Constructing an Interpolatory Subdivision Scheme from Doo-Sabin Subdivision, Project Report: MEC-572

Dasharadhan Mahalingam
ID: 112950875
dasharadhan.mahalingam@stonybrook.edu

Sravana Sumanth Koneru
ID: 112526331
koneru.sravanasumanth@stonybrook.edu

Stony Brook University

May 18, 2020

Abstract

In this project we implement a new **interpolatory subdivision scheme derived from the Doo-Sabin subdivision scheme**[3]. Firstly, it presents the relations among four-point interpolatory subdivision scheme, a cubic B-spline curve subdivision scheme, and the Chaikin's algorithm with uniform quadratic B-spline curves. And by taking these relations to the surface case, an interpolatory surface subdivision scheme from the Doo- Sabin subdivision scheme (a generalization of the Chaikin's algorithm) to surface subdivision is derived. The variable tension parameter can be used to effectively control the resulting limit surface of the proposed subdivision scheme.

Keywords: *Subdivision; Interpolation Subdivision; Approximating Subdivision; Doo-Sabin Subdivision Scheme; Surface Interpolation.*

1 Literature Survey

Subdivision schemes have emerged as popular modelling tool in computer graphics due to their simplicity and expressive power. Subdivision schemes are efficient tools for computer-aided curve and surface design. Most of the important methods for curve design can be approached by subdivision processes. The common subdivision schemes are based on chopping corners of a control polygon, hence they are not interpolatory, and this is a drawback in some applications. A **subdivision scheme** provides the means to reconstruct a smooth surface from a coarse control mesh with arbitrary topology. The subdivision schemes are manily divided into two general classes:

1. **Approximatory Subdivision Scheme:**
   In approximating subdivision schemes, the limit surface approximates the initial surface and after that, the newly generated contour plots are not in the limit surfaces.

2. **Interpolatory subdivision Scheme:**
   In interpolating subdivision schemes, the control points of the original mesh and newly generated control points are interpolated on limit surface.

1.1 Chaikins Algorithm

For subdivision in curves, **George M Chaikin** introduced an algorithm [2] for high speed curve generation.This algorithm is recursive and used only integer addition, one-bit right shifts, complementation and comparisons, and produces a list of new points which constitute the curve. This curve consists of concatenated segments, where each segment is smooth and open. The curve may be arbitrarily complex, that is, it may be smooth or discontinuous, and it may be open, closed, or self intersecting. It is also noted that Chaikin's curve has been shown to be equivalent to a quadratic B-spline curve (a piecewise quadratic Bezier curve) [7] when two new adjacent vertices of chaikins scheme are averaged. However, Chaikin's method avoids the analytical definition of B-splines and provides a simple, elegant curve drawing mechanism.

1.2 Four point interpolatory subdivision scheme

A four point interpolatory subdivision scheme [5] is introduced by N.Dyn, D,Levin and J.A.Gregory. This scheme makes use of the tension parameter to control the shape of the surface. The common subdivision schemes are based on chopping corners of a control polygon, hence they are not interpolatory, and this is overcome by this four-point interpolatory scheme and this also provides good overall control over the curve.

1.3 Catmull-Clark Subdivision scheme

Parallel to the curve subdivisions, developments are done in surface subdivision schemes. The subdivision surfaces were discovered simultaneously by Edwin Catmull and Jim Clark and this scheme is named as **Catmull-Clark Subdivision Scheme**[1]. This was introduced as a generalization of bi-cubic uniform B-spline surfaces with arbitrary topology. This subdivision scheme gained huge popularity for its speed and efficiency. This subdivision scheme was used for the first time in animation in a Pixar's short **Geri's Game** which proved this scheme to be efficient and successful.

1.4 Doo Sabin Subdivision Scheme

One of the other key development in Subdivision surfaces is **Doo-Sabin Subdivision scheme**[4] introduced by Daniel Doo and Macolm Sabin. This subdivision surface is introduced as a generalization of b-quadratic uniform B-spline surfaces. Just like Catmull-Clark subdivision, this scheme also proved to be efficient and fast.

The subdivision surfaces algorithms are recursive in nature. The process starts with a given polygonal mesh. A Refinement Scheme is then applied to this mesh. This process takes that mesh and subdivides it, creating new vertices and new faces. The positions of the new vertices in the mesh are computed based on the positions of nearby old vertices. In some refinement schemes, the positions of old vertices might also be altered (possibly based on the positions of new vertices). This process produces a denser mesh than the original one, containing more polygonal faces. This resulting mesh can be passed through the same refinement scheme again and so on. The limit subdivision surface is the surface produced from this process being iteratively applied infinitely many times. But in reality, we apply the iterations for a limited number of times.

One of the main problems faced in the subdivision schemes is the emergence of ”Artifacts”. For an effective subdivision, it should be possible for the users to provide data and achieve something close to the mental image. The difference between the expected subdivision and the obtained subdivision is called an **Artifact**.[8]. In the paper, the author also provides a way with tension parameters to obtain a artifact free subdivision surface.

2 Introduction

As mentioned above, a **subdivision scheme** provides the means to reconstruct a smooth surface from a coarse control mesh with arbitrary topology. It is also mentioned that, interpolating schemes are required to match the original position of vertices in the original mesh, where as Approximating schemes do not. Approximating schemes can and will adjust these positions as needed. So, approximating schemes are smoother, but the user has less overall control of the outcome. So, the author tries to bring out a new interpolatory scheme from already existing efficient approximating schemes and make them displacement-safe.
For this, first we see the curve schemes, namely the four point interpolatory subdivision scheme, the cubic B-spline curve subdivision scheme, and the Chaikin’s algorithm that generates uniform quadratic B-spline curves. By further generalizing the relations to the surface case, we derive an interpolatory subdivision scheme from the Doo-Sabin subdivision scheme.

3 Theory

3.1 Curve Subdivision schemes

In this section, we study the subdivision schemes for the curve case. Then a variable tension parameter is also introduced for the shape control. The construction makes use of the relations among three subdivision schemes - the Chaikin’s algorithm, B-spline curve subdivision and a four point interpolatory curve subdivision scheme.
Given the initial Control Polygon CP_0 defined by control vertices $(P_0^0, P_1^0, ..., P_i^0, ...)$ the chaikins scheme is defined by:

1. For each pair of control vertices P_i^k, P_{i+1}^k of CP_k, where k is the level of refinement, the two refined veritces are given by:

$$P_{2i}^{k+1} = \frac{3}{4}P_i^k + \frac{1}{4}P_{i+1}^k \tag{1}$$

$$P_{2i+1}^{k+1} = \frac{1}{4}P_i^k + \frac{3}{4}P_{i+1}^k$$

2. Now, we connect the refined vertices $(P_0^{k+1}, P_1^{k+1}, ..., P_i^{k+1}, ...)$ to form the control polygon CP_{k+1} of the next refinement. By averaging the two adjacent refined vertices of the Chaikin's scheme, we obtain the geometric rules for the cubic B-spline curve subdivision as follows:

$$\bar{P}_{2i}^{k+1} = \frac{1}{2}(P_{2i-1}^{k+1} + P_{2i}^{k+1}) = \frac{1}{8}P_{i-1}^k + \frac{3}{4}P_i^k + \frac{1}{8}P_{i+1}^k$$

$$\bar{P}_{2i+1}^{k+1} = \frac{1}{2}(P_{2i}^{k+1} + P_{2i+1}^{k+1}) = \frac{1}{2}(P_i^k + P_{i+1}^k)$$

3. Now we define the moved vertices, in the process of introducing four point interpolatory subdivision scheme. The moved vertices for $P_{2i}^{k+1}, P_{2i+1}^{k+1}$ from the Chaikin's scheme are $Q_{2i}^{k+1}, Q_{2i+1}^{k+1}$ and are given by:

$$Q_{2i}^{k+1} = P_i^k + 8\omega(P_{2i}^{k+1} - P_{2i-1}^{k+1}) = P_i^k + 2\omega(P_{i+1}^k - P_{i-1}^k)$$

$$Q_{2i+1}^{k+1} = P_{i+1}^k - 8\omega(P_{2i+2}^{k+1} - P_{2i+1}^{k+1}) = P_{i+1}^k - 2\omega(P_i^k - P_{i+2}^k)$$

where ω is **tension parameter** and $0 < \omega < \frac{1}{8}$.

4. Now by further averaging two adjacent vertices of the moved vertex array $[Q_i^{k+1}]_i$, we obtain the **the four point interpolatory subdivision scheme:**

$$\bar{Q}_{2i}^{k+1} = \frac{1}{2}(Q_{2i-1}^{k+1} + Q_{2i}^{k+1}) = P_i^k,$$

$$\bar{Q}_{2i+1}^{k+1} = \frac{1}{2}(Q_{2i}^{k+1} + Q_{2i+1}^{k+1}) = (\frac{1}{2} + \omega)(P_i^k + P_{i+1}^k) - \omega(P_{i-1}^k + P_{i+2}^k),$$

In the same way as we produced four point interpolatory scheme from Chaikin's scheme, we obtain a new interpolatory subdivision scheme from Doo-Sabin subdivision by averaging the moved vertices of new faces of the Doo-Sabin scheme.

3.1.1 Variable tension parameter for shape control

For obtaining a displacement safe scheme [6], we introduce a variable local tension parameter in the subdivision scheme that is dependent to local vertices by defining the moved vertices as:

$$Q_{2i}^{k+1} = P_i^k + 8\omega_i^k(P_{2i}^{k+1} - P_{2i-1}^{k+1}) = P_i^k + 2\omega_i^k(P_{i+1}^k - P_{i-1}^k)$$

$$Q_{2i+1}^{k+1} = P_{i+1}^k - 8\omega_{i+1}^k(P_{2i+2}^{k+1} - P_{2i+1}^{k+1}) = P_{i+1}^k - 2\omega_{i+1}^k(P_i^k - P_{i+2}^k)$$

where $\omega_i^k, \omega_{i+1}^k$ are the local tension parameters attached to vertices P_i^k, P_{i+1}^k

3.2 Doo-Sabin Subdivision Scheme

First we define the Doo-sabin scheme and its algorithm and then we use it to obtain the new interpolatory surface subdivision scheme. Given a initial control mesh defined by organized faces, edges, and vertices. Each vertex is a space point and is called a **control vertex**. Each **edge** is a line segment bounded by two vertices. Each **face** is formed by a loop of edges. Each vertex is shared by a finite number of neighboring faces and each edge is shared by exactly two faces. Refined meshes are generated through repeated subdivisions. Each refinement iteration includes the following main steps:

1. For each face with m vertices $V_1, \cdots, V_m$, new vertices $V_1', \cdots, V_m'$ corresponding to the old face vertices are defined by:

$$\boxed{V_i' = \sum_{j=1}^{m} \alpha_{ij} V_{ij},}$$

where,

$$\alpha_{ij} = \begin{cases} \frac{m+5}{4m}, i = j \\ \frac{3+2cos[2(i-j)\pi/m]}{4m}, i \neq j \end{cases}$$

2. For each face, a new face created by connecting $V_1', \cdots, V_m'$ to replace the old one. Faces constructed in this way are named as **type-F** faces.

3. For each edge, a new four-sided face is formed for every edge of the old control mesh by connecting the images of the edge endpoints on each of the faces sharing the edge. These are called **Type-E** faces.

4. For each vertex, a new face is formed for every vertex of the old control mesh by connecting the images of the vertex on each of the faces surrounding the vertex. These are called **Type-V** faces. (All faces shown in Figure 1)

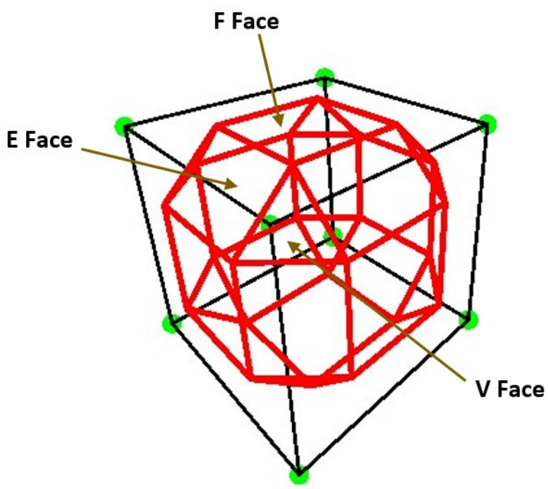


Figure 1: Faces obtained from Doo-Sabin Scheme

3.3 An interpolatory surface subdivision scheme from Doo-Sabin subdivision

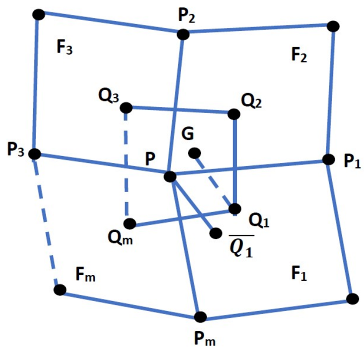


Figure 2: Moved vertices of Doo-Sabin Subdivision

Just like we have seen in the curve case, we define moved vertices for the obtained vertices from the Doo-Sabin scheme. For any old vertex Suppose that faces sharing P are $F_1, \ldots, F_m$ and the new vertices adjacent to P are $Q_1, \ldots, Q_m$, where m is the valence of an old vertex P. The moved vertices corresponding to Q_i(i=1,..,m) are defined as:

$$\boxed{\bar{Q_i} = P_i + \omega_p(Q_i - G)},$$

$$G = \frac{Q_1 + ... + Q_m}{m},$$

where $0 < \omega_p \leq 1$ is the tension parameter attached to local vertex. Like in the curve case, if all the moved vertices are grouped with a topological rule of the Doo-Sabin scheme, and then all of the moved vertices of eacch face, we produce the required interpolatory subdivision scheme.

The core algorithm of the new interpolation subdivision scheme is as follows:

1. For every face F, a new face vertex F_{k+1} is obtained as an average of all the moved vertices of the new face vertices.

2. For every edge E, a new edge vertex E_{k+1} is obtained as an average of four moved vertices which are adjacent to the two end vertices of E and topologically lie in the two faces which share E.

3. Create new edges by connecting each new face vertex to new edge vertices of the corresponding face, and connecting each old vertex to new edge vertices of corresponding edges incident to the old vertex.

4. Create new faces formed by individual loops of new Edges.

By following the steps given in the above steps, the output obtained is the new proposed interpolatory scheme.

3.4 Introducing variable tension parameter for surface shape control

As seen in the curve case, for the displacement safe scheme in the surface subdivision scheme a variable tension parameter is introduced as follows:
An interpolatory surface subdivision scheme is displacement-safe, if there exist 0 <C<1, such that for every edge and face, we have

$$d_E^k = ||E^{k+1} - \frac{P_1 + P_2}{2}|| <= C||P_1 - P_2||,$$

where P_1,P_2 are vertices of edge

$$d_F^k = ||F^{k+1} - \frac{P_1 + P_2 + P_3 + P_4}{4}|| <= min\{||P_i - P_{(i+1)\%4}||\},$$

where P_1,...,P_4 are vertices of face and To make the interpolatory scheme displacement scheme like curve case, we use ω_p as,

$$\omega_p = min\{1, \frac{\lambda}{max(m,4)} min\{\frac{min\{||P_i - P||, ||P_{(i+1)\%m} - P||\}}{||Q_i - G||}\}\}$$

where $0 < \lambda \leq 1$ is also a tension parameter for defining ω_p and for adjusting the shape of the limit surface.

4 Implementation, Results and Discussions

In this sections, the results of the implementation of the above mentioned algorithm in C++ are shown. These results are also compared with the examples given in the original paper. We also show results for various tension parameters and the change in limit surface according to the change in tension parameters. The implementation of the above mentioned algorithm is given below in a step by step manner:

- **Step 0**: We take a well defined polygon mesh(Figure 3):

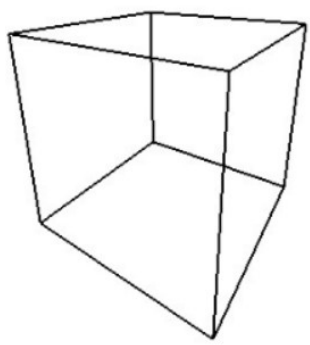


Figure 3: Initial polygon mesh (Step 0)

- **Step 1**: We determine new points obtained from Doo-Sabin subdivision scheme (Section 3.2) (Figure 4).

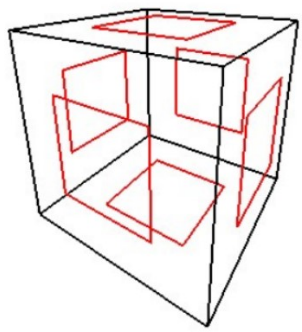


Figure 4: Mesh with Doo Sabin subdivision scheme (Step 1)

- **Step 2**: Now we find the moved vertices from each new vertex obtained from Doo-Sabin refinement (Figure 5).

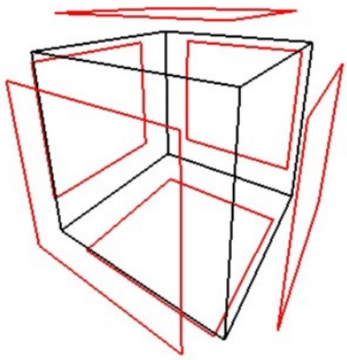


Figure 5: Faces From moved vertices (Step 2)

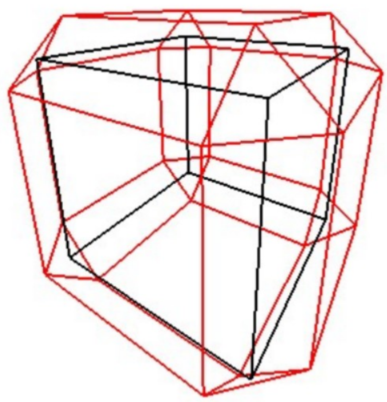


Figure 6: Determining new edge faces (Step 3)

- **Step 3**: As mentioned in section 3.2 for each face we create a new face by connecting the moved vertices. In this step we find the E(Edge) faces (Figure 6).
- **Step 4** We find the new face vertex F_{k+1} as an average of all the moved vertices of the moved face vertices. The new face vertices are shown as blue dots in Figure 7.
- **Step 5**: Similarly, We find the new edge vertex E_{k+1} as the average of the four vertices of the new face formed from each edge E. The formed new edge vertices are shown as green points in Figure 8.

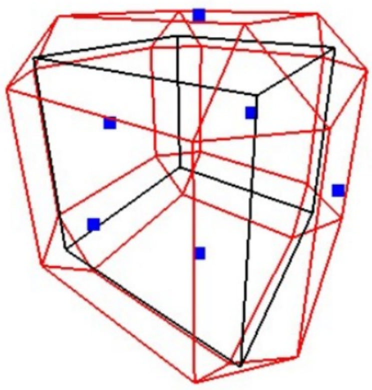


Figure 7: New Face vertices(Step 4)

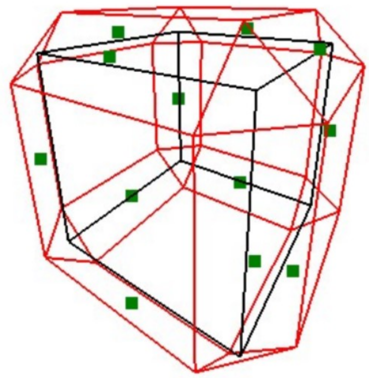


Figure 8: New Edge vertices(Step 5)

- **Step 6**: The new edge vertices E_{k+1} are shown by green points and the new face vertices F_{k+1} are shown as blue points in Figure 9. Connect each new face vertex to new edge vertices of the corresponding face connect each old vertex to new edge vertices of corresponding edges connecting the two old vertices to obtain the required interpolatory subdivision scheme. The result is shown in Figure 10.

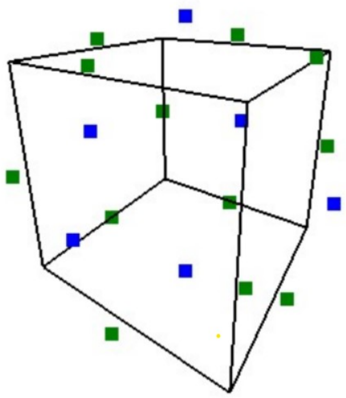


Figure 9: New edge (green) and face (blue) vertices obtained

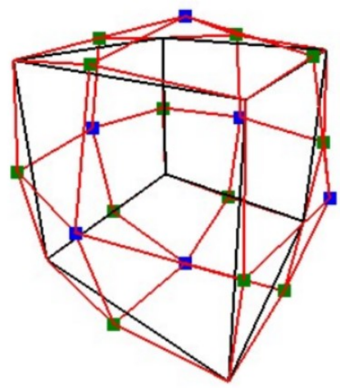
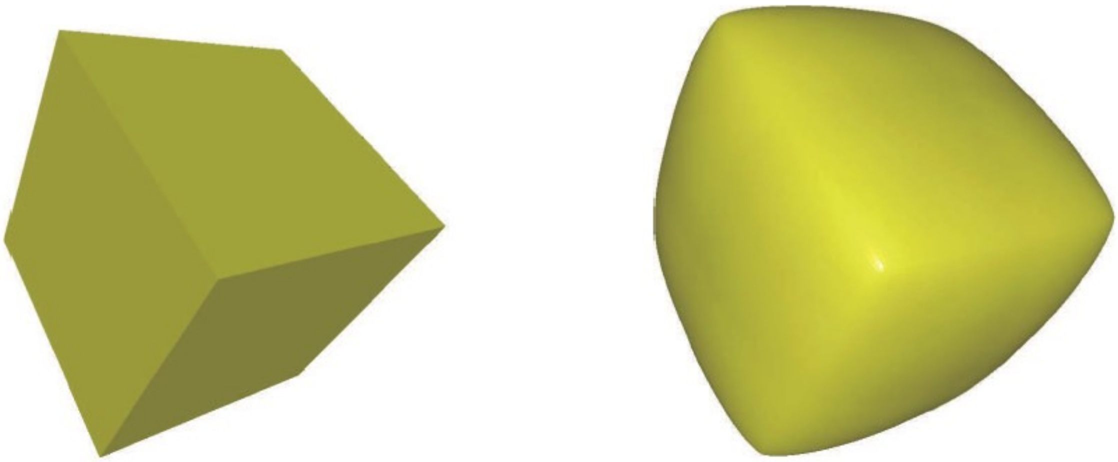


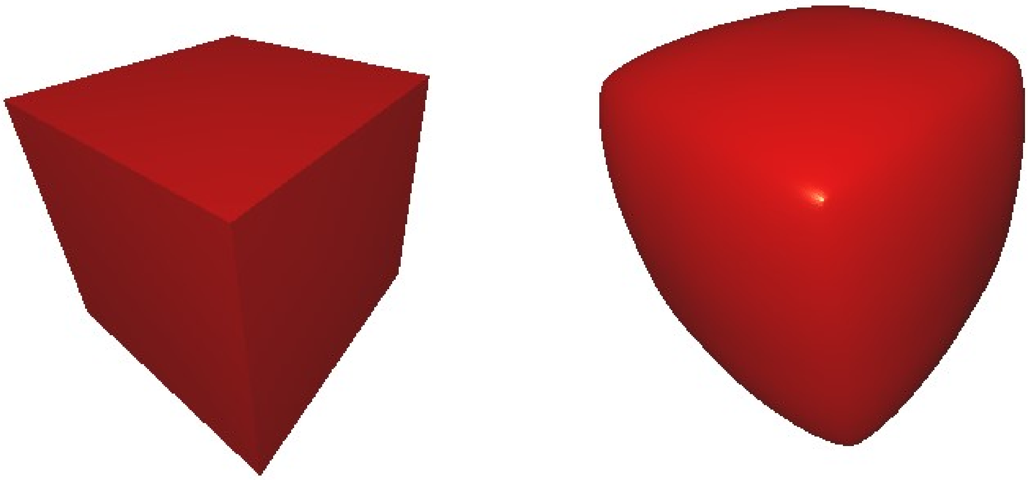
Figure 10: New interpolatory subdivision

Here, we try to implement the new interpolatory subdivision scheme for the models given in the original paper to validate the obtained results. In example 1 we implement the new interpolatory subdivision scheme on a simple cube with tension parameter, $\omega_p = 1$. This is compared to the model given in the paper (Figure 11). We can see that the model obtained exactly replicates the model obtained from the paper(Figure 12).



(a) Original control mesh shown (b) limit surface produced with $\omega_p = 1$

Figure 11: Interpolation of a Cube shown in paper

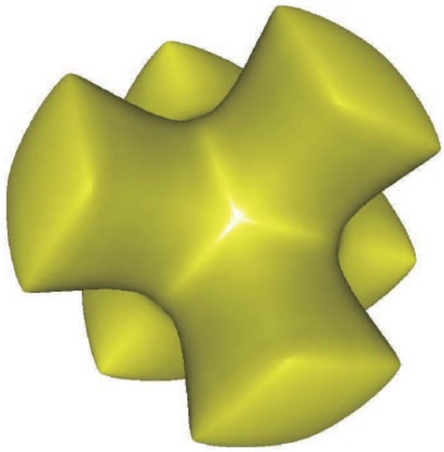


(a) original control mesh (b) Limit surface produced with $\omega_p = 1$

Figure 12: Interpolation of a Cube implemented

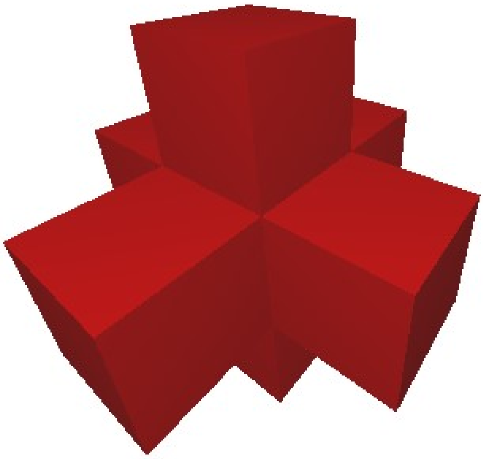
We also compare the results for a 3D cross mesh shown in the original paper for tension parameter, $\omega_p = 1$ (Figure 13 and Figure 14).

(a) Original control mesh

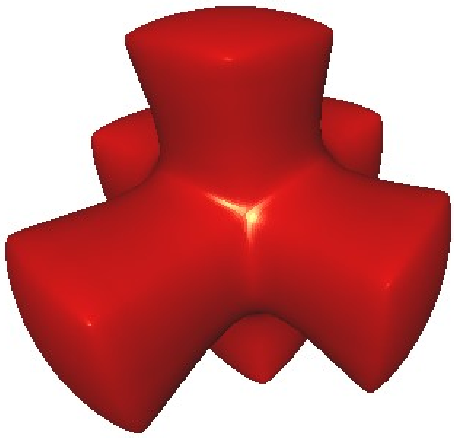


(b) Limit surface produced with $\omega_p = 1$

Figure 13: Interpolation of a 3D Cross Mesh shown in paper

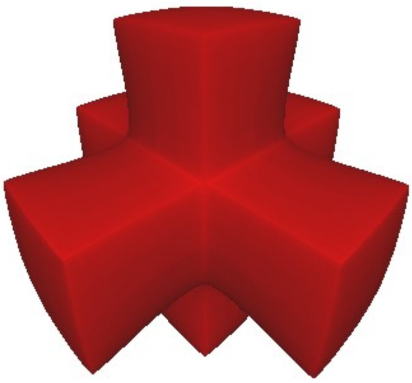


(a) Original control mesh

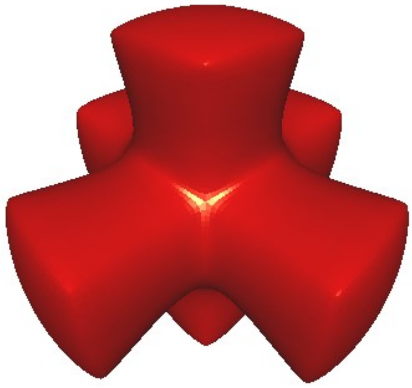


(b) Limit surface produced with $\omega_p = 1$

Figure 14: Interpolation of 3d control mesh implemented



(a) ω_p=0.5



(b) ω_p=1

Figure 15: Comparison of obtained limit surface of a 3D cross mesh with different tension parameters

In the **Figure 11** we compare the new interpolation scheme on a 3D cross mesh with ω_p = 0.5 and 1 (Figure 15). The total implementation of the obtained **GUI** is shown below (Figure 16):

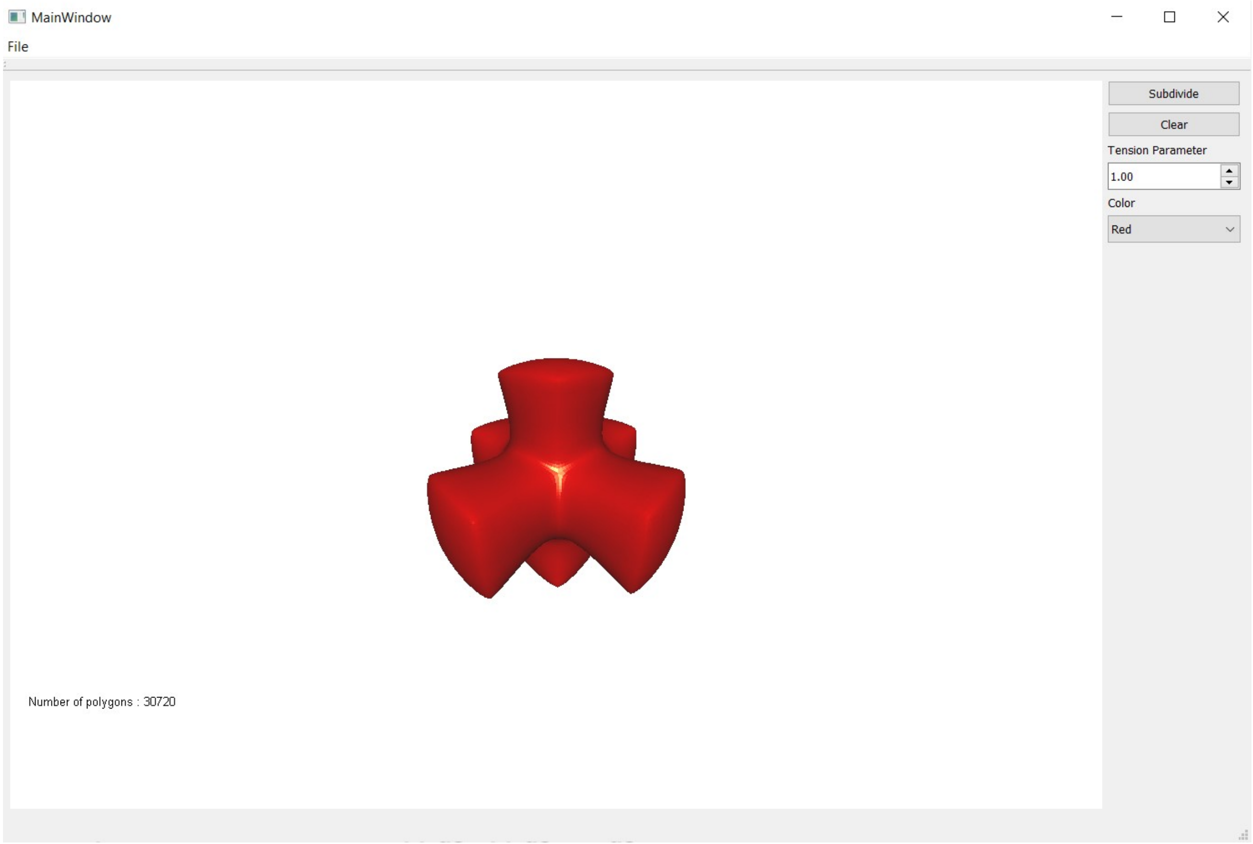


Figure 16: Application Window

5 Conclusions

In this project we reproduce and implement the new interpolation subdivision scheme derived from Doo-Sabin subdivision[3]. From the result and observation, it is fair to say that this new interpolatory scheme produces high quality smooth limit surfaces without any artifacts. The paper also discusses about a variable tension parameter for the subdivision surface scheme to give a displacement safe curve, but in the implementation we used a global tension parameter throughout the surface for obtaining the limit surface. The new interpolatory surface subdivision scheme obtained from Doo-sabin scheme is fast, effective and also artifact free.

References

[1] Edwin Catmull and James Clark. "Recursively generated B-spline surfaces on arbitrary topological meshes." In: *Computer-aided design* 10.6 (1978), pp. 350–355.

[2] George Merrill Chaikin. "An algorithm for high-speed curve generation." In: *Computer graphics and image processing* 3.4 (1974), pp. 346–349.

[3] Chongyang Deng and Weiyin Ma. "Constructing an Interpolatory Subdivision Scheme from Doo-Sabin Subdivision." In: Sept. 2011, pp. 215–222. DOI: 10.1109/CAD/Graphics.2011.80.

[4] Daniel Doo and Malcolm Sabin. "Behaviour of recursive division surfaces near extraordinary points." In: *Computer-Aided Design* 10.6 (1978), pp. 356–360.

[5] Nira Dyn, David Levin, and John A Gregory. "A 4-point interpolatory subdivision scheme for curve design." In: *Computer aided geometric design* 4.4 (1987), pp. 257–268.

[6] Martin Marinov, Nira Dyn, and David Levin. "Geometrically controlled 4-point interpolatory schemes." In: *Advances in multiresolution for geometric modelling*. Springer, 2005, pp. 301–315.

[7] Richard F Riesenfeld. "On Chaikin's algorithm." In: *Computer Graphics and Image Processing* 4.3 (1975), pp. 304–310.

[8] Malcolm Sabin and Loıc Barthe. *Artifacts in recursive subdivision surfaces*. 2003.