On Exact Error Bounds for View-Dependent Simplification

Elmar Eisemann, Xavier Décoret

Volume 0 (1981), Number 0 pp. 1–11

On Exact Error Bounds for View-Dependent Simplification

Elmar Eisemann and Xavier Décoret

ARTIS-GRAVIR/IMAG-INRIA,France†

Abstract

In this article we present an analytical closed-form expression to ensure exact error bounds for view-dependent simplification which is of importance for several algorithms. The present work contains proofs and solutions for the general 2D case and particular 3D cases.

Most preceeding works rely on coarse heuristics, that might fail and/or restrict movements or object representations. We introduce the notion of validity regions as the complete set of possible simplifications respecting a given error bound between the object and its simplification. The approach handles arbitrary polygonal viewcells which allow for free movement in the interior. We show how to compute these regions for mesh points and faces. Since the validity region of a face accounts for all its points, properties like silhouette preservation and textures are gracefully handled. This is not the case if the error is controlled only at the face's vertices or edges.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling

1. Introduction

Simplification has become a very important topic in today's research. Nowadays, models are made out of thousands of triangles, and complex scenes like forests can even contain millions of polygons. Graphics hardware is constantly improving, but not capable of displaying such complex scenes in real-time, thus simplification remains an important issue.

Clark [Cla76] was probably the first to point out that distant objects do not need the same precision as closer ones. Since then, several approaches have been proposed to automatically create simpler representations of an input model. A key point is to control the error caused by using a coarser replacement. This is a difficult problem for a single viewpoint but even harder if one considers to keep the same representation for a certain viewing region (*viewcell*).

In this article, we are interested in measuring the geometric error associated with a given simplification and viewcell. Furthermore it is possible to detect for each mesh part the viewpoints that reveal the maximal error. Reviewing the simplification from this very viewpoint gives the user an idea of the subjective/qualitative error.

A huge amount of literature on simplification exists, but no work on a general error analysis has been published so far. Interestingly, most previously proposed error bounds, even for the simpler case of a single viewpoint, are based on heuristics that are invalid in the general case. Viewcell approaches (where a simplification has been precalculated and is used while an observer stays inside a certain region, the viewcell) received a lot of interest because run-time simplification becomes inefficient for very complex models. The query-cost exceeds the gain from rendering the simplified version. In this article we define and solve the problem of *exact* (not only upper) error bounds analytically in two-dimensional scenes and with arbitrary precision in 3D. In practice models are often well-behaved, making heuristics work well, which produce arbitrarily large errors only in mostly pathological cases. This is probably why the resulting error has not yet been closely examined. Nevertheless our analysis can be applied and is of interest in situations where faithfulness is a must. Also we proof with our work that algorithms with strict error bounds still allow for aggressive simplification. We show this with a simple example application. This proof of concept is not the main contribution of the paper, which is the theoretical foundation and definition of the geometric error for viewcell-depedent simplification.

In this article we underline the actual complexity of the

† Elmar.Eisemann@inrialpes.fr, Xavier.Decoret@inrialpes.fr, ARTIS is a team of the GRAVIR/IMAG laboratory, a joint effort of CNRS, INRIA, INPG and UJF

problem, present how error bounds can be visualized, calculated and assured. It serves as an invitation to further explore a field that existed for more than 30 years. Simplification still contains lots of basic open questions to which our work gives some answers and provides a deep understanding.

After discussing previous work (section 2), the definition of the problem will be presented (section 3). The calculation of exact validity regions is not trivial and we will derive it in several steps. Starting with mesh points we show how we can indirectly obtain the exact solution using special viewpoints. We describe how to find them for different types of viewcells (section 4) and extend the approach from mesh points to faces (section 5). We then discuss our work, conclude and give an outlook on future research (sections 6,7).

2. Previous Work

General simplification algorithms can be divided into two classes. The first selects an appropriate model during run-time based on the current viewpoint. This often involves hierarchical structures [Hop96,Hop97,XV96,ESV99, ESSS01]. The second class usually implies the use of alternative representations [SDB97, DSSD99, JWS02, JW02, WM03] which are created according to a viewcell. Seen from this region, the corresponding simplified representation should closely match the original model's appearance.

Run-time based approaches test simple queries to decide whether more details are needed. Most algorithms use edge collapses (introduced in [Hop96]) and work with bounding spheres and special silhouette criteria to obtain a better quality. Unfortunately, bounding spheres around edges do not provide a real conservative bound, neither closeness relations between vertices as used by Luebke et al. [LE97]. In [LE97] some ground rules are specified. Polygon soups should be accepted as valid input, and a fine-grained tradeoff between fidelity and aggressive simplification should be possible. The analysis in this article fulfills both conditions.

Another run-time approach, that represents a transition to offline simplification is presented in [COM98]. This algorithm establishes upper bounds on geometry and texture and is able to limit the actual screen error. Nevertheless the bound is not tight and only given in accordance to a certain viewing distance, not for arbitrary viewcells. The method is best suited for dense objects of high complexity.

Most viewcell-based approaches involve alternative representations which can be rendered efficiently and only one query (membership of the observer to a viewcell) has to be evaluated. Typically much information about the scene can be stored in form of textures on a simplified model [MS95, SDB97]. As pointed out by Wilson et al. [WM03], invisible parts may cause problems. They perform a reconstruction based on several viewpoints to maximize visibility. In some cases this might not be sufficient. Imagine a wall almost perpendicular to the observer's sightline. No geometry is hidden, but texture information can appear distorted from a different viewpoint due to the grazing angle. Geometrical error bounds are not established on such representations.

Jeschke et al. [JWS02, JW02] manage to provide an error bound for a particular case. Around a cubic viewcell, they create enclosing cubes on which the scene is projected, whereas inside the first one the original geometry is used. If the user moves on the diagonals of the viewcell, they are able to deduce the position of the layers for a faithful representation. Décoret et al. [DSSD99] provide an error measure for parallax effects in two-dimensional scenes for segment viewcells. We derive a more general error bound which does not directly depend on the alternative representation being used. We ensure fidelity and do not restrict the observer's movement to a segment.

Our approach relates to Cohen et al. [CVM*96], where a hull is defined around the initial model. Simplification operations are only considered valid if the object remains inside this boundary. Cohen et al.'s approach is not view-dependent and does not take texture into account, as it uses the Hausdorff distance (see section 3). Inspired by their work we are interested in determining maximum envelopes for a view-dependent, point-wise measure.

3. Basic Definitions

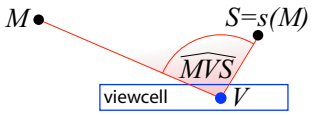

Figure 1: *Angular distance is evaluated for each viewpoint V in viewcell $\mathcal{V}$.*

The input of the problem is a mesh $\mathcal{M}$ and a viewcell $\mathcal{V}$. The *mesh* is defined by vertices V_i and faces $\mathcal{F}_j$, and consists of all the *points* inside those faces (usually denoted M). The *viewcell* $\mathcal{V}$ is a set of *viewpoints* (usually denoted V).

The goal is to measure the distance between $\mathcal{M}$ and a simplified mesh $\mathcal{S}$. A *simplification* is a mapping from the points of $\mathcal{M}$ to the points of $\mathcal{S}$:

$$s : \mathcal{M} \mapsto \mathcal{S}, M \rightarrow s(M)$$

Note that s can be many-to-one, i.e. several points can be "simplified" to the same place.

A classic approach to measure the error is to compare $\mathcal{M}$ and $\mathcal{S} = s(\mathcal{M})$ using the Hausdorff distance. This metric measures the geometric change between the mesh and its simplification. If every point on the mesh has a color (e.g. through texturing), two meshes can have the same shape (i.e. a null Hausdorff distance) but look very different. For that reason, distance measures between $\mathcal{M}$ and $\mathcal{S}$ should be considered for each point (not only vertex) on the mesh. Thus, in a first step, we will focus on single points on the mesh.

For a single viewpoint V, the *distance* between M and $S :=$

$s(M)$ is defined as the angle $\widehat{MVS}$ under which the segment $[M,S]$ is seen from V (fig. 1). It is important to notice, that for a fixed view frustum, the angle implies a bound on the projected screen distance and vice versa.

For a viewcell $\mathcal{V}$, we define the error as the maximum angle under which a point M and its simplification S can be seen from within the viewcell.

$$e_{\mathcal{V}}(M,S) := \max_{V \in \mathcal{V}} \widehat{MVS} \tag{1}$$

We say a simplification is *valid* for a given error $\Theta \in [0,\pi/2)$, if for all mesh points $M \in \mathcal{M}$ the error between M and its simplification $s(M)$ is smaller than Θ.

4. Validity Regions of Points

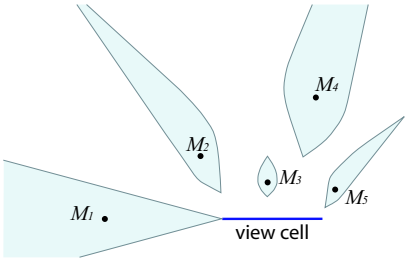

Figure 2: *Some examples for validity regions of mesh points based on a given segment viewcell and a fixed error bound. Notice that the shapes are not symmetric, nor always polyhedral and might be unbounded. The image was obtained via sampled cone intersections.)*

The *validity region* for an error bound Θ of a mesh point M and a viewcell $\mathcal{V}$ is defined as the set:

$$VR^{\Theta}_{\mathcal{V}}(M) := \{S \mid e_{\mathcal{V}}(M,S) \leq \Theta\} \tag{2}$$

This definition is equivalent to:

$$VR^{\Theta}_{\mathcal{V}}(M) = \bigcap_{V \in \mathcal{V}} \{S \mid \widehat{MVS} \leq \Theta\} \tag{3}$$

From this we can obtain:

$$VR^{\Theta}_{\mathcal{V} \cup \mathcal{W}}(M) = VR^{\Theta}_{\mathcal{V}}(M) \cap VR^{\Theta}_{\mathcal{W}}(M) \tag{4}$$

Intuitively eq. (4) states that a bigger viewcell leads to smaller validity regions. In eq. (3), the set $\{S|\widehat{MVS} \leq \Theta\}$ is a cone of apex V in direction $\overrightarrow{VM}$ and aperture Θ. Therefore the validity region corresponds to the intersection of cones.

The remainder of this section describes the determination of the exact shape of the validity region. Fig. 2 shows some examples and one sees that the shape is not polyhedral, thus depending on an infinity of viewpoints.

From eq. (3) we see that validity regions are convex, because the set is defined as the intersection of convex cones. Convexity implies star-shape which means that the validity region of M can be represented radially by its *extents* with respect to M and a given direction $\vec{d}$. This gives a convenient parametrization of the set we look for.

$$VR^{\Theta}_{\mathcal{V}}(\vec{d})(M) := \delta VR^{\Theta}_{\mathcal{V}}(M) \cap r(M,\vec{d}) \tag{5}$$

where $r(M,\vec{d})$ denotes the ray from M in direction $\vec{d}$ and $\delta VR^{\Theta}_{\mathcal{V}}(M)$ refers to the boundary of the validity region.

For a single viewpoint V it is simple to calculate the extents. It corresponds to the intersection between the view cone borders and a ray through M in a direction $\vec{d}$ (fig. 3).

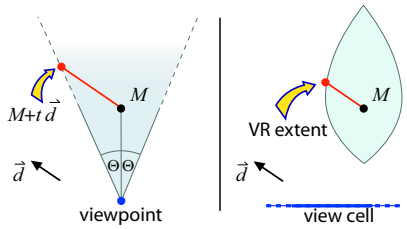

Figure 3: *Validity region extents of points, intersection with cone (left), hyperplane viewcell (right)*

For an error bound Θ the validity region extent in direction $\vec{d}$ of M for a viewpoint V is given by $M + t\vec{d}$, where t is specified by the (smaller) positive result of eq. (6). If there is none, the validity region is unbounded in this direction.

$$\frac{-\sin^2\Theta < \overrightarrow{VM}|\vec{d} > \pm \sin\Theta\cos\Theta\sqrt{1 - < \overrightarrow{VM}|\vec{d} >^2}}{< \frac{\overrightarrow{VM}}{||\overrightarrow{VM}||}|\vec{d} >^2 - \cos^2\Theta} \tag{6}$$

$< | >$ denotes the standard scalar product, $||\ ||$ the L_2 norm.

For general viewcells, it is difficult to calculate the validity region. On the other hand, for a finite set of viewpoints the solution can be found by evaluating expression (6) several times. The key idea is thus to find a finite set of viewpoints in the viewcell that can be used to bound the whole validity region extent.

Max Viewpoints: It can be shown that for a closed viewcell there has to be a special viewpoint, the so-called *max viewpoint*, V_{max}, such that $VR^{\Theta}_{\mathcal{V}}(\vec{d})(M) = VR^{\Theta}_{\{V_{max}\}}(\vec{d})(M)$. V_{max} depends on the mesh point, the viewcell, the error bound and the direction.

In other words, the cone associated with the viewpoint V_{max} is actually responsible for the validity region extent in the direction $\vec{d}$. Denoting the validity region extent E, it follows $\widehat{EV_{max}M} = \Theta$ (fig. 4). The max viewpoint thus gives the validity region extent (expression (6)). This gives us an an indirect way to find the exact answer to our problem. The existence of these points will follow from the discussion in

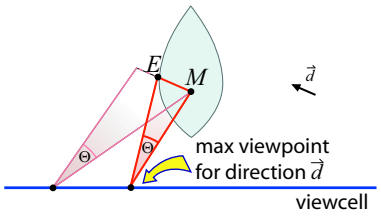

Figure 4: *For a closed viewcell and a direction $\vec{d}$, there is one specific viewpoint, the max viewpoint, which implies the validity region extent E of mesh point M.*

the special case of polygonal viewcells but it is generally true for closed viewcells.

The inscribed angle theorem states that in 2D all points that see a given segment under a fixed angle Θ lie on a bicircle, the outer contour of two intersecting circles. In 3D, the set is invariant under rotation around axis $\vec{d}$, thus leading to a so-called bialy, a torus without hole. This shape also played an important role in [LKR*96], where it was derived in a different manner for height field simplifications during run-time. All points inside this bialy see the segment under an angle greater than Θ. This shows that the bialy defined by the mesh point, the validity region extent and the max viewpoint is tangent to the viewcell and leads to the observation:

$$VR^{\Theta}_{\mathcal{V}}(M) = VR^{\Theta}_{\delta\mathcal{V}}(M)$$

Hence, it is sufficient to calculate the validity region extent for the borders of a volumetric viewcell.

Due to eq. (4), each part of a polygonal viewcell can be treated separately by intersecting the corresponding results. In 3D we can restrict ourselves to faces, in 2D to simple line segments. It thus suffices to suppose one codimensional viewcells. We start by considering hyperplanes.

Hyperplane Viewcells: The bialy's tangency is key to finding the max viewpoint. Due to the bialy's rotationally invariant shape around direction $\vec{d}$ the tangency point has to lie on the orthogonal projection of the line $S(t)$ on the plane. This implies that for hyperplane viewcells the 3D case can be solved in 2D. Fig. 5 shows a bialy and the possible restriction to 2D. The parametrization of a simplification point $S(t) = M + t\vec{d}$ leads to a parameterized bialy of points that "see" the segment $[MS(t)]$ under the angle Θ. The idea is to find the parameter t to obtain tangency (fig. 6). Instead of working on the bialy/bicircle itself, we treat the two circular parts separately.

Without loss of generality, the viewcell corresponds to the x-axis and the mesh point $M = (m_1, m_2)^\top$, with $m_2 > 0$. Then the circle is given by the equation for the center $C(t)$ and the radius $r(t)$.

$$C(t) = M + \frac{t}{2}(\vec{d} \pm \cot\Theta \vec{d}^{\perp})$$

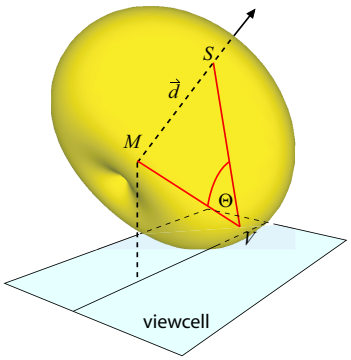

Figure 5: *The 3D shape of the points seeing the segment $[M,S]$ under the same angle. The max viewpoint has to be situated in the plane containing the projection of the line defined by M, $\vec{d}$ and the viewcell plane's normal.*

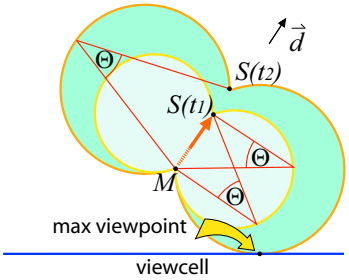

Figure 6: *Parameterizing a simplification point leads to a parameterized bicircle (consisting of two circles). The max viewpoint is the tangent point on the viewcell.*

$$r(t) = \frac{t}{2\sin\Theta}$$

where ± indicates the considered part of the bicircle. As the viewcell corresponds to the x-axis and M is situated in the upper half-space, the tangent/max viewpoint must have coordinates $(x_{max}, 0) = C(t_{max}) - (0, r(t_{max}))$. Solving the implied linear system leads to the coordinates of the max viewpoint:

$$V_{max} = (m_1 + m_2 * \frac{\sin\Theta d_1 \pm \cos\Theta d_2}{\pm\cos\Theta d_1 - \sin\Theta d_2 + 1}, 0) \quad (7)$$

$$t_{max} = \frac{2\sin\Theta m_2}{\pm\cos\Theta d_1 - sin\Theta d_2 + 1} \quad (8)$$

again, ± depends on the considered part. More details can be found in the appendix A.

At this point, we get a first interesting result. If we parameterize t, which corresponds to the distance between the validity region extent and the mesh point M, via the direction $\vec{d} = (\cos\gamma, \sin\gamma)^\top$, given by an angle γ, we get:

$$t_{max}(\gamma) = \frac{\sin\Theta m_2}{\pm\cos(\Theta \pm \gamma) - 1} \quad (9)$$

which is an equation describing a hyperbola. The validity region of a mesh point for a line viewcell is thus formed by two intersecting hyperbolas (fig. 7). In 3D this region is rotated around the normal of the hyperplane viewcell. We have thus an exact description of the validity region of a point in three dimensions in the case of a hyperplane viewcell.

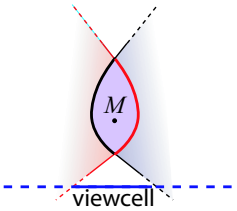

Figure 7: *For a line viewcell the validity region of a point is bounded by two hyperbolas.*

One Codimensional Viewcells: In the next step we want to restrict the hyperplane viewcells to polygonal areas. We will describe how to solve the problem in the 2D case and explain how to transfer the idea to the 3D case.

When working with two separated circles instead of the bicircle, we actually find two "max viewpoints", one for each circular part. Although one of these two might not be a max viewpoint in the strict sense, to avoid confusion and with a slight abuse of notation, we will refer to both of them as max viewpoints. It is not problematic to test several points, as long as their number is finite. In particular it is important to consider both sides of the bicircle, because the position of a max viewpoint for a segment viewcell is in no direct relation to the one for the corresponding line viewcell (fig. 8).

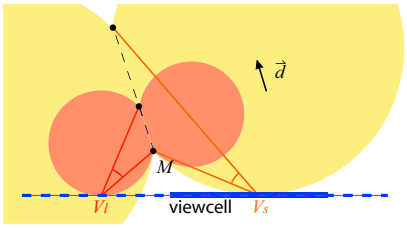

Figure 8: *For the line viewcell the max viewpoint in direction $\vec{d}$ is situated on the left, whereas for the segment viewcell it is on the right.*

Let $\mathcal{V}_S$ be a segment viewcell and $\mathcal{V}_L$ its extension to a line viewcell. Also, let V_s (resp. V_l) be the max viewpoint for $\mathcal{V}_S$ (resp. $\mathcal{V}_L$). If $V_l \in \mathcal{V}_S$ then $V_s = V_l$. We now want to show that if $V_l \notin \mathcal{V}_S$ then V_s is an extremity of $\mathcal{V}_S$ (fig. 9). V_l has to be in the interior of the tangent bicircle at $\mathcal{V}_S$ (inscribed circle theorem). Therefore the segment $[V_s, V_l]$ lies also in the interior of the bicircle of V_s. $\mathcal{V}_S \cap (V_s, V_l) \neq \emptyset$ contradicts the tangency property, thus V_S is an extremity.

In the 3D case, we can establish the same distinction. The proof is essentially the same, the bialy for the hyperplane is

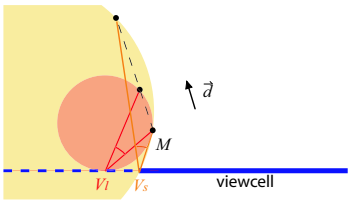

Figure 9: *If the max viewpoint (for one part of the bicircle) for the line viewcell does not lie on the segment viewcell, its max viewpoint has to be on an extremity.*

contained in the bialy for the viewcell, therefore the space between the two has to be empty. Unfortunately in 3D, this border is not a point, but a segment. To solve for the validity extent it can still be treated as a line and then the segment's extremities are taken into account. Still, finding max viewpoints for the line involves higher order polynomials and we did not yet succeed to obtain an analytic expression. Nevertheless, it is possible to approach the result arbitrarily close using numerical methods. It is *not* a trivial result that numerically stable solutions can be provided. The proof comes from the geometrical interpretation that we established. The idea is to embed the line into a plane and look at iso-values for the sizes of the bialys (which are in direct relation to the lengths of the validity region extents). The result are half-moon shaped areas around the actual max viewpoint of the plane (fig 10). Thus, on a line embedded in this plane at most three extrema can occur. All viewpoints for which the validity region is unbounded lie inside a cone. Therefore one can quickly determine which part of the viewcell is not needed for the determination of the validity extent. To restrict the search space one can pick an arbitrary viewpoint V of the remaining viewcell and define a minimum distance such that all viewpoints exceeding this distance are less restrictive than V. These properties make it possible to compute the validity extent in a stable way with arbitrary precision.

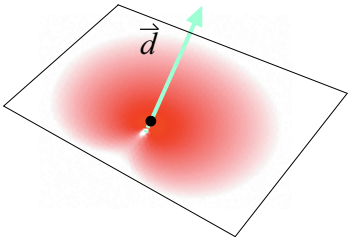

Figure 10: *For a given direction the sizes of the bialys are represented on a plane (red low, white high). The halfmoon-shapes can be explained with our geometrical interpretation.*

To resume, so far we have shown that it is possible to calculate the validity region of points given a finite set of (not necessarily bounded) polygonal viewcells. For a given point M and a direction $\vec{d}$, possibly given via an existing simplification point S, the max viewpoints are classified for each

face of the viewcell. This is done by considering the hyperplane using eq. (7) and discarding viewpoints outside the viewcell face. Next the borders of the viewcell are examined. This gives a finite set for which eq. (3) is evaluated, what leads to the minimal extent.

This result could be applied for e.g. simplification of point clouds or sampled geometry (or texels of a textured mesh). Such applications lie out of the scope of this article. The result is general enough to be used with any simplification algorithm. If the creation of the impostor was not performed using this error bound, the verification of validity still remains possible. Concerning level of detail changes, when the observer approaches the object, our error measure delivers the actual distance at which a representation change is needed. This can be done by determining the bialys for the desired error and assuring that the minimal distance keeps the viewer outside of every single one.

5. Validity Regions of Faces

In this section we will extend the notion of validity regions of points to mesh faces, therefore taking color information of points (like texture) into account. Mathematically a face contains an infinity of points. Thus to avoid sampling, we want to establish a way to assure the validity of the modification of a whole face. This part is quite technical and currently our result is two-dimensional.

The solution is not straightforward because validity regions for points inside the face cannot be simply interpolated from the validity regions of the vertices. In other words, if the vertices of a face are moved within their validity region, mesh points on the face do not necessarily remain in their respective validity region, as illustrated in fig. 11. This directly implies that algorithms basing their error estimate only on edges or vertex distances cannot succeed in establishing an upper bound on the error of the actual image (neither in 2D nor in 3D). A comparison of any algorithm like this with a method involving our analysis could thus be arbitrarily biased in our favor. In other words heuristic can and will lead to arbitrarily large errors (at least in special cases).

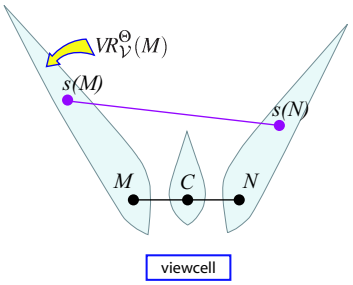

Figure 11: *When M and N are moved within their validity region (to s(M) and s(N) respectively) not all points on [MN] can be simplified on [s(M),s(N)]. E.g. C's validity region does not intersect [s(M)s(N)]*

To define a validity region for faces, we must first fix the way a face is "moved" during the simplification process, thus specifying the "movement" of the corresponding mesh points. We chose a directional projection onto a simplification plane. This is actually less restricting than it might sound and has several benefits. The order of the mesh points and a certain injectivity and connectivity is kept during simplification. Compared to a perspective projection, no specific viewpoint has to be selected and faces could still be cut into parts to allow different directions per patch. Finally it should be mentioned that the analysis is independent of the way the final representation is stored. If the obtained simplification is texture based, it can still be transformed under perspective projection to gain memory. Only the resolution changes, not the position of the samples. The approach of this section can also be used to find optimal texture resolutions for all viewpoints inside the viewcell (see [Eis04]). The idea is to choose as the simplification direction the surface orientation, thus representing texels on the surface. In this part, we assume projections along the plane's normal to ease explanations, although the approach described generalizes to arbitrary directions.

A simplification plane is said to be *valid for a face* if each point on the face, when projected onto the simplification plane, remains inside its corresponding validity region (fig. 12).

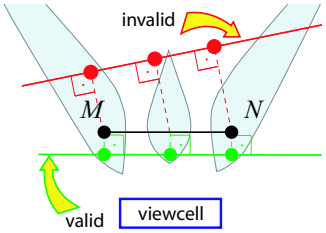

Figure 12: *A simplification plane is valid for the face $[M,N]$ if **all** face points remain inside their corresponding validity region after being transferred.*

A plane can be represented by its Hough transform: a normal and an offset from the origin [Hou62]. Fixing a normal (and thus projection direction) leaves us with the question of finding the possible offsets for a valid simplification plane. For a single mesh point the offset is given via the validity region extents. For a mesh face, this is more difficult.

From eq. (8), we observe that for a line viewcell the extents of a validity region vary linearly with the distance of the mesh point to the viewcell. In this case it is sufficient to ensure validity at the face's extremities to obtain validity for all points on the face. This also holds in the 3D case for hyperplane viewcells. Thus if all the max viewpoints fall inside the segment viewcell, we encounter a linear behavior and it is actually sufficient to test the vertices of the face.

This observation leads to the idea of decomposing the

mesh face into several parts, for which we will determine the offsets separately. Having detected the linear part, we will see that the remaining parts are non-linear, but involve only one single viewpoint. Once offsets have been found for each part, an intersection leads to the correct solution for the face. An empty set means that there is no valid simplification plane for this projection direction, hence at least one point cannot be simplified. Realize that for each face there are always valid simplification planes, in particular the one containing the face itself.

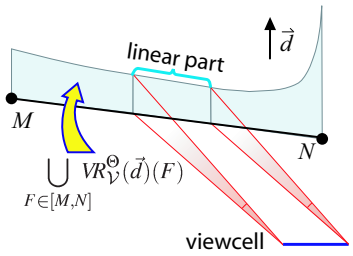

Figure 13: *Validity extents in direction $\vec{d}$ of the face's points form the shape above, which contains a linear part. Accordingly the face will be decomposed into a linear region, where it is sufficient to test the extremities and the non-linear regions, which have to be dealt with separately.*

Detecting the Linear Part: The determination of the linear part is actually equivalent to the detection of those points on the face for which the max viewpoint of the corresponding line viewcell falls into the segment viewcell. Due to eq. (7), we can determine for each point on the face the corresponding coordinates of the max viewpoint for a line viewcell. Actually there is a small subtlety; eq. (7) corresponds to a mesh point in the upper half-space, therefore faces should be clipped, see appendix B. There is a linear correspondence between the position of the mesh point and the max viewpoint for hyperplane viewcells. Now, if the face is extended to a line, we have two points on this line for which the max viewpoint corresponds to an extremity of the segment viewcell. These borders of the linear part can be inferred by solving for $(m_1, m_2)^\top$ in eq. (7) where the max viewpoint corresponds to the segment viewcell's extremities (fig. 13).

Supposing the face is given by the segment $[M,N]$ we will refer to the line through the face, as the *face line*, here given by $l(\alpha) := M + \alpha(\vec{MN}) = M + \alpha(w_1, w_2)^\top$. Without loss of generality, one viewcell extremity is $(e,0)^\top$. Thus, the equation to solve is: $V_{max}(l(\alpha)) = (e,0)^\top$. The solution is given by:

$$\alpha_e = \frac{e-(m_1+cm_2)}{w_1+cw_2},\ c := \frac{\sin\Theta d_1 \pm \cos\Theta d_2}{\pm\cos\Theta d_1 - \sin\Theta d_2 + 1} \quad (10)$$

($\pm$ corresponds to the two sides of the bicircle). If $(w_1 + cw_2) = 0$ all mesh points share the same max viewpoint. If this viewpoint lies inside (respectively outside) the segment, the whole face is linear (respectively non-linear).

Dealing with the non-linear part: The way we detected the linear part actually implies one property of the non-linear part; it only depends on a single viewpoint: one extremity of the viewcell. Thus we only need to examine how the validity region of a face behaves for a single view*point*.

We will consider the validity region extent of points on a face line. For each such point P, the validity region extent is given by the intersection of a line passing through P in the projection direction and the viewpoint's view cone. If each side of the cone is treated separately as a line and we plot those intersections for every point on the face line, we get a graph as shown in fig. 14. It may contain "false" intersections which represent intersections with the line instead of the ray. This is unproblematic, as a false intersection implies a more restrictive intersection for the other view cone ray. The right part of fig. 20 shows an example.

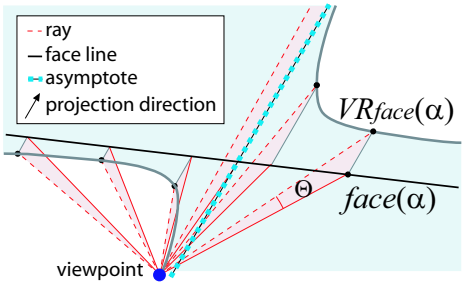

Figure 14: *The validity region extents of the mesh points on the face are given by the intersection between a ray from viewpoint with angle Θ and a line from the mesh point in projection direction.*

The graph visualizes the validity region extents of all points on the face line. A valid simplification plane has to remain inside the hull described by the portion of the graph corresponding to the mesh face (blue region in fig. 14). Thus to find the offsets of our plane, we need to find tangents with normal $\vec{d}$ at the contained curve parts.

The following reasonings will have to be performed for Θ and $-\Theta$ (both branches of the view cone). The resulting graphs delimit the simplification plane's offset. Both cases are similar and we will only focus on Θ.

Let's develop a mathematical description. Given a point M, the viewpoint V and the projection direction $\vec{d}$, we are looking for the intersection between the two lines $l_1(\alpha) := M + \alpha\vec{d}$ and $l_2(\alpha) := V + \alpha R_\Theta(V - M)$, where R_Θ describes a rotation of angle Θ.

To ease the calculus, but without loss of generality, we can assume that $V = (0,0)^\top$, $\vec{d} = (0,1)^\top$. M is on the face line given by $face(\alpha) = (0,m)^\top + \alpha(\cos\gamma, \sin\gamma)^\top$. This last assumption is actually a restriction, as it is now impossible that the face describes a vertical segment, not lying on the y-axis. On the other hand, this is not crucial as the projection direction was assumed to be vertical. We simply exclude this

case where a face would be simplified to a point. This makes sense and the remaining case could be treated separately. The final curve corresponds to:

$$VR_{face}(\alpha) := \alpha \frac{\alpha \sin(\Theta+\gamma) + m \cos\Theta \cos\gamma}{\alpha \cos(\Theta+\gamma) - m \sin\Theta \cos\gamma} \qquad (11)$$

We are now interested in the tangent points with a normal equal to the projection direction. As the projection direction has been chosen to be $(0,1)^\top$, we are actually interested in local minima and maxima of this quadratic rational. The curvature can only have two different signs, one for each side of the definition gap. Thus any point for which the derivative vanishes is automatically an extremum, there cannot be any saddle points and the function remains monotonic for the branches separated by the extremum. Due to this monotonicity, it is sufficient to test the extremities of the face if the extrema do not correspond to mesh points.

Finding extrema is equivalent to finding roots of the derivative. The resulting expression is at most quadratic and therefore possible to solve. The discussion is quite technical, because the function varies from a line, to a parabola, hyperbola and a "real" rational function depending on the angle between the face and the projection direction. The resulting formulae can be found in appendix C.

Concluding Face Validity: To summarize, let's explain a way to detect the validity of a face. (To accelerate the approach a more profound discussion of the functions would be necessary and the interested reader is referred to [Eis04]).

For both parts of the bicircle, that is to say for both circles we find the linear region on the face. Then we calculate the maximum offset in the projection direction for the linear part by evaluating eq. (6) for its extremities. For the remaining non-linear parts there is only one viewcell extremity to be considered. The tangents at the curve are found for Θ and $-\Theta$. If the extrema correspond to mesh points these are evaluated, otherwise only the face's extremities need to be checked. The process is repeated for $-\vec{d}$, to get the lower offset bound for the simplification plane.

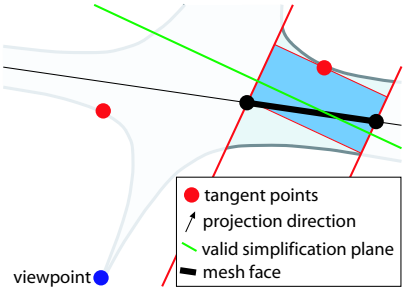

Figure 15: *For Θ and $-\Theta$ we obtain functions representing the validity extents of all points on the face line. We are actually only interested in those points on the face line corresponding to mesh points. Therefore both functions are cropped. There are two cases: an extremum corresponds to a mesh point (upper curve) or only the extremities have to be tested because of monotonicity (lower curve).*

6. Discussion

In this paper we examine the exact error in the case of viewcell-dependent simplification and describe how they can be visualized and represented geometrically. We treat several 3D cases and provide closed-form solutions for the 2D situation.

The innovative contribution is the notion of validity regions, which correspond *exactly* to the region of points which are close enough to be used as a simplification. We explain how their exact calculation can be achieved. Thus, we were able to establish an *exact* error bound. In other words, we give an answer to the question whether a simplification is valid, how much we can simplify and for which viewpoint the error will be most evident.

In addition, we point out the importance of considering all points in a face to respect texture. The error measure naturally leads to silhouette and parallax effect preservation. This approach combines local point with mesh information and avoids sampling. Movement of the observer is not restricted and arbitrary polygonal viewcells are possible. As mentioned before heuristics that only test observed vertex distances or edge lengths can result in arbitrarily large errors. Figure 16 depicts such a situation where the error converges to infinity. It is impossible to perform an unbiased comparison.

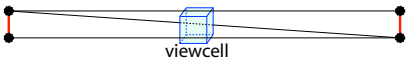

Figure 16: *The vertical edges are small as seen from the viewcell, but their collapse makes the triangles disappear.*

The evaluation of the validity regions is very fast due to its closed-form representation. It takes $\approx$ 1 sec. for 290.000 point validity extents ($\approx$ 0.0036 ms per evaluation) on a Pentium 1.5 GHz (face validity is usually about 3-4 times slower). Nevertheless we completely acknowledge that good heuristics can be simpler and often lead to acceptable results (especially for finely tesselated models). This is not surprising, as for far away geometry a small viewcell behaves like a point viewcell and small triangles behave like points, but it is *not* equivalent. This paper allows to evaluate the error resulting from simple heuristics and compare to the now available exact reference. E.g. for impostors [SDB97] often a single viewpoint is chosen. This necessity becomes apparent in form of a common problem of viewcell-based approaches. Representations can change drastically from one viewcell to the other. This is especially true because little attention has been paid to the actual error committed by the replacement. In our approach the proximity is guaranteed throughout the

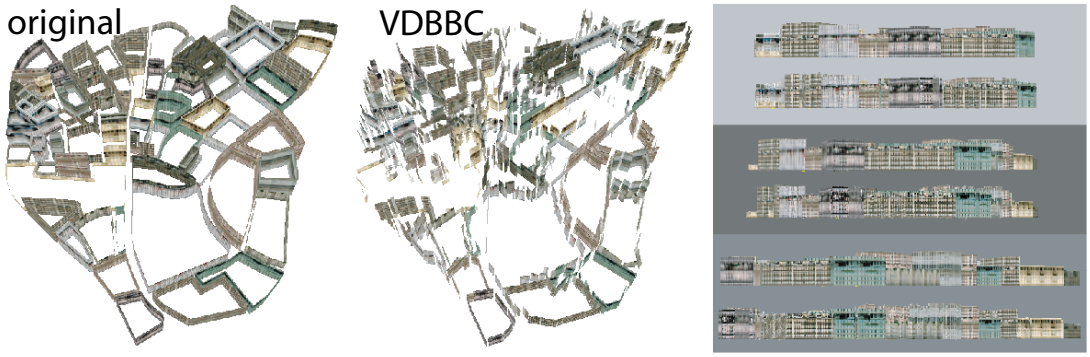

Figure 17: *A city model (Courtesy of J. Dorsey) is simplified on only 115 quads. The viewcell is a halfspace at the bottom of the image. Even though from above both representations seem to differ a lot (left, middle), seen from the inside of the viewcell both look quite similar (right). In each of the three sample views the original is on the top and the simplification underneath.*

viewcell and everywhere on the geometry. Overlapping of viewcells or blending thus become meaningful.

Theoretically this metric could be useful in several contexts, like simplification envelopes [CVM*96]. Geometry modifications are accepted if the shape is close enough to the initial one. This approach would be applicable for any edge-collapse or vertex deletion algorithm if a simplification function can be defined. This is also possible for point clouds or by sampling in 3D (for example at texture resolution). This is somewhat related to [COM98], but involving viewcells. Our distance metric can also be directly used during the simplification process [Fre00]. Other papers could benefit from our work too. One example, is a recent paper on soft-shadows [AHL*06]. Here the occluder was represented using a depth map and the error was estimated based on the gap between depth samples. Our analysis allows a classification of this error and predicts the region on the ground for which the error is biggest.

We also implemented a view-dependent billboard clouds (VD-BBC) approach whose run-time scales linearly with the geometric complexity. BBC [DDSD03] are a simplification via planes. These planes are represented via textured and alpha mapped quads on which geometry is projected. The selection is performed based on a heuristic, working on a discretization of the plane space. Following the density (a value representing the amount of geometry (faces) that can be simplified on the plane) the algorithm greedily proceeds until all faces have been simplified. For VD-BBC, we exploit the validity regions to determine on which planes a face can be projected in two dimensions.

A first example is shown in figure 17. The method was applied to a 4194 triangle city model. Due to its simplicity the model does not seem to leave much possibilities to simplify. Nevertheless, the resulting representation contains only 115 textured quads while maintaining a proximity of 0.1° (≈ 5 pixels) neglecting the vertical error. The viewcell in this example is a halfspace situated at the lower bottom of the image. In particular it is interesting to see how the structure of the city is maintained for nearby parts, whereas far away areas are aggressively simplified. Especially the structure becomes unrecognizable from above, but appears close to the original as seen from the viewcell. As in the original BBC approach small cracks might be visible where neighboring faces project to different planes, but these openings are completely bounded by our approach.

The second example is shown in Fig. 18. The scene is a billboard forest around a centering segment viewcell. The left part depicts the 2D representation to which we applied the VD-BBC approach. The upper left quadrant shows a quarter of the scene and its geometric complexity. The lower left quadrant shows that our method successfully simplifies distant geometry. The red segments show the orientation of the created billboards and all shown green segments have been simplified on these. The figure shows only those billboards representing more than 100 trees. Finally the right half shows an example of this simplification. All green trees simplify to the same billboard whose orientation is shown in red. In particular you might notice that the simplified geometry forms a star. This results from the segment viewcell, which is aligned horizontally. The parallax effects are thus less pronounced along this direction. All in all we realize that even exact bounds still allow for aggressive simplification in higher polygon count scenes. Only $< 2.3\%$ of the original complexity remained. On the one hand this does not have a real signification because we could simply add more geometry to further improve these statistics. On the other hand it gives a good idea of the quality of the approximation.

7. Future Work

Lots of interesting avenues arise from this work. We solved validity in 3D for the special case of hyperplanes and point validity in a numerically stable way. An analytical closed form expression, as the one we presented for the 2D case, would be an interesting result. Treating arbitrary (non-polytope) viewcells also remains a challenge.

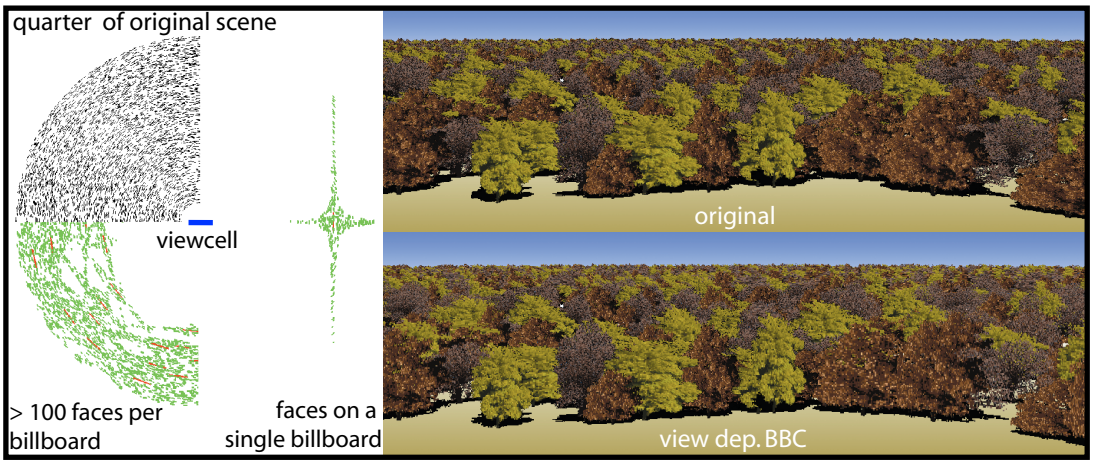

Figure 18: *The left two quadrants of the left part of the figure show the original scene complexity (top) and a selection of billboards containing more than* 100 *trees (the red segments indicate the billboards, the green elements the simplified trees). In this scene the viewcell is a centered segment. Its shape influences the validity regions and leads to a star like set of trees that are simplified on the same plane. Seen from the viewcell the original and the billboard cloud representation look very similar, even though less than* 2.3% *(compared to the original) of the geometry is involved.*

At the moment visibility is not involved in the calculation of the validity region. For two-dimensional scenes, several algorithms exist to calculate the visible part of the viewcell (e.g. [Hal02]). This information could be used, but algorithms are generally computationally expensive and might have numerical issues.

Perception starts playing an important role in simplification [WLC*03, LT00] and we would like to incorporate this kind of information in our approach. In particular we want to investigate what visual errors arise from a representation. Due to the way rasterization works on current cards, billboards can have a different appearance for grazing angles.

We would like to further explore other applications of our work, in the spirit of Cornish et al. [CRL01]. One particular idea consists in considering a light source as an observer to create simpler but more or less equivalent shadow casters.

Acknowledgments Thanks go to S. Lefebvre for his help with pBuffers and 3DSLib and the reviewers, P. Barla, S. Camerer, M. Eisemann, S. Grabli, S. Hornus, S. Paris, E. Turquin and H. de Almeida Bezerra for their comments.

References

[AHL*06] ATTY L., HOLZSCHUCH N., LAPIERRE M., HASENFRATZ J.-M., HANSEN C., SILLION F.: Soft shadow maps: Efficient sampling of light source visibility. *Computer Graphics Forum* (2006). (to appear).

[Cla76] CLARK J.: Hierarchical geometric models for visible surface algorithms. In *Communications of the ACM 19(10)* (1976).

[COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *Proc of SIGGRAPH* (1998), ACM Press.

[CRL01] CORNISH D., ROWAN A., LUEBKE D.: View-dependent particles for interactive non-photorealistic rendering. In *Proc. of Graphics Interface* (2001).

[CVM*96] COHEN J., VARSHNEY A., MANOCHA D., TURK G., WEBER H., AGARWAL P., BROOKS F., WRIGHT W.: Simplification envelopes. In *Proc. of Computer graphics and interactive techniques* (1996), ACM Press.

[DDSD03] DÉCORET X., DURAND F., SILLION F., DORSEY J.: Billboard clouds for extreme model simplification. In *Proc. of SIGGRAPH* (2003), ACM Press.

[DSSD99] DÉCORET X., SILLION F., SCHAUFLER G., DORSEY J.: Multi-layered impostors for accelerated rendering. *Proc. of Eurographics 18*, 3 (1999).

[Eis04] EISEMANN, ELMAR: *View-Dependent Object Simplification and its Application in the Case of Billboard Clouds*. Dea, UJF/INPG/ENSIMAG, 2004.

[ESSS01] EL-SANA J., SOKOLOVSKY N., SILVA C. T.: Integrating occlusion culling with view-dependent rendering. In *Proc. of the conference on Visualization* (2001), IEEE Computer Society.

[ESV99] EL-SANA J., VARSHNEY A.: Generalized view-dependent simplification. In *Computer Graphics Forum* (1999).

[Fre00] FREY P. J.: About surface remeshing. In *Proc. of 9th International Meshing Roundtable* (2000).

[Hal02] HALL-HOLT, OLAF: Kinetic visibility. *Ph. D.*

thesis, Department of Computer Science, Stanford University (2002).

[Hop96] HOPPE H.: Progressive meshes. *Computer Graphics 30*, Annual Conference Series (1996).

[Hop97] HOPPE H.: View-dependent refinement of progressive meshes. In *Proc. of SIGGRAPH* (1997).

[Hou62] HOUGH P.: Method and means for recognizing complex patterns. *US Patent 3,069,654* (1962).

[JW02] JESCHKE S., WIMMER M.: Textured depth meshes for real-time rendering of arbitrary scenes. In *Proc. of EuroGraphics Workshop on Rendering* (2002), Eurographics Association.

[JWS02] JESCHKE S., WIMMER M., SCHUMAN H.: Layered Environment-Map Impostors for Arbitrary Scenes. In *Proc. Graphics Interface* (2002).

[LE97] LUEBKE D., ERIKSON C.: View-dependent simplification of arbitrary polygonal environments. In *Proc. of SIGGRAPH* (1997).

[LKR*96] LINDSTROM P., KOLLER D., RIBARSKY W., HODGES L., FAUST N., TURNER G.: Real-time continuous level of detail rendering of height fields. In *Proc. of SIGGRAPH* (1996).

[LT00] LINDSTROM P., TURK G.: Image-driven simplification. *ACM Transactions on Graphics 19*, 3 (2000).

[MS95] MACIEL P. W. C., SHIRLEY P.: Visual navigation of large environments using textured clusters. In *Proc. of Symp. on Interactive 3D graphics* (1995), ACM Press.

[SDB97] SILLION F., DRETTAKIS G., BODELET B.: Efficient impostor manipulation for real-time visualization of urban scenery. In *Proc. of Eurographics* (1997).

[WLC*03] WILLIAMS N., LUEBKE D., COHEN J. D., KELLEY M., SCHUBERT B.: Perceptually guided simplification of lit, textured meshes. In *Proc. of Symposium on Interactive 3D Graphics* (2003).

[WM03] WILSON A., MANOCHA D.: Simplifying complex environments using incremental textured depth meshes. In *Proc. of SIGGRAPH* (2003), ACM Press.

[XV96] XIA J. C., VARSHNEY A.: Dynamic view-dependent simplification for polygonal models. In *Proc. of the conf. on Visualization* (1996), IEEE-CS Press.

Appendix A: V_{max} classification

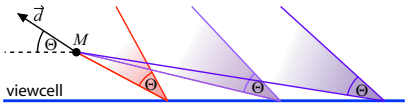

Figure 19: *If the angle of the projection direction and line viewcell lies in* $[\pi-\Theta,\pi+\Theta]$*, there is no meaningful max viewpoint for this side of the bicircle.*

Working with two circles instead of the bicircle might lead to an intersection with the part that lies in the interior of the bicircle. It can be shown that no meaningful max viewpoint occurs if the angle between $\vec{d}$ and the line viewcell lies in $[\pi-\Theta,\pi+\Theta]$. The case $\pi-\Theta$ leads to no intersection at all. For this "side" of the bicircle, the validity region extent is unbounded (fig. 19). Even without this test, false detections will be unproblematic (in the worst case, one unnecessary evaluation is performed).

Appendix B: Clipping faces

Faces intersected by the extension of a segment viewcell to a line, should be split because eq. (7) assumes the point M to lie in the upper half space. Of course, it is possible to deduce a similar equation for the lower one, but splitting and culling for polygonal viewcells implies that less geometry needs to be considered. The correctness is proven similar to the proof that max viewpoints lie on the boundary.

Appendix C: Tangents for Face Validity

If one supposes that the function can be simplified to $c\alpha$ (with $c \neq 0$) by division, we obtain $\alpha(\sin(\Theta+\gamma)-c\cos(\Theta+\gamma))+m\cos\gamma(\cos\Theta+c\sin\Theta)=0$ as $\cos\gamma\neq 0$. Assuming $m\neq 0$, the constant part implies $c=-\cot\Theta$. Leading to $\sin(\Theta+\gamma)+\tan\Theta\cos(\Theta+\gamma)=0$, and thus $\cos\gamma=0$. This case had been excluded, as the face would be vertical. If $m=0$, $\mathcal{VR}_{face}$ simplifies to $tan(\Theta+\gamma)\alpha$. A linear function, except if $cos(\Theta+\gamma)=0$, then the ray becomes parallel to the projection direction $(0,1)^\top$ (validity extents are unbounded).

The case $\sin(\Theta+\gamma)=0$ leads to a real hyperbola and both branches are monotonic. There cannot be a tangent with normal $(0,1)\top$ at a hyperbola. One exception occurs if the numerator is completely zero (therefore $m=0$), here we have a linear function, the x-axis. (fig. 20).

The case $\cos(\Theta+\gamma)=0$ leads to a parabola (fig. 20) except for $m=0$, when the function is always undefined (see first case). In the parabola case the extremum is at

$$\alpha_{parabola} = -\frac{m\sin\Theta\cos\gamma}{2\sin(\Theta+\gamma)}cot\Theta \qquad (12)$$

For a quadratic numerator, a linear denominator that cannot be simplified by division the extrema are:

$$\alpha_{1,2} = \frac{m\cos\gamma}{\cos(\Theta+\gamma)}\left(\sin\Theta \pm \sqrt{\frac{\cos\gamma\sin\Theta}{\sin(\Theta+\gamma)}}\right) \qquad (13)$$

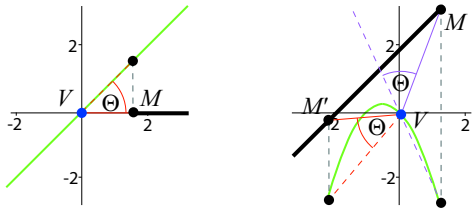

Figure 20: *Special Cases: Left: line, Right: Parabola (right cone is an example for the false intersections*