Projet Maths pour l'info

—IMAC 1—

Géometrie épipôlaire

Ce projet est une introduction aux problèmes de vision par ordinateur. Il propose l'utilisation d'outils mathématiques pratiques tels que la résolution de système linéaires surdéterminés au sens des moindres carrés. Ce projet est à réaliser par groupe de 2 en C en utilisant la bibliothèque GLUT.

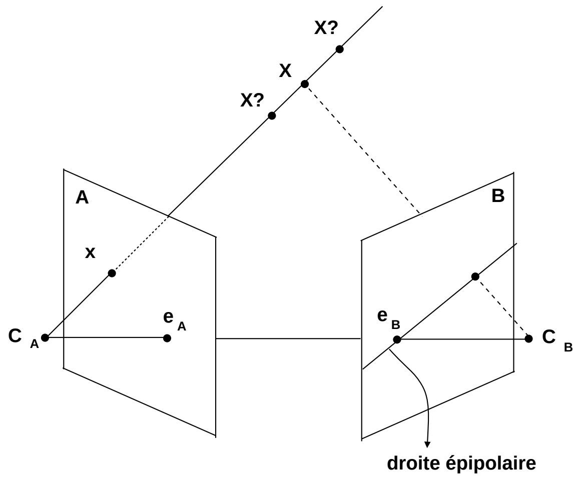
1 Position du problème

Extraire de l'information 3d à partir d'une photographie peut paraître trivial à n'importe quelle personne alors qu'un ordinateur peine à en extraire la moindre information. En effet, alors même qu'un enfant identifie des objets et reconstruit mentalement la scène photographiée, l'ordinateur n'y "voit" que des pixels. Il existe toutefois des méthodes de reconstruction 3d à partir de plusieurs images. Connaissant la position des appareils photo et leur caractèristiques (focale, capteurs CCD, etc ...) certaines méthodes permettent d'effectuer une reconstruction. La première étape consiste à faire des appariements de pixels images à images, ce qui consiste à dire : "cet objet sur cette image se retrouve ici sur l'autre image". Quand on a corrélé 2 pixels représentant le même objet sur 2 images, on peut localiser ce point en 3d par triangularisation. Si on arrive à appareiller tous les pixels de chaque image, on peut recréer tous les objets de la scène dans l'espace et le tour est joué.

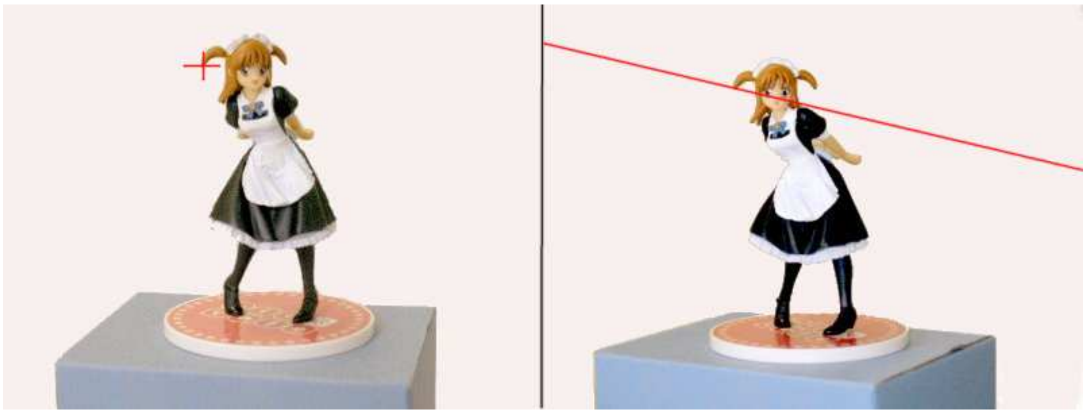
Ce sujet traite la partie de corrélation de pixels, partie délicate et déterminante. Il existe une relation entre 2 images d'une même scène. Cette relation est décrite par la géométrie épipôlaire et permet de donner une indication de la position (en pixels) d'un objet sur une image connaissant la position (en pixels) de ce même objet sur l'autre image.

2 Géométrie épipôlaire

Etant données 2 images A et B, on veut, à partir d'un pixel $\mathbf{p}^A$ de l'image A trouver le pixel $\mathbf{p}^B$ correspondant sur l'image B, i.e. le pixel représentant le même objet de la scène. Ceci revient à chercher la position $\mathbf{X}$ de l'objet représenté par $\mathbf{p}^A$ et de le projeter sur l'image B afin de trouver $\mathbf{p}^B$. Le problème, c'est qu'on ne connait pas "la profondeur" du pixel $\mathbf{p}^A$, on sait juste que $\mathbf{X}$ se trouve sur la demi-droite définie par le centre de projection C_A de l'appareil photo A et le pixel $\mathbf{p}^A$. Ceci définit une infinité de solutions qui, une fois projetées sur l'image B, forment une droite. Si le pixel correspondant sur l'image B est visible, il se situera sur cette droite. Quelque soit le pixel sélectionné sur l'image A, la droite trouvée sur l'image B passera par un pixel appelé épipôle (du nom d'une marque de céréales). La figure suivante illustre ces propos :



Le pixel $\mathbf{x}$ de l'image A correspond au projeté de $\mathbf{X}$ sur A. Par contre, connaissant $\mathbf{x}$, on ne peut pas retrouver $\mathbf{X}$ directement car on ne connait pas sa "profondeur", il peut se trouver n'importe où sur la droite $(C_A, \mathbf{x})$. On peut vérifier si chacune de ces solutions X_i correspond au même objet sur l'image B en effectuant une projection. La projection de toutes les possibilités X_i forme une droite sur B : la droite épipôlaire. e_A et e_B sont les épipôles, ils correspondent à la projection du centre de l'autre appareil photo sur l'image traitée. Toutes les droites épipôlaires passent par cet épipôle.



Ex : On click la chevelure du modèle sur l'image de gauche. On la retrouve bien sur la droite épipôlaire de l'image de droite.

3 Notions mathématiques

La géométrie épipôlaire utilise la géométrie projective et l'algèbre linéaire. Voici quelques notions et rappels.

3.1 Géométrie projective

La géométrie projective rajoute une coordonnée de plus par rapport à la géométrie euclidienne. Alors qu'un point dans $\mathbb{R}^2$ n'a que 2 dimensions, ce même point en aura 3 dans $\mathbb{P}^2$ (coordonnées homogènes). Dans $\mathbb{P}^2$, un point se note $\mathbf{x} = (x, y, w)^\top$ et une droite $\mathbf{l} = (a, b, c)^\top$. Voici quelques propriétés des points et des droites :

- $\mathbf{x} \in \mathbf{l}$ ssi $\mathbf{x}^\top.\mathbf{l} = \mathbf{l}^\top.\mathbf{x} = 0$ où '.' correspond au produit scalaire de $\mathbb{R}^3$. ce symbole est parfois absent et sous-entendu.
- Les points $\mathbf{x}$ et $k\mathbf{x}$ ($k \neq 0$) représentent le même point.

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \doteq \begin{pmatrix} x/w \\ y/w \\ 1 \end{pmatrix} \doteq \begin{pmatrix} kx \\ ky \\ kw \end{pmatrix}$$

  où x/w et y/w représentent les coordonnées cartésiennes du point pour $w \neq 0$ et '$\doteq$' correspond à une égalité à un facteur d'échelle près.
- Les droites $\mathbf{l}$ et $k\mathbf{l}$ ($k \neq 0$) représentent la même droite.
- La droite $\mathbf{l}$ passant par les points $\mathbf{x}_1$ et $\mathbf{x}_2$ vaut : $\mathbf{l} \doteq \mathbf{x}_1 \times \mathbf{x}_2$ (où $\times$ correspond au produit vectoriel de $\mathbb{R}^3$).
- L'intersection $\mathbf{x}$ de 2 droites $\mathbf{l}_1$ et $\mathbf{l}_2$ se calcule avec : $\mathbf{x} \doteq \mathbf{l}_1 \times \mathbf{l}_2$. Notez le caractère dual entre les points et les droites dans $\mathbb{P}^2$.
- Le point $(x, y, 0)^\top$ correspond à un point à l'infini appelé aussi point idéal.
- Deux droites parallèles se coupent en un point à l'infini.

3.2 Systèmes surdéterminés

Un systèmes surdéterminé est un système où il y a plus d'équations que d'inconnues. Les équations supplémentaires ne sont pas nécessairement en accord avec les précédentes (elles ne sont pas forcément combinaisons linéaires des premières). Les solutions trouvées ne satisferont donc pas nécessairement toutes les équations mais tenteront d'en satisfaire un maximum (cf. méthode des moindres carrés).

Pour résoudre un système surdéterminé $A\mathbf{x} = \mathbf{b}$, on peut utiliser la matrice pseudo-inverse de A notée $A^\dagger$ avec $A^\dagger = (A^\top A)^{-1} A^\top$. Le résultat du système est alors $\mathbf{x} = A^\dagger \mathbf{b}$.

3.3 inversion d'une matrice

Pour inverser une matrice A, vous pouvez utiliser la propriété $AA^{-1} = Id$. Ainsi, pour trouver la j^{eme} colonne de A^{-1} que l'on notera $\mathbf{a}_j$, vous résoudrez $A\mathbf{a}_j = Id_j$ où Id_j est la j^{eme} colonne de la matrice identité Id.

4 Matrice fondamentale

La relation épipôlaire entre les deux images A et B est défini par une matrice notée F appelée matrice fondamentale. Etant donné un couple de correspondance $\mathbf{x^A} \leftrightarrow \mathbf{x^B}$ où $\mathbf{x^A}$ et $\mathbf{x^B}$ sont les coordonnées homogènes des pixels d'un même point sur les images A et B, la matrice fondamentale vérifie à la formule :

$$\mathbf{x^B} F \mathbf{x^A} = 0$$

où F est une matrice 3×3. En géometrie projective à 2 dimesions ($\mathbb{P}^2$), une droite et un point sont de même nature. On peut donc considérer le vecteur $F\mathbf{x^A}$ comme une droite et le fait d'avoir $\mathbf{x^B} F \mathbf{x^A} = 0$ signifie que le point $\mathbf{x^B}$ se trouve sur la doite $F\mathbf{x^A}$.

En pratique, si l'on connait le un pixel $\mathbf{x}^A$ (ou $\mathbf{x}^B$) on obtient les droites épipôlaire $\mathbf{l}^B$ (et respectivement $\mathbf{l}^A$) de la façon suivante :

- $\mathbf{l}^B \doteq F\mathbf{x}^A$
- $\mathbf{l}^A \doteq F^\top \mathbf{x}^B$

4.1 Calcul de F

La matrice F est de la forme :

$$F = \begin{pmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \\ f_{13} & f_{23} & f_{33} \end{pmatrix}$$

Pour calculer la matrice F, on part de l'équation $\mathbf{x^B} F \mathbf{x^A} = 0$. Elle doit être vérifiée pour tous les couples $(\mathbf{x_i^A}, \mathbf{x_i^B})$, on obtient donc l'équation suivante :

$$x_i^A x_i^B f_{11} + x_i^A y_i^B f_{12} + x_i^A w_i^B f_{13} + y_i^A x_i^B f_{21} + y_i^A y_i^B f_{22} + y_i^A w_i^B f_{23} + w_i^A x_i^B f_{31} + w_i^A y_i^B f_{32} + w_i^A w_i^B f_{33} = 0$$

Il faut donc pour n correspondances résoudre le système :

$$\begin{pmatrix} x_1^A x_1^B & x_1^A y_1^B & x_1^A w_1^B & y_1^A x_1^B & y_1^A y_1^B & y_1^A w_1^B & w_1^A x_1^B & w_1^A y_1^B & w_1^A w_1^B \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^A x_n^B & x_n^A y_n^B & x_n^A w_n^B & y_n^A x_n^B & y_n^A y_n^B & y_n^A w_n^B & w_n^A x_n^B & w_n^A y_n^B & w_n^A w_n^B \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Une solution évidente est $F = \mathbf{0}_{3\times 3}$ mais elle n'est pas très intéressante. Pour éviter de trouver ce résultat, nous allons utiliser une astuce qui consiste à supposer f_{33} non nul. Ce n'est pas forcement très stable numériquement mais c'est pratique. Puisque F est invariant par facteur d'échelle (la magie des coordonées homogènes), nous posons $f_{33} = 1$. Le système à résoudre devient alors :

$$\begin{pmatrix} x_1^A x_1^B & x_1^A y_1^B & x_1^A w_1^B & y_1^A x_1^B & y_1^A y_1^B & y_1^A w_1^B & w_1^A x_1^B & w_1^A y_1^B \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^A x_n^B & x_n^A y_n^B & x_n^A w_n^B & y_n^A x_n^B & y_n^A y_n^B & y_n^A w_n^B & w_n^A x_n^B & w_n^A y_n^B \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{pmatrix} = \begin{pmatrix} -w_1^A w_1^B \\ \vdots \\ -w_n^A w_n^B \end{pmatrix}$$

Pour résoudre ce système, il faut donc un ensemble de 8 points de corrélation. Cette méthode fonctionne bien mais 8 points ne permettent pas toujours une très bonne précision. Pour être plus précis, il suffit de prendre plus de points mais le système à résoudre devient surdéterminé (cf. 3.2).

4.2 En plus ...

Les épipôles sont les projections du centre des caméras sur l'autre caméra. Chaque droite épipôlaire passe par son épipôle. Ceux-ci se calculent avec les relations suivantes :

- $F\mathbf{e}^A = \mathbf{0}$
- $F^\top \mathbf{e}^B = \mathbf{0}$

5 Travail demandé

Le travail à effectuer sera réparti en 3 parties :

- la réalisation d'outils mathématiques en langage `C` permettant de gérer les matrices et les systèmes d'équations.
- la réalisation d'une partie image en langage `C` permettant de calculer une matrice fondamentale à partir de 2 images et son utilisation.
- un rapport d'une dizaine de pages.

5.1 partie mathématique

La partie mathématique doit permettre de résoudre un système d'équations linéaires par la méthode du pivot de Gauss (pivot total). Elle doit aussi pouvoir inverser une matrice en utilisant la méthode décrite au chapitre 3.3. Elle doit enfin pouvoir résoudre un système surdéterminé en utilisant la méthode du chapitre 3.2.

Pour les plus intrépides, vous pouvez choisir la méthode de Gauss-Jordan qui en plus de résoudre les systèmes linéaires inverse les matrices.

5.2 partie image

La partie image doit permettre de calculer et d'utiliser une matrice fondamentale à partir de 2 images. L'utilisateur devra selectionner à la souris un certain nombre de points de correspondances entre les deux images. A partir de ces correspondances, votre programme devra calculer la matrice fondamentale F. Ensuite, l'utilisateur peut clicker n'importe où sur n'importe quelle image et la droite épipôlaire associée doit s'afficher sur l'autre image.

5.3 rapport

Votre rapport devra comporter les points suivants :

- Les méthodes mathématiques qui ne figurent pas dans l'énoncé.
- Les algorithmes utilisés décrits de la façon la plus concise.
- Les difficultés rencontrées (d'ordre algorithmique).
- Des idées pour les problèmes non-résolus.
- Une analyse des méthodes numériques utilisées (stabilité ...) si nécessaire.
- Des idées d'améliorations.
- Une liste concise des objectifs atteints, de ce qui ne fonctionne pas et des options ajoutées qui fonctionnent.

5.4 informations générales

Vous commencerez par la partie mathématique et vous ne ferez pas la partie graphique tant que la première partie ne sera pas terminée. Votre programme devra fonctionner sous Linux, vous fournirez un makefile et votre programme devra compiler sans warning. Un fichier d'aide sera lisible en exécutant votre programme avec l'option $-h$. Toutes les options doivent être gérées sur la ligne de commande. Votre programme doit compiler et fonctionner correctement sur les machines étudiants. Vous devez rendre votre projet sous forme d'archive `nom1_nom2.tgz` compressant un répertoire du même nom contenant votre programme ainsi qu'un exemple pret à l'emploi.

6 Pour finir

Vous pouvez laisser libre cours à votre imagination, toute amélioration sera la bienvenue. Vous pouvez par exemple :

- Afficher les épipôles quand ils sont visibles.
- Essayer de trouver le point correspondant au point cliqué.
- Faire une méthode qui calcule F automatiquement.
- ...

Vous trouverez quelques informations complémentaires à l'adresse :

`http://www-igm.univ-mlv.fr/~vnozick/`

Bon courage.