A Subdivision Scheme to Model Surfaces with Spherelike Features

Koen Beets
Limburgs Universitair Centrum
Expertise Centre for Digital Media
Universitaire Campus
B-3590, Diepenbeek, Belgium
koen.beets@luc.ac.be

Johan Claes
Limburgs Universitair Centrum
Expertise Centre for Digital Media
Universitaire Campus
B-3590, Diepenbeek, Belgium
johan.claes@luc.ac.be

Frank Van Reeth
Limburgs Universitair Centrum
Expertise Centre for Digital Media
Universitaire Campus
B-3590, Diepenbeek, Belgium
frank.vanreeth@luc.ac.be

ABSTRACT

In this paper, we introduce a novel subdivision method able to generate smooth surfaces which locally tend to minimize variations in curvature. The method is based on a tensor product of a subdivision scheme for circle splines, which is then generalized to arbitrary quadrilateral meshes.

Although they involve a geometric construction, our rules are applied in a uniform way, without the need for applying different rules for different vertices or for different stages in the subdivision process. This results in a more general and natural way to obtain circular curvatures, unlike other approaches involving subdivision curves able to generate circles. Surfaces of revolution are just a basic example, as circular features can be distributed freely over the surfaces generated via our methods.

Keywords

Curve and surface modeling, interpolatory subdivision, curvature minimization, circle splines

1. INTRODUCTION

Subdivision surfaces are widely used in the graphics community. A major advantage is their ability to generate surfaces with arbitrary topology in a uniform representation based on a freely editable coarse polygonal mesh. Due to their close relationship with multiresolution and wavelet analysis, their practical applications further benefit from a vast amount of theoretical knowledge. We refer the interested reader to the Siggraph 2000 course by Zorin et al. [Zor00] and to the book by Warren and Weimar [War02] for excellent introductions to subdivision techniques, with many pointers to the continuously developing literature.

In this paper, our attention goes to surfaces which locally resemble sphere regions. Conventional subdivision methods seem to be inadequate as although the control points in a certain region are all located on the same sphere, the resulting surface usually exhibits a highly varying curvature. This variation results especially problematic for the schemes which directly interpolate the control points provided by the user instead of only approximating them.

Our study of the related literature started with curve representations based on circle blending. Compared to global optimization techniques, such as the Minimized Variation Curve [Seq92], local blending is much cheaper to compute. Furthermore, such global techniques have the additional disadvantage that a local change in the input may have a global impact on the resulting curve, something highly undesirable during interactive modeling.

Circular spline schemes usually employ 4 consecutive vertices P_0, P_1, P_2, P_3 to generate a curve segment with minimal curvature variation between P_0 and P_1. An interpolation scheme combines the segments resulting from the circle C_1 through P_0, P_1, P_2 and the circle C_2 through P_1, P_2, P_3 (see Figure 1). Various interpolation schemes have been proposed. Wenz blends the two circles using simple linear interpolation of 2 point positions on the base arcs [Wen96]. To improve the tangent continuity at the joints of segments, Szilvasi-Nagi and Vendel propose trigonomically weighted interpolation, which guarantees G^2 continuity [Szi00]. To further

WSCG 2005 conference proceedings, ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

improve the curve quality, Séquin et al. constructed a C^2 circular curve blending scheme [Seq05]. Instead of interpolating point positions, they suggest a trigonometrical interpolation between tangent angles. As Séquin et al. argue, this approach remedies the cusps and sharp loops which can appear in the previous schemes, in particular when the control polygon features sharp corners.

In a theoretical work, Chris Doran shows how Clifford algebras help to define a circle blending with an arbitrary level of G continuity [Dor03]. His employment of conformal transformations leads to curves which are – in the G^2 case – identical to the curves presented by Séquin et al. [Seq05]. For us this forms an extra argument to also adopt angle based interpolation. Doran furthermore extended his approach to sphere blending, but unfortunately this did not yet lead to practical methods for surface construction.

Circle blending techniques are interesting to generate curves, but for our subdivision surfaces, we also need a subdivision approach for the underlying curves. In 1987, Dyn et al. introduced the first interpolating subdivision scheme for curves, known in the literature as the four-point scheme or also as the DLG scheme [Dyn87]. Starting from a coarse control polygon, new points are recursively introduced between each pair of old points. The new point positions are defined as the central point of a spline which interpolates the four immediate neighbors.

As the four-point scheme generates curves with highly varying curvature, Séquin and Yen constructed a circular subdivision scheme. Their new point positions are now calculated to lie at the centre of an angle-based interpolated arc [Seq01].

Sabin and Dodgson provided yet another solution to create subdivision curves with more continuous curvature [Sab04]. With a particular definition of curvature as a kind of normalized cross product, they ensure that the new point's curvature averages the curvature of its immediate neighbors. Additionally, to obtain a more even spacing of vertices after subdivision, they position new vertices closer to the shortest edge adjacent to the current edge. The resulting scheme is C^2 in practical situations, just as the four-point scheme.

In the literature, also some subdivision surface schemes incorporating circular arcs are described. Nasri et al. started from an interpolating subdivision algorithm for piecewise C^1 circular spline curves, based on biarcs [Nas01]. This is used to create a modified version of the Doo-Sabin scheme for surfaces, where the standard rules are combined with the circular rules on user-defined edges. This way surfaces with piecewise circular boundaries can be created as well as seamless connections between different surfaces along a common circular boundary. A disadvantage of their technique is that the curvature changes abrupt at the control vertices. Also, to create circular or spherical regions, extra vertices have to be added to the control polygon, while it is not always clear how to add these.

In a similar approach, Morin et al. describe a non-stationary subdivision scheme for surfaces of revolution [Mor01]. With their technique, the user has to mark the desired sections of the curve as circular. Between circular arcs, standard Catmull-Clark subdivision rules are applied. Both Morin et al.'s and Nasri et al.'s schemes are approximating, while we want to create a scheme that interpolates all of its control vertices. Also, we intend to have a more continuously varying curvature everywhere, more than only at certain indicated circular regions.

Our subdivision surface scheme is inspired by Kobbelt's interpolating scheme for quadrilateral meshes [Kob96]. Kobbelt first created a tensor product of the four-point scheme for curves and then extended this to meshes with arbitrary topology.

The rest of this paper is organized as follows: In section 2 we describe a subdivision scheme for curves, which minimizes local variations in curvature. Afterwards we employ the same idea to construct a subdivision scheme for surfaces, and explain how it is constructed. Next, we propose some applications for which the spherelike scheme is very well suited. Finally we present some results which we compare to results of well-known subdivision schemes, and we formulate our conclusion.

2. THE CURVE SCHEME

The subdivision scheme for curves works as follows: For every couple of adjacent vertices P_1 and P_2, we consider the 4 consecutive vertices (P_0, P_1, P_2, P_3). Since every circle can be determined by three vertices, one can fit exactly one circle C_1 going through (P_0, P_1, P_2), and also one circle C_2 through (P_1, P_2, P_3). This situation is illustrated in Figure 1. Between P_1 and P_2, both circles have an arc *arc1* and *arc2*, which is parameterized to have parameter $u=0$ at P_1, and $u=1$ at P_2.

The scheme blends both arcs between P_1 and P_2, creating a curve segment minimizing curvature changes. First we calculate the tangent vectors t_1 and t_2 in P_1, and their average t. The arc arc_{avg} which has a tangent vector equal to t in P_1 is created. On this arc we take the central vertex to be S, and insert it into the new curve.

After several iterations of the recursive subdivision scheme, we obtain a smooth segment, which is shown as a fat dashed line between P_1 and P_2 in Figure 1.

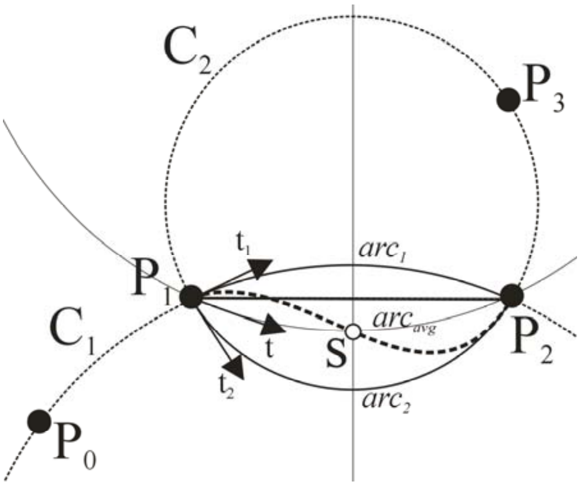


Figure 1: Blending two circle segments between P_1 and P_2. Every iteration, the tangents t_1 and t_2 to the circles C_1 and C_2 are calculated in P_1. Then the arc passing through P_1 and P_2, and having the average tangent t is calculated. The vertex S, lying in the middle of this arc_{avg} is added to the curve.

The algorithm we use is based on Séquin's circular subdivision scheme for curves [Seq01]. Séquin rotates a vertex P with distance $f(u) = b * sin(u*t(u)) / sin(t(u))$ from P_1 lying on P_1P_2 an angle $\varphi(u) = (1-u)t(u)$ around the axis $rot\_axis = P_0P_1 X P_1P_2$, where $b = |P_1P_2|$. The resulting vertex S lies on the average arc arc_{avg}. For u=0.5, this is shown in Figure 2.

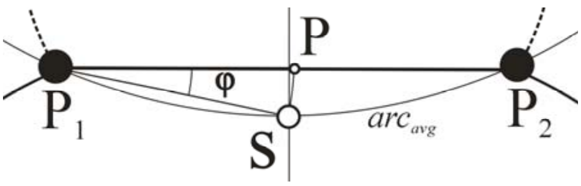


Figure 2: Obtaining S using matrix rotation

We instead suggest not rotating this point P using matrix rotation, but instead we propose to use a geometric construction to obtain S, using vector calculation. Let $P_{12} = (P_2 - P_1) / |P_2 - P_1|$.We calculate the cross product $N = rot\_axis X P_{12}$, which is a unity vector, because both rot_axis and P_{12} are unity vectors and are perpendicular to each other.

Then we obtain:

$$S = P_1 + cos(\varphi) * f * P_{12} + sin(\varphi) * f * N$$

Advantages for using this method instead of matrix rotations are faster calculations of the new points, and higher accuracy.

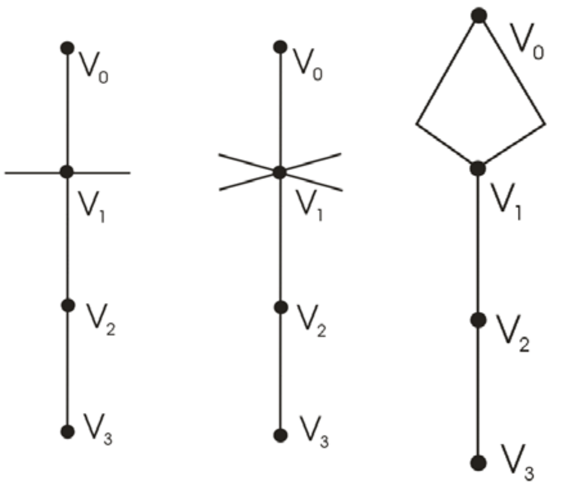
3. THE SURFACE SCHEME

In this paragraph we describe how we extend the techniques presented in the previous section to surfaces, generating smooth and interpolating surfaces of arbitrary topology. Starting from a coarse control mesh, the algorithm recursively refines the mesh. At each iteration, the number of faces is multiplied by four. The algorithm works as follows: First, all edges are split into two, while all original vertices are retained. Then, new face vertices are placed inside every face. Finally, the new mesh is reconnected, replacing every old n-sided face with n new quadrangles. In the next paragraphs we describe the algorithm in more detail.

First, all edges are split into two, using the rule for curves. For every edge which is not part of a boundary, we take the two end vertices V_1 and V_2, and locate V_0 and V_3 (see Figure 3).

Figure 3a illustrates the situation in the regular case. We apply the curve algorithm to these four vertices, and split the original edge in two by inserting a vertex E. If, however, the valence of a vertex belonging to the curve is different from 4, we use other rules.

Suppose we have a vertex v with valence different from four. There are two different cases: If v has an even valence, we take the most central edge to obtain the other vertices used for calculating the edge split . This is illustrated in Figure 3b. If v has an odd valence, we choose V_0 or V_3 as the vertex which is the furthest away from V_1, and belonging to the central face. This is illustrated in Figure 3c.



(a)	(b)	(c)
Regular case	**Even valence, different from 4**	**Odd valence**

Figure 3: Different situations around an edge with vertices V_1 and V_2. In subfigure a, the situation with a regular vertex V_1 is shown. Subfigure b

shows a vertex V_1 with an even valence different from 4, while subfigure c displays a vertex V_1 with an odd valence. In each situation, the vertices V_0 and V_3 are located.

When all vertices V_1, V_2, V_3 and V_4 are found, a new vertex E is inserted between V_1 and V_2. This is illustrated in Figure 4.

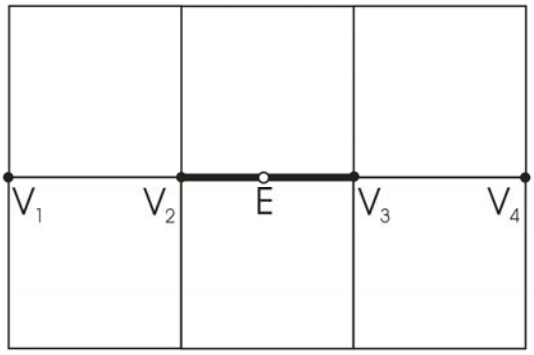


Figure 4: The mask for splitting edges using spherelike interpolating subdivision for surfaces. Vertex *E* is inserted in every edge, using the algorithm for curves. This algorithm is applied to the vertices *(V_1, V_2, V_3, V_4)*.

Secondly, a face vertex F is created inside every face. The creation of a face point in the regular case is illustrated in Figure 5. We apply the scheme for curves to the vertices (V_1, V_2, V_3, V_4) and to the vertices (V_5, V_6, V_7, V_8). Note that both will not give the same result. Thus we add a new face vertex with the average coordinates. Since the scheme is interpolating, existing vertices are left unchanged. Finally, the old faces are discarded, and new faces are created by connecting every old vertex with its two adjacent edge splits, and with a face vertex of an adjacent old face.

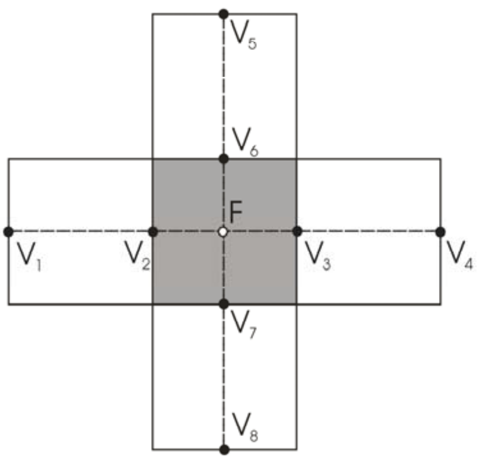


Figure 5: Creation of a new face vertex *F* inside the gray quadrangle, in the regular case. The points V_i are used for calculating *F*. *F* becomes the average of the curve subdivision applied to *(V_1, V_2, V_3, V_4)* and *(V_5, V_6, V_7, V_8)* respectively.

In the extraordinary case, we employ a different rule for generating new face vertices. In this case, we can not simply select 2 paths of edges passing through the centre of the face, since the number of edge splits in the face will be different from 4. So there is a problem picking the second edge vertex. We discern two different cases here: If the number of vertices n is even, we calculate the $n/2$ curve subdivisions using the new edge vertices. Then we take the average of these results. This is shown in Figure 6 (left). If the face has an odd number of vertices, say n, we take the new face vertex to be the average of n calculations of the curve scheme, using the vertices of the face. This is illustrated in Figure 6 (right).

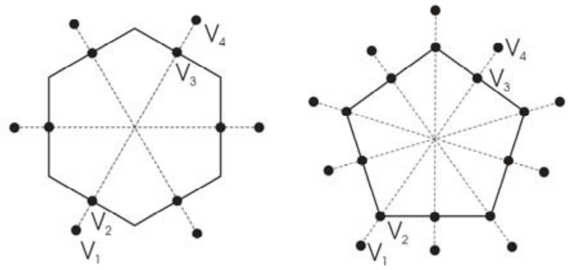


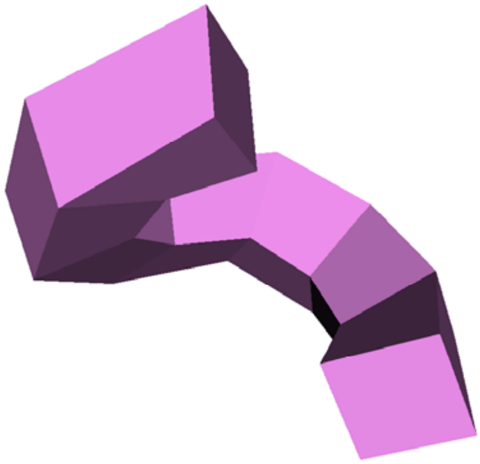
Figure 6: Calculating face points in the extraordinary case: a face with an even number of vertices (left), and a face with an odd number of vertices (right).

4. MODELING APPLICATIONS

There are many applications which may benefit from using the interpolating spherelike subdivision scheme. We enumerate some examples.

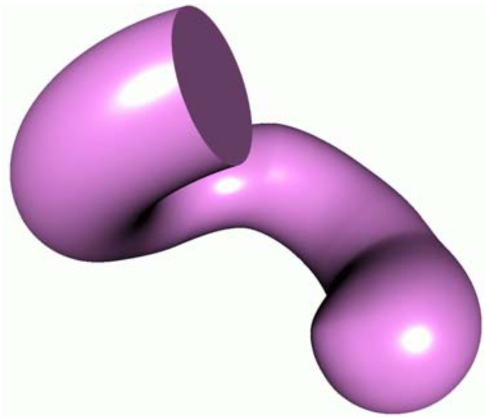
A first application would be the smoothing of polygonal objects, since the scheme does not shrink the object. Selective smoothing of edges is also possible.

Another application is the efficient generation of cylinder-like objects and tubes. Starting from a random curve in 3D, one can sweep a circle over this curve, with the curve going through the centre of the circle. Throughout this path, the diameter of the circle may change, or the circle may change shape. At the end points, one may use a sphere to produce a round end point, or a flat plane. An example object generated using this technique is shown in Figure 7.



(a)

Control mesh



(b)

Subdivided surface

Figure 7: An example tubular object generated using the interpolating spherelike scheme.

Also, objects like surfaces of revolution can be represented easily using the spherelike scheme. An example chess pawn is shown in Figure 8b, along with the control mesh in Figure 8a.

Finally the scheme can be used to support boolean operations. An example would be to create a smooth spherelike blending between two cylinders, where one cylinder cuts another cylinder under an angle.

5. RESULTS

Figure 8 shows a visual comparison of a chess pawn, subdivided using different subdivision schemes. Our spherelike scheme combines the advantages of both interpolating and approximating subdivision: it generates the smooth surface of approximating schemes like Catmull-Clark, while still interpolating the control points. This interpolation is an important feature for application of subdivision surfaces in engineering applications. The interpolating spherelike scheme generates a round pawn while preserving the features well. The surface generated by Kobbelt's scheme is not round enough, while the Catmull-Clark surface lacks the necessary features. Clearly these schemes need a different – and more complex – control mesh to generate a realistic pawn.

6. CONCLUSION

We presented a new subdivision scheme for surfaces, which is interpolating, and which minimizes local variations in curvature. It is well suited to efficiently produce surfaces with spherelike regions.

For general use, subdivision surfaces are not suitable yet. Gonsor and Neamtu present a list of problems with subdivision in engineering applications [Gon01]. Several problematic properties of subdivision surfaces are mentioned which may be alleviated by our scheme: The scheme is interpolating, while still generating good quality surfaces, in contrast to existing interpolating schemes. Secondly, our scheme creates surfaces with good curvature. Finally, our scheme is locally refinable, while still maintaining exactly the same shape, unlike most other schemes.

Future work includes a thorough analysis of the behavior of the scheme, and adding support for adaptive subdivision.

7. ACKNOWLEDGMENTS

The authors are pleased to acknowledge that this work has been partially funded by the European Fund for Regional Development and the Flemish Government.

8. REFERENCES

[Dor04] Doran, C. Circle and Sphere Blending with Conformal Geometric Algebra. Submitted to Computer Aided Geometric Design, 2004

[Dyn87] Dyn, N., Gegory, J., and Levin, D. A 4-point Interpolatory Subdivision Scheme for Curve Design. Computer Aided Design, Volume 4, pp. 257-268, 1987

[Gon01] Gonsor, D., and Neamtu, M. Can subdivision be useful for geometric modeling applications? Boeing Technical Report #01-011, http://www.math.vanderbilt.edu/~neamtu/papers/report.pdf.gz, 40pp, 2001

[Kob96] Kobbelt, L. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. Computer Graphics Forum, Volume 15, Issue 3, pp. 409-420, 1996

[Mor01] Morin, G., Warren, J., and Weimer, H. A. Subdivision Scheme for Surfaces of Revolution. Computer Aided Geometric Design, Volume 18, Issue 5, pp. 483-502, 2001

[Nas01] Nasri, A., van Overveld, C., and Wyvill, B. A Recursive Subdivision Algorithm for Piecewise

Circular Spline. Computer Graphics Forum, Volume 20, Number 1, pp. 33-45, 2001

[Sab04] Sabin, M., and Dodgson, N. A Circle-Preserving Variant of the Four-Point Subdivision Scheme, Proceedings of the 6th International Conference on Mathematical Methods for Curves and Surfaces, Accepted, 2004

[Seq92] Séquin, C., and Moreton, H. Functional Optimization for Fair Surface Design. SIGGRAPH 1992 Proceedings, pp. 167-176, 1992

[Seq01] Séquin, C., and Yen, J. Fair and Robust Curve Interpolations on the Sphere. SIGGRAPH 2001 Sketch, p. 182, 2001

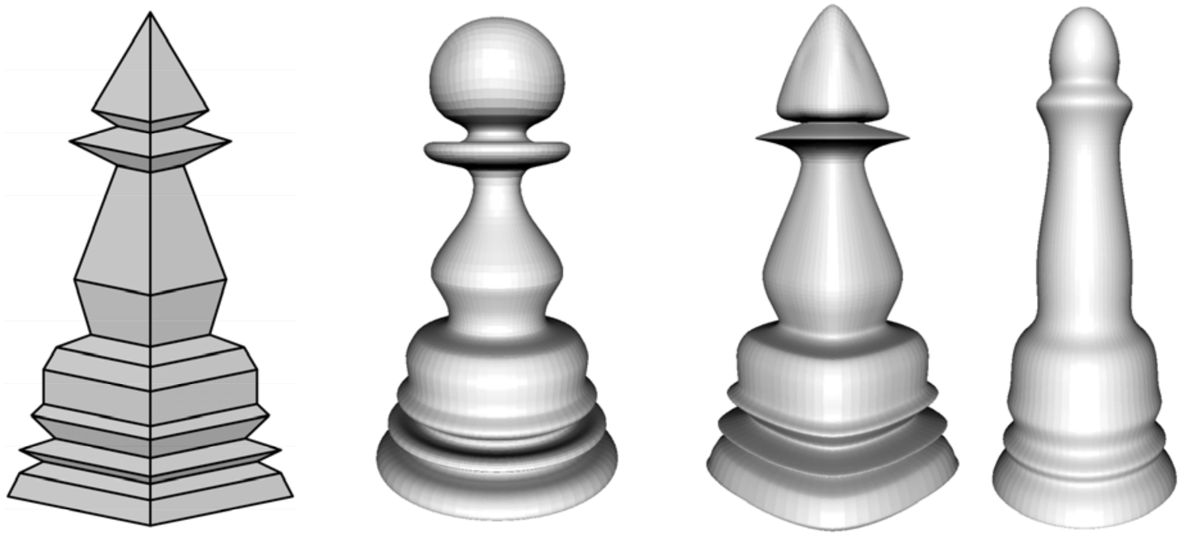
[Seq05] Séquin, C., Lee, K., and Yen, J. Fair, G^2- and C^2-Continuous Circle Splines for the Interpolation of Sparse Data Points. Journal of Computer-Aided Design, Volume 37, Issue 2, pp 201-211, 2005

[Szi00] Szilvási-Nagy, M., and Vendel, T.P., Generating Curves and Swept Surfaces by Blended Circles, Journal of Computer Aided Geometric Design, Volume 17, Issue 2, pp 197-206, 2000

[War02] Warren, J., and Weimer, H. Subdivision Methods For Geometric Design: A Constructive Approach. published by Morgan Kaufmann. ISBN: 1558604464, 2002

[Wen96] Wenz, H.-J. Interpolation of Curve Data by Blended Generalized Circles. Computer Aided Geometric Design, Volume 13, Issue 8, pp. 673-680, 1996

[Zor00] Zorin D., Schröder P. Subdivision for Modeling and animation, SIGGRAPH 2000 course notes, 2000



(a) **Control mesh** **(b)** **The interpolating spherelike scheme** **(c)** **Kobbelt's scheme** **(d)** **Catmull-Clark**

Figure 8: A visual comparison of an object of revolution, subdivided with various subdivision schemes.