Télécom Physique Strasbourg - Université de Strasbourg - TIC Santé 2A

TP 3 : Images et Vision

Introduction

Objectifs du TP :
- Utilisation de la librairie OpenCV
- Extraction de points caractéristiques, mise en correspondance, assemblage d'images.

Pour la compilation des programmes, on utilisera cmake et le compilateur g++ sous le système Linux. On pourra trouver la documentation d'OpenCV sur le site `http://docs.opencv.org` et un bref résumé de certaines fonctions dans le fichier `http://docs.opencv.org/2.4/opencv_cheatsheet.pdf`.

Application

Nous souhaitons créer une mosaïque à partir de deux images d'une même scène prises de deux points de vue différents. Entre les deux prises, la caméra a effectué une **translation parallèle au plan de l'image**. Nous allons calculer cette translation de façon automatique en recherchant des correspondances entre des points des deux images (voir fig. 1(b)).

Extraction de points caractéristiques

1. Lire et visualiser les images t1.png et t2.png. Les convertir en images en niveaux de gris.
2. Utiliser la fonction `cornerHarris` pour détecter les point caractéristiques de ces deux images. Visualiser les résultats en faisant varier les différents paramètres. On n'oubliera pas de normaliser les images pour la visualisation.
3. Dans la suite, on prendra les paramètres (blockSize = 2, apertureSize = 3, k = 0.04). Normaliser entre 0 et 255 les images résultant de l'utilisation de `cornerHarris` et conserver la liste des points ayant une intensité supérieure à $\tau = 125$. Dans la suite, on appelle C_1 la liste des points de la première image (C_2 pour la seconde). Afficher des cercles autour de ces points dans les images RGB originales (voir fig. 1(a)). S'agit-il bien de points facilement identifiables dans les images ?

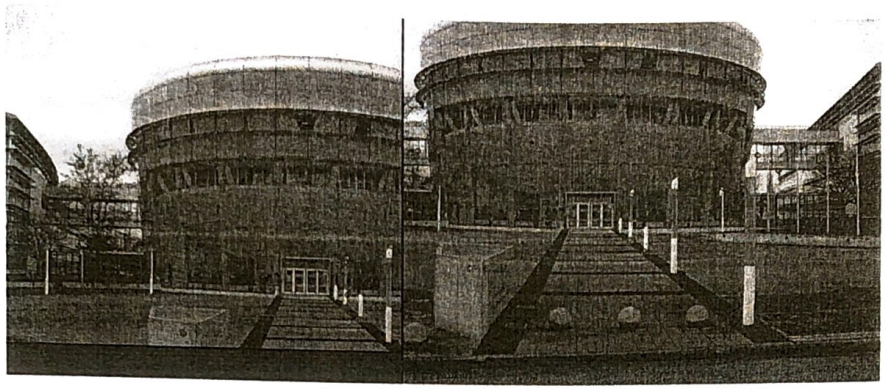
Mise en correspondence

1. Pour chaque point $p_1(x_1, y_1)$ de C_1, trouver le point p_2 de C_2 le plus similaire en comparant les voisinages. On comparera les voisinages en utilisant la fonction de similarité SSD définie comme suit :
$$\sum_{u,v}(I_1(x_1+u, y_1+v) - I_2(x_2+u, y_2+v))^2,$$
avec $-w \leq u, v \leq w$ et $w = 7$ pixels.
2. Copier les deux images côte-à-côte dans une image plus grande. Afficher ces correspondences en tracant des lignes entre elles (voir fig. 1(b)).

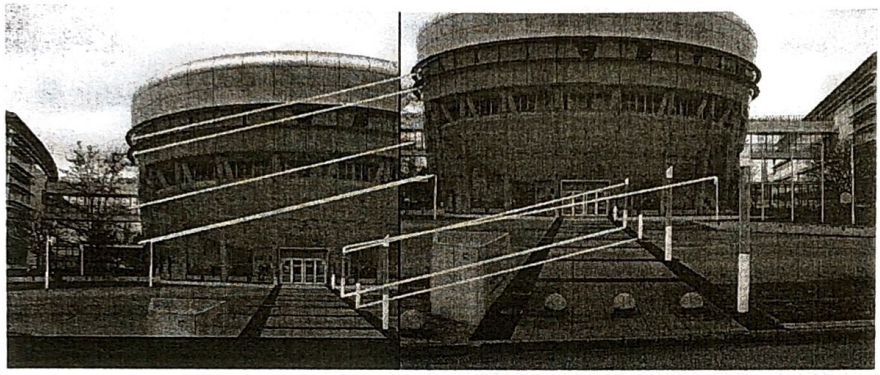
3. Eliminer les correspondances qui ne sont pas symétriques, c'est à dire les correspondances (p_1, p_2) telles que p_1 n'est pas le point de C_1 le plus similaire à p_2. Afficher ces nouvelles correspondances et les comparer aux précédentes.

Création de la mosaïque

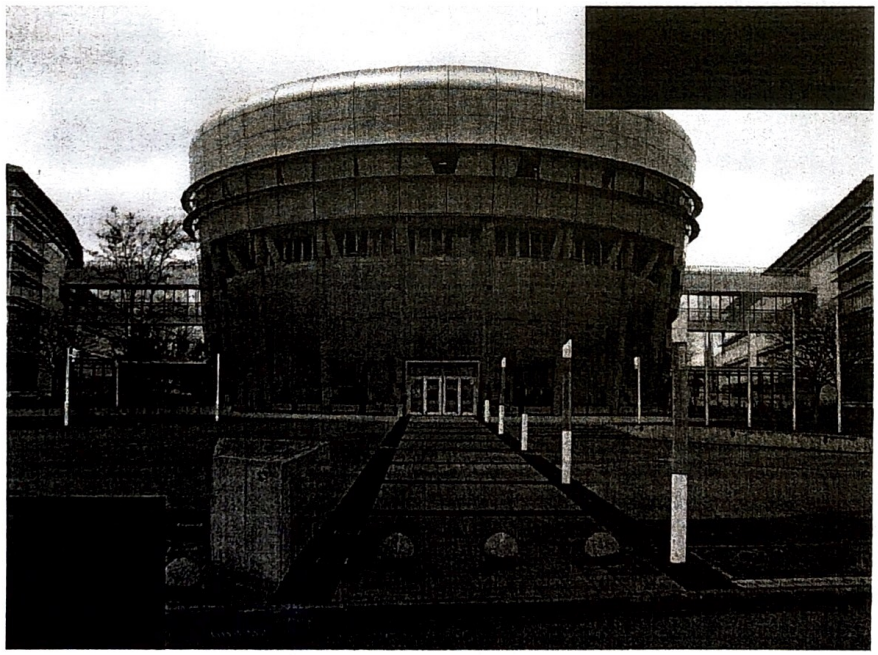
1. Calculer la translation T entre les deux images comme la moyenne des vecteurs de mise en correspondance.
2. Utiliser T pour copier les deux images d'origine dans une image plus grande et visualiser la mosaïque.
3. Si la mosaïque n'est pas parfaite, c'est que les correspondances et donc la translation estimée ne sont pas suffisamment précises. Proposer une méthode pour éliminer les mauvaises correspondances, sachant que la majorité des correspondances sont correctes.
4. Tester la méthode implémentée pendant le TP sur les images médicales c1.png et c2.png (voir fig. 2). Il pourra être nécessaire de faire varier certains paramètres lors de la détection des points caractéristiques.



(a)

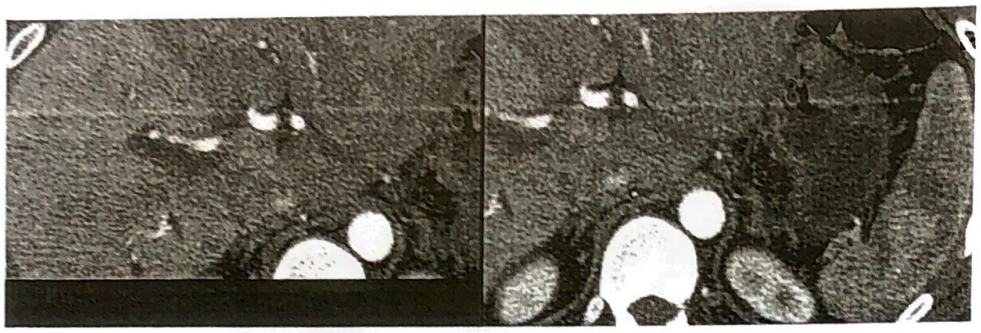


(b)

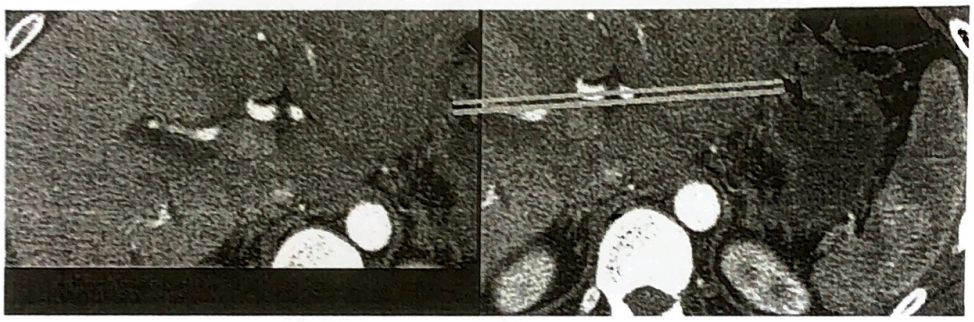


(c)

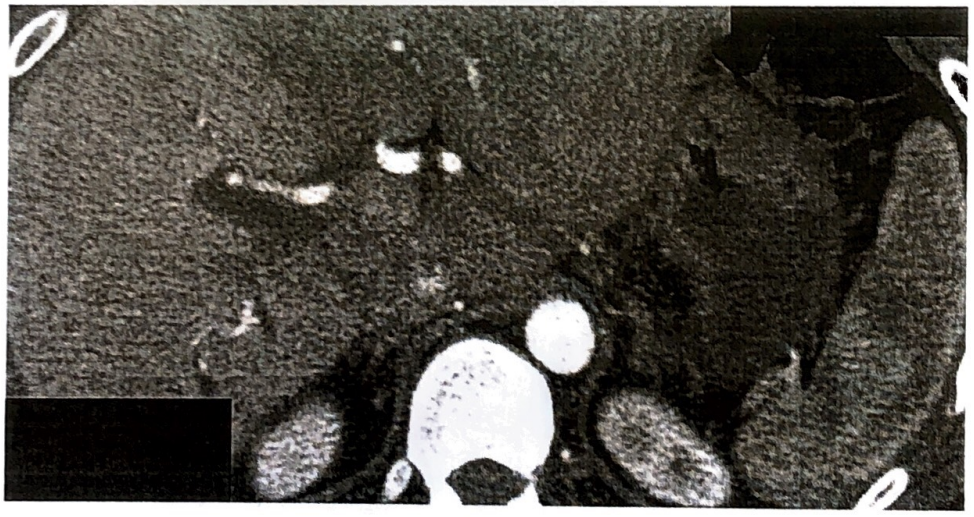
FIGURE 1 – (a) Points caractéristiques. (b) Correspondances. (c) Mosaique.



(a)



(b)



(c)

FIGURE 2 – (a) Points caractéristiques. (b) Correspondances. (c) Mosaique.