



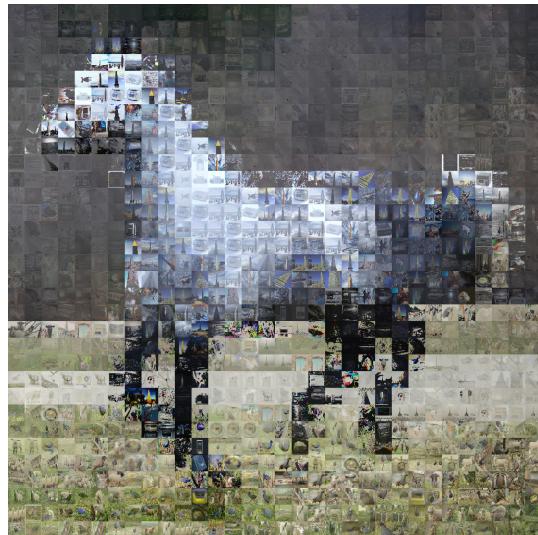
HAI804I - HAI809I
Projet Image et Compression
Mosaïque d'images avec critères avancés
Compte rendu n°6

Jalbaud Lucas

Reynier Théo

M1 IMAGINE
Faculté des Sciences
Université de Montpellier

30 Mars 2025



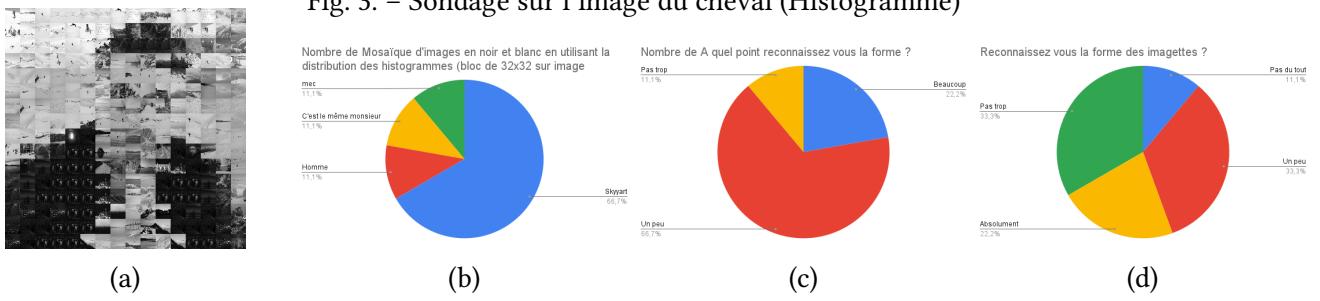
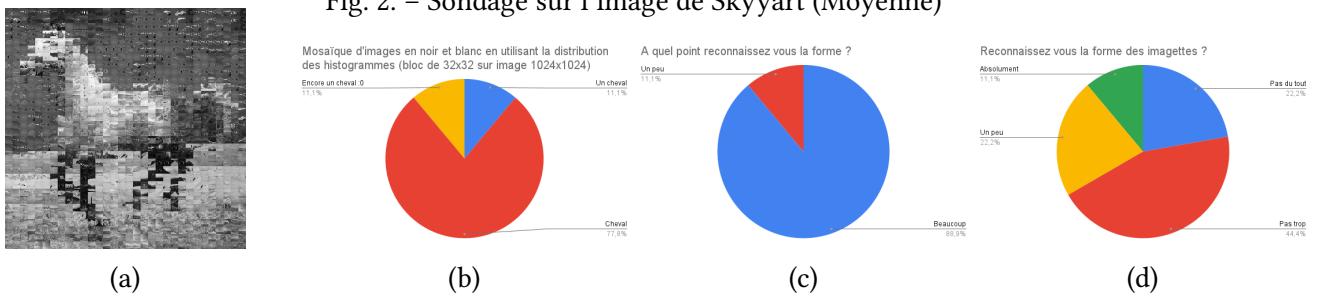
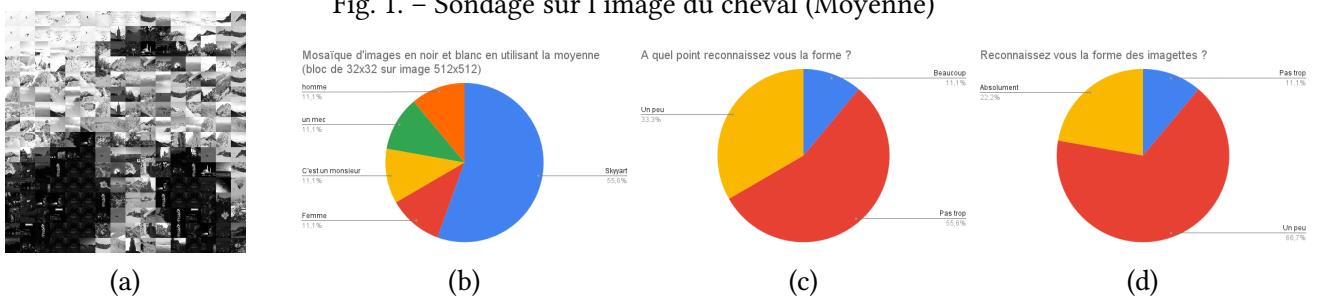
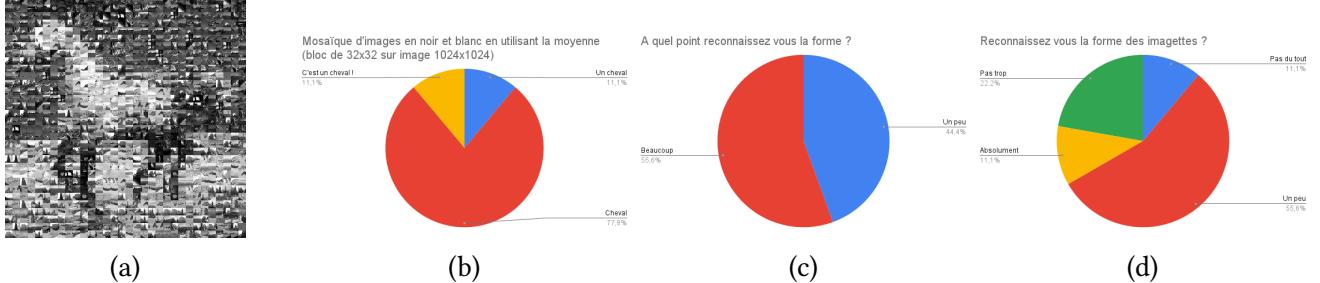
Résumé

Ce compte rendu de la semaine du 24 au 29 mars illustre un sondage réalisé sur la reconnaissance des mosaïques, l'implémentation d'une génération de mosaïque en utilisant une spécification de l'histogramme des blocs de l'image d'origine et également un début d'algorithme de détection des formes dans l'image pour la mosaïque.

1 Sondage

On a recueilli l'avis de personnes extérieur au projet et réalisé des statistiques sur la reconnaissance d'une image et des imagettes sur une mosaïque.

1.1 Niveaux de gris



On a également demandé quelle méthode était selon eux la meilleure :



Fig. 5. – Avis sur la meilleure méthode entre Moyenne et Histogramme

1.2 Couleur

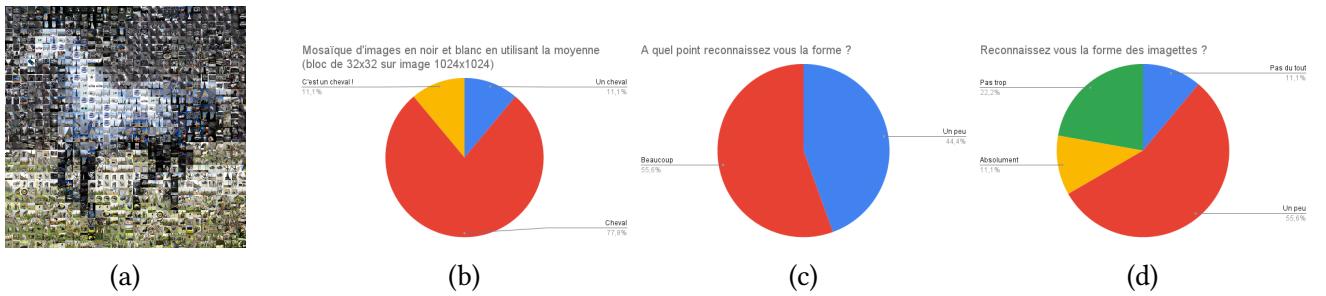


Fig. 6. – Sondage sur l'image du cheval (Moyenne)

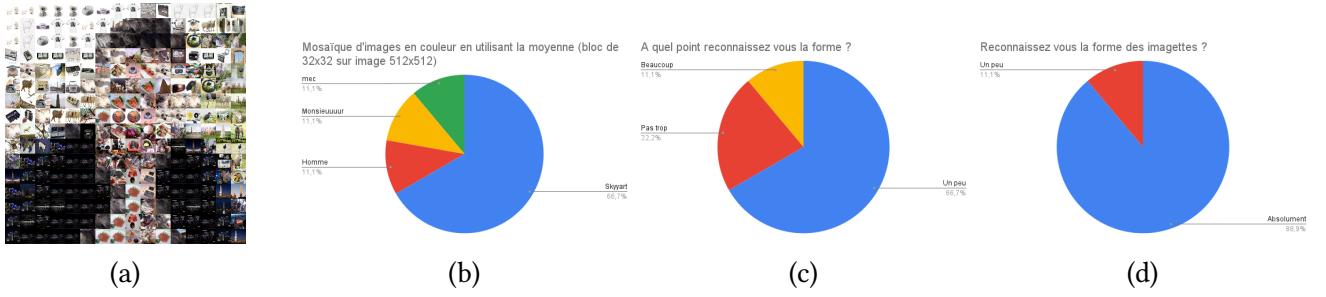


Fig. 7. – Sondage sur l'image de Skyyart (Moyenne)

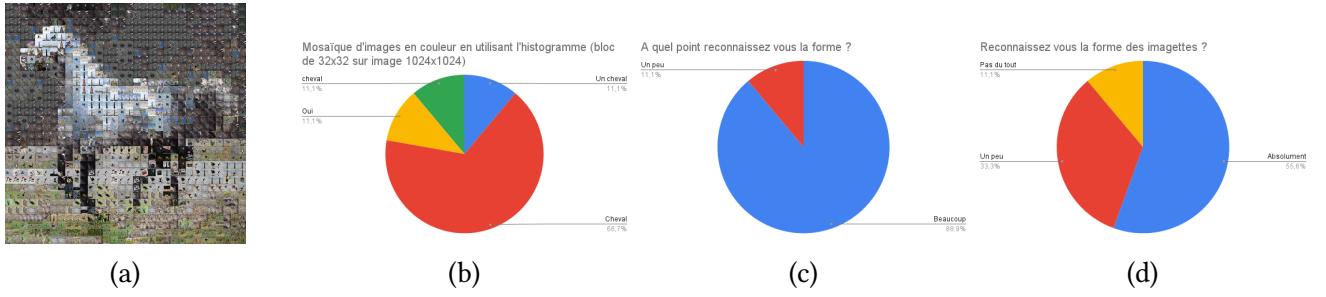


Fig. 8. – Sondage sur l'image du cheval (Histogramme)

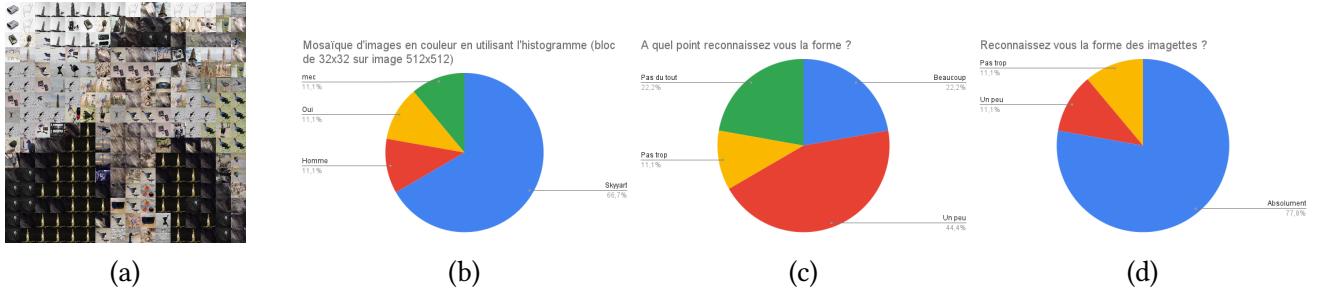


Fig. 9. – Sondage sur l'image de Skyyart (Histogramme)

On a également demandé quelle méthode était selon eux la meilleure :

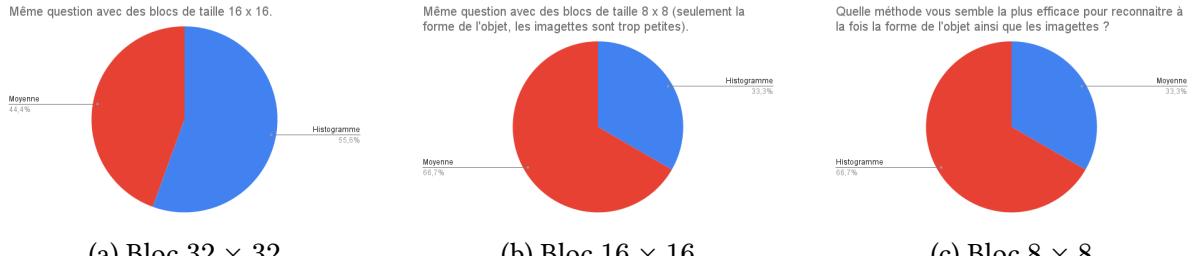


Fig. 10. – Avis sur la meilleure méthode entre Moyenne et Histogramme

1.3 Résumé des résultats

Dans l'ensemble, les avis sont favorables à la reconnaissance des objets cachés derrière la mosaïque. Les imagettes sont néanmoins beaucoup partagés. Selon les graphiques, les personnes trouvent que le critère de l'histogramme est meilleur que le critère de la moyenne.

On a également demandé s'il trouvé que l'ajout de couleur aux images, leur permettait de mieux reconnaître l'image globale ainsi que les imagettes.



Fig. 11. – Sondage sur l'utilité de la couleur

2 Spécification d'histogramme

2.1 Niveaux de gris

On reprend le programme `mosaique_moy.cpp` qui permet de remplacer les blocs par des imagettes en utilisant leurs moyennes et on écrit un nouveau programme `mosaique_specifier.cpp` qui reprend le fonctionnement du programme principal, mais cette fois-ci en spécifiant les niveaux de gris de chaque bloc aux imagettes. La signature de cette fonction est la suivante :

Entrée : Les chemins de l'image d'entrée, de sortie, la valeur de la taille des blocs et le nombre d'imagettes utilisés.

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

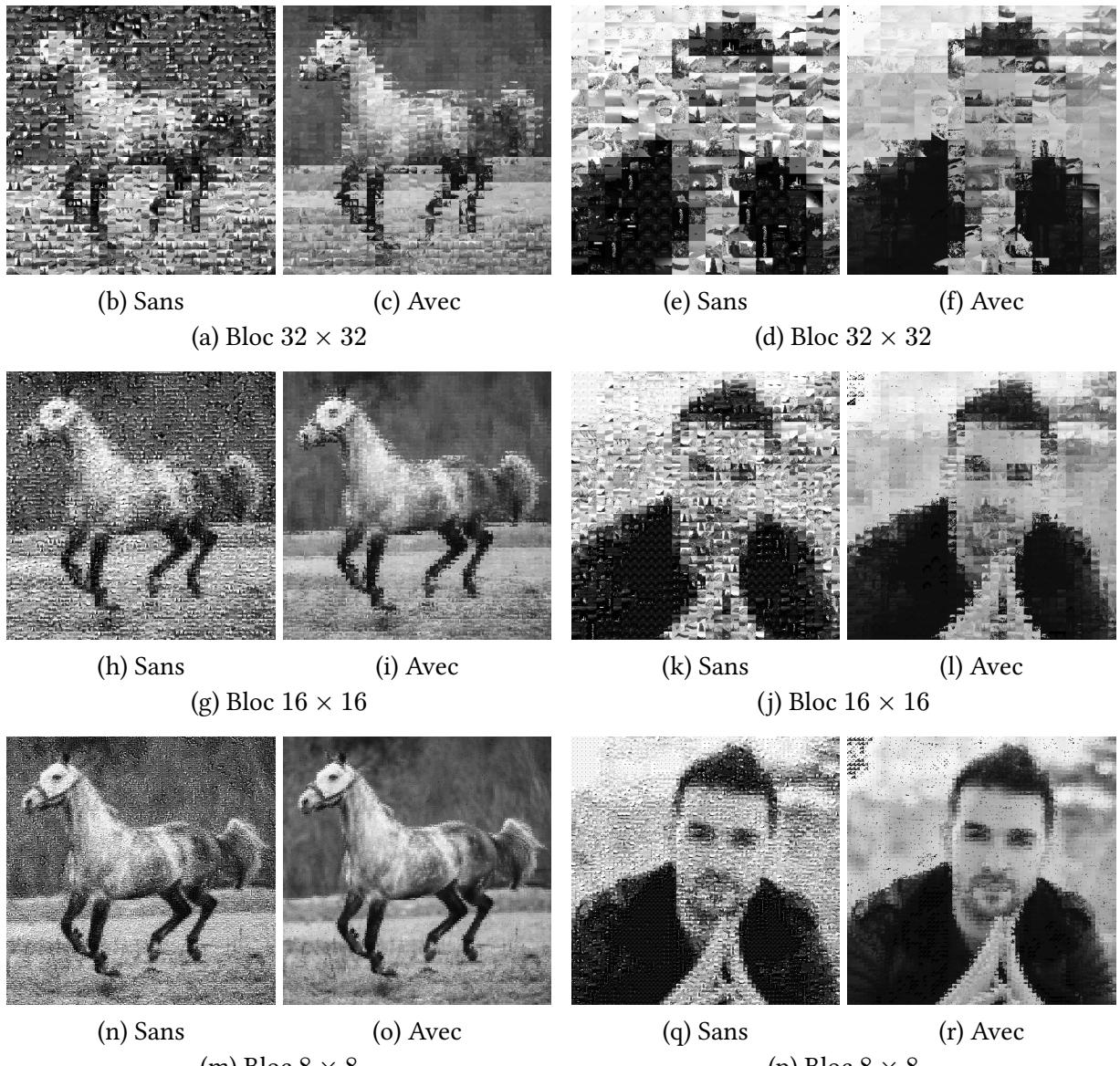


Fig. 12. – Comparaison moyenne/histogramme sur une image 1024×1024 et 512×512

On constate une amélioration dans la reconnaissance de l'objet derrière la mosaïque, mais on perd de la reconnaissance au niveau des imagettes. Au niveau des performances, le programme a une exécution en temps similaire au programme de référence et on gagne un peu plus de PSNR.

2.2 Couleur

On reprend le programme `mosaique_moy_RGB.cpp` qui permet de remplacer les blocs par des imagettes en utilisant leurs moyennes et on écrit un nouveau programme `mosaique_specifier_RGB.cpp` qui reprend le fonctionnement du programme principal, mais cette fois-ci en spécifiant les couleurs RGB de chaque bloc aux imagettes. La signature de cette fonction est la suivante :

Entrée : Les chemins de l'image d'entrée, de sortie, la valeur de la taille des blocs et le nombre d'imagettes utilisées.

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



Fig. 13. – Comparaison moyenne/histogramme sur une image 1024×1024 et 512×512

On constate les mêmes résultats pour la version couleur, ainsi qu'une performance du programme similaire

3 Détection des formes

La détection des formes se fait en 5 étapes appliquées sur toutes les imagettes et les blocs de l'image. On commence par réduire le bruit présent dans notre image en appliquant un filtre moyenneur.



Fig. 14. – Application d'un filtre moyenneur

Ensuite, on applique un filtre Laplacien pour pouvoir mettre en évidence les contours de l'image.

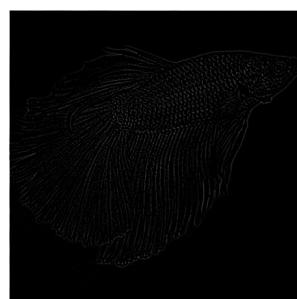


Fig. 15. – Laplacien

Plutôt que de regarder les contours de l'image avec les gradients horizontal et vertical on calcule le vecteur du gradient sous sa forme polaire.



Fig. 16. – Laplacien avec Phase

Après ça il suffit d'affecter à la norme du gradient tous les passages à zéro.



Fig. 17. – Laplacien avec Gradient

Enfin, on applique un seuillage par hystérésis pour affiner les contours et conserver seulement les plus cohérents.



Fig. 18. – Laplacien avec hystérésis

On peut maintenant calculer l'histogramme de chaque image des contours pour pouvoir ensuite la comparer à d'autres images et trouver les formes les plus proches, indépendamment du niveau de gris. Pour le moment, nous n'avons pas réussi à obtenir un résultat convaincant pour la détection des contours, car la forme n'est pas visible sur la mosaïque d'imagettes.

La signature de cette fonction est la suivante :

Entrée : Les chemins de l'image d'entrée, de sortie, la valeur de la taille des blocs et le nombre d'imagettes utilisées.

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

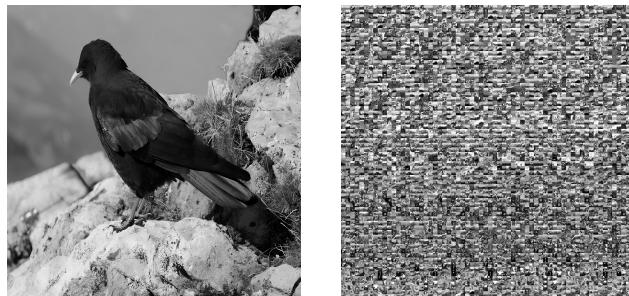


Fig. 19. – Résultat d'un appel avec taille des blocs 32×32 et nombre d'imagettes = 1000

4 Conclusion

Cette semaine, on a réuni les différents résultats obtenus dans un sondage afin de se donner une idée des avis de personnes extérieures. On a également implémenté une méthode utilisant la spécification d'histogramme sur le critère de la moyenne. Cette méthode améliore la reconnaissance de l'objet de base sans trop modifier les imagettes de la mosaïque.

Pour les dernières semaines à venir, on va essayer de corriger l'algorithme du filtre laplacien ainsi qu'intégrer nos divers programmes dans une application afin de pouvoir faire des démonstrations simples.