

A new method for cutting splats of models with sharp features

Rafael Fernandes Ivo
Departamento de computação - UFC
CRAB
Fortaleza - Brazil

Creto Augusto Vidal
Departamento de computação - UFC
CRAB
Fortaleza - Brazil

Joaquim Bento Cavalcante Neto
Departamento de computação - UFC
CRAB
Fortaleza - Brazil

Abstract—Surface splatting is a fast rendering technique for point-based models that usually delivers good-quality renderings. However, the intrinsic characteristic of this technique is the treatment of points as if they were discs. That requires special processing near sharp edges and corners in order to prevent rendering artifacts from appearing. In this work, we present a clever way of clipping splats near sharp edges and corners by a classification of neighboring splats, which belong to distinct intersecting surfaces. Those neighboring splats that take part in the clipping process are called *clip partners*. Their concavity or convexity with respect to one another are determined, as well as their relation with the clipped splat. In our approach, there is no need of distance computations nor of other complex operations during the rendering process, since the classification can be performed offline during a pre-processing phase. The examples presented in this paper demonstrate the importance of correctly clipping splats for high quality rendering.

Keywords—clip partners; edge classification; surface splatting

I. INTRODUCTION

In recent years, many researches have focused on efficient representation, modeling, processing and rendering of geometries sampled by points. There are two main reasons for this interest: the increasing complexity of polygonal graphics models and popularization of 3D scanners. To fill the gaps between samples more efficiently, representations based on points are usually extended to representations based on *splats* [1], where the surface is locally approximated by a small disk or by a small ellipse.

For many applications, the ability to render edges and corners, where the surface's continuity is only C^0 , is essential. These discontinuities often appear in models generated by Boolean operations (CSG) [2], [3] and physical simulations of cracking and breaking of materials [4]. Due to the nature of the splats, an infinite amount of splats would be required to represent edges perfectly. However, if the location of these discontinuities are known, the splats can be adapted either by changing the sampling, by rearranging the samples or by clipping.

The artifacts in sharp features can be minimized by increasing the sampling, but even if surface splats can be processed extremely fast by exploiting the programmable features of current graphics hardware, it is preferable to minimize the number of splats that are needed to represent a given object, since the processing time is nevertheless proportional to the

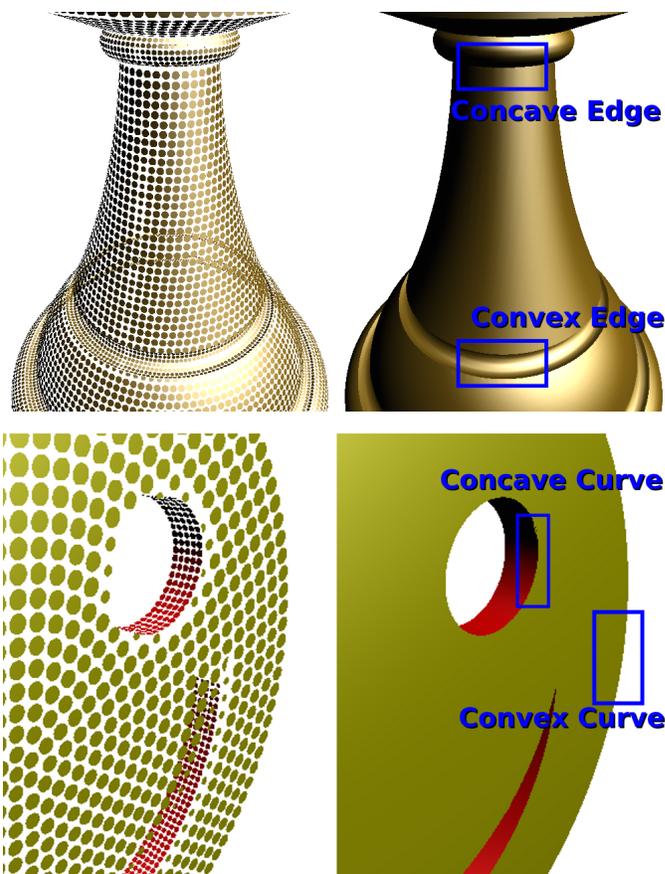


Fig. 1. Models rendered with the adaptive technique of clipping splats. Even on samplings with different densities (above) or not aligned (below), clippings fit the edges properly.

number of geometric primitives. For this reason, this work aims at achieving high quality renderings of models with sharp features, even if the density of samples is low.

In this work, the samples are adapted by clippings to represent correctly the discontinuity curve without any modeling information. This edge can be concave or convex (Figure 1 above) and this curve can be concave or convex too (Figure 1 below), even over different splat densities around it. To adapt the splats on the sharp features, a set of splats is formed for each sample. This set of splats is classified according to the

orientation of concavity and convexity of its elements, as well as according with its interaction with the splat to be clipped. That classification is used to guide the choice of the best option of clipping and to adjust the splat to the edge curve. The contributions of this work can be listed as follows:

- Adaptation splats near curved edges, without adding new samples or rearranging the existing ones. This allows for good renderings of less detailed models;
- Offline classification of splats in order to allow for the removal of fragments at rendering time without loss of performance;
- Independence from modeling information, which increases the applicability of the method.

II. RELATED WORK

Points were first proposed as rendering primitives by Levoy and Whitted [5] in 1985. After that work, several approaches of point-based rendering, using techniques of reconstruction in image space [6], [7] or re-sampling in object space [8], [9], were proposed. In contrast, *surface splatting* [1] associate normal vectors and radius to each point to consider them as disks or ellipses in object space in order to avoid the gaps in rendered images. However, in many surfaces, regions around edges and corners need to be rendered properly. This is relevant since the overlap of splats in those areas usually shows artifacts.

The first work to treat edges was Adams and Dutr’s article [2]. In that work, smooth surfaces represented with a set of splats were subjected to boolean operations against each other. Usually, the boolean operation that computes the intersection of these smooth surfaces forms edges and corners in the resulting surface. Therefore, an approach for correctly rendering the resulting model was needed. The intersection lines were detected by an octree, the leaves of which hold samples of both surfaces involved in the boolean operation. All splats that cross one of those lines are analyzed and, possibly, replaced by other smaller splats.

Notice that using Adams and Dutr’s method [2] requires an infinite amount of splats to represent edges continuously. Pauly *et al.* [3] introduced a special class of splats that represent edges explicitly. Each splat in this class is represented by two discs with the same center. During rasterization, each disc is clipped against the plane defined by the other in order to obtain a linear approximation of the curve of intersection in screen space. Despite the better edge representation, the step of projecting points on the curve of intersection of CSG models is a very time consuming process. In addition, the splats are clipped against a single splat, i.e., the method needs a larger amount of samples to represent a curved edge faithfully.

Zwicker *et al.* [10] used informations from previous discontinuity detection methods or modeling information to adapt the splats near the edges through the insertion of clip lines. These clip lines are calculated at the intersection of the tangent planes of the splats on each side of the edge. However, when more than two clip lines affect a splat, the results can be ambiguous, making it impossible to represent complex intersections.

Wicke *et al.* [11] created a list for each splat near the intersection line of the objects involved in a Boolean operation. The elements in this list are called *clip partners* and they are close enough to the splat to be able to clip it. After the splats’ clip partners are formed, clippings are performed during rasterization of the ellipse. The coordinates of each fragment are mapped to the object coordinate system and used to determine whether the fragments are on the surface. However, the dependency on modeling information, CSG in this case, to determine whether a point belongs to the surface or not, limits the applicability of the method, because many models either are obtained by 3D scanning technology, or simply do not have that kind of information available.

III. CLIP PARTNERS

Wicke *et al.* [11] used the term *clip partner* to name the splats responsible for clipping another splat. This same term is used here to denote the subset of neighboring splats that participate in the process of clipping a splat in order to adapt it to the edge’s curve. Since a clipped splat also clips its *clip partners*, it uses only neighborhood relations that have symmetry. The neighborhood estimation used here is written as: splat S_i is a neighbor of splat S_j whenever:

$$r_i + r_j < d_{ij}, \quad (1)$$

where r_i and r_j are the radii of the splats S_i and S_j , respectively, and d_{ij} is the distance between their centers. In the case of elliptical splats, r_i and r_j are the major axes of the ellipses. The set containing splat S_i ’s neighbors is called $N(S_i)$.

The clippings are needed only between splats belonging to different surfaces. Thus, a discontinuity detection method can remove all neighbors of a splat that share the same surface with it. A simple and efficient alternative is Kobbelt’s detection method [12], which states that a splat does not share the same surface with its neighbors if the dot product between its normal vector and its neighbor’s normal vector is less than a threshold chosen by the user. In other words, the set of neighbors of a splat S_i which belong to different surfaces is:

$$N_{so}(S_i) = \{S_j \in N(S_i) \mid \mathbf{n}_i \cdot \mathbf{n}_j < w_s\}, \quad (2)$$

where \mathbf{n}_i and \mathbf{n}_j are the normal vectors to S_i and S_j , respectively, and w_s is the threshold chosen to detect splats of different surfaces. Since the dot product is a symmetric operation:

$$S_i \in N_{so}(S_j) \Leftrightarrow S_j \in N_{so}(S_i). \quad (3)$$

Even after $N_{so}(S_i)$ is set, some neighbors do not cross the splat S_i to require clipping. Thus, the intersection between two splats is identified to form S_i ’s *clip partners* set, $N_{cp}(S_i)$. To identify the intersection between two disks in object space, some steps are necessary: 1) define the intersection line of the planes formed by splats involved, 2) calculate the intersection between this line and both disks forming line segments, 3) check whether these segments have some piece in common. In case the intersection persists until this last test, we conclude

that the disks intersect each other and the neighbor is added to the subset N_{cp} of the splat and vice versa. Otherwise, the splats do not clip each other and their relationship is disregarded.

A. Planes' Intersection Line

The first step in performing collision detection between two circles or ellipses is to find out whether their planes cross each other and, if they do, to calculate the intersection line. Consider splats S_i and S_j , where $S_i \in N_{so}(S_j)$ and $S_j \in N_{so}(S_i)$. The planes of the splats S_i and S_j are defined as

$$\begin{aligned} \alpha_i &= \{\mathbf{P} \mid \mathbf{n}_i \cdot \mathbf{P} = \mathbf{n}_i \cdot \mathbf{C}_i\} \\ \alpha_j &= \{\mathbf{P} \mid \mathbf{n}_j \cdot \mathbf{P} = \mathbf{n}_j \cdot \mathbf{C}_j\}, \end{aligned} \quad (4)$$

where \mathbf{C}_i and \mathbf{C}_j are the splats' centers and \mathbf{n}_i and \mathbf{n}_j are splats' normal vectors.

The direction of the intersection line is calculated as

$$\mathbf{d} = \mathbf{n}_i \times \mathbf{n}_j. \quad (5)$$

When the cross product in Equation 5 is a null vector, the two planes are parallel or coincide. However, this condition should not be satisfied because it is contrary to the definition of N_{so} and it is known that the angle between the normal vectors \mathbf{n}_i and \mathbf{n}_j is greater than the predefined threshold w_s . Then, the vectors \mathbf{d} , \mathbf{n}_i and \mathbf{n}_j form a basis for the Euclidean space \mathbb{R}^3 . Thus, the canonical origin and these basis-vectors form a reference frame relative to which any point in space can be written as

$$\mathbf{P} = t_i \mathbf{n}_i + t_j \mathbf{n}_j + t \mathbf{d}. \quad (6)$$

Probably, \mathbf{d} , \mathbf{n}_i and \mathbf{n}_j form a non-orthogonal basis, but not necessarily. In the case $\mathbf{n}_i \cdot \mathbf{n}_j = 0$ (orthogonal), these vectors form an orthogonal basis.

Equation 6 can be used to represent the parametric equation of the intersection line, if t is defined as the parameter of the line and $t_i \mathbf{n}_i + t_j \mathbf{n}_j$, as the position vector, relative to the canonical frame at point \mathbf{P}_o where the line crosses the plane $\mathbf{n}_i \mathbf{n}_j$. The parameter t is considered to be zero at this point. Thus, by using the dot product of \mathbf{P}_o with \mathbf{n}_i and \mathbf{n}_j , we obtain the following system of equations with unknowns t_i and t_j :

$$\begin{cases} \mathbf{P}_o \cdot \mathbf{n}_i = t_i(\mathbf{n}_i \cdot \mathbf{n}_i) + t_j(\mathbf{n}_j \cdot \mathbf{n}_i) \\ \mathbf{P}_o \cdot \mathbf{n}_j = t_i(\mathbf{n}_i \cdot \mathbf{n}_j) + t_j(\mathbf{n}_j \cdot \mathbf{n}_j) \end{cases} \quad (7)$$

By finding the solutions of the system shown in Equation 7, \bar{t}_i and \bar{t}_j , and substituting in Equation 6, all the points located at the intersection of the planes formed by splats S_i and S_j are found. In other words, the intersection line is

$$\mathbf{P} = \mathbf{P}_o + t \mathbf{d}, \text{ where } \mathbf{P}_o = \bar{t}_i \mathbf{n}_i + \bar{t}_j \mathbf{n}_j \quad (8)$$

B. Intersection Between Line and Splat

Near the edges, it is possible that a splat belong to N_{so} of other samples without intersecting them, in which case clipping is unnecessary. It is also possible that the intersection line of two planes crosses only one of the two splats. In this case, clipping only one of the splats can create holes in the

surface. For this reason, it is very important to know when a line intersects a circular or elliptical splat.

The intersection line of the planes is defined by the point \mathbf{P}_o and the vector \mathbf{d} found in Equation 8. The unit vectors \mathbf{e}_u and \mathbf{e}_v along the principal axes of an elliptical splat, the unit vector \mathbf{n} , normal to the surface, and its center, \mathbf{C} , form a splat reference frame where the parameters of the intersection line of the planes can be written. Note that this line belongs to the splat's plane. Thus, after the transformation to the splat's reference frame, the components in the splat normal direction are zero. Thus, \mathbf{P}_o and \mathbf{d} can be rewritten as

$$\begin{aligned} \mathbf{P}_o &= (P_{ou}, P_{ov}) \leftrightarrow \mathbf{P}_o = P_{ou} \mathbf{e}_u + P_{ov} \mathbf{e}_v \\ \mathbf{d} &= (d_u, d_v) \leftrightarrow \mathbf{d} = d_u \mathbf{e}_u + d_v \mathbf{e}_v. \end{aligned} \quad (9)$$

In the 2D coordinate system defined by the vector-basis \mathbf{e}_u and \mathbf{e}_v , the parametric equation of the intersection line can be written as

$$\mathbf{P}(t) = \mathbf{P}_o + t \mathbf{d} \quad (10)$$

and the equation of the ellipse as

$$\frac{u^2}{r^2} + \frac{v^2}{s^2} = 1 \quad (11)$$

where r and s are the ellipse's semi-axes with the same directions of \mathbf{e}_u and \mathbf{e}_v , respectively. The intersection points of the line with the splat are the roots of the follow quadratic equation in t

$$At^2 + 2Bt + C = 0, \text{ where } \begin{cases} A = \frac{d_u^2}{r^2} + \frac{d_v^2}{s^2} \\ B = \frac{P_{ou} d_u}{r^2} + \frac{P_{ov} d_v}{s^2} \\ C = \frac{P_{ou}^2}{r^2} + \frac{P_{ov}^2}{s^2} - 1 \end{cases} \quad (12)$$

A circular splat is a particular case of a elliptical splat where $r = s$. Thus, the intersection of a line with the splat can be calculated with a simplified version of Equation 12 as

$$At^2 + 2Bt + C = 0, \text{ where } \begin{cases} A = \mathbf{d} \cdot \mathbf{d} \\ B = \mathbf{P}_o \cdot \mathbf{d} \\ C = \mathbf{P}_o \cdot \mathbf{P}_o - r^2 \end{cases} \quad (13)$$

The solutions of Equation 12 or Equation 13, t_1 and t_2 , are used to determine the resulting line segment of intersection between the intersection line of the planes and the splat.

C. Intersection Between Line Segments

Since the intersection segments of the line with the splats have been determined, a final intersection test is done to check the intersection of these two splats. This test is necessary, because, even if these splats intersect the intersection line of the planes, it is possible that these splats do not touch each other (Figure 2a).

Let t_{i1} and t_{i2} be the solutions of Equation 12 resulting from the intersection of the line with the splat S_i , where $t_{i2} > t_{i1}$. Similarly, let t_{j1} and t_{j2} be the solutions of Equation 12 resulting from the intersection of the line with the splat S_j , where $t_{j2} > t_{j1}$. If the intervals $[t_{i1}, t_{i2}]$ and

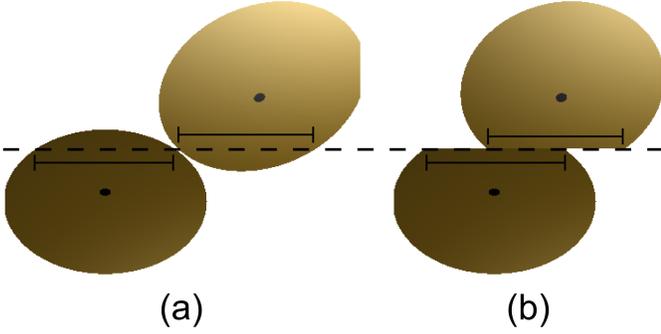


Fig. 2. Necessity of the segments intersection test for detection of false positives. (a) Although both splats intercepting the intersection line, if the segments do not intersect, they do not cross each other and they do not need clipping. (b) If the segments have some intersection, clipping is necessary and done on the intersection line.

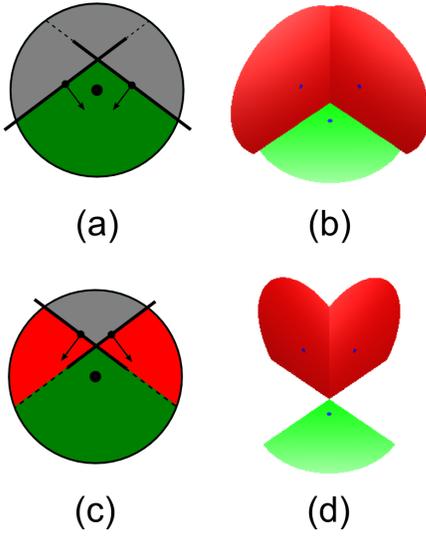


Fig. 3. Ambiguity case do not treated by the Zwicker's method [10]. (a) Scheme where the clipping against both clip partners fits well the splat. (b) Model that exemplifies the scheme (a). (c) Scheme where the clipping against both clip partners creates holes in the surface. (d) Model that exemplifies the scheme (c).

$[t_{j_1}, t_{j_2}]$ are not disjoint, the splats cross each other and need to be clipped (Figure 2b), otherwise, there is no intersection, and any indication to the contrary is a false positive.

After this last test, the neighbors belonging to $N_{so}(S_i)$ of a given splat S_i , which intersect it are inserted in the subset $N_{cp}(S_i)$ and called *clip partners*.

IV. TYPES OF CLIPPING

A. Splat Clipping Problem

In order to adapt a splat to an edge, the splat must be clipped against the planes of its clip partners. These clip partners may either share the same center [3] or have distinct centers [10], [11].

Zwicker *et al.* [10] generalized the idea of cutting splats, eliminating the restriction that the splats should have coincident centers. In this method, a splat detected as belonging to

an edge is clipped by the intersection line of that splat's plane with the plane of its clip partner. The amount of clip partners used to clip the sample is left as a user-defined parameter, however, that choice is not trivial. Using only the closest clip partner has problems in the representation of curved edges, like in Pauly's method [3], and do not represent the corners.

Clipping against all clip partners brings a more serious problem: ambiguity. Figure 3 exemplifies two cases that are treated in the same manner by Zwicker's method, which, however, should receive distinct treatment. In the case illustrated in figures 3a and 3b, the union of clipped areas for each one of the clip partners fits correctly the green splat. However, the case exemplified in figures 3c and 3d have the same intersection lines of the previous case and cannot be treated identically, because some areas are clipped in front of splats of other surface. In this case, one way to properly adjust the green splat is to clip the intersection area of clipping areas of each clip partner. By performing each clipping independently, a fragment is removed if it is in the clipping area of at least one of the clip partners. Thus, regardless of the case, a fragment is removed if it is in the union of clipping areas. To avoid such ambiguities, the clip partners must be analyzed a priori, then a correct clipping on the splat can be performed.

Wicke *et al.* [11] used the two closest splats to classify a 3D point as inside or outside an object represented by splats. That classification changes when the two splats involved form a concave or a convex area. This is particularly important for CSG classifications, but in this case, this technique was developed to better cut the splats near the edges. All splats that cross the intersection line between the two objects involved in the Boolean operation are rasterized differently. Each fragment is mapped to the object space and then classified against the splats of the other object. The 3D point corresponding to the fragment is said to be external or internal in relation to the other object, and depending on the Boolean operation performed, the fragment is retained or deleted. The classification is done using the following definition: a point is viewed by a splat when:

$$(\mathbf{P} - \mathbf{C}_i) \cdot \mathbf{n}_i > 0, \quad (14)$$

where \mathbf{P} is the tested point and \mathbf{C}_i and \mathbf{n}_i are the center and the unit normal vector of a splat S_i , respectively. This definition is very important and is used recurrently in this work. In addition to using information from the CSG modeling to decide when a fragment should be clipped, Wicke's method perform a search for the two closest clip partners to a given fragment. This slows down rendering by adding more calculations in the rasterization step.

B. Adaptive Splat Clipping

As in [11], the clip partners need to be classified according to their orientation to decide between intersection or union of the clipped areas. To identify concave or convex areas among the clip partners set, first, the centers of all the elements in the set are used in the computation of their geometric center. Next, the set is classified as a concave set, if the computed geometric

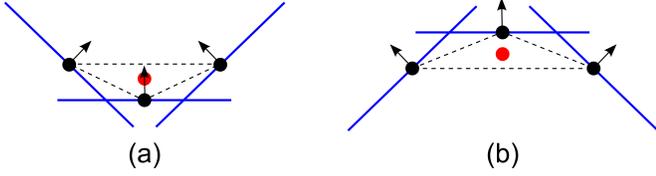


Fig. 4. Classification between concave and convex areas. (a) In concave areas, all the elements “see” the midpoint of their centers. (b) In convex areas, no element “see” the midpoint.

center is “seen” by all elements in the set (Figure 4a); the set is classified as a convex set, if the geometric center is not “seen” by at least one element in the set (Figure 4b). In other words:

$$\begin{aligned} N_{cp} \text{ is concave} &\Leftrightarrow \forall S_i \in N_{cp}; (\mathbf{M} - \mathbf{C}_i) \cdot \mathbf{n}_i > 0 \\ N_{cp} \text{ is convex} &\Leftrightarrow \forall S_i \in N_{cp}; (\mathbf{M} - \mathbf{C}_i) \cdot \mathbf{n}_i < 0 \end{aligned} \quad (15)$$

where:

- N_{cp} is the clip partners set of a splat S ;
- \mathbf{C}_i and \mathbf{n}_i are the center and the normal vector of the clip partner S_i , respectively;
- \mathbf{M} is the geometric center of the clip partners’s centers.

Only the clip partners classification in convex or concave areas is not enough to choose between union and intersection of the clipped areas. It is necessary to check the cases in which S forms concave or convex areas with its clip partners. The same classification as described in Equation 15 can be used in this case. Thus:

$$\begin{aligned} S \cup N_{cp} \text{ is concave} &\Leftrightarrow \forall S_i \in N_{cp}; (\mathbf{C}_i - \mathbf{C}) \cdot \mathbf{n} > 0 \\ S \cup N_{cp} \text{ is convex} &\Leftrightarrow \forall S_i \in N_{cp}; (\mathbf{C}_i - \mathbf{C}) \cdot \mathbf{n} < 0 \end{aligned} \quad (16)$$

where, \mathbf{C} and \mathbf{n} are the center and the normal vector of the splat S .

Crossing the information obtained in the classifications described in equations 15 and 16, the choice between union and intersection of the clipped areas can be done correctly. Choices are made for each combination of cases as described in Figure 5. In all subfigures, the green splat has his clip partners set classified as concave or convex, as well as his interaction with that set. In each subfigure, the left image shows an example that has the classification of N_{cp} and $S \cup N_{cp}$ shown below and the clipped area is highlighted in red. The right image illustrates the same case in different point of view (the green splat is clipped by his clip partners rendered in red). Below each subfigure, it is shown the type of clipping depending on the classifications of N_{cp} and $S \cup N_{cp}$.

V. RASTERIZATION

All the steps presented in the previous are performed offline, i.e., during a pre-processing phase. However, the clipping of the splat is actually done during its rasterization, the first step in splat rendering. Splat rasterization determines the image pixels that are affected by the splat’s projection.

To render clipped splats, fragments must be removed after being rasterized by some splat rasterization method. However,

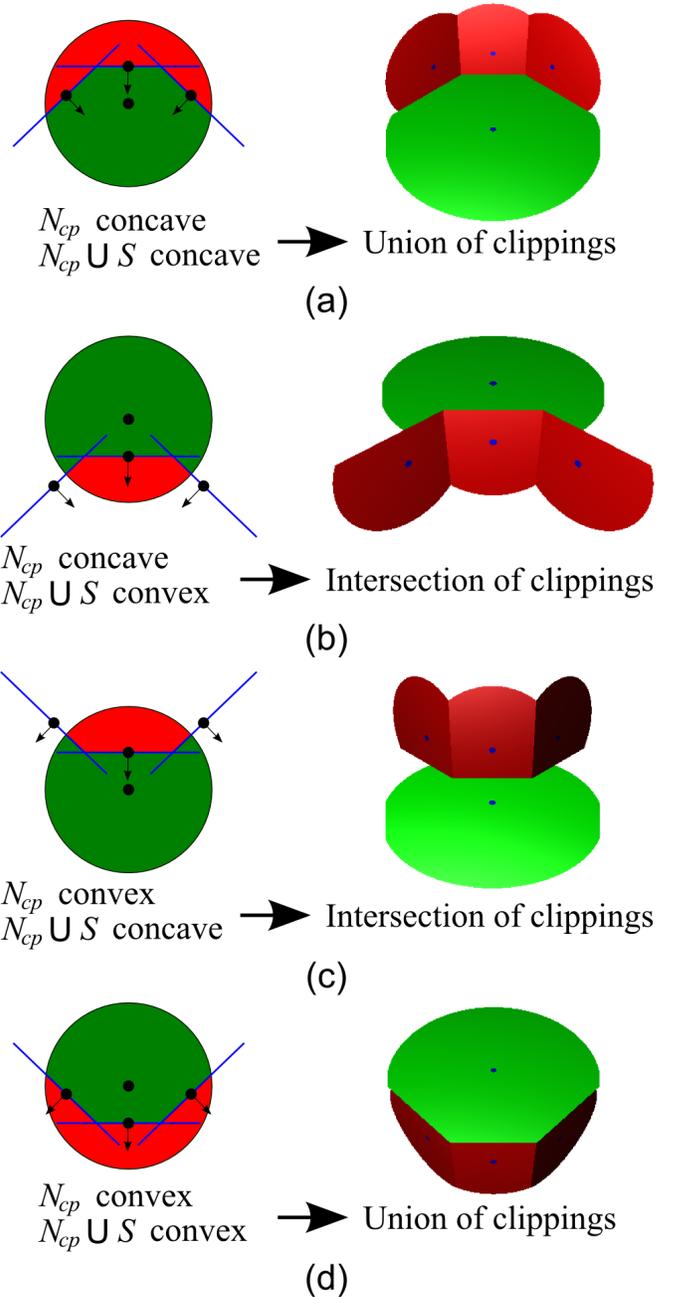


Fig. 5. Choosing between the union or intersection of the clippings according to the classifications of concavity and convexity between clip partners and between surfaces.

to decide when a fragment is removed, the fragment is mapped to a 3D point in object space. Thus, splat rasterization methods which use affine mappings [1], [13], [10] are not adequate. Thus, Botsch’s rasterization technique [14], [15] is used. This technique is efficient and does not show perspective errors.

An $s \times s$ image-space square will be rasterized in the frame-buffer. Thus, a local ray casting is performed among all the square’s pixels centered in the pixel touched by the splat’s center projection. An efficient evaluation of s uses the splat radius r and the depth value of the splat’s center

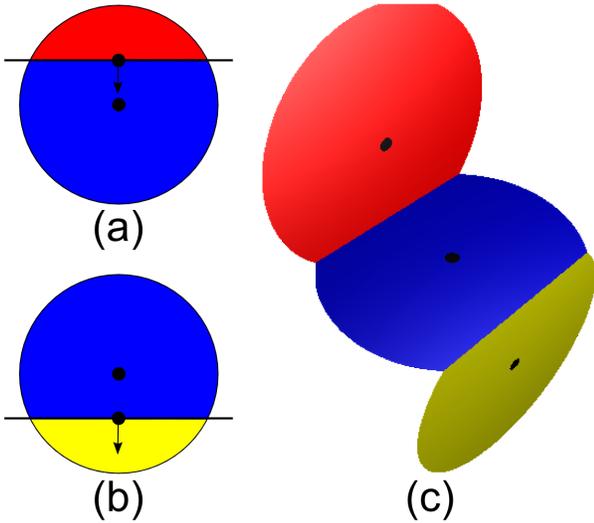


Fig. 6. Choosing the clipping directions. (a) If the clip partner sees the splat’s center, the clipping direction is its normal opposite. (b) If the clip partner does not see the splat’s center, the clipping direction is the same as its normal. (c) Example of both cases of clipping direction choice.

$\mathbf{C} = (c_x, c_y, c_z)$ in camera coordinates:

$$s = 2r \cdot \frac{n}{c_z} \cdot \frac{h}{t - b} \quad (17)$$

where, n , t and b are the viewer parameters near, top and bottom, and h is the viewport’s height (in pixels). If the splat is elliptical, the value of the major axis is used as r .

Let \mathbf{P} be the intersection of a ray with the splat’s plane. If the splat is circular, the fragment is discarded when the distance from \mathbf{P} to the splat’s center \mathbf{C} is greater than r . If the splat is elliptical, let \mathbf{u} and \mathbf{v} be unit vectors in the direction of the two principal axes. Let (u, v) be the coordinates of the point \mathbf{P} in the splat’s coordinate system. If $u^2 + v^2 \leq 1$, the pixel is rendered, otherwise, the fragment is discarded.

For splats that have clip partners, the point \mathbf{P} is also tested against the clipping planes. If the splat S sees its clip partners’ centers, i.e., $S \cup N_{cp}$ is concave, the clip partners’ normal vectors are directed to center of S and the clipping areas consist of points not seen by the clip partners (Figure 6a). If $S \cup N_{cp}$ is convex, the clip partners’ normal vectors are not directed to the center of S and the clipping areas consist of points seen by the clip partners (Figure 6b). Thus, if $S \cup N_{cp}$ is concave, then:

$$\mathbf{P} \text{ is clipped by } S_j \Leftrightarrow (\mathbf{P} - \mathbf{C}_j) \cdot \mathbf{n}_j < 0, \quad (18)$$

and, if $S \cup N_{cp}$ is convex, then

$$\mathbf{P} \text{ is clipped by } S_j \Leftrightarrow (\mathbf{P} - \mathbf{C}_j) \cdot \mathbf{n}_j > 0, \quad (19)$$

where $S_j \in N_{cp}$.

With the definition of clipping areas for each clip partner according to the concavity or convexity of the edge, it remains to perform union or intersection of these clipping areas, depending on the classification defined in Figure 5. In case of union, the point is discarded if it is in the clipping area of

at least one of the clip partners. In case of intersection, the point must be in the clipping areas of all clip partners to be dropped. That is, if the choice is union of clipping areas, then:

$$\mathbf{P} \text{ is clipped} \Leftrightarrow \exists S_j \in N_{cp}; \mathbf{P} \text{ is clipped by } S_j, \quad (20)$$

otherwise:

$$\mathbf{P} \text{ is clipped} \Leftrightarrow \forall S_j \in N_{cp}; \mathbf{P} \text{ is clipped by } S_j. \quad (21)$$

VI. RESULTS

Several types of artifacts arising from naive renderings of point-based models can be improved with increased sampling. However, the surface splatting’s complexity is directly proportional to the amount of samples to be projected. Thus, to achieve the interactive frame rates required in various real-time applications, we seek to render models correctly even when the sampling conditions are not so favorable.

To compare the proposed method with previous methods, we used three strategies for clipping the splats: 1) each splat’s clip partner cut it independently of the others; 2) just the closest splat’s clip partner cut it; 3) the clippings are adjusted to the edge using the proposed method shown in Section IV. The first two strategies are implementations of Zwicker’s method[10]. Methods such as those proposed in [2] and [3] were not tested because they use modeling information and add new samples to the model. Wicke’s method [11] has similar results to those achieved by our technique, but in addition to using modeling information, the clipping classification is performed for each rasterized fragment, which considerably reduces the performance gained by using fewer samples.

In the first strategy, the ambiguity of clippings leads to appearance of holes over the curved edges. Figure 7a shows the chess tower model rendered using that strategy. The holes were highlighted in red. The enlarged images at the right show the two cases where it is necessary to use the intersection of the clipping areas: 1) convex edge and clip partners forming a concave area (Figure 7a above); 2) concave edge and clip partners forming a convex area (Figure 7a below). In both cases, the clipped area is larger than necessary, which causes holes over the edge to appear. The proposed method adapts the splats to the edge’s curvature because of previous classifications and the use of intersection of the clipping areas in the two cases cited above. Figure 7b shows the same chess tower model rendered using our strategy. The same areas are highlighted to show the correct treatment of the curved edges.

Using the second strategy, the results are better than those found using the previous strategy for models with very small splats. However, in flat areas, it is preferable to use a smaller number of splats, thus reducing the complexity of the rendering. When the sizes of the splats are very different around the edge, the largest splats make jagged edges (figures 8b and 8c), because they still cross clip partners other than the nearest. All clip partners must be used in processing the splat’s clipping to adapt it to the curve of the edge (figures 8d and 8e). The second strategy does not treat corners.

Figure 9 shows one of the motivations of this work: to achieve good renderings of models with lower sampling.

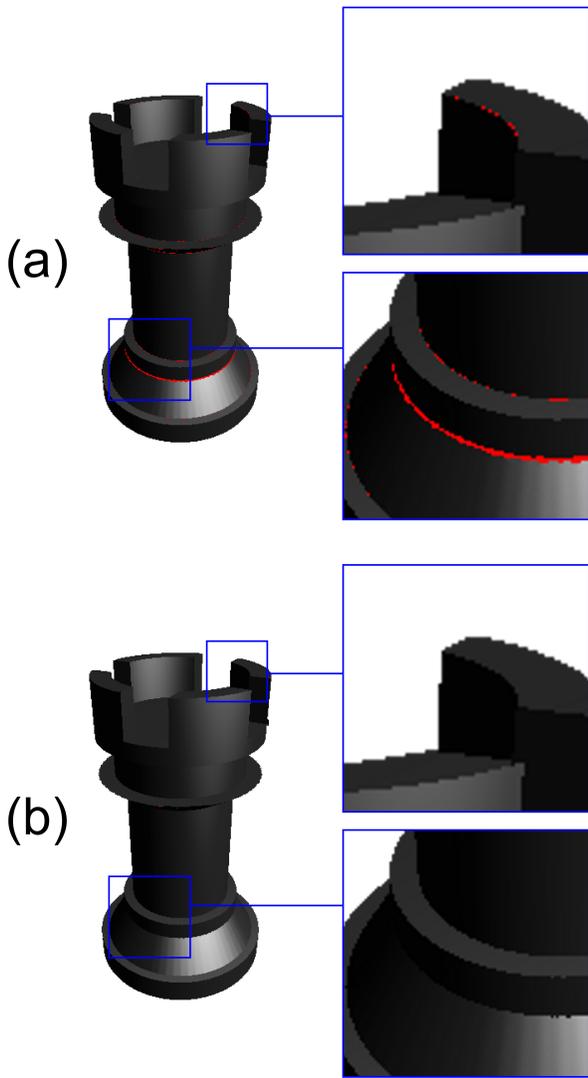


Fig. 7. Comparing the adaptive method with the independent clipping of each clip partner. (a) The tower model rendered using the first strategy. We can see the presence of holes, highlighted in red, in the curved edges of the model, because of the ambiguity problem. (b) The same tower model rendered using the adaptive method. The edge classification and clip partners classification allow clip the splats in different ways depending on the edge's curve and the edge's concavity or convexity.

Models with high sampling around the edges have good rendering regardless of the method used, however the detection of splats to be clipped, the projection of the samples and even the surface reconstruction step become more time consuming. Figure 9 shows the decrease in visual quality for several simplification of a model. It is clearly noticeable that the rendering quality is not as tightly related to geometric complexity for the proposed strategy (figures 9e and 9f) as for the first strategy (figures 9c and 9d).

Figure 10 shows some models with sharp features rendered using the adaptive clipping method.

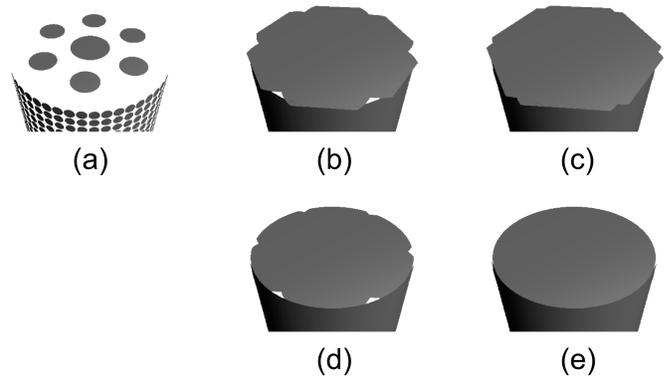


Fig. 8. Comparing the adaptive method with the clipping of only nearest clip partner. (a) Distribution and relative size of the model's splats. (b) Clipping against only the nearest clip partner. (c) Resulting image when completely fill the surface using the second strategy. (d) Clipping against all clip partners, but in adaptive way. (e) Resulting image when completely fill the surface using adaptive strategy.

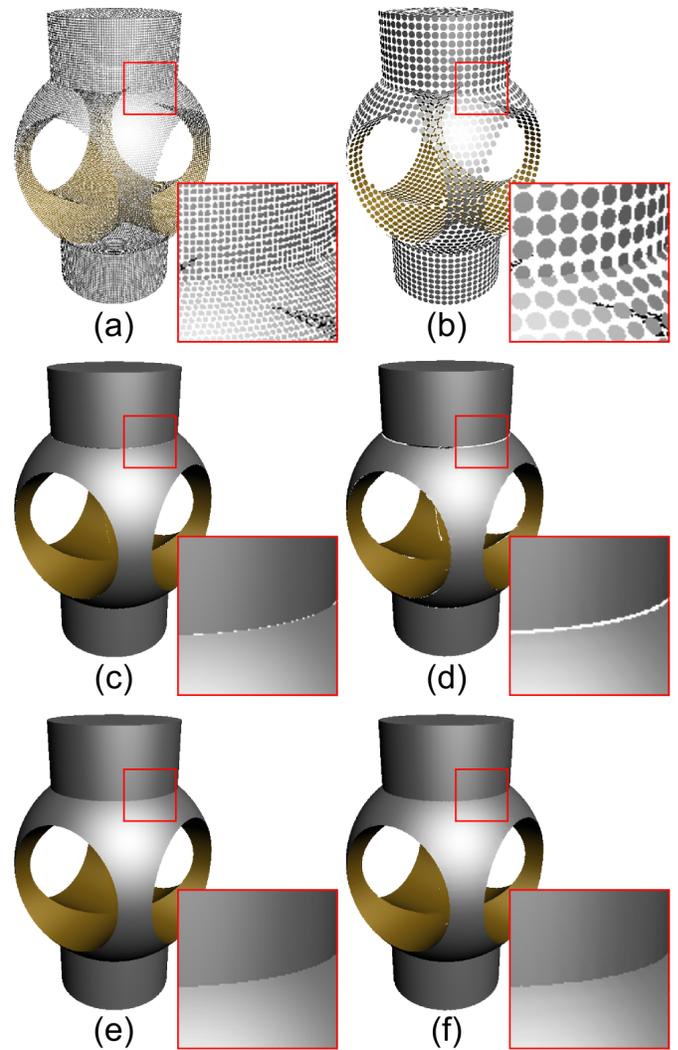


Fig. 9. Comparing the techniques in relation to the number of splats of the model. (a) A model with 45734 samples. (b) The same model in (a), but with only 4276 samples. (c) The model in (a) rendered using the first strategy. (d) The model in (b) rendered using the first strategy. (e) The model in (a) rendered using our strategy. (f) The model in (b) rendered using our strategy.



Fig. 10. Models rendered using the proposed clipping method.

VII. CONCLUSION AND FUTURE WORKS

Although 3D scanners generate thousands of points and modern graphic cards render models with high rates of splats per second, as with any other rendering primitive, the processing costs are still proportional to the number of primitives that we use to represent a given object. This is why complexity reduction for splat-sampled geometry is as important as it is for triangle meshes. For this reason, this work aims at achieving high quality renderings of models with sharp features, but with low density of samples.

In this paper, a proper way of clipping splats was presented and discussed. The proposed technique fits well the clipped splats to the edges, using classification of neighbors, which belong to other adjacent surfaces. The clip partners and edge classification help to adapt splats near sharp features through clippings, thus avoiding the appearance of artifacts. This method assumes that the clip partners orientations are consistent, i.e., all clip partners form a concave or convex area, with no inflections. This kind of restriction is not very strong, even in simplified models. However, the lower the amount of samples, the greater are the chances of inflection points to occur. A possible future work is to separate the set of clip partners in several subsets, in which there is no inflection points.

Clipping still fails in certain splats near the corners, where some clip partners may be above the splat and others below, which causes incorrect classification. The same suggestion proposed earlier can also be applied here: a subset of clip partners above the splat and other subset of clip partners below the splat. Another possible solution would be to build a polyline that connects all segments of the intersection, adapting the clipping optimally. However, the excessive cost of this computation for such rare cases makes this approach

unattractive.

REFERENCES

- [1] M. Zwicker, H. Pfister, J. van Baar, and M. Gross, "Surface splatting," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 371–378.
- [2] B. Adams and P. Dutré, "Interactive boolean operations on surfel-bounded solids," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. ACM, 2003, pp. 651–656.
- [3] M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross, "Shape modeling with point-sampled geometry," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. ACM, 2003, pp. 957–964.
- [4] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross, and L. J. Guibas, "Meshless animation of fracturing solids," in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH '05. ACM, 2005, pp. 957–964.
- [5] M. Levoy and T. Whitted, "The use of points as display primitive," Computer Science Department, University of North Carolina at Chapel Hill, Tech. Rep., 1985.
- [6] J. P. Grossman and W. J. Dally, "Point sample rendering," in *Proceedings of Eurographics Workshop on Rendering 98*, 1998, pp. 181–192.
- [7] H. Pfister, M. Zwicker, J. van Baar, and M. Gross, "Surfels: Surface elements as rendering primitives," in *Proceedings of SIGGRAPH 00*, 2000, pp. 335–342.
- [8] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, and C. Silva, "Point set surfaces," in *Proceedings of IEEE Visualization 01*, 2001, pp. 21–28.
- [9] S. Fleishman, D. Cohen-Or, M. Alexa, and C. T. Silva, "Progressive point set surfaces," *ACM Transactions on Graphics* 22, vol. 4, 2003.
- [10] M. Zwicker, J. Rasanen, M. Botsch, C. Dachsbacher, and M. Pauly, "Perspective accurate splatting," in *Proceedings of Graphics Interface 2004*, ser. GI '04. Canadian Human-Computer Communications Society, 2004, pp. 247–254.
- [11] M. Wicke, M. Teschner, and M. Gross, "Csg tree rendering for point-sampled objects," in *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference*, ser. PG '04. IEEE Computer Society, 2004, pp. 160–168.
- [12] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature sensitive surface extraction from volume data," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '01. ACM, 2001, pp. 57–66.
- [13] M. Botsch and L. Kobbelt, "High-quality point-based rendering on modern gpus," in *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, ser. PG '03. IEEE Computer Society, 2003, pp. 335–343.
- [14] M. Botsch, M. Spornat, and L. Kobbelt, "Phong splatting," in *Proceedings of Symposium on Point-Based Graphics 2004*, 2004, pp. 25–32.
- [15] M. Botsch, A. Hornung, M. Zwicker, and L. Kobbelt, "High-quality surface splatting on today's gpus," *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics*, vol. 0, pp. 17–141, 2005.