On the Visibility of Point Clouds

Sagi Katz
Technion–Israel Institute of Technology
sagi.katz@gmail.com

Ayellet Tal
Technion–Israel Institute of Technology
ayellet@ee.technion.ac.il

Abstract

Is it possible to determine the visible subset of points directly from a given point cloud? Interestingly, in [7] it was shown that this is indeed the case—despite the fact that points cannot occlude each other, this task can be performed without surface reconstruction or normal estimation. The operator is very simple—it first transforms the points to a new domain and then constructs the convex hull in that domain. Points that lie on the convex hull of the transformed set of points are the images of the visible points. This operator found numerous applications in computer vision, including face reconstruction, keypoint detection, finding the best viewpoints, reduction of points, and many more. The current paper addresses a fundamental question: What properties should a transformation function satisfy, in order to be utilized in this operator? We show that three such properties are sufficient—the sign of the function, monotonicity, and a condition regarding the function's parameter. The correctness of an algorithm that satisfies these three properties is proved. Finally, we show an interesting application of the operator—assignment of visibility-confidence score. This feature is missing from previous approaches, where a binary yes/no visibility is determined. This score can be utilized in various applications; we illustrate its use in view-dependent curvature estimation.

1. Introduction

The last decade has witnessed a vast increase in the use of range imaging and 3D scanning devices. A point cloud, sampled from a surface, is the standard output of these devices. This paper explores an operator that determines the visibility of a point cloud, given a viewpoint.

As points cannot occlude each other (unless they accidentally fall on the same line from the viewpoint), the traditional way to solve the problem is to reconstruct the surface [1, 8] and determine visibility on the reconstructed surface. In [7] an elegant operator is proposed to determine the sought-after subset directly from the point set, without surface reconstruction or normal estimation, as demonstrated in Figure 1. This operator, termed the *HPR (Hidden Point Removal)* operator, is supported by theoretical guarantees.

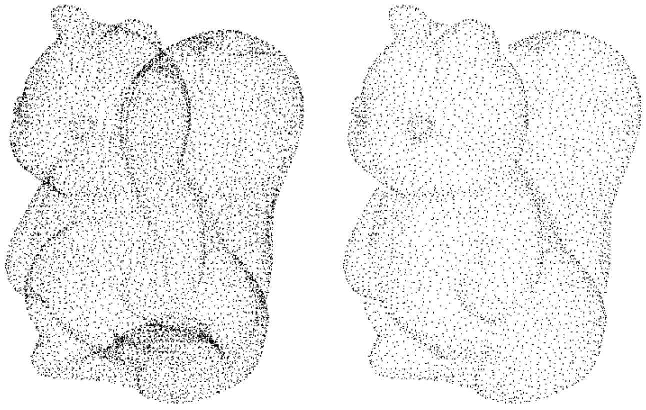

(a) Input (b) Output

Figure 1. **Given points sampled from a surface and a viewpoint** (a), only the points that would be visible if the surface were known, are extracted (b).

This operator has found numerous applications, both within computer vision and computer graphics, and in other domains. Some of these applications are surface reconstruction [3, 12], 3D face reconstruction [9], 3D keypoint detectors [17], finding the best views of 3D shapes [14, 16], finding silhouettes [15], orienting point sets [2], flock animation [10], 3D pose estimation [4], object relighting [18], determining meaningful points [6], and even wireless communication [11, 13].

Given a point cloud P and a viewpoint C, the operator consists of two steps:

1. **Point transformation:** A function maps every point $p_i \in P$ to an inverted domain. Assuming, without loss of generality, that the viewpoint is at the origin, the *flipping* transformation function is defined as

$$\widehat{p_i} = F(p_i) = p_i + 2(R - ||p_i||)\frac{p_i}{||p_i||}, \qquad (1)$$

where the parameter $R > 0$ is the radius of a sphere.

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.159

2. **Convex hull construction:** The convex hull of the transformed points and the viewpoint is calculated.

The main result of [7] is that the points that reside on the convex hull of Step 2 are the images of the visible points.

Our goal in this paper is to generalize the HPR operator by identifying the properties that should be satisfied by any function used in Step 1. The main contribution of this paper is establishing three such properties: the sign of the function, monotonicity, and a condition regarding the function's parameter (Section 2). This is not only interesting theoretically, but may also be practical, as various applications can benefit from different functions that suit their needs.

We analyze the generalized HPR operator (*GHPR*) that satisfies these requirements (Section 3). We also prove its correctness in the limit and provide guarantees for the more practical case of finite sampling (Section 4).

An additional contribution is demonstrating the usefulness of the operator for a new type of applications, which require a score reflecting the confidence in the visibility, rather than a binary visibility decision (Section 5). We present an example of such application: view-dependent curvature estimation.

2. Properties of the transformation function

Figure 2 describes the pseudo code of the generalized HPR operator (GHPR) for visibility detection. This section focuses on the first step of the operator. Our goal is to identify the properties of the transformation function of Step 1, which are essential for the correctness of the visibility operator.

```
GHPR($P$,$C$,$\gamma$)
    Pre-processing: Move the points in $P$ s.t. C is the origin
    Step 1: Apply the transformation in Equation 2 $\forall p_i \in P$, using $\gamma$;
    Step 2: Apply a convex hull algorithm to the set of transformed points
    Output: Return the set of points whose images reside on the convex hull
```

Figure 2. Pseudo-code of the GHPR operator

We are given a point set $P \subset \mathbb{R}^D$ sampled from a continuous surface, a point $p_i \in P$ and a viewpoint C. Let f be a 1-dimensional continuous kernel function $f : \mathbb{R}^+ \to \mathbb{R}^+$ that, given the distance of p_i from C, outputs an updated distance after applying f. We assume that f is invertible. We define $F_f : \mathbb{R}^D \to \mathbb{R}^D$ to be a radial transformation, as follows:

$$F_f(p_i, C) = \begin{cases} C + \frac{p_i - C}{\|p_i - C\|} f(\|p_i - C\|) & p_i \neq C \\ p_i, & p_i = C \end{cases} .$$

If we define, without loss of generality, a coordinate system where C is at the origin, we get:

$$F_f(p_i) = \begin{cases} \frac{p_i}{\|p_i\|} f(\|p_i\|), & p_i \neq 0 \\ 0 & p_i = 0 \end{cases} . \quad (2)$$

Claim 2.1 *The inverse transformation of F_f is:*

$$F_f^{-1}(\widetilde{p_i}) = \begin{cases} \frac{\widetilde{p_i}}{\|\widetilde{p_i}\|} f^{-1}(\|\widetilde{p_i}\|), & \widetilde{p_i} \neq 0 \\ 0 & \widetilde{p_i} = 0 \end{cases} , \quad (3)$$

where f^{-1} is the inverse kernel, such that $f^{-1}(f(d)) = d$.

We identify the following sufficient properties, which should hold in order for a kernel, f, to be used for visibility determination (i.e., they will be used in the subsequent sections):

1. $f'(d) < 0$, or in other words, $f(d)$ is monotonically decreasing, s.t. $f(\|p_i\|) < f(\|p_j\|)$ iff $\|p_i\| > \|p_j\|$. This means that points that are closer to the viewpoint become farther away after being transformed.

2. $f(d) > 0$. This condition assures that the orientation of the points is maintained relative to the viewpoint after the transformation.

3. For a γ-controlled kernel f_γ, it is required that for any $d_1, d_2 \in \mathbb{R}^+$, s.t. $d_1 > d_2$ and for any $0 < \epsilon < 1$, there exists a value of $\gamma = \Gamma$, s.t.

$$1 > \frac{f_\Gamma(d_1)}{f_\Gamma(d_2)} > 1 - \epsilon. \quad (4)$$

It should be noted that the left-hand side of the equation results directly from the monotonicity of f and therefore, it is correct for any value of γ. The additional requirement on the right-hand side will be shown useful for proving, in Section 4, the correctness in the limit of GHPR, as well as for providing guarantees when P is a sample of the surface.

Exemplary kernels: It can be easily verified that the following kernels satisfy the required properties.

1. **Mirror/Linear Kernel**: This kernel, which was used in [7], can be written as $f_{mirror}(d) = \gamma - d, \gamma \geq max_{p_i \in P}(\|p_i\|)$. It describes flipping around an imaginary spherical mirror, centered at C, with a user specified radius $R = \frac{1}{2}\gamma$.

2. **Exponential Inversion Kernel:** This kernel is defined as $f_{exponential}(d) = d^\gamma$, where $\gamma < 0$ is a parameter.

3. **Natural Exponential Kernel:** This kernel is defined as $f_{natural}(d) = e^{-\gamma d}, \gamma > 0$.

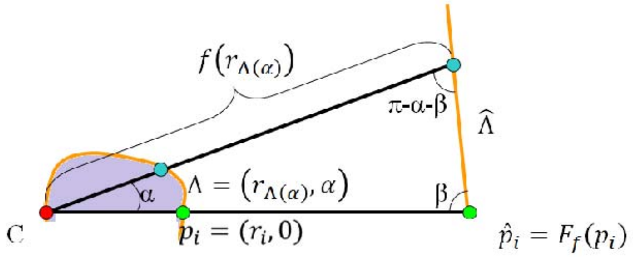


Figure 3. The curve Λ is transformed to the line $\hat{\Lambda}$ by F_f. The empty region associated with Λ is in purple.

We note that in [6] a related operator is introduced: the *Target-Point Occlusion (TPO)* operator. TPO finds the points that would occlude a given point from outside observers. This operator can be generalized very similarly to what is proposed in this paper, with modifications only to properties 1 and 3 above (the function should be monotonically increasing, rather than decreasing).

3. Empty regions & The visibility condition

According to the visibility condition, for a point p_i to be visible, its image $F_f(p_i)$ should lie on the convex hull of the set of transformed points. This means that the curve that is the source of an edge of that convex hull, should be empty of points of P. This is illustrated by the purple region in Figure 3. Intuitively, if the empty region between $p_i \in P$ and the viewpoint is large, our confidence in the visibility should be large as well. Therefore, if we could determine the empty region associated point p_i, we could threshold its size.

This section explores this curve. It begins by defining a family of curves, Λ-*curves*, which are transformed to lines by the GHPR operator in $\mathbb{R}^2$ (Section 3.1). Then, we discuss the empty region induced by a Λ-curve (Section 3.2). We prove the relationship between the size of the empty region and the angle between the x-axis and the line to which the Λ-curve is transformed. Finally, we use this relationship to define a mathematical condition for visibility, which can be efficiently computed (Section 3.3). This section generalize the proofs that were given in [7] for a specific function, to the family of functions that satisfy the conditions presented in Section 2.

3.1. Boundaries of empty regions – Λ-curves

We are given a kernel f, a point cloud P, a viewpoint C, and a point $p_i \in P$.

Definition 3.1 Λ-***curve:*** *Given p_i, $\Lambda(\alpha) = (r_\Lambda(\alpha), \alpha)$ is a parametric curve that is the pre-image under F_f of a line passing through a transformed point $F_f(p_i)$.*

The following lemma can be easily proved.

Lemma 3.1 *A Λ-curve passes through the viewpoint $C = (0, 0)$.*

From Definition 3.1, it is clear that this curve is important for understanding the empty region associated with p_i. This is so, since after computing the convex hull during the second stage of the GHPR operator, p_i is visible if $\hat{p_i}$ is on the convex hull, that is to say, all the transformed points reside to one side of a straight line (half-space). We are therefore interested in determining the shape of Λ.

Without loss of generality, attach a polar coordinates system (R, α), such that its origin is at the viewpoint C, and the x-axis ($\alpha = 0$) lies on the line connecting C with p_i (Figure 3). The line $\hat{\Lambda} = \{\hat{q} = F_f(q) | q \in \Lambda\}$ represents the straight line to which the curve Λ is transformed by F_f. This line creates an angle β with the x-axis. Any point on the Λ-curve, having a polar angle α, is transformed to a point on the line $\hat{\Lambda}$ with distance $f(r_{\Lambda(\alpha)})$ from C.

Using the Law of Sines we get: $\frac{f(r_i)}{sin(\pi-\alpha-\beta)} = \frac{f(r_\Lambda(\alpha))}{\sin\beta}$. Therefore, for $p_i \neq C$, we define the parametric equation of $\Lambda(\alpha)$ with respect to this polar coordinate system as:

$$\Lambda(\alpha) = (r_\Lambda(\alpha), \alpha) = \left(f^{-1}(\frac{f(r_i)\sin\beta}{\sin(\alpha+\beta)}), \alpha \right). \quad (5)$$

Since the inverse function f^{-1} is assumed to exist, Λ always exists. Figure 4 shows the Λ-curve profiles for our exemplary kernels.

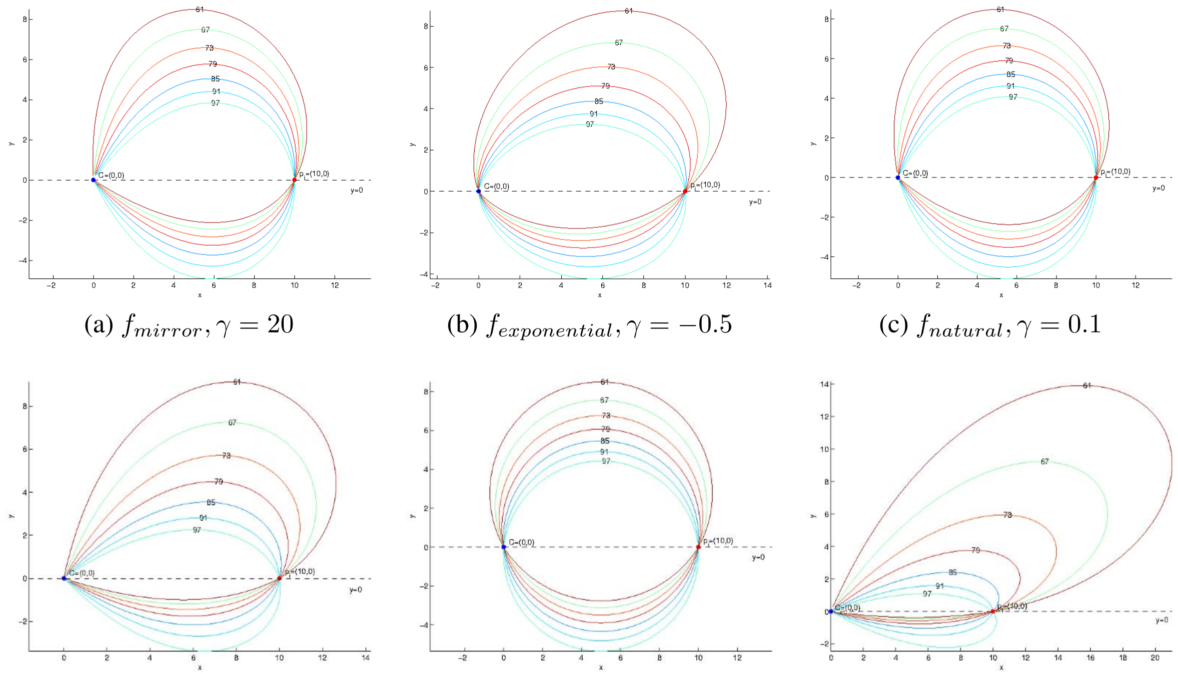
3.2. Empty regions induced by the Λ-curves

We are interested in finding the largest empty region, which indicates our confidence in the visibility result. We start by defining the region associated with a Λ-curve. Then, in Lemma 3.2 we show a way to measure the relative sizes of such regions. The result of this lemma lets us characterize in Lemma 3.3 the largest empty region, formally define it (Definitions 3.3, 3.4) and prove that the definition is correct, i.e.,that the region is indeed empty of points (Lemmas 3.4, 3.5).

Definition 3.2 Ω_Λ***-region:*** *Let Ω_Λ be the region bounded by the x-axis and the curve $\Lambda(\alpha) = (r_\Lambda(\alpha), \alpha), 0 \leq \alpha < \pi$, s.t. $p_q = (r_q, \theta) \in \Omega_\Lambda$ iff $0 < r_q < r_\Lambda(\theta)$.*

The next lemma proves the relation between the size of Ω_Λ and the angle β between the line $\hat{\Lambda}$ and the line connecting the viewpoint to point p_i. In particular, it proves that as the first increases, the latter decreases.

Lemma 3.2 *Let Ω_{Λ_1} and Ω_{Λ_2} be two regions associated with a point $p_i \in P$ and defined by Λ_1 and Λ_2, where the lines $\hat{\Lambda}_1$ and $\hat{\Lambda}_2$ create angles β_1 and β_2 with the x-axis, respectively. Then, $\Omega_{\Lambda_1} \subseteq \Omega_{\Lambda_2} \iff \beta_1 > \beta_2$.*



(a) $f_{mirror}, \gamma = 20$ (b) $f_{exponential}, \gamma = -0.5$ (c) $f_{natural}, \gamma = 0.1$

(d) $f_{mirror}, \gamma = 40$ (e) $f_{exponential}, \gamma = -1$ (f) $f_{natural}, \gamma = 0.01$

Figure 4. **Λ-curve profiles** for different kernels and parameter values for a single point, $p_i = (10, 0)$ and $C = (0, 0)$. Each graph contains several curves, each corresponds to a different value of β. It can bee seen that the size of the region captured between the x-axis and the curve above the x-axis increases as β gets smaller.

Proof: We prove one direction of the lemma; the other direction is similar. Assume that $\beta_1 > \beta_2$; we want to prove that $\Omega_{\Lambda_1} \subseteq \Omega_{\Lambda_2}$. This is done by showing that the boundary, Λ_1, of the region associated with β_1, is closer to the origin compared to Λ_2, for every possible angle $0 < \alpha < \pi$. This means that for a region boundary $\Lambda = (r_\Lambda(\alpha), \alpha)$, $r_\Lambda(\alpha)$ increases when β decreases. Therefore, we need to show that $\frac{\partial r_\Lambda}{\partial \beta} < 0$.

Applying the derivation by β to $r_\Lambda = f^{-1}\left(\frac{f(r_i)\sin\beta}{\sin(\alpha+\beta)}\right)$ from Equation (5), results with

$$\frac{\partial r_\Lambda}{\partial \beta} = \qquad (6)$$

$$= f(r_i)\frac{cos\beta sin(\alpha+\beta) - sin\beta cos(\alpha+\beta)}{sin^2(\alpha+\beta)} f^{-1'}\left(\frac{f(r_i)sin\beta}{sin(\alpha+\beta)}\right)$$

$$= f(r_i)\frac{sin(\alpha)}{sin^2(\alpha+\beta)} f^{-1'}\left(\frac{f(r_i)sin\beta}{sin(\alpha+\beta)}\right).$$

In order to find the sign of $\frac{\partial r_\Lambda}{\partial \beta}$, we note that: (1) $f(r_i) > 0$ by definition. (2) $\sin(\alpha) > 0$ for $0 < \alpha < \pi$ and therefore, $\frac{sin(\alpha)}{sin^2(\alpha+\beta)} > 0$. (3) Since $f'(f^{-1}(x)) \neq 0$ (since f is strictly monotonically decreasing) then $f^{-1'}(x) = \frac{1}{f'(f^{-1}(x))}$, and by the derivative of inverse function rule, we get

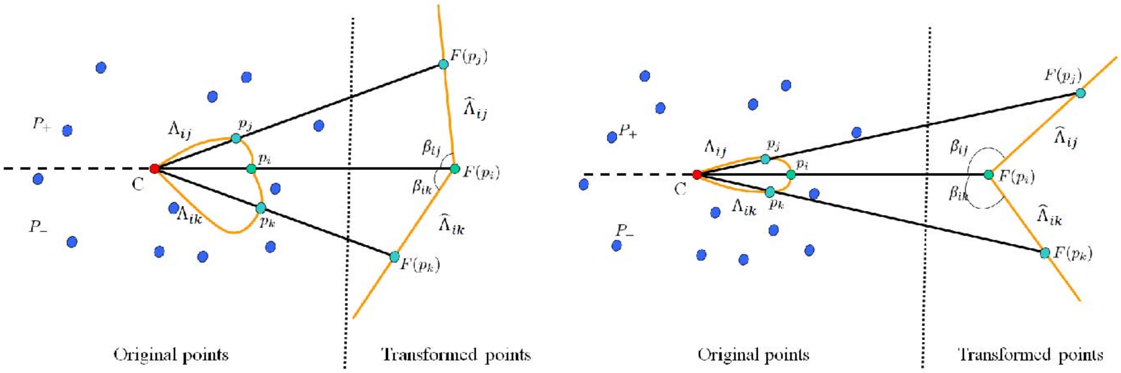
$$f^{-1'}\left(\frac{f(r_i)sin\beta}{sin(\alpha+\beta)}\right) = \frac{1}{f'(f^{-1}(\frac{f(r_i)sin\beta}{sin(\alpha+\beta)}))}.$$

For Λ we required that $f'(d) < 0$, so $f^{-1'}(\frac{f(r_i)sin\beta}{sin(\alpha+\beta)}) < 0$. Therefore, the term $\frac{\partial r_\Lambda}{\partial \beta}$ is always negative.□

Lemma 3.3 *For a given point $p_i \in P$, the Λ_{max}-curve, which passes through p_i and defines the largest empty region $\Omega_{\Lambda_{max}}$, passes through at least one additional point $p_j \in P$.*

Proof: Suppose, by way of contradiction, that Λ_{max} does not pass through an additional point. Then, $\hat{\Lambda}_{max}$ does not pass through an additional transformed point and we can define a line $\hat{\Lambda}$, with a smaller β angle, which would still create an empty half space. According to Lemma 3.2, this half space originates from a larger region, which contradicts the assumption.□

Combining the above lemmas suggests that in order to find the largest empty region for a point p_i, one needs to find another point in the transformed domain, for which the line connecting the two transformed points, creates the smallest possible angle β. We therefore define a region associated with a pair of points:



(a) p_i is considered visible (b) p_i is considered invisible

Figure 5. **Visibility condition.** If the angle $\beta_{ij} + \beta_{ik}$ between the lines (to which the Λ-curves are transformed) is greater than π, then the point is invisible; otherwise it is visible.

Definition 3.3 ***A region $\Omega_{i,j}$ associated with two points $\mathbf{p_i}$ and $\mathbf{p_j}$:*** *$\Omega_{i,j}$ is the region associated with a Λ-curve transformed by F_f to the line segment connecting p_i with p_j.*

Until now we considered only points in the upper half-plane. We now extend our definitions to the whole set P. We divide P into two subsets: P_+ and P_-, where P_+ contains all the points above the x-axis and P_- contains the points below it. The region associated with a point p_i is defined as follows:

Definition 3.4 ***A region Ω_i associated with a point*** *p_i is*

$$\Omega_i = (\bigcap_{p_m \in P_-, m \neq i} \Omega_{i,m}) \bigcup (\bigcap_{p_n \in P_+, n \neq i} \Omega_{i,n}).$$

Lemma 3.4 *Let $p_j \in P_+$ and $p_k \in P_-$ be the points that minimize the sizes of $\Omega_{i,j}$ and $\Omega_{i,k}$ correspondingly. The region Ω_i can be calculated by $\Omega_i = \Omega_{i,j} \cup \Omega_{i,k}$.*

Proof: We choose a point $p_k \in P_-$ below the bisecting line, such that $\Omega_{i,k}$ is minimized. This means that β_{ik} is maximized among all the points that are above the bisecting line. Using the result of Lemma 3.2, $\Omega_{i,k}$ is contained in all $\Omega_{i,m}$ for $p_m \in P_-$. Therefore, $\Omega_{i,k} = \bigcap_{p_m \in P_-} \Omega_{i,m}$ and similarly, $\Omega_{i,j} = \bigcap_{p_n \in P_+} \Omega_{i,n}$.□

Lemma 3.5 *Ω_i is empty of points from P.*

Proof: In order to show that Ω_i is empty, it is enough to show that $\Omega_{i,j}$ and $\Omega_{i,k}$, as defined in Lemma 3.4, are empty. Assume, by way of contradiction, that $\Omega_{i,j}$ is not empty. Then, it contains a point p_q for which the line $\hat{\Lambda}_{i,q}$ creates an angle β_{iq} with the x-axis, such that $\beta_{iq} > \beta_{ij}$. This contradicts the fact that $\Omega_{i,j}$ is minimized and maximizes $\beta_i j$. Similarly, it is possible to show that $\Omega_{i,k}$ is empty.□

3.3. Condition on visibility

To determine visibility, we should threshold the points according to the size of their associated empty region. Intuitively, points associated with large regions are visible. Using Lemma 3.2, we can threshold the values of β instead of directly thresholding the size of the empty region. We formulate our condition for visibility as:

$$\beta_{i,j} + \beta_{i,k} \leq threshold, \qquad (7)$$

where j and k are the indices of neighboring points as defined in Lemma 3.4.

One way to apply the condition is to find, for each point $p_i \in P$, two other points $p_j \in P_+$ and $p_k \in P_-$ that minimize the β angles, as shown in Figure 5. However, this method is inefficient.

Instead, setting $threshold = \pi$ gives rise to an efficient method (e.g., $O(n \log n)$ in 2D and 3D), the *generalized-HPR (GHPR)* operator. This is so since $\beta_{i,j} + \beta_{i,k} \leq \pi$ means that $\hat{p}_i$ is on the convex hull of $\hat{P}$, where P is transformed to $\hat{P}$, as illustrated in Figure 5:

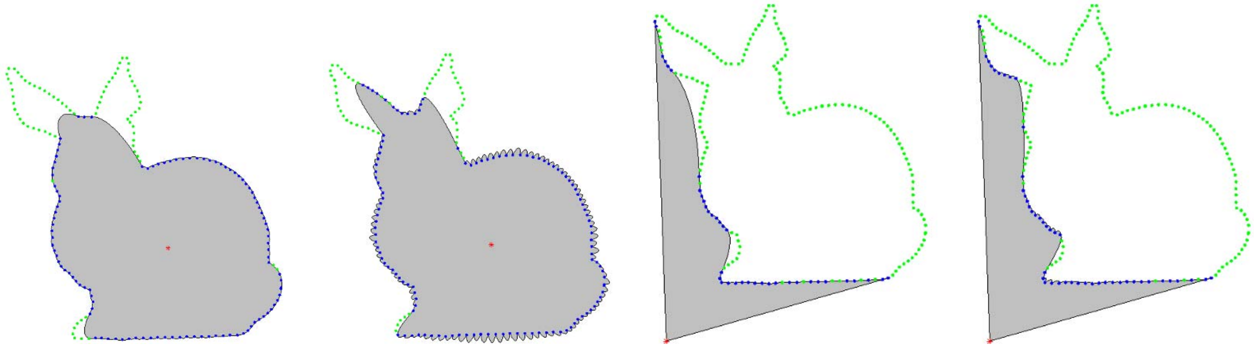
$$GHPR(P) \equiv (P \cap \{C\}) \cup \{p | \hat{p} \in convexhull(\hat{P} \cup \{C\})\}.$$

Note that using the convex hull construction is more than just accelerating the computation. A theoretical beauty of this approach is that it essentially provides a reduction of the problem of visibility detection to that of convex hull construction—linking two seemingly different problems.

Figure 6 shows examples of applying the GHPR operator. In these drawings, blue points on the boundary of the gray regions are visible, while the green points are occluded.

4. Correctness & Properties

In this section, we first prove the correctness of the GHPR operator in the limit, where the distance between



(a) $\gamma = -0.05$, C is inside P

(b) $\gamma = -0.01$, C is inside P

(c) $\gamma = -0.05$, C is external to P

(d) $\gamma = -0.01$, C is external to P

Figure 6. **The visible region,** marked in gray, as viewed from a given point in red, calculated using the exponential inversion kernel. Blue points on the boundary of the gray region are visible, whereas green points, which are not on the boundary of the gray region, are occluded. It can be seen that the size of the region detected as visible increases as γ gets closer to 0.

the points approaches 0. Then, we provide theoretical guarantees for the case where P is a finite sample of S. The proofs generalize those of the HPR operator [7], which assume the special case of a mirror kernel. They rely on the three properties introduced in Section 2 and are given in the supplementary.

The next three lemmas assume that the input consists of the set of all the points of the surface S and a viewpoint C. Let $V \subseteq S$ be the set of visible points from C and $GHPR(P) \subseteq S$ be the set detected as visible by the GHPR operator, using a kernel f_γ with parameter γ.

Lemma 4.1 *$GHPR(P) \subseteq V$, i.e., every point detected visible by the GHPR operator is indeed visible from C.*

Lemma 4.2 *For every valid kernel, f_γ, there exists a value Γ s.t. $\lim_{\gamma \to \Gamma} GHPR(P) = V$, assuming $T = \inf\{\|p - C\| \; |p \in S\} > 0$.*

In other words, Lemma 4.2 guarantees that when $\gamma \to \Gamma$, the set of points detected by the GHPR operator is equal to the set of visible points—an accurate solution is obtained. If the condition for the value of γ does not hold, we can further analyze which points are detected visible by the GHPR operator, by considering the influence of the curvature on the results, as follows.

Lemma 4.3 *Let S be an infinitesimal surface patch around p. Then, $p \in GHPR(P)$ if and only if p is visible and the curvature κ at p is below a threshold κ_Λ:*

$$\kappa_\Lambda = \frac{(\sin(\beta)^3 - \sin(\beta))(f_b(d)f(d)d + f_a(d)d - 2f_a^2(d)) - f_a(d)d\sin(\beta) + d^2\sin(\beta)^3}{(f_a(d)^2 + (d^2 - f_a(d)^2)\sin(\beta)^2)^{\frac{3}{2}}},$$
$$\textit{where } f_a(d) = f(d) \cdot (f^{-1})'(f(d)), f_b(d) = f(d) \cdot (f^{-1})''(f(d)). \tag{8}$$

In conclusion, given a kernel f with a specific parameter, the GHPR algorithm correctly detects the visible points if

1. Locally, the surface is either convex or concave with sufficiently low curvature.

2. The surface is close enough to the viewpoint C (i.e., $d = ||p||$ is sufficiently small).

3. The angle between the surface normal and the line of sight is sufficiently small.

This means that misclassification errors are expected to occur around regions whose tangent plane is parallel to the line of sight or with large surface perturbations. Furthermore, these errors become worse for far-away points.

Now assume that P is a ρ-sample of S, with $\rho > 0$, i.e., for every sample $p \in P$, there exists another point $q \in P$ whose distance to p is smaller than ρ. We consider a point to be $\epsilon - visible$, if moving it by ϵ will make it visible. Using these definitions, we extend the correctness lemmas to the more practical case of sampled data.

Formally, let $V_\epsilon \subseteq P$ be the set of ϵ-visible points from C. We assume that the distance of S to C is at least $T > 0$ and that the sample is sufficiently dense.

Theorem 4.4 *For every valid γ, there exists $\epsilon > 0$ such that every point detected visible is $\epsilon-$visible.*

Theorem 4.5 *For sufficiently large $\epsilon > 0$, there exists a value of γ s.t. a point is detected visible only if it is $\epsilon-$visible.*

In summary, we proved that assuming that the sample is sufficiently dense, for every γ, there exists an ϵ, such that every point marked visible by the operator is $\epsilon-$visible. Moreover, for sufficiently large ϵ, there exists γ, such that every point marked visible by the operator is indeed $\epsilon-$visible.

5. Application: Assigning Visibility Scores

As mentioned before, point visibility is a fundamental component in many applications, both within computer vision and in other communities [3, 9, 12, 14, 15, 16, 17, 18]. Traditionally, in these applications, a binary decision is utilized, i.e., for each point, it is determined whether it is visible or not.

We propose an alternative—rather than producing a binary visibility decision, we will generate the degree of confidence in the visibility results. In many situations, this additional information is beneficial. For instance, in registration of point clouds generated by range cameras, points having high visibility scores may get high weights, hence influencing the registration more than points for which the scanner is likely to err. In the sequel, however, we discuss a different use— view-dependent curvature estimation.

Our goal is to attach to each point, which was detected visible, a *Visibility Score (VS)*. The key idea is to assign large visibility scores to points associated with large empty regions. Therefore, we wish to efficiently measure the size of the empty region.

To do that, recall that the size of the empty region is large when the sum of the β angles is small. For points on the convex hull that are detected as visible, these angles can be directly calculated from the convex hull. Hence, for a visible point $p_i \in P$ in 2D, we define VS as the angle between the edges on the convex hull that are adjacent to $\hat{p}_i$. Similarly, in 3D we define the visibility score as the sum of angles between the edges of the triangles on the convex hull, which are adjacent to $\hat{p}_i$.

Figure 7(left) shows results of our method, using the exponential inversion kernel, where dark blue indicates high values of VS and red marks lower values. The middle column shows the same results from another angle, to demonstrate that invisible points get the value 0.

As shown in Equation (8), VS is affected by the local curvature, the angle between the normal and the line of sight, and the distance of the surface to the viewpoint. For example, the creases in the hand model (Figure 7(a)) have a low value of VS due to deep concavities. In the Bimba model (Figure 7(b)), it can be seen that regions whose tangents create almost-perpendicular angles with the line of sight receive low scores.

View-dependent curvature estimation: Judd et al. [5] presented a method for extracting apparent ridges of surfaces (represented as meshes), which are the maxima of the view-dependent curvature. As shown below, VS behaves like a view-dependent curvature and can, therefore, be used for direct drawing the apparent ridges of point sets, without surface reconstruction.

The right column in Figure 7 shows the results of the commonly-used estimation of the curvature of P, subtracting the sum of angles from 2π. This estimation is performed on the reconstructed polygonal models.

Our curvature is estimated similarly, as the sum of the angles adjacent to the point, but this is done on the convex hull of the transformed set $\hat{P}$. The results show that our estimation indeed finds the features with high curvatures, and it is less sensitive to noise. Intuitively, this can be explained by the fact that the calculated visibility is a global measure and is therefore more resistant to small noise, compared to the conventional calculation of the curvature.

6. Conclusion

This paper has addressed the detection of visible points from a viewpoint. It generalizes the HPR operator, by answering the fundamental question of which properties a transformation function should satisfy in order to be applicable to visibility calculation. The paper enumerates three such properties and uses them to analyze the operator. It proves the operator's correctness in the limit and provides guarantees for the real-life case of finite sampling.

The operator is very simple, fast, and easy to implement. Moreover, it can be applied to points in any dimension, though we have focused in the paper on the practical cases of point sets in two dimensions and in three dimensions.

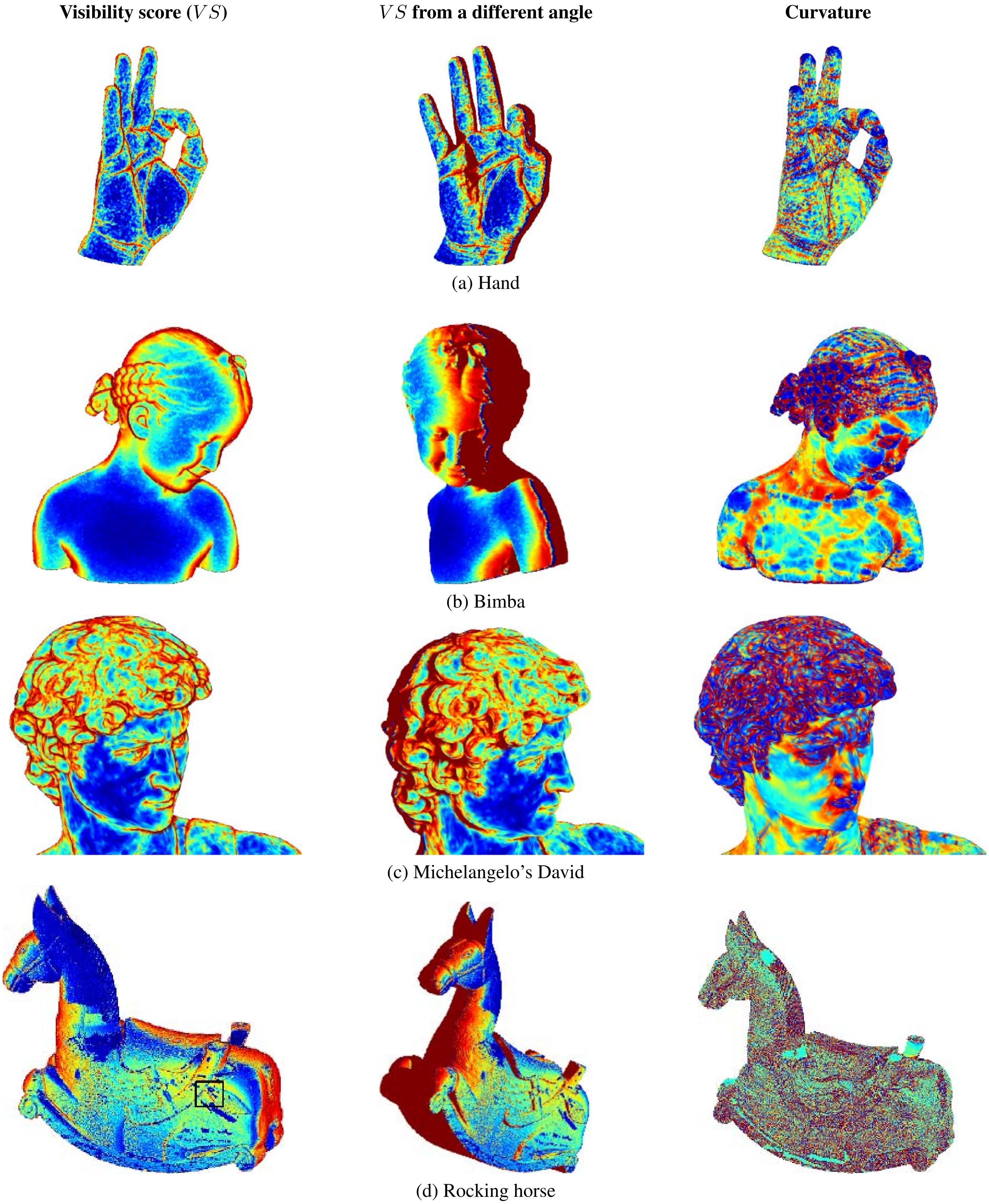
Last but not least, we present an additional benefit of this operator—its ability to compute a continuous visibility score of points, rather than a binary score, as commonly done. We show one concrete use of this score to directly compute the view-dependent curvature of a point set, skipping reconstruction. We demonstrate that, surprisingly, our results are good even though we are not using the reconstructed surfaces. With the growing popularity of scanning devices and 3D point sets, we believe that many applications will follow.

Future direction: An interesting question that this paper has not discussed is what a good kernel is. We expect it to be application-specific. Moreover, we believe that for applications where the shape of the empty region is known (e.g. cellular communication), it may be possible to design specific kernels.

Further analysis in terms of the value of γ and in terms of noise could also be performed.

Last but not least, sufficient conditions were identified in this work. But what are the necessary conditions? This question is open and is highly intriguing.

Acknowledgements: This research was supported in part by the Israel Science Foundation (ISF) 1420/12, the Binational Science Foundation (BSF) 2012376,, the Technion Funds for Security Research, and the Ollendorff Foundation.



Visibility score (VS)

VS from a different angle

Curvature

(a) Hand

(b) Bimba

(c) Michelangelo's David

(d) Rocking horse

Figure 7. **Results of calculating the point visibility score VS.** The left column shows VS as colors over a surface, where dark blue indicates highly-visible regions and red regions are invisible ($VS = 0$). The middle column shows the same results from another angle, without re-calculating VS. The right column shows estimation for the curvature; it can be compared to our results in the left column.

References

[1] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. Silva. State of the art in surface reconstruction from point clouds. In *Eurographics STAR*, 2014. 1

[2] J. Cao, Y. He, Z. Li, X. Liu, and Z. Su. Orienting raw point sets by global contraction and visibility voting. *Computers & Graphics*, 35(3):733 – 740, 2011. 1

[3] Y. Chen, B. Chen, S. Lai, and T. Nishita. Binary orientation trees for volume and surface reconstruction from unoriented point clouds. *Computer Graphics Forum*, 29(7):2011–2019, 2010. 1, 7

[4] C. Cheng, H. Chen, T. Lee, S. Lai, and Y. Tsai. Robust 3D object pose estimation from a single 2D image. In *Visual Communications and Image Processing*, pages 1–4, 2011. 1

[5] T. Judd, F. Durand, and A. E. Apparent ridges for line drawing. *ACM Transactions on Graphics*, 26(3), 2007. 7

[6] S. Katz and A. Tal. Improving the visual comprehension of point sets. In *CVPR*, 2013. 1, 3

[7] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. *ACM Transactions on Graphics*, 26(3):24:1–11, 2007. 1, 2, 3, 6

[8] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 2013. 1

[9] I. Kemelmacher-Shlizerman and R. Basri. 3D face reconstruction from a single image using a single reference face shape. *PAMI*, 33(2):394–405, 2011. 1, 7

[10] M. Klotsman and A. Tal. Animation of flocks flying in line formations. *Artificial life*, 18(1):91–105, 2012. 1

[11] F. Li, J. Luo, C. Zhang, S. Xin, and Y. He. Unfold: uniform fast on-line boundary detection for dynamic 3D wireless sensor networks. In *ACM Int. Symposium on Mobile Ad Hoc Networking and Computing*, pages 14:1–14:11, 2011. 1

[12] R. Mehra, P. Tripathi, A. Sheffer, and N. J. Mitra. Visibility of noisy point cloud data. *Computers & Graphics*, 34(3):219–230, 2010. 1, 7

[13] R. Mochaourab and E. Jorswieck. Optimal beamforming in interference networks with perfect local channel information. *IEEE Transactions on Signal Processing*, 59(3):1128 –1141, 2011. 1

[14] M. Mortara and M. Spagnuolo. Semantics-driven best view of 3D shapes. *Computers & Graphics*, 33(3):280–290, 2009. 1, 7

[15] M. Olson, R. Dyer, H. Zhang, and A. Sheffer. Point set silhouettes via local reconstruction. *Computers & Graphics*, 35(3):500–509, 2011. 1, 7

[16] E. Shtrom, G. Leifman, and A. Tal. Saliency detection in large point sets. In *ICCV*, 2013. 1, 7

[17] F. Tombari, S. Salti, and L. Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013. 1, 7

[18] G. Yang and Y. Liu. 3D object relighting based on multiview stereo and image based lighting techniques. In *IEEE Int. Conf. on Multimedia and Expo*, pages 934–937, 2009. 1, 7