Discrete world problems : comment ouvrir les trous d'un objet

Aldo Gonzalez-Lorenzo, Alexandra Bac, Jean-Luc Mari
Aix-Marseille Université, CNRS, LSIS UMR 7296

Résumé

Un objet discret est un ensemble de pixels, voxels ou leur analogue en dimension supérieure. Un objet discret 3D peut contenir des trous : composantes connexes, tunnels, anses ou cavités. Ouvrir les trous d'un objet discret consiste à éliminer tous ses trous en enlevant certains de ses points. Une façon de le faire est de prendre un point dans l'objet et de le dilater en restant contractile : les points restants sont ceux qui doivent être enlevés.

Nous avons développé deux algorithmes pour ouvrir les trous d'un objet discret en dimension quelconque. Les deux algorithmes s'appuient sur la transformée de distances de l'objet, mais ils diffèrent sur la manière dont la dilatation est faite.

Définitions

Un **objet discret** d-dimensionnel est un ensemble fini de de points avec de coordonnées entières. Il est muni d'une **rélation d'adjacence** qui définie sa topologie.

Un point est **simple** pour un objet discret X si son addition ou suppresion ne change pas le type d'homotopie de l'objet. Cette notion peut être etendue à un ensemble de points.

Un objet discret est **contractile** si on peut le réduire à un point avec une séquence de suppresion de points simples. Il n'a donc pas de trous.

Une **ouverture homotopique** pour un objet discret X est un sous-ensemble Y tel que X-Y est contractile. Donc, l'ouverture homotopique est l'ensemble de points qu'on doit enlever de l'objet pour qu'il n'ait plus de trous.

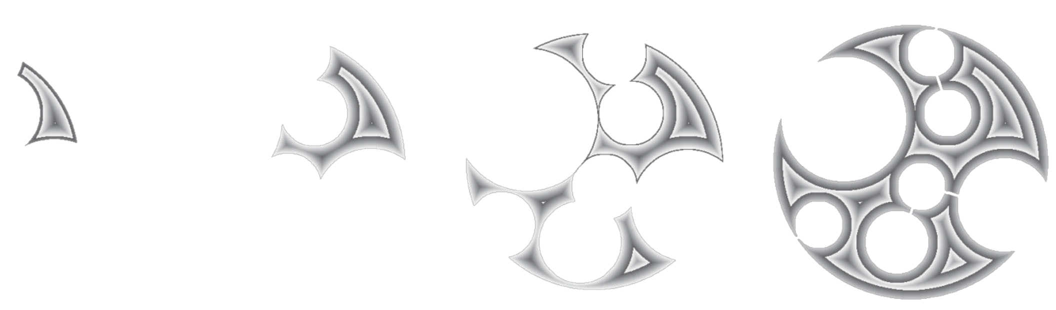
Algorithme 1 : propagation aléatoire

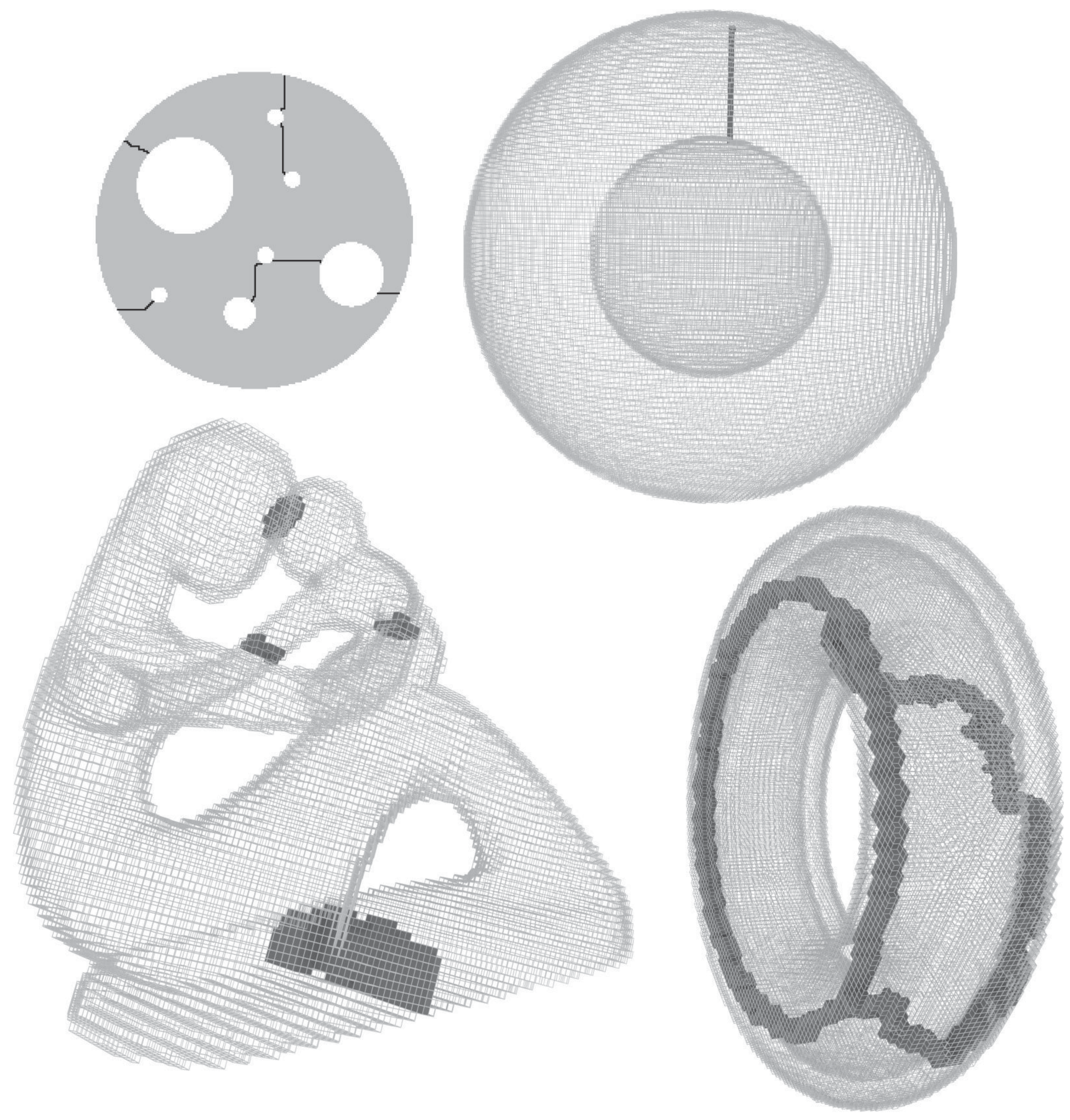
Idée de base : prendre comme sous-ensemble contractile initial un point dans l'objet et ajouter des points simples de l'objet tant que c'est possible.

Priorité aux points plus profonds pour une propagation plus uniforme.

Avant d'ajouter un point simple, on teste si on pourrait ajouter une boule discrète de rayon *r* centrée sur le point. Cela produit des coupures épaisses.

Plusieurs points avec la même transformée de distance ? Choisir un à l'hasard.





Algorithm 1: Homotopic opening with random propagation

Input: $X \subset \mathbb{Z}^d$, $r \in \mathbb{Z}$
Output: Homotopic opening for X
$C \leftarrow \{x\}$ for a random $x \in X$ such that $dt_X(x)$ is maximal;
$S \leftarrow N^*(x) \cap (X - C)$;
while $S \neq \emptyset$ **do**
  $x \leftarrow$ random point in S such that $dt_X(x)$ is maximal;
  $S \leftarrow S - \{x\}$;
  if $(B(x,r) - C)$ *is simple for* C *and* x *is simple for* C **then**
    $C \leftarrow C \cup \{x\}$;
    $S \leftarrow S \cup (N^*(x) \cap (X - C))$;
  return $X - C$;

[>] Complexité : O(n (log n + r^{2d})) pour des objets 3D

[>] L'ouverture homotopique est anysotropique

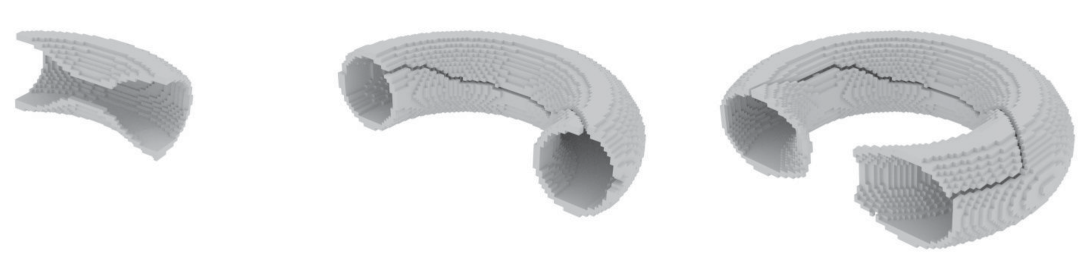
[>] Un grand rayon de boule donne des coupures plus «droites»

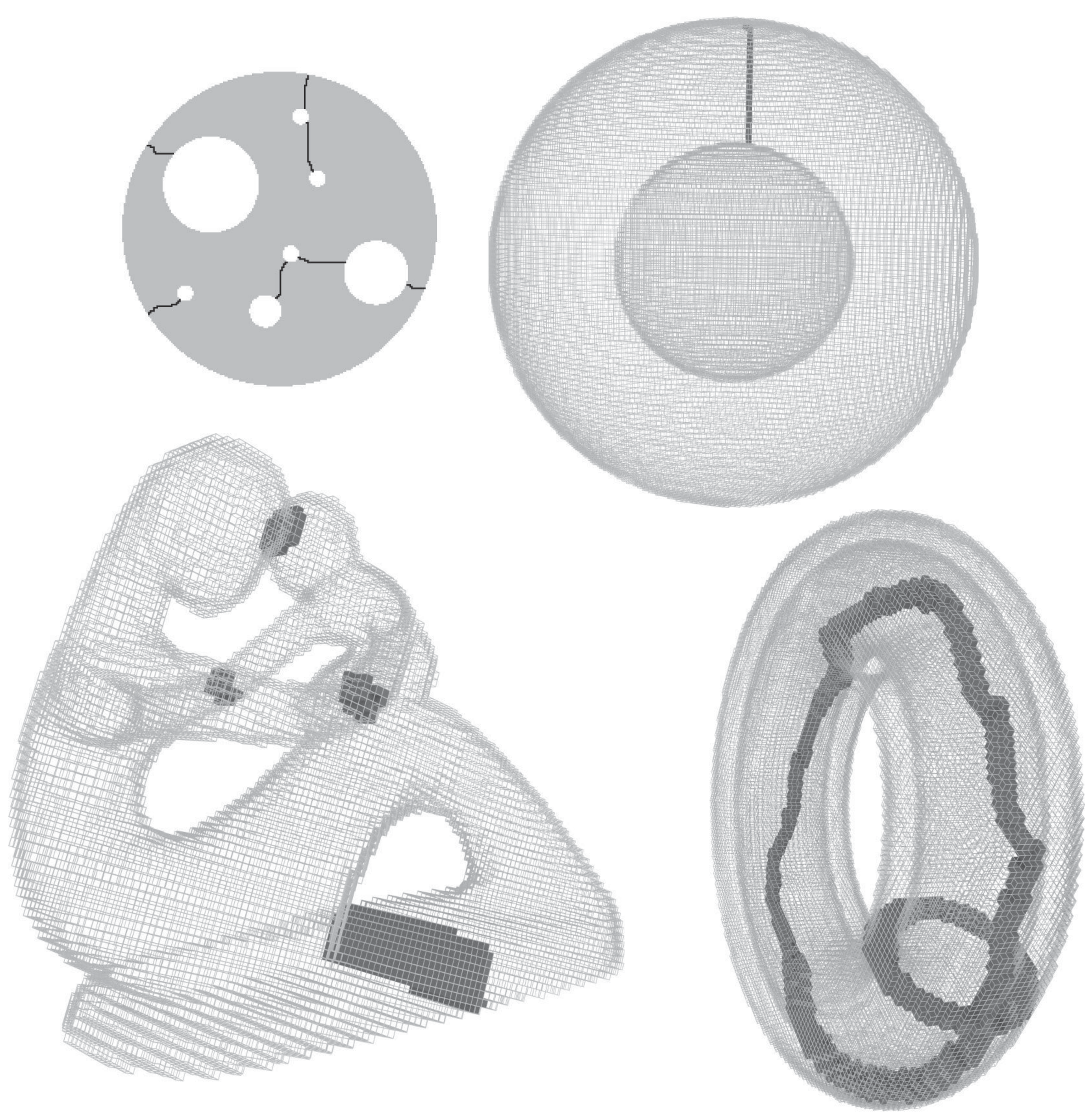
[<] Coupures étranges sur des objets avec des parties plates

Algorithme 2 : propagation par couches

Beaucoup de points équidistants au complémentaire ? La propagation aléatoire peut donner des résultats étranges.

Solution : propager le sous-ensemble contractile par couches. On restreint la propagation à un voisinage du sous-ensemble. Dès qu'elle est finie, on recalcule le voisinage.





Algorithm 2: Homotopic opening with propagation by layers

Input: $X \subset \mathbb{Z}^d$, $r \in \mathbb{Z}$
Output: Homotopic opening for X
$C \leftarrow \{x\}$ for a random $x \in X$ with highest dt_X value;
repeat
  $m \leftarrow \max\{dt_X(x) \mid x \in N^*(C) \cap X, x \text{ simple for } C\}$;
  $L \leftarrow dt_X^{-1}([m-1, m]) \cap N^*(C)$;
  foreach $x \in L$ **do**
    if $(B(x,r) - C)$ *is simple for* C *and* x *is simple for* C **then**
      $C \leftarrow C \cup \{x\}$;
until *idempotency*;
return $X - C$;

[<] Complexité : O(n^2 r^{2d}) pour des objets 3D (pire)

[>] Meilleurs résultats sur des objets avec des parties plates

Morale

Bien que l'ouverture homotopique d'un objet discret soit bien définie, nous n'arrivons pas a mesurer sa qualité.

Par example, les deux ouvertures suivantes ont le même nombre de pixels.

C'est donc un discrete world problem !