Convex and Concave Decomposition of Digitized Shapes Using Plane Probing and Visibility

Jacques-Olivier Lachaud[1,2][0000−0003−4236−2133] and Tristan Roussillon[2][0000−0003−2524−3685]

[1] Université Savoie Mont Blanc, CNRS, LAMA, F-73000 Chambéry, France
jacques-olivier.lachaud@univ-smb.fr
[2] Université Lyon, CNRS, INSA, LIRIS, F-69000 Lyon, France
tristan.roussillon@liris.cnrs.fr

Abstract. In this paper, we consider the geometrical analysis of oriented digital surfaces, which form the boundary of connected voxel sets. We present a method for the detection and reconstruction of its convex and concave parts. The proposed method composes two tools: the first one is a probing approach for the extraction of locally extremal points, the second one, based on the notion of visibility, joins some pairs of points whenever the straight segment between them stays sufficiently close to the input surface without crossing it. Finally, outer and inner candidate segments are assembled separately into facets, thus building polyhedral approximations of the local convex and concave zones.

Keywords: surface approximation · digital geometry · digital convexity

1 Introduction

Numerous studies focus on capturing the geometry of 3D shapes based on sampled data, whether in the form of point clouds, polygonal meshes, or digital surfaces. In this paper, we consider digital surfaces, defined as boundaries of connected voxel sets. They may be seen as closed, oriented, quadrangular meshes with vertices of integer coordinates.

Given a digital surface, we present a method for the detection and reconstruction of its convex and concave parts. The geometry of the convex hull is exactly recovered when the input data is digitally convex. This is therefore a first step towards a polygonal reconstruction that respects the convex and concave parts, which would be useful for many tasks in geometry processing.

Several methods could be considered for our purposes. Global approaches, e.g., based on the α-shape [2,11] or on a decimation strategy [5], are not expected to better capture the local geometry of the surface than local approaches. In the field of digital geometry, numerous works focus on the recognition of planar parts (see, e.g., [1]). In particular, one may use a plane-probing algorithm [10] to extract some tangent triangles. However, even when the input data is digitally convex, this approach is not able to recover all facets of the convex hull. Alternatively, we propose in this paper a probing algorithm to address the different problem of deciding whether a point is extremal or not.

In 2D, very good polygonal reconstructions of digital curves are obtained by local convex hull computations, done either explicitely [14,4,15] or implicitly [6,9,12]. We know only one attempt to extend this approach to 3D [13], but the proposed algorithm only works for a limited class of input. We go further in this direction, using the more recent notion of full convexity and tangency [7,8].

The outline is as follows. Section 2 defines the notion of convex and concave visibilities along a digital surface and establishes their links to convexity. Section 3 presents a probing approach to speed up the detection of convex and concave vertices. Section 4 explains the global reconstruction algorithm of convex and concave zones. Section 5 concludes the paper and offers some perspectives.

2 Convex and Concave Visibility

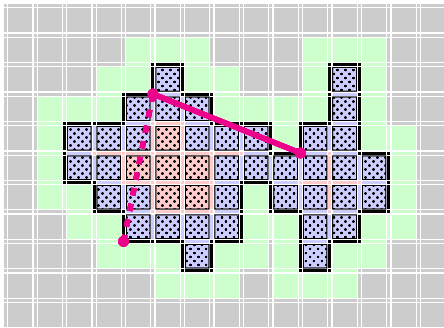
The input data is a set K of d-dimensional cells, i.e. *voxels*. Our objective is to identify the convex and concave zones of the digital boundary of K, which is an oriented digital surface.[3] We introduce the concept of *visibility* within exterior and interior parts of the digital surface, a kind of oriented alternative to the *tangency* to a digital set [8, Definition 4]. Remarkably, Theorem 2 shows that convex hull edges and facets of K are indeed visible when K is digitally convex.

Background definitions. There is a natural grid cell decomposition of the Euclidean space $\mathbb{R}^d$ with open unit cells (segments, squares, cubes, etc) such that their vertices are exactly the digital space $\mathbb{Z}^d$. We denote this cubical complex $\mathcal{C}^d$, while its k-dimensional cells are denoted by $\mathcal{C}^d_k$. For any subset Y of $\mathbb{R}^d$, the *star* of Y is $\mathrm{Star}\,(Y) := \{c \in \mathcal{C}^d, \bar{c} \cap Y \neq \emptyset\}$, where $\bar{c}$ is the topological closure of c. The *cover* of Y is $\mathrm{Cover}\,(Y) := \{c \in \mathcal{C}^d, c \cap Y \neq \emptyset\}$. Obviously the cover of a set is included in its star. The 0-dimensional cells of $\mathcal{C}^d$ are identified with $\mathbb{Z}^d$ and are called (lattice) points, or often *pointels* in the literature when thought of as vertices of higher dimensional cells. Given a subset Y of $\mathbb{R}^d$, its *convex hull* is denoted by $\mathrm{CvxH}\,(Y)$. The *set of extremal points* of a convex set C is denoted by $\mathrm{Extr}\,(C)$ (when C is a polytope, they are often called *vertices*). For a grid cell c, which is a convex set, we have $\mathrm{Extr}\,(c) = \bar{c} \cap \mathbb{Z}^d$. We recall one characterization of full convexity [7,8], and the related notion of tangency:

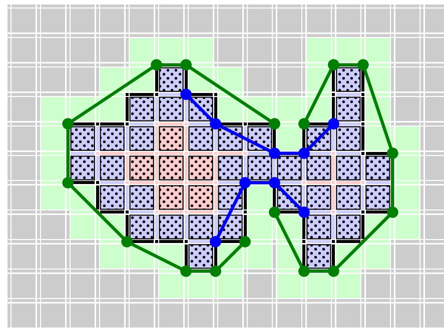
Definition 1. *A digital set* $X \subset \mathbb{Z}^d$ *is* digitally convex *iff* $\mathrm{CvxH}\,(X) \cap \mathbb{Z}^d = X$. *It is* fully convex *iff* $\mathrm{Star}\,(\mathrm{CvxH}\,(X)) = \mathrm{Star}\,(X)$. *A digital subset* $A \subset X \subset \mathbb{Z}^d$ *is* tangent *to* X *whenever* $\mathrm{Star}\,(\mathrm{CvxH}\,(A)) \subset \mathrm{Star}\,(X)$. *The elements of* A *are called* cotangent *in* X.

Otherwise said, if we consider two pointels $A = \{p, q\}$ of the digital surface, with X its set of pointels, they are cotangent in X whenever the straight line segment $[pq]$ touches cells that are also touched by X. Tangency defines a kind of visibility, since p and q can "see" each other if and only if their line of sight

[3] A more generic definition of oriented surface could be used [3], but the presentation is made easier if we consider the surface to be the boundary of a set of cells.



(a) Example of tangent edges in K (●—●) and non tangent edges in K (● - -●)



(b) Examples of convex K-visible edges (●—●) concave K-visible edges (●—●)

Fig. 1: Decomposition of the cellular grid $\mathcal{C}^d$ into disjoint set of cells and illustration of tangency and visiblity. The input of set of voxels K are hatched as ⚃, and $\mathcal{C}^d = \mathrm{Outer}\,(K) \cup \mathrm{Out}\,(K) \cup \mathrm{Bd}\,(K) \cup \mathrm{In}\,(K) \cup \mathrm{Inner}\,(K)$. The set X_K of 0-cells of $\mathrm{Cl}\,(K)$ is decomposed into the inner pointels I_K (the 0-cells of $\mathrm{Inner}\,(K)$) and the boundary pointels Z_K (the 0-cells of $\mathrm{Bd}\,(K)$).

stays close to the input digital surface. However this line of sight may cross the digital surface. Therefore we exploit the fact that we know where is the interior of the surface (the input set of interior voxels K), to keep this line of sights either exterior or interior. Convex vertices (resp. concave vertices) are visible to each other in the exterior (resp. interior) of the surface.

Let $K \subset \mathcal{C}^d_d$ be the input set of interior voxels to the surface. For any subset L of $\mathcal{C}^d$, its (combinatorial) *closure* $\mathrm{Cl}\,(L)$ is $\{c \in \mathcal{C}^d, \mathrm{Star}\,(c) \cap L \neq \emptyset\}$. For simplicity, the k-dimensional cells of a subset L of $\mathcal{C}^d$ are denoted by L_k. We denote by X_K the 0-cells of $\mathrm{Cl}\,(K)$. The digital set X_K is subdivided into two disjoint subsets: the set of *inner pointels* I_K is defined as $\{x \in X_K, \mathrm{Star}\,(x) \subset \mathrm{Cl}\,(K)\}$ and the set of *boundary pointels* Z_K is $X_K \setminus I_K$. We subdivide the cubical complex $\mathcal{C}^d$ into several disjoint subsets (see Fig. 1)

$$\begin{aligned}
\mathrm{Outer}\,(K) &:= \{c \in \mathcal{C}^d, \mathrm{Extr}\,(c) \subset (\mathbb{Z}^d \setminus X_K)\},\\
\mathrm{Out}\,(K) &:= \{c \in \mathcal{C}^d, \mathrm{Extr}\,(c) \cap (\mathbb{Z}^d \setminus X_K) \neq \emptyset \text{ and } \mathrm{Extr}\,(c) \cap (Z_K) \neq \emptyset\},\\
\mathrm{Bd}\,(K) &:= \{c \in \mathcal{C}^d, \mathrm{Extr}\,(c) \subset Z_K\},\\
\mathrm{In}\,(K) &:= \{c \in \mathcal{C}^d, \mathrm{Extr}\,(c) \cap (Z_K) \neq \emptyset \text{ and } \mathrm{Extr}\,(c) \cap I_K \neq \emptyset\},\\
\mathrm{Inner}\,(K) &:= \{c \in \mathcal{C}^d, \mathrm{Extr}\,(c) \subset I_K\}.
\end{aligned}$$

The analogy of $\mathrm{Bd}\,(K)$ with the topological boundary is clear, since it represents the cells whose star touches K and the complement of K.

Visibility and local convexity/concavity. As said above, tangency is not restricted enough to capture local convex edges and facets of a digital set. We introduce convex/concave visibility for this more specific purpose (see also Fig. 1).

Definition 2. *The elements of a non-empty set of points* $A := \{p_1, \ldots, p_n\}$ *of* Z_K *are* convex (resp. concave) K-visible *if and only if* $\mathrm{Cover}\left(\mathrm{CvxH}\left(A\right)\right) \subset \mathrm{Out}\left(K\right) \cup \mathrm{Bd}\left(K\right)$ *(resp.* $\mathrm{Cover}\left(\mathrm{CvxH}\left(A\right)\right) \subset \mathrm{In}\left(K\right) \cup \mathrm{Bd}\left(K\right)$*).*

The convex hull of the points visible from a point x of Z_K tells if x is salient:

Definition 3. *Let* $x \in Z_K$*. The point* x *is a* convex (resp. concave) corner *iff exactly one cell of* K *(resp. exactly one cell of* $\mathcal{C}^d_d \setminus K$*) is incident to* x*. The* convex visibility cone $\mathcal{V}^{\mathrm{Out}}_K(x)$ *of* x *is the set of points* y *of* Z_K *such that* x, y *are convex* K*-visible. The* concave visibility cone $\mathcal{V}^{\mathrm{In}}_K(x)$ *of* x *is the set of points* y *of* Z_K *such that* x, y *are concave* K*-visible. The point* x *is* locally convex (resp. concave) on the boundary of K *iff* $x \in \mathrm{Extr}\left(\mathrm{CvxH}\left(\mathcal{V}^{\mathrm{Out}}_K(x)\right)\right)$ *and* x *is a convex corner, resp.* $x \in \mathrm{Extr}\left(\mathrm{CvxH}\left(\mathcal{V}^{\mathrm{In}}_K(x)\right)\right)$ *and* x *is a concave corner.*

Note that we require x to be a convex (resp. concave) corner in order to define it as a locally convex (resp. concave) corner. At first glance, this condition may seem unnecessary, since if it is not met, x cannot be locally convex or concave anyway. However, a concave corner may appear as a vertex of the convex hull of its convex K-visible points and vice versa. To eliminate this undesirable edge case, we only consider convex (resp. concave) K-visible points from a convex (resp. concave) corner.

From now on, we focus on convex definitions and properties, but analogous results hold by replacing convex with concave. We assemble the locally convex points into polyhedral objects as follows.

Definition 4. *Let* $A \subset Z_K$*. The set* $\mathrm{CvxH}\left(A\right)$ *is called a* locally convex face of K *whenever* $\mathrm{CvxH}\left(A\right)$ *is a face of* $\mathrm{CvxH}\left(\cup_{a \in A} \mathcal{V}^{\mathrm{Out}}_K(a)\right)$*.*

Lemma 2 (in Appendix A, page 13) shows that incident vertices, edges, etc, to a locally convex face are also locally convex. It is clear that the locally convex vertices and faces of K form a disjoint set with the locally concave vertices and faces of K, since their extrema must belong to either the convex corners or the concave corners. The following theorem justifies this definition of local convexity, since it coincides with the convex hull of Z_K when X_K is fully convex.

Theorem 1. *Let* $K \subset \mathcal{C}^d_d$ *and* X_K *be fully convex. The vertices and the faces of* $\mathrm{CvxH}\left(Z_K\right)$ *are locally convex vertices and locally convex faces of* K*.*

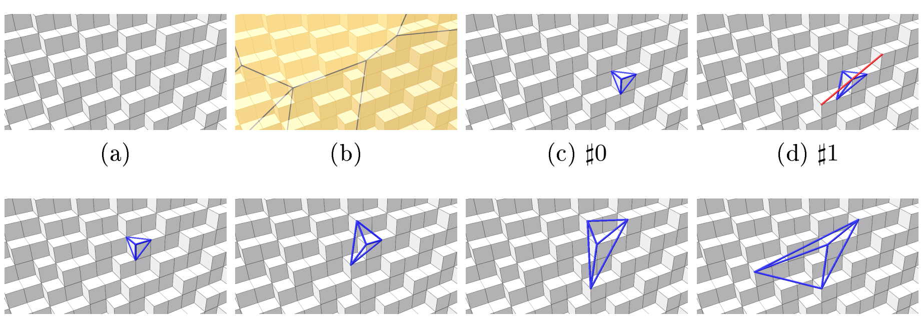
Proof. First it is clear that any vertex of $\mathrm{CvxH}\left(Z_K\right)$ must be a convex corner of K and is also convex K-visible (belonging to $\mathrm{Bd}\left(K\right)$). We use here Theorem 2 (Appendix A, page 14), which tells that any vertex and face of $\mathrm{CvxH}\left(Z_K\right)$ are convex K-visible. Now, let $A \subset Z_K$ be such that $F := \mathrm{CvxH}\left(A\right)$ is a face of $\mathrm{CvxH}\left(Z_K\right)$. So $F \subset \partial \mathrm{CvxH}\left(Z_K\right)$ and any $a \in A$ is a vertex of $\mathrm{CvxH}\left(Z_K\right)$. Clearly $\mathcal{V}^{\mathrm{Out}}_K(a) \subset Z_K$ so $\mathrm{CvxH}\left(\mathcal{V}^{\mathrm{Out}}_K(a)\right) \subset \mathrm{CvxH}\left(Z_K\right)$. If a is not locally convex, then it is a convex combination of points of $\mathcal{V}^{\mathrm{Out}}_K(a) \subset \mathrm{CvxH}\left(Z_K\right)$, and it cannot be a vertex of $\mathrm{CvxH}\left(Z_K\right)$, a contradiction. Thus a is locally convex.

The face $F = \mathrm{CvxH}\left(A\right)$ is a subset of each $\mathrm{CvxH}\left(\mathcal{V}^{\mathrm{Out}}_K(a)\right)$ for $a \in A$ (since any subset of its vertices are convex visible according to Theorem 2). So $F \subset \cup_{a \in A} \mathrm{CvxH}\left(\mathcal{V}^{\mathrm{Out}}_K(a)\right) \subset \mathrm{CvxH}\left(\cup_{a \in A} \mathcal{V}^{\mathrm{Out}}_K(a)\right)$.

If we assume that there is a $y \in F$ with $y \in \mathrm{int}(\mathrm{CvxH}\left(\cup_{a\in A}\mathcal{V}_K^{\mathrm{Out}}(a)\right))$, then it follows that $y \in \mathrm{int}(\mathrm{CvxH}\left(Z_K\right))$ since any point of $\mathcal{V}_K^{\mathrm{Out}}(a)$ belongs to Z_K. But this is a contradiction to F being a face of $\mathrm{CvxH}\left(Z_K\right)$. So either $y \in \partial\mathrm{CvxH}\left(\cup_{a\in A}\mathcal{V}_K^{\mathrm{Out}}(a)\right)$ or y is outside this convex set. The second case is impossible since $F \subset \mathrm{CvxH}\left(\cup_{a\in A}\mathcal{V}_K^{\mathrm{Out}}(a)\right)$ as shown above. So F lies in the boundary of $\mathrm{CvxH}\left(\cup_{a\in A}\mathcal{V}_K^{\mathrm{Out}}(a)\right)$. A convex set lying on the boundary of a convex set Y such that each of its extremal points are extremal points of Y is by definition a face of Y. So F is a locally convex face of K. □

3 Extremal Point Detection

In this section, we consider a digital set $Z \subset \mathbb{Z}^3$. The dimension of the space is restricted to $d = 3$ to better align with the illustrations, but it can be extended to arbitrary d without difficulty. A point $\mathbf{z} \in Z$ is *extremal* if and only if $\mathbf{z} \in \mathrm{Extr}\left(\mathrm{CvxH}\left(Z\right)\right)$. In addition, $\mathbf{z} \in Z$ is *locally extremal* if and only if it is extremal for the convex hull of some non-empty subset $Z' \subseteq Z$ containing $\mathbf{z}$. In theory, Z' could be equal to $\{\mathbf{z}\}$ or to Z. In the first case, the definition would be too local, as every point would be considered locally extremal. In the second case, it would not be local enough, as only extremal points would be considered locally extremal. The challenge is to find the right trade-off, which the approach described below manages remarkably well in practice.



(a) (b) (c) ♯0 (d) ♯1

(e) ♯0 (f) ♯3 (g) ♯6 (h) ♯8

Fig. 2: Z is the set of pointels of a digital surface, denoted by Z_K in Section 2 and shown in gray colors in (a). $\mathrm{CvxH}\left(Z\right)$ is shown in (b). An initial tetrahedron is shown in (c). In this case, the algorithm finds a point alignment (in red) after one update and the starting point is thus labelled as not extremal. An execution from a vertex of $\mathrm{CvxH}\left(Z\right)$ is shown from (e) to (h). The base facet of the tetrahedron aligns with a supporting plane after eight iterations and the starting point is labelled as locally extremal.

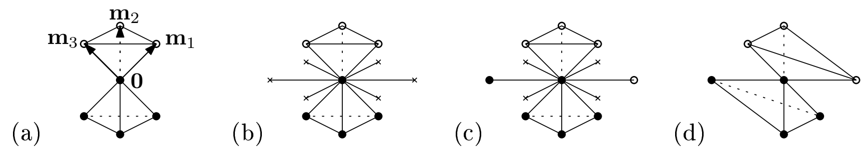


Fig. 3: Illustration of Definition 5 (a) and Definition 7 (b-d). Points in (resp. not in) Z are depicted with black (resp. white) disks. A valid tetrahedron is shown in (a). Among the six possible operations, depicted with crosses in (b), a valid one is shown in (c). Having applied this operation, the resulting tetrahedron is shown in (d). One can check that it is valid.

We propose a probing algorithm that labels a point $\mathbf{z} \in Z$ as locally extremal or not extremal. Informally, the algorithm starts with an initial tetrahedron. The apex is placed at $\mathbf{z}$ and the opposite triangular face stands below it. The triangle is then iteratively updated with elementary vector operations using only queries "is $\mathbf{x}$ in Z?" (see Fig. 2). The algorithm stops if it detects a point alignment that makes $\mathbf{z}$ not extremal or if the normal vector to the triangle gives a direction for which $\mathbf{z}$ is extremal with respect to the set Z' of visited points, thus locally extremal. It is very similar to the H-algorithm [10] but solves a different problem.

In more details, by translation, we assume without loss of generality that the point to consider is the origin $\mathbf{0}$. The tetrahedron is then conveniently represented by a 3×3 matrix $\mathbf{M}$ with integer coefficients, where the three column vectors are denoted by $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$. See Fig. 3 (a) and Definition 5.

Definition 5 (Fig. 3 (a)). *Let $Z \subset \mathbb{Z}^3$ be a digital set containing $\mathbf{0}$. A tetrahedron is a matrix $\mathbf{M} \in \mathbb{Z}^{3\times3}$, which is valid with respect to Z if*

$$-\mathbf{m}_1, -\mathbf{m}_2, -\mathbf{m}_3 \in Z, \quad \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \notin Z, \quad \det(\mathbf{M}) = 1. \tag{1}$$

The update of the tetrahedron is conveniently represented by a 3×3 matrix with integer coefficients taken from a set of six matrices with determinant 1:

Definition 6. *Let Σ be the set of all permutations over $\{1, 2, 3\}$. For a permutation $\sigma \in \Sigma$, let $\boldsymbol{U}_\sigma$ be a 3×3 matrix having -1 at the intersection between the $\sigma(1)$-th column and the $\sigma(2)$-th row and the same entries as $\mathbf{I}$, the 3×3 identity matrix, elsewhere. Let $\mathbb{U}$ be the set $\{\boldsymbol{U}_\sigma \mid \sigma \in \Sigma\}$.*

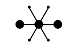
Note that computing $\mathbf{M}\boldsymbol{U}_\sigma$ amounts to subtracting the $\sigma(2)$-th column of $\mathbf{M}$ to the $\sigma(1)$-th column. As shown in Fig. 3 (b), the six possible results (determined by the six permutations $\sigma \in \Sigma$) form the vertices of an hexagon around the origin. Any such operation is possible if the resulting tetrahedron is valid.

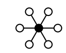
Definition 7 (Fig. 3 (c-d)). *Let $Z \subset \mathbb{Z}^3$ be a digital set containing $\mathbf{0}$ and let $\mathbf{M} \in \mathbb{Z}^{3\times3}$ be a valid tetrahedron. An operation is a function $f : \mathbb{Z}^{3\times3} \mapsto \mathbb{Z}^{3\times3}$ such that given $\mathbf{M}' = f(\mathbf{M})$, there exists a matrix $\boldsymbol{U}_\sigma \in \mathbb{U}$ that satisfies $\mathbf{M}' = \mathbf{M}\boldsymbol{U}_\sigma$. Moreover, f is valid if $\mathbf{M}'$ is valid, i.e., if*

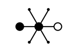
$$-\mathbf{m}_{\sigma(1)} + \mathbf{m}_{\sigma(2)} \in Z, \quad \mathbf{m}_{\sigma(1)} - \mathbf{m}_{\sigma(2)} \notin Z. \tag{2}$$

The algorithms halts if there is no valid operation, but also whenever two points symmetric with respect to $\mathbf{0}$ are in Z. Otherwise it continues to probe:

Definition 8. *Let $Z \subset \mathbb{Z}^3$ be a digital set containing $\mathbf{0}$ and let $\mathbf{M} \in \mathbb{Z}^{3\times 3}$ be a valid tetrahedron. The state* $\text{STATE}(\mathbf{M})$ *of* $\mathbf{M}$ *is defined as:*

NO — *if* $\exists \sigma \in \Sigma$ *s.t.* $-\mathbf{m}_{\sigma(1)} + \mathbf{m}_{\sigma(2)}, \mathbf{m}_{\sigma(1)} - \mathbf{m}_{\sigma(2)} \in Z$, 

YES — *if* $\forall \sigma \in \Sigma,\ -\mathbf{m}_{\sigma(1)} + \mathbf{m}_{\sigma(2)}, \mathbf{m}_{\sigma(1)} - \mathbf{m}_{\sigma(2)} \notin Z$,  (3)

PROBING — *otherwise* 

Finally, our algorithm ISEXTREMAL reads as follows:
Input: a digital set Z containing $\mathbf{0}$, a valid tetrahedron $\mathbf{M}$.
while $\text{STATE}(\mathbf{M}) = \text{PROBING}$ **do**
⌊ Find a valid operation f and apply it: $\mathbf{M} \leftarrow f(\mathbf{M})$
return $\text{STATE}(\mathbf{M})$

The initial tetrahedron is usually a (positively oriented) orthonormal basis. By rotations, we can assume it is $\mathbf{I}$ without loss of generality.

If the algorithm terminates, the output is either NO or YES. In the former case, $\mathbf{0}$ is not extremal because it lies in a segment joining two other points of Z. In the latter case, $\mathbf{0}$ is locally extremal and more precisely, extremal with respect to the set of visited points. We can show that the algorithm indeed terminates for any finite Z and that it returns YES if $\mathbf{0}$ is a vertex of $\text{CvxH}(Z)$ (see Theorem 3 and Theorem 4 in Appendix B, page 14).

Note that finding a valid operation is quite easy: we have to check if there exists $\sigma \in \Sigma$ such that $-\mathbf{m}_{\sigma(1)} + \mathbf{m}_{\sigma(2)} \in Z$ and $\mathbf{m}_{\sigma(1)} - \mathbf{m}_{\sigma(2)} \notin Z$. This can be done by at most six calls to a set-membership predicate. There are usually several valid operations. The above-mentioned termination results do not depend on a particular choice. In some cases, however, the way we choose the operations may change the final answer. In the experiments, we follow the approach previously used in the H-algorithm [10]: we use an in-sphere criterion and, if several points are in a cospherical position, we resort to a lexicographic order.

Algorithm ISEXTREMAL provides in practice a fast way to discard a large amount of non-extremal points, especially along digitizations of smooth surfaces. As an example, the ellipsoid $x^2 + 25y^2 + 25z^2 - 90 \leq 0$ is digitized at grid steps $0.5, 0.1, 0.05$ and 0.01 to obtain four sets of voxels. For each one, we have processed the set of boundary pointels Z after surface tracking. The results are reported in the table below, where n_c denotes the number of corners, i.e., points $\mathbf{z} \in Z$ such that $\forall k \in \{1,2,3\}, \mathbf{z} \pm \mathbf{e}_k \in Z, \mathbf{z} \pm \mathbf{e}_k \notin Z$, from where the algorithm is initialized, and n_e is the number of points detected as locally extremal. Note that $n_c \geq n_e$ by definition and $n_e \geq |\text{CvxH}(Z)|$. Experimentally, n_e is equal or slightly greater than $|\text{CvxH}(Z)|$, while it is far less than n_c.

grid step	$\|Z\|$	n_c	n_e	$\|\text{CvxH}(Z)\|$
0.5	984	112	112	112
0.1	24.808	2.032	1.128	1.128
0.05	99.448	7.784	3.064	3.064
0.01	2.488.104	186.664	33.864	33.784

Algorithm 1: Given the interior voxels K, returns a surface meshing of the convex zones as a pair of vertices and faces joining them.

```
Function BuildLocallyConvexZones( In K: subset of C_d^d ) : Mesh;
begin
    E ← ∅ ;                                                          // (I)
    Z ← Cl(K) ∩ Z^d ;
    foreach convex corner z of K do
        M ← GetValidTetrahedron(z, Z) ;
        if IsExtremal(M, Z − z) then  E ← E ∪ {z} ;
    V ← GetVisibilityPairs(K, E) ;                                   // (II)
    (C_v, C_e) ← GetLocallyConvexVerticesAndEdges(E, V) ;            // (III)
    C_f ← GetLocallyConvexFacets(C_v, C_e, V) ;                      // (IV)
    return (C_v, C_f)
```

Algorithm 2: Given the interior voxels K and extremal convex corners E, returns all the visible K-convex pairs.

```
Function GetVisibilityPairs( In K: voxels, In E: points ) : set of
 (point,point);
begin
    R ← ∅;
    foreach (p, q) ∈ E × E do
        if p ≠ q and Cover(CvxH({p, q})) ⊂ Bd(K) ∪ Out(K) then
            R ← R ∪ {(p, q)};
    return R
```

4 Reconstruction of Locally Convex and Concave Zones

General presentation. Algorithm 1 provides the general algorithm for reconstructing the locally convex zones along the boundary of a set of interior voxels K. It is composed of 4 stages:

(I) Filtering the convex corners with the probing algorithm IsExtremal
(II) Computing the convex visibility between these points (Algorithm 2)
(III) Extracting the locally convex vertices and edges among them (Algorithm 3)
(IV) Extracting the locally convex facets between them (Algorithm 4)

Computing the locally concave zones is a completely symmetric algorithm and is not detailed here (just starts from concave corners). Note that the true C++ implementation uses indices instead of sets to speed up computations.

Algorithm 3: Given points E and their visibility V, filters them to return the locally convex vertices and edges.

```
Function GetLocallyConvexVerticesAndEdges( In E: points, In V:
 set of (point,point) ) : (points, set of (point,point)) ;
Var M: map point to bool, N: map (point,point) to bool;
begin
    S ← ∅ ;                                   // set of candidate convex edges
    foreach p ∈ E do M[p] ← true;
    foreach (p,q) ∈ V do N[(p,q)] ← false;
    foreach p ∈ E do
        Q ← ConvexHull({p} ∪ {q ∈ E, (p,q) ∈ V});
        if Q is not full dimensional or p ∉ Extr(Q) then
            M[p] ← false;
            continue;
        foreach facet f = (p_0,...,p_{k-1}) of Q do
            if ∃i ∈ {0,...,k-1}, p_i = p then
                S ← S ∪ {(p,p_{i-1})} ∪ {(p,p_{i+1})} ;     // indices modulo k
    foreach (p,q) ∈ V do
        if (p,q) ∉ S or (q,p) ∉ S then continue;
        L ← {r ∈ E, (p,r) ∈ V and (p,r) ∈ S and (r,p) ∈ S};
        L ← L ∪ {r ∈ E, (q,r) ∈ V and (q,r) ∈ S and (r,q) ∈ S};
        Q ← ConvexHull(L);
        if (p,q) is an edge of Q then N[(p,q)] ← true;
    C_v ← {p ∈ E, M[p] = true};
    C_e ← {(p,q) ∈ V, N[(p,q)] = true};
    return (C_v, C_e)
```

Computational considerations. Note that more than 95% of the computation time is related to stage (II), the computation of all the visible pairs. Its complexity grows quadratically with the number of candidate extremal points. This is why it is crucial to filter the convex corners as most as possible and this is exactly what does the probing algorithm IsExtremal. Furthermore, many of the visibility tests are done between candidate points that are very far away from each other and probably unrelated. Instead of using a heuristic (like assuming that points further away from a given distance are not visible), we use a coarse-to-fine pyramid approach to check covering and tangency, that can be proven to be exact. We do not have the place to develop it here, but we achieve between 10× to 200× speed-up with this technique.

Results and timings. We have tested our method on a large amount of digital surfaces. Some results are shown in Fig. 4 and Table 1 (computations made on a Macbook Pro, apple M2 Max, 64 Go RAM). The convex and concave parts are accurately detected, and their polyhedral reconstruction closely matches the convex hull in the most salient regions as shown in Fig. 4 (d) and (g). Remarkably,

Algorithm 4: Given the locally convex vertices C_v and edges C_e, the visibility V, returns the locally convex facets joining them.

```
Function GetLocallyConvexFacets( In C_v: points, In C_e, V: set of
  (point,point) ) : set of facets ;
begin
  F ← ∅;
  foreach p ∈ C_v do
    Q ← ConvexHull({p} ∪ {q ∈ C_v, (p,q) ∈ V});
    foreach facet f ∈ Q do
      if p ∈ Extr(f) and every edge of f is in C_e then
        // Some faces may be present several times
        f' ← LowestCircularPermutation(f);
        F ← F ∪ {f'} ;                        // (3,1,6) → (1,6,3)
  return F
```

shape	$\lvert Z_k \rvert$	convex zones				concave zones				time
		n_c	n_e	n_v	n_f	n_c	n_e	n_v	n_f	(s)
`twospheres`	21.758	3.258	1.580	1.580	1.339	2.664	0	0	0	5
`torusknot`	96.622	15.196	2.930	2.930	2.765	13.347	5	5	0	31
`sharpsphere`	119.846	16.715	3.111	3.111	2.546	15.011	412	412	271	40
`goursat`	104.984	7.776	2.160	2.160	2.408	7.376	288	288	246	24
`leopold`	144.440	33.672	1.664	1.664	1.538	32.256	856	856	744	19

Table 1: For each digital surface of Fig. 4, we provide the overall running time together with the number of input points ($|Z_k|$), corners (n_c), extremal points (n_e), locally convex/concave points and faces (n_v and n_f).

the columns n_e and n_v are identical, which means that the probing algorithm IsExtremal identifies all the locally convex and concave vertices. Thanks to algorithm IsExtremal and the above-mentioned additional optimization tricks, our method achieves low running times relative to the difficulty of the problem. Note that the concave parts are located inside the volume and are thus not visible on Fig. 4 (c), (e) and (h). Many parts are of course neither convex nor concave and addressing them remains a key direction for future work.

5 Conclusion

In this paper we have proposed an original definition of local convexity and concavity along a digital surface. The definition, based on the notion of visibility, is sound in the sense that it matches the convex hull when the digital surface is digitally (fully) convex. Computing local convexity/concavity requires the identification of extremal points. We have proposed a fast probing algorithm that quickly prune corners to retain extremal points. The computation of visibilities

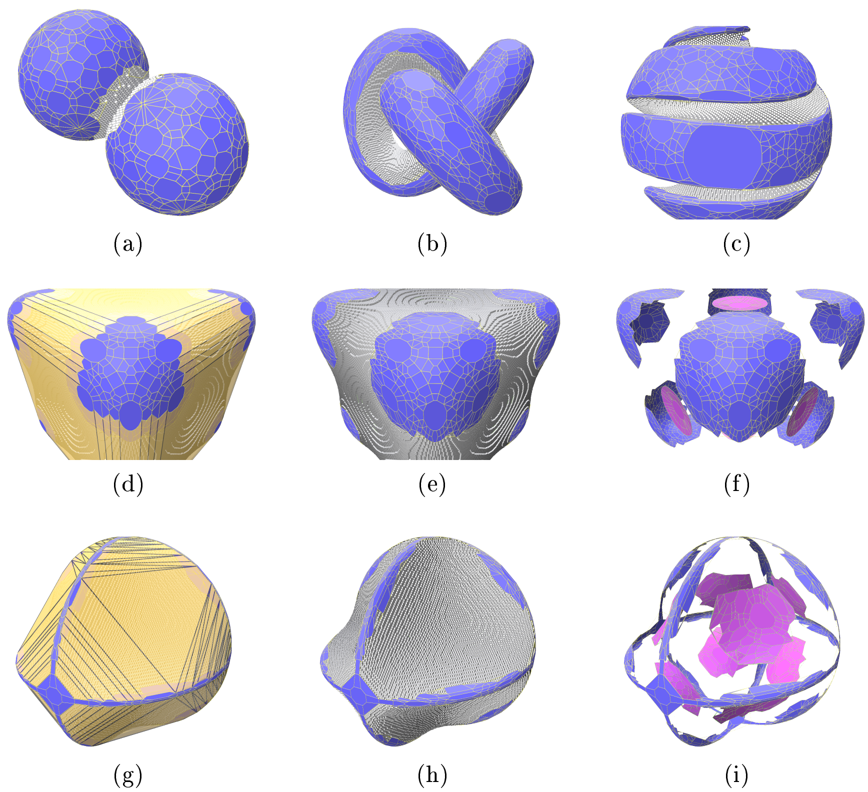


Fig. 4: Reconstructions of `twospheres`, `torusknot`, `sharpsphere`, `goursat` and `leopold` are respectively shown in (a), (b), (c), (d)-(f) and (g)-(i). The digital surface, the convex hull, the convex and concave parts are respectively colored in grey, yellow, blue and pink. Statistics and timings are collected in Table 1.

between extremal points is thus greatly reduced. Locally convex/concave edges and faces are then quickly determined with local convex hull computations. Experiments illustrate that the decomposition of the digital surface into convex and concave zones is exact and not approximate.

One of the limitations of the work is the computation of visibility between all pairs of extremal points. A pyramid approach is used to speed up these computations, but computing visibility remains the bottleneck of the method. We are currently investigating "compatibility criteria" between extremal points to prune further the number of tested pairs. Our final objective remains the full polyhedral decomposition of the digital surface into convex, concave, and saddle parts. A plane-probing variant could be able to detect potential saddle zones. It remains to understand how to combine this information to triangulate these zones in a deterministic way.

Acknowledgments. This work is partially supported by the French National Research Agency within the StableProxies project (ANR-22-CE46-0006).

References

1. Brimkov, V., Coeurjolly, D., Klette, R.: Digital planarity – a review. Discrete Applied Mathematics **155**(4), 468–495 (2007)
2. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. **13**(1), 43–72 (Jan 1994). https://doi.org/10.1145/174462.156635
3. Herman, G.T.: Oriented surfaces in digital spaces. CVGIP: Graphical models and image processing **55**(5), 381–396 (1993)
4. Klette, G.: Recursive calculation of relative convex hulls. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) Discrete Geometry for Computer Imagery. pp. 260–271. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
5. Kobbelt, L., Campagna, S., Seidel, H.P.: A general framework for mesh decimation. In: Proceedings of the Graphics Interface 1998 Conference, June 18-20, 1998, Vancouver, BC, Canada. pp. 43–50 (June 1998), http://graphicsinterface.org/wp-content/uploads/gi1998-6.pdf
6. Lachaud, J.O., Provençal, X.: Two linear-time algorithms for computing the minimum length polygon of a digital contour. Discrete Applied Mathematics **159**(18), 2229–2250 (2011). https://doi.org/10.1016/j.dam.2011.08.002
7. Lachaud, J.O.: An alternative definition for digital convexity. In: Lindblad, J., Malmberg, F., Sladoje, N. (eds.) Discrete Geometry and Mathematical Morphology - First International Joint Conference, DGMM 2021, Uppsala, Sweden, May 24-27, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12708, pp. 269–282. Springer (2021). https://doi.org/10.1007/978-3-030-76657-3_19
8. Lachaud, J.O.: An alternative definition for digital convexity. J. Math. Imaging Vis. **64**(7), 718–735 (2022). https://doi.org/10.1007/s10851-022-01076-0
9. Lachaud, J.O., Provençal, X.: Dynamic minimum length polygon. In: Aggarwal, J.K., Barneva, R.P., Brimkov, V.E., Koroutchev, K.N., Korutcheva, E.R. (eds.) Combinatorial Image Analysis. pp. 208–221. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
10. Lachaud, J.O., Provençal, X., Roussillon, T.: Two Plane-Probing Algorithms for the Computation of the Normal Vector to a Digital Plane. Journal of Mathematical Imaging and Vision **59**(1), 23 – 39 (Sep 2017). https://doi.org/10.1007/s10851-017-0704-x
11. Portaneri, C., Rouxel-Labbé, M., Hemmer, M., Cohen-Steiner, D., Alliez, P.: Alpha wrapping with an offset. ACM Transactions on Graphics (TOG) **41**(4), 1–22 (2022)
12. Roussillon, T., Sivignon, I.: Faithful polygonal representation of the convex and concave parts of a digital curve. Pattern Recognition **44**(10-11), 2693–2700 (Nov 2011), https://hal.archives-ouvertes.fr/hal-00643573
13. Schulz, H.: Polyhedral approximation and practical convex hull algorithm for certain classes of voxel sets. Discrete Applied Mathematics **157**(16), 3485–3493 (2009). https://doi.org/10.1016/j.dam.2009.04.008, combinatorial Approach to Image Analysis
14. Sklansky, J., Chazin, R.L., Hansen, B.J.: Minimum-perimeter polygons of digitized silhouettes. IEEE Transactions on Computers **21**(3), 260–268 (1972)
15. Wiederhold, P.: Computing the minimal perimeter polygon for sets of rectangular tiles based on visibility cones. Journal of Mathematical Imaging and Vision **66**(5), 873–903 (2024). https://doi.org/10.1007/s10851-024-01203-z

A Properties related to visibility

We gather here some properties related to visibility and necessary to give some theoretical guarantees for local convexity.

We first state without proof a few elementary properties of the sets defined in Section 2.

Lemma 1. *For any $K \subset \mathcal{C}^d_d$, the set of cells $\mathcal{C}^d$ is the disjoint union of* $\mathrm{Outer}\,(K)$, $\mathrm{Out}\,(K)$, $\mathrm{Bd}\,(K)$, $\mathrm{In}\,(K)$, $\mathrm{Inner}\,(K)$*. Furthermore* $\mathrm{Outer}\,(K)$, $\mathrm{Bd}\,(K)$ *and* $\mathrm{Inner}\,(K)$ *are closed.* $\mathrm{Out}\,(K)$ *and* $\mathrm{In}\,(K)$ *are open.*

The next lemma tells that the definition of locally convex face is sound, i.e., all faces incident to a locally convex face are also locally convex.

Lemma 2. *For any $\emptyset \neq A' \subset A \subset Z_K$, if* $\mathrm{CvxH}\,(A)$ *is a locally convex face of K, then* $\mathrm{CvxH}\,(A')$ *is a locally convex face of K too.*

Proof. Assume $\mathrm{CvxH}\,(A)$ is a locally convex face of K. Let $C := \mathrm{CvxH}\left(\cup_{a\in A}\mathcal{V}^{\mathrm{Out}}_K(a)\right)$ and $C' := \mathrm{CvxH}\left(\cup_{a\in A'}\mathcal{V}^{\mathrm{Out}}_K(a)\right)$. We have $\mathrm{CvxH}\,(A') \subset C'$. If $\mathrm{CvxH}\,(A') \not\subset \partial C'$ then $\mathrm{CvxH}\,(A') \cap \mathrm{int}(C') \neq \emptyset$ (it must cross its interior). But obviously $C' \subset C$, which implies $\mathrm{int}(C') \subset \mathrm{int}(C)$. It follows that $\mathrm{CvxH}\,(A') \cap \mathrm{int}(C) \neq \emptyset$. This contradicts $\mathrm{CvxH}\,(A') \subset \mathrm{CvxH}\,(A) \subset \partial C$ ($\mathrm{CvxH}\,(A)$ is a face of C) since ∂C and $\mathrm{int}(C)$ are disjoint. □

We show below that the pointels I_K are indeed interior points as they belong to the interior of the convex hull of X_K.

Lemma 3. *Let $K \subset \mathcal{C}^d_d$ non-empty. Then X_K and Z_K have the same convex hull (i.e.* $\mathrm{CvxH}\,(Z_K) = \mathrm{CvxH}\,(X_K)$*). Furthermore* $\mathrm{CvxH}\,(Z_K)$ *is full dimensional, and I_K belongs to* $\mathrm{int}(\mathrm{CvxH}\,(Z_K))$*, which is not empty.*

Proof. Since the convex hull is an increasing operator and $Z_K \subset X_K$, we have $\mathrm{CvxH}\,(Z_K) \subset \mathrm{CvxH}\,(X_K)$. To show the reverse inclusion, X_K being finite, the extremal points V of $\mathrm{CvxH}\,(X_K)$ form a subset of X_K. Let $v \in V$. Let i be an arbitrary direction. If $v + \mathbf{e}_i \in X_K$ and $v - \mathbf{e}_i \in X_K$, it means that v is a linear combination of two points of X_K and cannot be an extremal point of $\mathrm{CvxH}\,(X_K)$. So either $v^+ := v + \mathbf{e}_i$ or $v^- := v - \mathbf{e}_i$ or both are not in X_K. Assume it is v^+ wlog. So the 1-cell c^+ of boundary $\{v, v^+\}$ must be in $\mathrm{Out}\,(K)$ and v must belong to Z_K. Therefore, $V \subset Z_K$ and we conclude that $\mathrm{CvxH}\,(X_K) = \mathrm{CvxH}\,(V) \subset \mathrm{CvxH}\,(Z_k)$.

Besides, if $x \in I_K$, it follows that all neighbors $N := (x \pm \mathbf{e}_i)_{i\in\{1,\ldots,d\}}$ are in X_K too. Since $\mathrm{CvxH}\,(N)$ is full dimensional, $x \in \mathrm{int}(\mathrm{CvxH}\,(N)) \subset \mathrm{int}(\mathrm{CvxH}\,(X_K)) = \mathrm{int}(\mathrm{CvxH}\,(Z_K))$. □

It is quite intuitive that the boundary of the convex hull of a digitally convex digital set should be close to the boundary points of this digital set. However this result if false if we consider classical digital convexity. For instance just pick two points separated by a primitive lattice vector of ∞-norm strictly greater than 1. The full convexity is thus necessary in the following property:

Theorem 2. *Let* $K \subset \mathcal{C}^d_d$. *If* X_K *is fully convex, then* $\mathrm{Cover}\,(\partial \mathrm{CvxH}\,(Z_K)) \subset \mathrm{Out}\,(K) \cup \mathrm{Bd}\,(K)$. *Hence any edge or facet of* $\mathrm{CvxH}\,(Z_K)$ *are convex* K*-visible.*

Proof. Let F be any facet of $\mathrm{CvxH}\,(X_K)$, which is also a facet of $\mathrm{CvxH}\,(Z_K)$ by Lemma 3. Hence $F \subset \partial \mathrm{CvxH}\,(Z_K)$. Let $c \in \mathrm{Cover}\,(F)$, so $c \cap F \neq \emptyset$, and let $y \in c \cap F$. Let $V := \mathrm{Extr}\,(c)$ be the vertices of c.

Since $y \in F$, we have $y \in \partial \mathrm{CvxH}\,(Z_K) \subset \mathrm{CvxH}\,(Z_K) = \mathrm{CvxH}\,(X_K)$. So $\mathrm{Star}\,(y) = \{c\} \subset \mathrm{Star}\,(\mathrm{CvxH}\,(X_K)) = \mathrm{Star}\,(X_K)$, the latter equality coming from full convexity of X_K. At least one of the extremal point of c must belong to X_K, and c cannot be in $\mathrm{Outer}\,(K)$.

We show now that no point of V can be in I_K. Assume there is some $x \in V$ with $x \in I_K$. Then $c \in \mathrm{Inner}\,(K)$ or $c \in \mathrm{In}\,(K)$. If $c \in \mathrm{Inner}\,(K)$, then $V \subset I_K$. By Lemma 3, we know that $I_K \subset \mathrm{int}(\mathrm{CvxH}\,(Z_K))$. Hence $\mathrm{CvxH}\,(V) \subset \mathrm{CvxH}\,(\mathrm{int}(\mathrm{CvxH}\,(Z_K))) = \mathrm{int}(\mathrm{CvxH}\,(Z_K))$. It follows that c lies entirely in the interior of $\mathrm{CvxH}\,(Z_K)$, which is a contradiction with $y \in \partial \mathrm{CvxH}\,(Z_K)$.

If $c \in \mathrm{In}\,(K)$, then $\mathrm{Star}\,(c) \subset \mathrm{In}\,(K)$ since $\mathrm{In}\,(K)$ is open (see Lemma 1). Now $\overline{\mathrm{In}\,(K)} \subset \mathrm{Cl}\,(K)$. It follows that:

$$
\begin{aligned}
& \mathrm{CvxH}\,(\mathrm{Star}\,(c)) \subset \mathrm{CvxH}\,(\mathrm{Cl}\,(K)) = \mathrm{CvxH}\,(Z_K) && (\mathrm{CvxH}\,(\cdot) \text{ is increasing}) \\
\Rightarrow\ & \mathrm{int}(\mathrm{CvxH}\,(\mathrm{Star}\,(c))) \subset \mathrm{int}(\mathrm{CvxH}\,(Z_K)) && (\text{Taking their interior}) \\
\Rightarrow\ & \mathrm{CvxH}\,(\mathrm{Star}\,(c)) \subset \mathrm{int}(\mathrm{CvxH}\,(Z_K)) && (\text{since } \mathrm{Star}\,(c) \text{ is open}) \\
\Rightarrow\ & c \subset \mathrm{int}(\mathrm{CvxH}\,(Z_K)) && (\text{since } c \in \mathrm{Star}\,(c))
\end{aligned}
$$

There is again a contradiction since $y \in c$ and $y \in \partial \mathrm{CvxH}\,(Z_K)$.

We have just shown that $\mathrm{Cover}\,(F) \subset (\mathcal{C}^d \setminus (\mathrm{Outer}\,(K) \cup \mathrm{In}\,(K) \cup \mathrm{Inner}\,(K)))$, hence $\mathrm{Cover}\,(F) \subset (\mathrm{Out}\,(K) \cup \mathrm{Bd}\,(K))$ by Lemma 1. Furthermore, $\partial \mathrm{CvxH}\,(Z_K)$ is the union of finitely many convex facets (F_j). Thus $\mathrm{Cover}\,(\partial \mathrm{CvxH}\,(Z)) = \mathrm{Cover}\,(\cup_j F_j) = \cup_j \mathrm{Cover}\,(F_j) \subset \mathrm{Out}\,(K) \cup \mathrm{Bd}\,(K)$. □

It immediately implies the tangency of convex hull facets.

Corollary 1. *Let* $K \subset \mathcal{C}^d_d$. *If* X_K *is fully convex, then the vertices of any facet of* $\mathrm{CvxH}\,(Z_K)$ *are cotangent in* Z_K *(i.e.* $\mathrm{Star}\,(\partial \mathrm{CvxH}\,(Z_K)) \subset \mathrm{Star}\,(Z_K)$*).*

Proof. It suffices to notice than $\mathrm{Star}\,(Z_K) = \mathrm{Out}\,(K) \cup \mathrm{Bd}\,(K) \cup \mathrm{In}\,(K)$. □

B Properties related to probing

In this section, we focus on the termination of algorithm IsExtremal. We first show that it returns Yes if $\mathbf{0}$ is a vertex of $\mathrm{CvxH}\,(Z)$ and then show that it terminates for any finite Z, whatever the relative position of $\mathbf{0}$.

Theorem 3. *Let* $Z \subset \mathbb{Z}^3$ *be a digital set containing* $\mathbf{0}$ *and let* $\mathbf{I}$ *be a valid initial tetrahedron. If* $\mathbf{0}$ *is a vertex of* $\mathrm{CvxH}\,(Z)$, *Algorithm* IsExtremal *returns* Yes *after a number of iterations bounded from above by* $2\sqrt{3}$ *times the total area of the facets of* $\mathrm{CvxH}\,(Z)$ *incident to* $\mathbf{0}$.

Proof. First, note that since $\mathbf{I}$ is assumed to be a valid tetrahedron, $-\mathbf{e}_1, -\mathbf{e}_2, -\mathbf{e}_3$ and $\mathbf{0}$ are all in Z by Definition 5, which means that $\mathrm{CvxH}(Z)$ is full-dimensional.

Every facet $\mathcal{F}$ of $\mathrm{CvxH}(Z)$ incident to $\mathbf{0}$ has an orthogonal vector $\mathbf{v}_\mathcal{F} \in \mathbb{Z}^3$ and an area $a_\mathcal{F}$ that satisfy $\|\mathbf{v}_\mathcal{F}\|_2 \leq 2a_\mathcal{F}$. Denoting by $\mathbf{v}$ (resp. a), the sum of $\mathbf{v}_\mathcal{F}$ (resp. $a_\mathcal{F}$) over all facets $\mathcal{F}$ incident to $\mathbf{0}$, we have again $\|\mathbf{v}\|_2 \leq 2a$. Therefore, $\|\mathbf{v}\|_1 \leq \sqrt{3}\|\mathbf{v}\|_2 \leq 2\sqrt{3}a$.

We denote by $\mathcal{H}(\mathbf{v})$ the half-space $\{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}\cdot\mathbf{v} < 0\}$. By construction, $\mathcal{H}(\mathbf{v})$ contains $\mathrm{CvxH}(Z) \setminus \{\mathbf{0}\}$, therefore $Z \setminus \{\mathbf{0}\}$. Note that two nonzero points $\mathbf{x}, -\mathbf{x}$ cannot be both in $\mathcal{H}(\mathbf{v})$, therefore cannot be both in Z. Consequently, algorithm IsExtremal, if it terminates, cannot return No but only Yes in that case.

The algorithm will stop before crossing the boundary of $\mathcal{H}(\mathbf{v})$, while the maximum number of iterations to reach it can be determined through a decreasing sequence of integers (see also [10, Lemma 5, Lemma 6, Theorem 1]).

More formally, we denote by $\mathbf{M}$ the current tetrahedron and c_1, c_2, c_3 the entries of $\mathbf{v}^\top\mathbf{M}$. At initialization, $\mathbf{M} = \mathbf{I}$ and c_1, c_2, c_3 are just the coordinates of $\mathbf{v}$. Since $\mathbf{M}$ is a valid tetrahedron (1), we have for all $k \in \{1, 2, 3\}$, $-\mathbf{m}_k \in Z$, which implies $-\mathbf{m}_k \in \mathcal{H}(\mathbf{v})$. Therefore, $(-\mathbf{m}_k) \cdot \mathbf{v} < 0$ and, equivalently, $c_k > 0$.

If there is a valid operation f, we denote by $\mathbf{M}' := f(\mathbf{M})$ the new tetrahedron. There is a matrix $\boldsymbol{U}_\sigma$ such that $\mathbf{M}' = \mathbf{M}\boldsymbol{U}_\sigma$ and by permutations, we assume w.l.o.g. that σ is the identity. We thus have $\mathbf{v}^\top\mathbf{M}' = (c_1 - c_2, c_2, c_3)$. In addition, since f is valid (2), we have $-\mathbf{m}_1 + \mathbf{m}_2 \in Z$, which implies $-\mathbf{m}_1 + \mathbf{m}_2 \in \mathcal{H}(\mathbf{v})$. Therefore, $(-\mathbf{m}_1 + \mathbf{m}_2) \cdot \mathbf{v} < 0$ and, equivalently, $c_2 - c_1 > 0$. In other words, the entries c_1, c_2, c_3 are strictly positive integers and their sum is strictly decreasing from $\|\mathbf{v}\|_1$ at every iteration. As a consequence, the number of iterations is less than $\|\mathbf{v}\|_1$, which concludes the proof. □

Theorem 4. *Let $Z \subset \mathbb{Z}^3$ be a* finite *digital set containing $\mathbf{0}$ and let $\mathbf{M}$ be a valid initial tetrahedron. Algorithm* IsExtremal *terminates after a finite number of iterations.*

Proof. Let $C_\mathbf{M}$ be the cone formed by all convex combinations of $-\mathbf{m}_1$, $-\mathbf{m}_2$, $-\mathbf{m}_3$, i.e., $\{-\sum_{k\in\{1,2,3\}} c_k\mathbf{m}_k \mid c_1, c_2, c_3 \in \mathbb{R}_{\geq 0}\}$. If there is a valid operation f, we denote by $\mathbf{M}' := f(\mathbf{M})$ the new tetrahedron. There is a matrix $\boldsymbol{U}_\sigma$ such that $\mathbf{M}' = \mathbf{M}\boldsymbol{U}_\sigma$ and by permutations, we assume w.l.o.g. that σ is the identity.

For all $\mathbf{x} \in C_\mathbf{M}$, we have by definition $\mathbf{x} = -c_1\mathbf{m}_1 - c_2\mathbf{m}_2 - c_3\mathbf{m}_3$ with nonnegative c_1, c_2, c_3. We can write also

$$\begin{aligned}\mathbf{x} = -c_1'\mathbf{m}_1' - c_2'\mathbf{m}_2' - c_3'\mathbf{m}_3' &= -c_1'(\mathbf{m}_1 - \mathbf{m}_2) - c_2'\mathbf{m}_2 - c_3'\mathbf{m}_3 \\ &= -c_1'\mathbf{m}_1 - (c_2' - c_1')\mathbf{m}_2 - c_3'\mathbf{m}_3,\end{aligned}$$

for any $c_1', c_2', c_3' \in \mathbb{R}$. The nonnegativity of c_1, c_2, c_3 implies $c_1', c_3' \geq 0$ and $c_2' \geq c_1' \geq 0$, i.e. the nonnegativity of c_1', c_2', c_3'. Thus, $\mathbf{x} \in C_{\mathbf{M}'}$.

As a result, $C_{\mathbf{M}'}$ contains $C_\mathbf{M}$. By induction, the new cone $C_{\mathbf{M}'}$ also contains all past cones and therefore, all vertices of past tetrahedra. Consequently, Algorithm IsExtremal selects at every iteration a point $z \in Z$ to be a vertex of the next tetrahedron and this point cannot be selected again afterwards. Since Z is finite, Algorithm IsExtremal terminates after a finite number of iterations. □