

TS 225 – Projet Image

Rectification d'image par homographie
Application à la construction d'une mosaïque d'images

1) Préambule

1.1) Objectifs

Les applications concernant des images ou de la vidéo sont extrêmement nombreuses. De manière générale, elles sont également relativement complexes à mettre en œuvre. Le projet a pour but d'appliquer les enseignements fondamentaux de traitement des images vus au cours du semestre dans un contexte applicatif réel et non trivial.

1.2) Mode d'évaluation

Le projet sera effectué en binôme (en cas d'imparité du nombre total d'étudiants, un trinôme sera formé) pendant 13h20 et sera mené sous *Python*.

L'évaluation du travail comporte deux volets : un travail continu et un rapport d'environ 20 pages.

Le rapport doit être rendu dans un format pdf. Il est important de faire figurer les différentes images de la chaîne complète de traitement. Les images accompagnant les commentaires des résultats doivent notamment être choisies et présentées de sorte à mettre en évidence les phénomènes mentionnés (prendre garde au choix de la palette et des échelles de couleurs, aux grandissements, aux graphiques en surimpression, aux légendes, etc.). Les équations doivent être générées avec un éditeur d'équations, en utilisant une police de caractères homogène avec le reste du document, et être numérotées afin d'être référencées dans le texte. Les commentaires doivent être pertinents et tout résultat doit être justifié. Les programmes seront mis en annexe en double colonne en utilisant une police de caractères relativement petite tout en restant suffisamment lisible. Le rapport devra également comporter un bilan de l'organisation et du déroulement du projet qui comprendra, en particulier, la liste des tâches entreprises avec une évaluation du temps passé par chacun des membres du groupe lors de chaque séance et, le cas échéant, entre les séances.

La qualité du code produit et une bonne organisation seront des points pris en compte pour la note de travail continu.

Une interface *Python* peut également être proposée et sera comptabilisée en tant que bonus.

1.3) Contact

Le projet est encadré par M. Donias (Bureau S222, mél : marc.donias@enseirb-matmeca.fr).

2) Recalage d'images et construction d'une mosaïque

2.1) Contexte

Une mosaïque d'images résulte de la combinaison de plusieurs images « recouvrantes » (voir figure 1) représentant chacune partiellement une même scène observée sous des angles de vues différents.

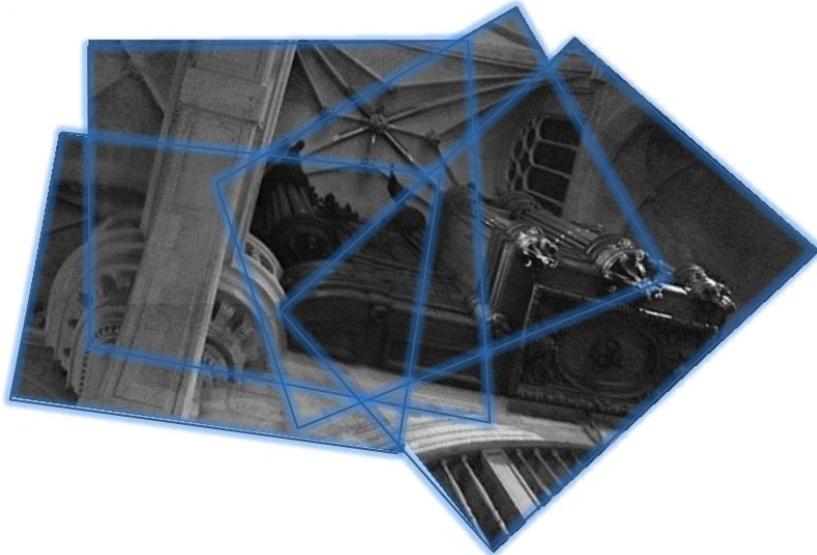


Figure 1 – Exemple de mosaïque d'images.

Une telle représentation, plus compacte que l'ensemble des images originales tout en conservant la majeure partie de l'information portée par celles-ci, a pour application principale de produire une vue unifiée, agrandie ou panoramique. Par exemple, en télédétection, plusieurs vues satellites ou aériennes sont recalées et fusionnées pour former une carte couvrant une région. En pratique, une mosaïque d'image trouve également sa place dans d'autres contextes :

- en compression, le stockage et la transmission peuvent être réalisés à travers une mosaïque d'images couplée à des images de différences et des paramètres indiquant la posture de chaque image originale.
- en indexation multimédia, une courte séquence vidéo est efficacement résumée par une image unique.

A partir d'un ensemble d'images, la construction d'une mosaïque nécessite, dans un premier temps, d'estimer les paramètres de transformations géométriques liant les différentes prises de vues à un espace commun et, dans un second temps, de recalier ou de rectifier chacune des images dans cet espace. Les problématiques concernent la détermination d'un espace commun adéquat, la nature des transformations géométriques et l'estimation de leurs paramètres, les techniques de rectification et d'interpolation, ou encore la stratégie de combinaison ou de fusion des parties communes ou redondantes.

2.2) Algorithmique

La méthode développée dans ce projet vise à produire une mosaïque d'images en tenant compte des hypothèses ou contraintes suivantes :

- les images traitées sont reliées 2 à 2 par une transformation géométrique appelée « homographie » (restriction à des scènes « lointaines », ne présentant pas d'occultations ou de distorsion radiale, etc.),

- des ensembles de points composées au minimum de quatre points (ancres) sont connus dans chacune des images, toute ancre d'une image ayant au moins une ancre correspondante dans au moins une autre image de sorte qu'un graphe d'ancres relie l'ensemble des images traitées (démarche supervisée nécessitant une connaissance a priori),
- un temps de calcul relativement faible par image à traiter (calculs matriciels requis sous *Matlab*).

2.2.1) Homographie

Dans une image numérique acquise par des dispositifs classiques (appareil photographique « plan », caméra sans distorsions radiales, etc.), les objets rectangulaires tels qu'une feuille de papier, un livre, une affiche, une fenêtre ou une porte apparaissent le plus souvent simplement quadrangulaires : leurs angles projetés peuvent ne pas être droits et leurs côtés opposés deux à deux n'apparaissent pas nécessairement parallèles. Les transformations affines (translation, rotation, homothétie, symétrie centrale ou axiale) ne sont alors pas adaptées pour rectifier de telles images c'est-à-dire restaurer les géométries photographiées de sorte à les faire apparaître rectangulaires. D'une manière générale, elles ne permettent également pas de rectifier des prises de vues partielles et recouvrantes d'une même scène (mosaïque d'images) afin de réconcilier leurs contenus et former une vue unique englobante.

Le terme « homographie » désigne une application projective du plan définie par un quotient de deux fonctions affines qui permet de transformer un quadrangle quelconque en un autre quadrangle quelconque. Cette transformation convient pour la rectification qui nous intéresse ici car elle inclut l'aspect projectif d'une prise de vue induit par un appareil photographique « plan » et, en particulier, inclut les transformations affines. Une homographie est définie par une matrice H à 9 coefficients :

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \quad (1)$$

La relation qui lie tout point « source » $M_1 = (x_1, y_1)$ à un point « destination » $M_2 = (x_2, y_2)$ à travers une homographie H s'écrit:

$$\begin{cases} x_2 = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \\ y_2 = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \end{cases} \quad (2)$$

De manière évidente, une homographie est définie à un coefficient multiplicateur près et le paramètre h_{33} peut toujours être fixé à 1. Par conséquent, les 8 paramètres libres d'une homographie peuvent être déterminés de manière exacte, c'est-à-dire unique et non ambiguë, dès lors que 4 paires de points appariés (M_1, M_2), correspondants réciproques au sens de la transformation géométrique, sont connus. Leur détermination s'obtient en résolvant le système linéaire suivant :

$$AX = B \quad (3)$$

où $X = [h_{11} \dots h_{32}]^T$ est un vecteur formé des 8 inconnues de H , A et B sont réciproquement une matrice de taille 8×8 et un vecteur de longueur 8 constitués de combinaisons des points (x_1, y_1) et (x_2, y_2) , issues d'une réécriture de l'équation (2) pour chacun des 4 points appariés.

Si moins de points appariés sont connus, la matrice H admet une infinité de valeurs possibles tandis que si davantage de points appariés sont connus, les paramètres de H sont déterminés aux sens des moindres carrés selon :

$$\hat{X} = \underset{X}{\operatorname{argmin}} \|AX - B\|^2. \quad (4)$$

D'une manière générale, lorsque la solution est considérée unique, la matrice H s'obtient par le calcul d'une pseudo-inverse :

$$\hat{X} = (A^T A)^{-1} A^T B. \quad (5)$$

2.2.2) Principes généraux de la construction d'une mosaïque

Les paramètres de l'homographie liant deux images étant estimés, il est possible de les rectifier c'est-à-dire de les placer dans un référentiel commun et de les fusionner afin de construire une image « étendue ». La formation d'une mosaïque d'images consiste alors à itérer ce processus autant de fois que nécessaire en fonction du nombre d'images disponibles. Plusieurs stratégies sont possibles selon l'ordre de traitement des images (incrémental ou simultané) ou la technique choisie pour fusionner les parties communes.

Une image numérique de taille $W \times H$ est une matrice de taille $H \times W$ représentant une partie bornée, définie par une « boite englobante » $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$, et échantillonnée d'une fonction continue bidimensionnelle. Le cas le plus basique consiste à considérer un pas d'échantillonnage égal à 1 et des coordonnées x_{\min} , x_{\max} , y_{\min} , y_{\max} entières telles que $W = x_{\max} + 1 - x_{\min}$ et $H = y_{\max} + 1 - y_{\min}$. De manière évidente, toute transformation géométrique non basique (rotations, changements d'échelle, homothéties, homographies, etc.) conduit à une nouvelle image numérique dont un nombre très variable de points d'échantillonnage ne sont pas définis (voir figures 2a et 2b) et inscrite dans une boite englobante plus grande.

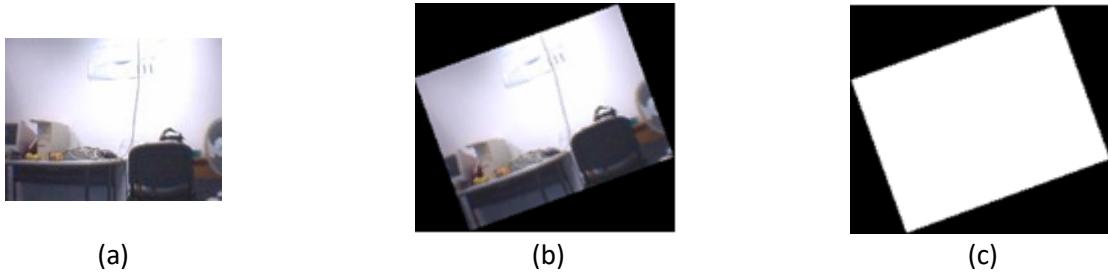


Figure 2 – (a) Image initiale, (b) Image obtenue par rotation (points non définis en noir), (c) Masque binaire définissant le domaine de validité

L'utilisation conjointe d'un domaine de validité (voir figure 2c) sous la forme d'un masque noté m , par exemple binaire (par convention, une valeur de masque de 1 indiquera un point valide et pourra être représenté en blanc tandis qu'une valeur de masque égale à 0 indiquera un point non défini et pourra être représenté en noir), facilite la gestion des images numériques lors de transformations géométriques, de combinaisons logiques ou arithmétiques.

Ainsi, l'application de la transformation géométrique T à l'image f_1 qui conduit à l'image $f_2 = T(f_1)$ nécessite, tout d'abord, d'établir la boite englobante de l'image transformée f_2 et, ensuite, de former les matrices « image » et « masque » en échantillonnant la boite englobante, en calculant les points initiaux associés par transformation géométrique inverse T^{-1} et en déterminant leurs valeurs par interpolation (plus proche voisin, bilinéaire, cubique, etc.) : par exemple, un point de l'image « masque » est mis à 1 seulement si le point

initial associé est dans le domaine de validité de l'image f_1 et vaut 1 (0 dans le cas contraire) et la valeur de la matrice « image » est déterminée uniquement dans ce cas (valeur arbitraire 0 dans le cas contraire).

Par ailleurs, la combinaison f_3 définie par la moyenne de deux images f_1 et f_2 s'écrit simplement à l'aide de leurs masques binaires respectifs m_1 et m_2 :

$$f_3(x, y) = \begin{cases} \frac{f_1(x, y)m_1(x, y) + f_2(x, y)m_2(x, y)}{m_1(x, y) + m_2(x, y)} & \text{si } m_1(x, y) + m_2(x, y) \neq 0 \\ \text{non défini} & \text{si } m_1(x, y) + m_2(x, y) = 0 \end{cases} \quad (6)$$

tandis que le masque m_3 associé à f_3 s'exprime par :

$$m_3(x, y) = \max(m_1(x, y), m_2(x, y)) \quad (7)$$

Avant toute combinaison, les matrices masques et images des différentes images doivent être complétées de sorte à être définies sur des boîtes englobantes identiques d'un même référentiel.

Il est à noter qu'une image originale de taille $W \times H$ peut être initialement représentée par une boîte englobante $[1, W] \times [1, H]$, une matrice f codant l'intensité de chaque point et un masque m « plein » correspondant à une matrice uniquement composée de valeurs égales à 1.

3) Travail à réaliser

Il s'agit d'implémenter la technique proposée sous *Python* et de justifier de son fonctionnement correct à travers, par exemple, les étapes des sections suivantes.

3.1) Estimation des paramètres d'une homographie

Ecrire une fonction (primitive « identification ») qui permet d'estimer les paramètres d'une homographie reliant deux ensembles « source » et « destination » composés chacun de 4 points ordonnés :

- prototype `H = homography_estimate(x1, y1, x2, y2)`,
- `x1` et `y1` sont les vecteurs regroupant respectivement les 4 coordonnées horizontales et verticales de points « sources » ordonnés,
- `x2` et `y2` sont les vecteurs regroupant respectivement les 4 coordonnées horizontales et verticales de points « destination » ordonnés en conformité avec les points « source »,
- `H` est une matrice 3×3 regroupant les 9 paramètres de l'homographie estimée.

Ecrire une fonction (primitive « transformation ») qui applique une homographie à un ensemble de points :

- prototype `(x2, y2) = homography_apply(H, x1, y1)`,
- `H` est une matrice 3×3 regroupant les 9 paramètres de l'homographie appliquée,
- `x1` et `y1` sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales de points « sources »,

- **x2** et **y2** sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales de points « destination » calculés en appliquant l'homographie aux points « source ».

La validation simultanée des deux fonctions s'obtient en choisissant deux ensembles « source » et « destination » de 4 points quelconques, en estimant l'homographie les reliant, en appliquant l'homographie obtenue aux points de l'ensemble « source » et en vérifiant que les points transformés sont bien identiques à ceux de l'ensemble « destination »

3.2) Extraction

Ecrire une fonction (primitive « extraction ») qui extrait un contenu quadrangulaire d'une image et le transforme en une image rectangulaire de dimensions arbitraires et choisies :

- prototype **I2 = homography_extraction(I1, x, y, w, h)**,
- **I1** est l'image « source » représentant un contenu quadrangulaire,
- **x** et **y** sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales des 4 coins du contenu quadrangulaire présent dans **I1**,
- **w** et **h** sont respectivement la largeur et la hauteur du contenu rectangulaire créé,
- **I2** est l'image « destination » rectangulaire créée.

Une étape interne de calcul est l'identification de l'homographie (primitive « identification ») reliant les 4 coins ordonnés de l'image « destination » aux 4 points « source » **x** et **y** dans le sens adéquat « image destination -> image source ».

La validation s'obtient selon les étapes suivantes :

- choix et affichage d'une image « source » représentant un contenu quadrangulaire,
 - pointé à la souris et récupération des 4 points « source » correspondant aux coins ordonnés du contenu quadrangulaire,
 - choix des dimensions de l'image « destination » rectangulaire à créer,
- calcul et affichage de l'extraction (primitive « extraction »).

3.3) Projection

Ecrire une fonction (primitive « projection ») qui remplace un contenu quadrangulaire d'une image par une image rectangulaire quelconque :

- prototype **I2 = homography_projection(I1, I2, x, y)**,
- **I1** est l'image « source » rectangulaire à projeter,
- **I2** est l'image « destination » représentant un contenu quadrangulaire,
- **x** et **y** sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales des 4 coins du contenu quadrangulaire présent dans **I2**.

Une étape interne de calcul est l'identification de l'homographie (primitive « identification ») reliant les 4 coins ordonnés de l'image « source » aux 4 points **x** et **y** dans le sens adéquat « image source -> image destination ».

La validation s'obtient selon les étapes suivantes :

- choix et affichage d'une image « source »,
- choix et affichage d'une image « destination » représentant un contenu quadrangulaire,
- pointé à la souris et récupération des 4 points « destination » correspondant aux coins ordonnés du contenu quadrangulaire,
- calcul et affichage de la projection (primitive « projection »).

3.4) Projections croisées

Ecrire une fonction (primitive « projections croisées ») qui échange les contenus quadrangulaires d'une image :

- prototype **I = homography_cross_projection(I, x1, y1, x2, y2)**,
- **I** est l'image contenant les contenus quadrangulaires,
- **x1** et **y1** sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales des 4 coins du premier contenu quadrangulaire présent dans **I**,
- **x2** et **y2** sont les vecteurs regroupant respectivement les coordonnées horizontales et verticales des 4 coins du second contenu quadrangulaire présent dans **I**.

Deux approches différentes seront mises en œuvre au moyen :

- soit d'une combinaison des primitives « extraction » et « projection » permettant de réaliser l'opération en deux passes,
- soit d'un carré virtuel intermédiaire permettant de réaliser l'opération en une seule passe avec une qualité supérieure à l'approche précédente.

La validation s'obtient selon les étapes suivantes :

- choix et affichage d'une image représentant deux contenus quadrangulaires,
- pointé à la souris et récupération de deux ensembles de 4 points correspondant aux coins ordonnés des contenus quadrangulaires,
- calcul et affichage de l'extraction (primitive « extraction »).

3.5) Mosaïque constituée de deux images

A partir d'une image initiale de taille suffisamment importante, extraire deux imagettes « quadrangulaires » recouvrantes par la technique vue précédemment (mode « extraction »).

Au sein de chacune des imagettes obtenues, identifier deux ensembles de 4 points correspondants aux mêmes sites de la scène observée.

Calculer les paramètres de l'homographie liant les deux quadrangles dans les deux sens possibles.

Rectifier la première imagette de sorte à la projeter dans la deuxième imagette, avec et sans élargissement, avec et sans combinaison des parties communes.

Faire de même en inversant le rôle des deux imagettes.

Rectifier les deux imagettes au sein d'un espace rectangulaire commun « élargi » englobant l'espace des deux imagettes. Vérifiez que la fusion obtenue est indépendante de l'ordre des deux images traitées

3.6) Mosaïque constituée de plusieurs images

Etendre le procédé précédent au cas de plusieurs imagettes : des ensembles de 4 points (ancres) dans chacune des imagettes traitées se correspondant 2 à 2 seront supposés connus et des stratégies séquentielle ou parallèle seront mises en œuvre. Bruiter les imagettes traitées de sorte à comparer les différentes implémentations.

Vérifier que l'ordre de traitement des images n'influe pas sur le résultat. Le cas échéant, proposer une stratégie conduisant au résultat attendu.

Appliquer la démarche à des images réelles en prenant garde à la mise en concordance de dynamiques d'intensité éventuellement différentes.

3.7) Améliorations

Proposez une méthode pour éviter l'apparition de discontinuités à l'intérieur de la mosaïque (dans le cas, par exemple, où les bords des images d'origine sont plus sombres ou clairs que leurs parties centrales). Identifiez d'autres raisons justifiant cette amélioration.

Bonus : rendez l'approche entièrement automatique en calculant des points remarquables (algorithmes SIFT, SURF, ORB, BRIEF, etc.) et en identifiant des paires de points correspondants par une approche robuste (RANSAC).

Bonus : implémentez une interface graphique mettant en œuvre l'ensemble des traitements.

3.8) Contexte

Faites une recherche de logiciels utilisant la technique des mosaïques d'images en présentant l'application visée (contexte, intérêt, etc.). Précisez notamment leurs différences avec l'implémentation réalisée.