

De la photographie numérique à la photographie computationnelle

CES8AH - École des Mines de Nancy

Séance 10

Frédéric Sur

Dans cette séance on s'intéresse à la création automatique de mosaïques d'images (panorama). Vous êtes fortement encouragé à expérimenter en modifiant les différents paramètres et à utiliser vos propres images. N'hésitez pas non plus à consulter la documentation Matlab.

Les deux premières sections suivent le tutoriel Matlab :

<http://fr.mathworks.com/help/vision/examples/find-image-rotation-and-scale-using-automated-feature-matching.html>

1 Extraction de points d'intérêt et description de régions d'intérêt

La page wikipedia expliquant les points d'intérêt et descripteurs SIFT est très bien faite :

http://fr.wikipedia.org/wiki/Scale-invariant_feature_transform

On utilisera en fait les points SURF :

http://fr.wikipedia.org/wiki/Speeded_Up_Robust_Features

Que fait :

```
>> I1 = imread('cameraman.tif');
>> points_I1 = detectSURFFeatures(I1);
>> figure, imshow(I1); hold on;
>> plot(points_I1.selectStrongest(15));
```

Plus d'informations sur l'extraction des zones d'intérêt avec `doc detectSURFFeatures`.

La variable `points_I1` est un *objet* Matlab. Pour en savoir plus :

```
>> points_I1
>> doc SURFPoints
```

Comparer à :

```
>> scale = 0.7;
>> J = imresize(I1, scale);
>> theta = 30;
>> I2 = imrotate(J,theta);
>> points_I2 = detectSURFFeatures(I2);
>> figure, imshow(I2); hold on;
>> plot(points_I2.selectStrongest(15));
```

Essayez différentes échelles et angle de rotation.

Ensuite, on construit des descripteurs des régions circulaires extraites de chacune des images :

```
>> [features_I1, points_I1_b] = extractFeatures(I1, points_I1);
>> [features_I2, points_I2_b] = extractFeatures(I2, points_I2);
```

2 Appariement des points d'intérêt

On apparie les points d'intérêt dont les descripteurs sont les plus proches.

```
>> indexPairs = matchFeatures(features_I1, features_I2);
>> matched_I1 = points_I1_b(indexPairs(:,1));
>> matched_I2 = points_I2_b(indexPairs(:,2));
```

Constatez dans `doc matchFeatures` que la méthode d'appariement est bien celle vue en cours.

On visualise les points appariés par (deux méthodes sont possibles) :

```
>> figure;
>> showMatchedFeatures(I1, I2, matched_I1, matched_I2);
>> title('Points appariés selon la proximité des descripteurs');
>> figure;
>> showMatchedFeatures(I1, I2, matched_I1, matched_I2, 'montage');
>> title('Points appariés selon la proximité des descripteurs');
```

Des « faux » appariements sont visibles. On va les « nettoyer » en imposant qu'une similitude globale envoie les points de la première image sur la seconde.

```
>> [tform, inliers_I1, inliers_I2] = estimateGeometricTransform(...
    matched_I1, matched_I2, 'similarity');
>> figure;
>> showMatchedFeatures(I1, I2, inliers_I1, inliers_I2);
>> title('Points appariés après RANSAC');
>> figure;
>> showMatchedFeatures(I1, I2, inliers_I1, inliers_I2, 'montage');
>> title('Points appariés après RANSAC');
```

3 Panorama par construction d'une mosaïque

Suivez le tutoriel décrit dans la documentation Matlab :

<http://fr.mathworks.com/help/vision/examples/feature-based-panoramic-image-stitching.html>

Vous serez peut-être amené à changer le répertoire où les images `building` sont cherchées.

Modifiez le script du tutoriel et affichez les appariements entre points d'intérêt de deux images consécutives.

Vous testerez d'autres jeux d'images, par exemples celui de la page suivante :

<http://study.marearts.com/2013/11/opencv-stitching-example-stitcher-class.html>

Essayez avec le jeu d'images disponible ici :

<http://home.cse.ust.hk/~cstws/research/641D/mosaic/>

Pourquoi n'est-il pas adapté à la méthode du script Matlab ? Quelle amélioration pouvez-vous proposer ?