

Texture-Optimization

1. Lire et comprendre le fichier '*Texture_optimization_TP_exercice.ipynb*'
2. Compléter la fonction *TextureOptimization*
Indication: on utilisera les fonctions définies en en-tête
3. Expérimenter pour plusieurs images (dossier 'textures') en particulier déterminer:
 - 3.1. L'influence de l'initialisation
 - 3.2. Le rôle du sous-échantillonnage spatial (paramètre 'stride')
 - 3.3. L'influence du sous-échantillonnage spatial (paramètre 'N_subsampling')
 - 3.4. Le rôle du nombre d'échelle ('scale')
 - 3.5. L'influence de la taille des patches (voir 'patch_size' et 'patchsize')
 - 3.6. Le rôle du nombre d'itérations
4. Calculer la carte de correspondance au plus proche voisin f des patches entre l'image d'entrée u et de sortie v . On pourra se servir de la fonction '*Patch_NN_search*' pour estimer $f(x)$. Afficher cette carte de correspondance $f(x) = (y_1, y_2)$ en définissant une image $c(x) = (R, G, B)$ dont les canaux de couleurs sont $R=y_1/W$, $G=y_2/H$, $B=0$

Patch-Match

5. Télécharger les sources depuis <https://github.com/ElIBock/patchmatch>
6. Visualiser les résultat de l'algorithme PatchMatch sur des exemples de paires d'images
7. Application à la stéréovision : visualiser le champ de déplacement entre 2 images d'une même scène
(par exemple: <http://vision.middlebury.edu/stereo/data/scenes2014/>)
8. Application à la synthèse de texture en vous inspirant de 'TextureOptimization'