendicular to each other and also two-manifold ones, while the fourth is non-manifold and coplanar to one of the other three.
- E4N2: two of its four incident faces are two-manifold and coplanar, while each of its other two is non-manifold and perpendicular to the other three.
- E5N: four of its five incident faces are two-manifold and lie in a hyperplane, while the fifth is non-manifold and perpendicular to the rest of them.
- E6: all six incident faces are two-manifold.
- E6N1: three of its six incident faces are perpendicular to each other and also manifold ones, while each of its remaining three faces is non-manifold and coplanar to one of the first three.

[2] The characterization as manifold or non-manifold for edges in 3D-OPP's and faces in 4D-OPP's is resumed in [Pérez,03a].

[3] The nD-OPP's can be represented and/or decomposed by a set of configurations or equivalence classes. See [Pérez,03b] for an introduction to this topic.

- E6N2: all of its six incident faces are non-manifold.
- Non valid edge 1: its two manifold faces are coplanar.
- Non valid edge 2: its two non-manifold faces are coplanar.

It results interesting that the number, classifications and positions of the incident faces to an edge in 4D-OPP's are analogous to the way that a set of edges are incident to a vertex in 3D-OPP's.

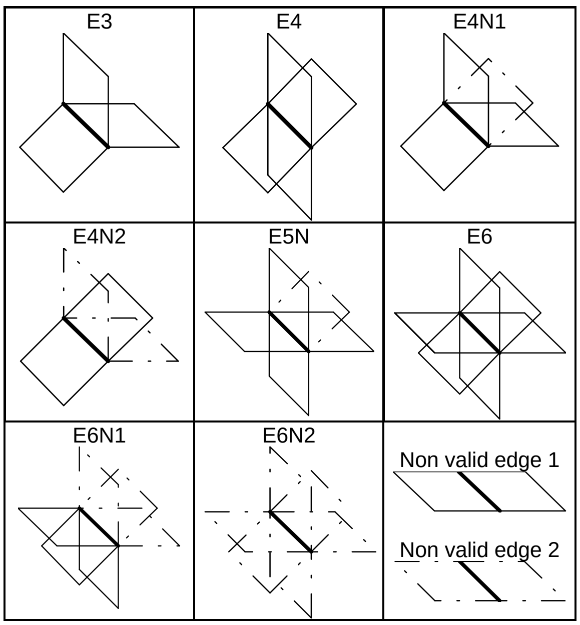


Table 2. Edges present in 4D-OPP's (dotted lines indicate non-manifold faces and continuous lines indicate manifold faces).

3.3 Classifying the Π_0 in Polyhedra Through its Cones of Faces

A **polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite set of plane polygons such that every edge of a polygon is shared by exactly one other polygon (adjacent polygons) [Preparata,85]. A **pseudo-polyhedron** is a bounded subset of the 3D Euclidean Space enclosed by a finite collection of planar faces such that every edge has at least two adjacent faces, and if any two faces meet, they meet at a common edge [Tang,91].

Edges and vertices, as boundary elements for polyhedra, may be either two-manifold (or just manifold) or non-manifold elements. In the case of edges, they are (non) manifold elements when every points of it is also a (non) manifold point, except that either or both of its ending vertices might be a point of the opposite type [Aguilera,98]. A manifold edge is adjacent to exactly two faces, and a manifold vertex is the apex (i.e., the common vertex) of only one **cone of faces**. Conversely, a non-manifold edge is adjacent to more than two faces, and a non-manifold vertex is the apex (i.e., the common vertex) of more than one **cone of faces** [Rossignac,91].

3D vertex	Configuration(s)	Classification
V3	b, f, o, u	Manifold
V4	j	Manifold
V4N1	g, p	Non-manifold
V4N2	k	Non-manifold
V5N	d, s	Non-manifold
V6	e, l, t	Non-manifold in configurations e and t. Manifold for configuration l.
V6N1	h, q	Non-manifold
V6N2	n	Non-manifold

Table 3. 3D-OPP's vertices classification.

Using the concept of cones of faces it is easy to construct an algorithm to determine the classification of a vertex as manifold or non-manifold in any polyhedron or pseudo-polyhedron. Using this algorithm over the possible vertices in 3D-OPP's we have the results presented in Table 3 which coincide with those presented by [Aguilera,98].

3.4 Classifying the Π_1 in 4D Polytopes Through its Hyper-Cones of Volumes

Due to the analogy between 3D-OPP's vertices described in terms of their incident manifold or non-manifold edges, and 4D-OPP's edges described in terms of their incident manifold or non-manifold faces, the next logical step is to extend the concept of cones of faces presented in section 3.3 to classify 4D polytopes' edges as manifold or non-manifold.

Faces, edges and vertices, as boundary elements for 4D polytopes, may be either manifold or non-manifold elements. [Coxeter,63] has stated that a manifold face is adjacent to exactly two volumes, and now we suggest that a manifold edge is the common edge (apex) of only one **hyper-cone of volumes**. Conversely, it has been suggested that a non-manifold face is adjacent to more than two volumes [Aguilera,02], and now we suggest that a non-manifold edge is the common edge (apex) of more than one **hyper-cone of volumes**.

Using the concept of hyper-cones of volumes, it is easy to extend the algorithm for obtaining the vertex classification for 3D-OPP's used for section 3.3, to allow us classifying an edge, as manifold or non-manifold, in any 4D polytope or 4D pseudo-polytope. The algorithm will be defined with the following steps (1 to 6):

1 Get the set of Π_3's that are incident to edge *A* (a Π_1).
2 From the set of Π_3's select one of them.
3 The selected Π_3 has two Π_2's that are incident to *A*, get one of them and label it as *START* and *ANOTHER*.

4 Repeat
4.1 If the number of Π_3's to *ANOTHER* is more than one, then A is a non-manifold Π_1. End.
4.2 The *ANOTHER* Π_2 is common to another Π_3, find it.
4.3 The Π_3 has another Π_2 that is common to *A*, find it and label it as *ANOTHER*.
4.4 Until *START = ANOTHER* (it has been found a hyper-cone of volumes).
5 If there are more Π_3's to analyze then *A* is non-manifold (there are more hyper-cones of volumes). End.
6 Otherwise, *A* is manifold (*A* is the common edge of only one hyper-cone of volumes). End.

See the next code for an implementation of the algorithm in a high level language, Java [Gosling,00]. For this code, an edge "e" is evaluated to classify it as manifold or non-manifold. If the edge is manifold (and for instance, the apex of only one hyper-cone of faces), then the method returns true, otherwise, the edge is non-manifold (it is the apex of more than one hyper-cone of faces) and it returns false.

```
boolean isManifoldEdge(Polytope p, Edge e)
{
  Volume volumes[ ]=getVolumesIncidentToEdge(p,e);
  Volume v = selectAndRemoveVolume(volumes);
  Face f1 = getIncidentFaceToEdge(v, e);
  Face start = f1;
  Face another = f1;
  do{ //do-while begins
    if(getNumberOfIncidentVolumesToFace
                                    (volumes,another) > 1)
          return false;
    v=removeVolumeIncidentToFace(volumes,another);
    another=getIncidentFaceToEdge(v,another,e);
  } while(another != start); //do-while ends
  if(volumes.length > 0) return false;
  return true;
}
```

4D edge	Classification through hyper-cones of volumes	3D vertex	Classification through cones of faces
E3	Manifold	V3	Manifold
E4	Manifold	V4	Manifold
E4N1	Non-manifold	V4N1	Non-manifold
E4N2	Non-manifold	V4N2	Non-manifold
E5N	Non-manifold	V5N	Non-manifold
E6	*Non-manifold* when 2 or 6 hypervolumes are incident to it. *Manifold* when 4 hypervolumes are incident to it.	V6	*Non-manifold* for configurations e and t. *Manifold* for configuration l.
E6N1	Non-manifold	V6N1	Non-manifold
E6N2	Non-manifold	V6N2	Non-manifold

Table 4. 4D-OPP's edges classifications and their analogy with 3D-OPP's vertices.

4. Results

Using the algorithm presented in section 3.4 over the possible edges in 4D-OPP's we have that the edges' classifications are analogous to the 3D-OPP's vertices' classifications. Table 4 shows the edges' classifications given by the extended algorithm and their analogous 3D results.

4.1 Classifying the Π_{n-3} in nD Polytopes Through its nD Hyper-Cones of Π_{n-1}'s

Due to the analogy found between 3D vertices and 4D edges with the extension of the concept of cones of faces, is feasible to generalize the algorithm presented in section 3.4 to classify the Π_{n-3} as manifold or non-manifold in nD polytopes through their nD hyper-cones of Π_{n-1}'s. The proposed general algorithm is the following:

1 Get the set of Π_{n-1}'s that are incident to Π_{n-3} *A*.
2 From the set of Π_{n-1}'s select one of them.
3 The selected Π_{n-1} has two Π_{n-2}'s that are incident to Π_{n-3} A, get one of them and label it as *START* and *ANOTHER*.
4 Repeat
4.1 If the number of incident Π_{n-1}'s to *ANOTHER* is more than one, then *A* is a non-manifold Π_{n-3}. End.
4.2 The *ANOTHER* Π_{n-2} is common to another Π_{n-1}, find it.
4.3 The Π_{n-1} has another Π_{n-2} that is common to *A*, find it and label it as *ANOTHER*.
4.4 Until *START = ANOTHER* (it has been found a nD hyper-cone of Π_{n-1}'s).
5 If there are more Π_{n-1}'s to analyze then Π_{n-3} *A* is non-manifold (there are more nD hyper-cones of Π_{n-1}'s). End.
6 Otherwise, Π_{n-3} *A* is manifold (A is the common Π_{n-3} of only one nD hyper-cone of Π_{n-1}'s). End.

4.2 The Eight Types of Π_{n-3}'s in nD Orthogonal Pseudo-Polytopes

Due to the analogy between vertices in 3D-OPP's and edges in 4D-OPP's (see Table 4), we can extend their properties to propose the eight types of Π_{n-3}'s in nD Orthogonal Pseudo-Polytopes. Such Π_{n-3}'s will be referred as $\Pi_{n-3}3$, $\Pi_{n-3}4$, $\Pi_{n-3}4N1$, $\Pi_{n-3}4N2$, $\Pi_{n-3}5N$, $\Pi_{n-3}6$, $\Pi_{n-3}6N1$ and $\Pi_{n-3}6N2$. In this nomenclature (just as the used in sections 3.1 and 3.2) "Π_{n-3}" indicates the (n-3)-dimensional element (i.e. vertices in 3D-OPP's and edges in 4D-OPP's), the first digit shows the number of incident Π_{n-2} (i.e. edges in 3D-OPP's and faces in 4D-OPP's), the "N" is present if at least one non-manifold Π_{n-2} is incident to the Π_{n-3} and the second digit is included to distinguish between two different types that otherwise could receive the same name.

For each Π_{n-3} we can expect the following properties:

- $\Pi_{n-3}3$: all three incident Π_{n-2}'s are manifold and perpendicular to each other.
- $\Pi_{n-3}4$: all four incident Π_{n-2}'s are manifold, they lie on a hyperplane, and can be grouped in two couples of *co-hyperplanar* Π_{n-2}'s.
- $\Pi_{n-3}4N1$: three of its four incident Π_{n-2}'s are perpendicular to each other and also manifold ones, while the fourth is non-manifold and *co-hyperplanar* to one of the other three.
- $\Pi_{n-3}4N2$: two of its four incident Π_{n-2}'s are manifold and *co-hyperplanar*, while each of its other two is non-manifold and perpendicular to the other three.
- $\Pi_{n-3}5N$: four of its five incident Π_{n-2}'s are manifold and lie in a hyperplane, while the fifth is non-manifold and perpendicular to the rest of them.
- $\Pi_{n-3}6$: all six incident Π_{n-2}'s are manifold.
- $\Pi_{n-3}6N1$: three of its six incident Π_{n-2}'s are perpendicular to each other and also manifold ones, while each of its remaining three Π_{n-2}'s is non-manifold and *co-hyperplanar* to one of the first three.
- $\Pi_{n-3}6N2$: all of its six incident Π_{n-2}'s are non-manifold.

5. Future Work

The results of this article are being used in studying the extension for the Extreme Vertices Model (EVM) [Aguilera,98] to the fourth dimensional space (EVM-4D). The EVM-4D will be a representation model for 4D Orthogonal Polytopes that will allow queries and operations over them. However, the fact related to a model purely geometric (four geometric dimensions) is not restrictive for our research, because it will be used under geometries as the 4D spacetime. The first main application for the EVM-4D covers the visualization and analysis for multidimensional data under the context of a Geographical Information System (GIS).

6. Acknowledgements

This work is supported by the "Consejo Nacional de Ciencia y Tecnología" (CONACyT), México (Project W35804-A).

7. References

[Aguilera,98] Aguilera,A. Orthogonal Polyhedra: Study and Application. PhD Thesis, Universitat Politècnica de Catalunya. 1998.

[Aguilera,02] Aguilera Ramírez, Antonio & Pérez Aguila, Ricardo. Classifying the n-2 Dimensional Elements as Manifold or Non-Manifold for n-Dimensional Orthogonal Pseudo-Polytopes. Proceedings of the 12th International Conference on Electronics, Communications and Computers CONIELECOMP 2002, pp. 59-63. February 24 to 27, 2002. Acapulco, Guerrero, México.

[Coxeter,63] Coxeter, H.S.M. Regular Polytopes. Dover Publications, Inc., New York. 1963.

[Feiner,90] Feiner, S., and Beshers, C. Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds. In Proceedings of The ACM Symposium on User Interface Software and Technology, ACM Press / Addison-Wesley, 76-83. 1990.

[Gosling,00] Gosling, J., Joy, B., Steele, G., and Bracha G. The Java Language Specification. Second Edition. Sun Microsystems, 2000.

[Herman,98] Herman, G.T. Geometry of Digital Spaces. Birkhäuser, Boston. 1998.

[Joan-Arinyo,88] Joan-Arinyo, R. Aportació a l'Estudi de la Transcodificació dels Models de Fronteres i Arbres Estesos en Model de Geometria Constructiva de Sòlids. PhD Thesis (written in Catalan). Escola Tècnica Superior d'Enginyers Industrials de Barcelona, Universitat Politècnica de Catalunya. 1988.

[Lees,99] Lees, J. M. Geotouch: Software for Three and Four Dimensional GIS in the Earth Sciences. Computers & Geosciences 26, 7, 751-761. 1999.

[Pérez,03a] Pérez Aguila, Ricardo & Aguilera Ramírez, Antonio. Extreme Edges: A New Characterization for 1-Dimensional Elements in 4D Orthogonal Pseudo-Polytopes. Submitted for Evaluation to the 13th International Conference on Electronics, Communications and Computers CONIELECOMP 2003. To be held february 24 to 26, 2003, Cholula, Puebla, México.

[Pérez,03b] Pérez Aguila, Ricardo. Presenting the 'Test-Box' Heuristic for Determining the Configurations for the n-Dimensional Orthogonal Pseudo-Polytopes. Submitted for Evaluation to the 13th International Conference on Electronics, Communications and Computers CONIELECOMP 2003. To be held february 24 to 26, 2003, Cholula, Puebla, México.

[Preparata,85] Preparata F. P., and Shamos, M. I. Computational Geometry: an Introduction. Springer-Verlag. 1985.

[Rossignac,91] Rossignac, J.R., and Requicha, A.A. Constructive Non-Regularized Geometry. Computer Aided Design 23, 1, 21-32. 1991.

[Tang,91] Tang, K., and Woo, T. Algorithmic Aspects of Alternating Sum of Volumes. Part 2: Non-Convergence and Its Remedy. Computer-Aided Design 23, 6, 435-443. 1991.

[Weeks,02] Weeks, J. R. The Shape of Space. Second Edition. Marcel Dekker, Inc., New York. 2002.

[Wegenkittl,97] Wegenkittl, R., Löffelmann, H., and Gröller, E. Visualizing the Behavior of Higher Dimensional Dynamical Systems. In Proceedings of the IEEE Visualization 1997, 119-125. 1997.

Presentando una Metodología para la Visualización del Desenvolvimiento de un Hipercubo 4D

Antonio Aguilera Ricardo Pérez-Aguila
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas – Puebla (UDLAP)
Cholula, Puebla 72820, México.

Abstract

Este artículo presenta un método para desenvolver al hipercubo y formar la cruz tridimensional (teseracto) que corresponde al hiperaplanamiento de su frontera. El envolver el hipercubo implicará aplicar el método a la inversa. También se presenta un método para visualizar dichos procesos. Las transformaciones a aplicar incluyen rotaciones alrededor de planos (propias del espacio 4D). Dichos procesos son visualizados a través de un sistema de animación por computadora.

Enviado a revisión a la 2da. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2003). A llevarse a cabo del 31 de julio al 2 de agosto de 2003, Orlando, Florida, EE.UU.

Presentando una Metodología para la Visualización del Desenvolvimiento[1] de un Hipercubo 4D

Antonio Aguilera Ricardo Pérez-Aguila
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA)
Universidad de las Américas - Puebla
Cholula, Puebla 72820, México

RESUMEN

Este artículo presenta un método para desenvolver al hipercubo y formar la cruz tridimensional (teseracto) que corresponde al hiperaplanamiento de su frontera. El envolver el hipercubo implicará aplicar el método a la inversa. También se presenta un método para visualizar dichos procesos. Las transformaciones a aplicar incluyen rotaciones alrededor de planos (propias del espacio 4D). Dichos procesos son visualizados a través de un sistema de animación por computadora.

Palabras Claves: Modelado 4D, Animación 4D, Geometría Computacional, Interrogaciones y Razonamiento Geométrico.

1. INTRODUCCIÓN

Coxeter [5], Rucker [12], Kaku [9], Robbin [10] y Banchoff [2] inician sus introducciones al estudio del espacio 4D aplicando tres métodos de visualización sobre el hipercubo: observación de sus sombras (proyección), sus intersecciones con el espacio 3D y a través de sus "*unravelings*".

Examinar las sombras de un politopo consiste en que si es posible hacer dibujos de sólidos 3D cuando éstos son proyectados sobre un plano, entonces es posible hacer dibujos o modelos tridimensionales de los politopos 4D cuando estos son proyectados sobre un hiperplano [5].

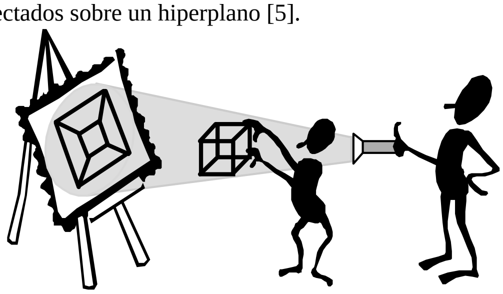


Figura 1. Proyección de un cubo sobre un plano.

En el primer caso, y siguiendo la analogía presentada en *"Flatland"* [1] si un ser 3D quiere mostrar un cubo a un ser 2D (un "*flatlander*"), entonces el primero deberá proyectar la sombra del cuerpo sobre el plano en el que el *flatlander* habita. En este caso, la figura proyectada podría ser, por ejemplo, un cuadrado dentro de otro cuadrado (figura 1) llamada proyección central.

Para el caso en que un ser 4D quisiera mostrarnos un hipercubo, él debe proyectar la sombra de éste sobre el espacio 3D en que vivimos. El cuerpo proyectado podría ser un cubo dentro de otro cubo [9] (figura 2) también llamada proyección central. Sabemos que un cubo proyectado sobre un plano es sólo una representación aproximada. Análogamente, el hipercubo proyectado sobre el espacio 3D es también una representación aproximada del real.

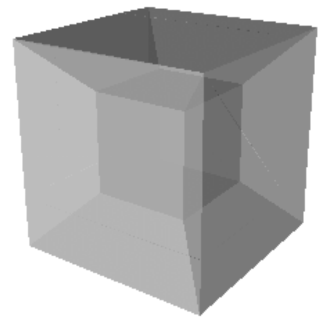


Figura 2. Proyección de un hipercubo en el espacio 3D.

El método de los "*unravelings*" consiste en que si un cubo puede ser desenvuelto en una cruz bidimensional compuesta por las seis caras que forman su frontera (figura 3) entonces, y en forma análoga, un hipercubo puede ser también desenvuelto en una cruz tridimensional compuesta por los ocho cubos que forman su frontera [9]. C. H. Hinton nombró a esta cruz tridimensional ***teseracto*** (figura 4).

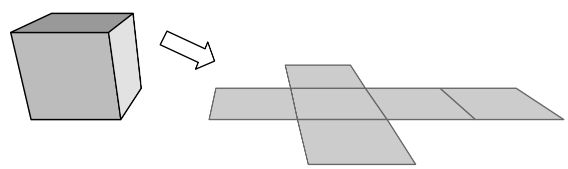


Figura 3. Desenvolvimiento de un cubo.

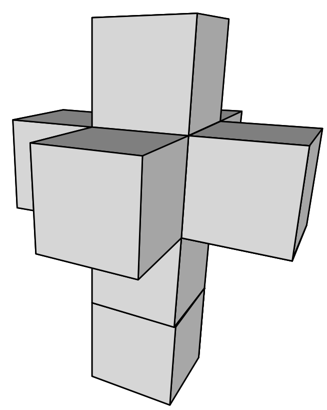


Figura 4. El hipercubo desenvuelto (teseracto).

Un *flatlander* visualizará la cruz bidimensional pero no tendrá la capacidad para ensamblarla nuevamente en un cubo (aún cuando contase con las instrucciones específicas), debido a

[1] Todas las referencias consultadas utilizan el verbo inglés "*unravel"* para indicar la acción de hacer coincidir los volúmenes (o las caras) de un hipercubo 4D (o un cubo) con un hiperplano (o un plano). Los términos en castellano utilizados en este artículo para hacer referencia a tal acción serán *desenvolver* o *hiperaplanar* (4D). También el término "*unravelings"* deberá entenderse como el conjunto de volúmenes (o caras) de un hipercubo (o un cubo) a los que ya fue aplicada la acción de desenvolvimiento.

que es necesaria la traslación de sus caras correspondientes en dirección de la tercera dimensión y la rotación alrededor de un eje (transformaciones físicamente imposibles en el espacio 2D). Pero durante el proceso de ensamblado, este *flatlander* sí podrá visualizar la proyección de las caras del cubo sobre el espacio 2D en que habita.

Por analogía, nosotros podremos visualizar la cruz tridimensional pero no tendremos la capacidad para ensamblarla nuevamente en un hipercubo, debido a que es necesaria la traslación de sus cubos frontera en dirección de la cuarta dimensión y la rotación alrededor de un plano (transformaciones físicamente imposibles en el espacio 3D).

Analizar el hipercubo es también interesante debido a que puede ser realizado usando el recurso de la analogía con el cubo y las visualizaciones descritas antes. Hilbert [7] ha determinado que un hipercubo está formado dieciséis vértices, veinticuatro caras y por ocho cubos (que también son llamados celdas o volúmenes). Coxeter [4] también agrega que cada cara es compartida por dos cubos que no se encuentran en el mismo espacio tridimensional dado que forman un ángulo recto a través de una rotación alrededor del plano de soporte de la cara compartida. Estas propiedades pueden ser claramente visibles a través de la proyección del hipercubo propuesta por Claude Bragdon (figura 5) (véase [11] para un análisis sobre la obtención de esta proyección).

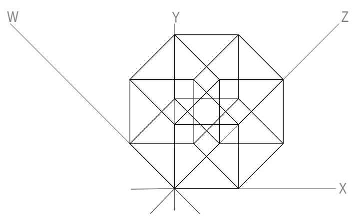


Figura 5. El hipercubo con proyección de Bragdon.

2. PROBLEMA

Kaku [9] y Banchoff [2] describen con detalle el modelo de representación del hipercubo a través de sus "*unravellings*" y mencionan la incapacidad física de un ser 3D para envolverlo nuevamente debido a las transformaciones que se requieren. Kaku [9] y Banchoff [2] también describen que si presenciáramos el proceso de envolvimiento, siete de los ocho cubos que forman la cruz desaparecerían repentinamente debido a que ya se han movido hacia la cuarta dimensión. Sin embargo no proporcionan una metodología que indique las transformaciones y sus parámetros necesarios para ejecutar dicho procedimiento. A pesar de dicha incapacidad nuestra, lo que sí podemos es visualizar una proyección de los cubos de la frontera del hipercubo en nuestro espacio 3D durante su desenvolvimiento y ensamblado.

Este artículo presenta un método para desenvolver al hipercubo y formar la cruz tridimensional (teseracto) que corresponde al hiperaplanamiento de su frontera (figura 6). El envolver el hipercubo implicará aplicar el método a la inversa. Las transformaciones a aplicar incluyen rotaciones alrededor de planos. Dicho proceso podrá ser visualizado a través de un sistema de animación por computadora.

3. METODOLOGÍA PARA DESENVOLVER UN HIPERCUBO

En primer lugar habrán de tomarse las siguientes consideraciones a fin de hacer más fácil el proceso:

- La posición del hipercubo en el espacio 4D.
- Seleccionar un hiperplano (subespacio 3D inmerso en el hiperespacio) hacia el que los volúmenes serán dirigidos.
- Establecer ángulos de giro que garanticen que todos los volúmenes quedarán completamente inmersos en el hiperplano seleccionado.
- Durante su movimiento hacia el hiperplano seleccionado, todas los volúmenes deberán mantener una relación de adyacencia de cara con otro volumen.

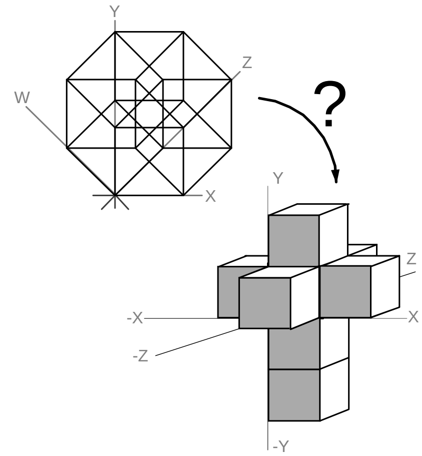


Figura 6. El paso del hipercubo al teseracto.

La posición del hipercubo en el espacio 4D es esencial ya que de ella dependerán los planos de rotación alrededor de los cuales deberán girar los volúmenes para ser posicionados sobre un hiperplano. Por lo tanto se determinará que uno de los vértices del hipercubo coincida con el origen, que seis de sus caras coincidan cada una con alguno de los planos XY, YZ, ZX, XW, YW y ZW y que todas las coordenadas sean positivas (véase [2] para la metodología para obtener las coordenadas de los vértices del hipercubo). Las coordenadas a usar se presentan en la tabla 1 (cada vértice es numerado arbitrariamente).

Vértice	***X***	***Y***	***Z***	***W***
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

Tabla 1. Las coordenadas del hipercubo a desenvolver.

Así como la posición del hipercubo en el espacio 4D tiene relación con los planos de rotación a utilizar, también la tendrá el hiperplano seleccionado sobre el que los volúmenes finalmente serán posicionados. Si se observan las coordenadas de los vértices del hipercubo, se encontrará que ocho de ellas presentan W=0, esto se traduce en que uno de los volúmenes del hipercubo (el formado por los vértices 0-1-2-3-4-5-6-7) tiene por hiperplano de soporte a W=0. Seleccionar el hiperplano W=0 es conveniente ya que uno de los volúmenes ya esta "naturalmente inmerso" en el espacio 3D y por lo tanto no requerirá transformaciones posteriores.

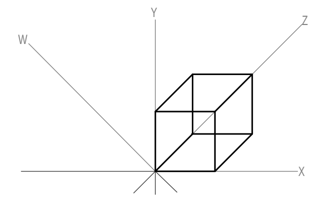
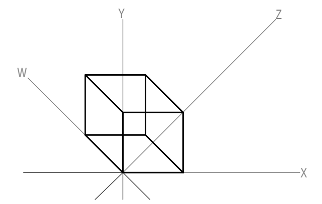
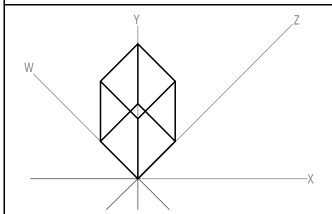
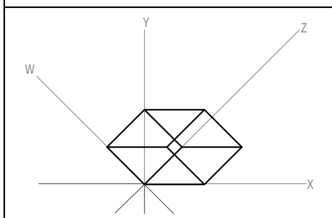
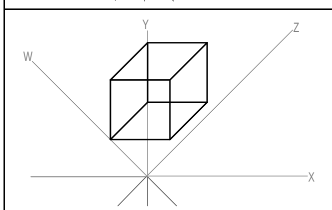
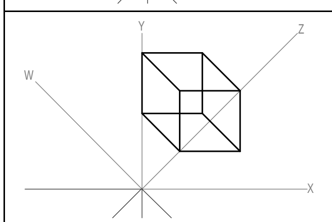
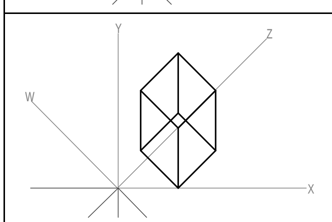
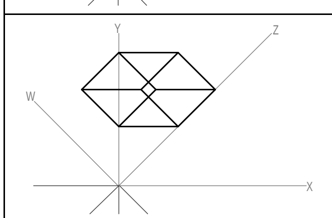
Volumen	*Etiqueta y Vértices*
	Volumen A (0-1-2-3-4-5-6-7)
	Volumen B (0-1-2-3-8-9-10-11)
	Volumen C (0-2-4-6-8-10-12-14)
	Volumen D (0-1-4-5-8-9-12-13)
	Volumen E (8-9-10-11-12-13-14-15)
	Volumen F (4-5-6-7-12-13-14-15)
	Volumen G (1-3-5-7-9-11-13-15)
	Volumen H (2-3-6-7-10-11-14-15)

Tabla 2. Los ocho volúmenes del hipercubo.

Ahora también es conveniente identificar los volúmenes que forman al hipercubo a través de sus vértices y asignarles una etiqueta para futuras referencias. Hasta ahora ya se tiene un volumen identificado, el formado por los vértices 0-1-2-3-4-5-6-7 y será llamado volumen A. Véase la tabla 2.

Dado que el volumen A ya había sido descrito como "naturalmente inmerso" en el espacio 3D y por lo tanto no requerirá de transformaciones, es por lo tanto el volumen que ocupará la posición central de la "cruz" y será llamado en lo sucesivo el "volumen central".

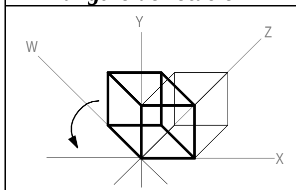
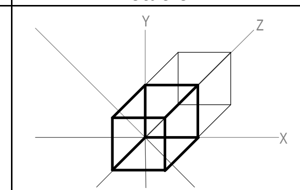
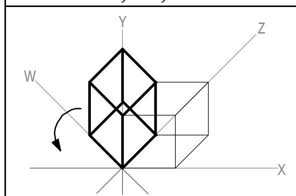
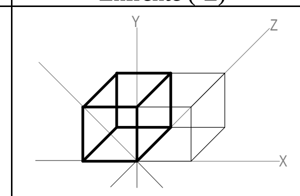
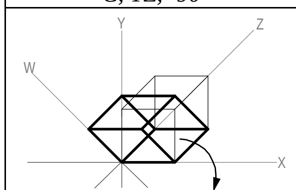
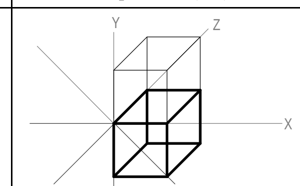
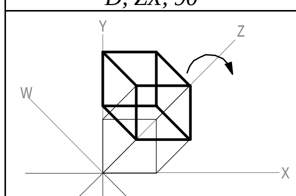
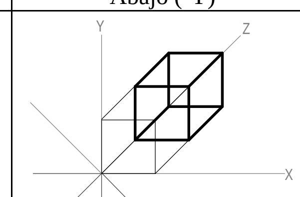
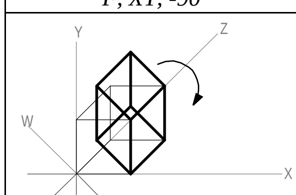
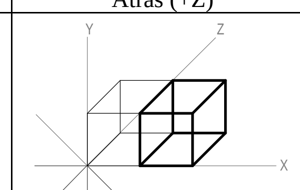
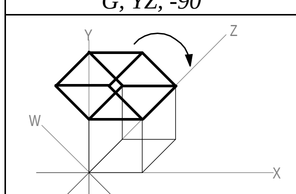
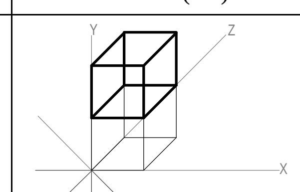
Volumen adyacente (previo a la rotación), plano y ángulo de rotación	*Posición en el espacio 3D y en el teseracto después de la rotación*
 *B, XY, 90°*	 Enfrente (-Z)
 *C, YZ, -90°*	 Izquierda (-X)
 *D, ZX, 90°*	 Abajo (-Y)
 *F, XY, -90°*	 Atrás (+Z)
 *G, YZ, -90°*	 Derecha (+X)
 *H, ZX, -90°*	 Arriba (+Y)

Tabla 3. Transformaciones aplicadas a los volúmenes adyacentes.

De los volúmenes restantes, aquellos que tengan adyacencia de cara con el volumen central podrían ser rotados con facilidad hacia nuestro espacio 3D debido a que su plano de rotación es claramente identificable. Estos volúmenes rotarán alrededor del plano de soporte de la cara que compartan con el cubo central y que serán llamados "volúmenes adyacentes". Los volú-menes adyacentes son B, C, D, F, G y H. El volumen restante, E, será llamado "volumen satélite" y se tratará más adelante.

Todos los volúmenes adyacentes girarán ángulos rectos, así se garantiza que su coordenada W sea igual a cero, pero es importante tener en cuenta la dirección de giro ya que de lo con-trario los volúmenes podrían una vez rotados coincidir con el volumen central. Los planos de rotación y la dirección para cada volumen adyacente son presentados en la tabla 3 (en las imáge-nes se incluye también al volumen central sólo para referenciar la posición inicial y final del volumen correspondiente).

En este punto ya tenemos a 7 de los 8 volúmenes del hipercubo colocados en su posición final (los volúmenes A, B, C, D, F, G y H). El volumen que ha de presentar la serie de transformaciones más compleja es el E, esto es por las siguientes dos particularidades:

- Su hiperplano de soporte es paralelo al del volumen central, por lo tanto no existe ningún tipo de adyacencia con éste (de ahí que no fue designado como volumen adyacente).
- De las posiciones por ocupar en la "cruz" aún falta aquella que corresponde al volumen más alejado del volumen central (en la parte inferior, según la figura 4). El volumen que ocupará esta posición será el E, es por esta razón por la que fue llamado con anterioridad volumen satélite.

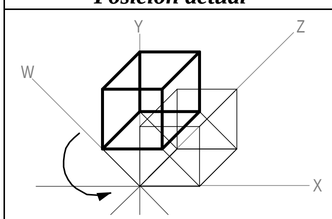
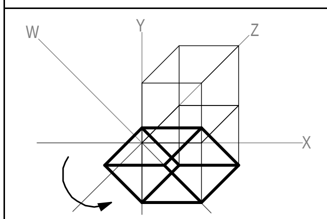
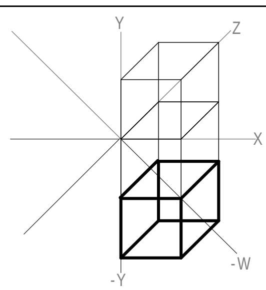
Posición actual	***Transformación***
	Rotación de los volúmenes D y satélite alrededor del plano ZX (90°).
	Volumen D en su posición final. Rotación del volumen satélite 90° alrededor de la cara compartida con el volumen D (plano paralelo a ZX).
	Volumen satélite en su posición final (parte inferior de la cruz sobre el eje -Y).

Tabla 4. Transformaciones asociadas al volumen satélite (volumen E).

Al inicio del documento se menciona la necesidad de que los volúmenes durante su movimiento hacia el hiperplano seleccionado deberán mantener una relación de adyacencia de cara con otro volumen. Los volúmenes B, C, D, F, G y H son los seis que comparten una cara con el volumen central (el cual se mantiene estático durante todo el proceso). Para determinar las transformaciones que se aplicarán al volumen satélite, es necesario determinar con qué volumen compartirá una cara. El volumen central ya se encuentra descartado, y de los restantes cualquiera puede serlo. En este trabajo, el volumen D será seleccionado como aquel con el que el volumen satélite compartirá una cara durante el hiperaplanamiento.

Para el volumen D ya se había determinado su plano de rotación y la dirección correspondientes (plano ZX, 90°) que lo llevarán a su posición final. El volumen satélite inicialmente tendrá también estos parámetros de movimiento. Esta es una forma de asegurar que ambos volúmenes compartan la cara correspondiente.

Cuando el volumen D ha finalizado sus movimientos y ha quedado en su posición final, el hiperplano de soporte del volumen satélite será perpendicular al hiperplano seleccionado y la cara compartida será paralela al plano ZX. El movimiento adicional que deberá aplicarse al volumen satélite será un giro de 90° alrededor del plano representado por la cara compartida.

La serie de movimientos a ejecutar sobre el volumen satélite se resumen en la tabla 4 (los volúmenes central y D se muestran también).

Ahora han sido determinadas las transformaciones con las que el hipercubo será desenvuelto. Para envolverlo sólo habrá que aplicar el proceso presentado pero a la inversa (considerando que la dirección de los ángulos sea contraria a la usada para desenvolver).

4. IMPLEMENTACIÓN

Rotaciones en el Espacio 4D

Banks [3] y Hollasch [8] han identificado que si en el espacio 2D una rotación es dada alrededor de un punto y en el espacio 3D es dada alrededor de una línea, entonces en el espacio 4D, en forma análoga, deberá estar dada alrededor de un plano.

Hollasch [8] considera que las rotaciones en el espacio 3D deben ser consideradas como rotaciones paralelas a un plano 2D en lugar de rotaciones alrededor de un eje. Hollasch [8] apoya esta idea considerando que dado un origen de rotación y un punto destino en el espacio 3D, el conjunto de todos los puntos rotados para una matriz de rotación dada coinciden en un solo plano, el cual es llamado el plano de rotación. Además, el eje de rotación en el espacio 3D coincide con el vector normal del plano de rotación. El concepto de plano de rotación es consistente con el espacio 2D debido a que todos los puntos rotados coinciden en el mismo y único plano. Finalmente, usando las ideas anteriores, Hollasch [8] construye las seis matrices de rotación 4D básicas alrededor de los planos principales en el espacio 4D (los planos XY, YZ, XZ, XW, YW y ZW) basado en el hecho de que sólo dos coordenadas cambian para una rotación dada (las coordenadas cambiantes correspon-den al plano de rotación).

Usando estas ideas, Duffin [6] generaliza el concepto de rotación en un espacio nD ($n \geq 2$) como la rotación de un eje Xa en dirección hacia un eje Xb. El plano descrito por los ejes Xa y Xb es lo que Hollasch [8] definió como plano de rotación. Duffin [6] presenta la siguiente matriz general de rotación:

$$R_{ab}(\theta) = \left[r_{ij} \left| \begin{array}{ll} r_{ii} = 1 & i \neq a, i \neq b \\ r_{aa} = \cos\theta & \\ r_{bb} = \cos\theta & \\ r_{ab} = -\sin\theta & \\ r_{ba} = \sin\theta & \\ r_{ij} = 0 & elsewhere \end{array} \right. \right]$$

La matriz $R_{ab}(\theta)$ es una matriz identidad excepto en las intersecciones de las columnas a y b con los renglones a y b. Debido a que en un espacio *n*D existen C(n,2) planos principales, este número es precisamente el número de rotaciones principales (y básicas) para tal espacio.

A partir de estos conceptos, se debe considerar que una rotación puede ser referenciada usando dos notaciones: usando los ejes que describen el plano de rotación o usando los ejes que describen el subespacio (n-2)D que se encuentra fijo durante la rotación. En este documento las rotaciones en el espacio 4D han sido referenciadas usando la segunda notación.

Proyecciones 4D-3D-2D

Banks [3] establece que las mismas técnicas utilizadas para la proyección de objetos 3D sobre planos 2D pueden ser aplicadas para la proyección de politopos 4D sobre hiperplanos 3D (nuestro espacio 3D por ejemplo). Entonces se tendrá que una proyección paralela 4D-3D (o bien, la eliminación de la coordenada W de los puntos del politopo) es:

$$P(x, \quad y, \quad z, \quad w) \mapsto P'(x, \quad y, \quad z)$$

Una proyección perspectiva 4D-3D se define cuando el centro de proyección se encuentra sobre el eje W a una distancia *pw* del origen. Si el hiperplano de proyección es W = 0 entonces se tendrá que un punto P será proyectado como:

$$P(x \quad y \quad z \quad w) \mapsto P'\left(\frac{x \cdot pw}{pw - w}, \quad \frac{y \cdot pw}{pw - w}, \quad \frac{z \cdot pw}{pw - w}\right)$$

Debido a que una proyección 4D-3D producirá un volumen como la "sombra" de un politopo 4D, Hollasch [8] considera válido procesar tal volumen con alguna de las proyecciones 3D-2D (paralela o perspectiva) para ser finalmente proyectado en una pantalla de computadora. De esta manera, se tendrán cuatro posibles proyecciones 4D-3D-2D:

- Proyección Perspectiva 4D-3D - Proyección Perspectiva 3D-2D.
- Proyección Perspectiva 4D-3D - Proyección Paralela 3D-2D.
- Proyección Paralela 4D-3D - Proyección Perspectiva 3D-2D.
- Proyección Paralela 4D-3D - Proyección Paralela 3D-2D.

Por ejemplo, el hipercubo presentado en la Figura 1 tiene aplicadas las proyecciones perspectiva 4D-3D y perspectiva 3D-2D.

5. RESULTADOS

En la Tabla 5 se presentan algunas fases de la secuencia del desenvolvimiento del hipercubo. En las imágenes 1 a 6 las rotaciones aplicadas son ±0°, ±15°, ±30°, ±45°, ±60° y ±75° (el sentido de la rotación depende del volumen adyacente). En la imagen 7, la rotación aplicada es ±82°; el volumen satélite se aprecia como un plano -un efecto producido por la proyección 4D-3D aquí seleccionada. En la imagen 8, la rotación aplicada es ±90°; los volúmenes adyacentes finalizan sus movimientos. En las imágenes 9 a 14, el volumen satélite se mueve independientemente y las rotaciones aplicadas respectivamente son +15°, +30°, +45°, +60°, +75° y +90°.

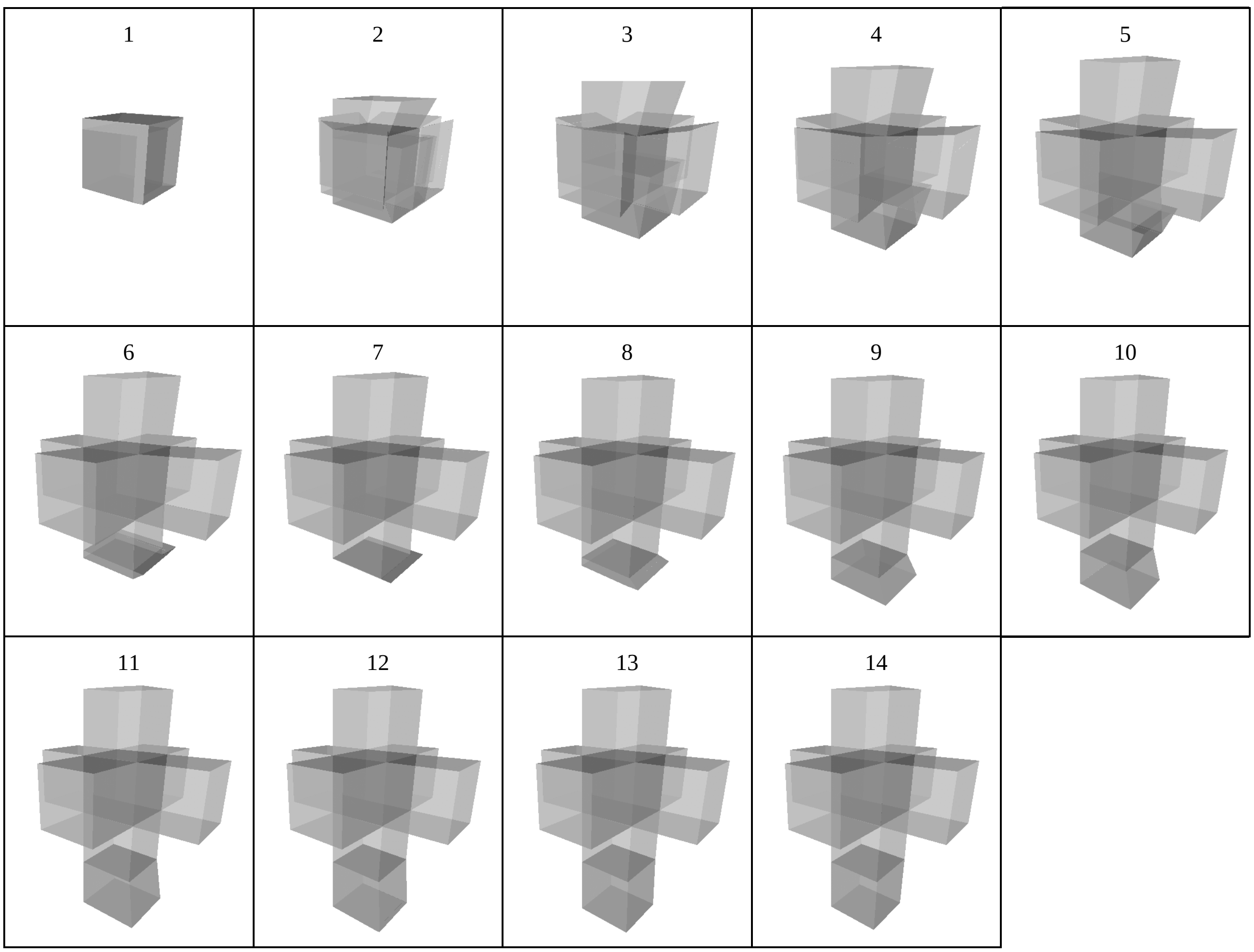


Tabla 5. Desenvolviendo al Hipercubo 4D (véase el texto para los detalles).

Actualmente, el resultado obtenido en esta investigación es usado eficazmente como material didáctico en la Universidad de las Américas - Puebla, México.

6. TRABAJO FUTURO

Observando los *unravelings* para un cuadrado (un cubo 2D), el cubo y el hipercubo 4D; podemos generalizar al ***hiper-teseracto n-dimensional*** ($n \geq 1$) como el resultado del desenvolvimiento de un hipercubo (n+1)-dimensional con las siguientes propiedades:

- El hipercubo (n+1)-dimensional tendrá 2(n+1) celdas n-dimensionales sobre su frontera.
- Una celda central permanecerá estática durante el proceso de desenvolvimiento/ envolvimiento.
- 2(n+1)-2 celdas serán adyacentes a la celda central. Todas las celdas compartirán una celda (n-1)-dimensional con la celda central.
- Una celda satélite no será adyacente a la celda central debido a que sus hiperplanos de soporte son paralelos. Ésta será adyacente a cualquiera de las celdas adyacentes (compartirá una celda (n-1)-dimensional con la celda adyacente seleccionada).
- Todas las celdas adyacentes y satélite durante el proceso de desenvolvimiento/ envolvimiento rotarán ±90° alrededor del hiperplano de soporte de las celdas (n-1)-dimensionales compartidas.

Por ejemplo, el hiper-teseracto 4D es el resultado del desenvolvimiento de un hipercubo 5D. El hiper-teseracto 4D estará compuesto por 10 hiper-volúmenes, uno de ellos será el hiper-volumen central (estático), ocho serán adyacentes al hiper-volumen central (comparten un volumen) y el último será el hiper-volumen satélite (éste comparte un volumen con cualquiera de los volúmenes adyacentes). Véase la Figura 7. Los hiper-volumenes adyacentes y satélite rotarán alrededor de un volumen o un hiperplano durante el proceso de desenvolvimiento/envolvimiento.

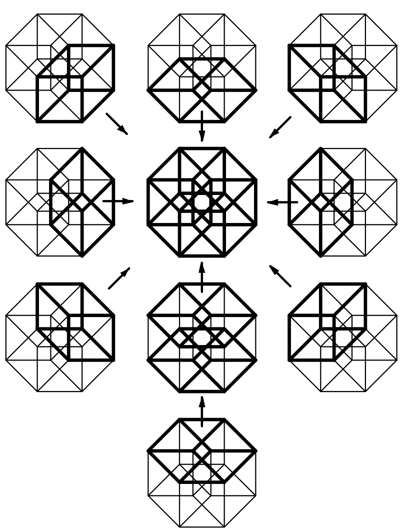


Figura 7. Las posibles relaciones de adyacencia entre el hiper-volumen central y los hiper-volúmenes adyacentes y satélite que formarán al hiper-teseracto 4D.

En este trabajo se ha propuesto un método para el desenvolvimiento del hipercubo 4D y obtención del teseracto. También se ha propuesto una generalización para describir las propiedades del hiper-teseracto n-dimensional, el resultado del desenvolvimiento de un hipercubo (n+1)-dimensional. En el espacio 5D las rotaciones tienen lugar alrededor de un volumen, mientras que en el espacio 6D tienen lugar alrededor de un hiper-volumen y así sucesivamente. Esta es una de las direcciones a seguir en nuestra investigación a fin de obtener los parámetros necesarios para llevar a efecto el desenvolvimiento del hipercubo 5D. Además, otra dirección a seguir tiene relación con las rotaciones alrededor de planos arbitrarios en el espacio 4D (análogamente a las rotaciones alrededor de ejes arbitrarios en el espacio 3D). Al definir los procedimientos necesarios para la rotación alrededor de planos arbitrarios, la posición del hipercubo puede no ser relevante.

BIBLIOGRAFÍA

[1] Abbott, E. A. Flatland: A Romance of Many Dimensions. New American Library, 1984.

[2] Banchoff, Thomas F. Beyond the Third Dimension. Scientific American Library, 1996.

[3] Banks,David. Interactive Manipulation and Display of Two-Dimensional Surfaces in Four-Dimensional Space. *Proceedings of the 21st annual conference on Computer graphics*, Orlando, FL USA, Pages 327 - 334, July 24 - 29, 1994.

[4] Coxeter, H.S.M. Regular Polytopes. Dover Publications, Inc., New York, 1963.

[5] Coxeter, H.S.M. Fundamentos de Geometría. Editorial Limusa, 1984.

[6] Duffin, K, William, B. Spiders: A new user interface for rotation and visualization of n-dimensional points sets. *Proceedings of the 1994 IEEE Conference on Scientific Visualization.* 1994.

[7] Hilbert, D. & Cohn-Vossen, S. Geometry and the Imagination. Chelsea Publishing Company, 1952.

[8] Hollasch, Steven Richard. Four-Space Visualization of 4D Objects. Thesis for the Master of Science Degree. Arizona State University, 1991.

[9] Kaku, Michio. Hyperspace: A Scientific Odyssey Through Parallel Universes, Time Warps, and the Tenth Dimension. Oxford University Press, 1994.

[10] Robbin, Tony. Fourfield: Computers, Art & The 4th Dimension. Bulfinch Press, 1992.

[11] Rucker, Rudolf V. B. Geometry, Relativity and the Fourth Dimension. Dover Publications, Inc., New York, 1977.

[12] Rucker, Rudy. The Fourth Dimension. Houghton Mifflin Company, Boston, 1984.