Improving the Visual Comprehension of Point Sets

Sagi Katz
Technion
sagi.katz@gmail.com

Ayellet Tal
Technion
ayellet@ee.technion.ac.il

Abstract

Point sets are the standard output of many 3D scanning systems and depth cameras. Presenting the set of points as is, might "hide" the prominent features of the object from which the points are sampled. Our goal is to reduce the number of points in a point set, for improving the visual comprehension from a given viewpoint. This is done by controlling the density of the reduced point set, so as to create bright regions (low density) and dark regions (high density), producing an effect of shading. This data reduction is achieved by leveraging a limitation of a solution to the classical problem of determining visibility from a viewpoint. In addition, we introduce a new dual problem, for determining visibility of a point from infinity, and show how a limitation of its solution can be leveraged in a similar way.

1. Introduction

The last decade has witnessed a massive growth in the availability of range imaging and 3D scanning devices, from high-end industrial devices to low-end consumer market devices. These devices inherently produce 3D point sets from which, in some cases, 3D surfaces are reconstructed. In other cases, the point sets are left in their original form, for further processing or analysis.

2D images, by definition, contain data viewed from a specific position and therefore, in some manner, reflect human vision. Conversely, 3D point sets are usually produced from multiple views and do not follow human vision. Therefore, for human observers, point sets are often difficult to comprehend, as demonstrated by looking at the rocking-horse in Figure 1(a), which was produced using a scanner. Why is this set difficult to grasp? Obviously, one reason is that when drawing all the points, both the fore and the back points are visible, which is confusing. Solving this problem by removing the back points is helpful, but some problems still remain (Figure 1(b)). First, there are still too many points, which "hide" the subtle features, such as the saddle and the harness. Second, due to different sampling densities, false edges appear, e.g., on the neck of the horse.

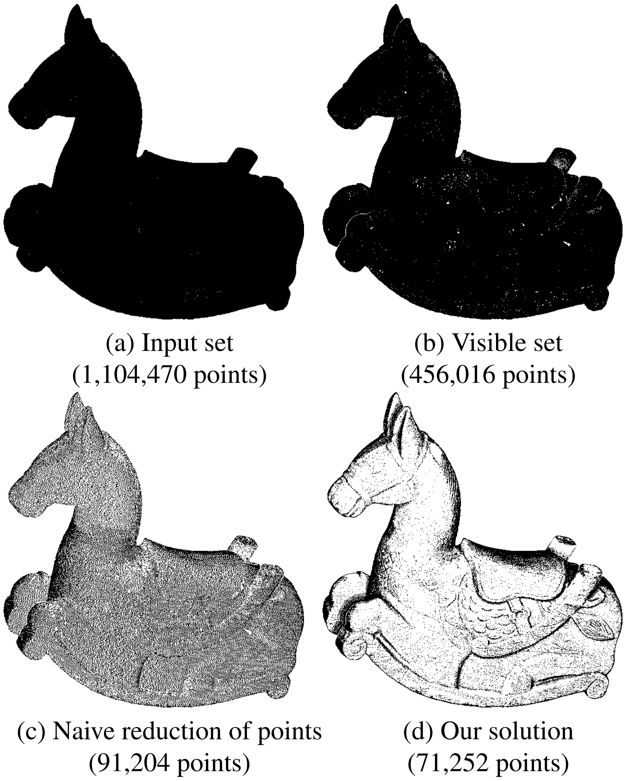
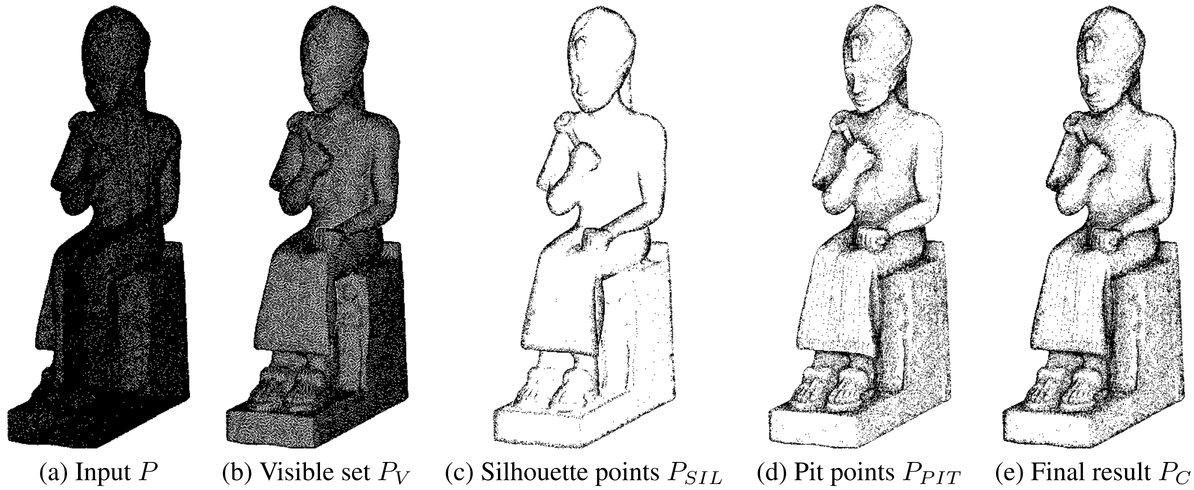

(a) Input set (1,104,470 points)
(b) Visible set (456,016 points)
(c) Naive reduction of points (91,204 points)
(d) Our solution (71,252 points)

Figure 1. **Extracting a subset of points that enhance visual comprehension.** After removing the hidden points from (a), only the visible points are presented in (b). This set is still too dense for the fine features to be noticeable. Drawing every 5^{th} visible point (c) results in loss of the features. In (d), the result of our solution is shown: points on the silhouette and on subtle features, such as on the leaf-shaped relief on the horse's back, are maintained.

Our goal is to perform reduction of points, as viewed from a given viewpoint, such that the object's fine features become apparent, thus improving the visual comprehension of the entire object, as demonstrated in Figure 1(d). The regions that are desired to appear bright, should have a lower density than regions that are desired to appear dark. The question is how we can characterize these regions and determine the appropriate subset of points.

Data compression was extensively discussed for im-



(a) Input P (b) Visible set P_V (c) Silhouette points P_{SIL} (d) Pit points P_{PIT} (e) Final result P_C

Figure 2. **Algorithm outline.** Given a point set P (193,252 points), the visible set of points P_V is extracted. Then, silhouette points P_{SIL} and pit points P_{PIT} are extracted as subsets of P_V. Finally, P_{SIL} and P_{PIT} are combined to create our result (30,266 points).

ages [10, 14] and videos [11]. While the goal there is mostly to maintain the visual quality, our goal is to improve it. Recently, several methods were suggested for point cloud simplification. Some aim at accelerating surface reconstruction, by first reducing the number of input points, and then performing reconstruction on the reduced set [3, 6]. Other methods aim at reducing the data for further processing, such as smoothing or modeling [7, 9]. Their results do not reveal some fine features of the 3D shapes. In [8], in addition to the visible points [5], silhouette points, extracted by local reconstruction, are rendered. This method is highly sensitive to noise. In contrast, our method improves the visual perception not only around the silhouettes, but also in internal regions. Moreover, we benefit from slight noise, such as depth measurement noise.

Our algorithm is based on two key ideas. First, we introduce the dual problem to the classical *Hidden-Point Removal (HPR)* problem, termed the *Target-Point Occlusion* (TPO) problem. In both cases, we are given a set of points and one specific point. HPR aims at detecting the visible (or hidden) points, seen from the viewpoint. Conversely, in TPO the goal is to find the subset of points that are visible to outside observers, positioned along the lines that connect these points with a given target point. These points are considered occluders of the target point. While HPR was solved in [5], this paper suggests a solution to TPO.

Second, we show that, interestingly, the false negatives of the operators that approximate HPR and TPO, are the subset of points we wish to preserve for achieving good visual comprehension of the sampled shape. In particular, we show how to extract points that belong to locally-concave regions by determining false negatives of the HPR operator. Similarly, we extract silhouette points by using false negatives of the TPO operator.

The contribution of this paper is hence threefold. First and foremost, we present a novel algorithm for meaningful view-dependent data reduction of point sets. Our algorithm is inherently simple and easy to implement, and is flexible in its ability to handle points in various dimensions. Moreover, we show how to handle both general point sets and range data. Second, we introduce the problem of target-point occlusion and present an operator that solves it in the limit and approximates it in the finite case. Finally, we suggest an algorithm for detecting the silhouettes of an object directly from the point cloud.

2. Algorithm Outline

Given a point set P and a viewpoint C, our algorithm aims at selecting a subset of P, which preserves the features and the patterns of the object (Figure 2). The naive approach of reducing the number of points by taking every k^{th} point, blurs the silhouettes and the fine features, as it treats all the points similarly (Figure 1(c)). Instead, we are inspired by the stippling-style art form, which simulates varying degrees of shading, by controlling the density of the points. Dense regions appear dark, whereas sparse regions appear bright.

Formally, we seek a visually-comprehensible subset P_C of the visible points P_V, $P_C \subseteq P_V \subseteq P$. In order to produce a compelling subset, we require P_C to have the following two properties: First, in order to highlight the outline of the object, the likelihood of a point $p_i \in P_V$ to belong to P_C should be proportional to the angle between the surface normal and the line-of-sight. For instance, points on the silhouette should belong to P_C. Second, to enhance the shading effect on the viewed point set, the likelihood of a point $p_i \in P_V$ to belong to P_C should be large if it belongs to a deep concavity and small if it belongs to a locally-convex region. This is based on the observation that deep concavities tend to be less illuminated, compared to convex regions, due to a reduced amount of light that reaches them.

Our algorithm is based on two dual operators. The first operator is the HPR operator of [5] (Section 3), used for approximating the set of visible points, P_V, from a viewpoint. The second operator is used for approximating the set of occluding points from infinity (Section 4). Both operators have a curvature-related limitation, which is leveraged for extracting the following two subsets (Section 5): $P_{SIL} \subset P_V$ is the set of points that are located near silhouette regions, as viewed from the given viewpoint. $P_{PIT} \subset P_V$ is the set of points that are located near concave regions, as viewed from the viewpoint. Finally, we define the comprehensible set P_C to be the union of these two subsets (Section 6). Hence, P_C contains points that are either near silhouette regions or near concave regions, as desired:

$$P_C = P_{PIT} \cup P_{SIL}.$$

3. The Hidden-Point Removal (HPR) Operator

Given a set of points P and a viewpoint C, the goal is to find the set of points P_V, which would be visible if the surface from which the points were sampled, were known. Determining the visibility of a point set is an intriguing problem, as points cannot occlude each other. The most common way to compute visibility is therefore to reconstruct the surface [2] and determine visibility on it. Reconstruction, however, is a difficult problem, both theoretically and practically. Early attempts to address the problem directly from the set, skipping reconstruction, were performed during ray tracing. Since both rays and points are singular primitives, the algorithms assume either "thick" rays [13] or "finite-area" points [4, 12]. Unfortunately, assumptions need to be made regarding the thickness of the rays or the area of the points and normals must be estimated.

Katz et al. [5] address this problem regardless of rendering and without normal estimation. Instead, an elegant operator, the *Hidden Point Removal (HPR)* operator, is proposed, which is supported by theoretical guarantees. The HPR operator consists of two stages:

1. **Inversion:** An inversion function maps every point $p_i \in P$ to an inverted domain. Many inversion functions are possible and in [5] $F(p_i) = p_i + 2(R - ||p_i||)\frac{p_i}{||p_i||}$ is used, where R is a user parameter.

For our problem, we found the following function to be advantageous. Assuming, without loss of generality, that C is at the origin, the inversion function is defined as

$$F_\gamma(p_i) = \begin{cases} p_i\|p_i\|^{\gamma-1}, & p_i \neq 0 \\ 0 & p_i = 0 \end{cases}, \qquad (1)$$

where $\gamma < 0$ is a parameter.

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

The main result of [5] is that the points that reside on the convex hull of Step 2 are the images of the visible points. Intuitively, "how much" a point is visible, depends on the size of the empty region that lies between the point and the viewpoint. The larger the size, the "more visible" a point is. It is proved that points that fall on the convex hull in the inverted domain are those that are associated with large empty regions.

While the HPR operator is powerful, deep concavities in the object result in *false negatives* (i.e., points that should be visible but are not detected as such by the operator). This can be explained by the following lemma, which relates between the correct detection of the visible points to the curvature κ, the distance between the surface and the viewpoint d, and the angle β between the surface normal and the line of sight.

Lemma 3.1 Point $p \in \{HPR_\gamma(P)\}$ if p is visible and the curvature κ at p satisfies:

$$\kappa < \frac{\gamma(1-\gamma)\sin(\beta)(\cos^2(\beta) - \gamma\sin^2(\beta))}{d(\gamma^2\sin^2(\beta) + \cos^2(\beta))^{\frac{3}{2}}}.$$

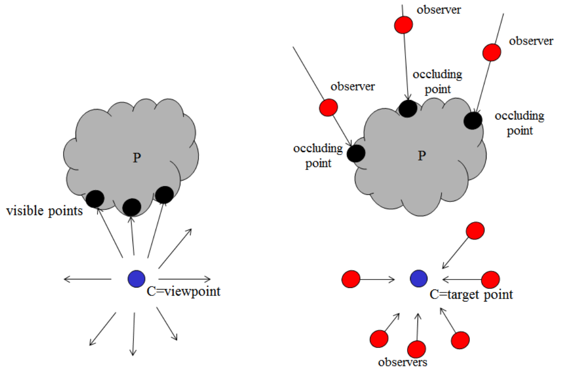
This lemma, whose proof for the inversion function that we use, is given in the appendix, implies that the HPR operator cannot resolve visibility in regions of high curvature. It is guaranteed to detect the points correctly only if the curvature κ is below a threshold, i.e., the local surface is convex or the local surface is concave and the curvature is small enough. In this paper, we show how to benefit from this limitation, in order to extract our desired subset of points.

4. The Target-Point Occlusion (TPO) Operator

This section introduces a dual problem to the from-point visibility, namely the *Target-Point Occlusion* (TPO). Given a set of points P and a target point C, the goal is to find the subset of P, which would occlude C from outside observers positioned at ∞, if the surface from which the points were sampled, existed.

Figure 3 illustrates the difference between the two problems. In the HPR problem, the rays coming out of viewpoint C, intersect the visible points. In the TPO problem, on the other hand, the rays come towards C from infinity and intersect the *occluding points* before reaching the target point. Hence, each of these points occludes C from an observer, which is located in the direction from which the ray is coming, but farther than the occluding point. Obviously, when a ray reaches C, the sample point on the ray is not occluded by any other point in this direction.

Definition 4.1 Occluding point: *Given a point set P and a target point C, a point $p_i \in P$ is an occluding point to C,*



(a) From-point visibility (b) Target-point occlusion

Figure 3. **From-point visibility vs. target-point occlusion.** We are given a point set P (sampled from the gray surface) and a point C (blue). The black points are the visible points from C in (a) and the points occluding C from observers at infinity in (b).

if p_i is visible to some observer located in $O = C + t\frac{p_i - C}{|p_i - C|}$, where $t > \sup_{p_j \in P}\langle p_j - C, \frac{p_i - C}{|p_i - C|}\rangle$.

To approximate the solution to the target-point occlusion problem, we present an operator which, similarly to the HPR operator, is composed of two stages: (1) point transformation and (2) convex hull computation. Surprisingly, the only modification needed is changing the parameter of the transformation function we utilize. In particular, in Equation (1), we use $\gamma < 0$ for from-point visibility and $0 < \gamma < 1$ for target-point occlusion; see Figure 4.

The correctness of this operator is expressed in the following lemmas, whose proofs are provided in the supplementary material. Specifically, we show that in the limit, when the point set is an infinite sampling of the surface, the operator extracts the occluding subset accurately.

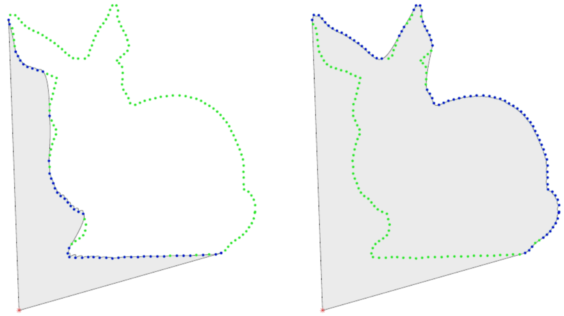
Let P be an infinite sampling of a surface S, $O \subseteq S$ be the set of points that occlude the target point C (i.e., the ground truth), and $TPO_\gamma(P) \subseteq S$ be the set of points extracted by our operator.

Lemma 4.1 *$TPO_\gamma(P) \subseteq O$, i.e., every point p marked occluding by the operator is indeed an occluding point.*

Lemma 4.2 *When $\gamma \rightarrow {}_{+}0$, the set of points marked by the TPO operator, is equal to the set of occluding points.*

Intuitively, Lemma 4.1 suggests that in the limit, the number of false positives of the operator is 0. Lemma 4.2 adds that the number of false negatives is also 0 when $\gamma \rightarrow {}_{+}0$. (The supplementary material also proves guarantees for the case of finite sampling.)

We are particularly interested in understanding the locations where false negatives appear. The following Lemma, which is the equivalent of Lemma 3.1, addresses it.



(a) From-point visibility, $\gamma = -0.01$ (b) Target point occlusion, $\gamma = 0.01$

Figure 4. **Results for from-point visibility & target-point occlusion for a 2D point set.** Blue points were detected visible (occluding), while green points were detected as invisible (non-occluding). The red point is the viewpoint (target point).

Lemma 4.3 *$p \in \{TPO_\gamma(P)\}$ if p is visible to an observer positioned on the ray p-C, farther away than any point in P, and the curvature κ at p satisfies*

$$\kappa > \frac{\gamma(1-\gamma)\sin(\beta)(\cos^2(\beta) - \gamma\sin^2(\beta))}{d(\gamma^2\sin^2(\beta) + \cos^2(\beta))^{\frac{3}{2}}}.$$

Therefore, while the HPR operator is limited in its ability to correctly identify visible points around locally-concave regions, in respect to the viewpoint, the TPO is limited in its ability to correctly identify occluding points around locally-convex regions. These dual limitations are the basis for our point reduction method.

5. Controlled Reduction of Points

Our goal is to extract a subset of the visible points, which satisfies the two requirements discussed in Section 2. Recall that we look for (1) points at locally-concave regions, which create a natural shading effect, and (2) points around the silhouettes regions, which by definition, have normals perpendicular to the line of sight. We note that the latter tend to belong to locally-convex regions.

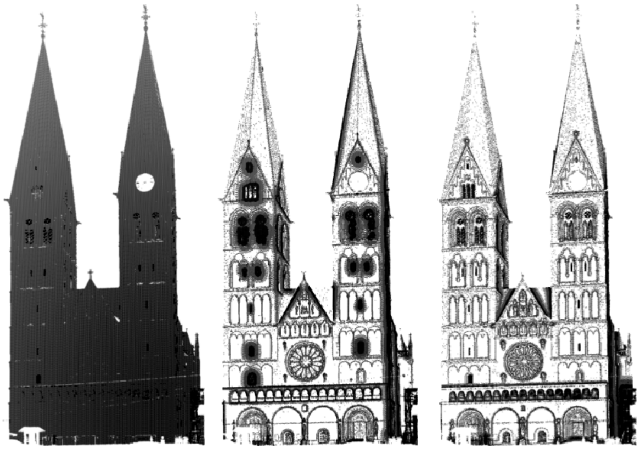
Recall Lemma 3.1, which states that when applying the HPR operator with $\gamma < 0$, the operator will produce false negatives around locally-concave regions. But these are precisely the points we seek after. We term these false negatives as *pit points* and define them as:

$$P_{pits} = P_V \setminus HPR_{\gamma_p}(P). \tag{2}$$

Similarly, from Lemma 4.3, we learn that the TPO operator will produce false negatives around locally-convex regions. These false negatives are the *tip points*, defined as:

$$P_{tips} = P_V \setminus TPO_{\gamma_t}(P_V). \tag{3}$$

Since, as illustrated in Figure 4, the tip points need not necessarily be a subset of the visible points (e.g., the back



(a) Input range data (b) Tip points (c) Pit points

Figure 5. **Tip & pit points for range data.** These points allow easy comprehension of the object.

of the bunny), we apply the TPO operator on the set of visible points, rather than on the entire point set. Recall from Lemma 4.3 that the curvature bound depends also on β, the angle between the tangent and the line-of-sight. Since we wish to extract points with a small β (i.e., silhouettes), we use γ_t that is close to 0. We can now define $P_{sils} = P_{tips}$.

The problem with the realization of Equations (2)-(3) is that P_V is unknown. Below, we propose solutions both for range data and for generic point clouds.

The case of range data: Range data, generated by range scanners, are samples of distances from the camera to the scene being scanned. These samples are represented similarly to images, in a 2D matrix, adding the z coordinates.

Due to the fact that this data consists of samples that are seen from a given viewpoint (scanner), it provides the visible point set from the scanner location. Therefore, in this case $P_V = P$ and the pit points are simply

$$P_{pits} = P \setminus HPR_{\gamma_p}(P).$$

Similarly, the set of silhouette points is defined as

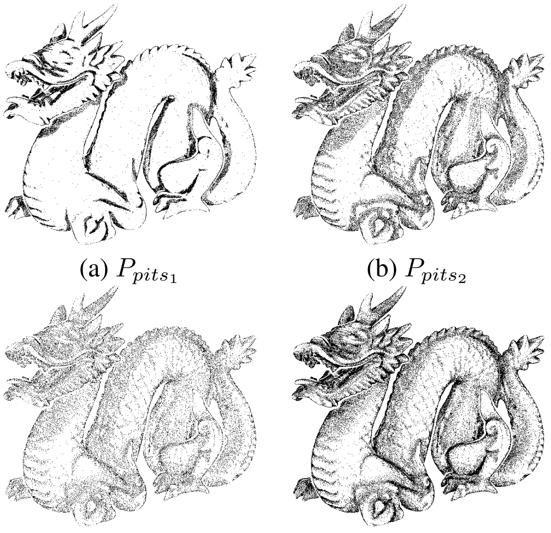
$$P_{sils} = P \setminus TPO_{\gamma_t}(P).$$

Figure 5 shows the pit points and the tip points for a range data set, for $\gamma_t = -\gamma_p$, which are points on locally-concave regions and on locally-convex regions, respectively. While the details of the facade of the church cannot be identified in the input, they are clearly perceived when rendering only the tip or only the pit points. For all the other results shown in this paper, $|\gamma_t| << |\gamma_p|$, so that only convex points near the silhouettes are extracted.

The case of generic point clouds: In this case, we do not have available the ground truth of P_V. We therefore need to approximate it. Fortunately, such approximation exists—it is the result of applying the HPR operator on a point set. We therefore apply the HPR operator twice, with different parameters and define the set of pit points as

$$P_{pits} = HPR_{\gamma_v}(P) \setminus HPR_{\gamma_p}(P),$$

where $\gamma_p < \gamma_v < 0$. Hence, points that belong to P_{pits} pass the curvature threshold for γ_v, but not the threshold for γ_p.



(a) P_{pits_1} (b) P_{pits_2}

(c) P_{pits_3} (d) $P_{PIT} = \bigcup_{i=1}^{i=3} P_{pits_i}$

Figure 6. **The final set of pit points**P_{PIT}**.** The union of three sets of pit points creates an appearance of a gradual change.

In order to control the sensitivity of the pit point extraction, we define *a sensitivity parameter* $e_p = \frac{\gamma_v}{\gamma_p}$. The set of pit points is now reformulated as

$$P_{pits}(P, \gamma_v, e_p) = HPR_{\gamma_v}(P) \setminus HPR_{\frac{\gamma_v}{e_p}}(P). \tag{4}$$

When $e_p \rightarrow 1$, less and less points are considered as pit points, and when $e_p = 1$, $P_{pits} = \emptyset$. Figures 6(a)-(c) show the pit points extracted for different values of e_p.

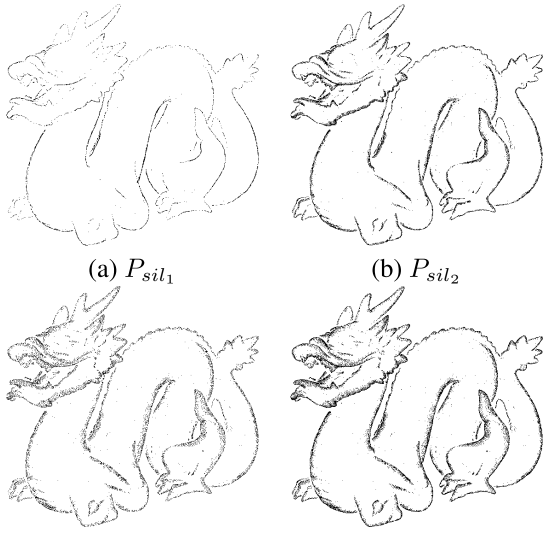
Similarly, changing P_V to $HPR_{\gamma_v}(P)$ in Equation 2, results with a method to extract point in the silhouette regions:

$$P_{sil}(P, \gamma_v, e_t) = HPR_{\gamma_v}(P) \setminus TPO_{-\frac{\gamma_v}{e_t}}(HPR_{\gamma_v}(P)).$$

6. Producing the Visually-Comprehensible Set

In the last step of the algorithm, we wish to find the subset of points, P_C, which is not only much smaller than the input set, but also visually “describes” the object in an eye-pleasing manner. One way to achieve this is to create a visual effect of soft illumination. For this, we would like the density of the points to gradually change from being sparse in “lit“ regions to being dense in “shaded“ regions.

As illustrated in Figure 6(a)-(c), the pit points create a rather sharp transition between regions that are densely populated with points and regions that are sparse. To achieve



(a) P_{sil_1} (b) P_{sil_2}

(c) P_{sil_3} (d) $P_{SIL} = \bigcup_{i=1}^{i=3} P_{sil_i}$

Figure 7. **The final set of silhouette points P_{SIL}.** The union of three sets of pit points creates an appearance of gradual change.

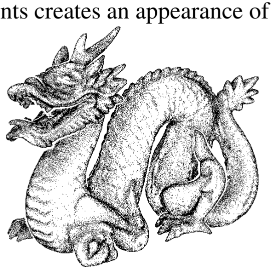


Figure 8. **The final result** $P_C = P_{PIT} \cup P_{SIL}$.

the gradual density change effect, we create a cascade of N pit point sets, each is calculated with a different parameter e and has a different tone level, as explained hereafter.

Let $E = \{e_1, e_2 \cdots e_N\}$ be a vector of parameters to Equation (4) and $Tone = \{tone_1, tone_2 \cdots tone_N\}$ be a vector that controls the relative level of tone for each level of sensitivity. Intuitively, E ”thresholds” the curvature of the points considered, whereas $Tone$ represents the ”darkness” for each level.

We define the pit points at level j to be

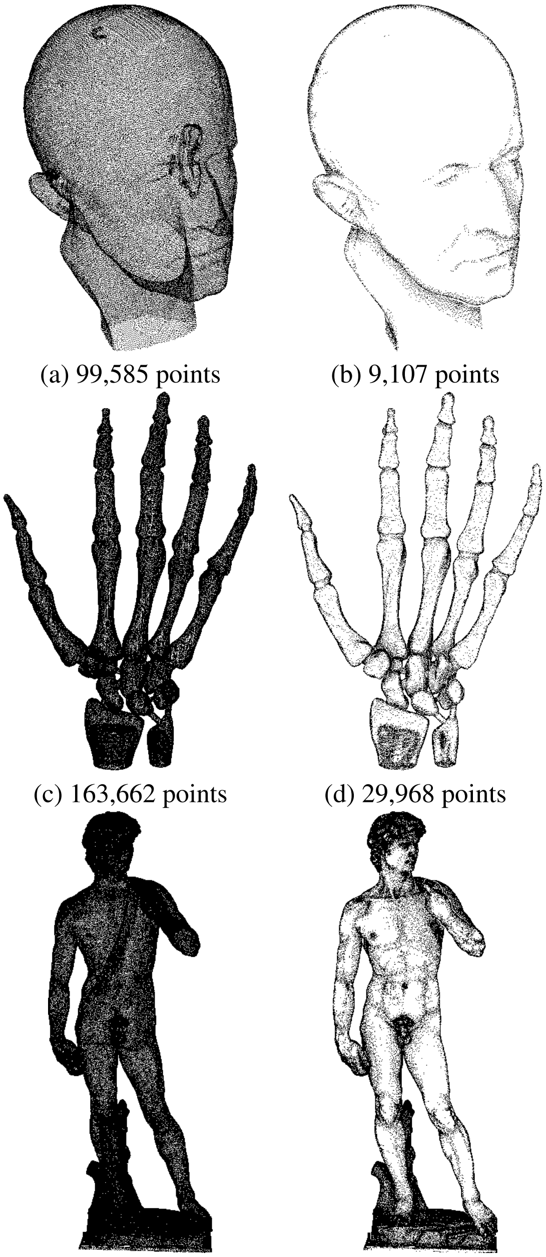
$$P_{pits_j} = \{p_i | p_i \in P_{pits}(P, \gamma_v, e_j), t_i \leq tone_j\}, j = 1...N.$$

Here, t_i is a random variable sampled uniformly from the range $[0, 1]$. If $tone_j = 1$, then all the points are taken for level j. If $tone_j = 0.5$, then roughly half of the points in $P_{pits}(P, \gamma, e_j)$ will belong to the final set. In general, for $tone_j = \alpha$, α of the points are drawn.

The final set of pit point P_{PIT} is defined as:

$$P_{PIT} = \{p_i | p_i \in \bigcup_j P_{pits_j}\}.$$

Figure 6 demonstrates that the fine features, such as the “diamond-shaped” texture, become apparent and the transitions between dark and bright regions are gradual.



(a) 99,585 points (b) 9,107 points

(c) 163,662 points (d) 29,968 points

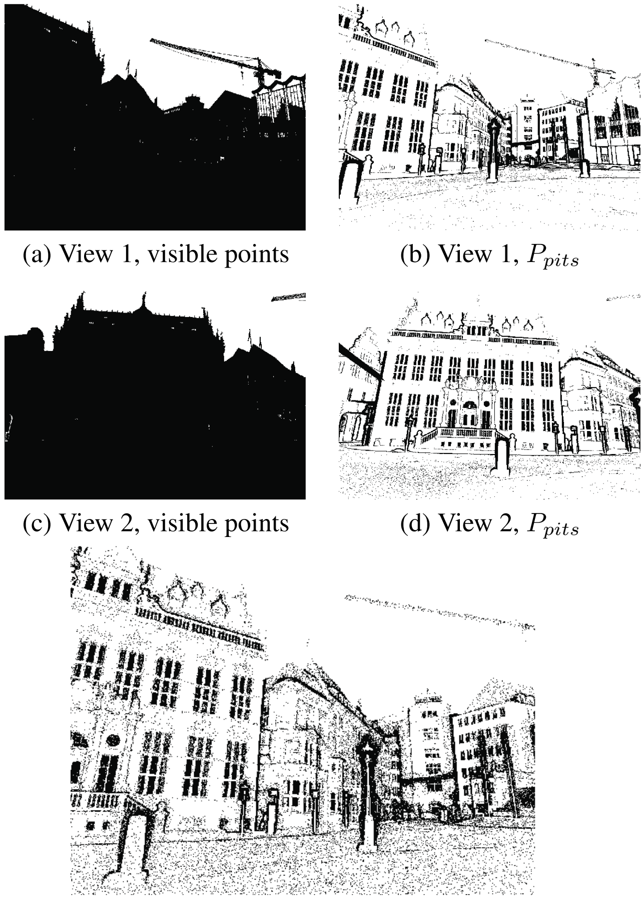
(c) 253,716 points (d) 72,250 points

Figure 9. **Results.** Our results (right) not only improve the visual comprehension of the objects (left), but are also eye-pleasing.

The set P_{SIL} is produced similarly, and creates “roundness” near the silhouette (Figure 7). Recall that the final set is defined as (Figure 8): $P_C = P_{PIT} \cup P_{SIL}$.

7. Results

This section shows additional results. In all the examples, we use the following default parameters: $N = 3$, $\gamma_v = -0.00001$, $TONE = (1, 0.5, 0.25)$, $E = (0.01, 0.002, 0.001)$. Figure 9 shows results for several



(a) View 1, visible points

(b) View 1, P_{pits}

(c) View 2, visible points

(d) View 2, P_{pits}

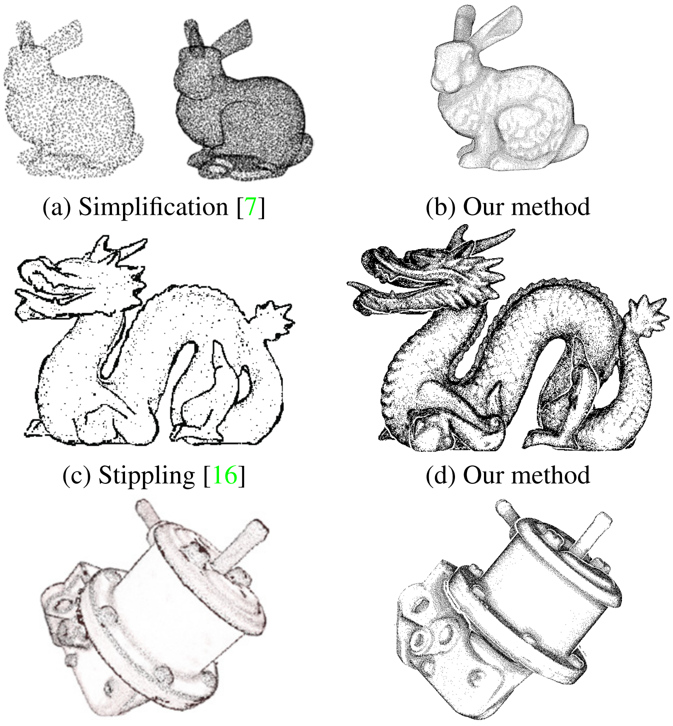
(e) P_{PIT}

Figure 10. **Result on range data consisting of 15 million points.** While the presentation of the whole set of points is not informative, our reduced set provides a visually descriptive view.

generic point sets. Our method reduces the number of points considerably, while not only improving the visual comprehension of the objects, but also producing aesthetic drawings. In particular, fine features, such as the facial features, have become more apparent.

Figure 10 shows our result for range data from [1], which was scanned by a Riegl VZ-400 laser scanner. It contains 15 million samples captured from a single point. It is easy to see that while the input data (left) is too dense to understand, the reduced sets make it possible to comprehend.

Comparisons: In Figure 1, we compare our work to the straightforward solution of removing every k^{th} point, and show that our method maintains the fine features and textures. Figure 11(a)-(b) demonstrates a similar effect, when comparing our work to the simplification method of [7]. In Figure 11(c)-(d) we compare our results to stippling of point sets [16]. Though stippling has a somewhat different goal, it also aims at rendering points in an aesthetic manner [15, 16]. It is evident that our technique manages to preserve the subtle details, such as the 3D texture on the dragon's side of the body. Figure 11(e)-(f) compares our result to that of [8], where the goal is quite similar to ours—direct visualization of point sets. Their method produces very nice results, however our method creates a self-shadowing effect, which improves depth perception. In addition, in contrast to [8], our method can be applied on noisy point sets and does not render all the visible points, which may not be useful if the set is too large.



(a) Simplification [7]

(b) Our method

(c) Stippling [16]

(d) Our method

(e) Result of [8]

(f) Our method

Figure 11. **Comparison to previous methods.** We manage to maintain the 3D texture compared to [16]. While [8] uses points and lines for visualization, our method requires only points. In addition, our method creates an effect of self-shadowing, which improves the depth perception.

Limitations: Our current Matlab implementation does not allow real-time rendering of very large point sets (7 seconds on 500,000 points). The bottleneck of the method is the convex hull computation ($O(n \log n)$). In the future, it would be possible to significantly improve the run-time performance, using GPU-accelerated convex hull construction.

8. Conclusion

This paper has presented an algorithm for view-dependent data reduction of a point set, which improves the visual comprehension of the sampled object. The utility of the algorithm was demonstrated by producing compelling results on a variety of point sets—both unorganized sets and range data. Our algorithm neither reconstructs the surface, nor re-samples the data or estimates normals. Instead, it works directly on the input set. The algorithm is easy to implement, while supported by theoretical guarantees. The resulting set can be utilized in various analysis problems.

Our algorithm is based on two key ideas. First, we discuss and analyze two dual visibility problems: the classical from-point visibility and the new target-point occlusion. We utilize the HPR operator to approximate the from-point visibility and propose an operator that approximates the solution to the target-point occlusion problem. Second, we show how exploiting the limitations of these operators—the fact that their correctness depends on the local curvature—lets us define a subset of points, whose presentation improves the perception of the given point cloud.

As byproducts, our paper presents algorithms for both directly computing occlusion and extracting silhouettes of point sets, while avoiding surface reconstruction. These algorithms are important in their own right in computer vision and in computer graphics.

Acknowledgements: This research was supported by the Israel Science Foundation (ISF) 1420/12, the Argentinian Research Fund, and the Ollendorff Foundation. We thank aim@shape repository & Stanford CG Lab for their models.

References

[1] D. Borrmann and J. Elseberg. http://kos.informatik.uni-osnabrueck.de/3dscans/. 7

[2] T. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge, 2006. 3

[3] T. K. Dey, J. Giesen, and J. Hudson. Decimating samples for mesh simplification. In *CCCG*, pages 85–88, 2001. 2

[4] G. Guennebaud, L. Barthe, and M. Paulin. Deferred splatting. *Comput. Graph. Forum*, 23(3):653–660, 2004. 3

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

[6] P.-F. Lee and B.-S. Jong. Point-based simplification algorithm. *WSEAS Trans. Comp. Res.*, 3(1):61–66, 2008. 2

[7] C. Moenning and N. Dodgson. A new point cloud simplification algorithm. In *VIIP*, 2003. 2, 7

[8] M. Olson, R. Dyer, H. Zhang, and A. Sheffer. Point set silhouettes via local reconstruction. *Computers & Graphics*, 35(3):500–509, 2011. 2, 7

[9] M. Pauly, M. Gross, and L. Kobbelt. Efficient simplification of point-sampled surfaces. In *VIS*, pages 163–170, 2002. 2

[10] W. B. Pennebaker and J. L. Mitchell. *JPEG still image data compression standard*. Springer, 1993. 2

[11] I. E. Richardson. *The H.264 advanced video compression standard*. Wiley, 2010. 2

[12] S. Rusinkiewicz and M. Levoy. Qsplat: A multiresolution point rendering system for large meshes. In *SIGGRAPH*, pages 343–352, July 2000. 3

[13] G. Schaufler and H. Jensen. Ray tracing point sampled geometry. In *Rendering Techniques*, pages 319–328, 2000. 3

[14] D. Taubman and M. W. Marcellin. *JPEG2000 image compression fundamentals, standard and practice*. Kluwer Academic Publishers, 2004. 2

[15] H. Xu and B. Chen. Stylized rendering of 3D scanned real-world environments. In *NPAR*, 2004. 7

[16] N. Zakaria and H.-P. Seidel. Interactive stylized silhouette for point-sampled geometry. In *GRAPHITE*, 2004. 7

A. Proof of Lemma 3.1

Proof: Let $\widehat{p} = F(p) = (\widehat{x}, \widehat{y})$ denote the image of p and $\widehat{\Lambda}$ be the line through $\widehat{p}$ along the convex hull. Without loss of generality, we define a coordinate system as follows (Fig. 12): The vantage point C is at the origin; the Y-axis is parallel to $\widehat{\Lambda}$; the X-axis is perpendicular to the Y-axis. $p = (x, y)$ in Euclidean coordinates and (d, θ) in polar coordinates. Since $\theta = \tan^{-1}(y/x)$, then $\theta = \beta - \pi/2$. Thus,

$$(x, y) = \left(\cos(\theta)(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}}, \sin(\theta)(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}} \right).$$

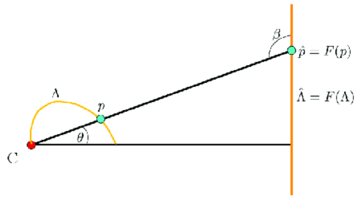


Figure 12. The figure shows line $\widehat{\Lambda}$ parallel to the Y-axis and its inversion source Λ through p. The curvature κ_Λ of Λ at p is the maximal curvature beyond which p is marked invisible by the HPR.

We are interested in curve Λ, which is the inversion source of $\widehat{\Lambda}$, and in its curvature κ_Λ. p will be marked visible if the curvature κ at p is smaller than κ_Λ and marked hidden otherwise. Varying x and y along Λ and taking their derivatives with respect to θ yields:

$$\begin{aligned}
\dot{x} &= -\frac{(\gamma-1)}{\gamma}\sin(\theta)(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}}, \\
\dot{y} &= (\cos(\theta) - \frac{\cos^2(\theta)-1}{\gamma\cos(\theta)})(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}}, \\
\ddot{x} &= \frac{(1-\gamma)(\gamma\cos^2(\theta)-\cos^2(\theta)+1)}{\gamma^2\cos(\theta)}(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}}, \\
\ddot{y} &= \frac{\sin(\theta)(\gamma+1-\gamma^2\cos^2(\theta)+2\gamma\cos^2(\theta)-\cos^2(\theta)}{\gamma^2\cos^2(\theta)}(\frac{\widehat{x}}{\cos(\theta)})^{\frac{1}{\gamma}}.
\end{aligned}$$

Using the definition of the curvature $\frac{\dot{x}\ddot{y}-\dot{y}\ddot{x}}{(\dot{x}^2+\dot{y}^2)^{3/2}}$,

$$\kappa_\Lambda = \frac{(\frac{\hat{x}}{\cos(\theta)})^{\frac{2}{\gamma}}(\gamma-1)(\cos(\theta)^2+\gamma\cos(\theta)^2-1)}{\gamma^2\cos(\theta)^2\left(\frac{\sin(\theta)^2(\frac{\hat{x}}{\cos(\theta)})^{\frac{2}{\gamma}}(\gamma-1)^2}{\gamma^2}+\frac{(\frac{\hat{x}}{\cos(\theta)})^{\frac{2}{\gamma}}(\gamma\cos(\theta)^2-\cos(\theta)^2+1)^2}{\gamma^2\cos(\theta)^2}\right)^{\frac{3}{2}}}.$$

Expressing κ_γ in terms of d and β, using the identities $\beta = \frac{\pi}{2} + \theta$, $x = d\cos\theta = d\sin\beta$, $y = d\sin\theta = -d\cos\beta$, and $\widehat{x} = d^\gamma \sin\beta$, we obtain

$$\kappa_\Lambda = \frac{\gamma(1-\gamma)\sin(\beta)(\cos^2(\beta)-\gamma\sin^2(\beta))}{d(\gamma^2\sin^2(\beta)+\cos^2(\beta))^{\frac{3}{2}}}. \square$$