Volume 2021 (4002), Number 1 pp. 1–5 COMPUTER SCIENCE *seminar*

Ray Tracing of Parametric Surfaces with Focus on Subdivision Surfaces

Lukas Wolf[1]

[1]OTH Regensburg, Germany

Abstract

Subdivision surfaces are used to algorithmically generate smooth surfaces for 3D models. This work gives an overview about subdivision surfaces and related techniques by exploring the Catmull-Clark scheme. Especially the use of subdivision surfaces for cinematic rendering including ray tracing and displacement mapping is investigated. In the basic displacement mapping approach the quality of the displacement depends on fine tessellation, which correlates to the memory consumption. Novel methods like tessellation-free displacement mapping do not have that dependency. Furthermore, the Ptex format is addressed that has been developed specifically for storing subdivision surface related textures, which allows seamless model rendering.

1. Introduction

In 3D modeling there are various techniques to create detailed surface structures. The most basic variant is the use of triangular meshes. Two more advanced widespread approaches are Non-Uniform Rational B-Splines (NURBS) surfaces [PT96, p. 128–138] and subdivision surfaces. Both variants use coarse control meshes consisting of vertices. In the NURBS approach curves are created by using B-splines to interpolate vertices. B-splines are evaluated either analytically or by approximation also via subdivision. One evaluation method is the de-Boor algorithm [DBDB78]. Subdivision surfaces in comparison create new vertices based on the mesh control vertices to reach a higher level of detail. The algorithm is applied recursively multiple times to create a smoother approximation of a curve. Analytical evaluation offers a continuous representation of curves, whereas subdivision creates discrete geometry. In the limit continuous curves are approximated.

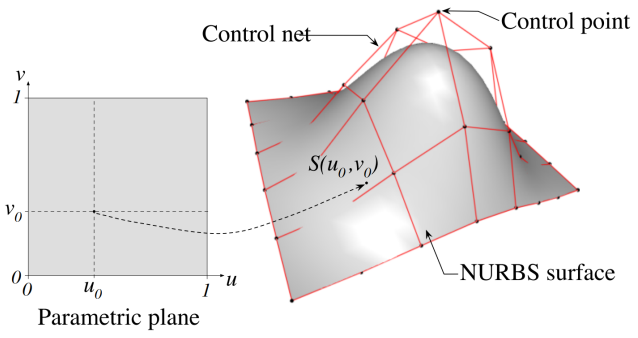


Figure 1: The image [ELG06, Fig. 1] shows a NURBS surface and its corresponding parametric plane. NURBS curves are aligned in two dimensions as tensor product.

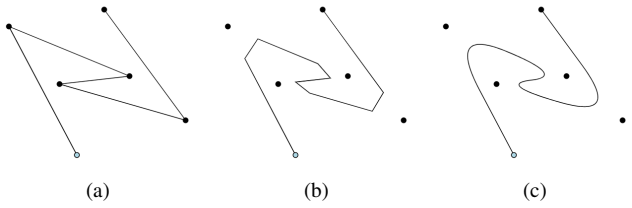


Figure 2: Chaikin's subdivision algorithm is demonstrated for a curve. The images show: (a) zero iterations (the control mesh), (b) one iteration and (c) the limit shape.

To model NURBS surfaces control vertices are aligned in quadrilateral patches as shown in Figure 1. B-splines are applied in two dimensions [LLL17, p. 24–25]. The surface will then be interpolated accordingly. When working with subdivision surfaces considerable amount of geometry is generated. Subdivision surfaces allow precomputation of this geometry, which can speed up the rendering process. This is also possible for NURBS surfaces when the approximated surface points are stored. The possibly high memory consumption needs to be considered, also for moderately sized models. On-the-fly computation either for NURBS or subdivision surfaces on the other hand requires only a fraction of that memory. However, the just-in-time evaluation raises the computation time, which is especially a problem in interactive or real-time applications.

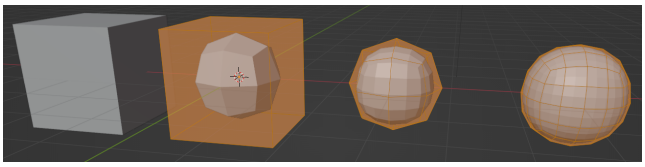


Figure 3: Catmull-Clark subdivision of a cube. The control mesh of the previous iteration is displayed in orange. From left to right iterations of level 0 to 3 are shown. The scene has been modelled with the free graphics software Blender.

2. Subdivision methods

There are different algorithms available for subdivision. Chaikin's algorithm [Cha74] is capable of generating arbitrary curves using subdivision. The newly created vertices represent the control mesh for the next iteration as seen in Figure 2. The limit shape represents a quadratic B-spline curve. Doo and Sabin [DS78] extend this algorithm for surface modeling. This method and Catmull-Clark subdivision [CC78] are the most popular ones used for quadrilateral meshes. Catmull-Clark and Doo-Sabin subdivision can be generalized [LLL17, p. 75–79] to support non-uniform meshes. An example of subdivision surfaces can be seen in Figure 3, which shows three iterations of Catmull-Clark subdivision. The resulting mesh of one iteration represents the control mesh for the next iteration. For example, the second mesh from the left is identical to the control mesh around the third mesh displayed in orange. Algorithms optimized for triangular meshes are Loop subdivision [Loo87], Butterfly subdivision [DLG90] and $\sqrt{3}$-subdivision [Kob00]. In production the Reyes pipeline [CCC87] is one popular approach to render subdivision surfaces. This is also used by animation studios like Pixar, which implemented that approach in its RenderMan engine[†].

The following chapters focus especially on Catmull-Clark subdivision, which is widely used in movie animation [DKT98]. The techniques creases, feature-adaptive subdivision, displacement mapping and the Ptex format are evaluated as well as ray tracing of subdivision surfaces.

2.1. Catmull-Clark subdivision

In character animation [DKT98] quadrilateral meshes are often used. Quadrilaterals are specifically suitable to model tube shapes like arms and legs and to preserve symmetries. As a result, Catmull-Clark subdivision, which is based on quads, became especially popular in movie and character animation.

The subdivision depends on mesh vertices, edge points and face points [CC78] as displayed in Figure 4 (a). Edge points represent the center of an edge between two vertices. Face points are at the centroid of a face. For each vertex and edge point a related point is calculated according to the subdivision rules equation 1 and equation 2 [CC78].

[†] https://renderman.pixar.com

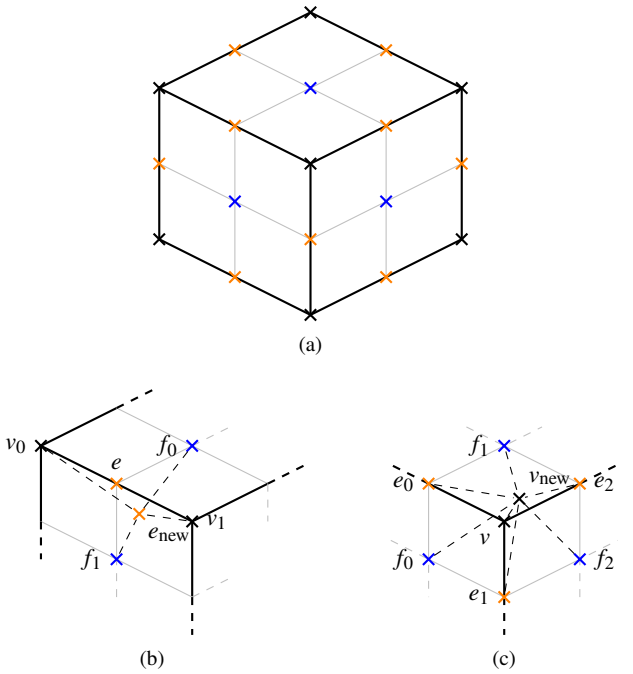


Figure 4: This figure highlights the relevant points in Catmull-Clark subdivision to calclulate the subdivided vertices and edge points. New edge points become vertices in the subdivided mesh. (a) Edge points (orange) and face points (blue) need to be calculated based on the vertices (black). (b) Calculation of the to e corresponding subdivided edge point e_{new}. (c) Calculation of the to v corresponding subdivided vertex v_{new}.

$$e_{\text{new}} = \frac{v_0 + v_1 + f_0 + f_1}{4} \tag{1}$$

$$\begin{aligned} v_{\text{new}} &= \frac{Q}{n} + \frac{2R}{n} + \frac{(n-3)v}{n} \\ Q &= \frac{1}{n}\sum_{j=0}^{n-1} f_j \\ R &= \frac{1}{n}\sum_{j=0}^{n-1} e_j \end{aligned} \tag{2}$$

Equation 1 gives the formula for the calculation of the new edge point. The new edge point depends on the two vertices and face points adjacent to the edge point, but not on the edge point itself. This can be seen in Figure 4 (b). The new point related to the vertex is calculated by equation 2. It depends on the position of the vertex and its adjacent edge and face points, which is displayed in Figure 4 (c). The parameter n, valence of a vertex, represents the number of edges that are connected to vertex v, The subdivision rules apply for arbitrary valence. These new points span a new mesh, which represents the subdivided model and which can be seen as the control mesh for the next iteration.

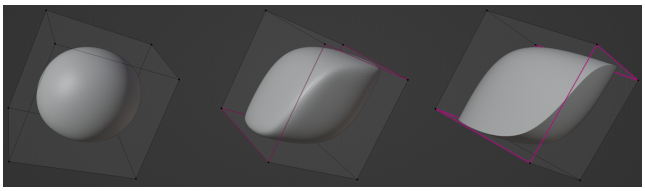
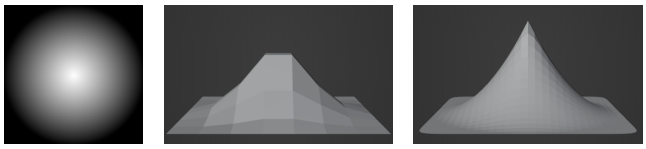


Figure 5: This figure visualizes the difference between no creases (left), semi-sharp creases (middle) and infinitely sharp creases (right). Catmull-Clark subdivision of level 6 has been applied to all models and the base mesh in each case is the unit cube. The edges tagged as crease are visible on the control mesh of the right model. Blender has been used for scene modeling.



(a) (b) (c)

Figure 6: In this figure displacement mapping is demonstrated on a 7×7-plane. Images were rendered with Blender. (a) The used displacement map is shown where dark indicates weak displacement and light high displacement. (b) Displacement mapping has been applied to the plane without further subdivision. (c) By Catmull-Clark subdivision beforehand, here level 2, the displacement improves visibly in quality.

3. Creases

Creases are an extension to subdivision surfaces and are used to model sharp edges. Hoppe et al. [HDD*94] describe a variant of that feature, the infinitely sharp crease. Edges of the control mesh are tagged as sharp. When subdividing, sharp creases are created at these edges whereas the remaining surfaces gradually become smoother. An example of this type of crease is displayed in the right model of Figure 5. Creases are implemented by applying a different set of subdivision rules [HDD*94, DKT98] to tagged edges compared to regular edges.

In later work DeRose et al. [DKT98] propose the concept of semi-sharp creases. A crease of that type is shown in the middle model of Figure 5. This is a generalization to infinitely sharp creases, which allows parameterization of the crease's sharpness s. In an implementation the subdivision is split into two phases. In the first phase tagged edges are subdivided s times with the rules for infinitely sharp creases. For the following subdivision steps the regular rules are applied. Each tagged edge can be assigned with an individual value for s. If s holds a non-integer value, the resulting vertex position can be interpolated between the vertex positions of the upper and lower subdivision level.

4. Feature-adaptive subdivision

Feature-adaptive subdivision [NKF*16] means to only subdivide when necessary. This is the case for rendering so called features, which include the modeling of sharp and semi-sharp creases and extraordinary vertices, which are vertices with a different valence than four. These features need to be handled with subdivision surfaces and are hard to reconstruct with other parametric surfaces. Patches adjacent to such features are classified as irregular. Other patches are called regular and can be rendered as bi-cubic B-spline patches. This is especially desirable because of the potential utilization of dedicated tessellation hardware. Irregular patches are further recursively subdivided. In the process more regular patches are generated and the area represented by irregular faces shrinks.

5. Texturing

Textures on parametric surfaces are projected to the resulting curvature. When subdivision surfaces are used the corresponding parts of the texture map to their subdivided patch (subpatch). This mapping between texel and surface point can implicitly be calculated. In this chapter the texturing technique displacement mapping and the Ptex format are discussed.

5.1. Displacement mapping

Displacement maps were described by Cook [Coo84] and are an extension to bump maps [Bli78]. A bump map stores information regarding the height at a specific point. Bump mapping creates an illusion of depth. The silhouette of an object and therefore also its shadow remains unaffected. The illusion breaks especially when the surface is viewed from a flat angle.

Similarly to bump maps, displacement maps also store the height information as displayed in Figure 6 (a). In contrast to bump mapping the geometry of the object is actually modified, instead of creating an illusion, when applying displacement mapping. Vertices are moved inward or outward according to the displacement map and normals of the moved faces are updated. Textures of the surface are projected accordingly to the modified geometry. To reach a high level of fidelity, fine tessellation of the model is required. This tessellation can be reached manually in the modeling process or automated by subdivision. The difference of a coarse and a fine tessellated mesh is illustrated in Figure 6 (b) and (c).

5.2. Ptex

Prior texturing methods like the usage of atlases [LPRM02] have problems to render the edges of a texture. There are often visible artifacts at the seams between neighboring textures, such as cracks or in animated models aliasing. The focus of the Ptex format [BL08] is to overcome this issue.

In contrast to atlases, in Ptex it is required to map a separate texture to every face of the model. An earlier approach that utilizes per-face texturing is the Displaced Subdivision Surface [LMH00], but these do not have a particular handling for artifacts at seams. Ptex addresses the topic by providing mesh adjacency data that is used that is used for specific texture filtering at patch edges. The per-face approach further significantly reduces the modeling expense since the process of aligning a 2D texture atlas to the 3D

model, called surface unwrapping, is not required. The format is capable of handling features, displacement mapping and varying texture resolutions.

6. Ray tracing of subdivision surfaces

The basic approach to handle subdivision surfaces with ray tracing is to pre-tessellate the scene and apply ray tracing methods like path tracing [Kaj86] afterwards. As noted before, the main memory is prone to exceed with pre-subdivision, especially when fine tessellation is required for displacement mapping. Benthin et al. [BWN*15] propose a caching technique that maintains tessellation data. By that the memory consumption is limited to a fixed size. Patches are subdivided just-in-time only when necessary. In the following, two other approaches are described in more detail.

6.1. Compression schemes

To tackle the memory consumption problem for a priori tessellation, Selgrad et al. [SLM*16] propose a strategy using a two-level Bounding Volume Hierarchy (BVH) scheme. A standard BVH is used for the upper layer and the mesh of the scene is divided accordingly. When a certain flatness criterion (e.g. by reaching a specified subdivision level or by comparing the normals of neighboring patches) is met for the subpatches a compressed BVH is used as the second layer. The bounding box for the root node of the compressed BVH is aligned according to the respective patch. The idea of the compression scheme is to drop the actual geometry at the leaf nodes and only use its bounding box for ray intersection tests. By that the memory consumption can be largely decreased. Even though the original geometry is approximated by bounding boxes the rendered results are still of high accuracy due to the fine grained subdivision.

Furthermore, the compressed BVH is always balanced, which allows an implicit addressing of its nodes. Therefore pointers, (u, v)-coordinates or vertex indices [LMSS18] can be omitted at this level of the BVH. It is guaranteed that no cracks are produced by the approximation of the bounding boxes. This approach has been extended by Lier et al. [LMSS18] by providing a compact leaf representation. By that the BVH depth can also be decreased.

6.2. Tessellation-free displacement mapping

Thonat et al. [TBS*21] propose an approach to ray trace displacement maps that does not rely on fine tessellation. In the evaluation this method could be applied to interactive applications while remaining high quality in the results. Due to the absence of high tessellation, only a fraction of the used memory is required.

The described algorithm also uses two BVH layers where the first is the standard BVH and its leaf nodes respectively link to the root of the second layer, the Displaced Bounding Volume Hierarchy (D-BVH). The second layer BVH is built on mipmaps of the displacement map. Specifically, displacement maps are structured in minmax mipmaps as can be seen in Figure 7. The level 0 mipmap is equal to the displacement map. Higher levels store the minimum and maximum height for specific areas. The 1×1 mipmap is the top level and corresponds to the root of the D-BVH. Each texel of

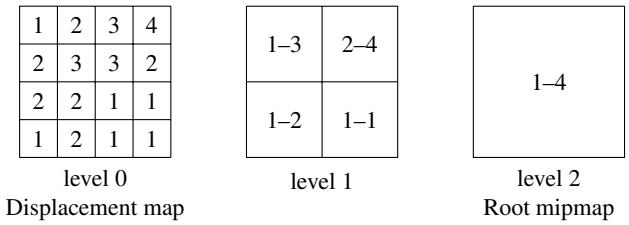


Figure 7: This figure shows a simplified example of three displacement mipmap levels. Level 0 is the original displacement map and its stored height information. The values refer to the offset from the base triangle and can also be negative. Four mipmap texels converge to one texel of the upper level, which provides the minimum and maximum height of that area. The 1×1 mipmap (here level 2) represents the root of the D-BVH.

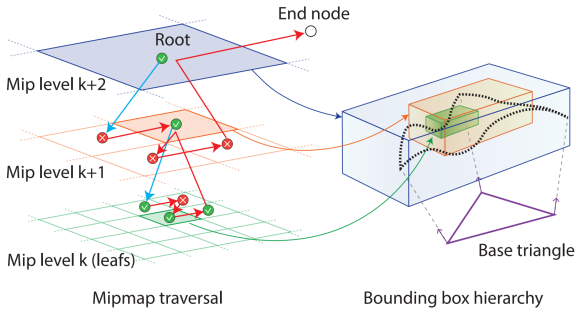


Figure 8: This figure is an illustration from Thonat et al. [TBS*21, Fig. 3] and showcases the ray traversal through the D-BVH. Each texel (left) corresponds to a bounding box (right). The dashed shape on the right represents the displaced projection of the base triangle. Green check-marks and red crossings in the BVH representation indicate a hit or a miss of the ray. If a texel was marked as hit then the corresponding four texels of the next lower mipmap are evaluated recursively. The recursion stops when reaching the leaf nodes or if no intersection was detected at a mipmap level.

these mipmaps correlates to a bounding box. The volume is determined by the partial area of the displacement map and the minimum and maximum height information. The root bounding box encapsulates the entire displaced projection of the base shape, as displayed in Figure 8 on the right side. When reaching a leaf node of the first layer BVH the ray traversal is delegated to the D-BVH. This traversal is illustrated in Figure 8.

This technique can still be used with subdivision surfaces. The advantage is that the level of detail reached by subdivision is now independent of the quality of displacement mapping. Subdivision and displacement mapping can be used orthogonally. This is also interesting for feature-adaptive subdivision where different levels of tessellation are used. By that, it is plausible to use displacement mapping usable independent of the subdivision level or patch size.

More recent works [Oga23] extend the tessellation-free displacement mapping approach to simplify the implementation. This is achieved by partially delegating the ray tracing process to texture space (nonlinear ray tracing) for the evaluation of displacements.

7. Conclusions

Modern movie production and 3D applications require highly detailed renderings and animations. Use cases for realistic lighting and shadows are increasing. Furthermore, the demand for interactive and real-time applications rises as well. A high degree of detail can be attained by subdivision surfaces in combination with features like creases and especially displacement mapping. To achieve realistic lighting, ray tracing is the preferred technique compared to rasterization. Current research addresses these challenges, especially to improve calculation times and reduce memory consumption while remaining high quality results. Caching strategies [BWN*15] and compression schemes [SLM*16, LMSS18] have been developed for ray tracing subdivision surfaces to attain better memory management. Further, tessellation-free displacement mapping approaches [TBS*21,Oga23] allow interactive modeling in a ray traced environment and decouple displacement quality and subdivision level. The utilization of the GPU [NKF*16] regarding subdivision surfaces and rasterization is well studied. Considering this with ray tracing instead of rasterization, this is still an active topic of research.

References

[BL08] BURLEY B., LACEWELL D.: Ptex: Per-face texture mapping for production rendering. *Computer Graphics Forum 27*, 4 (2008), 1155–1164. doi:https://doi.org/10.1111/j.1467-8659.2008.01253.x. 3

[Bli78] BLINN J. F.: Simulation of wrinkled surfaces. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1978), SIGGRAPH '78, Association for Computing Machinery, pp. 286–292. doi:10.1145/800248.507101. 3

[BWN*15] BENTHIN C., WOOP S., NIESSNER M., SELGRAD K., WALD I.: Efficient ray tracing of subdivision surfaces using tessellation caching. In *Proceedings of the 7th Conference on High-Performance Graphics* (New York, NY, USA, 2015), HPG '15, Association for Computing Machinery, pp. 5–12. doi:10.1145/2790060.2790061. 4, 5

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (1978), 350–355. doi:https://doi.org/10.1016/0010-4485(78)90110-0. 2

[CCC87] COOK R. L., CARPENTER L., CATMULL E.: The reyes image rendering architecture. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1987), SIGGRAPH '87, Association for Computing Machinery, pp. 95–102. doi:10.1145/37401.37414. 2

[Cha74] CHAIKIN G. M.: An algorithm for high-speed curve generation. *Computer Graphics and Image Processing 3*, 4 (1974), 346–349. doi:https://doi.org/10.1016/0146-664X(74)90028-8. 2

[Coo84] COOK R. L.: Shade trees. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1984), SIGGRAPH '84, Association for Computing Machinery, pp. 223–231. doi:10.1145/800031.808602. 3

[DBDB78] DE BOOR C., DE BOOR C.: *A practical guide to splines*, vol. 27. springer-verlag New York, 1978. 1

[DKT98] DEROSE T., KASS M., TRUONG T.: Subdivision surfaces in character animation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, Association for Computing Machinery, pp. 85–94. doi:10.1145/280814.280826. 2, 3

[DLG90] DYN N., LEVINE D., GREGORY J. A.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph. 9*, 2 (apr 1990), 160–169. doi:10.1145/78956.78958. 2

[DS78] DOO D., SABIN M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, 6 (1978), 356–360. doi:https://doi.org/10.1016/0010-4485(78)90111-2. 2

[ELG06] ESTRATAT M., LA GRECA R.: An interfacing module using configuration for declarative design of nurbs surfaces. In *Proceedings of the 9th International Conference on Computer Graphics and Artificial Intelligence (3IA'2006)* (2006), pp. 85–96. 1

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, Association for Computing Machinery, pp. 295–302. doi:10.1145/192161.192233. 3

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), SIGGRAPH '86, Association for Computing Machinery, pp. 143–150. doi:10.1145/15922.15902. 4

[Kob00] KOBBELT L.: $\sqrt{3}$-subdivision. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 103–112. doi:10.1145/344779.344835. 2

[LLL17] LIAO W., LIU H., LI T.: *Subdivision Surface Modeling Technology*. Springer, 2017. 1, 2

[LMH00] LEE A., MORETON H., HOPPE H.: Displaced subdivision surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 85–94. doi:10.1145/344779.344829. 3

[LMSS18] LIER A., MARTINEK M., STAMMINGER M., SELGRAD K.: A high-resolution compression scheme for ray tracing subdivision surfaces with displacement. *Proc. ACM Comput. Graph. Interact. Tech. 1*, 2 (aug 2018). doi:10.1145/3233308. 4, 5

[Loo87] LOOP C.: Smooth subdivision surfaces based on triangles [master thesis]. 2

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3 (jul 2002), 362–371. doi:10.1145/566654.566590. 3

[NKF*16] NIESSNER M., KEINERT B., FISHER M., STAMMINGER M., LOOP C., SCHÄFER H.: Real-time rendering techniques with hardware tessellation. *Computer Graphics Forum 35*, 1 (2016), 113–137. doi:https://doi.org/10.1111/cgf.12714. 3, 5

[Oga23] OGAKI S.: Nonlinear ray tracing for displacement and shell mapping. In *SIGGRAPH Asia 2023 Conference Papers* (New York, NY, USA, 2023), SA '23, Association for Computing Machinery. doi:10.1145/3610548.3618199. 4, 5

[PT96] PIEGL L., TILLER W.: *The NURBS book*. Springer Science & Business Media, 1996. 1

[SLM*16] SELGRAD K., LIER A., MARTINEK M., BUCHENAU C., GUTHE M., KRANZ F., SCHÄFER H., STAMMINGER M.: A compressed representation for ray tracing parametric surfaces. *ACM Trans. Graph. 36*, 1 (nov 2016). doi:10.1145/2953877. 4, 5

[TBS*21] THONAT T., BEAUNE F., SUN X., CARR N., BOUBEKEUR T.: Tessellation-free displacement mapping for ray tracing. *ACM Trans. Graph. 40*, 6 (dec 2021). doi:10.1145/3478513.3480535. 4, 5