

Estimation d'homographies et reconstruction de panoramas

David FILLIAT - Antoine MANZANERA
Cours ROB313 - ENSTA Paris

26 novembre 2019

1 Introduction

Dans ce TP, nous allons **estimer des homographies 2D** permettant de rectifier des images et de construire des panoramas. Pour cela, nous utiliserons le code MATLAB réalisé par **Cordelia Schmid**, Josef Sivic et Andrew Zisserman pour l'école d'été Reconnaissance visuelle et apprentissage automatique¹ de l'INRIA. Ce code a été légèrement modifié, vous le téléchargerez sur la page du cours. Ces programmes permettent d'**estimer des homographies entre images**, **de transformer des images selon ces homographies** et de **construire des panoramas en assemblant ces images transformées**.

Pour ce TP, vous devez rédiger un rapport que vous m'enverrez au format PDF d'ici la séance de cours suivante. Utilisez le format `Nom_Rapport_Homographies.pdf` pour nommer votre rapport.

2 Question 1 - Rectification d'images

Dans cette première partie, nous allons prendre une image **d'un plan déformé** par l'effet de perspective et la redresser (Figure 1).

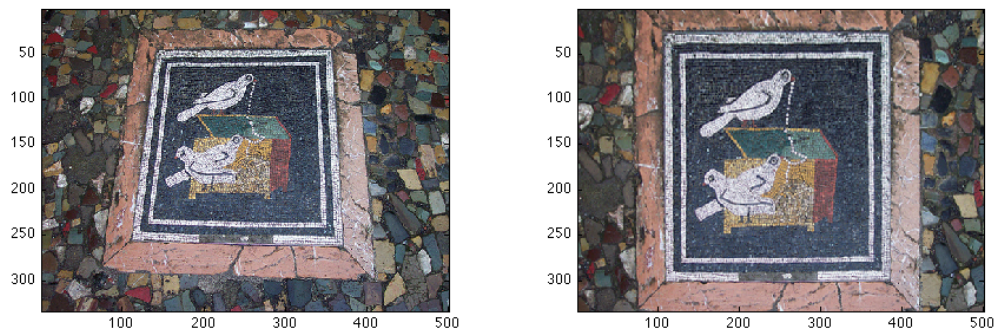


FIGURE 1 – Image d'origine (à gauche) et image redressée de telle sorte que la mosaïque soit carrée (à droite).

Pour commencer, chargez l'image :

```
imrgb = double(imread('Pompei.jpg'))/255;
```

Vous pouvez afficher l'image avec la fonction :

```
imagesc(imrgb);
```

1. <http://www.di.ens.fr/willow/events/cvml2010/>

Choisissez 4 points sur l'image d'origine (les coins de la mosaïque centrale par exemple) et les points en lesquels ils devront être transformés dans l'image destination (vous pouvez choisir les points que vous souhaitez). Mettez ces points sous forme de matrices **PtO** et **PtD** de taille 3×4 (un point en coordonnées homogènes par colonne).

Complétez la fonction `homography2d` pour implémenter l'estimation d'homographie par la méthode **DLT** vue en cours. Vous devez :

- créer la matrice A définissant le système à résoudre, en faisant attention de pouvoir traiter un nombre quelconque (supérieur à 4) de points,
- réaliser la SVD de la matrice $A : A = USV^T$,
- reconstruire H à partir du résultat de la SVD.

Puis utilisez cette fonction pour estimer l'homographie nécessaire pour redresser votre image :

```
H = homography2d(PtO, PtD);
```

Affichez enfin l'image transformée :

```
imagesc(vgg_warp_H(imrgb, H));
```

3 Question 2 - Assemblage d'images

Nous allons maintenant utiliser deux images prises depuis le même point de vue et en transformer une afin de pouvoir les superposer.

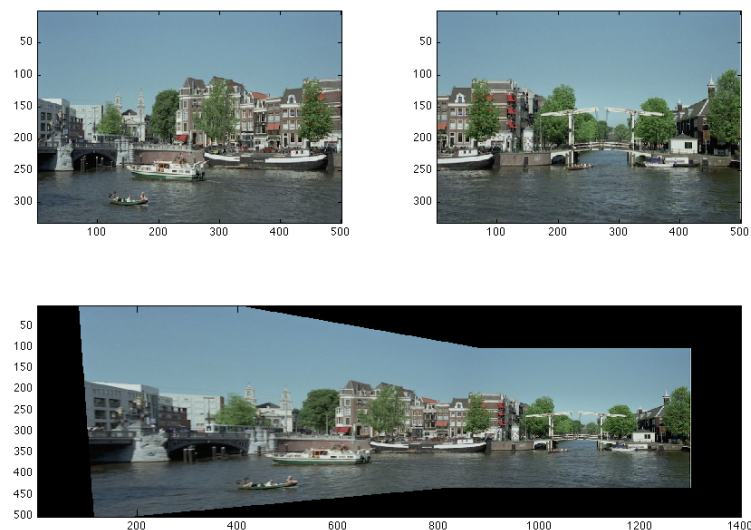


FIGURE 2 – Exemple d'assemblage de panorama.

A vous de trouver la procédure, vous aurez besoin des fonctions suivantes :

- $H = \text{homography2d}(x1, x2)$: fonction réalisée précédemment qui calcule l'homographie transformant des points $x1$ en points $x2$.
- `im_warped=vgg_warp_H(img, H, 'linear', bbox)` : transforme une image à l'aide de l'homographie H , et la place dans une image de taille `bbox = [xmin xmax ymin ymax]`. L'image est complétée par un fond noir.
- `im_fused = max(ima, imb)` : fusionne deux images de même taille en remplaçant le noir de l'une par la couleur de l'autre.

Vous définirez l'homographie à l'aide de points choisis manuellement dans les 2 images. Commencez par 4 points. Essayez ensuite avec un nombre de points plus important. Qu'est-ce que cela apporte ? Vous pouvez ensuite faire l'assemblage avec 3 images.