

## Compte Rendu TP3 POOCS

### Compression d'une Image par Maillage

JOHN GLIKSBERG

M1 IHPS @ UVSQ

#### Introduction

Le but de ce TD/TP est de fournir un outil qui génère une maille de triangles à partir d'une image pour réduire la quantité d'information stockée d'une image.

Le maillage consistera en un ensemble de triangles dont les points sont des pixels de l'image. Chaque pixel retiendra la couleur de l'image en ce point. On pourra retrouver un affichage de l'image d'origine en dessinant les triangles remplis de la couleur moyenne de celles de ses points.

La "compression" se fait sur la formation de triangles qu'on privilégiera dans les zones avec des grandes différences de couleur.

On n'alignera *pas* les côtés des triangles avec les lignes qu'on peut distinguer dans l'image mais on construira toujours de nouveaux triangles dans la zone qui diffère le plus de l'image (sauf lorsque l'on a atteint une résolution trop fine).

C'est ainsi un processus itératif car on ne prévoit pas où sera le prochain découpage.

On décide d'avance qu'on ne travaillera que sur des images en niveaux de gris. Pour la modélisation de notre outil on décide de se limiter à des fichiers en entrée au format `.pgm` et un enregistrement du maillage au format `.vtk`.

**Définitions:** Dans un maillage  $M$  donné, pour un triangle  $T$  dans  $M$  donné, si  $C_1$ ,  $C_2$  et  $C_3$  sont les couleurs des points de  $T$ , on définit  $C$  la couleur moyenne du triangle, et l'écart en  $T$  correspond à la différence absolue entre  $C$  et la couleur de l'image au barycentre des points du triangle.

Ceci nous permet d'une itération à l'autre de choisir le triangle au plus grand écart dans notre maillage. On travaille maintenant sur le barycentre  $P$  de ce triangle, c'est le point à partir duquel on va rendre notre maillage plus précis.

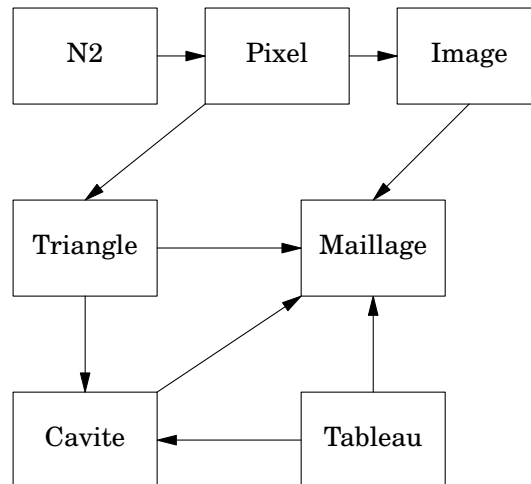
Le processus précis est décrit dans l'énoncé et consiste à vider une cavité de triangles autour de  $P$  et reconstruire de nouveaux triangles plus petits.

Cet algorithme de maillage privilégiera souvent trop certaines régions de l'image, mais en limitant la granularité à celle du pixel, on répartit l'importance donnée aux zones de l'image.

## Mise en place

On a implémenté les différentes parties de notre outil avec pour chacune une classe décrite entièrement dans un fichier `.hh` et un fichier `.cc` si ce n'est pas une classe template (dans quel cas on construit un objet `.o` pour cette classe).

Voici un graphe d'inclusion qui représente ces classes selon l'inclusion de leur *header*.



```
POOCS.hh  
Vect.hh Vect.cc  
Mat.hh Mat.cc  
MatG.hh MatG.cc  
MatC.hh MatC.cc  
main.cc  
Makefile
```

## Accesseurs

## Mutateurs

## Opérations d'algèbre linéaire

- |                                |                                 |
|--------------------------------|---------------------------------|
| • Vect Vect::operator+(Vect)   | addition vectorielle            |
| • Vect Vect::operator-(Vect)   | soustraction vectorielle        |
| • double Vect::operator*(Vect) | produit scalaire                |
| • Vect Vect::operator*(double) | produit vecteur par un scalaire |
| • Vect Mat::operator*(Vect)    | produit matrice vecteur         |