

Estimation d'une homographie

Florent GUIOTTE et Frédéric BECKER

29 mai 2015

Table des matières

1	Introduction	1
2	Calcul d'homographie	1
2.1	Question 1	1
2.2	Question 2	1
2.3	Question 3	2
2.4	Question 4	2
2.5	Question 5	2
3	Random sample consensus	3
3.1	Question 6	3
3.2	Question 7	4
3.3	Question 8	4
3.4	Question 9	5

1 Introduction

L'objectif de ce tp est d'estimer une homographie, puis de transférer une première image dans une deuxième avec cette homographie.

La deuxième partie se concentre sur la validation des points mis en correspondance de manière automatique, afin de pouvoir faire des mosaïques de manière fiable avec l'homographie.

2 Calcul d'homographie

2.1 Question 1

Pour calculer une homographie il faut au minimum 4 points.

2.2 Question 2

Pour calculer l'homographie, on implémente la fonction DLT qui prend en paramètres deux listes de points de l'image 1 puis de l'image 2, ces listes doivent faire correspondre les points entre les deux images, dans le même ordre. Pour obtenir ces deux listes de points on repère manuellement les point correspondant en cliquant sur les repères visuels des deux images, dans le même ordre (fig. 1). Notre implémentations estime l'homographie avec 5 points.

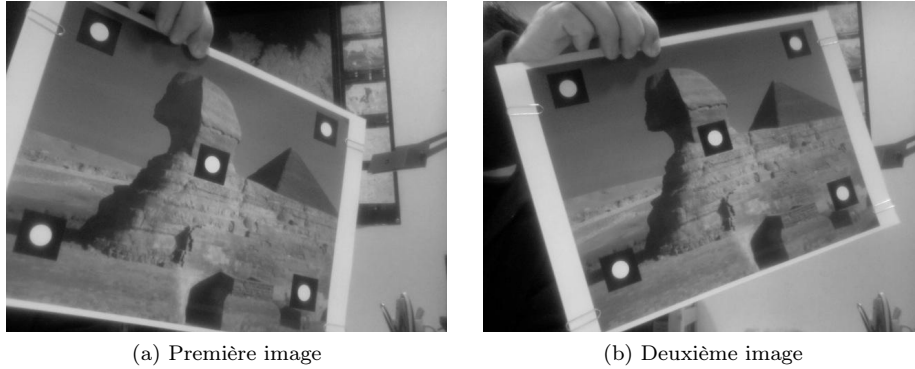


FIGURE 1 – Les deux images comportant les points à mettre en correspondance

2.3 Question 3

L'estimation semble précise. On peut le vérifier avec la question suivante, pour visualiser l'erreur.

2.4 Question 4

On transfère les points $(u2_i, v2_i)$ dans l'image 1 grâce à l'homographie, puis on compare les résultats avec les vrais points $(u1_i, v1_i)$. On calcule ainsi l'erreur commise et on l'affiche par un cercle centré sur le point calculé, de rayon dix fois l'erreur commise (fig. 2).

2.5 Question 5

D'après les questions précédentes, on peut maintenant calculer la correspondance des points d'une image à l'autre. Dans cette question, on transfère tous les points de la deuxième image dans la première en utilisant cette méthode. Le résultat nous donne notre première mosaïque (fig. 3). Pour la visualisation, chacune des deux images à une transparence de 50%, les zones communes apparaissent donc sans aucune transparence.

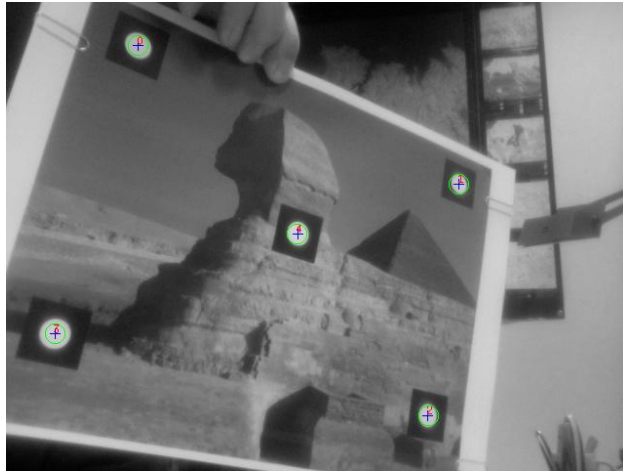


FIGURE 2 – Visualisation de l'erreur de calcul induise par l'homographie

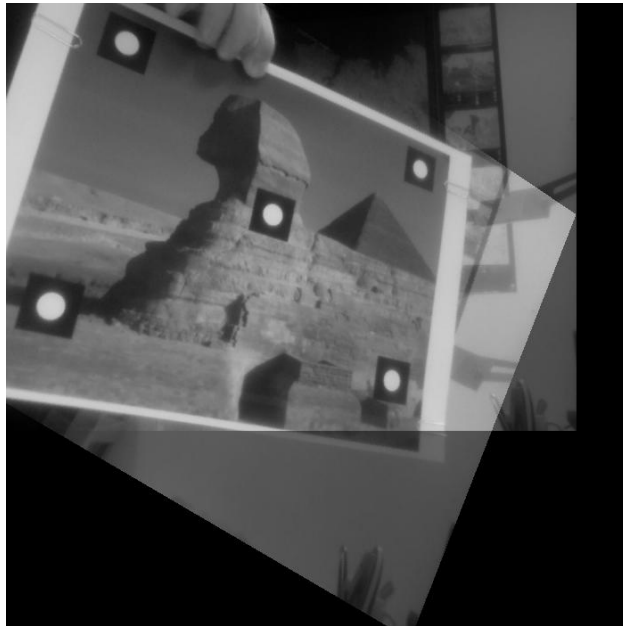


FIGURE 3 – Mosaïque obtenue par le transfère des points de la deuxième image dans la première.

3 Random sample consensus

3.1 Question 6

On affiche les mises en correspondances acquises automatiquement (via OpenCV, ou vpBrief, ou autre ...), figure 4.



FIGURE 4 – Point mis en correspondance de manière automatique

On peut constater quelques erreurs de mise en correspondance, pour les visualiser plus rapidement, on peut tracer des droites entre les correspondances, les outliers apparaissent naturellement à l'oeil (fig. 5).

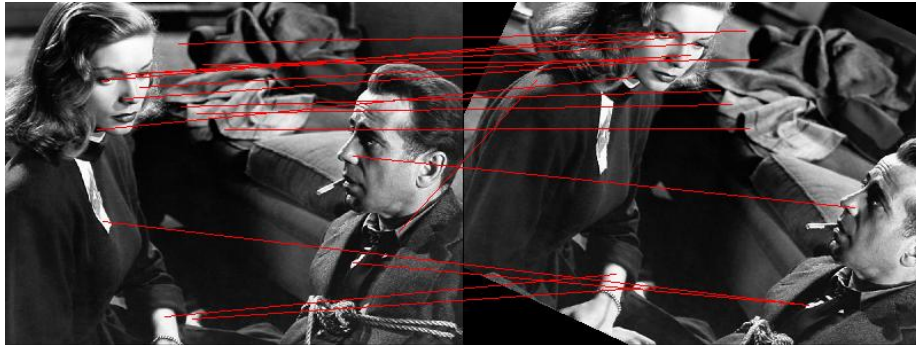


FIGURE 5 – Mise en correspondance graphique, les erreurs apparaissent.

3.2 Question 7

On applique directement l'algorithme développé dans la première partie. Les erreurs ne sont pas raisonnables. Les quelques erreurs de correspondances suffisent pour fausser le calcul de l'homographie.

3.3 Question 8

Pour pallier à ce problème, il faut impérativement détecter les erreurs de matching et les éliminer. On utilise un algorithme basé sur RANSAC (pour RANdom SAmple Consensus) permet de détecter rapidement les outliers des inliers dans notre mise en correspondance.

Le résultat de notre implémentation de RANSAC est visible figure 6. Les inliners sont tracés en vert tandis que les outliers sont tracés en rouge.

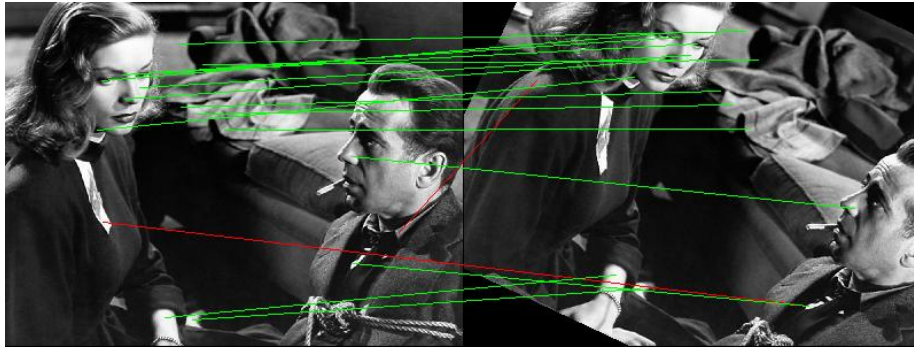


FIGURE 6 – Mise en évidence des inliners et outliers grâce à RANSAC.

3.4 Question 9

La contrainte à respecter pour faire une mosaïque avec cette technique est liée à l'homographie. En effet, l'homographie permet de trouver la transformation d'un plan par rapport à un autre (rotation + translation). Notre «image» à reconstituer doit donc représenter un plan.