# Reversible discrete volume polyhedrization using Marching Cubes simplification

**Article** *in* Proceedings of SPIE - The International Society for Optical Engineering · April 2004

DOI: 10.1117/12.525391



David Coeurjolly
French National Centre for Scientific Research


Isabelle Sivignon
French National Centre for Scientific Research

# Reversible discrete volume polyhedrization using Marching Cubes simplification

David Cœurjolly, Alexis Guillaume, Isabelle Sivignon

{dcoeurjo, alexis.guillaume}@liris.cnrs.fr, sivignon@lis.inpg.fr.

LIRIS (Lyon), LIS (Grenoble) - France

# Introduction

Discrete volumes ⇒exploitation and study are difficult:

- huge volume of data
- facet structure

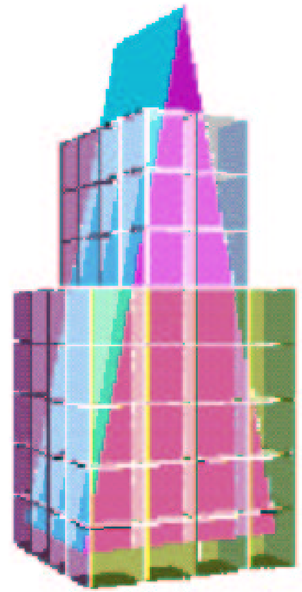
*Problem:* how to transform a discrete volume into a Euclidean Polyhedra ?

- topologically correct surface
- reversibility property

# State of the art

Digital plane segmentation : *Ok*

Digital Polyhedrization: *[Debled-Rennesson] [Vittone] [Klette] [Sivignon]*



*⇒ No method exists to ensure both the correct topology and the reversibility of the edges and the vertices of the surface*

# Introduction

Two approaches:

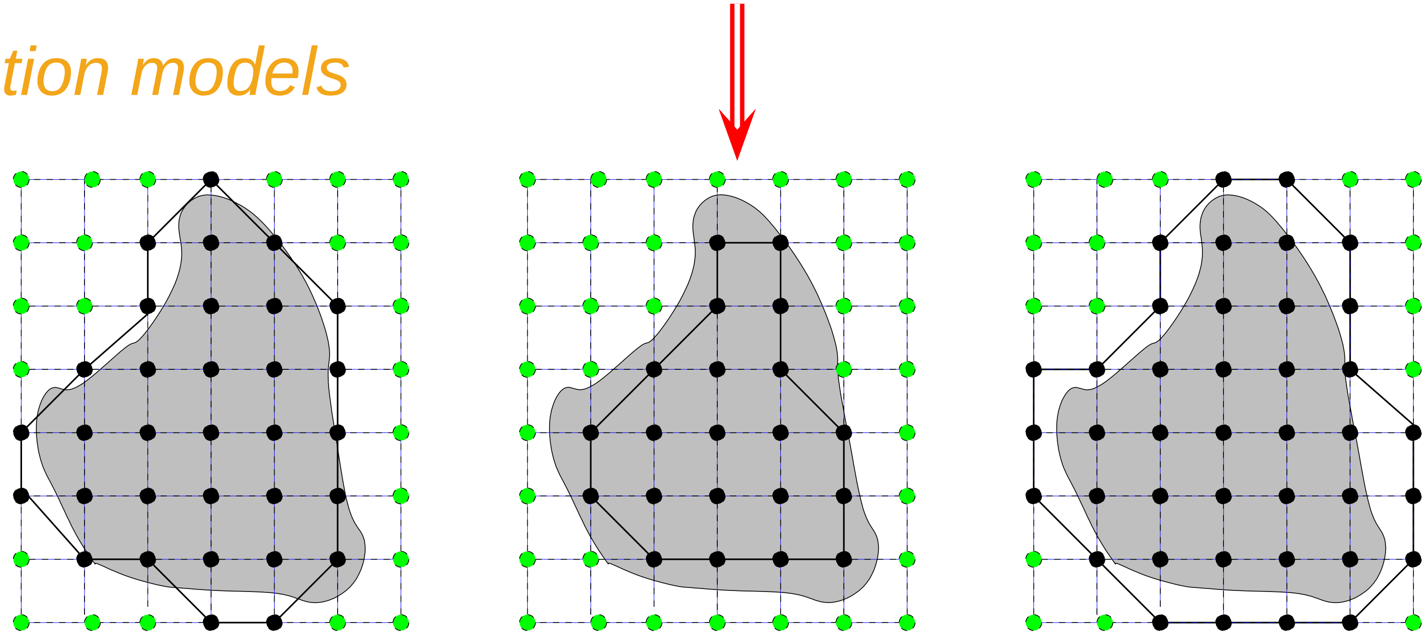
- Marching-Cubes algorithms: compute a triangulated reversible surface. Huge number of facets but reversible solution.
- Digital geometry solutions: segment the digital surface into pieces of digital planes, and then reconstruct a surface from this information. Hard to ensure both reversibility and correct topology.

*Idea:* Combine the two processes in order to decrease the number of facets of the MC triangulation.

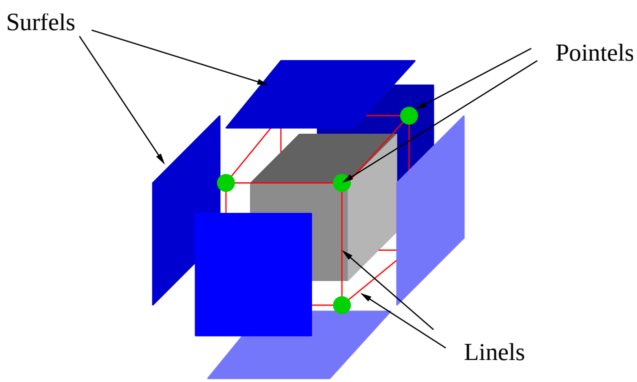
# Contents

# Preliminaries

## *Digitization models*



## *Cellular decomposition of $\mathbb{Z}^3$* (Digital surface = set of oriented surfels)



# Marching-Cubes algorithms

*Problem* : Given a density function $V : \mathbb{Z}^3 \rightarrow \mathbb{R}$, how to extract a triangulated iso-surface ?

# Marching-Cubes algorithms

*Problem* : Given a density function $V : \mathbb{Z}^3 \rightarrow \mathbb{R}$, how to extract a triangulated iso-surface ?

$\Rightarrow$ *Marching-Cubes algorithm* *[Lorensen-Cline 87]*

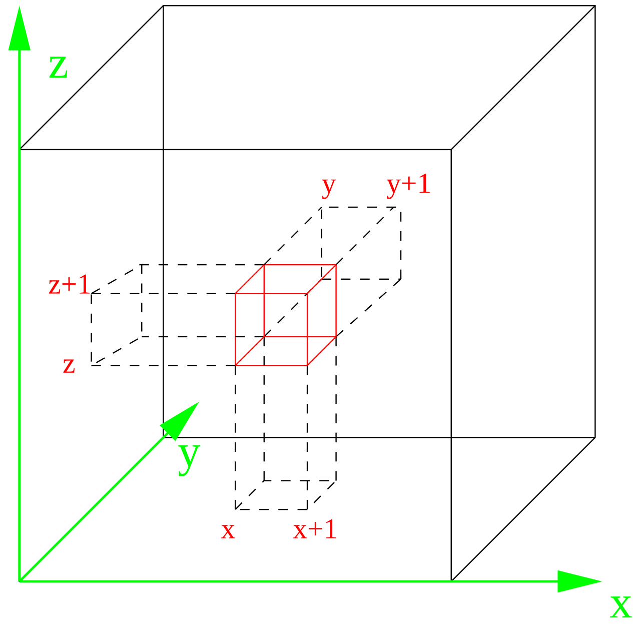
# Marching-Cubes algorithms

*Step 1: Cubic cell decomposition*
*Step 2: Local configurations*

*Step 3: Cubic cell displacement*

# Marching-Cubes algorithms

*Step 1: Cubic cell decomposition*

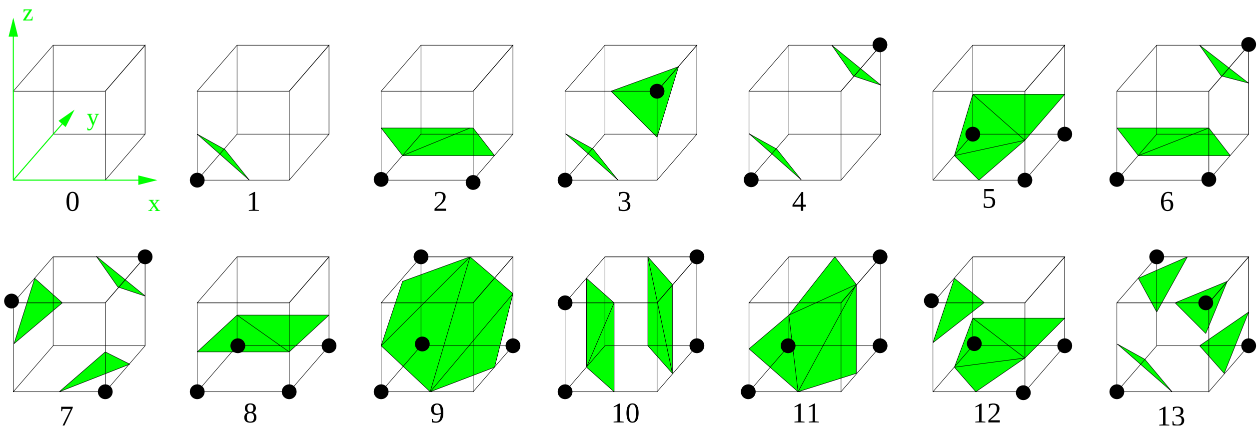



*Step 2: Local configurations*

*Step 3: Cubic cell displacement*

# Marching-Cubes algorithms

## *Step 1: Cubic cell decomposition*
## *Step 2: Local configurations*

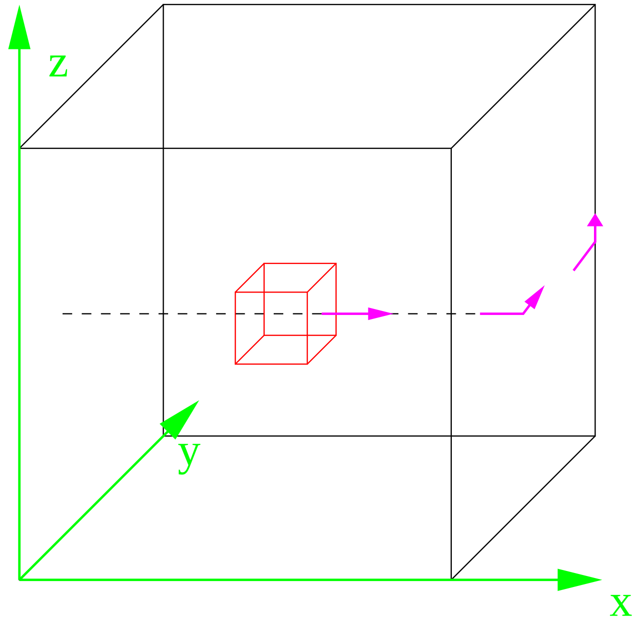



*+ Interpolation processes*

## *Step 3: Cubic cell displacement*

# Marching-Cubes algorithms

*Step 1: Cubic cell decomposition*
*Step 2: Local configurations*
*Step 3: Cubic cell displacement*

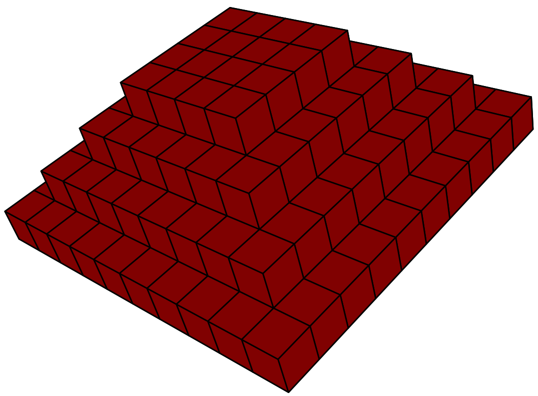


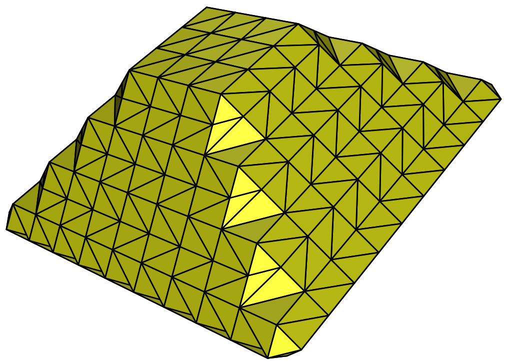
# Marching-Cubes global properties

Lemma 1 The triangulated surface is closed, oriented and without self-crossing *[Lachaud 96]*

Lemma 2 Let $V : \mathbb{Z}^3 \rightarrow \{0, 1\}$ be a binary object, and a threshold in $[0, 1]$. The Marching-Cubes surface is a reversible polyhedrization of the binary object according to the Object Boundary Quantization model.

Lemma 3 The MC vertices and boundary surfel centers coincide.





# Digital plane recognition

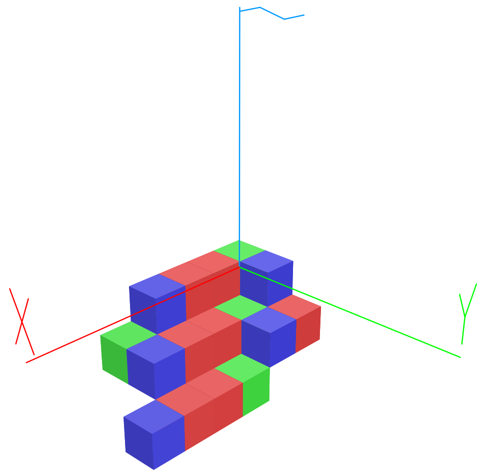
*Problem:* Given a set of voxels $\mathcal{V}$, does there exist a plane $P$ which digitization contains $\mathcal{V}$ ?

Many solutions:

- geometrical properties *[Stojmenovic-Tosic91] [Kim-Stojmenovic91] [Veelaert93]*
- arithmetical definition *[Debled95]*
- linear programming framework *[Francon et al.96] [Buzer02] [Vittone00]*

# Digital plane recognition

$\mathcal{V}$ =

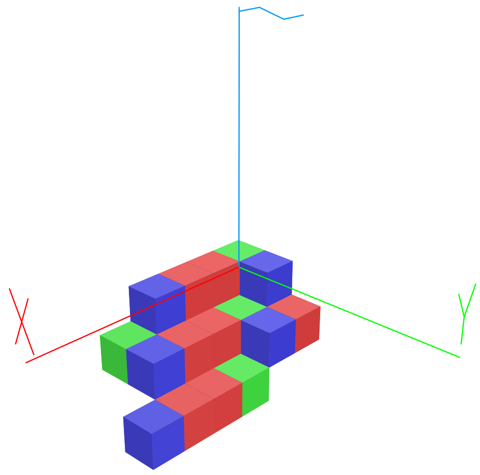



$\mathcal{V}$ is a piece of digital plane $\Rightarrow$there exist $(\alpha, \beta, \gamma)$ such that
$\mathcal{V} \subset \{(x, y, z) \in \mathbb{Z}^3 \mid 0 \leq \alpha x + \beta y + \gamma + z < 1\}$

Set of parameters $(\alpha, \beta, \gamma)$ = Preimage of $\mathcal{V}$
= Intersection of linear constraints.

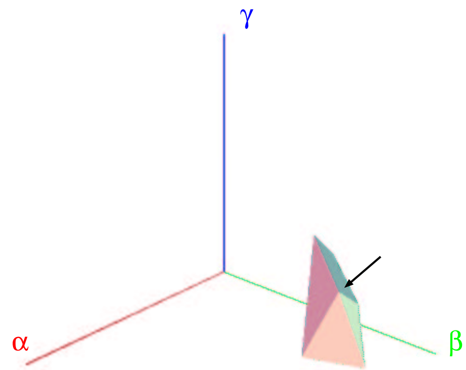
# Digital plane recognition

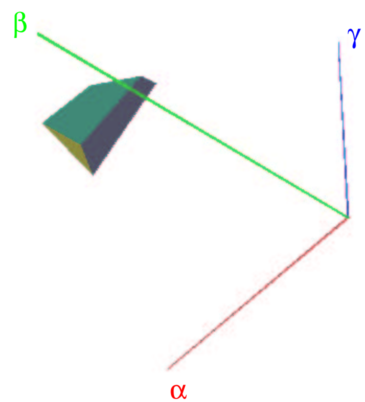
$\mathcal{V}$ =



$\mathcal{V}$ is a piece of digital plane $\Rightarrow$there exist $(\alpha, \beta, \gamma)$ such that

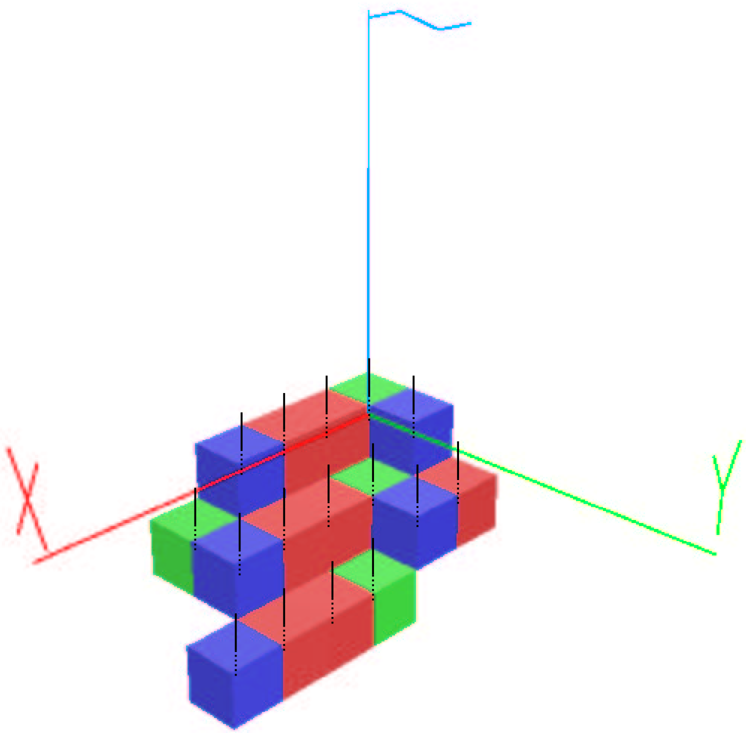
$$\mathcal{V} \subset \{(x, y, z) \in \mathbb{Z}^3 \mid 0 \leq \alpha x + \beta y + \gamma + z < 1\}$$



# Directional recognition algorithm

Any solution plane of this preimage crosses the segments $[p, p+d[$ where $d = (0, 0, 1)$.




$\Rightarrow$ Set of directions $D = \{(1,0,0),(0,1,0),(0,0,1),(-1,0,0),(0,-1,0),(0,0,-1)\}$

# Directional recognition algorithm

*Directional recognition algorithm:*
The directional recognition algorithm in direction $d$ on $\mathcal{V}$ computes the set of Euclidean planes that cross all the segments $[pq[$ where $p \in \mathcal{V}$ and $q$ is equal to $p + d$.

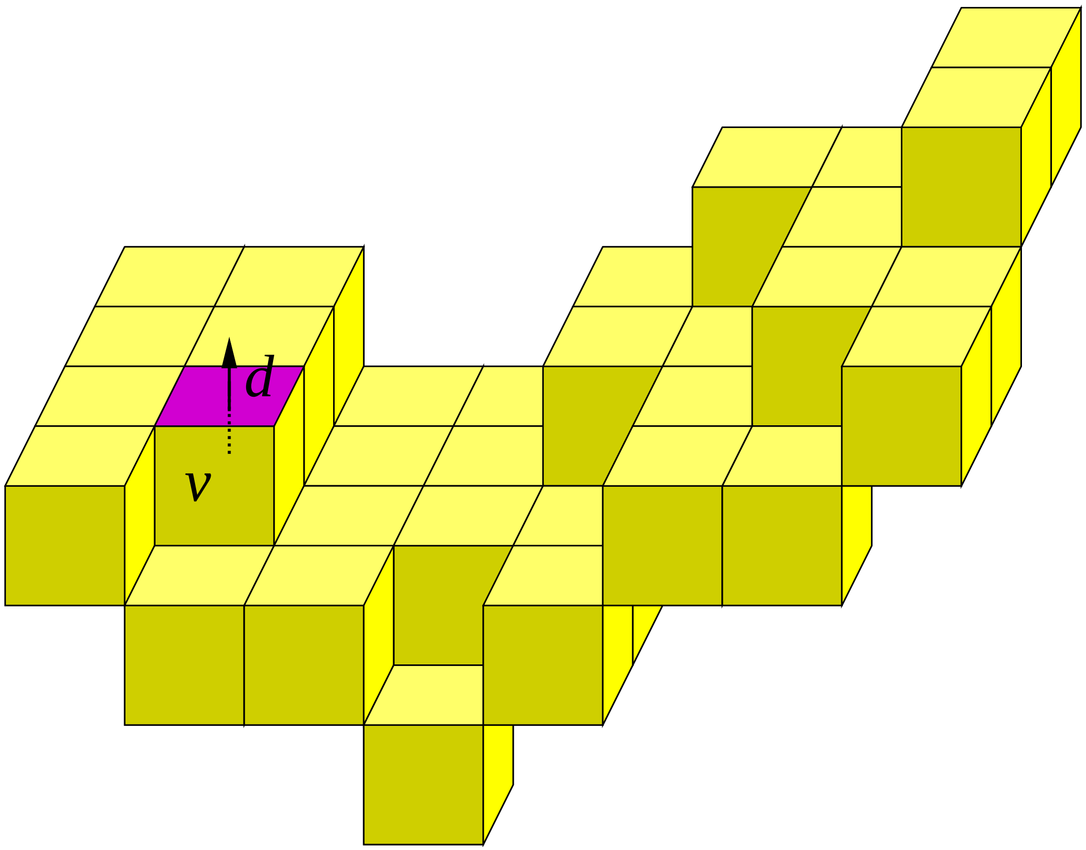
*Nota Bene:* $\Rightarrow$6 preimages for $\mathcal{V}$.
$\mathcal{V}$ is a piece of digital plane $\Leftrightarrow$ one out of the 6 preimages is not empty.

# Digital plane segmentation

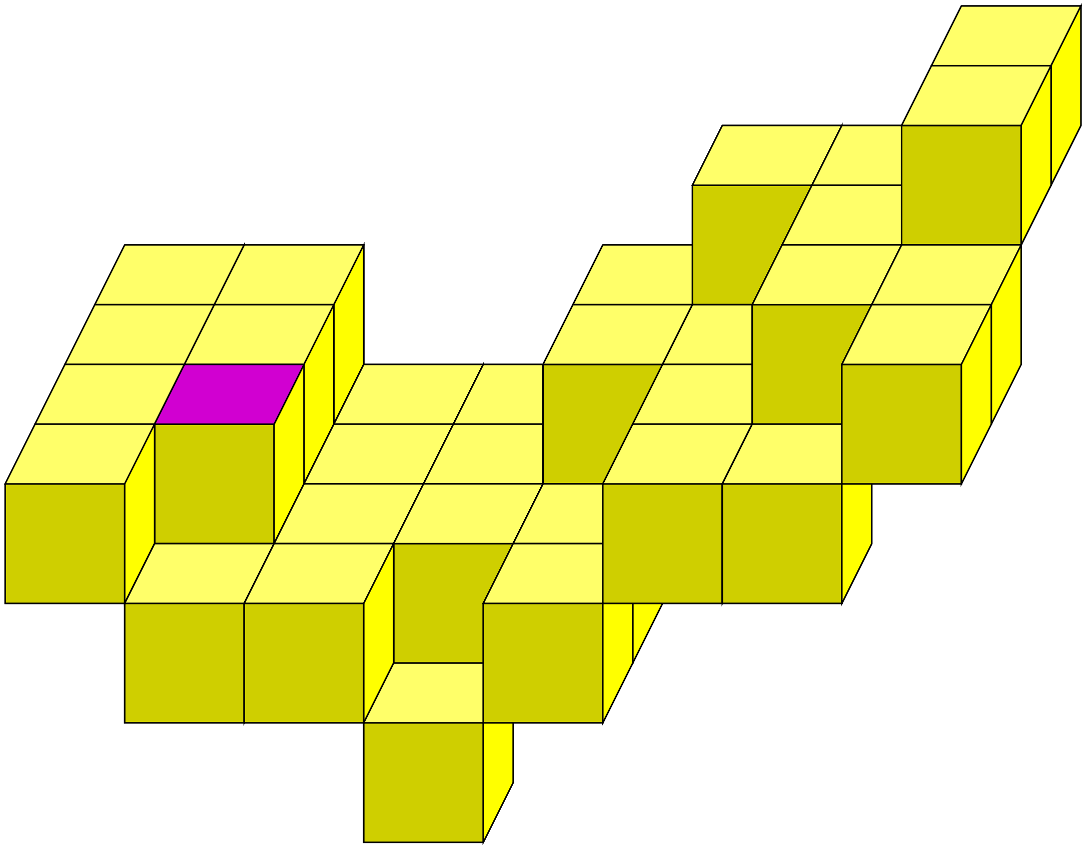
For each direction $d$

For each unlabelled voxel $v$

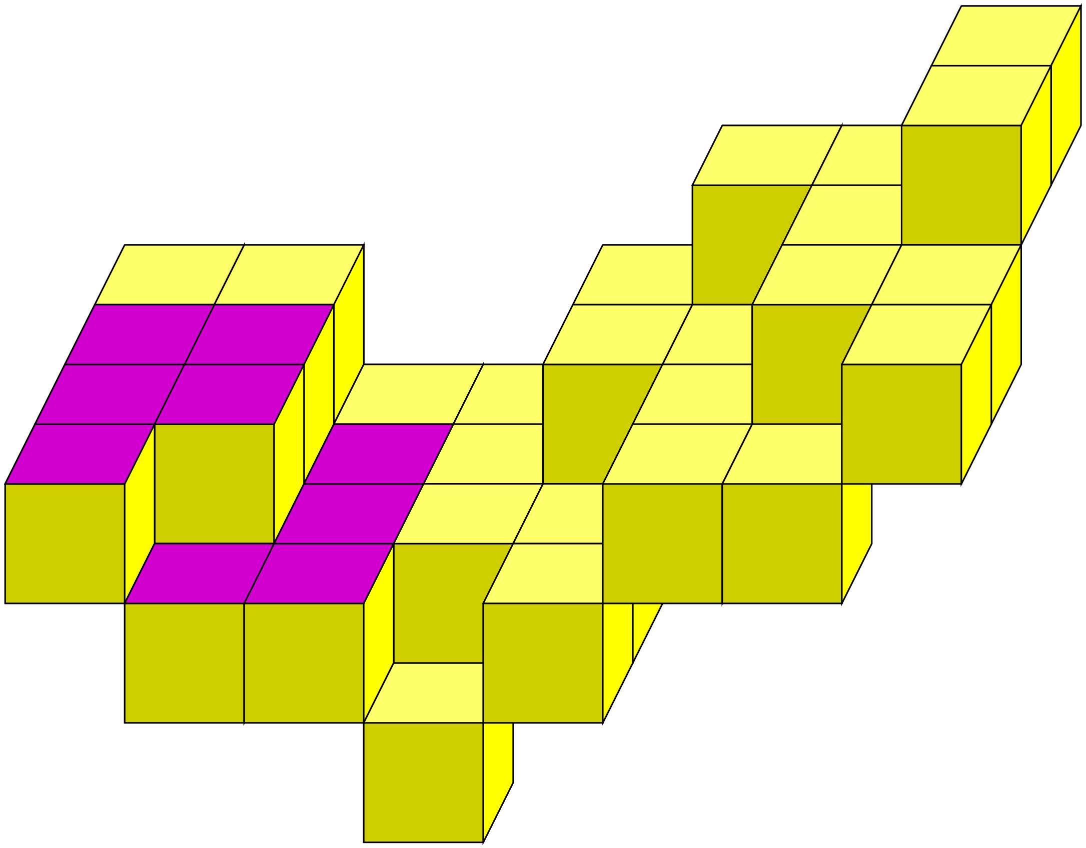
Apply incrementally the directional recognition algorithm in direction $d$ with seed $v$.



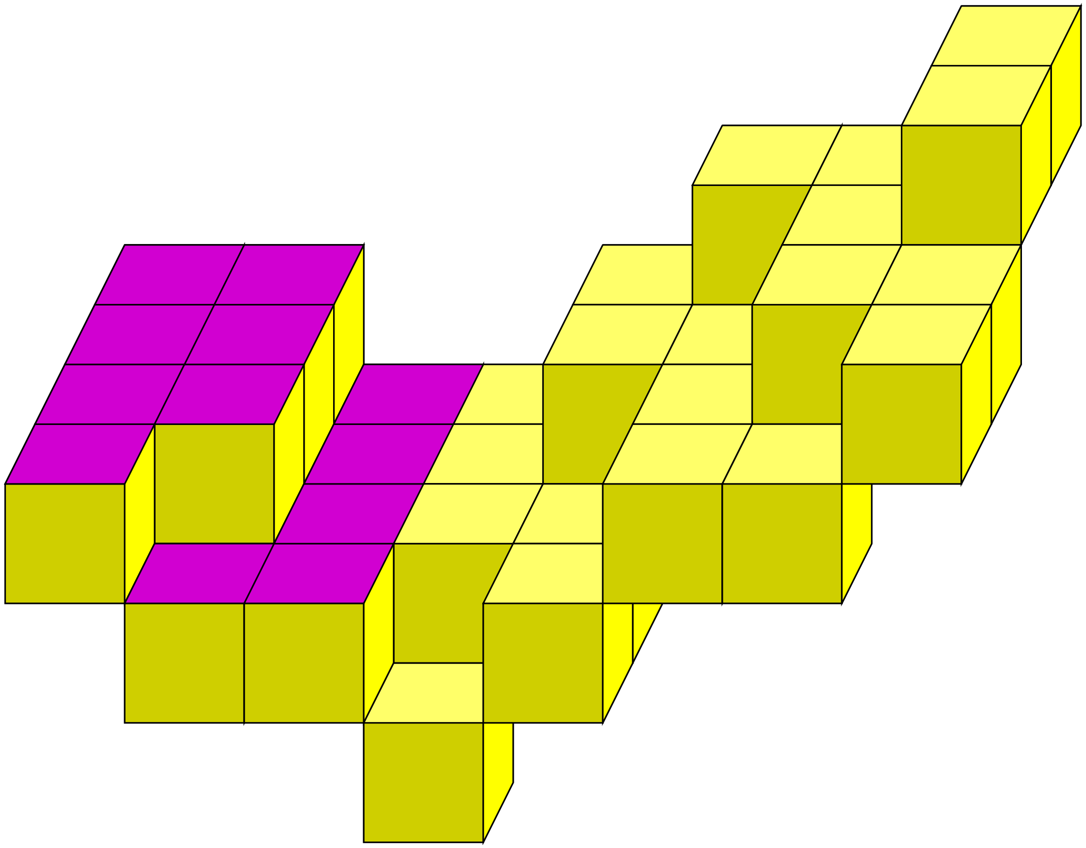
# Digital plane segmentation



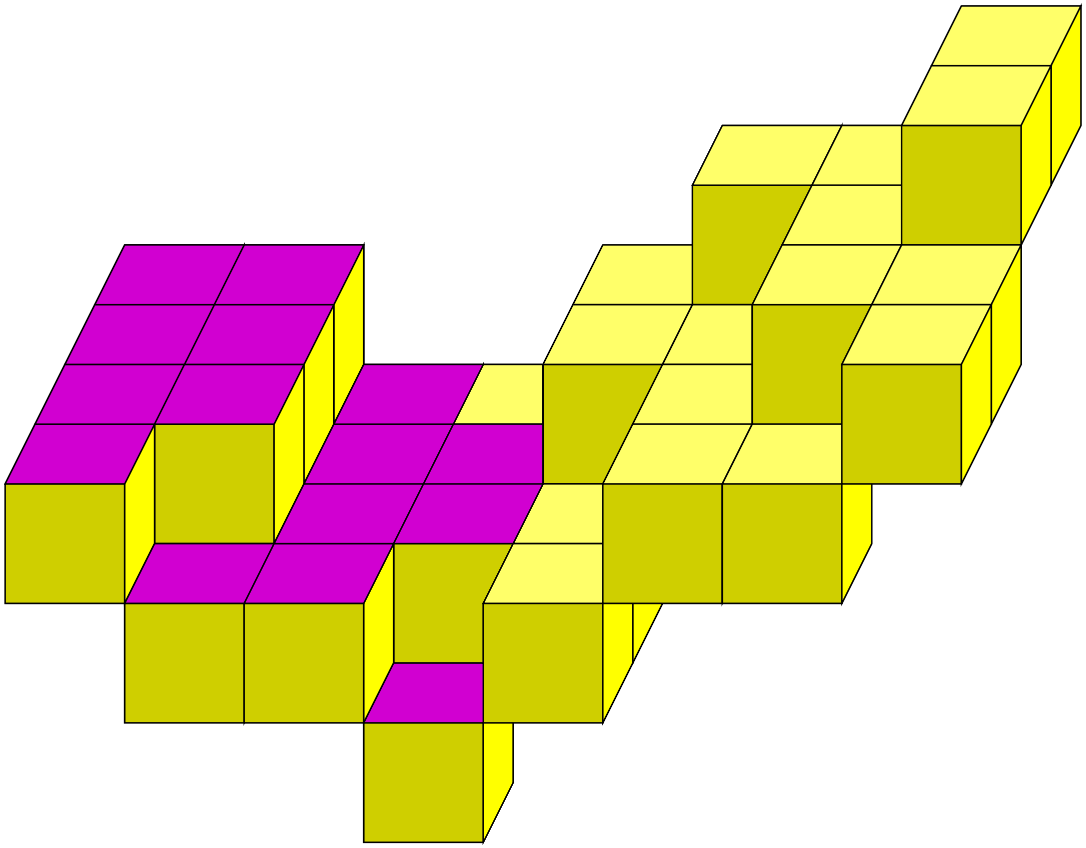
# Digital plane segmentation



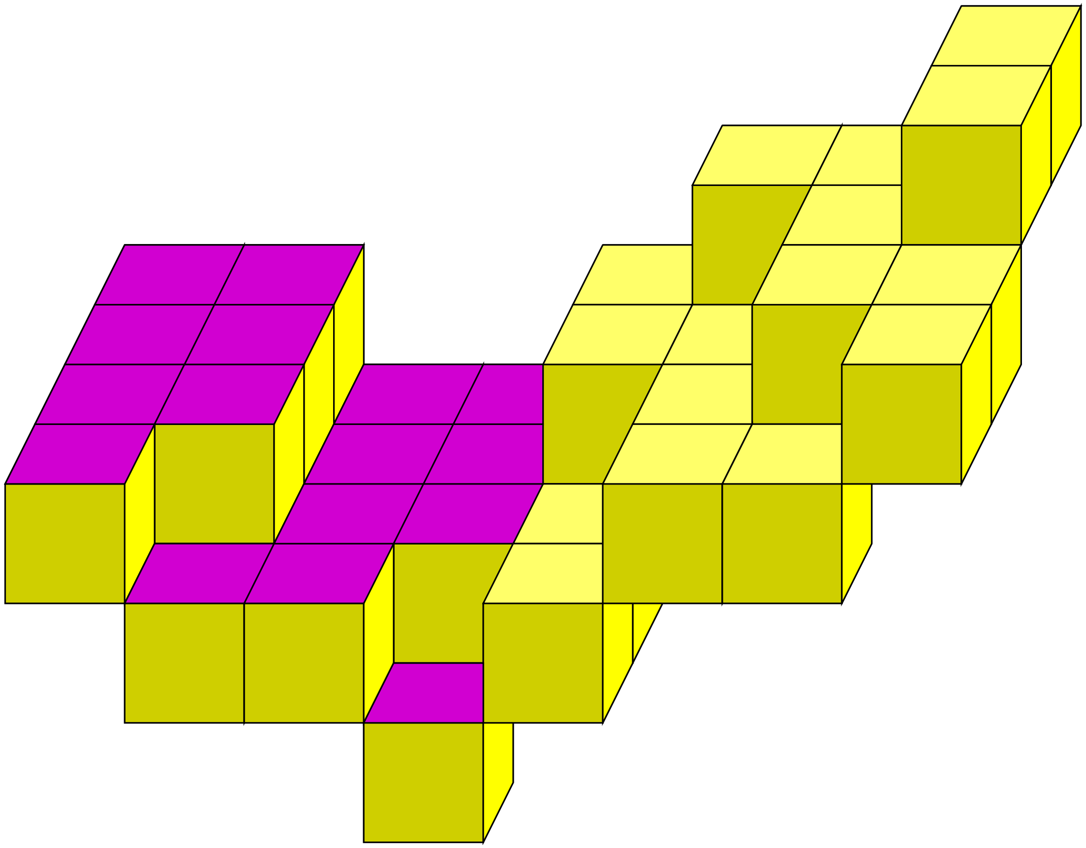
# Digital plane segmentation



# Digital plane segmentation

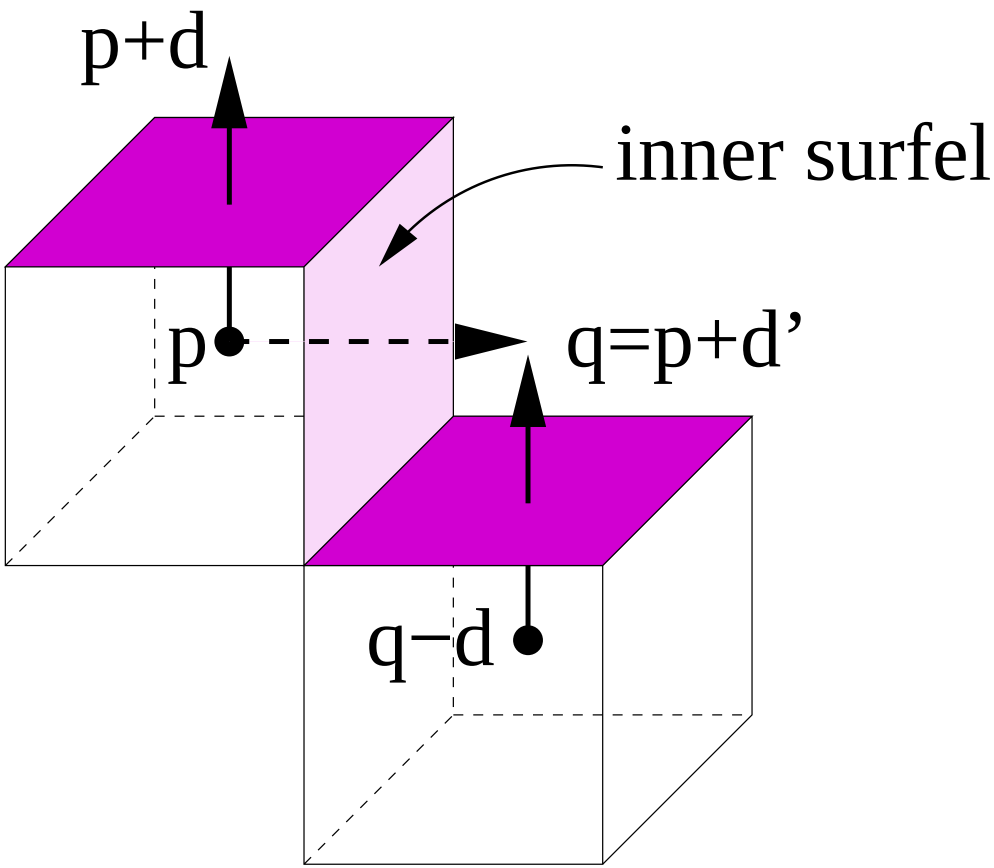


# Digital plane segmentation

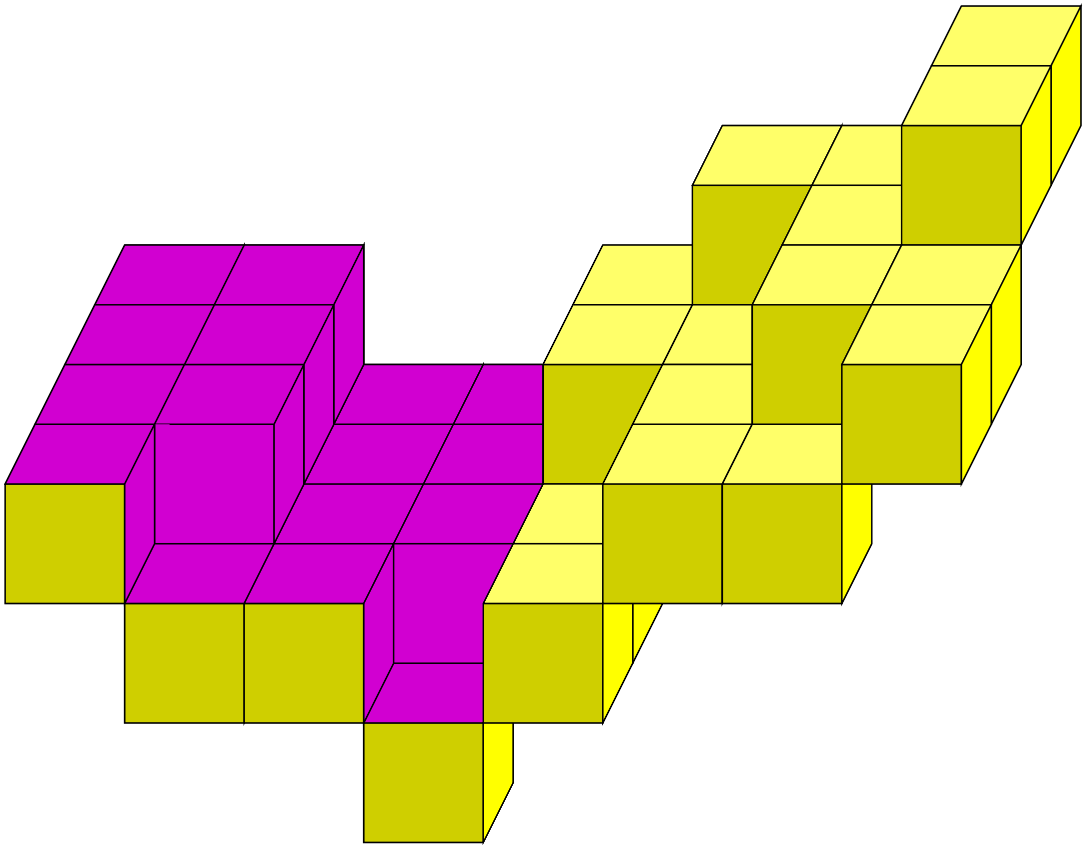


# Digital plane segmentation

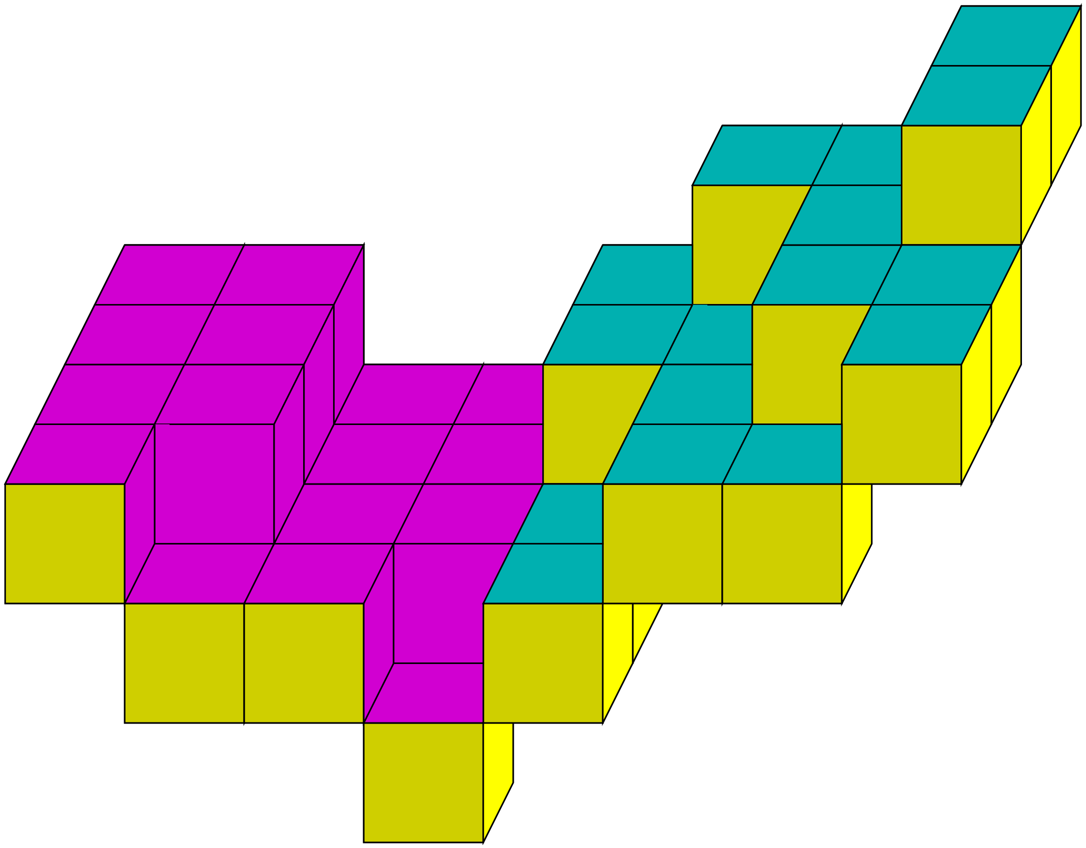
*Inner surfel of a plane $P$ in direction $d$:*



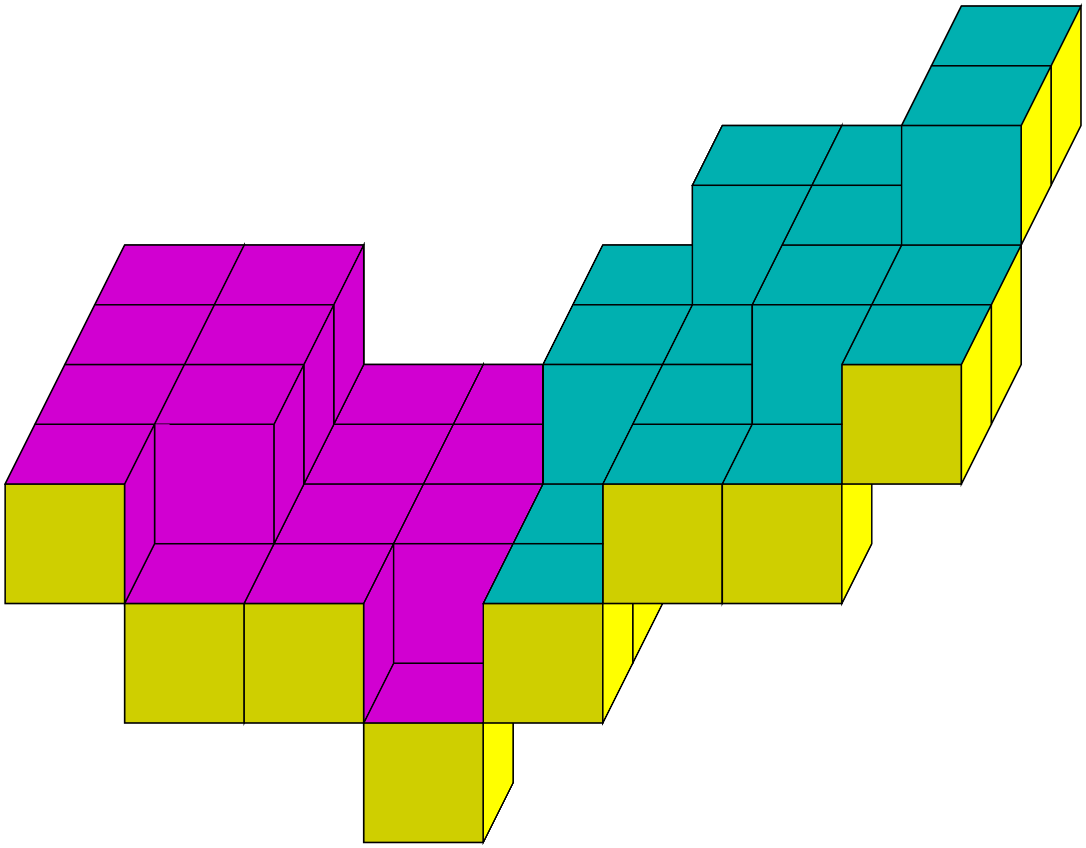
# Digital plane segmentation



# Digital plane segmentation

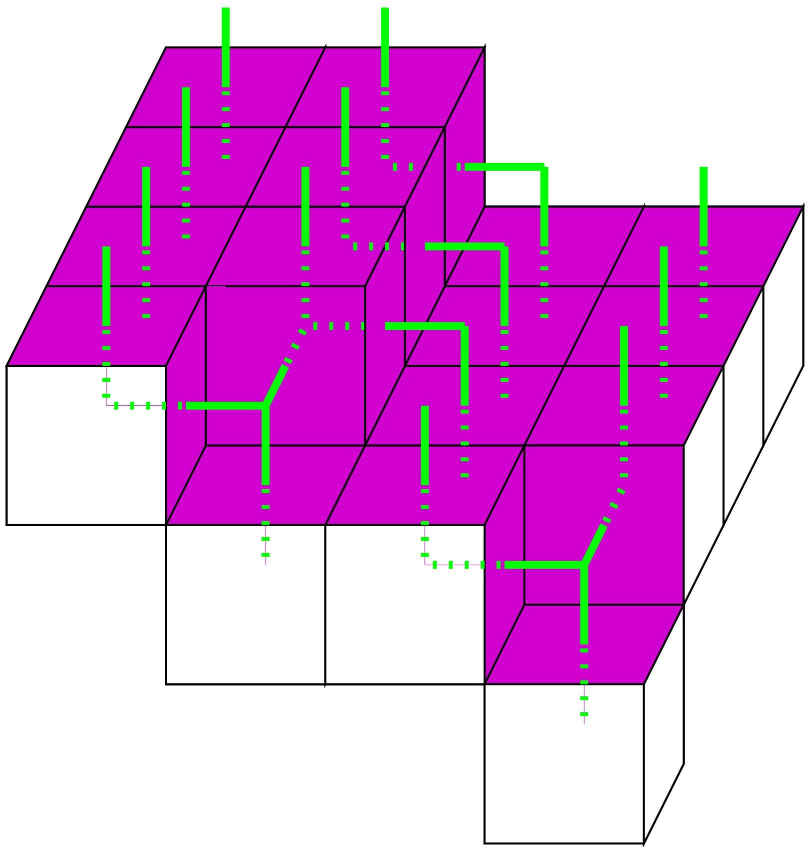


# Digital plane segmentation

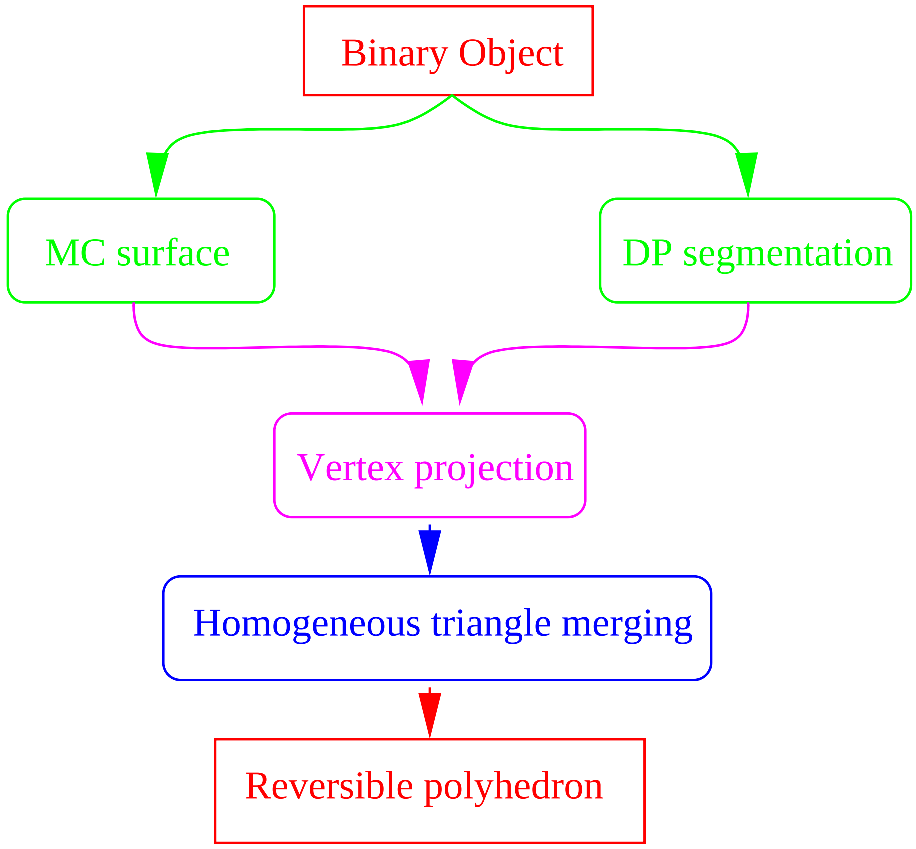


# Digital plane segmentation

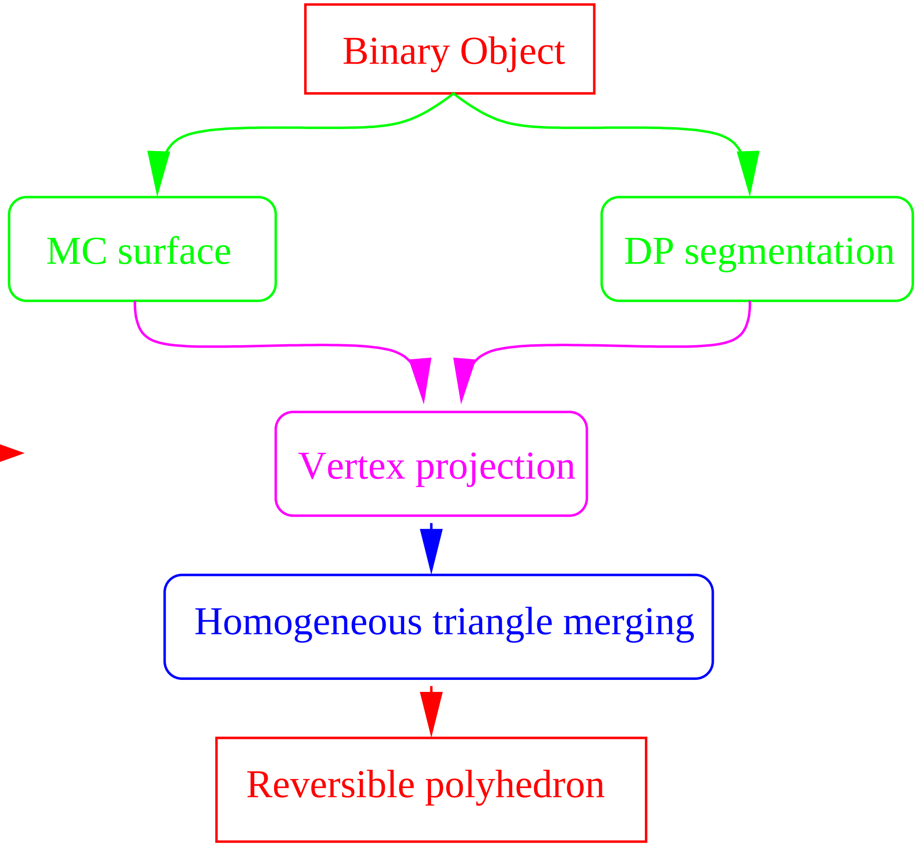
*Property:* All the solution planes of $P$ cross all the segments $[pq[$ where $\{p,q\}$ is a surfel labelled with $P$.



# Sketch of the algorithm

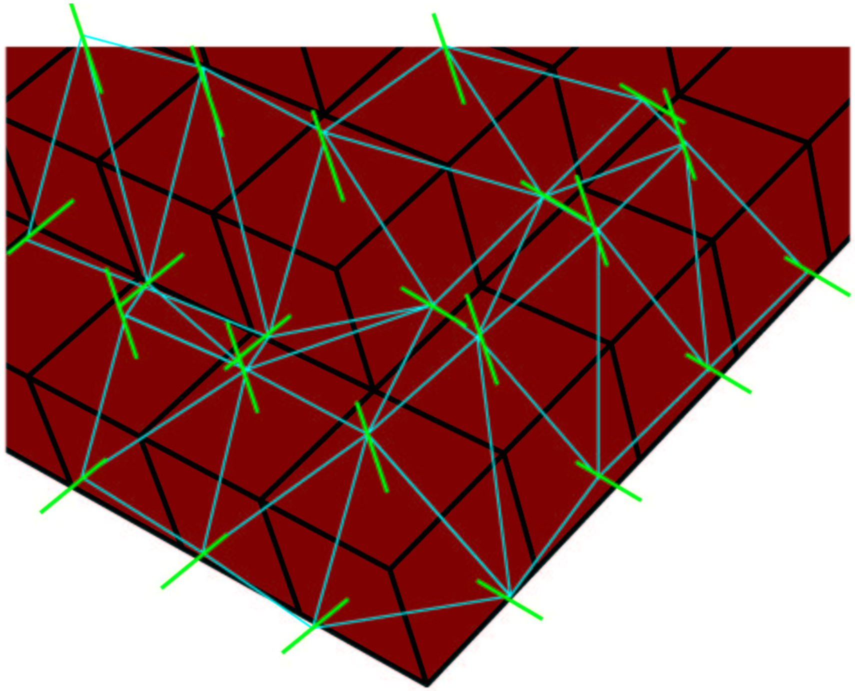


# Vertex projection



# Vertex projection

*Perpendicular projection of a surfel center onto a plane:* given a surfel defined by $(p, q) \in \mathbb{Z}^2$ with $d^1(p, q) = 1$, we center its projected onto the Euclidean plane in the $(pq)$ direction.



# Vertex projection

Given a set of surfels belonging to the same DP

*Step 1:* Extract an Euclidean plane from the DP preimage

*Step 2:* Project all MC vertices onto such a plane

Lemma 4 The polyhedron obtained at the end of the vertex projection step has got the reversibility property.

Proof hints : since the Euclidean plane comes from the DP preimage, all projected vertices belong to the $[p, q[$ segment.

# Vertex projection

Given a set of surfels belonging to the same DP

*Step 1:* Extract an Euclidean plane from the DP preimage

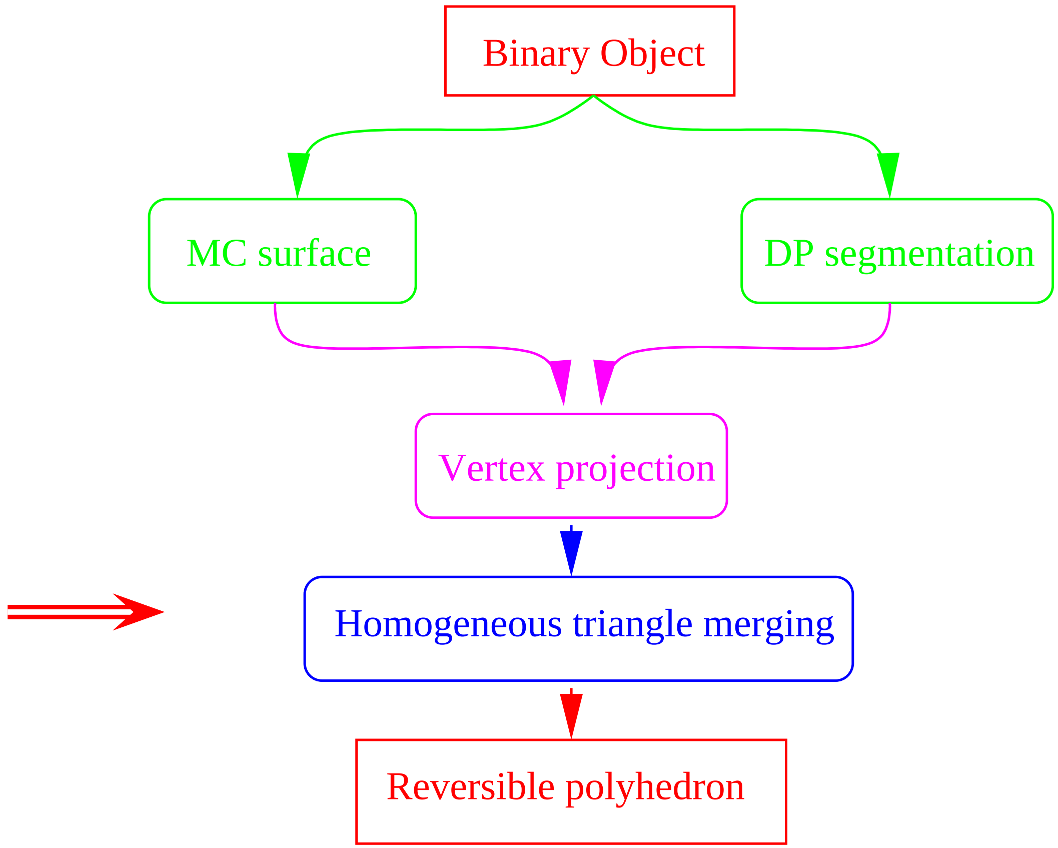
*Step 2:* Project all MC vertices onto such a plane

Lemma 4 The polyhedron obtained at the end of the vertex projection step has got the reversibility property.

Proof hints : since the Euclidean plane comes from the DP preimage, all projected vertices belong to the $[p, q[$ segment.

*Important: all Euclidean planes of the DP preimage can be used...*

# Homogeneous triangle merging

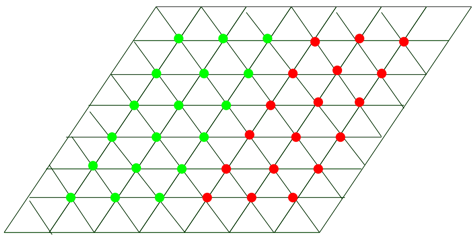


# Homogeneous triangle merging

*Homogeneous triangle :* a triangle of the MC is homogeneous if its vertices belong to the same Digital Plane.

# Homogeneous triangle merging

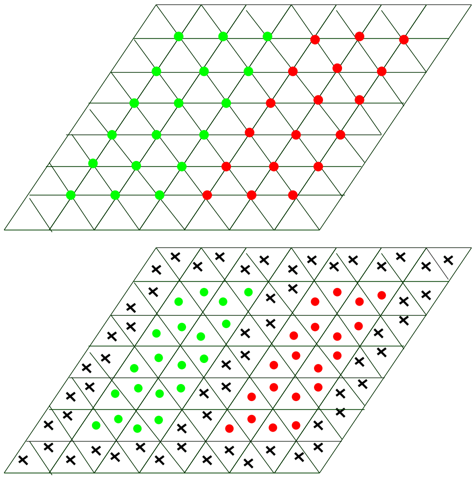
*Homogeneous triangle :* a triangle of the MC is homogeneous if its vertices belong to the same Digital Plane.



- DP segmentation labelling on vertices

# Homogeneous triangle merging

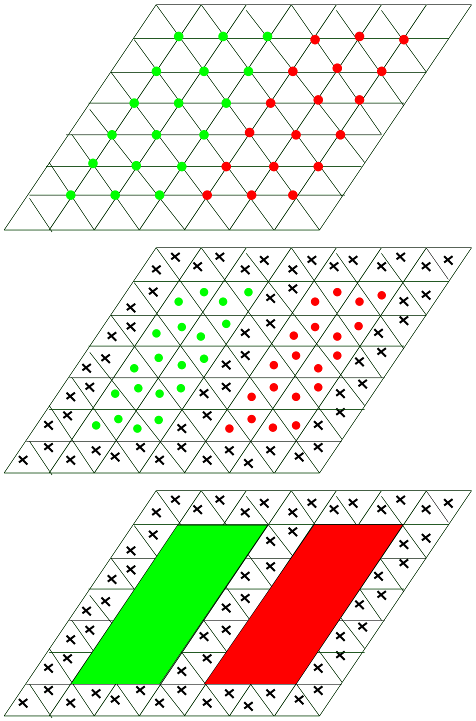
*Homogeneous triangle :* a triangle of the MC is homogeneous if its vertices belong to the same Digital Plane.



- DP segmentation labelling on vertices
- Homogeneous triangle labelling

# Homogeneous triangle merging

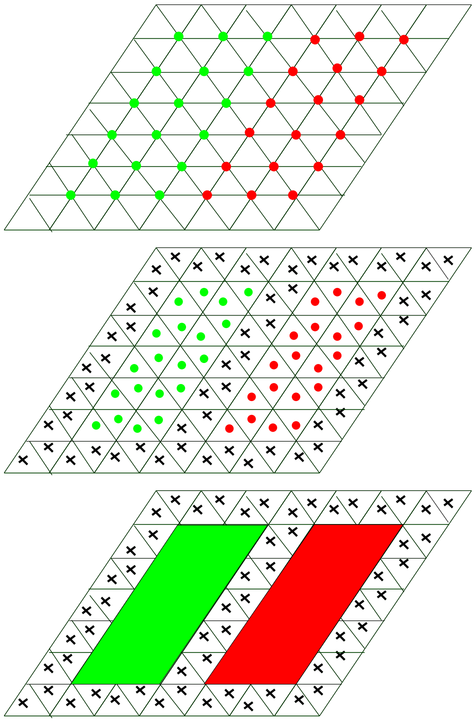
*Homogeneous triangle :* a triangle of the MC is homogeneous if its vertices belong to the same Digital Plane.



- DP segmentation labelling on vertices
- Homogeneous triangle labelling
- Connected homogeneous triangle set are merged

# Homogeneous triangle merging

*Homogeneous triangle :* a triangle of the MC is homogeneous if its vertices belong to the same Digital Plane.



- DP segmentation labelling on vertices
- Homogeneous triangle labelling
- Connected homogeneous triangle set are merged

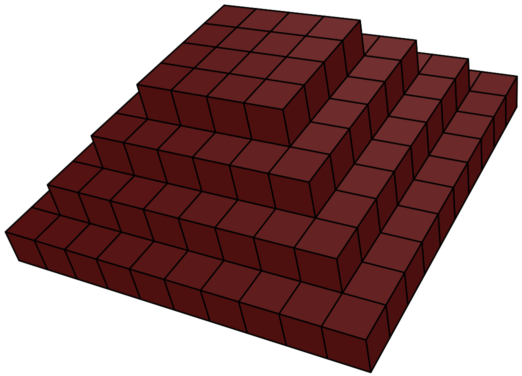
Homogeneous triangles are coplanar at the end of the vertex projection step

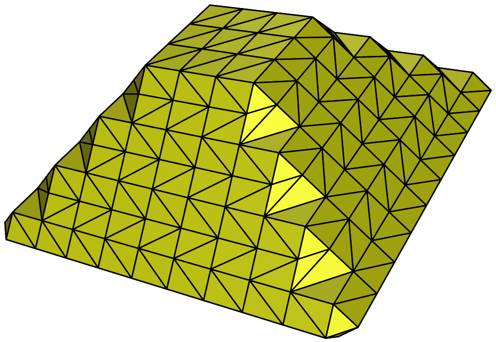
# Complete Algorithm

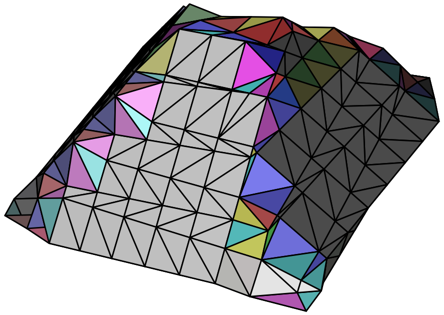
Let $\mathcal{S}$ be the surfels of the boundary of the object $O$
Compute the DPS of $\mathcal{S}$
Let MC the polyhedron given by the Marching-Cubes algorithm
**for each** vertex $v$ of MC **do**
    Find the surfel $s \in \mathcal{S}$ associated to $v$
    Project $v$ onto the representative Euclidean plane of the digital plane associated to $s$
**end for**
Merge adjacent coplanar triangles into polygonal facets.

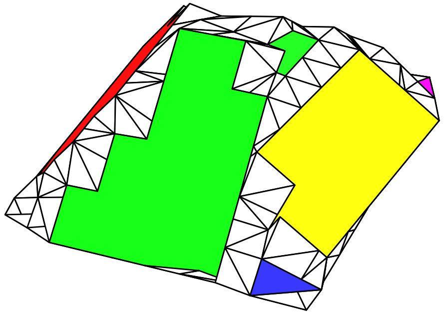
*Theorem* : The polyhedron obtained by the above algorithm has got the reversibility property and is topologically correct (closed, without self-crossing, oriented).

# Results 1 : Overall Algorithm

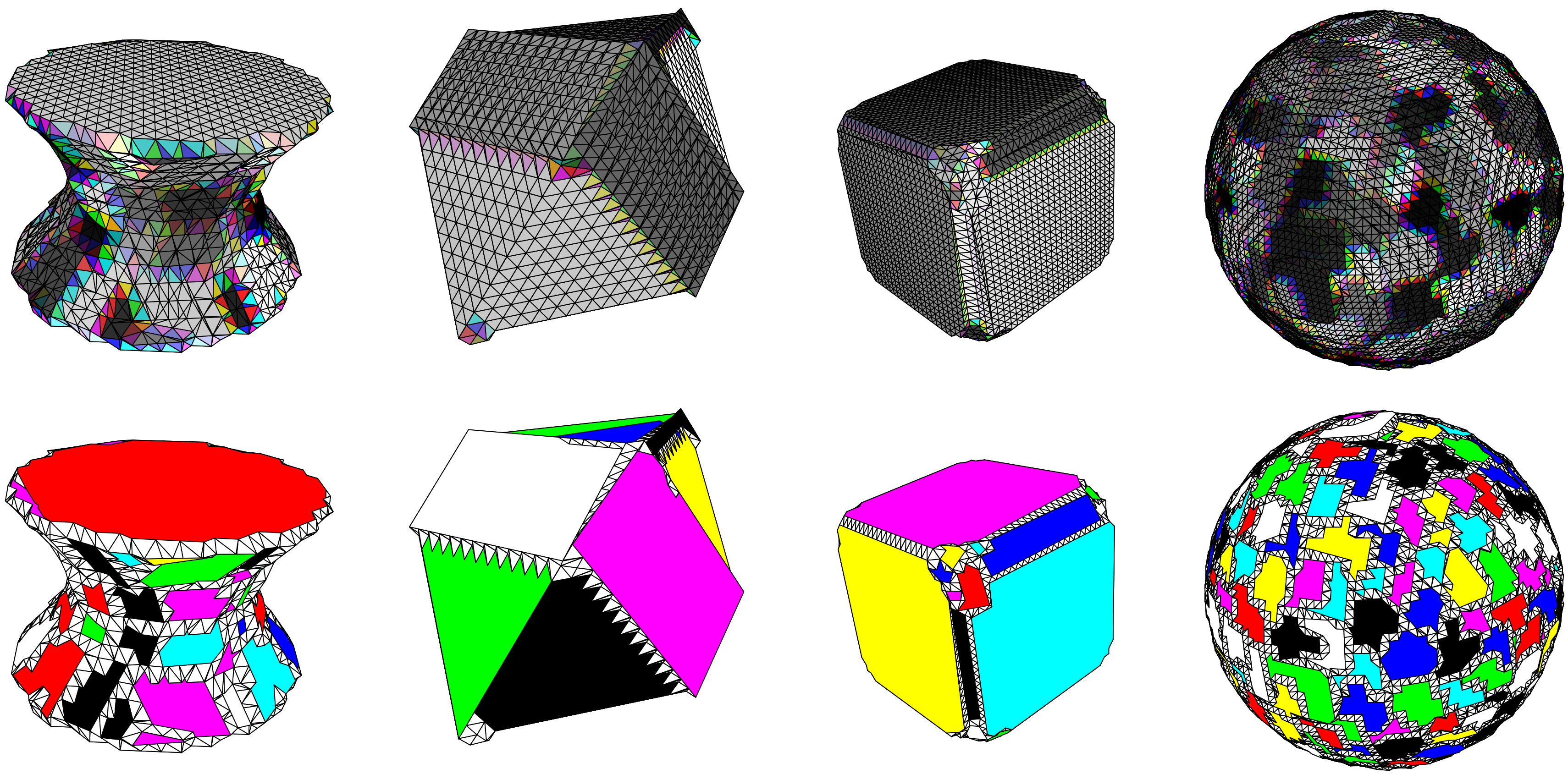








# Results 2



# Efficiency of the polyhedron

| Object | MC | simplified MC | percentage of removed facets |
|---|---|---|---|
| `pyramid` | 620 | 196 | 68% |
| `catenoid` | 5032 | 1427 | 72% |
| `pyramid6` | 4396 | 557 | 87% |
| `rounded_cube` | 9944 | 1621 | 84% |
| `sphere25` | 24632 | 8774 | 64% |

# Conclusion and Future works

*Main result:* algorithm to compute a topologically correct reversible polyhedrization of a binary volume based on a MC simplification

*Future Works:*

- Non-homogeneous triangle patch removal using appropriate choices of the representative Euclidean planes from DP preimages
- Generalization of the algorithm to $n-$dimensional polyhedrization based on $n-$dimensional MC surfaces.