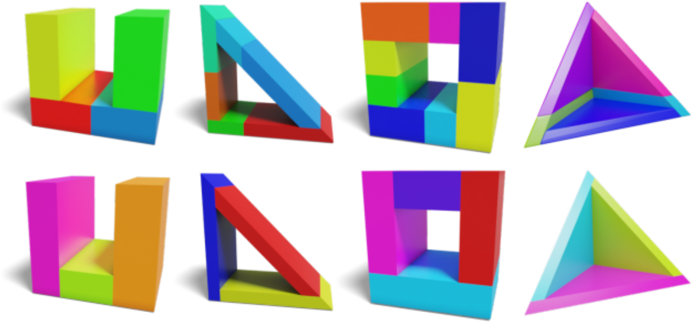
Approximate Convex Decomposition for 3D Meshes with Collision-Aware Concavity and Tree Search

XINYUE WEI* and MINGHUA LIU*, University of California San Diego, USA
ZHAN LING, University of California San Diego, USA
HAO SU, University of California San Diego, USA



Pourquoi?

On cherche à décomposer un maillage comme une union minimale de parties convexes.
Pourquoi?

Pourquoi?

On cherche à décomposer un maillage comme une union minimale de parties convexes.

Pourquoi?

- énormément d'algorithmes optimaux demandent des maillages convexes:
  - Detection de collisions (simulation physique)
  - Appartenance d'un point à une forme (rendu, simulation physique)
  - Optimisation mathématique

Pourquoi?

On cherche à décomposer un maillage comme une union minimale de parties convexes.

Pourquoi?

- énormément d'algorithmes optimaux demandent des maillages convexes:
  - Detection de collisions (simulation physique)
  - Appartenance d'un point à une forme (rendu, simulation physique)
  - Optimisation mathématique
- Beaucoup d'autres algorithmes en ont besoin dans leur pipeline
  - Extraction de squelettes
  - Génération de maillages tétrahédriques
  - Déformation de maillages

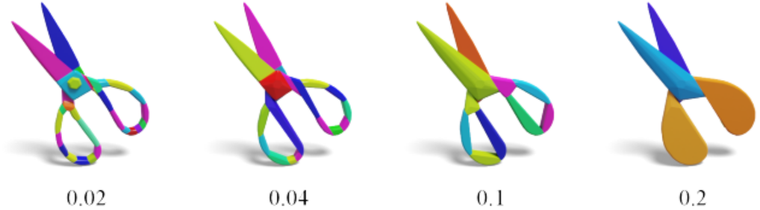
Solution exacte?

Solution exacte : NP-HARD [Chazelle et al. 1997]

Solution exacte?

Solution exacte : NP-HARD [Chazelle et al. 1997]

$\Longrightarrow$ Solution approchée: on définit un ϵ de tolérence à la concavité puis on prend les enveloppes convexes.



Comment?

Algorithm 1: Approximate Convex Decomposition

Input: A 2-manifold solid mesh $\mathcal{S}$, a concavity threshold ϵ
Output: Approximate convex decomposition $\mathcal{D}$

$Q \leftarrow \{\mathcal{S}\}$ // processing queue
$\mathcal{D} \leftarrow \emptyset$ // decomposition results
while Q *is not empty* **do**
  $C \leftarrow Q.dequeue()$
  if $\mathrm{Concavity}(C) < \epsilon$ **then** // Section 4
    $\mathcal{D}.enqueue(C)$
  else
    $\mathcal{P} \leftarrow \mathrm{MCTS}(C)$ // search for cutting plane, Section 6
    $\mathcal{P} \leftarrow \mathrm{Refine}(C, \mathcal{P})$ // Section 6.5
    $C_L, C_R \leftarrow \mathrm{Cut}(C, \mathcal{P})$ // Section 5
    $Q.enqueue(C_L, C_R)$
$\mathcal{D} \leftarrow \mathrm{Merge}(\mathcal{D}, \epsilon)$ // Section 6.5

Apports du papiers:

- Nouvelle métrique pour la concavité
- MCTS?

Métrique pour la concavité

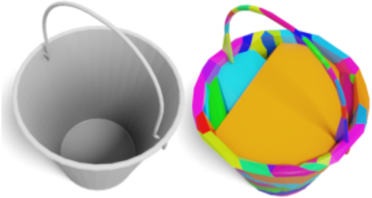
Caractérisation utile de la convexité:

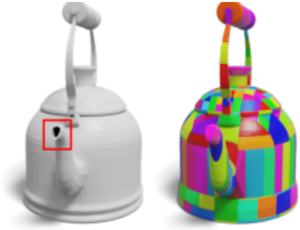
$$\Sigma \text{ convexe} \iff \Sigma = ConvexHull(\Sigma)$$

Donc concavité:

$$\text{Concavité}(\Sigma) = \mu(ConvexHull(\Sigma) \setminus \Sigma)$$

Deux métriques (μ) connues:
Distance entre les frontières / Différence de volume





Nouvelle métrique

Les deux métriques précédentes capturent des informations importantes mais complémentaires: $\implies$ besoin de les combiner.

Distance de Haussdorff:

$$H(A,B) = \max(\sup_{x \in A} d(x,B), \sup_{y \in B} d(y,A))$$

Nouvelle métrique

Les deux métriques précédentes capturent des informations importantes mais complémentaires: $\implies$ besoin de les combiner.

Distance de Haussdorff:

$$H(A,B) = \max(\sup_{x \in A} d(x,B), \sup_{y \in B} d(y,A))$$

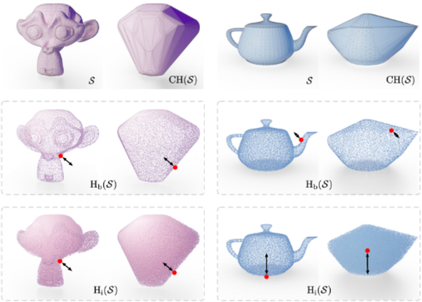
- Relativement peu coûteux sur des nuages de points donc:

$$H(A,B) \approx H(\text{Sample}(A), \text{Sample}(B))$$

Echantillionner pour approximer les distances

On introduit:

$$H_i(\Sigma) = H(\text{Sample}(\text{Int}(\Sigma)), \text{Sample}(\text{Int}(\mathit{ConvexHull}(\Sigma))))$$
$$H_b(\Sigma) = H(\text{Sample}(\partial\Sigma), \text{Sample}(\partial \mathit{ConvexHull}(\Sigma)))$$



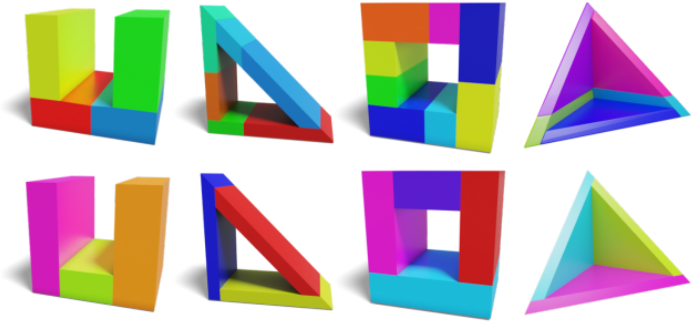
Mesure de la concavité

$$\text{Concavité}(\Sigma) = \max(H_i(\Sigma), H_b(\Sigma))$$

Trouver le plan de coupe

Auparavant on cherchait le plan de coupe P qui minimise:

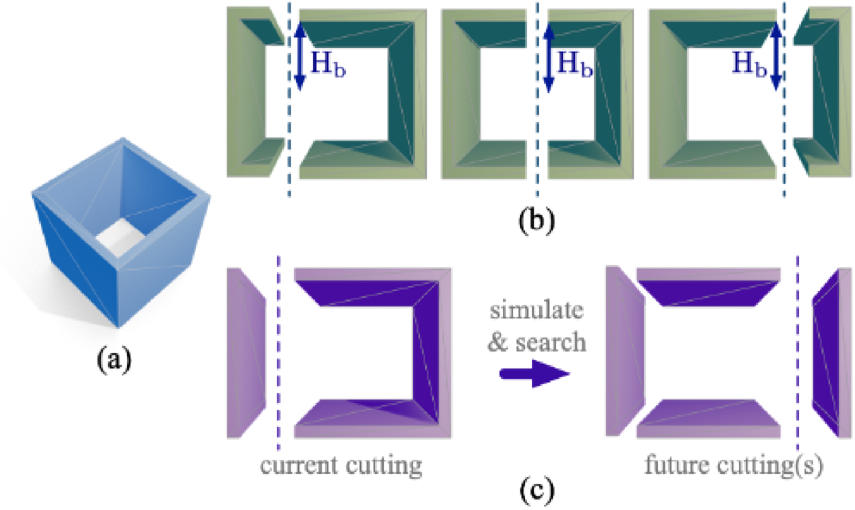
$$\min_P \ \max(\text{Concavité}(\text{Cut}_L(\Sigma, P)), \text{Concavité}(\text{Cut}_R(\Sigma, P)))$$



Mais approche trop locale $\implies$ chercher à optimiser plus globalement par un MTCS

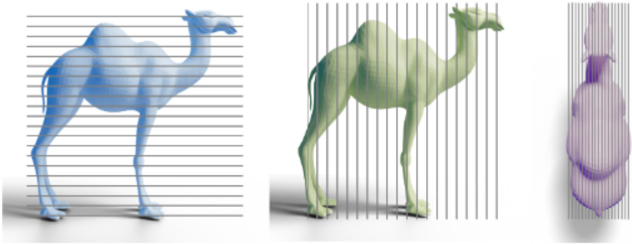
MCTS

On se sert d'un MCTS pour rendre la recherche plus globale:



Le MCTS fonctionne par arbre, par exemple pour des jeux où l'espace des configuration est dénombrable : pas le cas ici car l'espace des plans de $\mathbb{R}^3$ est isomorphe à $\mathbb{S}^2 \times \mathbb{R}^+$.

Donc:

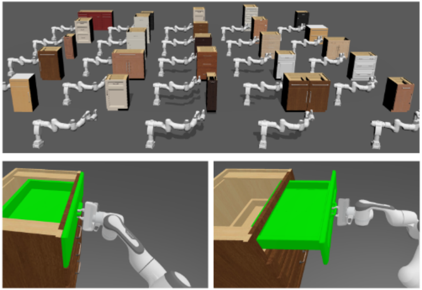


(Avant: Alignement par PCA)

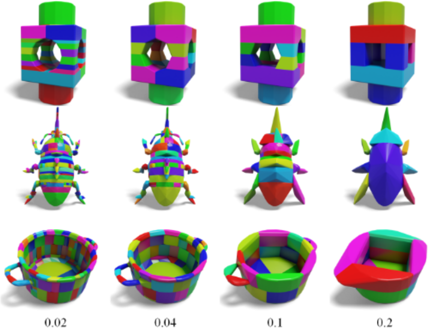
Résultats

Produit de meilleurs résultats:

- Grâce à la métrique: plus robuste
- Grâce au MCTS: moins de morceaux (et donc termine plus vite)



50% de succès → 80%



Conclusion

Approche pas fondamentalement nouvelle mais amélioration du pipeline.
Pourquoi j'aime: Contribution ingénieuse à un problème utile.
Critique: beaucoup de paramètres à régler...

Vrai atout du papier: vidéo de présentation magistrale