

Efficient Voxelization

Using Projected Optimal Scanline

François Palma
Nicolas Marguerit
Alexandre Réaubourg

January 4, 2022

Sommaire

1. État de l'art
2. Etude de l'article
3. Prototypage
4. Résultats & Benchmark

État de l'art

État de l'art

Voxelization

Exemple d'utilisation:

- ▶ Modélisation
- ▶ Simulation physique
 - ▶ De nouvelles simulations rendues possibles
- ▶ Eclairage volumétrique

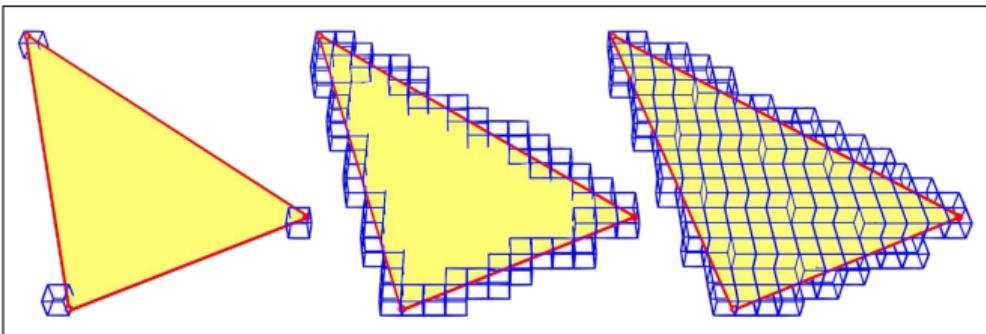
Etude de l'article

Etude de l'article

3D voxelization ? scanlines ?

Etude de l'article

Spécificité de l'algorithme



Etapes de la voxelization

```

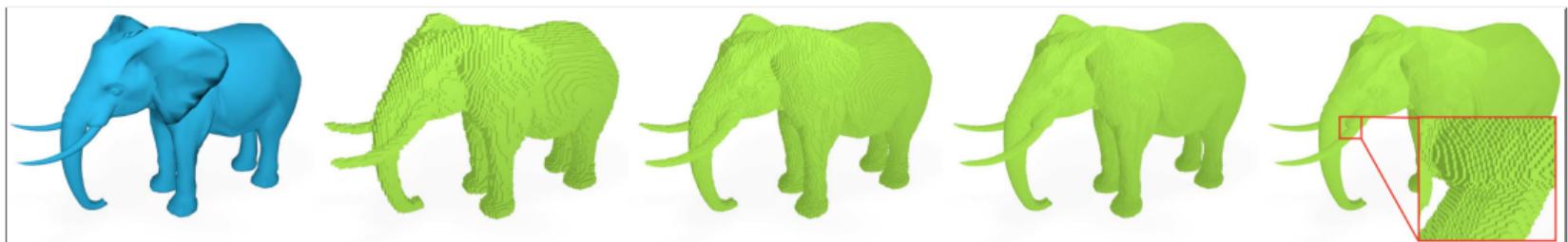
Algorithm 1 Triangle Voxelization
1: function VOXELOUTRIANGLE( $v_0, v_1, v_2, n$ )
2:    $\{P_0, P_1, P_2\} \leftarrow \text{GetVoxel}(v_0, v_1, v_2)$ 
3:    $i \leftarrow \text{DOMINANTAXISINDEX}(n)$ 
4:    $\text{SORTONAXIS}(P_0, P_1, P_2, i)$ 
5:    $\text{MARKLINEILV}(P_0, P_1, Q_0)$ 
6:    $\text{MARKLINEILV}(P_1, P_2, Q_1)$ 
7:    $\text{MARKLINEILV}(P_0, P_2, Q_2)$ 
8:    $Q_1 \leftarrow Q_0 \cup Q_1$ 
9:    $\text{FILLINTERIOR}(Q_1, Q_2, P_0, P_2, i)$ 
10:
11: function MARKLINEILV( $P_0, P_1, Q$ )
12:    $\Delta P[0] \leftarrow \text{SIGN}(P_1[0] - P_0[0])$ 
13:    $\Delta P[1] \leftarrow \text{SIGN}(P_1[1] - P_0[1])$ 
14:    $\Delta P[2] \leftarrow \text{SIGN}(P_1[2] - P_0[2])$ 
15:    $L[0] \leftarrow M[0] \leftarrow [P_1[1] - P_0[1]]||P_1[2] - P_0[2]]$ 
16:    $L[1] \leftarrow M[1] \leftarrow [P_1[0] - P_0[0]]||P_1[2] - P_0[2]]$ 
17:    $L[2] \leftarrow M[2] \leftarrow [P_1[0] - P_0[0]]||P_1[1] - P_0[1]]$ 
18:    $P_{current} \leftarrow P_0$ 
19:   while  $P_{current} \neq P_1$  do
20:      $(L_{init}, L_{index}) \leftarrow \text{Min}(L[0], L[1], L[2])$ 
21:      $P_{current}[L_{index}] \leftarrow P_{current}[L_{index}] + \Delta P[L_{index}]$ 
22:      $L \leftarrow L - L_{init}$ 
23:      $L[L_{index}] \leftarrow L_{init}$ 
24:      $\text{MARKVOXEL}(P_{current})$ 
25:      $Q.PUSHBACK(P_{current})$ 
26:
27: function FILLINTERIOR( $Q_1, Q_2, P_0, P_2, axis$ )
28:   for  $i = 0$  to  $P_2[axis] - P_0[axis]$  do
29:      $slice \leftarrow P_0[axis] + i + 1/2$ 
30:      $Q_{1,sub} \leftarrow \text{GetSubSEQUENCE}(Q_1, slice)$ 
31:      $Q_{2,sub} \leftarrow \text{GetSubSEQUENCE}(Q_2, slice)$ 
32:     while  $Q_{1,sub} \neq \emptyset \text{ OR } Q_{2,sub} \neq \emptyset$  do
33:        $P_{start} \leftarrow \text{GetNEXTINSLICE}(Q_{1,sub})$ 
34:        $P_{stop} \leftarrow \text{GetNEXTINSLICE}(Q_{2,sub})$ 
35:        $\text{MARKLINEILV}(P_{start}, P_{stop})$ 

```

Etude de l'article

Points clés

- ▶ Plus c'est rapide mieux c'est
- ▶ Parfaitement compatible avec le multithreading
- ▶ Plus on a de triangles plus c'est rapide
- ▶ L'approximation des entiers réduit les trous dans la couverture



Prototypage

Prototypage

Notre plan initial



Prototypage

Premiers jours

Prototypage

Difficultés rencontrées

Prototypage

Éclaircissement des zones d'ombres sur l'article

Résultats & Benchmark

Résultats & Benchmark

Quelques chiffres en comparaison

Résultats & Benchmark

Démo

Résultats & Benchmark

Perspectives d'évolution

Résultats & Benchmark

Conclusion