Eurographics Symposium on Point-Based Graphics (2007)
M. Botsch, R. Pajarola (Editors)

Surfel-Based Billboard Hierarchies for Fast Rendering of 3D-Objects

Mathias Holst and Heidrun Schumann

Institute of Computer Science and Electrical Engineering, University of Rostock, Germany

Abstract

In recent years point hierarchies have been shown to be efficient for rendering high-detailed objects. Since texture access is much faster than vertex processing we propose to increase efficiency of point based rendering of arbitrary surfaces using image-based techniques. For this purpose, we combine surfels and billboards. Such a surfel billboard *contains a snapshot of the geometry it represents, and can be used to replace this geometry, if the actual viewing direction is close to the direction the snapshot was taken from. Such surfel billboards can be arranged in a hierarchy to create an impostor for arbitrary view-dependant LOD. In this paper we develop a framework that contains automatic surfel billboard placement, surfel billboard hierarchy creation and a carefully adaption of texture sizes that considers probability of billboard validity and available texture memory. Furthermore, we show how surfel billboard hierarchies can be combined with traditional triangle-based multi-resolution techniques to realize a hybrid rendering with a seamless LOD-transition using a user-defined image error.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]:

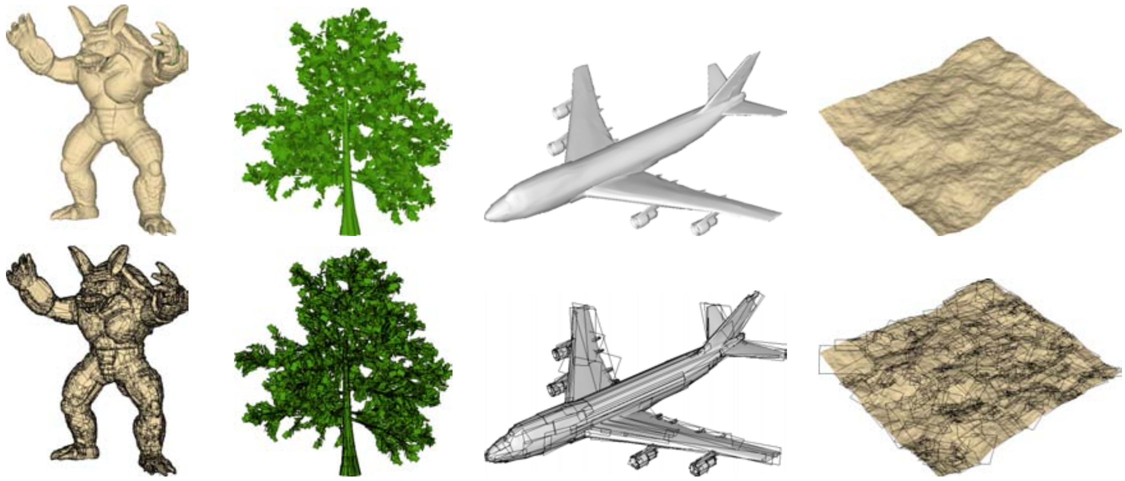
1. Introduction

A general problem in computer graphics is to display large scenes with many high-detailed objects with interactive frame rates. Due to increasing output resolutions and growing user/consumer expectation this problem does not decrease with modern graphical hardware. One way to attenuate this problem is to use several levels of detail (LOD) of the same model to adapt the required accuracy to its size in image-space. To specify such simplifications several strategies have been established over the last years:

- Mesh-Based: Local or global simplification operations are applied to the original mesh.
- Point-based: A given point representation of the original model is simplified by merging points to larger ones. In realtime applications points are generally parameterized with a normal and a radius to form a circular disc. These graphical primitives are also known as *surfels* (from surface elements).
- Impostor-based: A set of snapshots of the original model is taken beforehand and elements (*billboards*) of this set are blended into the image instead of the model. These billboards are called an *imposter* of the original geometry. This approach is known as image-based in contrast to the previous methods which are geometry-based.

All of these approaches are general but have a specific domain limited by efficiency. Mesh-based simplification is efficient for manifold meshes, but is not efficient for weak connected meshes, like trees, due to simplification constraints. Point based simplifications are effective for all surfaces in 3D, but are not efficient in any case. This is quite contrary to triangle meshes, because even for simple models many points have to be used for an adequate rendering. Image based approaches provide a very fast rendering but are often limited to special objects or to specific scenarios including a limited camera position/orientation, e.g. building or landscape exploration.

To overcome these drawbacks, we propose to combine all of these techniques using only one LOD data structure. The key idea is to combine unconnected surfels with billboards to form so called *surfel billboards*. Generally speaking, these are textured surfels. Such a billboard represents a small surface part on a specific LOD. Since texture access is much faster than vertex processing, this results in a significant framerate increase compared to point-based approaches. Therefore, this technique is well suited for arbi-



3963 surfel billboards 4687 surfel billboards 510 surfel billboards 456 surfel billboards

Figure 1: *Different types of 3D models rendered by surfel billboards (upper row). Same renderings with added surfel boundaries (lower row).*

trary objects as shown in fig. 1. By arranging these billboards in a hierarchy, similar to point trees, you are able to obtain an appropriate view-dependant LOD quickly. Every LOD can be seen as an impostor of the original geometry. To preserve texture memory we propose to reduce the hierarchy to important surfel billboards and limit the usage of them to lower LOD. For higher LOD a triangle-based multi-resolution hierarchy is used instead.

2. Related Work

The approach proposed in this paper combines elements of surfel-based rendering, image-based rendering and hybrid rendering. In the following we briefly summarize publications relating to these topics.

Surfel-Based Rendering The usage of points instead of polygons to render 3D objects was proposed in [LW85] and bases on the idea that if a polygon covers only one pixel in image space, it could be rendered as a point as well. This idea was advanced in [PZvBG00] to parameterize points as a circular disk in object space forming unconnected surface elements (short: surfels). To achieve a high-quality rendering result overlapping surfels have to be blended (called *splatting*). This was first described in [ZPvBG02].

To decrease the number of surfels LOD-point-hierarchies can be used. *QSplat* ([RL00]) was a first framework that focuses on this approach. How such point hierarchies can be generated by various subdivision schemes was shown in [PGK02]. Another approach to decrease the number of surfels was proposed in [HS07]. Here surfel trees are combined with normal mapping. In contrast, we propose a more general approach that also includes an optimized billboard placement and an adapted hierarchy creation. Furthermore it combines surfel billboards with triangle-based LOD to preserve texture-memory. Since texture information suffices to guarantee a high-quality rendering, a sophisticated splatting of billboards is not necessary compared to other point-based approaches.

Image-Based Rendering In the Quicktime VR System developed by Chen [Che95], cylindrical panoramic images are obtained from a fixed viewpoint and are re-projected for display. User movement is restricted to zooming and rotation around the object.

Layered impostors are proposed in [Sch98] to increase the degree of impostor validity. This was done by using a layered billboard composition, where layers are displaced against each other in respect to the viewing angle. This approach is well suited for city model explorations, where the camera movement is highly restricted, but can get non-efficient for more degrees of freedom because many impostors have to be created for one object. Contrarily, our approach is mainly designed for arbitrary camera positions and orientations.

Billboard clouds were proposed in [DDSD03]. Here the original object was separated into meaningful planes, and the geometry faces are projected to their closest plane, in order to form billboards. Afterwards, this billboard set can be used as an impostor. Similar to layered impostors only one LOD is described. Moreover this LOD is not view-dependant. This makes it unsuitable for large objects, like landscapes. In contrast we use a hierarchy of billboards to overcome these drawbacks by the possibility of arbitrary view-dependant LOD. For an comprehensive overview about image-based techniques we refer to [JWP05] at this point.

Hybrid Rendering The combination of different rendering strategies is an established approach in computer graphics. In [DCSD02] points, lines and triangles are combined for a fast rendering of complex plant ecosystems. Triangle-based LOD was combined with image caching in [HB04] and [SLS*96] to increase efficiency of LOD estimation and rendering. An extension to *QSplat* is the *POP* system [CN01]. Here, triangles are used as a replacement of points for the highest LOD and silhouette refinement. In [DVS03] a fast LOD estimation was proposed for such hybrid hierarchies which uses a serialization of the hierarchy. A first system that combines triangles for more than one high LOD and points for lower LOD was proposed in [CAZ01]. Similar to our approach a multi triangulation hierarchy (short MT, [FMP97]) was combined with a point hierarchy. Compared to this approach we use one surfel billboard to replace a certain set of triangles, instead of using some points to replace one triangle. Moreover, our simplification scheme for creating the surfel hierarchy can be adapted to the needs of the user (texture memory, rendering performance).

3. Surfel Billboards

We consider a surfel billboard as a replacement for a triangle set at lower levels of detail. This is called surfel billboard because it is not connected to other graphical primitives, and because it only represents a small surface part of the original geometry similar to the fundamental idea of surfels. To create a surfel billboard its plane has to be defined in object space, and a snapshot of the original geometry has to be taken onto this plane looking in normal direction. In the following we describe this procedure in more detail. We introduce criterions of validity for such a billboard. Furthermore, different rendering strategies are discussed.

3.1. Surfel Billboard Creation

In contrast to the well known billboard cloud approach [DDSD03] we use a given subdivision of the original mesh into disjoint triangle subsets. How this subdivision will be generated is explained in Sect. 4. To create a surfel billboard b for such a triangle subset $\mathcal{T}_b$ we propose to use a bounding box approach as it can be seen in fig. 2. This includes three steps:

1. Create the smallest oriented bounding box OBB_b of the triangle set $\mathcal{T}_b$ (fig. 2(a)).
2. Place the billboard b in the center of OBB_b, aligned and sized according to the largest plane (fig. 2(b)).
3. Create a texture of the original geometry (fig. 2(c)). Since a billboard can be seen from both sides two textures have to be generated.

The fastest algorithm known in literature to compute the smallest oriented bounding box (OBB) of a given point set is $O(n^3)$ [O'R85]. This can only be computed for small point sets in a justifiable time. Thus, an approximation has to be used. For this purpose, a principal component analysis (PCA) is applied to obtain the orientation of the OBB. This can be done in $O(n)$. Other efficient OBB calculation algorithm (e.g. [BHP99]) may be used as well.

To place and texture a surfel billboard we need parameters that describe its position, its size and texture information. Hence, we define a surfel billboard to have the following attributes:

- A center M. This naturally equals the center of its OBB.
- A local orthogonal right-hand coordinate system $(\mathbf{o}_1, \mathbf{o}_2, \mathbf{n})$ which equals the OBB orientation vectors starting with the largest dimension.
- Three lengths w, h and d that equal the half dimensions of the OBB.
- Its texture size (tw, th) and its texture position $(tx, ty)^T$ (potentially for both sides of the billboard). How these value are obtained is shown in Sect. 3.3.

3.2. Validity

The validity of a surfel billboard depends on two criteria:

1. The maximum deviation of the image position of a point on the original geometry and its projection on the billboard plane. This error is called e_p and depends on the viewing situation and the projection type used. We consider a perspective projection in the following.
2. The size of the billboard in image-space. If this size exceeds its texture size no correct rendering result can be guaranteed. We define the number of pixels that a texel covers in image space as the error e_t.

It is evident to define a surfel billboard b to be valid iff

$$valid(b) = e_p(b) \leq \varepsilon \wedge e_t(b) \leq \varepsilon, \tag{1}$$

where ε is a user defined parameter in pixel values. We now focus on the calculation of e_p and e_t.

The correct value of e_p is given by

$$e_p(b) = \max_{\mathbf{p} \in \mathcal{T}_b}(|ip(\mathbf{p}) - ip(proj_b(\mathbf{p}))|), \tag{2}$$

where $\mathcal{T}_b$ is the set of all points on the triangles which are replaced by billboard b, ip is the projection function from object-space to screen-space and $proj_b$ defines the projection of a point to the plane of b. Since the calculation of this term is time-consuming an approximation has to be used instead.

In case of perspective projection the image size s_{img} in pixels of a view-plane aligned line with length l and with eye-space z-coordinate z_{eye} can be approximated by

$$s_{img}(l, z_{eye}) = \frac{l}{z_{eye}} \cdot \underbrace{\frac{h}{2} \cot\left(\frac{\alpha}{2}\right)}_{pc}, \tag{3}$$

where h is the image height in pixel and α is the field-of-view angle defining a projection constant pc. We approximate e_p

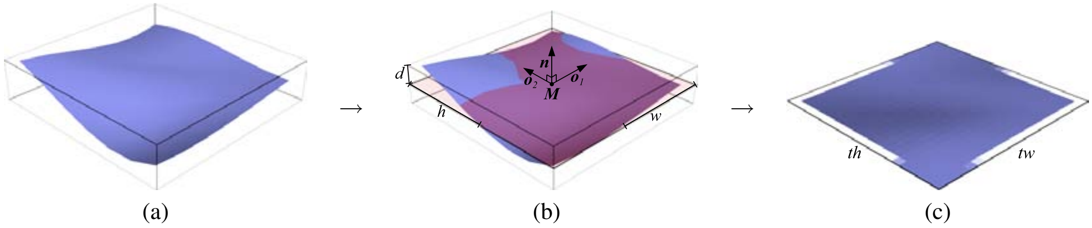


Figure 2: *Surfel billboard creation steps: Triangle set with its oriented bounding box (a). Inserted billboard plane (light red) and surfel billboard geometry (b). Texture size and (front-facing) texture taken from inverse normal direction (c).*

by s_{img} with $z_{M_{eye}}$ and $l = d|\sin(\beta)|$, where β is the angle between eye-space normal $\mathbf{n}_{eye}$ and eye-space midpoint M_{eye} as illustrated in Fig. 3(a). The absolute value bars are required, because e_p does not change when viewing the front- or backplane of the billboard. This yields

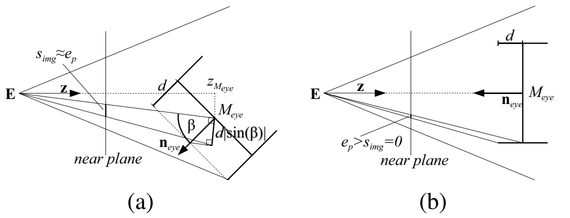


Figure 3: *(a) Geometry for calculating our approximation of e_p. (b) Approximation of e_p is 0, whereas the correct value of e_p is > 0.*

$$e_p(b) \approx \frac{d|\sin(\beta)|}{z_{M_{eye}}} \cdot pc = \frac{d\sqrt{1 - \frac{(M_{eye} \cdot \mathbf{n}_{eye})^2}{||M_{eye}||^2}}}{z_{M_{eye}}} \cdot pc. \quad (4)$$

To avoid square root computation it is convenient to use $e_p^2 \leq \varepsilon^2$ for validity estimation. Note that this approximation is not conservative because it can underestimate the correct value of e_p as shown in Fig. 3(b). In this example our approximation of e_p is 0 whereas the correct value is slightly larger. However, in general this deviation is very small because surfel billboards are mostly used for larger distances.

To calculate e_t we also choose an approximation of the billboard size in image space by using s_{img}:

$$e_t(b) = \max\left(\frac{s_{img}(2w, z_{M_{eye}})}{tw}, \frac{s_{img}(2h, z_{M_{eye}})}{th}\right). \quad (5)$$

With e_p and e_t the validity of a surfel billboard can easily be computed for arbitrary viewing situations.

3.3. Texturing

To compute efficient texture sizes tw, th we propose to use an individual scaling factor for each surfel billboard to adapt its size to texture size which depends on its depth and on the total validity (e_p and e_t). In doing so, first we get a smallest allowed distance z_{min} for every billboard using e_p. Afterwards we use e_t to calculate a valid texture size for this distance.

By choosing $e_p = \varepsilon$ the value of z_{min} can be obtained by

$$z_{min}(b) = z_{M_{eye}} = \frac{d|\sin(\beta)|}{\varepsilon} \cdot pc. \quad (6)$$

Since β depends on the actual viewing situation a realistic a priori upper bound has to be found. In doing so, we introduce a more intuitive user-defined parameter $\theta \in (0,1]$ that defines the probability of β to be as small as needed to satisfy ε at distance z_{min}. θ can be interpreted as the number of allowed viewing directions in ratio to all viewing directions. This corresponds to the ratio of the area of the two caps defined by β (two caps due to the absolute value bars in equ. 6) and the total area of the unit sphere (see Fig. 4):

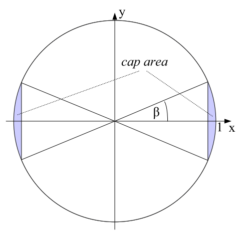


Figure 4: *Sphere caps area defined by angle β illustrated in 2D.*

$$\theta = \frac{4\pi(1-\cos(\beta))}{4\pi} = 1 - \cos(\beta). \quad (7)$$

Hence, z_{min} is given by

$$z_{min}(b) = \frac{d\sqrt{\theta(2-\theta)}}{\varepsilon} \cdot pc. \quad (8)$$

Reliable values of θ should be ≥ 0.25 in order to obtain valueable textures. In our framework a constant probability of 0.5 is used.

After defining z_{min}, now we adjust b's texture size in such a way that e_t equals ε at this distance. This yields

$$\begin{pmatrix} tw \\ th \end{pmatrix} = \begin{pmatrix} s_{img}(2w, z_{min}(b)) \\ s_{img}(2h, z_{min}(b)) \end{pmatrix} \frac{1}{\varepsilon} = \begin{pmatrix} w \\ h \end{pmatrix} \frac{2}{d_i\sqrt{\theta(2-\theta)}}. \quad (9)$$

A remarkable property of this function is that texture sizes do not depend on ε and will not have to be recalculated if its value changes.

Given this texture size a free area in the used texture atlas can be allocated and the texture position $(tx,ty)^T$ can be set accordingly.

We use two texture atlases with the same size and the same texture coordinates for every used surfel billboard to hold normal information for both sides. Using closed surfaces only the outside is often meaningful and viewpoint moving is restricted accordingly. In this case only one texture will be needed if the maximum angle between billboard normal and the normal of a face, replaced by the billboard, is $> \pi/2$. Note, that if all are $> \pi/2$ one texture (for the back) is sufficient, too. More textures will have to be added if other shading attributes are stored for each vertex/face, like color information or texture coordinates. To get a better approximation of the z-buffer value for a rendered pixel of a surfel billboard we additionally use the alpha value of the normal map to store the signed distance value of the original geometry to the billboard plane at this position (in the sense of depth-images).

3.4. Billboard Rendering

Surfel billboards either can be rendered as quadrilaterals or alternatively as points. If using quads more vertices have to be processed. On the other hand the number of pixels that have to be shaded is less compared to using points. This is due to the fact that the size of a point has to be conservatively approximated in image space, which results in useless fragments if the surfel plane is not view-plane aligned. Thus we use quads, because vertex and pixel shading is better balanced and the total number of shader calls is smaller. Note, that with the unified shader architecture available on latest graphics cards it might be more efficient to use points instead as done in [HS07].

When rendering a surfel billboard, it has to be decided whether to render its front or its back. This can be done in the per-vertex stage using the sign of $(E-X)\cdot\mathbf{n}$, where X is the processed vertex of the quad, E the eye position in object-space and $\mathbf{n}$ the billboard normal. If only one texture had been assigned to the billboard it also should be determined if the right side of the billboard is visible. In another case it can be dropped (e.g. by moving it to infinity). In the per-pixel stage the correct texture can then be used for shading. In addition a more accurate z-buffer value can be computed using the z-offset stored for every texel of the normal map in its α-value (see 3.3).

4. Hybrid LOD

To preserve texture memory surfel billboards should be rendered for greater distances only. However, it is not sufficient in every case to render the original model at smaller distances, because often a lower LOD can be used instead. Therefore, it is useful to combine surfel billboards with a traditional triangle based multi-resolution hierarchy to efficiently handle smaller distances. As also described in [CAZ01], for this purpose we use the elegant multi-triangulation hierarchy [FMP97], abbreviated MT. In contrast to this approach, our MT contains larger patches in its arcs. This results in a better mesh-coding and a faster rendering using triangle-strips. To create such a MT we apply the algorithm proposed in [HS06], which is based on a reduction of the original MT. The classical MT or other patch-based MTs like proposed in [CGG*05] may be used as well.

In the following we briefly describe the MT data structure and how it is used to define various LOD. Then we will explain how it can be combined with a surfel billboard hierarchy. Afterwards, it will be shown how this can be reduced to significant billboards in order to save texture memory. Finally, a fast LOD estimation algorithm will be described.

4.1. Basic MT-Framework

A MT is a directed acyclic graph (short DAG). Each node contains a scalar attribute that measures the error caused by the local mesh simplification associated with the node. For this purpose, we use the object-space QEM [GH97]. The outgoing arcs of a node contain the geometry simplified by the node and the ingoing arcs represent this geometry after applying the simplification. In general a MT contains one source node (n_s) and one drain node (n_d) which outgoing arcs (resp. ingoing arcs) form the lowest (resp. highest) LOD of the model.

To define a specific LOD based on a given $MT = (\mathcal{N},\mathcal{A})$ an arc subset $\mathcal{C} \subseteq \mathcal{A}$ (called *cut*) has to be defined. A valid cut has two unique properties:

1. $\forall\, a = (n_i,n_j), a' = (n_k,n_l) \in \mathcal{C} :\ \nexists n_j \rightarrow^* n_k$ ($\rightarrow^*$ denotes a path of any length)
2. $\mathcal{C}$ is *maximal* (no arc can be added to $\mathcal{C}$ without breaking property 1).

The first property ensures that there are no overlapping arc patches. The second property ensures that the whole original mesh is covered by the arc patches of the cut. Using QEM the simplification error e_i stored in each node n_i increases monotonically bottom-up. In this case a valid cut can easily be defined by:

$$\mathcal{C} = \{(n_i,n_j) \in \mathcal{A} :\ e_i > \delta \geq e_j\}, \tag{10}$$

where δ is a quality threshold. To get a view dependant LOD s_{img} can be used to measure the node error in image space. This requires a current distance value for every node. Therefore, every node n also stores a bounding sphere (M,r) that contains the geometry that this node simplifies. Using this, a bound on the image space error e_{img} is given by

$$e_{img}(n) = s_{img}(e, z_{M_{eye}} - r). \tag{11}$$

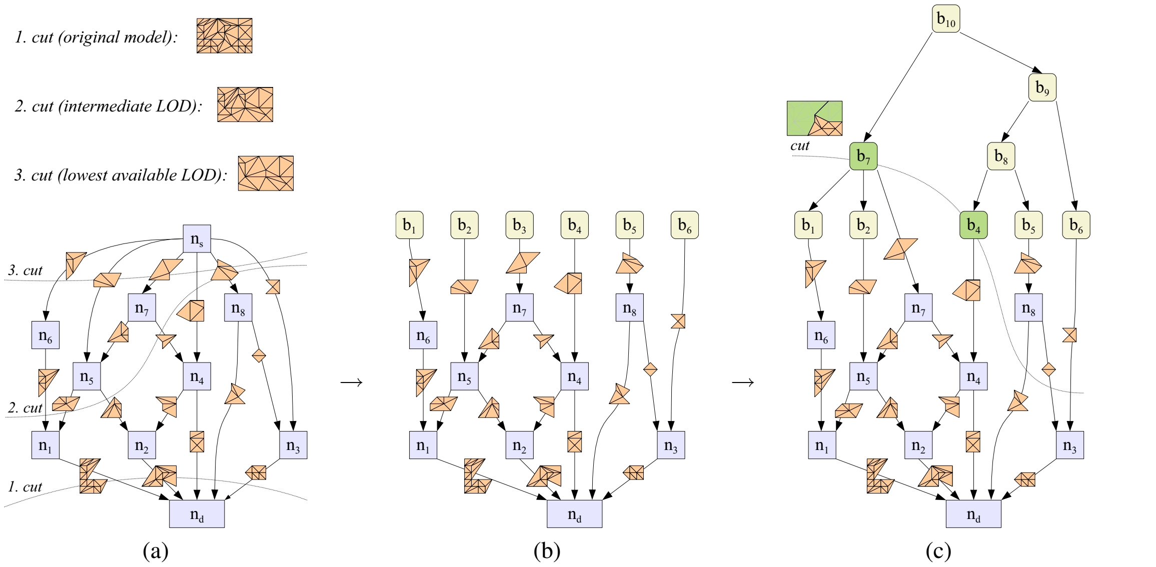


Figure 5: *Hybrid hierarchy creation: (a) MT with three cuts (i.e. three LOD). (b) Inserted base surfel billboards. (c) Final hybrid hierarchy with another cut containing triangles (marked) and surfel billboards (marked).*

To easily ensure that e_{img} increases bottom-up the bounding sphere of a node should also cover the bounding spheres of its child nodes. This all together yields

$$\mathcal{C} = \{(n_i, n_j) \in A : e_{img}(n_i) > \varepsilon \geq e_{img}(n_j)\}, \quad (12)$$

where ε is a pixel value, usually the same as used for the validity check of surfel billboards.

4.2. Combination of MT and Surfel Billboards

The simplification of triangular meshes is usually limited by many constraints. This does not guarantee that the lowest LOD that can be created by a MT only contains very few triangles. Therefore, the lowest LOD, defined by the outgoing arcs of the source node n_s, is used to create a surfel billboard hierarchy on top.

Each arc of the MT contains several triangles using as few triangle-strips as possible. As shown in [HS06] an average number of 10 triangles is a good trade-off between stripification efficiency and locality. This can be achieved by reducing the MT using the scheme also presented in [HS06]. In order to get a first image-based description of the lowest LOD, which can be obtained using a given MT (Fig. 5(a)), a surfel billboard is created for every outgoing arc of the source node (see section 3.1) and assigned as the new father of this arc (Fig. 5(b)). This does not guarantee to create flat billboards, but due to the small size of triangle-strips and the subsequent hierarchy reduction (explained in section 4.4) this is sufficient in the general case. After doing so, the isolated source node can be deleted.

After defining a set of base surfel billboards, they can be used to create a hierarchy, similar to point trees (Fig. 5(c)). This procedure is described in the following.

4.3. Surfel Billboard Hierarchy

A common technique to generate surfel or point hierarchies is to use an recursive object space subdivision scheme, like an octree. On the other hand the effectiveness of surfel billboards mostly depends on their depth value. Since space subdivision does not consider this, a more sophisticated procedure should be applied.

For this purpose, we use a simplification scheme based on the k-nearest neighbors. Since close surfels do not yield to flat billboards in any case as shown in Fig. 6 we decide to use a value of $k = 8$. This seems so be a good trade-off between accuracy and time consumption.

For the given set of surfel billboards $\mathcal{B}$ the subset $\mathcal{B}_i$ of nearest neighbor is computed for every surfel billboard $b_i \in \mathcal{B}$. Then for each $b_j \in \mathcal{B}_i$ the oriented bounding box OBB_{b_i,b_j} is calculated which includes the vertices of the triangle sets $\mathcal{T}_i$ and $\mathcal{T}_j$ replaced by b_i and b_j. Afterwards the triple (b_i, b_j, OBB_{ij}) is inserted into a priority queue in decreasing order sorted by a given weight w_{ij}. Its calculation is described later in this section. While processing the queue a pair (b_i, b_j) will be merged to a new surfel impostor b_{new} with child surfels b_i and b_j if both billboards are not already merged with another one. b_{new} is then resized according its OBB_{ij} and a proper texture size is determined as described in Sect. 3.3. The texture of the new surfel is rendered using the triangle sets $\mathcal{T}_i$ and $\mathcal{T}_j$. Finally, b_{new} is inserted with its unmerged neighbored billboards into the priority queue as

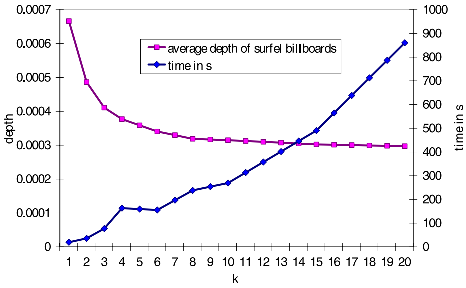


Figure 6: *Influence of k to the depth of created surfel billboards compared to required time for hierarchy creation exemplarily shown for the airplane model.*

described before. In doing so, a binary surfel imposter hierarchy can be generated.

To determine the specific weight w_{ij} for a particular merge operation we choose a metric which expresses the benefit of b_{new} created by the union of b_i and b_j. This includes two factors:

1. The plane size of b_{new} compared to its depth variance. This can be expressed by:

$$f_1 = 1 - \frac{d_{new}}{\sqrt{w_{new} h_{new}}}, \quad (13)$$

where a higher value is more beneficial.

2. The degree of coverage of the billboard plane by projected geometry. If this plane contains many transparent areas the contribution of b_{new} to a rendered image is small and vice versa. This can be approximated by counting the number of non-transparent pixel in a temporary framebuffer $\mathbf{FB}_{b_{new}}$ in which the triangle sets $\mathcal{T}_i$ and $\mathcal{T}_j$ are rendered:

$$f_2 = \frac{|\{\mathbf{p} \in \mathbf{FB}_{b_{new}} \mid \alpha_{\mathbf{p}} \neq 0\}|}{w_{\mathbf{FB}} h_{\mathbf{FB}}}. \quad (14)$$

Since both factors, f_1 and f_2, are in [0,1], we combine them by a fuzzy AND to form w_{ij}:

$$w_{ij} = f_1 f_2. \quad (15)$$

Because of nearest neighbor search, bounding box calculation and coverage estimation this algorithm has a complexity of $O(n^2)$, where n is the number of triangles in the lowest possible LOD afforded by the previously given MT. This is not sufficient for the general case and therefore it is necessary to reduce this effort. For this purpose, we propose a local estimation using a depth limitation. If a pair (b_i, b_j) should be inserted into the priority queue, we decide to change the OBB calculation in such a way that only child surfels l steps below b_i and b_j will be considered, as shown in Fig. 7 for $l = 1$. If this goes down into the MT-part of the hierarchy, then the triangle sets will be used as described before (Fig. 7(b)). In another case only the corner vertices of the children's billboard plane will be considered (Fig. 7(c,d)). In addition, these child-billboards will be rendered into the framebuffer for coverage estimation instead of the triangle sets. In doing so, OBB and coverage estimation complexity is constant and the complexity of the total algorithm is reduced to $O(n)$. Tests have shown that $l = 5$ is sufficient for a good result of OBB calculation and coverage estimation.

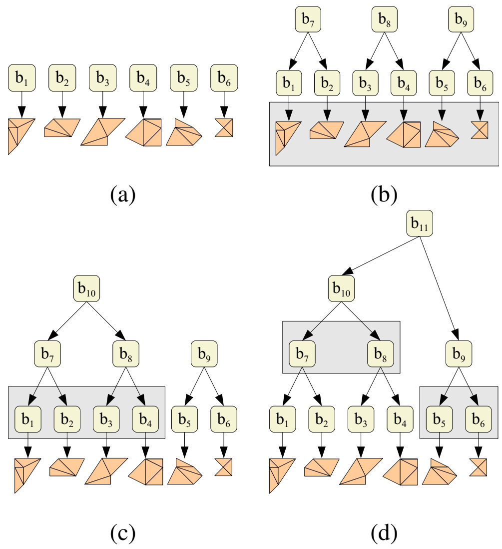


Figure 7: *Hierarchy creation using depth limitation $l = 1$: (a) Base set of surfel billboards. (b) Three new billboards are defined using the triangle sets at the bottom. (c) New billboard b_{10} is defined from billboards $\{b_1, \dots, b_4\}$ instead of triangle sets due to depth limitation. (d) Final hierarchy. (The influence area of new billboards is marked gray.)*

4.4. Billboard Hierarchy Reduction

Up to here, it is not guaranteed that all billboards inside the created billboard hierarchy are appropriate in the sense that they are valid for a long range. Thus, it is useful to filter nonrelevant billboards afterwards. A billboard b can be nonrelevant for two reasons:

1. The LOD represented by the surfel billboard is not coarser than the corresponding part of the lowest triangle-based LOD.
2. The range of validity is short due to the fact that its minimum distance is similar to the minimum distance of its father billboard or child billboards.

For the first case it is also possible to delete nodes of the MT part instead of deleting the surfel billboard. But since billboards are more expensive in the sense of texture memory we choose to delete surfel billboards.

To find billboards that are obsolete due to the first reason we compute a minimum distance z_{min} for each source node n of the MT part analogously to the z_{min} of a surfel billboard. This can be done by using $e_{img}(n) = \varepsilon$ which yields

$$z_{min}(n) = \frac{e_n}{\varepsilon} pc + r_n. \quad (16)$$

If a billboard b is connected to n by a MT arc and $z_{min}(b) \leq z_{min}(n)$ it can be erased (e.g. billboard b_3 in Fig. 5(c)). In order to find all of these billboards one bottom up breath-first scan of the billboard hierarchy is sufficient.

In a second scan we search for all billboards of a limited validity range. For this purpose, a similarity value ds is computed for each billboard b that measures the average ratio of the minimum distances of b's father p_b and children $\mathcal{C}_b$. Thus, ds is given by

$$ds(b) = \frac{z_{min}(p_b)}{|\mathcal{C}_b|} \sum_{c \in \mathcal{C}_b} \frac{1}{z_{min}(c)}. \quad (17)$$

If $ds(b)$ is smaller than a user-defined threshold ds_{max}, b can be deleted because its z_{min} is similar to that of its father and children. In doing so, for all billboards a quite similar value of ds can be achieved, which results in an even LOD transition. By adapting ds_{max} much texture memory can be saved. By testing we have found $ds_{max} = 1.5$ to be a good trade-off between texture memory and the number of surfel billboards selected for rendering, which means that a surfel billboard is in average valid if its distance is in $[z_{min}, 1.5z_{min})$.

4.5. LOD Estimation

To get a LOD of the original model for a specific viewing situation, the cut estimation defined in section 4.1 has to be extended to include billboards. Therefore, it has to be guaranteed that the simplification error monotonically grows bottom-up. Due to the hierarchy reduction step this is ensured.

Accordingly a cut can be defined by extending its definition for the MT part (equ. 12):

$$\begin{aligned} \mathcal{C} = \{(n_i \in \mathcal{N}, n_j \in \mathcal{N}) \in \mathcal{A} | e_{img}(n_i) > \varepsilon \geq e_{img}(n_j)\} \cup \\ \{(b \in \mathcal{B}, n_j \in \mathcal{N}) \in \mathcal{A} | !valid(b) \wedge \varepsilon \geq e_{img}(n_j)\} \cup \\ \{b \in \mathcal{B} | !valid(p_b) \wedge valid(p)\}, \end{aligned} \quad (18)$$

as shown in Fig. 5(c). Such a cut can easily be obtained by a top-down traversal of the hierarchy.

Common scenes often consist of one landscape object of a substantial depth range and many objects of a much smaller depth range. For the second type of objects we propose to use an iteration approach similar to sequential point trees [DVS03] to reduce the CPU load for the billboard selection of a particular LOD to $O(\log n)$. In doing so, all surfel billboards b are inserted in a vertex-buffer **VB** sorted by the z_{min} value of their father p_b in increasing order. For LOD estimation the smallest distance d_{min} of an object part is estimated by using the bounding sphere of the object. Since we know that a surfel billboard is only valid if its father is not, the greatest index i is determined afterwards for which $z_{min}(p_{\mathbf{VB}[i]}) \geq d_{min}$. Then the tail of the vertex buffer starting at i is processed by the graphics pipeline. In the per-vertex stage we compute the validity of the surfel billboard. For this purpose, all required information, including father information, are added to a billboard vertex as additional parameters. If a billboard is not valid it will be moved behind the back plane.

If the LOD only contains triangle-strips the whole surfel billboard vertex buffer will be processed, because the smallest z_{min} of a surfel billboard is greater than the closest distance of any object part. To prevent this, we also compute a greatest distance of the object using its bounding sphere again. If this greatest distance is less than the smallest z_{min} of a surfel billboard then vertex-processing of surfel billboards will be skipped. Vice versa processing of MT-nodes can be omitted if d_{min} is greater than any z_{min} of a source node.

5. Discussion and Results

On a standard PC equipped with a *Nvidia GeForce7800GS* we are able to render 60M triangles and 8M surfel-billboards per second. This difference appears, because surfel-quads can not be stored efficiently as a strip and because of the more complex shader programs that have to be used. But since the number of surfel-billboards is much lower than the number of triangles that have to be used for the same LOD this deficit is more than compensated. Due to this observation, we limit MT creation to simplification errors which yield to a minimum distance $z_{min} \leq 2s$ of the node that would be created (equ. 16), where s is the average bounding box dimension of the original object. For greater distances surfel billboards will be used instead. This distance is arbitrary and can be adapted to the available texture memory and object type. In doing so, it is possible to render hundreds of objects in real-time.

In table 1 you can see the number of surfel billboards and the required texture memory for some particular objects. For simple objects, like the height field, our approach performs very well. This is mainly due to hierarchy reduction which deletes many billboards. For very planar models, like the airplane model, image-based approaches work best. Our system impressively affirms this. But also for more complex objects like the tree model, our approach is well suited as can be seen in this table.

In Fig. 8 some LOD compositions are shown for various distances using the Stanford Bunny and the airplane model. As it can be seen, the number of triangles decreases very quickly to 0, whereas the number of billboards increases up to a specific distance and then rapidly falls again. This was expected, because the MT was only created for distances up

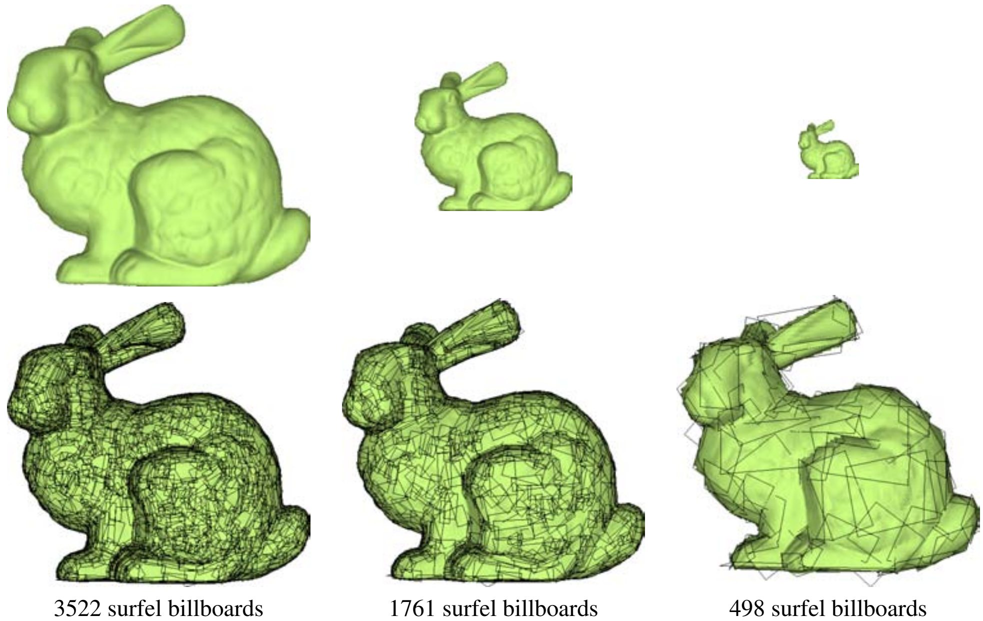


Figure 9: *Upper row: Different LOD of the Stanford Bunny based on object distance using* $\varepsilon = 1px$*. Lower row: Same LOD on smaller distances showing surfel planes.*

model	triangles	billboards	texture
airplane	22814	2168	≈8.6MB
bunny	104026	11017	≈20MB
armadillo	215914	17494	≈24MB
tree	50200	11785	≈25MB
height field	524288	5946	≈15MB

Table 1: *Object used in this paper with triangle numbers in lowest triangle-based LOD, number of surfel billboards in hierarchy and allocated texture memory.*

to 2. Indeed triangles are also used for greater distances (up to 5), but only because the corresponding surfel billboard is not valid yet. Moreover, it shows that our billboard selection scheme adapts the number of billboards to the distance similar to point trees. In Fig. 9 you can see some exemplarily LOD of the bunny model. To get a better feeling of the surfel billboard size the same LOD is shown below together with the quad boundaries of each surfel.

With our implemented framework many difficulties of other approaches are avoided. This includes:

- An automatic silhouette refinement, which is difficult to achieve by triangle-based simplification.
- In general mipmapping is not necessary, because surfel billboards are only valid for a small distance range.
- A sophisticated point-splatting scheme (multi-pass rendering and/or image postprocessing) is not necessary, because texture information suffice for shading.

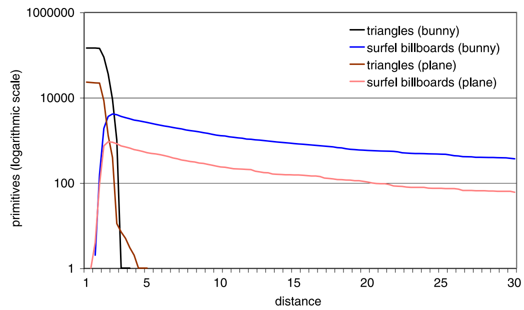


Figure 8: *Number of primitives used for various distances using the hybrid hierarchy of the bunny and the airplane model.*

On the other hand, our approach has also some disadvantages. These are:

- Its limitation to static objects. If the object shape changes over time, textures have to be recomputed. Moreover, the whole hierarchy has to be recreated. For joint-based dynamic objects it might be effective to use hierarchies for each object part, instead of one for the whole model.
- If the image resolution or the field of view angle change over time, textures and their sizes will have to be recomputed.

- If there is only little texture memory available, system memory will have to be used which reduces rendering performance.

6. Conclusion

In this paper we proposed a new way of rendering high-detailed meshes using so called surfel billboards. They combine the key-idea of unconnected surface elements (surfels) and image-based impostors. By arranging them in a hierarchy, you are able to form arbitrary view-dependant LOD. To reduce necessary texture memory we proposed a combination with triangle-based LOD. For this, a hybrid LOD hierarchy and its generation algorithm were described. In addition a reduction of this hierarchy to important surfel billboards and a simple but efficient LOD selection algorithm were defined.

Future work that our framework would benefit from should focus on integrating techniques to further increase rendering performance and memory efficiency. To achieve a higher rendering performance a fast backface culling and occlusion test should be added. To increase memory efficiency an out-of-core data management should be integrated which dynamically allocates and frees texture memory with regard to the actually used surfel billboards. Finally, the occlusion test could be advanced for dense populated scenes using displacement mapped billboard clouds [MJW07].

References

[BHP99] BAREQUET G., HAR-PELED S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. In *ACM-SIAM symposium on Discrete algorithms conf.proc.* (Philadelphia, PA, USA, 1999), Society for Industrial and Applied Mathematics, pp. 82–91.

[CAZ01] COHEN J. D., ALIAGA D. G., ZHANG W.: Hybrid simplification: combining multi-resolution polygon and point rendering. In *Vis'01 conf.proc.* (2001), IEEE Computer Society, pp. 37–44.

[CGG*05] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: Batched multi triangulation. In *Vis'05 conf.proc.* (2005), pp. 207–214.

[Che95] CHEN S. E.: Quicktime vr: an image-based approach to virtual environment navigation. In *SIGGRAPH'95 conf.proc.* (New York, NY, USA, 1995), ACM Press, pp. 29–38.

[CN01] CHEN B., NGUYEN M. X.: Pop: a hybrid point and polygon rendering system for large data. In *Vis'01 conf.proc.* (2001), IEEE Computer Society, pp. 45–52.

[DCSD02] DEUSSEN O., COLDITZ C., STAMMINGER M., DRETTAKIS G.: Interactive visualization of complex plant ecosystems. In *VIS '02 conf.proc.* (2002), pp. 219–226.

[DDSD03] DÉCORET X., DURAND F., SILLION F. X., DORSEY J.: Billboard clouds for extreme model simplification. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 689–696.

[DVS03] DACHSBACHER C., VOGELGSANG C., STAMMINGER M.: Sequential point trees. *ACM Trans. Graph. 22*, 3 (2003), 657–662.

[FMP97] FLORIANI L. D., MAGILLO P., PUPPO E.: Building and traversing a surface at variable resolution. In *Vis'97 conf.proc.* (1997), IEEE Computer Society Press, pp. 103–ff.

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH'97 conf.proc.* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 209–216.

[HB04] HILBERT K., BRUNNETT G.: A hybrid lod based rendering approach for dynamic scenes. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 274–277.

[HS06] HOLST M., SCHUMANN H.: Efficient rendering of high-detailed objects using a reduced multi-resolution hierarchy. In *GRAPP'06 conf.proc.* (Setúbal, Portugal, 2006), INSTICC Press, pp. 3–10.

[HS07] HOLST M., SCHUMANN H.: Normal mapping for surfel-based rendering. In *Journal of WSCG'07* (2007), pp. 219–226.

[JWP05] JESCHKE S., WIMMER M., PURGATHOFER W.: Image-based representations for accelerated rendering of complex scenes. In *EUROGRAPHICS 2005 State of the Art Reports* (Aug. 2005), Chrysanthou Y., Magnor M., (Eds.), EUROGRAPHICS, The Eurographics Association and The Image Synthesis Group, pp. 1–20.

[LW85] LEVOY M., WHITTED T.: *The Use of Points as a Display Primitive*. Tech. Rep. 85-022, Computer Science Department, University of North Carolina at Chapel Hill, 1985.

[MJW07] MANTLER S., JESCHKE S., WIMMER M.: *Displacement Mapped Billboard Clouds*. Tech. Rep. TR-186-2-07-01, Vienna University of Technology, 2007.

[O'R85] O'ROURKE J.: Finding minimal enclosing boxes. *International journal of computer and information sciences 14* (1985), 183–199.

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *VIS '02 conf.proc.* (Washington, DC, USA, 2002), pp. 163–170.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *SIGGRAPH'00 conf.proc.* (New York, NY, USA, 2000), pp. 335–342.

[RL00] RUSINKIEWICZ S., LEVOY M.: Qsplat: a multiresolution point rendering system for large meshes. In *SIGGRAPH'00 conf.proc.* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 343–352.

[Sch98] SCHAUFLER G.: Image-based object representation by layered impostors. In *VRST '98: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 1998), ACM Press, pp. 99–104.

[SLS*96] SHADE J., LISCHINSKI D., SALESIN D. H., DEROSE T., SNYDER J.: Hierarchical image caching for accelerated walkthroughs of complex environments. *Computer Graphics 30*, Annual Conference Series (1996), 75–82.

[ZPvBG02] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (2002), 223–238.