



HAI804I - HAI809I  
Projet Image et Compression  
Mosaïque d'images avec critères avancés  
Compte rendu n°3

Jalbaud Lucas

Reynier Théo

M1 IMAGINE  
Faculté des Sciences  
Université de Montpellier

9 Mars 2025



### Résumé

Ce compte rendu de la semaine du 3 mars au 9 mars illustre l'implémentation d'un critère moyenné pour chaque bloc de l'image, ainsi que l'implémentation d'une génération de mosaïque à partir du critère moyenné.

# 1 Extraction de la moyenne pour chaque bloc de l'image

L'objectif est d'implémenter une fonction qui divise une image en blocs de taille fixe et applique la valeur moyenne des pixels de chaque bloc à l'ensemble de ce bloc.

## 1.1 Segmentation en bloc

Tout d'abord, on découpe notre image en sous-blocs, tous de même taille. Chaque bloc est défini par un paramètre `tailleBloc (tB)`, qui représente la largeur ( $k$ ) et la hauteur ( $l$ ) du bloc en pixels.

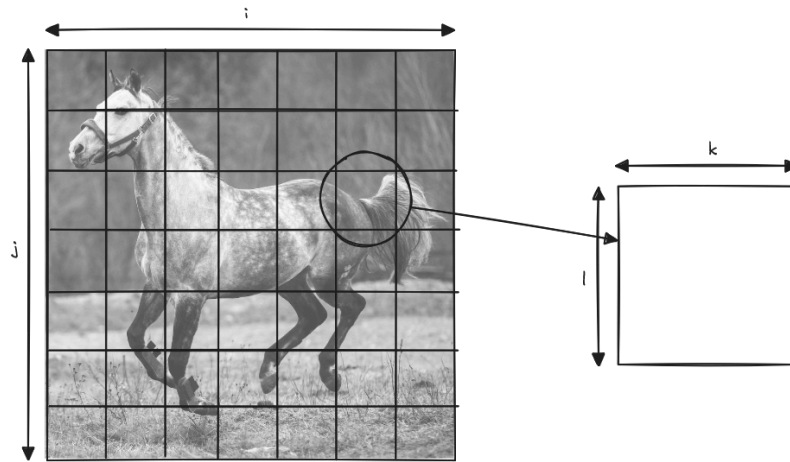


Fig. 1. – Segmentation de l'image en bloc

## 1.2 Calcul de la moyenne par bloc

Une fois l'image segmentée, on calcule la valeur moyenne des pixels de chaque bloc. Cette valeur est ensuite attribuée à chaque pixel du bloc. On écrit les formules suivantes ( $tB$  : Taille du bloc).

$$\text{Somme} = \sum_{k=0}^{tB-1} \sum_{l=0}^{tB-1} I(i \times tB + k, j \times tB + l)$$

$$\text{Moyenne} = \frac{S}{tB^2}$$

## 1.3 Implémentation

On écrit donc un programme `bloc_moyen.cpp` qui va découper l'image d'entrée en bloc, calculer la moyenne de chaque bloc et rendre une image découpée en blocs de pixels moyennés.

La signature de cette fonction est la suivante :

**Entrée** : Les chemins de l'image d'entrée, de sortie et la valeur de la taille des blocs.

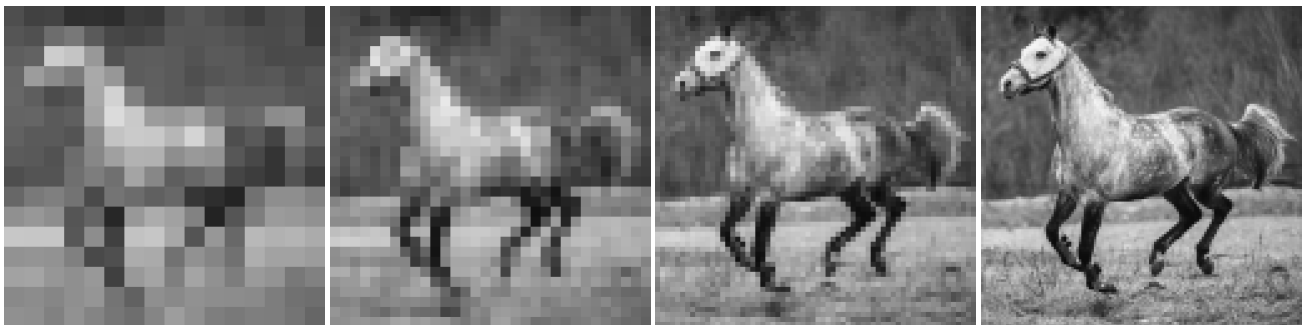
**Sortie** : L'image d'entrée segmentée en blocs.

## 1.4 Résultats

On teste ensuite notre segmentation sur des images en  $1024\text{px} \times 1024\text{px}$  puis  $512\text{px} \times 512\text{px}$  :



Fig. 2. – Image  $1024 \times 1024$



(a) Bloc  $64 \times 64$

(b) Bloc  $32 \times 32$

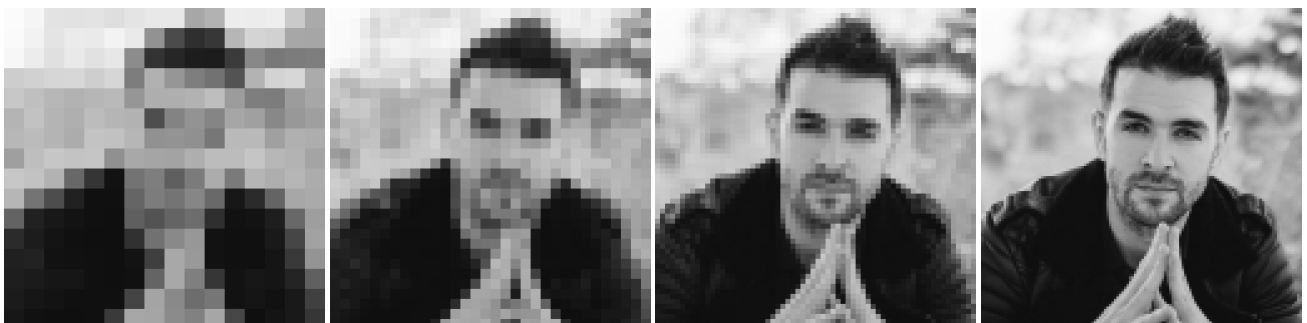
(c) Bloc  $16 \times 16$

(d) Bloc  $8 \times 8$

Fig. 3. – Segmentation selon la taille du bloc (image  $1024 \times 1024$ )



Fig. 4. – Image  $512 \times 512$



(a) Bloc  $32 \times 32$

(b) Bloc  $16 \times 16$

(c) Bloc  $8 \times 8$

(d) Bloc  $4 \times 4$

Fig. 5. – Segmentation selon la taille du bloc (image  $512 \times 512$ )

## 2 Génération de l'image mosaïque

### 2.1 Banque d'imagettes

Tout d'abord, on récupère une vaste variété d'images<sup>1</sup> pgm qui vont nous servir à créer notre mosaïque.

### 2.2 Création de la mosaïque

Afin d'obtenir une image mosaïque, on reprend notre segmentation précédente et on va remplacer nos blocs moyennés par des imagettes correspondantes. On compare la valeur moyenne de chaque imagette puis on choisit celle dont la différence avec la moyenne du bloc est la plus faible.

$$\partial = |\text{moyennes}[k] - \text{moyenne}|, k \in [0, 10000]$$

On doit également réduire la taille des imagettes afin qu'elle s'intègre à la taille des blocs.

$$x_{\text{source}} = \frac{k * nH_{\text{img}}}{tB}, k \in [0, tB[$$

$$y_{\text{source}} = \frac{l * nW_{\text{img}}}{tB}, k \in [0, tB[$$

$$\text{Bloc}[k \times tB + l] = \text{Imagette}[x_{\text{source}} \times nW_{\text{img}} + y_{\text{source}}]$$

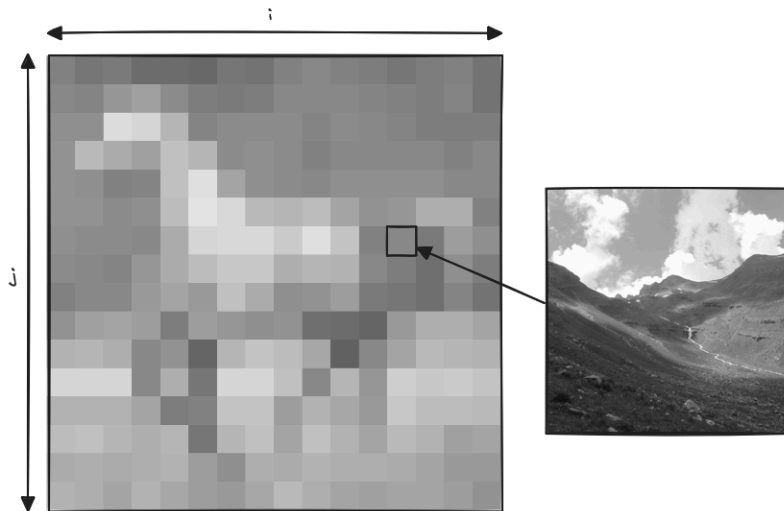


Fig. 6. – Remplacement du bloc par une imagette

### 2.3 Implémentation

On écrit un programme `mosaique_moy.cpp` qui reprend `bloc_moyen.cpp` et qui remplace les blocs moyennés par une imagette correspondante.

La signature de cette fonction est la suivante :

**Entrée :** Les chemins de l'image d'entrée, de sortie et la valeur de la taille des blocs.

**Sortie :** L'image d'entrée en mosaïque d'images.

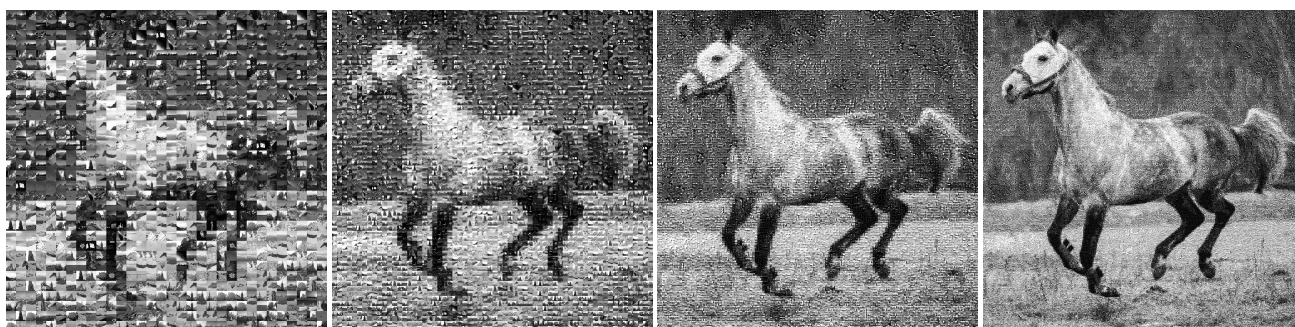
---

<sup>1</sup> <https://data.mendeley.com/datasets/kb3ngxfmjw/1>

On teste notre mosaïque sur les images précédentes :



Fig. 7. – Image  $1024 \times 1024$



(a) Bloc  $32 \times 32$

(b) Bloc  $16 \times 16$

(c) Bloc  $8 \times 8$

(d) Bloc  $4 \times 4$

Fig. 8. – Mosaïque selon la taille du bloc (image  $1024 \times 1024$ )



Fig. 9. – Image  $512 \times 512$



(a) Bloc  $32 \times 32$

(b) Bloc  $16 \times 16$

(c) Bloc  $8 \times 8$

(d) Bloc  $4 \times 4$

Fig. 10. – Mosaïque selon la taille du bloc (image  $512 \times 512$ )

### 3 Conclusion

Pour conclure, on a vu cette semaine comment segmenter notre image en bloc de pixels et comment remplacer ces blocs par une image dont la valeur moyenne en niveau de gris est équivalente à celle du bloc.



(a) Image originale



(b) Bloc  $16 \times 16$



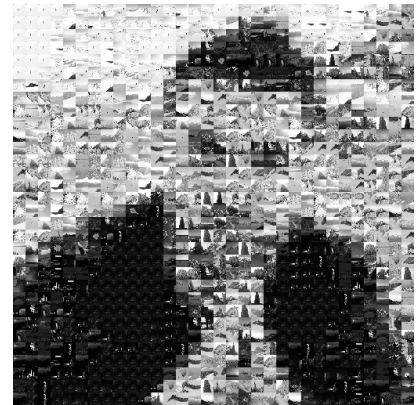
(c) Mosaïque d'images



(d) Image originale



(e) Bloc  $16 \times 16$



(f) Mosaïque d'images

Fig. 11. – Transformation d'une image en mosaïque d'images utilisant la moyenne