

ORDER-INDEPENDENT SEQUENTIAL THINNING IN ARBITRARY DIMENSIONS

Péter Kardos

Department of Image Processing and Computer Graphics
University of Szeged
Szeged, Hungary
email: pkardos@inf.u-szeged.hu

Kálmán Palágyi

Department of Image Processing and Computer Graphics
University of Szeged
Szeged, Hungary
email: palagyi@inf.u-szeged.hu

ABSTRACT

Skeletons are region based shape descriptors that play important role in shape representation. This paper introduces a novel sequential thinning approach for n -dimensional binary objects ($n = 1, 2, 3, \dots$). Its main strength lies in its order-independency, i.e., it can produce the same skeletons for any visiting orders of border points. Furthermore, this is the first scheme in this field that is also applicable for higher dimensions.

KEY WORDS

image representation, skeleton, sequential thinning, order-independency, digital topology

1 Introduction

Shape representation is fundamental in numerous applications in image processing and pattern recognition. Skeleton is an often applied shape feature, which summarizes the general form of objects [18]. Thinning serves as a traditional strategy for skeletonization that is based on an iterative peeling of objects [19]. Several parallel and sequential alternatives have been proposed for this method. Parallel thinning processes are composed of parallel reduction operations that delete all points satisfying their conditions simultaneously. Sequential thinning algorithms delete just one object point at a time. In both strategies the concept of simple points plays a key role. An object point is said to be a *simple point*, if its deletion (i.e., changing it to background point) preserves the topology [11]. In 2D, it means that no object is split nor completely deleted, no cavity is merged with the background nor another cavity, and no cavity is created [9]. There is an additional concept called hole in 3D binary pictures. Topology preservation implies that eliminating or creating any hole is not allowed. Using this notion, we consider Algorithm 1 as the general two-phase scheme for sequential thinning.

The basic criterion for a “deletable” point is usually that it must be a simple point but not a so-called *endpoint* (since preserving endpoints provides important geometrical information relative to the shape of the objects). The most important advantage of sequential thinning methods over parallel ones is that topology preservation can be easily ensured by removing only one simple point at a time

Algorithm 1:

```
repeat
  // Phase 1: contour tracking
  mark all simple points
  // Phase 2: reduction
  foreach marked point  $p$  do
    if  $p$  is “deletable” in the actual image then
      delete  $p$ 
until no points are deleted
```

[12, 14]. However, sequential algorithms usually suffer from the problem that they may extract various skeletons for different visiting orders of border points. One may say that any parallel thinning algorithm may be implemented as an order-independent sequential one. It holds, but that thinning scheme is quite different than Algorithm 1. In parallel thinning algorithms that are implemented on sequential computers, there is no re-checking of marked points before their deletion. That thinning scheme basically corresponds to Algorithm 2.

Algorithm 2:

```
repeat
  // Phase 1: contour tracking
  mark all “deletable” points
  // Phase 2: reduction
  foreach marked point  $p$  do
    delete  $p$ 
until no points are deleted
```

The first order-independent sequential thinning strategy (i.e., algorithms that produce the same skeletons for any visiting orders) for (8,4) images has been proposed by Ranwez and Soille [17]. Later, Iwanowski and Soille [7] gave solutions for two additional connectivities considering 2D images. The main disadvantage of these attempts lies in the fact that they do not retain endpoints which are important in the view of shape-preservation. This means that, as a pre-processing step, endpoints must be previously detected as anchors (i.e., those detected points are not deleted during the entire object reduction process). The latest so-

lution for order-independency has been developed by Kardos, Németh, and Palágyi [8]. Their method is based on a classification of simple points. For this purpose, simple points are partitioned into four classes in the first phase of an iteration, which means that this solution lies far from the sequential thinning scheme according to Algorithm 1.

In this paper we propose an order-independent thinning method which does not need any pre-processing step to determine endpoints, as an endpoint-criterion is built-in to it. It does not even need to distinguish different kinds of simple points, hence it is much closer to Algorithm 1 than the approach reported in [8]. A further major novelty of our new algorithm is that, unlike the earlier attempts, this can be applied in arbitrary dimensions. It is also important to emphasize that our method is not template-based.

2 Basic Notions and Results

The elements of a binary digital image can be considered as a set of points in the n -dimensional digital space denoted by \mathbb{Z}^n ($n = 1, 2, 3, \dots$). Two points $p = (p_1, \dots, p_n)$ and $q = (q_1, \dots, q_n)$ in \mathbb{Z}^n are said to be m -adjacent ($m = 1, \dots, n$) if both of the following conditions are satisfied:

- $|p_i - q_i| \leq 1$ ($i \in \{1, 2, \dots, n\}$),
- $\sum_{i=1}^n |p_i - q_i| \leq m$.

Note that these relations are reflexive and symmetric. In the 2D case, 1- and 2-adjacency are often referred to as 4- and 8-adjacency, and on the 3D cubic grid, 1-, 2-, and 3-adjacency corresponds to the 6-, 18-, and 26-adjacency relations, respectively [11]. Let $N_m(p) = \{q \mid q \text{ is } m\text{-adjacent to } p\}$ and $N_m^*(p) = N_m(p) \setminus \{p\}$.

The sequence $\langle s_0, s_1, \dots, s_k \rangle$ of distinct points is a j -path from point s_0 to point s_k in a non-empty set of points S if each point of the sequence is in S and for all i ($1 \leq i \leq k$) s_i is j -adjacent to s_{i-1} ($j \in \{1, 2, \dots, n\}$). If there is a j -path in S between two points, then we say that those points are j -connected in S . A set of points S is j -connected in the set of points $S' \supseteq S$ if any two points are j -connected in S' .

An n -dimensional $(n, 1)$ binary digital picture is a quadruple $(\mathbb{Z}^n, n, 1, B)$ [11], where $B \subseteq \mathbb{Z}^n$ is the set of black points, which are represented by “1”-s; its complement, $\mathbb{Z}^n \setminus B$ is the set of white points which have the value “0”. A *black component* is a maximal n -connected set of black points, while a *white component* is defined as a maximal 1-connected set of white points [11].

A black point p in the picture $(\mathbb{Z}^n, n, 1, B)$ is called as a *border point*, if $N_1(p) \setminus B \neq \emptyset$. A black point which is not a border point is said to be an *interior point*. It is obvious that any simple point is a border point, and any interior point is non-simple. The simplicity of a point in an $(n, 1)$ picture depends only on its n -neighborhood, hence it is a local property [4, 10, 13].

For shape preservation, thinning algorithms must not remove the endpoints of an object. Hall introduced three possible criteria to determine such points of 2D objects [6]. In 3D thinning, there have been several endpoint-characterizations proposed for extracting both medial surfaces and medial lines [2, 5]. In this paper, we use the following approach:

Definition 1 A black point $p \in B$ in picture $(\mathbb{Z}^n, n, 1, B)$ is an endpoint if there is no interior point in $N_1(p)$.

Here we note that this endpoint characterization has been applied in some existing 2D and 3D thinning algorithms [1, 3, 13, 16]. In 3D, thinning algorithms are capable of extracting two kinds of skeleton-like shape descriptors. Surface-thinning algorithms are to extract medial surfaces by preserving surface-endpoints, and curve-thinning algorithms produce centerlines by preserving curve-endpoints [15]. Note that using endpoint characterization according to Definition 1 yields surface-thinning algorithms in 3D.

3 The Proposed Algorithm

Now we are ready to introduce our new order-independent sequential thinning algorithm denoted by KP-OIST- nD ($n = 1, 2, \dots$). Its pseudo-code is shown in Algorithm 3. The two-phase scheme of the algorithm works as follows. In Phase 1, interior points are collected into the set I , while simple points are put into the set S . In Phase 2, the algorithm scans the collected simple points, and examines whether the actual point satisfies the deletion conditions. Note that it is not determined in what order those points should be visited. For the examinations, the set D is constructed, which contains the “deletable” black points in the actual n -neighborhood of p (i.e., the black points that were simple and non-endpoints in Phase 1 of the given iteration). By examining all subsets of D , we check all possible situations which could occur, if the algorithm would visit the points in $N_n(p)$ in any order. In other words, we examine whether p would be still simple if we would change some of its “deletable” black neighbors to white points. If in any of these possible situations p would not be simple any more, then the removal of p could result in order-dependency, but if p remains simple in all examined cases, then it may be deleted. The algorithm stops if stability is reached.

The effect of the proposed algorithm in 1D, 2D, and 3D is illustrated for three small binary images on Figs. 1-3.

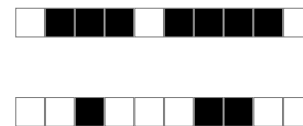


Figure 1. A 1D binary image (top) and its skeleton produced by Algorithm 3 (bottom).

Algorithm 3:

Input: picture $(Z^n, n, 1, X)$
Output: picture $(Z^n, n, 1, Y)$
 $Y = X$
repeat
 // Phase 1: contour tracking
 $I = \emptyset$
 $S = \emptyset$
 foreach p in Y **do**
 if p is an interior point **then** $I = I \cup \{p\}$
 if p is a simple point **then** $S = S \cup \{p\}$
 // Phase 2: reduction
 changed = false
 foreach $p \in S$ **do**
 deletable = true
 if $N_1(p) \cap I = \emptyset$ **then continue**
 $D = \{q \mid q \in N_n^*(p) \cap Y \cap S \text{ and } N_1(q) \cap I \neq \emptyset\}$
 foreach $\Delta \subseteq D$ **do**
 if p is not simple in $(Z^n, n, 1, Y \setminus \Delta)$
 then
 deletable = false
 break
 if deletable = true **then**
 $Y = Y \setminus \{p\}$
 changed = true
until changed = false

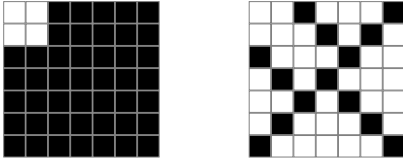


Figure 2. A 2D binary image (left) and its skeleton produced by algorithm KP-OIST-2D (right).

4 Verification

As algorithm KP-OIST- nD removes only one simple point at a time, it is obvious that the proposed method is topology-preserving. However, a more detailed explanation is needed to show that the algorithm is order-independent, as well. For the proof of this property, we will make use of the following proposition.

Proposition 1 *Let $q \in Y$ in the beginning of any iteration of algorithm KP-OIST- nD . Then, $N_1(q) \cap I$ is not changed during the actual iteration.*

Proof. Each point $r \in N_1(q) \cap I$ is an interior point, hence r is not simple. Since only simple points are taken into consideration as deletable points, r cannot be deleted. \square

Theorem 1 *Algorithm KP-OIST- nD is order-independent.*

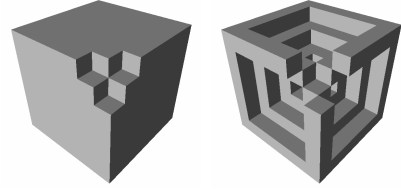


Figure 3. A 3D binary image (left) and its skeleton produced by algorithm KP-OIST-3D (right).

Proof. The proof will be carried out indirectly. Let us suppose that algorithm KP-OIST- nD is not order-independent. Then, there exists at least one point $p \in X$ such that p would be deletable in the beginning of an iteration (i.e., p would be the starting point of the contour tracking), but p is no more deletable, when it is visited later. This can only happen if there is a point $q \in N_n(p) \cap Y$ such that p is deletable in picture $(Z^n, n, 1, Y)$, but p is not deletable in $(Z^n, n, 1, Y \setminus \{q\})$. By Proposition 1, I remains unchanged during the given iteration, hence $N_1(p) \cap I \neq \emptyset$ will hold until the end of that iteration. Therefore, if p is not deletable in $(Z^n, n, 1, Y \setminus \{q\})$, then there must be a set $\Delta \subseteq D \setminus \{q\}$ such that p is not simple in picture $(Z^n, n, 1, (Y \setminus \{q\}) \setminus \Delta)$. However, as $\Delta \cup \{q\} \subseteq D$ and p is deletable in picture $(Z^n, n, 1, Y)$, p must be simple in $(Z^n, n, 1, Y \setminus (\Delta \cup \{q\}))$. It is obvious that $(Y \setminus \{q\}) \setminus \Delta = Y \setminus (\Delta \cup \{q\})$, thus the latter two pictures are the same, and as p can not be simple and non-simple at the same time, we came to a contradiction. \square

5 Discussion and Results

In experiments our method was tested on various 2D and 3D images. In the 2D case, we have also compared the results of our algorithm to the ones produced by the algorithm introduced by Ranwez and Soille [17]. That 2-dimensional thinning algorithm is denoted by RS-OIST-2D. Below, we present some illustrative examples. In Figs. 4-7, “skeletons” generated by the two algorithms under comparison are superimposed on the original objects. Figs. 8-11 show some 3D test images and their medial surfaces produced by our algorithm. Numbers in parentheses mean the count of object points.

If one examines the results for the 2D images, it is safe to state that algorithm KP-OIST- nD does not produce such skeletons with numerous unwanted side branches like algorithm RS-OIST-2D proposed by Ranwez and Soille [17]. As our deletion rule does not refer to any directions around the visited object point, it is obvious that it is invariant under the $k \times 90^\circ$ rotations and reflections ($k = 0, 1, 2, 3$). Due to the considered endpoint-characterization, algorithm KP-OIST- nD preserves 2-point thick segments, therefore it cannot produce 1-point thin skeletons as it is usual in numerous existing thinning algorithms [1, 3, 13, 16].

6 Conclusion

This paper has introduced the very first order-independent thinning method for higher dimensions. As a future work we are planning to extend this scheme in a way that it would be also applicable for some additional endpoint-characterizations. This could, for example, make possible to extract centerlines from 3D tubular objects.

Acknowledgements

This research was supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency, the European Union and the European Regional Development Fund under the grant agreement TÁMOP-4.2.1/B-09/1/KONV-2010-0005, and the grant CNK80370 of the National Office for Research and Technology (NKTH) & the Hungarian Scientific Research Fund (OTKA).

References

- [1] C. Arcelli, G. Sanniti di Baja, L. Serino, New removal operators for surface skeletonization, *Proc. 13th International Conference on Discrete Geometry for Computer Imagery, DGCI 2006, Lecture Notes in Computer Science 4245*, Szeged, Hungary, 2006, 119–135
- [2] G. Bertrand, Z. Aktouf, A 3D thinning algorithm using subfields, *SPIE Proc. of Conf. on Vision Geometry*, San Diego, CA, USA, 1994, 113–124
- [3] G. Bertrand, M. Couprie, Two-dimensional parallel thinning algorithms based on critical kernels, *Journal of Mathematical Imaging and Vision*, 31(1), 2008, 35–56
- [4] M. Couprie, G. Bertrand, New characterizations of simple points in 2D, 3D, and 4D discrete spaces, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4), 2009, 637–648
- [5] W. Gong, G. Bertrand, A simple parallel 3D thinning algorithm, *10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, USA, 1990, 188–190
- [6] R.W. Hall, Parallel connectivity-preserving thinning algorithm, in Kong, T.Y., Rosenfeld, A. (Eds.), *Topological Algorithms for Digital Image Processing, Machine Intelligence and Pattern Recognition*, 19, (Amsterdam, Elsevier Science, 1996) 145–179
- [7] M. Iwanowski, P. Soille, Order independence in binary 2D homotopic thinning, *Proc. 13th International Conference on Discrete Geometry for Computer Imagery, DGCI 2006, Lecture Notes in Computer Science 4245*, Szeged, Hungary, 2006, 592–604
- [8] P. Kardos, G. Németh, K. Palágyi, An order-independent sequential thinning algorithm, *Proc. 13th International Workshop on Combinatorial Image Analysis, IWCIA 2009, Lecture Notes in Computer Science 5852*, Playa del Carmen, Mexico, 2009, 162–175
- [9] T.Y. Kong, On topology preservation in 2-d and 3-d thinning, *International Journal of Pattern Recognition and Artificial Intelligence*, 9, 1995, 813–844
- [10] T.Y. Kong, Topology-preserving deletion of 1's from 2-, 3- and 4-dimensional binary images, *Proc. 7th Int. Workshop Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science 1347*, Montpellier, France, 1997, 3–18
- [11] T.Y. Kong, A. Rosenfeld, Digital topology: Introduction and survey, *Computer Vision, Graphics, and Image Processing*, 48, 1989, 357–393
- [12] P.C.K. Kwok, A thinning algorithm by contour generation, *Communications of the ACM*, 31(11), 1988, 1314–1324
- [13] A. Manzanera, T.M. Bernard, F. Preteux, B. Longuet, A unified mathematical framework for a compact and fully parallel n-D skeletonization procedure, *Vision Geometry VIII (VG'99)*, Denver, USA, 1999, 57–68
- [14] L. Lam, S.-W. Lee, C.Y. Suen, Thinning methodologies – A comprehensive survey, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14, 1992, 869–885
- [15] K. Palágyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm, *Graphical Models and Image Processing*, 61, 1999, 199–221
- [16] K. Palágyi, A 3D fully parallel surface-thinning algorithm, *Theoretical Computer Science*, 406, 2008, 119–135
- [17] V. Ranwez, P. Soille, Order independent homotopic thinning for binary and grey tone anchored skeletons, *Pattern Recognition Letters*, 23(6), 2002, 687–702
- [18] K. Siddiqi, S. Pizer, *Medial Representations: Mathematics, Algorithms and Applications* (New York, Springer, 2008)
- [19] C.Y. Suen, P.S.P. Wang, *Thinning Methodologies for Pattern Recognition, Series in Machine Perception and Artificial Intelligence 8* (Singapore, World Scientific, 1994)

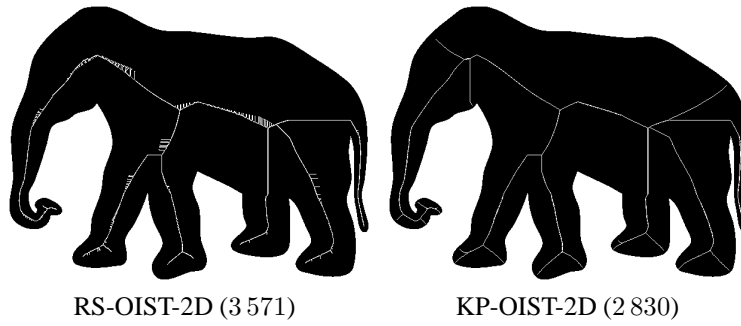


Figure 4. A 612×467 image with 179 293 object points of an elephant and its “skeletons”.

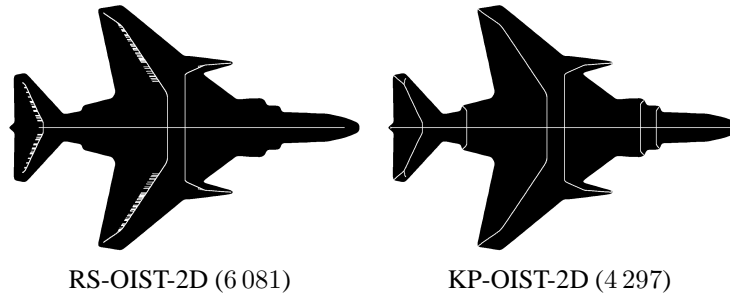


Figure 5. A 1437×1014 image with 487 620 object points of a plane and its “skeletons”.

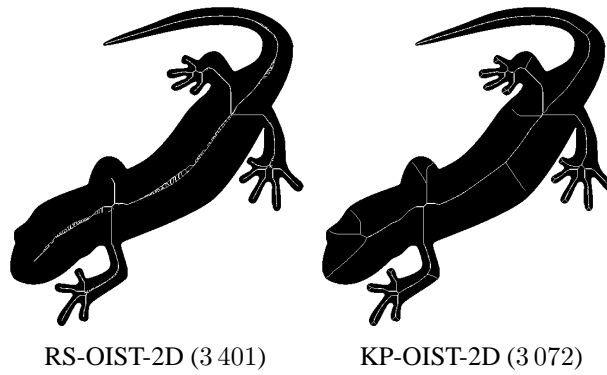


Figure 6. A 552×607 image with 108 615 object points of a salamander and its “skeletons”.

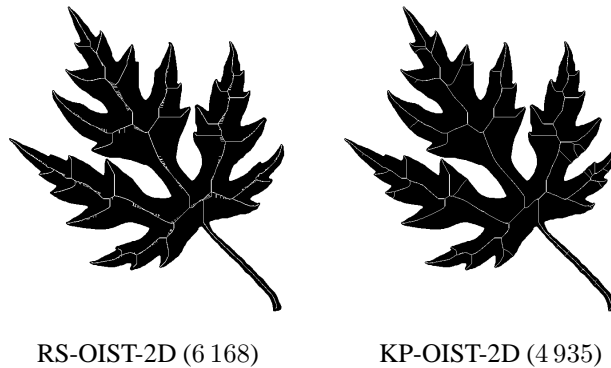


Figure 7. A 745×773 image with 152 611 object points of a leaf and its “skeletons”.

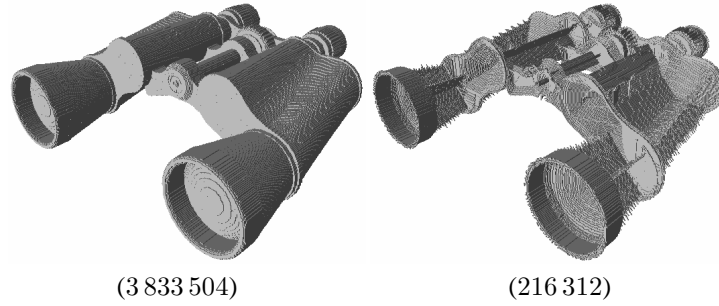


Figure 8. A $336 \times 133 \times 381$ 3D image of a binocular (left) and its medial surface produced by algorithm KP-OIST-3D (right).

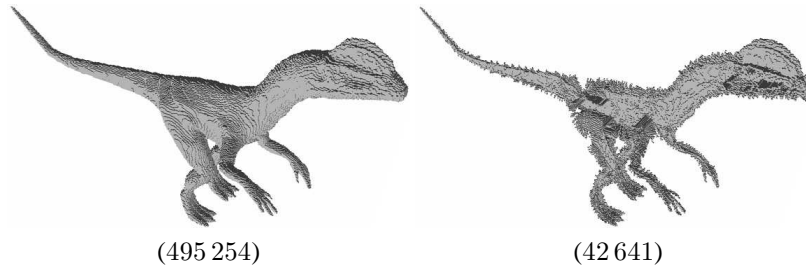


Figure 9. A $258 \times 474 \times 101$ 3D image of a dinosaur (left) and its medial surface produced by algorithm KP-OIST-3D (right).

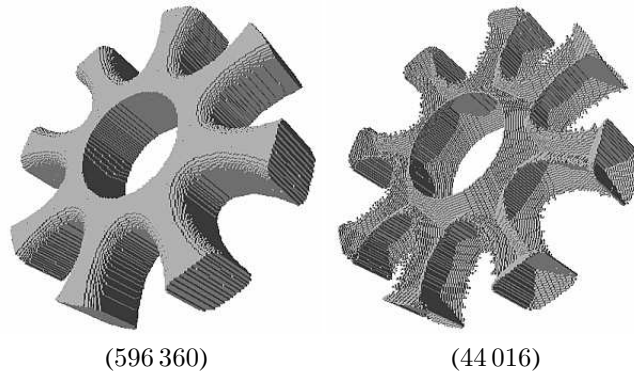


Figure 10. A $45 \times 191 \times 191$ 3D image of a gear (left) and its medial surface produced by algorithm KP-OIST-3D (right).

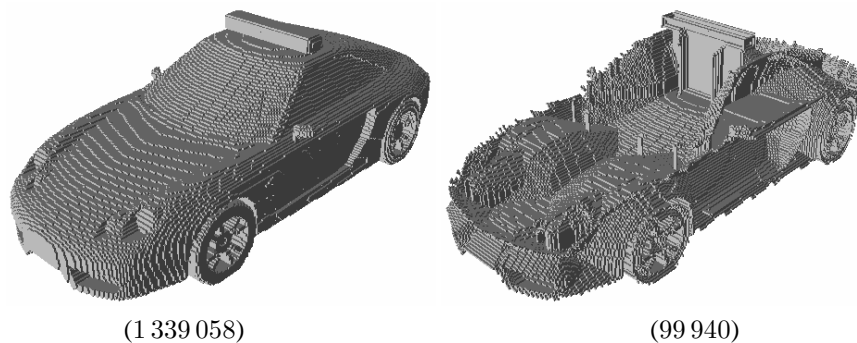


Figure 11. A $122 \times 93 \times 284$ 3D image of a Porsche (left) and its medial surface produced by algorithm KP-OIST-3D (right).