

Q-Note

Découpage Intelligent
dans Blender



Présentation & Stage



Amélioration des rendus

Le problème

Analyse

Approche préconisée

Réduction du temps d'exécution

Algorithme général



Perspectives

Moi

- 4 ans en Maths en Chine
- 1ère année à l'ENSAE ParisTech en Maths
- Stage depuis 1 juin

Mon Stage

- Statistique descriptive
- Amélioration des rendus

1

Le problème



$t_0 = 1516s$ $t_1 = 1947s$
 $t_2 = 408s$ $t_3 = 713s$

$T_0 = 1947s$

Nombre de
processeurs
: 4



$t_0 = 912s$ $t_1 = 1012s$
 $t_2 = 1721s$ $t_3 = 939s$

$T_1 = 1721s$

Temps de
rendu : T_i



$t_0 = 1698s$ $t_1 = 1765s$
 $t_2 = 788s$ $t_3 = 333s$

$T_2 = 1765s$

Question : Quel découpage à choisir (Rectangle Tiling & Multi-armed Bandit problems) ?

2 Analyse

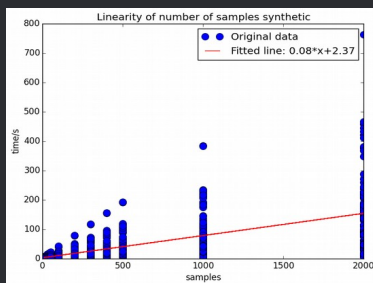
- Remarque 1 : Le temps de rendu dépend du nombre de samples et de la taille d'image (resolutions) choisis.



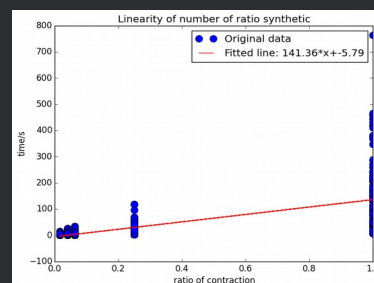
- Remarque 2 : Le temps de rendu dépend linéairement (le terme de constant proche de 0) du nombre de samples et de la taille d'image (resolutions) choisis.



- Remarque 4 : Etant donné le nombre de samples ou la taille, les temps de rendu pour les différentes images sont différents.



sample



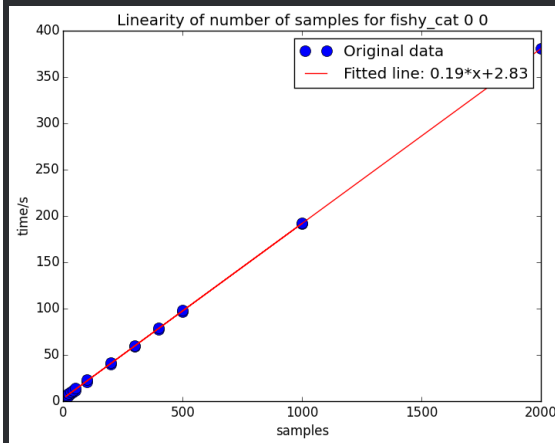
ratio de contraction

- Remarque 3 : Le nombre minimal de sample et la taille minimale qui nous permettent de savoir la pente de la droite dépendent de l'image à rendre. Il faut que le temps de rendu en réglant ces paramètres soit beaucoup plus important que l'erreur de notre mesure du temps. → restant à voir.



Linéarité entre le temps de rendu et le nombre de samples et la taille d'image

Samples



x: nombre de samples

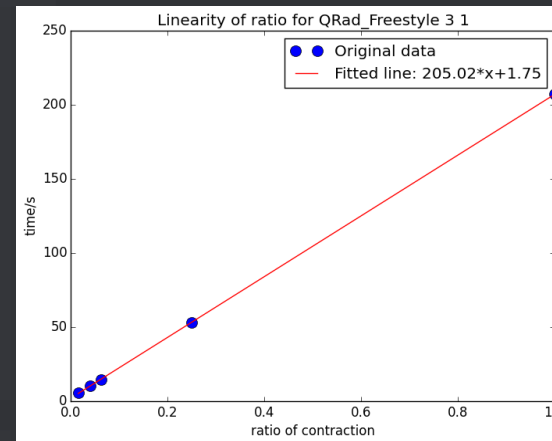
y: temps de rendu

La droite : obtenue par la régression linéaire

On a effectué cette démarche sur 96 image différentes.

Les r-squareds sont plus grand que 0,997 (qualité de régression). Les valeurs absolues des termes de constant (ordonnée à l'origine) sont moins que 3s.

Ratio de contraction



x: ratio de contraction

de la taille d'image

y: temps de rendu

La droite : obtenue par la régression linéaire

On a effectué cette démarche sur 80 image différentes.

Les r-squareds sont plus grand que 0,999 (qualité de régression). Les valeurs absolues des termes de constant (ordonnée à l'origine) sont moins que 8s.

3

Approche préconisée

- Découper l'image de façon fine par rapport au nombre de processeurs avec le nombre de samples faible et / ou la taille petite (le temps de cette démarche : le temps d'exploration T_{explr})
- En fonction des informations obtenues, prédire la meilleure façon de découpage quand le nombre de samples est important et la taille d'image grande.



12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3

0	1	2	3
---	---	---	---

Validité de l'approche

Cette approche est valide en condition que le temps d'exploration plus le temps de rendu en effectuant la façon de découpage préconisée soit moins important que le temps de rendu en effectuant la façon de découpage qu'on utilise maintenant.

Cas favorable : $T_{explr} \leq 26s$
 $T_{meilr} = 1721s$
 $T_{explr} + T_{meilr} \leq 1721 + 26$
 $= 1747s < 1947s$ (Supposons qu'on utilise le découpage de 2 par 2)

Contre exemple : La meilleure façon de découpage est celle qu'on utilise maintenant. Dans le cas extrême, si les temps de rendu sont équitables pour les 4 cases de l'image, on perdrait toujours le temps pour la prédiction. Ce cas est presque impossible à survenir et le nombre de processeurs pourrait être différent que 4.

4

Réduction du temps d'exécution



La mise en place de blender prend du temps chaque fois on rend une image ou une partie d'une image. Donc, on ne peut pas découper une image de façon très fine maintenant.



On tire profit de l'affichage du temps dans blender. En utilisant l'affichage du temps dans blender, on peut découper une image de façon très fine et savoir le temps de rendu pour chaque petite partie en faisant le rendu seulement une fois.

Après modification :

```
Fra:1 Mem:797.96M (0.00M, Peak 892.04M) | Time:02:24.23 | Remaining:02:52.77 | Mem:482.04M,  
Peak:482.04M | Scene_cat, RenderLayer | Path Tracing Tile 10/16  
Fra:1 Mem:797.96M (0.00M, Peak 892.04M) | Time:02:32.79 | Remaining:02:35.68 | Mem:482.04M,  
Peak:482.04M | Scene_cat, RenderLayer | 0 240 tile finished
```


5

Algorithme général pour trouver le découpage “optimal” – Algorithme de Tétris

But

Trouver la meilleure façon de découpage (= la façon de découpage la plus équitable) étant donné les temps de rendu pour les petites parties d'image pour un nombre de samples beaucoup plus faible et / ou une taille beaucoup plus petite et le nombre des processeurs.

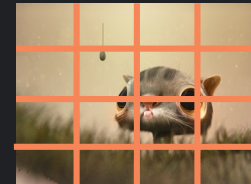
Contrainte

Il faut que chaque partie de découpage soit un rectangle.

Hypothèse

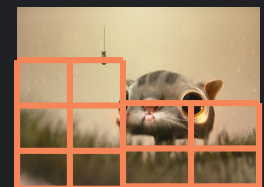
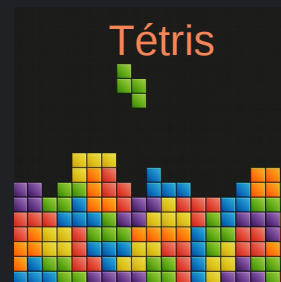
le temps de rendu total d'image = la somme des temps de rendu de chaque partie de découpage = la somme des temps de rendu de chaque petit rectangle pour une façon de découpage fine

→ La façon de découpage “idéale” : le temps de rendu de chaque partie de découpage = le temps total de rendu / nombre (noté n_{proc}) de processeurs (le temps de rendu moyen T_m)



Algorithme de Tétris

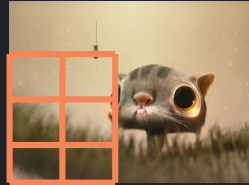
Remplir l'image avec n_{proc} rectangles dont le temps de rendu est le plus proche de T_m





Algorithme de Tétris

- Parmi les rectangles possibles, trouver le rectangle dont le temps de rendu est le plus proche de T_m



- Changer le bord de début et effectuer le même processus



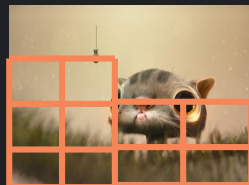
chercher

Répéter la recherche

Répéter le processus sur les trois autres bords

Choisir

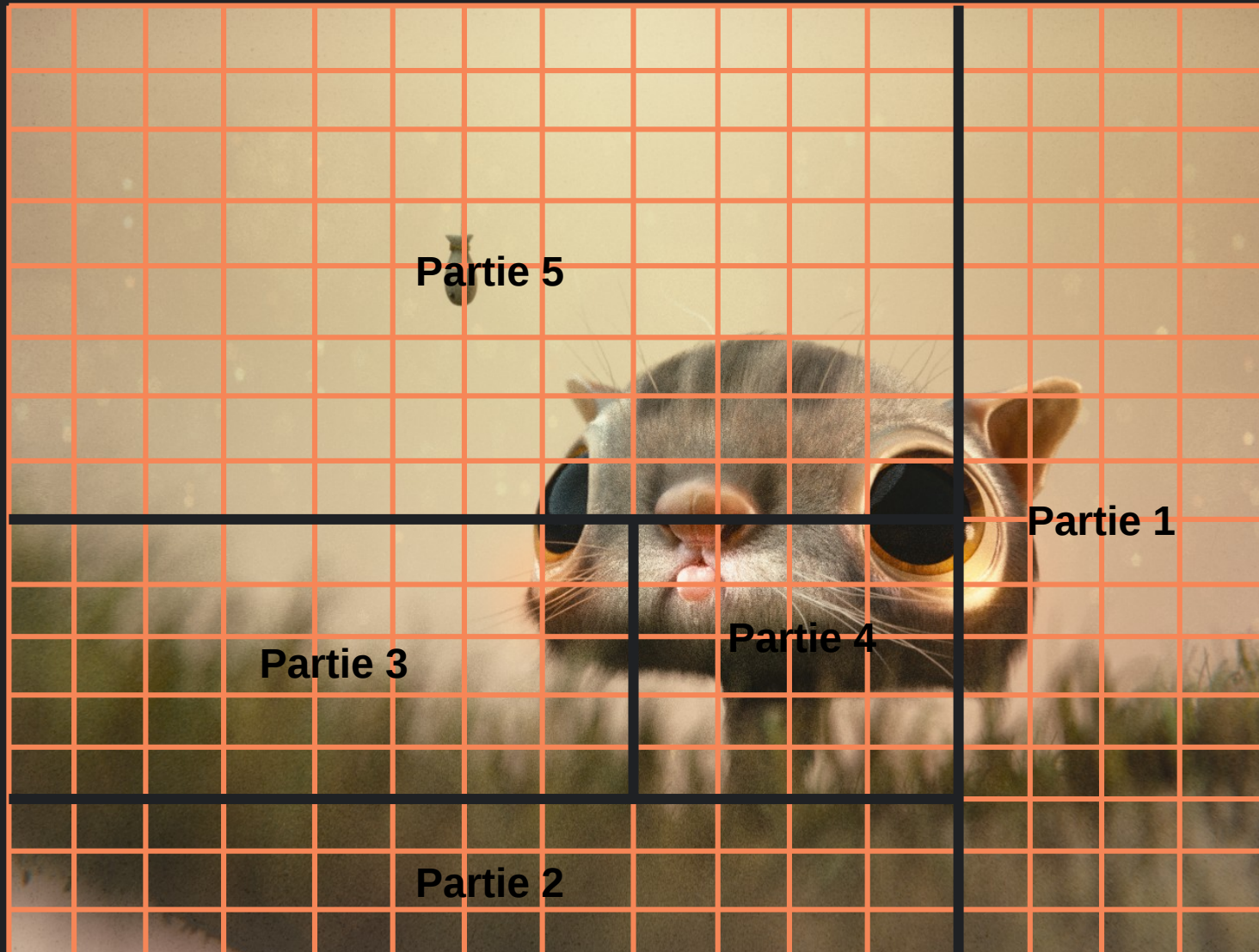
- Parmi les rectangles possibles, trouver le rectangle dont le temps de rendu est le plus proche de T_m



Parmi les quatre façons de découpage trouver, choisir celle qui prend le moindre temps

5

Résultat d'une exécution de l'algorithme de
Tétris (découpage en 16 par 16 ; nproc = 5)



Le temps de rendu
avec le nombre de
sample grand et une
taille grande

T1 = 1005s

T2 = 1011s

T3 = 996s

T4 = 952s

T5 = 971s

1

Modifier le fichier
subframe.py pour que
on puisse effectuer un
découpage de
n'importe quelle façon
sur les q-rads

2

Déterminer le nombre
de samples et la taille
à utiliser dans la
prédiction de façon de
découpage pour
chaque image