

Minimal decomposition of a digital surface into digital plane segments is NP-hard

Isabelle Sivignon David Coeurjolly

Laboratoire LIRIS - Université Claude Bernard Lyon 1
Bâtiment Nautibus - 8, boulevard Niels Bohr
69622 Villeurbanne cedex, France
`isabelle.sivignon@liris.cnrs.fr`
`david.coeurjolly@liris.cnrs.fr`

Abstract

This paper deals with the complexity of the decomposition of a digital surface into digital plane segments (DPS for short). We prove that the decision problem (does there exist a decomposition with less than k DPS ?) is NP-complete, and thus that the optimisation problem (finding the minimal number of DPS) is NP-hard. The proof is based on a polynomial reduction of any instance of the well-known 3-SAT problem to an instance of the digital surface decomposition problem. A geometric model for the 3-SAT problem is proposed.

1 Introduction

Digital objects are defined as sets of grid points in \mathbb{Z}^n . Those objects carry redundant geometrical information due to their discrete structure: an object is represented as a set of elementary cells (called pixels in 2D, voxels in 3D). The definition of digital linear structures like digital lines [12] and digital planes [10, 2] originated a lot of works dealing with the decomposition of the contour of a digital object into digital linear primitives. Such a decomposition actually apprehends global geometrical properties of those objects. Many decomposition strategies may be designed and the number of parts computed by the algorithms may be a first criterion to compare the results. In this work, we focus on the complexity of the optimal (minimal number of parts) decomposition problem. In the 2D case, it has been shown that decomposing a digital curve into digital line segments can be solved in linear time [7]. In the 3D case of surfaces, many decomposition algorithms have been proposed [8, 6, 13, 14], offering comparisons on the number of faces recognized by different algorithms. Nevertheless, no optimality results exist, and no complexity study has been carried out. Related results have been recently proposed in [3] concerning the NP-completeness of the construction of an integer lattice polyhedron P with minimal number of convex facets such that $P \cap \mathbb{Z}^3$ corresponds to the input 3D digital object.

In computational geometry, the decomposition of a shape (*e.g.* a polygon) into a minimal number of elements (*e.g.* convex polygons) usually leads to NP-complete problems [9]. A problem is in the NP class of algorithms if it can only be solved in polynomial time by a non-deterministic machine [5]. In other words, if a problem is NP-complete, no efficient in time solution exists to solve it. The remaining option is to consider approximation algorithms with or without heuristics.

Prior to a complexity study, the problem has to be formalized. In the sequel, we consider 6-connected sets of voxels which surface \mathbb{S} is defined as the set of object voxels sharing a face with the background. The surface is a set of 18-connected voxels, and maximal digital naive planes [11, 1, 10, 2] are used for the decomposition. A digital plane segment (DPS for short) is maximal if no surface voxel may be added to it. In the following, we consider a sequential decomposition algorithm: given a voxel on the surface (called a *seed*), we construct the maximal digital naive plane segment adding iteratively voxels that are 18-connected to the DPS initialized with the seed. Then, a new seed is considered from the set of remaining voxels in \mathbb{S} . In this algorithm, both the propagation process in the maximal DPS recognition process and the initialization of seeds must be taken into account.

The optimization problem we consider is defined as follows:

Min-DSD (Digital Surface Decomposition): Given a digital object surface \mathbb{S} , find the minimal decomposition of \mathbb{S} into maximal digital naive plane segments using a sequential algorithm.

In order to study the complexity of an optimisation problem, the related decision problem has to be considered:

k-DSD: Given a digital object surface \mathbb{S} and a number $k \in \mathbb{N}^*$, does there exist a decomposition of \mathbb{S} into k maximal digital naive plane segments using a sequential algorithm.

In this article, we prove that k -DSD is in NP-complete whatever the propagation heuristic used and the initialization of seeds. Furthermore, the only requirement on the digital plane segments topology is connectivity.

To prove that a problem \mathcal{P} is NP-competete, a classical scheme is to exhibit a polynomial reduction of all instances of a classical NP-problem, denoted \mathcal{P}_{NP} into an instance of \mathcal{P} . Then, we have to prove that a solution of \mathcal{P} also leads to a solution of \mathcal{P}_{NP} . Since \mathcal{P}_{NP} is known to be NP-complete, we could conclude that \mathcal{P} is also in NP [5]. In the literature, the Boolean Satisfiability Problem (SAT) is a decision problem classically used in complexity theory since it was the first known NP-complete problem. An instance of SAT is a boolean expression written using only AND, OR, NOT, variables and parentheses. The decision problem is: given an expression, is there an assignment of the variables such that the expression is TRUE ? The problem remains NP-complete even if the expression is written in conjunctive normal form with three variables per clause, yielding the 3-SAT problem. An expression ϕ has the form:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (\neg x_6 \vee x_3 \vee \neg x_5) \wedge \dots, \quad (1)$$

where each x_i is a binary variable (and $\neg x_i$ its negation) can appear several

times in the expression.

In the following we define a reduction of any instance of the 3-SAT problem to an instance of the k-DSD problem. The construction process, defining geometrical objects for variables, variable instances and clauses, is presented in Section 2, while the NP-completeness proof derived from this construction is given in Section 3.

2 A geometric model for 3-SAT

Given a 3-SAT expression ϕ , we show how to construct a geometric discrete object. This construction is a two steps process:

- first define geometric elements for variables, instances of variables and clauses;
- then define how those basic components are organized in the 3D space.

2.1 General considerations

All the basic elements we define further are composed of two parts:

- construction part: the surface is made of planes parallel to axis planes and only aims at defining a 6-connected object. The minimal number of digital planes needed to cover the construction part is fixed for each basic element, and thus do not take part in the optimization problem;
- optimization part: remaining part after the computation of maximal construction planes. It takes advantage of digital planes properties to geometrically encode the 3-SAT elements.

For each basic element, we provide an illustration¹ of an example set of seeds (black voxels) which may be used in a decomposition algorithm. Moreover, those sets have the remarkable property that any two seeds cannot be covered by a single DPS.

The underlying basic idea for this construction is the following: the optimal decomposition chosen for a variable object generates a “signal” sent to clause objects through “wires” representing instances of variables. This kind of geometric construction of 3-SAT is a classical way to prove NP-completeness of geometric problems (see [15, 4] for instance). The construction of basic elements relies on several properties of digital planes structure that we set forth here (see Figure 1 for illustrations):

Property 1 *For the three configurations (a), (b) and (c) represented in Figure 1, one DPS cannot simultaneously cover the two light-coloured voxels and all*

¹Most illustrations of this paper are originally color artworks. To make the understanding of the paper easier from B&W printings, color images are available on <http://liris.cnrs.fr/isabelle.sivignon/SatDSD.html>

the other ones. In the configuration represented in Figure 1(d), the three voxels cannot be covered by one DPS, but any two voxels can.

Proof 1 The proofs of those properties are straight forward using either digital naive plane structural basic properties or their arithmetical definition [11, 6, 10].

In the following, we refer to those configurations as Property 1(a), (b), (c) and (d).

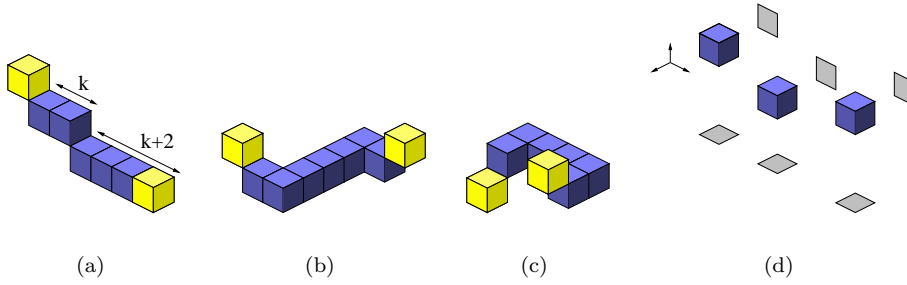


Figure 1: Four configurations used for the reduction process.

2.2 Variable and Clause objects

An illustration of a variable element is given in Figure 2. Optimal decompositions are represented in Figure 2 (b) and (c): five DPS are used to cover the construction part of the object, and two more DPS are required to cover the remaining optimization part (upper part on the figure, and see Property 1(d)). Regardless of the construction part decomposition, the only two optimal decompositions for the optimization part are those represented in 2 (b) and (c): the upper voxels are either covered by the left or the right part. Actually, any other decomposition is either not optimal or not composed of maximal DPS. Those two possibilities encode true and false assignments of the variable respectively. From those decomposition schemes and Property 1(d), seven seeds can be defined on each variable object.

Variable elements are linked to clauses thanks to wires that are connected on the area circled in red on Figure 2(a). Without loss of generality, positive instances of the variables are connected on the right part, whereas negative instances are connected on the left part. The length (along y axis) of the variable v depends on the the maximum number of positive or negative instances of v in an expression, so that all the connections can be made. Note that the length of the variable does not change the optimal number of DPS required for the decomposition.

A clause element is depicted in Figure 3. It is composed of a transversal plateau on which three terminals are plugged. Since each clause has three literals

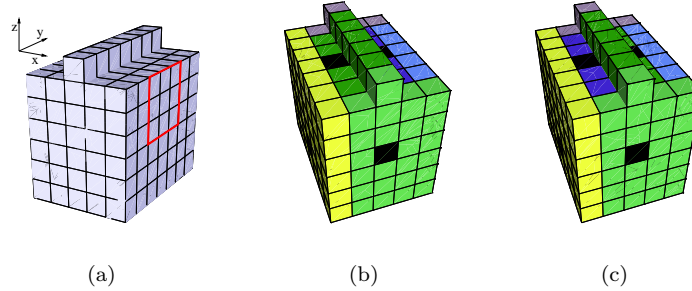


Figure 2: Geometric discrete object encoding a variable: (a) general view with wire plugging area, (b) truth assignment, (c) false assignment.

(recall that 3-SAT is considered) , each clause element has three incoming wires. Five construction planes are required in order to decompose five out of the six faces of the plateau. Construction planes related to the three terminals will be taken into account in the wire definition.

Concerning the upper face, which is actually the optimization part, and the three terminal extremities, Property 1(d) implies that the three terminals extremities cannot be covered by a single DPS. On the contrary, any couple of terminals can be entirely covered by a single DPS. An illustration of the optimal decomposition of a clause object, and more particularly of a DPS covering two out of the three terminals extremities is depicted in Figure 3(b). We see that the third terminal extremity is not covered.

From the optimal decomposition, we can define six seeds on each clause object surface.

The wires linking variable elements to a clause are plugged on the areas circled in red in Figure 3(a), one on each terminal extremity.

When several variables and clauses are considered, Figure 4 illustrates how these objects are put together in the 3D space: variables and clauses are lined up on two axis parallel to the y axis. The definition of wires connecting variables and clauses relies on this spatial construction.

2.3 Linking variables and clauses

Variable elements are connected to clause elements through wires, that represent variable instances: if a variable v appears a clause c , a wire links the geometric elements of v and c . Those wires aim at “transmitting” the truth value of a variable to the clause it belongs to. Before defining the geometric construction of wires, we describe the transmission process.

Figure 5 illustrates how the truth value of a variable is transmitted to a clause through a wire. This figure represents a vertical cut of the optimization part of a variable, a wire and a clause terminal. Figure 5(a) illustrates the propagation of a true assignment while Figure 5(b) shows how a false assignment

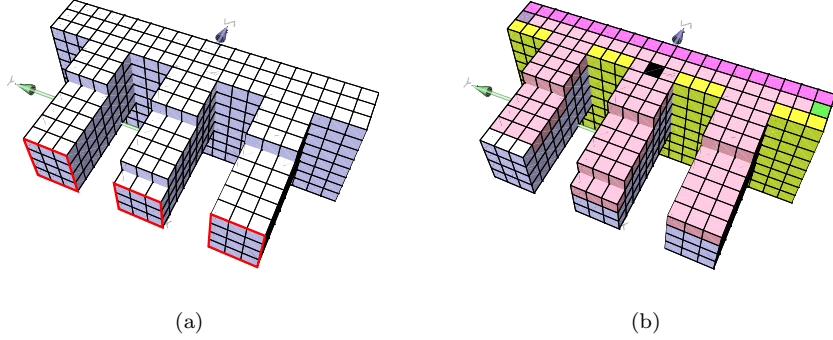


Figure 3: Geometric discrete object encoding a clause: (a) general view, (b) optimal decomposition.

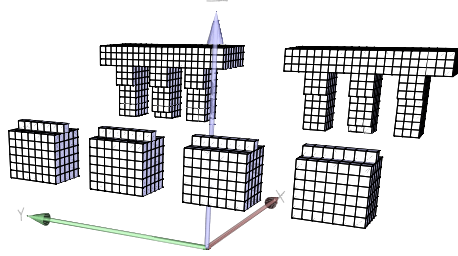


Figure 4: Positions of (four) variables and (two) clauses objects in the 3D space.

is transmitted to a clause. From the construction we propose, the vertical cut of a variable-wire-clause connection can be thought of as a 2D digital curve that we decompose in digital straight segments, using DSS properties.

We call “transmission voxels” the two voxel circled in red in Figure 5 (intermediate transmission voxels are circled in black). We consider an optimal decomposition of the surface into DPS. Using Property 1(a) and (b), if the left transmission voxel is covered by a variable plane, then the right transmission voxel (which is also an extremity of a clause terminal) is covered by a wire plane. On the contrary, if the left transmission voxel is not covered by a variable plane, then the right transmission voxel is not covered by a wire plane. Note that for the plateau, descent and ascent parts, the relative length of the steps are the key point of this transmission process: for instance, a single DPS cannot cover both the first and last voxel of the descent.

Since the left transmission voxel is covered by a variable plane if and only if the variable instance is set to the value “true” (Property 1(a)), the clause terminal extremity is covered by a wire plane if and only if the variable instance is set to the value “true”.

A variable v can appear as v or $\neg v$ in an expression ϕ , and we provide two

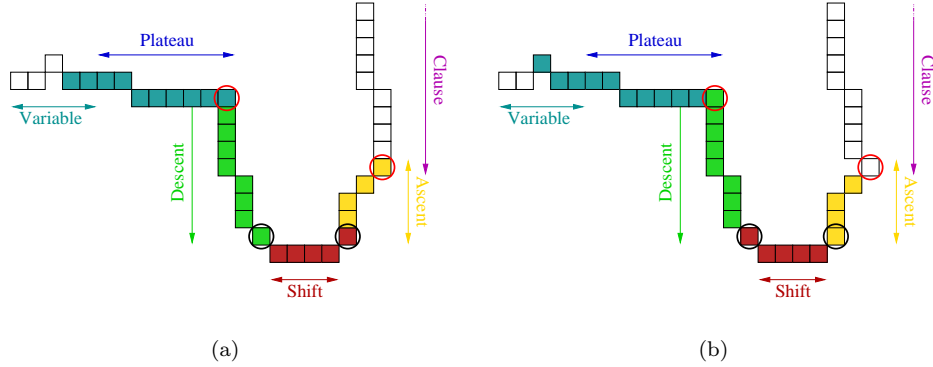


Figure 5: Vertical cut illustration of the transmission of a truth assignment through a wire: (a) the value “true” is transmitted; (b) the value “false” is transmitted.

types of wire for these two cases.

Following the spatial arrangement of variable and clause elements (see Figure 4), wires standing for positive instances are plugged on the variable side closest to clauses. On the contrary, wires corresponding to negative instances are plugged on the opposite side. We see that in the case of a negative wire, a U-turn towards clause objects is required.

Wires are basically composed of four parts, that are depicted in Figure 6(a) for a positive wire (see Figure 5 for an illustration of a vertical cut):

- a plateau (blue) generates the “signal” corresponding to the truth value of the variable;
- a descent (green) to a given level L : two distinct variable instances descend on two different levels;
- a shift movement (red) on the level L to reach the clause position;
- an ascent (yellow) from the level L to the clause terminal extremity.

The optimal decomposition of the plateau is made of one DPS only, for the construction part (bottom of the plateau). Indeed, the sides are covered with descent construction planes, and the upper part (optimization) is covered by a DPS coming from the variable decomposition. Figure 6(e) illustrates how a wire is connected to a variable.

Concerning the descent and ascent parts, optimal decompositions are depicted in Figure 6(b) and (c). Seven construction planes are required for both positive and negative wires. Moreover, the optimization part is a three steps surface such that the first step is k voxels long ($k \geq 3$), the second one is $k - 2$ voxels long and the third one is made of one voxel. In addition to the function

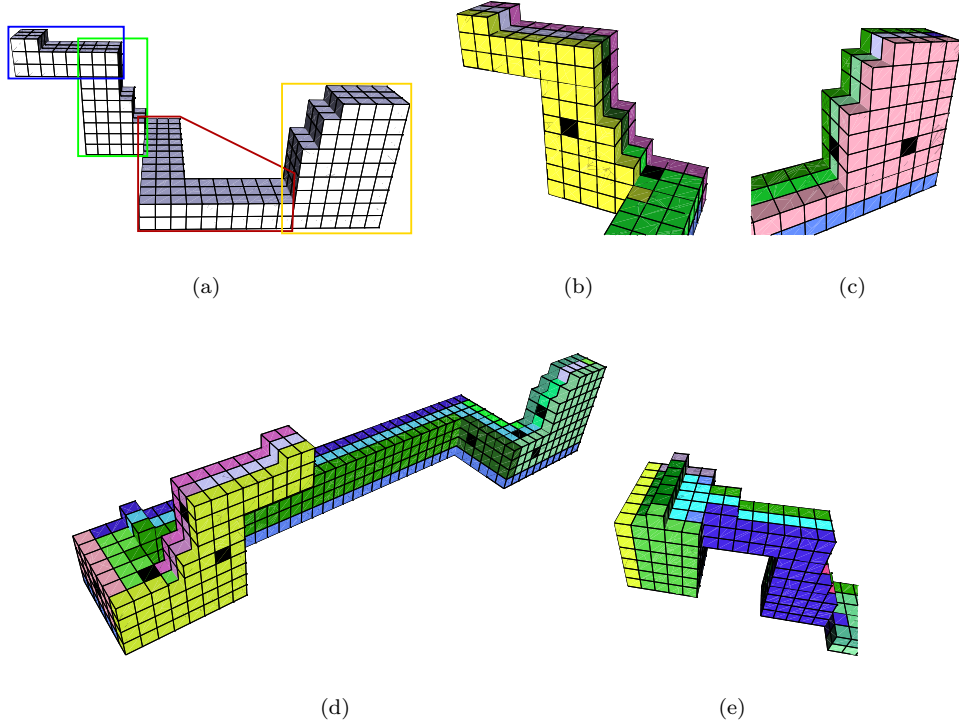


Figure 6: Wire between a variable and a clause: (a) case of a positive variable instance, with zooms on (b) the descent and (c) ascent parts; (d) case of a negative variable instance.

of this construction in the transmission process (see previous paragraph), the parameter k is used to ensure that every wire descend on a different level so that wires do not intersect.

The shift parts of positive and negative wires are different. In the case of a positive wire, only one DPS covers the optimization part (upper part). Five more construction planes are necessary for a negative wire, and two DPS are needed to cover the optimization part (see Figure 6(d)). The transmission of the truth value through the U-turn part of a negative wire is ensured thanks to Property 1(c).

All in all, optimal decomposition of wires leads to the definition of 11 seeds (eight for the construction part, three for the optimization part) for each positive wire and 17 (13 for the construction part, four for the optimization part) for each negative wire (see Figure 6(b), (c) and (d)).

2.4 Summary of the construction

We have proposed a polynomial reduction of any instance of the 3-SAT problem into an instance of the k-DSD problem. This reduction is based on the definition of variable and clause objects linked together through wires which pass the truth value of a variable on to clauses. The following section is dedicated to the proof of the relation between the two problems.

3 NP-completeness Proof

Let us consider a boolean 3-SAT expression ϕ and the corresponding discrete object surface \mathbb{S} . We denote c , v , v_p and v_n the number of clauses, variables, positive instances and negatives instances of variables in ϕ respectively.

Proposition 1 *k-DSD is in NP.*

Proof 2 Given a digital surface \mathbb{S} and solution D , verifying that $|D| \leq k$ and that it actually covers all the voxels of \mathbb{S} can easily be done in linear time in the number of voxels \mathbb{S} . \square

Proposition 2 *The size of \mathbb{S} is linear in the size of ϕ .*

Proof 3 The proof is straight forward considering the construction.

We shall now prove that the construction is a reduction of 3-SAT to k-DSD, *i.e.* that the expression ϕ is satisfiable if and only if \mathbb{S} admits a decomposition with k maximal DPS. We prove the two implications one after the other.

Lemma 1 *If the expression ϕ is satisfiable, then \mathbb{S} admits a decomposition with k maximal DPS.*

Proof 4 Assume that ϕ is satisfiable under some truth assignment T . The following algorithm builds a decomposition of the surface of \mathbb{S} into k maximal DPS:

1. label all the voxels belonging to a construction DPS regardless of T : $5v + 5c + 8v_p + 13v_n$ DPS are used to cover the entire construction part of \mathbb{S} ;
2. decompose each variable according to its truth assignment in T : this decomposition requires $2v$ DPS;
3. use $3v_p$ and $4v_n$ DPS to decompose the wires optimization parts, which may leave the tips of some wires (which are also the clause terminal extremities) uncovered;
4. since T satisfies ϕ , every clause has at least one incoming wire with a covered tip. Thus, every clause has at least one covered terminal extremity. Consequently, each clause optimization part can be covered with one single DPS.

All in all, $(5v + 5c + 8v_p + 13v_n) + 2v + 3v_p + 4v_n + c = 7v + 6c + 11v_p + 17v_n = k$ DPS are used in this decomposition. \square

In order to prove the reverse implication, we need to show that there is only one way of decomposing \mathbb{S} into k DPS. Next, we show that this unique solution leads to an satisfactory assignment of ϕ 's variables.

Lemma 2 *Consider a decomposition of \mathbb{S} with DPS with k DPS. Then the decompositions of variable, positive wire, negative wire and clause objects are respectively covered by 7, 11, 17, 6 DPS.*

Proof 5 Consider a decomposition D of \mathbb{S} with $|D| = k$. Suppose that there exist a variable object with a decomposition D_v such that $|D_v| > 7$. Extra planes are either construction or optimization planes. The fact D_v has more than five construction planes has no effect on the number of DPS required to cover other variables, wires and clauses. Thus, $|D| = 7(v - 1) + |D_v| + 6c + 11v_p + 17v_n > k$, which is a contradiction.

Now suppose that extra planes are used for the optimization part. With these DPS, one can at best ensure that the “true” value is transmitted to every clause objects linked to this variable. Nevertheless, the number of DPS required to cover wires and clauses does not change, and we still have $|D| = 7(v - 1) + |D_v| + 6c + 11v_p + 17v_n > k$, which is a contradiction.

On the contrary, if less than seven DPS are used to cover a variable object, it is easy to check that some voxels will remain uncovered even if more DPS are used for wires or clause objects.

Similar arguments can be used to show that positive and negative wires, and clause decompositions have to be composed of 11, 17 and 6 DPS. \square

Lemma 3 *If \mathbb{S} admits a decomposition into k maximal DPS, then ϕ is satisfiable.*

Proof 6 Suppose that \mathbb{S} admits a decomposition D with k DPS. Since $|D| = k$, from Lemma 2 the decomposition of every variable object is made of seven DPS. Variable objects can only be decomposed two ways into seven DPS, each of which encoding a truth assignment. This decomposition is made of 5 DPS for the construction part and 2 DPS for the optimization part (regardless of the algorithm used). Thus, covering all variables requires $7v$ DPS. In the same way, using Lemma 2 covering wires use $11v_p + 17v_n$ DPS. All in all, $k - 7v - 11v_p - 17v_n = 6c$ DPS remain for covering clause objects. The construction part of clause objects requires 5 DPS regardless of the rest of the decomposition. Thus c DPS remain to cover the clauses optimization parts. Since there are c clause objects, and c DPS remain, we know that the clause optimization parts are covered by one DPS only in D . This is possible if and only if every clause is satisfied, and thus ϕ is satisfied too. \square

Theorem 1 *k -DSD is an NP-complete problem.*

Proof 7 The result is derived from Lemma 1 and 3.

This theorem proves that the decision problem associated to Min-DSD is NP-complete. Thus, according to the theory of complexity, Min-DSD is said to be NP-hard.

4 Example

A software that generates a 3D object from a 3-SAT boolean expression is available on <http://liris.cnrs.fr/isabelle.sivignon/code.html>. This program also generates the seeds of the object, and a simple surface decomposition algorithm into maximal DPS is also provided to compute the decomposition derived from those seeds.

Figure 7 is an illustration of the digital surface encoding the expression $\phi = (a \vee \neg b \vee c)$. The optimal decomposition into maximal DPS is composed of 49 construction DPS and 17 optimization DPS. In Figure 7(a), the variable objects encode the assignment $(a = \text{true}, b = \text{true}, c = \text{false})$, and the optimal decomposition is represented. In Figure 7(b), the variable objects encode the assignment $(a = \text{false}, b = \text{true}, c = \text{false})$: in this case, since ϕ is not satisfied, the optimal decomposition cannot be achieved, and an extra DPS (in red) is added.

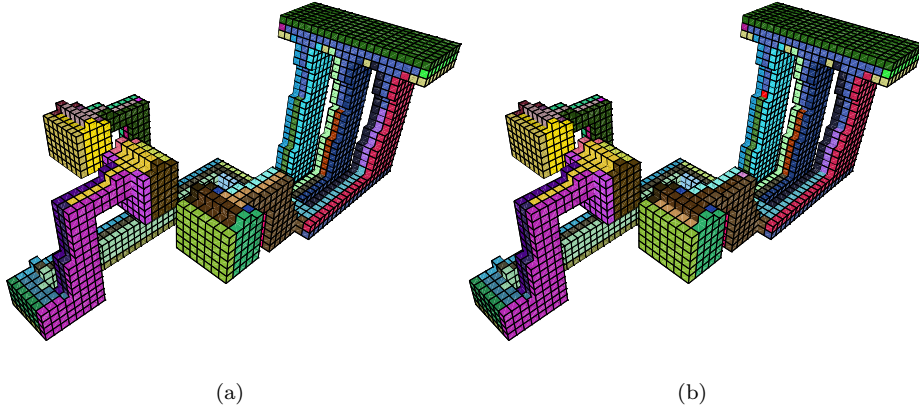


Figure 7: Discrete object encoding the expression $\phi = (a \vee \neg b \vee c)$: (a) optimal decomposition corresponding to the satisfaction of ϕ ; (b) ϕ is not satisfied and one more DPS is required to achieve a complete decomposition.

5 Conclusion and future works

In this article, we have proved that the decomposition of a digital object surface into a minimal number of maximal DPS using a sequential algorithm is NP-complete. This theoretical result concludes an important open problem in the discrete geometry community: no efficient algorithms exist to solve the Min-DSD problem. A logical consequence of this answer is that only heuristics can be used.

Among possible heuristics, important theoretical future works exist: does it exist a polynomial-time approximation scheme for the Min-DSD problem ? More precisely, is there a polynomial in time approximation of Min-DSD that produces a solution that is within ϵ factor of being optimal ?

By construction of variables, clauses and links, the genus of the obtained binary object depends on the number of cycle in the 3-SAT instance. If a hole-free object is considered, is k -DSD still in NPC, and thus Min-DSD still NP-hard ?

References

- [1] E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, 59(5):302–309, September 1997.
- [2] V. Brimkov, D. Coeurjolly, and R. Klette. Digital planarity - a review. Technical report, Laboratoire LIRIS, Université Claude Bernard Lyon 1, 2004. <http://liris.cnrs.fr/publis/?id=1933>.
- [3] Valentin Brimkov. Discrete volume polyhedrization is strongly NP-hard. Technical report, <http://www.citr.auckland.ac.nz/techreports/show.php?id=179>, March 2006.
- [4] Bernard Chazelle, David P. Dobkin, Nadia Shouraboura, and Ayellet Tal. Strategies for polyhedral surface decomposition: An experimental study. In *Symposium on Computational Geometry*, pages 297–305, 1995.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.
- [6] I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets*. PhD thesis, Université Louis Pasteur, 1995.
- [7] F. Feschet and L. Tougne. On the min dss problem of closed discrete curves. *Discrete Applied Mathematics*, 151(1-3):138–153, oct 2005.
- [8] J. Françon and L. Papier. Polyhedrization of the boundary of a voxel object. In *8th International Conference in Discrete Geometry for Computer Imagery*, volume 1568 of *LNCS*, pages 425–434. Springer-Verlag, 1999.
- [9] J. E. Goodman and J. O’Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

- [10] R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Series in Computer Graphics and Geometric Modelin. Morgan Kaufmann, 2004.
- [11] J.-P. Reveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Pasteur - Strasbourg, 1991.
- [12] A. Rosenfeld and R. Klette. Digital straightness. In *Int. Workshop on Combinatorial Image Analysis*, volume 46 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, August 2001.
- [13] I. Sivignon, F. Dupont, and J. M. Chassery. Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1):25–43, 2003.
- [14] I. Sivignon, F. Dupont, and J.-M. Chassery. Reversible polygonalization of a 3D planar discrete curve: Application on discrete surfaces. In *International Conference on Discrete Geometry for Computer Imagery*, pages 347–358, 2005.
- [15] Chris Worman. Decomposing polygons into diameter bounded components. In *Proceedings of the 15th Canadian Conference on Computational Geometry (CCCG'03)*, pages 103–106, 2003.