



Advanced Methods for Image Processing

Devoir 1, Assignment: Texture Synthesis

WOLSKI Axel

Chargé de TD : BUGEAU Aurelie

Master 2 Informatique Spécialité Image et Son 2017-2018

Octobre 2017

Contents

1	Explications	2
1.1	Texture Synthesis:	3
1.2	Inpainting:	5
2	Conclusion	7
3	Éléments bibliographiques	8

1 Explications

Lors du premier TP nous devons pouvoir créer des patches aléatoire de pixels (2) aléatoire d'une taille donnée à partir d'une texture (1). Cela afin de pouvoir utiliser par la suite ces patches dans les algorithmes que nous avons eu à coder. Cette fonction a été codée dans le fichier **randPatch.m**.



Figure 1: Exemple de texture

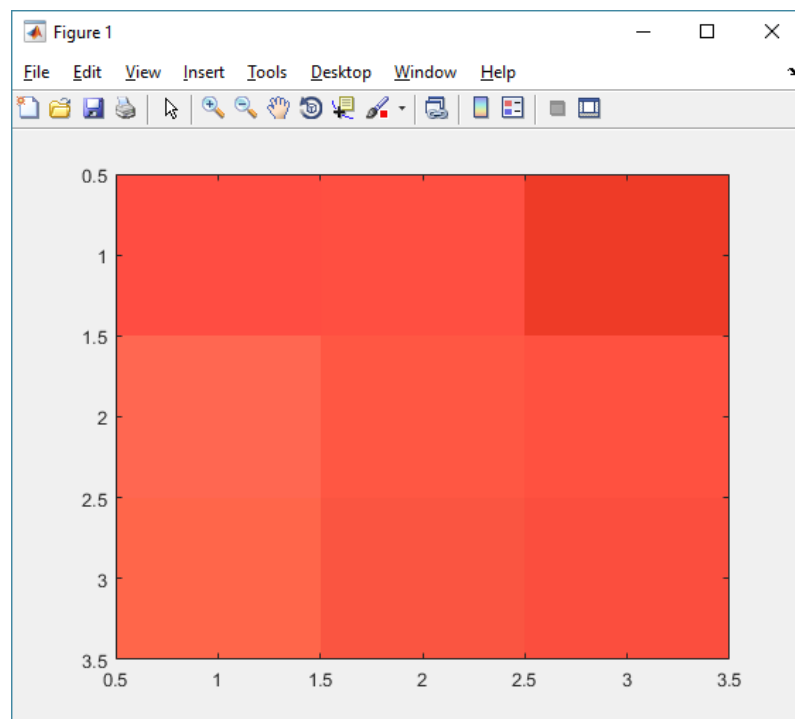


Figure 2: Patch random 3x3 à partir de la texture ci-dessus

1.1 Texture Synthesis:

Le première algorithme que nous avons eu à coder a été l'algorithme de "Texture Synthesis" présent dans le fichier **textureSynth.m**. Celui-ci permet à partir d'un patch aléatoire d'une texture, de reconstruire une texture similaire. Pour cela il faut construire un texture noir (3) que nous appellerons T, dans lequel nous placerons en son milieu le patch prélevé.

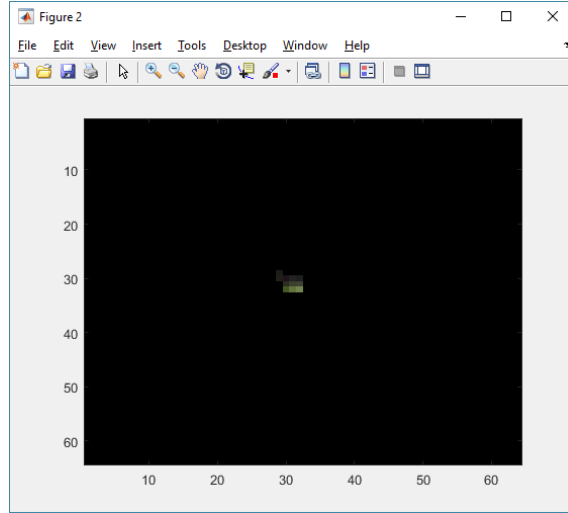


Figure 3: Texture T avec un patch en son centre

Nous allons créer également un mask, celui-ci nous permettra à l'aide d'une dilatation des récupérer le contour de pixels englobant la partie coloré de la texture T. Cela afin de permettre de coloré ce contour pas à pas en avançant en x et y, nous devons calculer la SSD (présente dans **SSD.m**) entre un patch sélectionné dans T en x et y et chaque patch possible pris dans la texture d'origine. Nous pourrions donc placer dans T la couleur correspondant à la texture dans le calcul de la SSD précédent à la valeur la plus basse qui correspond à la plus forte probabilité que le pixels corresponde. Cette méthode sera répétée jusqu'à ce qu'il n'y ai plus de pixel noir dans T.

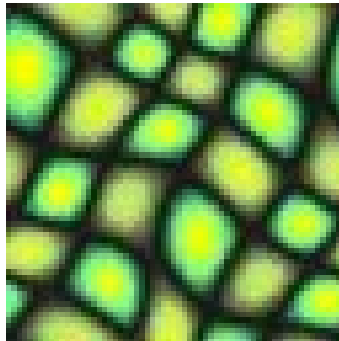


Figure 4: Texture d'origine utilisé pour Texture Synthesis

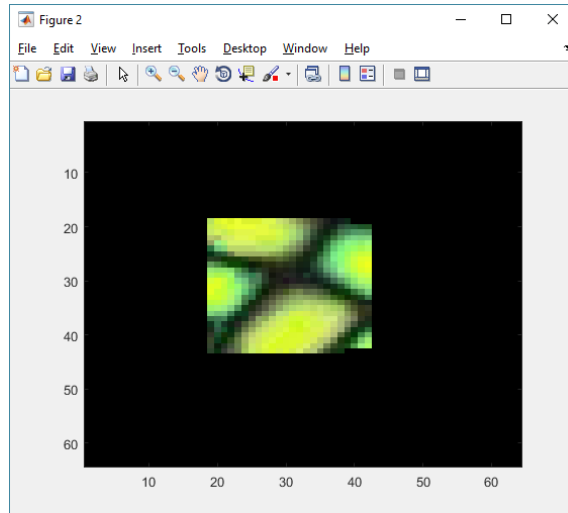


Figure 5: Texture Synthesis en cours de calcul

Afin de pouvoir remplir toute la texture T , il est nécessaire de réaliser une gestion des bords car il faudra sélectionner autrement les patches lorsque nous serons au bords de l'image. Dans le cadre du TP il nous a été possible de ne pas prendre en compte cette gestion des bords. J'ai quand même voulu essayé de coder une gestion des bords sans réussite, celle-ci se trouve dans **myPatch.m** est a été mis en commentaire.

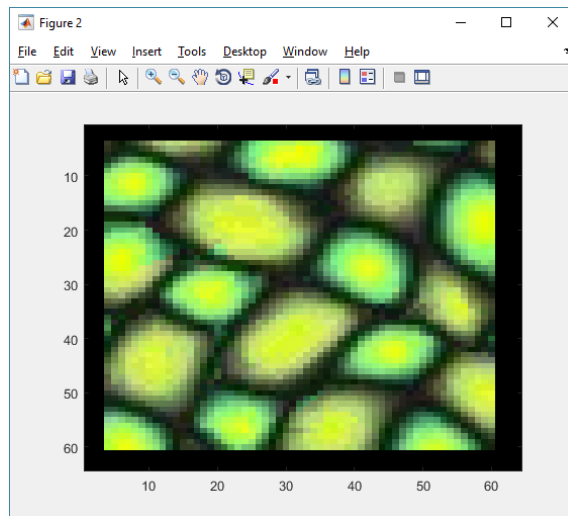


Figure 6: Résultat final obtenu avec Texture Synthesis

1.2 Inpainting:

Le Inpainting a un fonctionnement inverse à celui de Texture Synthesis, nous partons d'une texture qui a des contours et il faut recréer l'intérieur / ou la partie vide de l'image (7). Cette algorithme a été codé dans le fichier **inPainting.m**. Dans un premier temps j'ai repris mon code de Texture Synthesis. Je copie dans ma texture T la texture de base et j'applique par la suite à la texture une zone que je vais remplir de noir afin d'avoir la partie manquante de la texture. Par la suite j'applique le même traitement réalisé précédemment afin de remplir la zone noir, dans un premier temps j'ai voulu éroder le mask au lieu de le dilater car cela me semblé plus logique mais cela ne fonctionnait pas. J'ai donc réutilisé la dilatation comme dans Texture Synthesis.

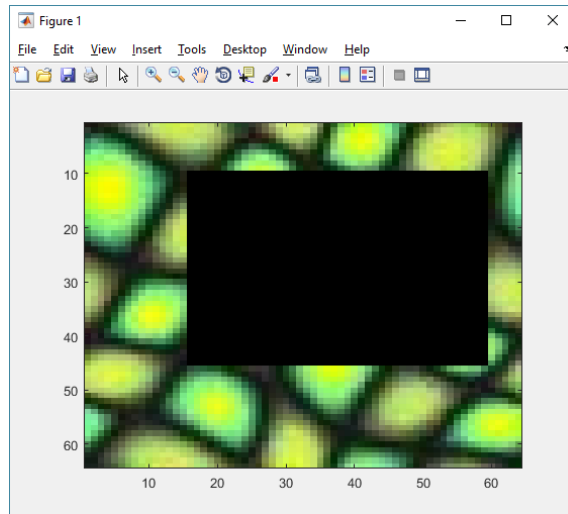


Figure 7: Texture de départ pour le inpainting

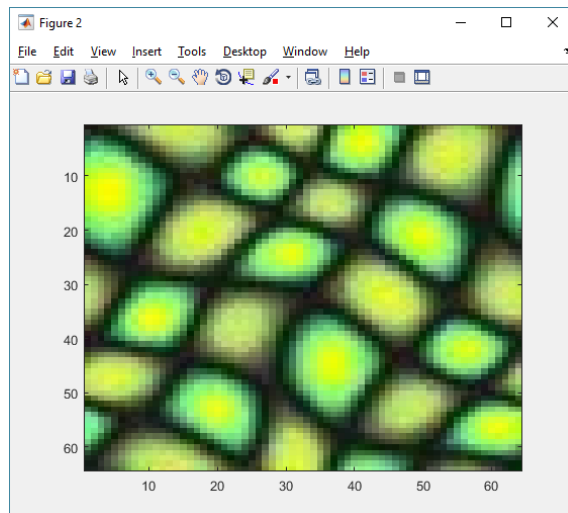


Figure 8: Résultat final de inpainting

J'ai par la suite voulu tester si cela fonctionné lorsque je donné à mon algorithme une image avec une zone noir (9). C'est la que j'ai pu me rendre compte que ma méthode ne fonctionne pas (10).

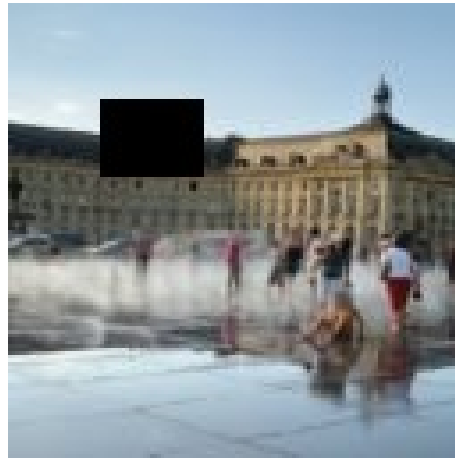


Figure 9: Image avec une zone manquante

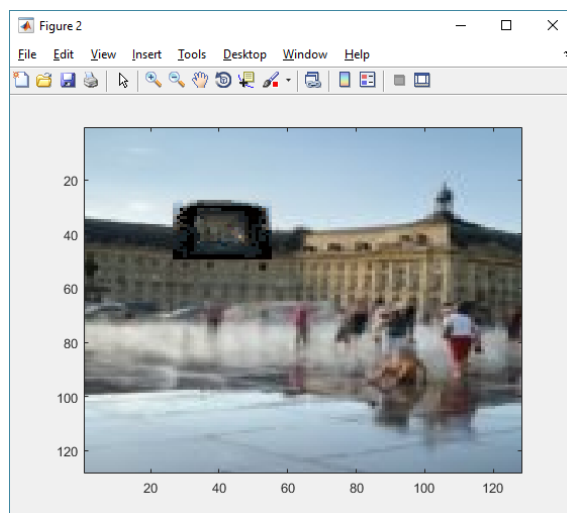


Figure 10: Résultat final de inpainting avec l'image

Afin que cela fonctionne, il me semble qu'il aurait été nécessaire de reprendre tous mon code et changé de méthode.

2 Conclusion

Les résultats obtenu avec l'algorithme de Texture Synthesis sont satisfaisant, j'ai pu le tester sur la deuxième texture **text2.png** et le resultat étant concluant.

L'algorithme de Inpainting fonctionne avec les textures en appliquant une zone noir sur la texture d'origine. Mais il ne fonctionne pas avec une image sur la quel un trou est présent. Afin que cela fonctionne pour ce cas la, il aurait fallu que je ne réutilise pas la méthode fais pour Texture Synthesis et essayer de coder la méthode présenté par mes camarades lors de leur exposé sur le inpainting en cours.

3 Éléments bibliographiques

- Region-Filling and Object Removal by Exemplar-Based Inpainting [1].
- Synthesizing Natural Textures [2].
- Texture Synthesis by Non-parametric Sampling [3].

References

- [1] P. Pérez A. Criminisi and K. Toyama. Region-filling and object removal by exemplar-based inpainting. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/criminisi_ip2004.pdf, 2004.
- [2] M. Ashikhmin. Synthesizing natural textures. <http://www.cs.utah.edu/~michael/ts/ts.pdf>, 2001.
- [3] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. <http://graphics.cs.cmu.edu/people/efros/research/EfrosLeung.html>, 1998.