



CR3 Projet Image 2.3

BELOT Mathieu, SERVA Benjamin, KERBAUL Loïc
Master 1 Imagine
Université de Montpellier

March 17, 2024

Contents

1 Réalisations de la semaine	2
1.1 Mosaïque d'images couleur	2
1.2 Création d'une interface graphique utilisant	2
2 Objectifs pour la semaine prochaine	5

1 Réalisations de la semaine

1.1 Mosaïque d'images couleur

Nous avons appliqué les méthodes utilisées la semaine dernière aux images ppm au lieu d'images pgm. Pour cela on a utilisé la base de données sur le site de [l'université de toronto](#) qui contient après extraction 41129 images couleurs de taille 32 par 32.



Figure 1: Image d'ours (à gauche) et mosaïque couleur de l'ours (à droite)

1.2 Crédit à une interface graphique utilisant

Remarques :

- L'application présentée ci-dessous n'implémente pour l'instant que la création d'image mosaïque à partir d'image en niveau de gris (au format *pgm*).
- Sur le dépôt GitHub, le code de l'application se trouve dans la branche *interface* (et non dans la branche *main*)

Afin de pouvoir utiliser nos programmes plus facilement, nous avons réalisé une interface graphique grâce à laquelle nous pouvons faire toutes les étapes de la création d'images mosaïques (redimensionnement des imagettes, listage des imagettes et calcul de leur moyenne, sélection de l'image à transformer). Au final, l'image mosaïque construite s'affiche sur notre application (voir figure 3), ainsi que le Peak Signal Noise Rate (PSNR) entre l'image initiale et l'image mosaïque.

Afin de pouvoir utiliser cette interface (qui utilise la librairie GTK), il est nécessaire d'installer le package *libgtkmm-3.0-dev*. Afin de tester notre application, nous avons fourni un ensemble de 192 imagettes (répertoire *imagettes* dans la branche *interface* sur le dépôt GitHub).

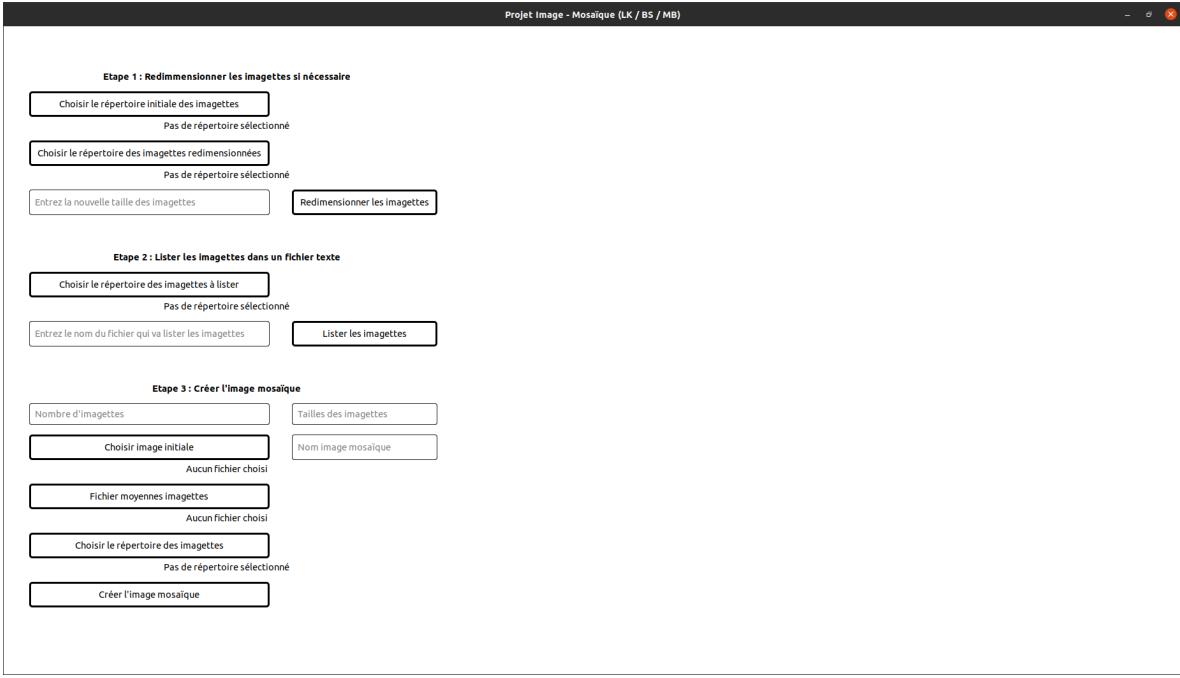


Figure 2: Etat de l'application à son lancement

Sur la capture d'écran plus bas, nous avons générée une image mosaïque en utilisant les 192 imagettes qui sont disponibles dans notre GitHub (uniquement dans la branche *interface*). Celles-ci ont pour cet exemple été redimensionné en 10×10 . Au final, le PSNR entre l'image initiale et la mosaïque est d'environ 9.8 dB, ce qui est assez bas (à cause notamment des imagettes utilisées).

Au niveau des commandes : L'application permet de gérer toutes les étapes de production de nos images mosaïque. Afin de la lancer, il suffit de faire les commandes (pour rappel, dans la branche *interface*) **make** puis **make run**. Ensuite, **make cleanCompiled** permet de supprimer les fichiers compilés sans supprimer tous ce qui concerne les imagettes, et **make cleanImagettes** qui fait l'inverse. Afin de tous supprimer, on peut aussi utiliser **make cleanAll** (à noter que les images mosaïques produites dans le répertoire *dataImg* ne seront pas supprimées).

Attention : Il n'est pas impossible que l'application se stoppe dans le cas où celle-ci serait mal utilisée (par exemple si le nombre renseigné d'imagettes ne correspond pas au nombre réel).

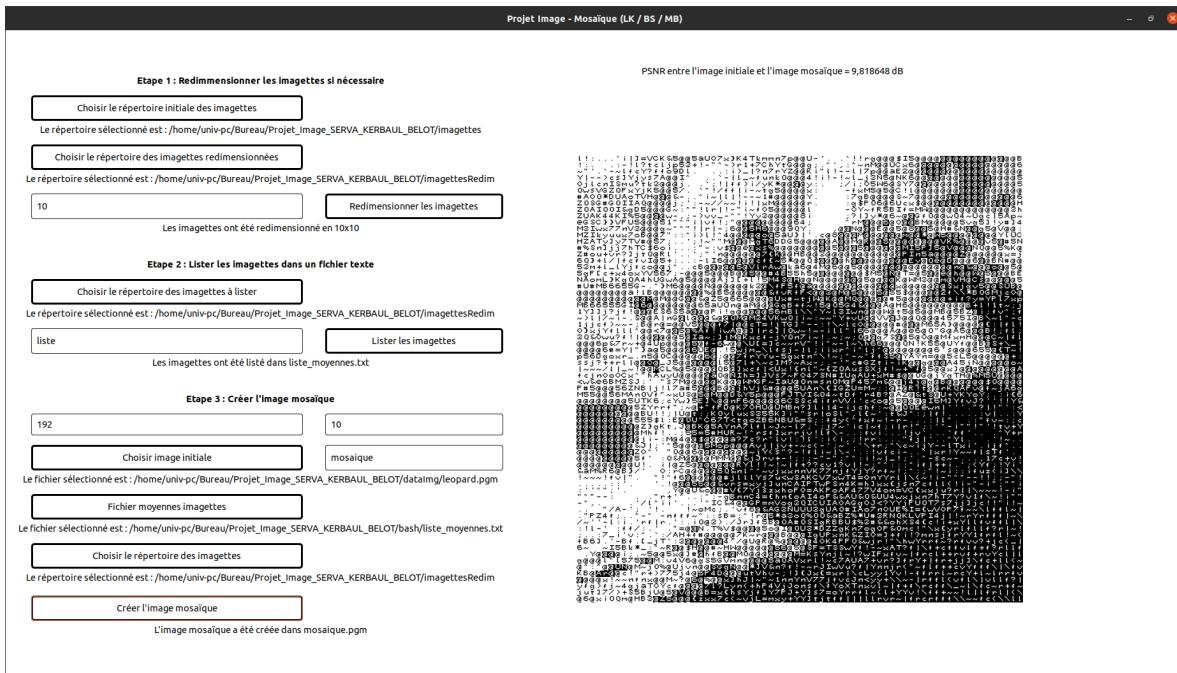


Figure 3: Application avec une image mosaïque générée

2 Objectifs pour la semaine prochaine

- Faire en sorte de pouvoir gérer les mosaïques d'images couleurs via l'application (comme on le fait actuellement pour les images en niveau de gris)
- Utiliser une autre méthode (que celle des moyennes) pour le choix des imagettes à utiliser
- Utiliser une autre méthode pour évaluer nos résultats (au lieu du PSNR qui nous donne des résultats assez bas dû à sa sensibilité à la moindre différence, et dû au fait qu'il ne prenne pas en compte la perception humaine)