Quelques notes pour le cours de
Vision par Ordinateur

Peter Sturm
INRIA Rhône-Alpes, Equipe-projet PERCEPTION
Peter.Sturm@inrialpes.fr
http://perception.inrialpes.fr/people/Sturm/

Table des matières

0 Généralités

0.1 Bibliographie générale

- R.I. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- O. Faugeras, *Three-Dimensional Computer Vision - A Geometric Viewpoint*, The MIT Press, Cambridge, MA, USA, 1993.
- R. Horaud et O. Monga, *Vision par ordinateur : outils fondamentaux*, Deuxième édition revue et augmentée Éditions Hermès, Paris, 1995.

0.2 Notations

Les matrices sont représentées comme ceci : P. Les vecteurs comme ceci : $\mathbf{a}$. Les scalaires comme ceci : s. Les éléments d'une matrice ou d'un vecteur sont notés comme scalaires indexés : P_{ij} ou a_i.

La matrice d'identité de taille $n \times n$ est notée : $\mathtt{I}_n$ (l'indexe peut être omis si la taille de la matrice découle du contexte). Le vecteur nul de longueur n est noté : $\mathbf{0}_n$ (ou $\mathbf{0}$ si la longueur découle du contexte).

Les vecteurs de coordonnées associés à des points 3-D sont notés en majuscules, ceux de points 2-D en minuscules : $\mathbf{Q}$ respectivement $\mathbf{q}$.

La transposée et l'inverse d'une matrice sont notées P^T respectivement P^{-1}. La transposée de l'inverse d'une matrice est notée : $\mathsf{P}^{-\mathsf{T}}$.

Les vecteurs sont implicitement interprétés comme étant des vecteurs colonne, ou bien des matrices consistant d'une seule colonne. La transposée d'un vecteur, notée $\mathbf{a}^\mathsf{T}$, est donc interprétée comme étant un vecteur ligne, ou bien une matrice consistant d'une seule ligne.

Le produit scalaire de deux vecteurs $\mathbf{a}$ et $\mathbf{b}$ de longueur n : $\sum_{i=1}^{n} a_i b_i$, peut alors être noté comme : $\mathbf{a}^\mathsf{T}\mathbf{b}$.

Le symbole $\times$ exprime le produit vectoriel de deux vecteurs de longueur 3. Le vecteur $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ est orthogonal à $\mathbf{a}$ et $\mathbf{b}$.

Le produit vectoriel peut être exprimé par une multiplication matrice-vecteur :

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \mathbf{b} \ .$$

La matrice ci-dessus – la « matrice associée au produit vectoriel du vecteur $\mathbf{a}$ » est anti-symétrique. Elle est notée $[\mathbf{a}]_\times$.

Le symbole $\sim$ exprime l'égalité entre deux vecteurs ou deux matrices, à un facteur scalaire non nul près : $\mathbf{a} \sim \mathbf{b}$ veut dire qu'il existe $s \neq 0$ avec : $\mathbf{a} = s\mathbf{b}$. La notion d'égalité à un facteur près est importante si les coordonnées homogènes sont utilisées ...

0.3 Décompositions de matrices

Bibliographie :
- G.H. Golub et C.F. Van Loan, *Matrix Computations*, 3rd edition, The John Hopkins University Press, 1996.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling et B.P. Flannery, *Numerical Recipes in C*, 2nd edition, Cambridge University Press, 1992.

0.3.1 Décomposition QR

La décomposition QR d'une matrice A de taille $m \times n$ est donnée par :

$$\mathsf{A} = \mathsf{QR} \ ,$$

où Q est une matrice orthonormale de taille $m \times m$ et R une matrice triangulaire supérieure de taille $m \times n$. Ceci est la définition standard de la décomposition QR.
Une autre variante est utilisée dans ce cours ; nous la formulons pour des matrices carrées :

$$\mathsf{A}_{m\times m} = \mathsf{B}_{m\times m}\mathsf{C}_{m\times m} \ ,$$

où B est triangulaire supérieure :

$$\mathsf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ 0 & B_{22} & \cdots & B_{2m} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{mm} \end{pmatrix}$$

et C orthonormale (une matrice de rotation pour nous).

0.3.2 Décomposition de Cholesky

Soit A une matrice symétrique et définie positive, de taille $m \times m$. Elle peut alors être décomposée en :

$$\mathsf{A} = \mathsf{B}\mathsf{B}^{\mathsf{T}} \ ,$$

où B est une matrice triangulaire supérieure[1] de taille $m \times m$.

[1] La définition standard de la décomposition est pour des matrices triangulaires inférieures.

1 Modélisation de caméra

Afin de pouvoir effectuer des calculs numériques ou des raisonnements géométriques à partir d'images, nous avons besoin d'un modèle qui décrit comment le monde 3-D se projette sur une image 2-D, projection réalisée par une caméra. Il existe beaucoup de modèles, qui décrivent plus ou moins bien les caractéristiques d'une caméra (optique, électronique, mécanique). Le modèle typiquement utilisé en vision par ordinateur présente un bon compromis entre la simplicité des équations associées et la proximité aux phénomènes physiques modélisés. Il s'agit du **modèle sténopé** (ou bien **modèle de trou d'épingle**, ou encore **pinhole model** en anglais) qui représente en effet une projection perspective.

1.1 Modélisation géométrique

Sur le plan géométrique, le modèle sténopé peut être décrit comme l'ensemble d'un **centre de projection** (ou aussi centre optique) et d'un **plan image** (ou bien d'une **rétine**). Un point 3-D est projeté le long du rayon qui le lie avec le centre de projection – son **point image** étant l'intersection de ce rayon avec le plan image (voir la figure 1). Ce rayon est parfois appelé **rayon de projection** ou **ligne de vue**. Cette projection est effectivement une projection perspective.

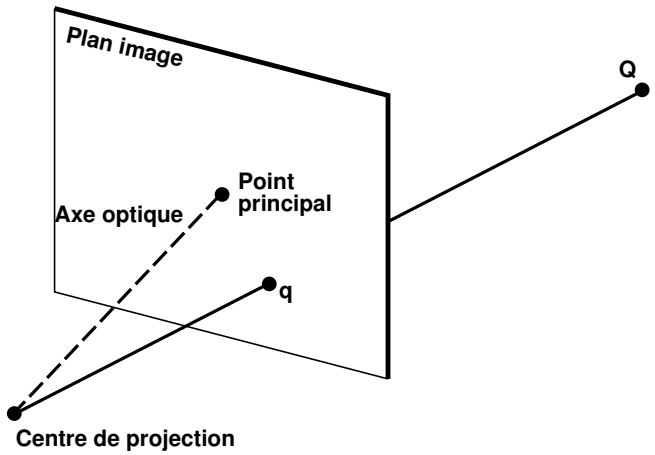


FIG. 1 – Le modèle sténopé.

Avant de continuer, nous introduisons deux notations à l'aide de la figure 1. La droite qui passe par le centre de projection et qui est perpendiculaire au plan image, est appelé **axe optique**. Le point d'intersection de l'axe optique avec le plan image est le **point principal**.

1.2 Modélisation algébrique

Nous venons de décrire un modèle géométrique pour des caméras. Pour effectuer des calculs, nous devons établir une description algébrique analogue. Dans la suite, nous allons faire ceci,

tout en identifiant des paramètres physiques de caméra. Afin de faire des calculs avec des points, nous avons besoin de coordonnées et donc de repères de coordonnées. Nous allons utiliser les coordonnées homogènes.

1.2.1 Étape 1 – Prise en compte de la distance focale

Nous allons dériver les équations de projection à l'aide de 4 repères (voir la figure 2). Commençons avec un repère 3-D attaché à la caméra – le **repère caméra**. Comme origine nous choisissons le centre de projection et comme axe des Z l'axe optique. Les axes des X et des Y sont choisis comme étant parallèles au plan image et perpendiculaires entre eux. En plus, les axes sont choisis tels qu'ils sont « alignés » avec les pixels.

Nous définissons un repère 2-D pour le plan image – le **repère image**. Son origine est le point principal. Ses axes des x et y sont parallèles aux axes des X et Y du repère caméra. En quelque sorte, le repère image peut être vu comme la projection orthogonale du repère caméra.

Nous introduisons la **distance focale** f comme étant la distance entre le centre de projection et le plan image (donc la distance entre le centre de projection et le point principal).

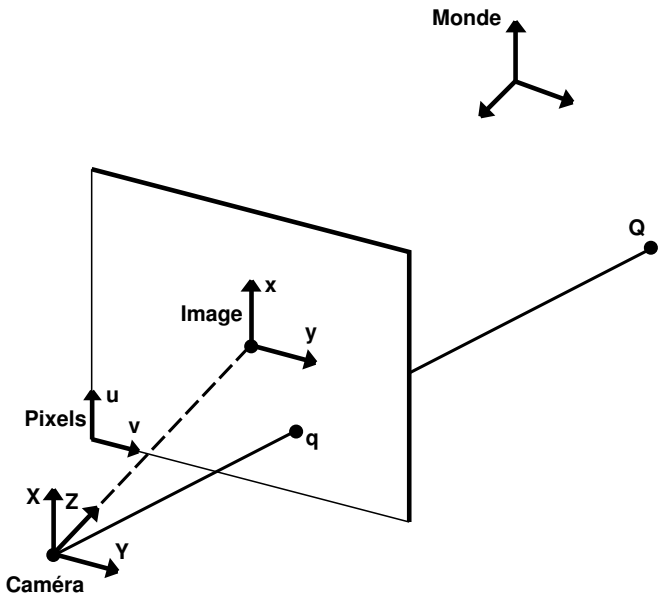


FIG. 2 – Les repères utilisés.

Nous pouvons maintenant dériver les équations de projection d'un point 3-D $\mathbf{Q}$, dont les coordonnées sont donnés par rapport au repère caméra (ce qui s'exprime par l'exposant c) :

$$\mathbf{Q}^c = \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} \quad .$$

Les coordonnées x et y du point image $\mathbf{q}$, projection de $\mathbf{Q}$, peuvent être calculées à l'aide des relations entre triangles similaires (voir cours) :

$$x = f\frac{X^c}{Z^c} \tag{1}$$

$$y = f\frac{Y^c}{Z^c} \ . \tag{2}$$

En coordonnées homogènes, on peut représenter ces équations par une multiplication matrice-vecteur :

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} \ . \tag{3}$$

Notons le symbole $\sim$ au milieu : l'égalité vectorielle n'est définie qu'à un facteur scalaire près (ce qui est le prix à payer pour l'utilisation de coordonnées homogènes) !

Nous vérifions maintenant si l'équation (3) est équivalente aux équations (1) et (2). Partons du côté droit de l'équation (3) :

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} = \begin{pmatrix} fX^c \\ fY^c \\ Z^c \end{pmatrix} \ .$$

Ceci est un vecteur de cordonnées homogènes. Si nous passons aux coordonnées non homogènes (en divisant par la dernière coordonnée), nous obtenons :

$$\begin{pmatrix} fX^c \\ fY^c \\ Z^c \end{pmatrix} \sim \begin{pmatrix} fX^c/Z^c \\ fY^c/Z^c \\ 1 \end{pmatrix} \ .$$

Il est maintenant facile de voir que ceci est équivalent aux équations (1) et (2).

1.2.2 Étape 2 – Prise en compte des pixels

Dans ce cours, nous considérons surtout des images digitales (donc adaptées pour la vision *par ordinateur*). Une image digitale est en effet une matrice de valeurs (e.g. niveaux de gris, couleurs RVB) – les cellules étant les **pixels**. Pour des traitements d'image par exemple, les pixels sont identifiés par des coordonnées – en effet, on les énumère tout simplement, une fois en direction horizontale, une fois pour la verticale. Ceci implique que nous ne pouvons pas directement utiliser les coordonnées 2-D définies dans la section précédente (i.e. le repère image), mais que nous devons prendre en compte un changement de repère.

Différents logiciels d'affichage ou éditeurs d'images utilisent parfois différents repères pour identifier les pixels. Soit le coin en haut à gauche, soit celui en bas à gauche de l'image est utilisé comme origine. L'axe de la première coordonnée est soit choisi horizontalement, soit

verticalement. Ici, pour simplifier les choses, nous définissons le **repère pixels** comme c'est montré sur la figure 2. Les coordonnées pixelliques sont notées u et v.

Le changement de repère nécessite alors tout d'abord une translation. En plus, il faut effectuer un changement d'unité : le repère *image* est un repère métrique – on mesure par exemple en mm ; dans le repère *pixels* l'unité est « nombre de pixels » – ce qui n'est pas une unité métrique. Surtout dans de vieilles caméras, les pixels ne sont parfois pas carrés, mais ils ont un côté plus grand que l'autre (c'est dû à des standards de télévision). Ceci implique que chaque axe devra subir un changement d'unité individuel.

Soient $-x_0$ et $-y_0$ les coordonnées du coin en bas à gauche de l'image, par rapport au repère image. Soit k_u la densité de pixels en direction de l'axe des u et k_v celle pour l'axe des v (exprimées en nombre de pixels par mm par exemple).

Considérons le point $\mathbf{q}$ avec les coordonnées x et y dans le repère image. Ses coordonnées par rapport au repère pixels sont obtenues en effectuant une translation, puis les changements d'unité décrits – en coordonnées homogènes :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad . \tag{4}$$

Nous pouvons maintenant combiner les équations (3) et (4) pour modéliser la projection complète du repère caméra vers le repère pixels :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} \quad . \tag{5}$$

Remarques. Pour donner des ordres de grandeur : la distance focale f varie typiquement de quelques mm jusqu'à une ou quelques centaines (au-delà, elle est souvent considérée comme étant infinie). La surface photosensible d'une caméra est typiquement un rectangle avec des bords de quelques mm de longueur. Pour une image de 512×512 pixels, correspondant à une surface photosensible de $5,12 \times 5,12mm^2$, nous avons alors des valeurs $k_u = k_v = 100\frac{1}{mm}$.

Pour des caméras avec une optique bien ajustée, le point principal se trouve normalement proche du centre de la surface photosensible, donc : $x_0 \approx y_0 \approx 2,56mm$ pour notre exemple.

1.2.3 Étape 3 – Prise en compte des déplacements

Jusqu'ici nous avons représenté les points 3-D dans un repère attaché à la caméra. Afin de prendre en compte les déplacements que les caméras vont effectuer, nous devons choisir un repère « statique », attaché à la **scène** 3-D. Nous introduisons donc un **repère monde** (voir la figure 2), qui peut être choisi de manière arbitraire (mais qui restera le même pour toute une application). La position des points 3-D **et** des caméras sont alors décrites par rapport à ce repère.

Nous modélisons dans la suite la position de la caméra. Elle comprend la **position** proprement dite – la position du centre de projection – et l'**orientation** de la caméra. La position

est représenté par un vecteur-3 $\mathbf{t}$ tel que

$$\begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}$$

sont les coordonnées homogènes du centre de projection. L'orientation de la caméra peut être exprimée par une rotation, ce qui sera représenté ici par une **matrice de rotation** [2] de taille 3×3 R. À la fin de cette section, nous rappelons brièvement quelques caractéristiques des matrices de rotation.

Soient X^m, Y^m, Z^m les coordonnées du point 3-D $\mathbf{Q}$, exprimées dans le repère monde. Le changement de repère du repère monde vers le repère caméra peut être écrit comme suit :

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \end{pmatrix} = \mathsf{R} \left(\begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathbf{t} \right) = \mathsf{R} \begin{pmatrix} X^m \\ Y^m \\ Z^m \end{pmatrix} - \mathsf{R}\mathbf{t} \quad .$$

Et en coordonnées homogènes :

$$\begin{pmatrix} X^c \\ Y^c \\ Z^c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix} \quad . \tag{6}$$

Vérifions que le centre de projection est effectivement l'origine du repère caméra :

$$\begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R}\mathbf{t} - \mathsf{R}\mathbf{t} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \quad .$$

Remarque. Nous avons exprimé la position et l'orientation d'une caméra. Des *déplacements* peuvent alors être modélisés par exemple par une suite de matrices de rotation R et de vecteurs de translation $\mathbf{t}$.

Rappels sur les matrices de rotation. Les matrices qui, pour nous, servent à représenter les rotations, sont les matrices *orthonormales* : leurs colonnes (et lignes) sont des vecteurs-3 mutuellement orthogonaux (leur produit scalaire s'annule) et de la norme 1. Ceci implique que le déterminant vaut ± 1. Pour une matrice de rotation, le déterminant doit valoir $+1$ (s'il vaut -1, la matrice représente une *réflexion*). Tout ceci implique que l'inverse d'une matrice de rotation est sa transposée : $\mathsf{R}\mathsf{R}^\mathsf{T} = \mathtt{I}$.

Chaque rotation peut être décomposée en trois rotations de base, autour des axes du repère actuel. Parmi les différentes permutations possibles, une souvent utilisée est :

$$\mathsf{R} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \quad .$$

Les angles α, β et γ sont associés aux axes des X, Y et Z respectivement. Il s'agit des **angles d'Euler**.

[2] Il existe d'autres représentations, e.g. les quaternions.

1.2.4 Modèle complet

Nous combinons simplement les résultats des sections précédentes, à savoir les équations (5) et (6) pour obtenir le modèle algébrique de caméra complet :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} X^m \\ Y^m \\ Z^m \\ 1 \end{pmatrix} \quad . \tag{7}$$

Nous identifions la matrice 3×3 K comme :

$$\mathsf{K} = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad .$$

Ainsi, les matrices de l'équation (7) peuvent s'écrire :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{K} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

ou bien :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{KR} & -\mathsf{KR}\mathbf{t} \end{pmatrix} \quad . \tag{8}$$

ou encore :

$$\mathsf{P} \sim \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix} \quad .$$

Nous appelons la matrice 3×4 P dans l'équation (8) la **matrice de projection** associée à la caméra. Elle projette des points 3-D sur des points 2-D, tous représentés en coordonnées homogènes.

1.2.5 Paramètres extrinsèques et intrinsèques

Nous pouvons regrouper les paramètres définissant la projection effectuée par une caméra, en deux ensembles : les **paramètres extrinsèques** – la matrice de rotation R et la position $\mathbf{t}$ – représentent la position et l'orientation de la caméra par rapport au « monde extérieur ».

Les **paramètres intrinsèques**, quant à eux, représentent les caractéristiques internes de la caméra, qui sont invariantes à sa position. Il s'agit des paramètres f, k_u, k_v, x_0 et y_0 rencontrés dans les sections 1.2.1 et 1.2.2. Ils sont regroupés dans une matrice triangulaire supérieure K (voir plus haut) :

$$\mathsf{K} = \begin{pmatrix} k_u f & 0 & k_u x_0 \\ 0 & k_v f & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad .$$

Nous voyons que ces 5 paramètres définissent 4 éléments de cette matrice. Nous définissons alors 4 paramètres intermédiaires (qui en plus ont l'avantage d'être « sans unité ») :

$$\begin{aligned} \alpha_u &= k_u f \\ \alpha_v &= k_v f \\ u_0 &= k_u x_0 \\ v_0 &= k_v y_0 \ . \end{aligned}$$

C'est ces 4 paramètres que l'on considèrera dans la suite comme les **paramètres intrinsèques** de la caméra. Leur interprétation est la suivante :

- α_u et α_v expriment la distance focale, en nombre de pixels (une fois en direction horizontale, une fois verticalement).
- u_0 et v_0 sont les coordonnées du point principal, exprimées dans le repère pixels.

La matrice K est alors :

$$\mathsf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \ . \tag{9}$$

Nous appelons K la **matrice de calibrage** de la caméra.

Remarque. Le rapport des longueurs de bord des pixels : $\tau = k_u/k_v$ est un paramètre important. Il est appelé **rapport d'échelle** et **aspect ratio** en anglais. Pour les caméras modernes, il vaut normalement 1, c'est-à-dire que les pixels sont carrés.

Remarque. Parfois, un paramètre intrinsèque supplémentaire est introduit, qui permet de modéliser des axes des u et v non perpendiculaires. Ce **skew**[3] s est introduit de la manière suivante dans la matrice de calibrage :

$$\mathsf{K} = \begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \ .$$

Dans la suite, nous négligeons ce paramètre (nous le considérons comme égal à 0).

[3] Il n'y a pas de terme français qui soit couramment utilisé.

2 Calibrage de caméra

Le calibrage d'une caméra a pour but principal la détermination de ses paramètres intrinsèques. Pourtant, il est difficile de dissocier les paramètres intrinsèques de ceux extrinsèques lors de calculs. C'est pourquoi le processus standard de calibrage détermine les deux ensembles de paramètres simultanément.

Le processus standard consiste à prendre une image d'un objet dont la structure est parfaitement connue – une **mire de calibrage**. Pour ce qui est de la mire montrée sur la figure 3 par exemple, les positions des « cibles » blanches sont connues avec une très haute précision, dans un repère attaché à la mire. Dans l'image, les cibles sont extraites par un traitement d'image. Il faut ensuite établir une *mise en correspondance* : pour chaque cible dans l'image, il faut déterminer la cible en 3-D dont elle est la projection. Ce processus est typiquement fait de manière semi-automatique.

FIG. 3 – La mire de calibrage utilisée dans le projet MOVI.

Nous avons alors un ensemble de n points 3-D avec des coordonnées connues, et des n points image correspondants, dont les coordonnées 2-D sont également connues. Pour chacune de ces correspondances, nous pouvons alors établir une équation de la forme (7) et tenter de les résoudre pour déterminer les paramètres de caméra inconnus.

L'équation (7) présente l'inconvénient que les inconnues sont multipliées entre elles, ce qui rend les équations non linéaires et difficiles à résoudre. Nous n'allons donc pas directement déterminer les paramètres intrinséques et extrinsèques, mais d'abord les 12 éléments de la matrice de projection P composée comme dans l'équation (8). Une fois P calculée, les paramètres intrinséques et extrinsèques peuvent en être extraits, comme ce sera montré plus bas.

2.1 Calcul de la matrice de projection

Nous disposons donc de n équation du style :

$$\mathbf{q}_p = \begin{pmatrix} u_p \\ v_p \\ 1 \end{pmatrix} \sim \mathsf{P}\mathbf{Q}_p \ , \tag{10}$$

où $\mathbf{q}_p$ et $\mathbf{Q}_p$, pour $p = 1 \ldots n$ sont les vecteurs de coordonnées homogènes des points 2-D et 3-D respectivement, et P la matrice de projection inconnue, de taille 3×4.

L'équation (10) étant définie à un facteur scalaire près seulement (exprimée par le symbole $\sim$), nous ne pouvons pas l'utiliser directement. Nous passons alors des coordonnées homogènes aux coordonnées standard, en divisant chaque côté de l'équation par sa troisième coordonnée, ce qui donne :

$$\begin{aligned} u_p &= \frac{(\mathsf{P}\mathbf{Q}_p)_1}{(\mathsf{P}\mathbf{Q}_p)_3} \\ v_p &= \frac{(\mathsf{P}\mathbf{Q}_p)_2}{(\mathsf{P}\mathbf{Q}_p)_3} \ . \end{aligned}$$

Multipliant ces équations par le dénominateur du côté droit respectif, nous obtenons :

$$\begin{aligned} u_p \, (\mathsf{P}\mathbf{Q}_p)_3 &= (\mathsf{P}\mathbf{Q}_p)_1 \\ v_p \, (\mathsf{P}\mathbf{Q}_p)_3 &= (\mathsf{P}\mathbf{Q}_p)_2 \ , \end{aligned}$$

ou bien :

$$\begin{aligned} u_p \, (P_{31}Q_{p,1} + P_{32}Q_{p,2} + P_{33}Q_{p,3} + P_{34}Q_{p,4}) &= P_{11}Q_{p,1} + P_{12}Q_{p,2} + P_{13}Q_{p,3} + P_{14}Q_{p,4} \\ v_p \, (P_{31}Q_{p,1} + P_{32}Q_{p,2} + P_{33}Q_{p,3} + P_{34}Q_{p,4}) &= P_{21}Q_{p,1} + P_{22}Q_{p,2} + P_{23}Q_{p,3} + P_{24}Q_{p,4} \ . \end{aligned}$$

Nous pouvons observer que ces équations sont linéaires en les éléments P_{ij} de la matrice de projection P, donc « aisées » à résoudre. Les équations ci-dessus peuvent être regroupées en un système d'équations, écrit sous la forme matricielle suivante :

$$\left(\begin{array}{ccc|ccc|ccc} Q_{1,1} & \cdots & Q_{1,4} & 0 & \cdots & 0 & -u_1Q_{1,1} & \cdots & -u_1Q_{1,4} \\ 0 & \cdots & 0 & Q_{1,1} & \cdots & Q_{1,4} & -v_1Q_{1,1} & \cdots & -v_1Q_{1,4} \\ \hline & \vdots & & & \vdots & & & \vdots & \\ & \vdots & & & \vdots & & & \vdots & \\ \hline Q_{n,1} & \cdots & Q_{n,4} & 0 & \cdots & 0 & -u_nQ_{n,1} & \cdots & -u_nQ_{n,4} \\ 0 & \cdots & 0 & Q_{n,1} & \cdots & Q_{n,4} & -v_nQ_{n,1} & \cdots & -v_nQ_{n,4} \end{array}\right) \left(\begin{array}{c} P_{11} \\ \vdots \\ P_{14} \\ \hline P_{21} \\ \vdots \\ P_{24} \\ \hline P_{31} \\ \vdots \\ P_{34} \end{array}\right) = \mathbf{0} \ ,$$

ou bien :

$$\mathsf{A}_{2n\times 12} \ \mathbf{x}_{12} = \mathbf{0}_{2n} \ .$$

La matrice A peut être calculée à partir des coordonnées des points et le vecteur $\mathbf{x}$ contient les inconnues – les 12 éléments de P. Chaque correspondance de points fournit deux équations, donc 6 correspondances ou plus sont en général suffisantes pour calculer P.

En pratique, les données sont normalement « bruitées » : on ne connaît les coordonnées des $\mathbf{Q}_p$ qu'avec une précision finie et surtout l'extraction des $\mathbf{q}_p$ dans l'image ne peut se faire sans imprécision. Le bruit implique qu'il n'y a en effet pas de solution $\mathbf{x}$ exacte, c'est-à-dire tel que $\mathsf{A}\mathbf{x}$ est exactement égal au vecteur nul. Il faut donc déterminer une solution $\hat{\mathbf{x}}$ qui soit la meilleure possible selon un certain critère. La possibilité que nous utilisons est de déterminer la solution aux moindres carrés :

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{i=1}^{2n} \left((\mathsf{A}\mathbf{x})_i \right)^2 \quad .$$

Afin de ne pas obtenir la solution triviale ($\mathbf{x} = \mathbf{0}$), nous devons ajouter une contrainte, par exemple sur la norme : $\|\mathbf{x}\| = 1$. Donc, nous déterminons :

$$\begin{aligned} \hat{\mathbf{x}} \quad = \quad & \arg\min_{\mathbf{x}} \sum_{i=1}^{2n} \left((\mathsf{A}\mathbf{x})_i \right)^2 \\ & \text{sous la contrainte } \|\mathbf{x}\| = 1 \quad . \end{aligned}$$

Comment résoudre ce système est expliqué dans le cours « Optimisation » du M2R IVR 2005/06, dont le poly sera distribué (la solution est typiquement obtenue en utilisant la *décomposition en valeurs singulières* ou *singular value decomposition* – SVD).

Nous avons mentionné que 6 correspondances de points seraient suffisantes pour obtenir une solution. Afin de réduire les effets du bruit dans les données, il est pourtant conseillé d'utiliser le plus de correspondances possible (en pratique, une ou plusieurs centaines de points sont normalement utilisés).

2.2 Extraction des paramètres intrinsèques et extrinsèques

De la matrice de projection P, nous pouvons finalement extraire les paramètres qui nous intéressent – la matrice de calibrage K, la matrice de rotation R et la position $\mathbf{t}$ de la caméra (R et $\mathbf{t}$ seront donnés par rapport au repère dans lequel les points 3-D $\mathbf{Q}_p$ sont donnés). D'après (8), nous avons :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{KR} & -\mathsf{KR}\mathbf{t} \end{pmatrix} \quad .$$

2.2.1 Extraction des paramètres intrinsèques

Soit $\bar{\mathsf{P}}$ la sous-matrice 3×3 composée des trois premières colonnes de P. Nous avons alors :

$$\bar{\mathsf{P}} \sim \mathsf{KR} \quad .$$

Si nous multiplions chaque côté de l'équation avec sa transposée :

$$\bar{\mathsf{P}}\bar{\mathsf{P}}^{\mathsf{T}} \sim \mathsf{KRR}^{\mathsf{T}}\mathsf{K}^{\mathsf{T}} \quad .$$

Puisque R est orthonormale : $\mathsf{R}\mathsf{R}^\mathsf{T} = \mathtt{I}$. Alors :

$$\bar{\mathsf{P}}\bar{\mathsf{P}}^\mathsf{T} \sim \mathsf{K}\mathsf{K}^\mathsf{T} \ . \tag{11}$$

Nous voyons à gauche une matrice symétrique et définie positive [4]. Sa décomposition de Cholesky (voir §0.3.2) donne une matrice triangulaire supérieure B avec :

$$\bar{\mathsf{P}}\bar{\mathsf{P}}^\mathsf{T} \sim \mathsf{B}\mathsf{B}^\mathsf{T} \ .$$

Si nous comparons cette équation avec (11), tout en nous rappelant que K est triangulaire supérieure, nous constatons alors que B est effectivement la matrice de calibrage K recherchée !

Ou presque : l'élément $(3,3)$ de K doit être égal à 1 (cf. l'équation (9)). Nous multiplions alors B avec le scalaire adéquat afin de satisfaire cette contrainte. Ensuite, nous pouvons extraire les paramètres intrinsèques individuels de B (ou K) d'après (9) :

$$\mathsf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \ .$$

2.2.2 Extraction des paramètres extrinsèques

Une fois les paramètres intrinsèques extraits, le calcul des paramètres extrinsèques n'est plus très difficile. D'après (8), nous avons :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{K}\mathsf{R} & -\mathsf{K}\mathsf{R}\mathbf{t} \end{pmatrix} \ .$$

Alors :

$$\mathsf{K}^{-1}\mathsf{P} \sim \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \end{pmatrix} \ .$$

Soit A la sous-matrice 3×3 constituée des trois premières colonnes de $\mathsf{K}^{-1}\mathsf{P}$. Nous avons donc : $\mathsf{A} \sim \mathsf{R}$. Nous pouvons rendre cette équation exacte (faire disparaître le symbole $\sim$) en multipliant A avec un scalaire λ approprié :

$$\lambda\mathsf{A} = \mathsf{R} \ .$$

Comment choisir λ ? L'égalité des matrices implique l'égalité de leurs déterminants, d'où nous obtenons :

$$\lambda^3 \det \mathsf{A} = \det \mathsf{R} = +1 \ .$$

Alors, nous choisissons

$$\lambda = \sqrt[3]{1/\det \mathsf{A}} \ .$$

Maintenant :

$$\lambda\mathsf{K}^{-1}\mathsf{P} = \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \end{pmatrix} \ .$$

(notons l'égalité exacte : $=$). Les trois premières colonnes de la matrice de gauche nous donnent alors directement la matrice de rotation R. Une fois R déterminée, le calcul du vecteur $\mathbf{t}$ est trivial.

[4] Le produit d'une matrice non singulière avec sa transposée est une matrice symétrique et définie positive.

3 Mosaïques d'images

Nous considérons le cas où nous disposons de plusieurs images prises par la même caméra, toutes *du même point de vue* (mais avec des orientations différentes). Il n'y a donc pas la possibilité de remonter à des informations 3-D (sans faire de la reconnaissance d'objets etc.). Par contre, on peut essayer de coller les images les unes aux autres afin de créer une image plus grande et plus complète de la scène. Le plus souvent, des images couvrant un tour de 360° sont prises ; collées ensemble, elles fournissent alors une *image panoramique*, qui pourra être utile pour la visualisation d'une scène de l'intérieur par exemple. Cette image panoramique, ou *mosaïque*, permettra par exemple la création d'images intermédiaires, à laquelle aucune des images originales ne correspond. Ainsi, un utilisateur peut visualiser la scène en « mode vidéo » [5].

Dans cette section, nous verrons comment coller des images prises du même point de vue, ce qui revient essentiellement à l'estimation de transformations projectives entre des paires d'images. Nous ne considérons au début que deux images ; l'application des résultats à une suite d'images étant relativement directe.

Le but est donc de trouver une transformation projective qui permette de coller une image à l'autre. Nous choisissons des repères de coordonnées comme au chapitre précédent. Puisque les images sont prises du même point de vue, nous simplifions les choses en adoptant ce point de vue (le centre de projection) comme origine du repère monde. Ainsi, la position de la caméra est évidemment donnée par : $\mathbf{t} = \mathbf{0}$.

Soient K la matrice de calibrage de la caméra, et R_1 et R_2 les matrices de rotation pour les deux prises d'image. Les deux matrices de projection sont alors, d'après (8) :

$$\begin{aligned} \mathsf{P}_1 &\sim \begin{pmatrix} \mathsf{K}\mathsf{R}_1 & \mathbf{0} \end{pmatrix} \\ \mathsf{P}_2 &\sim \begin{pmatrix} \mathsf{K}\mathsf{R}_2 & \mathbf{0} \end{pmatrix} . \end{aligned}$$

Considérons maintenant un point 3-D $\mathbf{Q}$ (donné en corodonnées homogènes) :

$$\mathbf{Q} = \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} .$$

Ses projections sont :

$$\mathbf{q}_1 \sim \mathsf{K}\mathsf{R}_1 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{12}$$

$$\mathbf{q}_2 \sim \mathsf{K}\mathsf{R}_2 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} . \tag{13}$$

[5] Voir e.g. QuickTime VR de Apple : `http ://www.apple.com/quicktime/qtvr/index.html`

Notons au passage que la projection de $\mathbf{Q}$ ne dépend pas de sa coordonnée T, ce qui veut dire en effet que tout point sur la ligne joignant $\mathbf{Q}$ au centre de projection, se projette sur le même point dans l'image.

En modifiant les équations (12) et (13) nous obtenons :

$$\begin{aligned} \mathsf{R}_1^\mathsf{T}\mathsf{K}^{-1}\mathbf{q}_1 &\sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ \mathsf{R}_2^\mathsf{T}\mathsf{K}^{-1}\mathbf{q}_2 &\sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad . \end{aligned}$$

D'où :

$$\mathsf{R}_2^\mathsf{T}\mathsf{K}^{-1}\mathbf{q}_2 \sim \mathsf{R}_1^\mathsf{T}\mathsf{K}^{-1}\mathbf{q}_1 \ ,$$

et finalement :

$$\mathbf{q}_2 \sim \mathsf{K}\mathsf{R}_2\mathsf{R}_1^\mathsf{T}\mathsf{K}^{-1}\mathbf{q}_1 \ . \tag{14}$$

Il existe alors une transformation projective :

$$\mathsf{H} \sim \mathsf{K}\mathsf{R}_2\mathsf{R}_1^\mathsf{T}\mathsf{K}^{-1} \ , \tag{15}$$

qui lie les deux projections d'un point 3-D. La définition de H ci-dessus ne fait pas intervenir les coordonnées du point 3-D de départ, ce qui veut dire que les deux projections de n'importe quel point 3-D sont liées par la même transformation H.

Notation. Les transformations projectives sont souvent aussi appelées **homographies**. En vision par ordinateur, le terme homographie est le plus souvent utilisé lorsqu'il s'agit d'une transformation projective en 2-D, comme ici.

Deux questions se posent maintenant :

- Comment déterminer l'homographie H entre deux images ?
- Comment l'utiliser pour coller des images les unes aux autres ?

3.1 Estimation de l'homographie

S'il est possible de mesurer les rotations que la caméra effectue (par exemple en la « motorisant » et à l'aide de capteurs d'angles) et si la caméra est calibrée (si la matrice de calibrage K est connue), alors on peut directement calculer H d'après (15). En pratique, on veut pourtant s'affranchir le plus possible de contraintes sur le matériel utilisé. Aussi, on voudra permettre à un utilisateur de prendre des images avec une caméra, tout en se tournant autour de lui-même. Donc, il sera difficile de connaître précisément la rotation qu'il effectue.

On va donc calculer l'homographie en utilisant les images elles-mêmes. Plus précisément, nous déterminons, dans un premier lieu, des correspondances de points entre deux images. Ceci peut être fait manuellement, mais il existe aussi des méthodes automatiques. La suite du processus est très similaire à ce qui a été fait lors du calibrage de caméra dans le chapitre précédent.

Soient $\mathbf{q}_1$ et $\mathbf{q}_2$ deux points correspondants, avec :

$$\mathbf{q}_1 = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \qquad \mathbf{q}_2 = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \quad .$$

Chaque correspondance de points nous donne alors une équation de la forme suivante sur l'homographie H :

$$\mathbf{q}_2 \sim \mathsf{H}\mathbf{q}_1 \quad .$$

Comme précédemment, nous devons passer des coordonnées homogènes aux coordonnées « standard » :

$$\begin{aligned} u_2 &= \frac{(\mathsf{H}\mathbf{q}_1)_1}{(\mathsf{H}\mathbf{q}_1)_3} \\ v_2 &= \frac{(\mathsf{H}\mathbf{q}_1)_2}{(\mathsf{H}\mathbf{q}_1)_3} \quad . \end{aligned}$$

Ensuite, nous multiplions chacune des équations avec le dénominateur du côté droit respectif :

$$\begin{aligned} u_2 \, (\mathsf{H}\mathbf{q}_1)_3 &= (\mathsf{H}\mathbf{q}_1)_1 \\ v_2 \, (\mathsf{H}\mathbf{q}_1)_3 &= (\mathsf{H}\mathbf{q}_1)_2 \quad . \end{aligned}$$

Nous pouvons expliciter ces équations comme suit :

$$\begin{aligned} u_2 \, (H_{31}u_1 + H_{32}v_1 + H_{33}) &= H_{11}u_1 + H_{12}v_1 + H_{13} \\ v_2 \, (H_{31}u_1 + H_{32}v_1 + H_{33}) &= H_{21}u_1 + H_{22}v_1 + H_{23} \quad . \end{aligned}$$

Toutes les équations, pour toutes les correspondances, peuvent alors être regroupées en un système d'équations linéaires :

$$\left(\begin{array}{ccc|ccc|ccc} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1u_2 & -v_1u_2 & -u_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1v_2 & -v_1v_2 & -v_2 \\ \hline \vdots & & & \vdots & & & \vdots & & \\ \vdots & & & \vdots & & & \vdots & & \\ \vdots & & & \vdots & & & \vdots & & \end{array}\right) \left(\begin{array}{c} H_{11} \\ H_{12} \\ H_{13} \\ \hline H_{21} \\ H_{22} \\ H_{23} \\ \hline H_{31} \\ H_{32} \\ H_{33} \end{array}\right) = \mathbf{0} \quad .$$

Pour ce qui est de la résolution de ce système, nous renvoyons aux remarques données en §2.1.

3.2 Application de l'homographie

L'homographie H nous indique en effet comment il faudrait transformer la première image afin de la faire se superposer sur la deuxième (au moins en ce qui concerne la partie commune des images). En même temps, elle nous donne les moyens d'introduire, dans la deuxième image, des parties qui ne sont visibles que dans la première image. La figure 5 montre l'exemple d'une mosaïque créée à partir des premières trois images montrées sur la figure 4. Ici, toutes les images étaient transférées vers la quatrième.

Visiblement, les trois premières images n'ont que de petites parties en commun entre elles. Donc, il n'était pas possible d'établir suffisamment de correspondances de points pour calculer les homographies associées. Pourtant, ayant calculé les homographies pour des paires d'images $(1,4), (2,4)$ et $(3,4)$, c'est-à-dire $\mathsf{H}_{14}, \mathsf{H}_{24}$ et H_{34}, il est possible de les « enchaîner » afin de calculer les autres homographies, par exemple :

$$\mathsf{H}_{13} \sim \mathsf{H}_{34}^{-1}\mathsf{H}_{14} \quad .$$

FIG. 4 – Quatre images des quais de Grenoble, prises du même point de vue.

Remarque. Nous n'avons traité que la « couche géométrique » du collage d'images. Il y a aussi une partie graphique ou de traitement d'images :

- coller des images les unes aux autres requiert d'abord le transfert des informations de couleur, de pixels dans une image vers des pixels d'une autre. En pratique, le transfert inclut une étape d'interpolation ;
- quand on tourne la caméra pour prendre les images, on se verra souvent confronté à un changement d'illumination apparente (si par exemple on se tourne progressivement vers le soleil, les images deviendront progressivement plus sombres). Pour créer des mosaïques de qualité il ne suffit alors plus de coller des images l'une à côté de l'autre, mais une correction photométrique appropriée est requise.

Remarque. Créer une mosaïque en collant plusieurs images à côté d'une autre, comme illustré ci-dessus, n'est pas la meilleure manière de procéder – il est préférable par exemple de créer une image cylindrique (plus de détails seront donnés lors du cours ...).

3.3 Calibrage en ligne

Rappelons comment l'homographie entre deux images dépend des rotations de la caméra et des paramètres intrinsèques :

$$\mathsf{H} \sim \mathsf{K}\mathsf{R}_2\mathsf{R}_1^{\mathsf{T}}\mathsf{K}^{-1} \quad .$$

Jusqu'ici, nous avons appliqué l'homographie telle qu'elle, sans nous soucier de ces paramètres. Il serait pourtant intéressant de déterminer par exemple la « rotation relative » entre

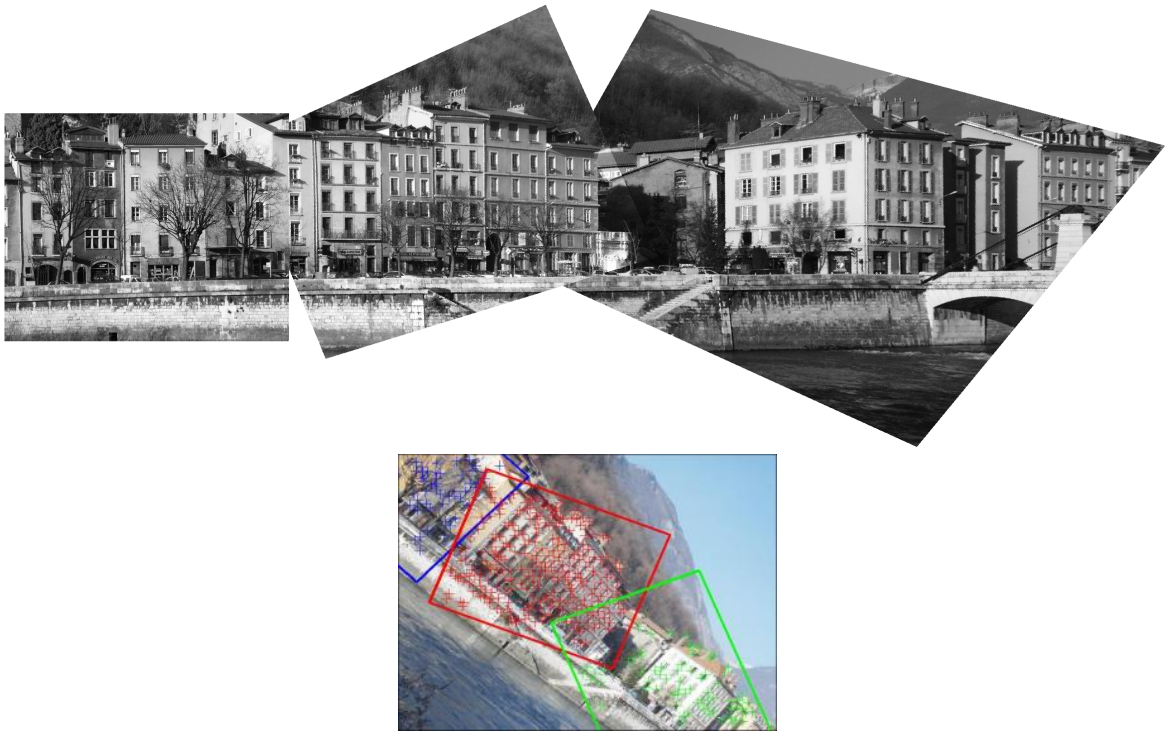


FIG. 5 – Mosaïque obtenue et illustration des positions relatives des trois premières images, vues dans la quatrième.

deux images : $\mathsf{R}_2\mathsf{R}_1^\mathsf{T}$. Ainsi, si des images couvrant $360°$ sont prises, il est effectivement possible de savoir quand on revient au début et ainsi de « peaufiner » la mosaïque.

Si la caméra est calibrée au préalable (donc *hors ligne*), nous pouvons calculer la rotation relative à partir de l'équation ci-dessus et de la connaissance de K : $\mathsf{R} \sim \mathsf{R}_2\mathsf{R}_1^\mathsf{T} \sim \mathsf{K}^{-1}\mathsf{H}\mathsf{K}$. Dans le cas contraire, nous pouvons essayer de procéder comme lors du calibrage, où nous avons décomposé une matrice de projection.

Transformons l'équation ci-dessus :

$$\mathsf{K}^{-1}\mathsf{H}\mathsf{K} \sim \mathsf{R}_2\mathsf{R}_1^\mathsf{T} \ .$$

Multiplier chaque côté de l'équation avec sa transposée donne :

$$\mathsf{K}^{-1}\mathsf{H}\mathsf{K}\ \mathsf{K}^\mathsf{T}\mathsf{H}^\mathsf{T}\mathsf{K}^{-\mathsf{T}} \sim \mathsf{R}_2\mathsf{R}_1^\mathsf{T}\ \mathsf{R}_1\mathsf{R}_2^\mathsf{T} \ .$$

Le côté droit se simplifie puisque $\mathsf{R}_1^\mathsf{T}\mathsf{R}_1 = \mathtt{I}$ et pareil pour R_2. Alors :

$$\mathsf{K}^{-1}\mathsf{H}\mathsf{K}\ \mathsf{K}^\mathsf{T}\mathsf{H}^\mathsf{T}\mathsf{K}^{-\mathsf{T}} \sim \mathtt{I} \ .$$

Il en découle :

$$\mathsf{H}\mathsf{K}\mathsf{K}^\mathsf{T}\mathsf{H}^\mathsf{T} \sim \mathsf{K}\mathsf{K}^\mathsf{T} \ .$$

Nous identifions :

$$\mathsf{A} = \mathsf{K}\mathsf{K}^\mathsf{T} \ .$$

Alors :

$$\mathsf{H}\mathsf{A}\mathsf{H}^\mathsf{T} \sim \mathsf{A} \ .$$

Cette équation n'est définie qu'à un facteur scalaire près. Nous pouvons la rendre exacte en trouvant un facteur scalaire adéquat pour la matrice H. Concrètement, il faut trouver le scalaire λ tel que les déterminants des deux côtés de l'équation sont égaux :

$$\det\left((\lambda\mathsf{H})\,\mathsf{A}\left(\lambda\mathsf{H}^\mathsf{T}\right)\right) = \det\mathsf{A} \ ,$$

ou bien :

$$\lambda^6\left(\det\mathsf{H}\right)^2\det\mathsf{A} = \det\mathsf{A} \ .$$

La solution est donnée par :

$$\lambda = \sqrt[3]{1/\det\mathsf{H}} \ .$$

Si nous identifions $\lambda\mathsf{H}$ par $\bar{\mathsf{H}}$, nous avons une équation *exacte* sur la matrice inconnue A :

$$\bar{\mathsf{H}}\mathsf{A}\bar{\mathsf{H}}^\mathsf{T} = \mathsf{A} \ .$$

Cette équation est linéaire en les éléments de A. Il peut être montré qu'une seule équation de ce type (c'est-à-dire une seule homographie) n'est pas suffisante pour entièrement calculer A. Avec deux homographies ou plus une solution unique pour A est pourtant possible en général (la solution est obtenue avec la méthode des moindres carrés linéaires).

Une fois la matrice A calculée, nous pouvons en extraire la matrice de calibrage K, grâce à sa forme triangulaire, par décomposition de Cholesky, comme en §2.2.1. Maintenant, la caméra est calibrée et nous pouvons par exemple calculer les rotations relatives entre des images, comme mentionné ci-dessus.

Malgré les points communs entre le processus de calibrage (§2) et la manière de déterminer K décrite dans cette section, nous pouvons constater une différence essentielle : pour le calibrage, une image d'un objet 3-D *parfaitement connu* était utilisée, tandis qu'ici nous n'avons en aucun lieu utilisé une telle information. En effet, nous avons calibré la caméra sans véritable mire de calibrage. Nous avons effectivement vu une première instance du paradigme d'*auto-calibrage*. L'information qui a permis de déterminer les paramètres intrinsèques de la caméra, est précisément la connaissance que la caméra ne se déplace pas, mais qu'elle tourne autour de son centre de projection.

Un calibrage **hors ligne** consiste à prendre une image d'une mire de calibrage, de calibrer, et puis l'application envisagée peut être lancée. L'image de la mire ne sert qu'à calibrer, tandis que les autres images ne servent qu'à l'application. Dans cette section, par contre, nous n'avons a priori que des images destinées à l'application (création d'une mosaïque). Pourtant, ces images se sont également avérées utiles pour le calibrage. Nous parlons donc aussi de **calibrage en ligne** à la place d'auto-calibrage.

Remarque. Ici, nous avons traité la matrice K comme étant totalement inconnue (en dehors de sa forme triangulaire supérieure). En pratique, il est souvent possible de faire des hypothèses sur par exemple la position

du point principal u_0, v_0. Ainsi, moins de paramètres intrinsèques sont à estimer et il est possible de créer des algorithmes spécialisés qui necéssitent moins d'images. En général (si tous les paramètres intrinsèques sont inconnus), il faut au moins 3 images (donc 2 homographies), mais ce qui est le plus important, c'est que leurs rotations relatives doivent s'effectuer autour d'au moins 2 axes différents (sinon les équations seront redondantes).

3.4 Bibliographie

- H.-Y. Shum et R. Szeliski, *Panoramic Image Mosaics,* Technical Report MSR-TR-97-23, Microsoft Research, 1997.
- M. Jethwa, A. Zisserman et A. Fitzgibbon, *Real-time Panoramic Mosaics and Augmented Reality,* British Machine Vision Conference, pp. 852-862, 1998.

4 Reconstruction 3-D à partir de deux images complètement calibrées

Nous considérons deux images d'une scène, prises de points de vue *distincts*. La création de mosaïques n'est alors pas possible, par contre nous pouvons remonter à des informations 3-D (les mosaïques ne contiennent que des informations 2-D).

Le scénario le plus simple pour la reconstruction 3-D concerne le cas où tout est connu sur les deux images : les paramètres intrinsèques des deux caméras ainsi que leur positionnement, donc les matrices de projection, P_1 et P_2. Le problème de base est alors de déterminer les coordonnées d'un point 3-D $\mathbf{Q}$, en partant des deux matrices de projection et des deux points correspondants dans les images, $\mathbf{q}_1$ et $\mathbf{q}_2$. Il faut alors trouver $\mathbf{Q}$ (vecteur-4 de coordonnées homogènes) tel que :

$$\begin{aligned} \mathsf{P}_1\mathbf{Q} &\sim \mathbf{q}_1 \\ \mathsf{P}_2\mathbf{Q} &\sim \mathbf{q}_2 \ . \end{aligned}$$

Comme nous l'avons expliqué dans le chapitre précédent, les « mesures » $\mathbf{q}_1$ et $\mathbf{q}_2$ sont bruitées. Donc, les équations ci-dessus n'ont pas de solution exacte et il faut trouver une solution pour $\mathbf{Q}$ qui soit la meilleure, d'après un certain critère.

Beaucoup de méthodes ont été proposées dans la littérature ; nous en enonçons quelques-unes :

- à partir des matrices de projection et des points image, deux rayons de projection peuvent être créés (les rayons passant par les centres de projection et les points image associés). Nous adoptons alors pour $\mathbf{Q}$ le « point du milieu » des deux rayons, c'est-à-dire le point qui est équidistant aux deux rayons et qui minimise la distance. Ce point est unique ; il se trouve sur la perpendiculaire commune des deux rayons ;
- une solution similaire à la précédente, mais plus simple, consiste à supposer que $\mathbf{Q}$ se trouve exactement sur l'un ou l'autre des rayons associés à $\mathbf{q}_1$ et $\mathbf{q}_2$. Parmi les points sur le rayon, on choisit celui qui minimise la distance vers l'autre rayon ;
- en partant des équations ci-dessus, nous pouvons en déduire quatre équations linéaires en les coefficients de $\mathbf{Q}$:

$$\begin{aligned} u_1 \left(\mathsf{P}_1\mathbf{Q}\right)_3 &= \left(\mathsf{P}_1\mathbf{Q}\right)_1 \\ v_1 \left(\mathsf{P}_1\mathbf{Q}\right)_3 &= \left(\mathsf{P}_1\mathbf{Q}\right)_2 \\ u_2 \left(\mathsf{P}_2\mathbf{Q}\right)_3 &= \left(\mathsf{P}_2\mathbf{Q}\right)_1 \\ v_2 \left(\mathsf{P}_2\mathbf{Q}\right)_3 &= \left(\mathsf{P}_2\mathbf{Q}\right)_2 \ . \end{aligned}$$

  Ce système peut être résolu aux moindres carrés, de la même manière que lors du calibrage ou du calcul de l'homographie pour la création de mosaïques.

La troisième méthode peut directement être étendue au cas où plus de deux images du point $\mathbf{Q}$ seraient disponibles.

Remarque. Toutes les méthodes ci-dessus sont sous-optimales (plus de détails lors du cours), mais donnent souvent des solutions raisonnables. Il existe une méthode optimale, un peu plus compliquée, voir : R. Hartley et P. Sturm, *Triangulation*, Computer Vision and Image Understanding, Vol. 68, No. 2, pp. 146-157, 1997.

5 Détermination de la pose d'un objet

La section 2 a traité du calibrage d'une caméra : étant donnée une image d'un objet dont la structure est parfaitement connue (e.g. à travers les coordonnées 3-D de points marquants), il est en général possible de retrouver et les paramètres intrinsèques de la caméra, et sa position et orientation relatives à l'objet. Il a été mentionné que pour ce faire, les projections d'au moins 6 points sur l'objet doivent être identifiées dans l'image.

Un sous-problème est la **détermination de pose** : le même scénario est considéré, sauf que la caméra a déjà été calibrée au préalable (ses paramètres intrinsèques sont connus). Le problème se réduit alors à la détermination de la position et l'orientation – la *pose* – de la caméra par rapport à l'objet (ou vice-versa). Une conséquence immédiate de la réduction du nombre d'inconnues est qu'il ne faut plus absolument identifier les projections de 6 points. En effet, nous verrons dans la section suivante qu'avec les projections de seulement 3 points, on peut déjà arriver à un nombre fini de solutions. Avec 4 points ou plus, il y a en général une solution unique pour la pose.

Dans la suite, nous nous concentrons sur un cas particulier de la problématique, notamment le calcul de pose si l'objet en question est *plan*. Ceci est par exemple très intéressant pour la réalité augmentée ou les studios virtuels (des exemples d'applications sont donnés lors du cours).

Nous examinons d'abord le calcul de pose pour le cas minimum où les projections de seulement 3 points sont utilisées, puis nous donnons une méthode qui peut prendre en compte plusieurs points.

5.1 Première méthode, utilisant 3 points

Nous considérons que la caméra est calibrée, c'est-à-dire que nous connaissons sa matrice de calibrage K. Les images de 3 points sont identifiées et représentées par les trois vecteurs de coordonnées homogènes suivants :

$$\mathbf{q}_1 = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \qquad \mathbf{q}_2 = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \qquad \mathbf{q}_3 = \begin{pmatrix} u_3 \\ v_3 \\ 1 \end{pmatrix} \quad .$$

Nous connaissons la structure de l'objet original, qui est ici constitué de 3 points seulement. La représentation de la structure que nous allons utiliser ici n'est pas la collection des coordonnées des points en 3-D (comme en §2 pour le calibrage), mais simplement les distances entre ces 3 points : d_{12}, d_{13} et d_{23}.

Le but de cette section est de calculer la position, en 3-D et par rapport à la caméra, des 3 points. Une fois ceci est établi, on peut par exemple calculer la rotation et la translation qui constituent le changement d'un repère attaché à l'objet, au repère caméra.

On peut formuler le problème ainsi : on cherche 3 points 3-D $\mathbf{Q}_1, \mathbf{Q}_2$ et $\mathbf{Q}_3$ (coordonnées données par rapport au repère caméra) tels que les distances entre eux ont les valeurs connues

et tels qu'ils se projettent sur les points image donnés :

$$dist\left(\mathbf{Q}_i, \mathbf{Q}_j\right) = d_{ij} \qquad \forall i, j \in \{1,2,3\} \tag{16}$$

$$\mathsf{P}\mathbf{Q}_i \sim \mathbf{q}_i \qquad \forall i \in \{1,2,3\} \ . \tag{17}$$

Nous cherchons les coordonnées des points 3-D par rapport au repère caméra (donc, $\mathsf{R} = \mathtt{I}$ et $\mathbf{t} = \mathbf{0}$ dans l'équation (8)), donc la matrice de projection P est :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{K} & \mathbf{0} \end{pmatrix} \ .$$

Maintenant, il nous faut trouver une paramétrisation des inconnues. Nous pouvons supposer sans crainte que les points 3-D sont des points finis (puisqu'il s'agit de points sur un vrai objet). Alors, les coordonnées homogènes des points peuvent être écrites

$$\mathbf{Q}_i = \begin{pmatrix} \bar{\mathbf{Q}}_i \\ 1 \end{pmatrix} \quad \forall i \in \{1,2,3\}$$

avec des vecteurs-3 $\bar{\mathbf{Q}}_i$.
Nous pouvons écrire la projection de ces points dans notre caméra :

$$\mathbf{q}_i \sim \mathsf{P}\mathbf{Q}_i \sim \mathsf{K}\bar{\mathbf{Q}}_i \quad \forall i \in \{1,2,3\} \ .$$

Puisque la matrice de calibrage K ainsi que les points image $\mathbf{q}_i$ sont connus, nous disposons déjà d'une certaine connaissance sur les coordonnées des points 3-D :

$$\bar{\mathbf{Q}}_i \sim \mathsf{K}^{-1}\mathbf{q}_i \quad \forall i \in \{1,2,3\} \ .$$

La seule indétermination réside en le fait que ces équations ne sont définies qu'à des facteurs scalaires près. Nous rendrons exactes les équations en explicitement introduisant les facteurs scalaires :

$$\bar{\mathbf{Q}}_i = \lambda_i \mathsf{K}^{-1}\mathbf{q}_i \quad \forall i \in \{1,2,3\} \ .$$

Les coordonnées des points 3-D sont alors données par :

$$\mathbf{Q}_i = \begin{pmatrix} \lambda_i \mathsf{K}^{-1}\mathbf{q}_i \\ 1 \end{pmatrix} \quad \forall i \in \{1,2,3\} \ .$$

Les seules inconnues sont les 3 scalaires λ_1, λ_2 et λ_3 (ces scalaires indiquent en fait la position des points 3-D le long de leurs rayons de projection).

Jusqu'ici, nous avons donc réduit le problème posé par les équations (16) et (17) : nous avons construit les $\mathbf{Q}_i$ tels que l'équation (17) est satisfaite. Le problème restant est de déterminer les valeurs de λ_1, λ_2 et λ_3 telles que :

$$dist\left(\mathbf{Q}_i, \mathbf{Q}_j\right) = d_{ij} \quad \forall i, j \in \{1,2,3\} \ .$$

Si nous écrivons

$$\mathsf{K}^{-1}\mathbf{q}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \quad \forall i \in \{1,2,3\} \ ,$$

nous pouvons écrire la distance entre deux points comme suit :

$$dist\left(\mathbf{Q}_i, \mathbf{Q}_j\right) = \sqrt{\left(\lambda_i X_i - \lambda_j X_j\right)^2 + \left(\lambda_i Y_i - \lambda_j Y_j\right)^2 + \left(\lambda_i Z_i - \lambda_j Z_j\right)^2} \quad \forall i,j \in \{1,2,3\} \ .$$

Nous nous débarassons dans la suite de la racine en considérant les carrés des distances. Ainsi, les 3 équations du problème sont :

$$\left(\lambda_i X_i - \lambda_j X_j\right)^2 + \left(\lambda_i Y_i - \lambda_j Y_j\right)^2 + \left(\lambda_i Z_i - \lambda_j Z_j\right)^2 = d_{ij}^2 \quad \forall i,j \in \{1,2,3\} \ . \qquad (18)$$

Nous disposons donc de 3 équations en les 3 inconnues λ_1, λ_2 et λ_3, ce qui implique qu'en général il n'y a qu'un nombre fini de solutions. Concrètement, nous avons 3 équations quadratiques. Donc, en général il y a 8 solutions (dont quelques-unes peuvent être complexes, qui ne nous intéressent évidemment pas). Une méthode numérique pour résoudre ces équations est donnée lors du cours.

5.1.1 Comment trouver une solution unique

Nous obtenons donc un maximum de 8 solutions réelles pour le triplet $(\lambda_1, \lambda_2, \lambda_3)$, donc 8 solutions pour la pose. En pratique, il est parfois le cas que les positions des points 3-D obtenues ne sont pas réalisables : si l'un des points est « derrière » la caméra (sa coordonnée Z est négative), nous pouvons éliminer cette solution d'emblée.

Pour chaque solution, il existe en effet une solution « miroir » qui consiste à changer le signe de chacun des 3 scalaires λ_i : on peut observer que cette action n'affecte pas la véracité des équations (18). Pour chaque tel couple de solutions, nous pouvons donc en rejeter au moins une, parce qu'elle contient des points 3-D derrière la caméra. Par conséquent, des 8 solutions numériques, au maximum 4 ont un sens physique.

Afin de trouver une solution unique, nous devons en général disposer de plus d'informations. Si nous disposons de la projection $\mathbf{q}_4$ d'un quatrième point sur l'objet plan, ceci est relativement facile : pour chacune des solutions hypothétiques pour les 3 premiers points, nous pouvons déterminer quelle serait la position du quatrième point. Si la solution est correcte, alors la projection du quatrième point par la matrice de projection P doit coïncider avec le point image $\mathbf{q}_4$. En général, une seule des solutions de départ « survivra ».

Remarque. En pratique, la projection du quatrième point ne coïncidera pas exactement avec le point image $\mathbf{q}_4$, ce qui est dû à des erreurs d'extraction de points dans l'image, etc. On retiendra alors la solution pour laquelle la projection est la plus proche de $\mathbf{q}_4$.

5.2 Deuxième méthode, utilisant plusieurs points

Dans le paragraphe précédent, nous avons vu qu'avec plus de 3 points, nous pouvons trouver une solution unique pour la pose. La méthode décrite a pourtant un inconvénient : la qua-

lité de la solution dépend de 3 points seulement (les autres étant utilisés uniquement pour disambiguer les solutions multiples). Ainsi, si l'erreur dans l'extraction des points image est grande, la pose résultante sera de mauvaise qualité. Au cas où nous disposions de plus de 3 points, nous voudrions donc déterminer une pose qui fasse en quelque sorte la « moyenne ». Une méthode pour ce faire est expliquée dans la suite.

Par rapport à la méthode précédente, nous adoptons comme inconnues la matrice de rotation R et le vecteur de position $\mathbf{t}$ de la caméra. Aussi, nous n'utilisons plus les distances entres des points comme représentation de la structure de l'objet plan, mais directement les coordonnées des points, par rapport à un repère attaché à l'objet. Puisque l'objet est plan, nous pouvons par exemple choisir le repère tel que les coordonnées Z des points sont égales à 0. Ainsi, les coordonnées 3-D des n points sont :

$$\mathbf{Q}_i = \begin{pmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{pmatrix} \quad \forall i \in \{1 \dots n\} \ .$$

La matrice de projection de la caméra, exprimée par rapport au repère attaché à l'objet, s'écrit (cf. (8)) :

$$\mathsf{P} \sim \begin{pmatrix} \mathsf{KR} & -\mathsf{KR}\mathbf{t} \end{pmatrix} = \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix} \ .$$

Les équations de projection sont données par :

$$\mathbf{q}_i \sim \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix} \mathbf{Q}_i \quad \forall i \in \{1 \dots n\} \ .$$

Nous rappelons que toutes les entités impliquées sont connues, à l'exception de R et $\mathbf{t}$.

Dans la suite, nous procédons comme lors du calibrage de caméra, où d'abord des paramètres intermédiaires sont estimés (là-bas, la matrice de projection), suivi par l'extraction des paramètres recherchés de ceux intermédiaires (là-bas, les paramètres intrinsèques et extrinsèques). Observons de plus près la projection d'un point 3-D $\mathbf{Q}_i$:

$$\begin{aligned} \mathbf{q}_i &\sim \mathsf{KR} \begin{pmatrix} \mathtt{I} & -\mathbf{t} \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{pmatrix} \\ &= \mathsf{KR} \begin{pmatrix} 1 & 0 & 0 & -t_1 \\ 0 & 1 & 0 & -t_2 \\ 0 & 0 & 1 & -t_3 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{pmatrix} \\ &= \mathsf{KR} \begin{pmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & -t_3 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \ . \end{aligned}$$

Il existe donc une transformation projective (et bijective en général) entre les points sur l'objet plan, et les points correspondants dans le plan image. Cette transformation est représentée

par la matrice 3×3 :

$$\mathsf{H} \sim \mathsf{K}\mathsf{R} \begin{pmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & -t_3 \end{pmatrix} \quad , \tag{19}$$

telle que :

$$\mathbf{q}_i \sim \mathsf{H} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \quad \forall i \in \{1 \dots n\} \ .$$

La méthode de calcul de pose consistera alors de deux étapes : d'abord, nous estimons la transformation projective H entre l'objet plan et le plan image ; puis, nous extrayons de H les paramètres de la pose, à savoir la matrice de rotation R et le vecteur $\mathbf{t}$.

Pour ce qui est de l'estimation de la transformation projective, nous renvoyons à la méthode expliquée au paragraphe 3.1. Cette méthode peut directement être utilisée ici – la seule différence est la « signification » des points 2-D considérés : en §3.1, il s'agit de la transformation entre deux plans image (pour la construction d'une mosaïque), tandis qu'ici l'un des plans est un plan contenu dans la scène 3-D.

Ayant déterminé H, nous examinons dans la suite comment extraire R puis $\mathbf{t}$, d'après l'équation (19). Nous allons d'abord déterminer les deux premières colonnes de R, puis la troisième colonne, puis le vecteur $\mathbf{t}$.

5.2.1 Calcul des deux premières colonnes de R

Puisque la matrice de calibrage K est connue, nous pouvons calculer la matrice $\mathsf{M} = \mathsf{K}^{-1}\mathsf{H}$, avec laquelle nous avons :

$$\mathsf{M} \sim \mathsf{R} \begin{pmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & -t_3 \end{pmatrix} \quad . \tag{20}$$

Nous allons utiliser les colonnes de la matrice R, c'est-à-dire les vecteurs-3 $\mathbf{r}_1, \mathbf{r}_2$ et $\mathbf{r}_3$ avec :

$$\mathsf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix} \ .$$

L'équation (20) s'écrit alors, après introduction explicite d'un facteur scalaire λ :

$$\lambda\mathsf{M} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & -\mathsf{R}\mathbf{t} \end{pmatrix} \ . \tag{21}$$

Dans un premier temps, nous allons déterminer les deux premières colonnes de R. Nous rappelons d'abord que les vecteurs colonne de R sont tous de la norme 1 (cf. §1.2.3). Ceci nous aide à déterminer le facteur scalaire λ. En effet, les deux relations suivantes doivent être satisfaites (une pour chacune des deux premières colonnes de M) :

$$\begin{aligned} \lambda^2 \left(M_{11}^2 + M_{21}^2 + M_{31}^2 \right) &= 1 \\ \lambda^2 \left(M_{12}^2 + M_{22}^2 + M_{32}^2 \right) &= 1 \ . \end{aligned}$$

Nous pouvons constater deux choses :

- il y a deux solutions pour λ, qui ne se distinguent que par leur signe : $\pm\lambda$;
- à cause du bruit (e.g. dans la position des points image, qui ont servi à calculer H puis M), il n'y a en général pas de solution exacte aux deux équations précédentes. Nous pouvons alors par exemple les résoudre séparément, et adopter comme solution commune la moyenne des solutions individuelles.

Dans la suite, nous dénommons par $\mathbf{m}_1, \mathbf{m}_2$ et $\mathbf{m}_3$ les trois colonnes de $\lambda\mathsf{M}$ (retenons que nous devons faire ce qui suit aussi pour $-\lambda$). En théorie, nous pourrions donc directement déterminer les deux premières colonnes de R par : $\mathbf{r}_1 = \mathbf{m}_1$ et $\mathbf{r}_2 = \mathbf{m}_2$ (d'après l'équation (21)). Encore une fois, le bruit dans les données interviendra en pratique : une des conditions sur les colonnes de R est qu'elles sont mutuellement perpendiculaires (cf. §1.2.3). Les vecteurs $\mathbf{m}_1$ et $\mathbf{m}_2$ calculés ne satisferont cette condition que de manière approximative.

Une méthode pour trouver des vecteurs $\mathbf{r}_1$ et $\mathbf{r}_2$ admissibles est expliquée dans la suite (plus de détails lors du cours). Soit $\mathbf{v}$ le bi-secteur des vecteurs-3 $\mathbf{m}_1$ et $\mathbf{m}_2$. Les trois vecteurs $\mathbf{v}, \mathbf{m}_1$ et $\mathbf{m}_2$ se trouvent dans le même plan Π. Dans ce plan Π, il existe deux directions qui forment un angle de $45°$ avec $\mathbf{v}$. Pour chacune de ces directions, nous déterminons le vecteur associé qui soit de longueur (ou norme) 1 et qui est « proche » de $\mathbf{m}_1$ ou $\mathbf{m}_2$. Les deux vecteurs choisis satisfont toutes les contraintes (ils sont perpendiculaires l'un par rapport à l'autre, et de norme 1), nous pouvons donc les adopter comme $\mathbf{r}_1$ et $\mathbf{r}_2$.

5.2.2 Calcul de la troisième colonne de R

Nous rappelons que les colonnes de R sont mutuellement perpendiculaires et de norme 1. Il n'existe que deux vecteurs $\mathbf{r}_3$ qui satisfont ces conditions par rapport à $\mathbf{r}_1$ et $\mathbf{r}_2$. Ces deux vecteurs pointent dans des directions opposées (l'un peut être obtenu à partir de l'autre par multiplication avec le scalaire -1).

Remarque. Un vecteur $\mathbf{r}_3$ qui est perpendiculaire à deux vecteurs $\mathbf{r}_1$ et $\mathbf{r}_2$ peut être déterminé, à un facteur scalaire près, par le produit vectoriel : $\mathbf{r}_3 \sim \mathbf{r}_1 \times \mathbf{r}_2$.

Nous avons donc a priori deux solutions pour la matrice de rotation :

$$\mathsf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \pm\mathbf{r}_3 \end{pmatrix} \ .$$

Il peut être observé que les déterminants des deux solutions sont de signe opposé (changer le signe de tous les éléments de l'une des colonnes d'une matrice fait changer le signe du déterminant). Puisque le déterminant d'une matrice de rotation doit être positif (en effet, il doit valoir $+1$, cf. §1.2.3), nous pouvons donc éliminer une des deux solutions pour $\mathbf{r}_3$.

5.2.3 Calcul du vecteur $\mathbf{t}$

Le calcul de $\mathbf{t}$ s'avère être trivial. Soit $\mathbf{m}_3$ la troisième colonne de $\lambda\mathsf{M}$. D'après l'équation (21), nous avons :

$$\mathbf{m}_3 = -\mathsf{R}\mathbf{t} \ .$$

Donc :

$$\mathbf{t} = -\mathsf{R}^\mathsf{T}\mathbf{m}_3 \ .$$

5.2.4 Remarques

Rappelons-nous que lors du calcul des deux premières colonnes de R, deux solutions pour le facteur scalaire λ étaient possibles. On obtient donc deux solutions globales pour la rotation et la position.

Il peut être montré que les deux solutions correspondent à des positions de la caméra des deux côtés de l'objet plan : les deux positions sont en effet des reflexions l'une de l'autre, dans le plan de l'objet. A priori, il n'est pas possible d'éliminer une des solutions.

Pour obtenir une solution unique (donc, la solution correcte), il faut que seulement un côté de l'objet soit normalement visible (ce qui est vrai par exemple pour une peinture). Ainsi, quand le repère attaché à l'objet est défini, on pourra fixer dans quel demi-espace une caméra peut se trouver : nous avions choisi que la coordonnée Z des points sur l'objet valent 0. Donc, la caméra doit se trouver dans seulement un des demi-espaces qui correspondent à une coordonnée Z supérieure ou inférieure à 0. Des deux solutions pour la position de la caméra, la mauvaise pourra alors être rejetée (en regardant le signe de la troisième coordonnée du vecteur $\mathbf{t}$).

5.3 Bibliographie

- R.J. Holt et A.N. Netravali, *Camera Calibration Problem : Some New Results,* CVGIP - Computer Vision, Graphics and Image Processing, Vol. 54, No. 3, pp. 368-383, 1991.
- R.M. Haralick, C. Lee, K. Ottenberg et M. Nölle, *Analysis and Solutions of the Three Point Perspective Pose Estimation Problem,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 592-598, 1991.
- D. Dementhon et L.S. Davis, *Model-Based Object Pose in 25 Lines of Code,* International Journal on Computer Vision, Vol. 15, No. 1/2, pp. 123-141, 1995.
- P. Sturm, *Algorithms for Plane-Based Pose Estimation,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 706-711, 2000.

6 Relations géométriques entre deux images prises de points de vue différents – Géométrie épipolaire

6.1 Introduction

Rappelons-nous le scénario de création de mosaïques (cf. §3) et une observation qui a été faite : deux images prises du *même* point de vue, sont liées par une transformation projective (ou homographie). Étant donnée la projection d'un point dans l'une des images, cette transformation permet de déterminer où le point se projette dans l'autre image. Dès que les images sont prises de points de vue différents, il n'existe en général plus de telle transformation projective bijective. Donc, étant donné un point dans une image, on ne pourra plus déterminer la position exacte du point correspondant dans l'autre image. Pourtant, nous verrons qu'il est possible de déterminer une droite dans l'autre image, sur laquelle le point correspondant doit se trouver. Donc, il existe des relations géométriques entre deux images prises de points de vue différents (la position du point correspondant dans l'autre image n'est pas quelconque, mais contrainte par une droite), même si elles sont moins fortes que pour le cas de deux images prises du même point de vue.

6.2 Cas de base : rechercher le correspondant d'un point

Considérons deux images de la même scène, prises de points de vue différents. La question clef de cette section est : étant donné un point $\mathbf{q}_1$ dans la première image, que peut-on dire sur la position du point correspondant $\mathbf{q}_2$ dans la deuxième image ?

Premièrement, examinons ce que l'on peut dire sur la position du point 3-D $\mathbf{Q}$ original, qui a été projeté sur $\mathbf{q}_1$. Nous supposons que la caméra est calibrée, donc on peut tracer le rayon de projection (la droite passant par le centre de projection et le point $\mathbf{q}_1$ sur le plan image). Tout ce que l'on peut dire c'est que $\mathbf{Q}$ doit se trouver quelque part sur ce rayon de projection.

La projection du rayon de projection dans la deuxième image donne lieu à une droite $\mathbf{l}_2$. Puisque le point 3-D $\mathbf{Q}$ doit se trouver le long du rayon de projection, sa projection dans la deuxième image doit alors se trouver sur la droite $\mathbf{l}_2$ (voir la figure 6). Il existe donc une relation géométrique entre les deux images (le point $\mathbf{q}_2$ ne peut pas se trouver n'importe où dans la deuxième image, mais bien sur la droite $\mathbf{l}_2$), mais elle n'est pas « bijective » (tous les points sur $\mathbf{l}_2$ sont des correspondances possibles pour le point $\mathbf{q}_1$).

Le fait que le point correspondant doit se trouver sur une droite est souvent appelé la « contrainte épipolaire » (cette expression deviendra plus claire dans la suite). Cette contrainte est très utile pour la mise en correspondance d'images : la recherche de correspondances peut être accélérée considérablement (il ne faut plus chercher dans toute l'image pour trouver le point correspondant) et le risque d'erreur est réduit.

Dans la suite, nous regardons la contrainte épipolaire d'un point de vue plus général.

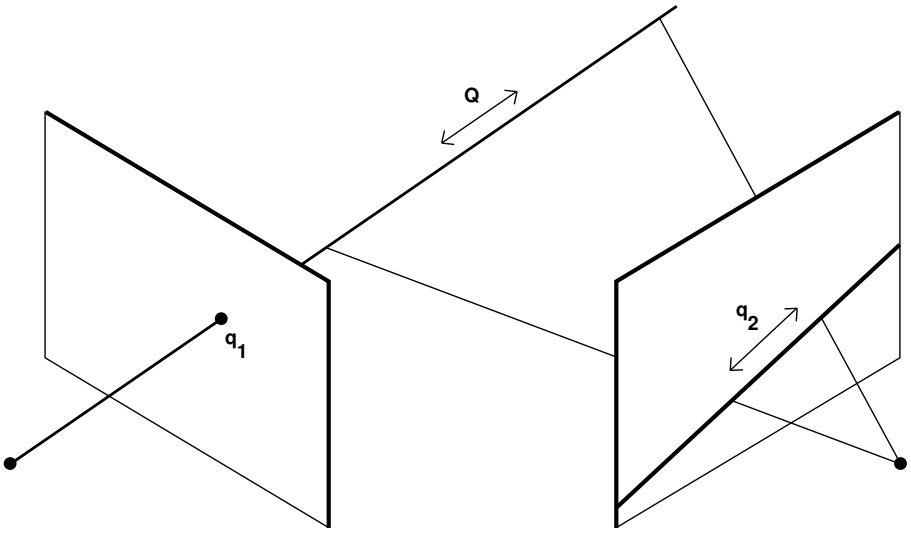


FIG. 6 – Géométrie épipolaire : le point 3-D se trouve sur le rayon de projection, le point $\mathbf{q}_2$ sur l'image de ce rayon.

6.3 La géométrie épipolaire

Avant de procéder, nous introduisons quelques notations (cf. les figures 6 et 7). Les centres de projection et le rayon de projection considéré dans le paragraphe précédent, définissent un plan que nous appellerons un *plan épipolaire*. Ce plan coupe les deux plans image en deux droites, les *droites épipolaires* – $\mathbf{l}_1$ dans la première image et $\mathbf{l}_2$ dans la deuxième. Le point dans la deuxième image qui correspond à $\mathbf{q}_1$ doit alors se trouver sur la *droite épipolaire associée* $\mathbf{l}_2$.

La configuration est parfaitement symétrique : le point dans la première image, qui correspond à un point $\mathbf{q}_2$ dans la deuxième image, se trouve sur la droite épipolaire associée, $\mathbf{l}_1$.

Considérons maintenant le cas où nous cherchons le correspondant d'un point $\mathbf{q}'_1$ qui se trouve sur la droite épipolaire $\mathbf{l}_1$ associée à $\mathbf{q}_1$. On peut observer que le plan épipolaire, ainsi que les deux droites épipolaires, sont les mêmes que ceux associés à $\mathbf{q}_1$. On peut en conclure que toutes les paires formées d'un point sur $\mathbf{l}_1$ et d'un autre sur $\mathbf{l}_2$, sont des correspondances possibles.

Considérons maintenant le cas d'un point $\mathbf{q}'_1$ qui ne se trouve pas sur la droite épipolaire $\mathbf{l}_1$ associée à $\mathbf{q}_1$. Il est clair que ceci donne lieu à un plan épipolaire différent. D'après la construction des plans épipolaires (définis par les deux centres de projection et un rayon), il est clair que ce deuxième plan contient la ligne qui joigne les deux centres de projection.

En répétant cette construction pour un nombre arbitraire de points, on peut alors « produire » le faisceau de plans épipolaires, qui est formé par tous les plans contenant les deux centres de projection. Au faisceau de plans épipolaires correspondent naturellement les deux faisceaux de droites épipolaires. La base du faisceau de plans épipolaires est la *ligne de base* – la droite qui lie les centres de projection. Quant aux faisceaux de droites épipolaires, leurs bases sont deux points – les **épipôles**.

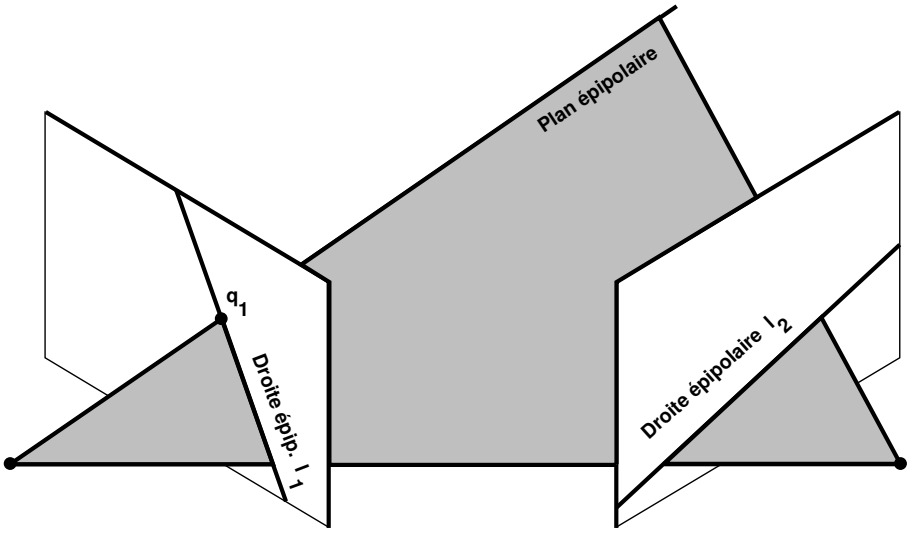


FIG. 7 – Les centres de projection et le rayon définissent un « plan épipolaire ». Ce plan coupe les plans image en les « droites épipolaires ».

En regardant la figure 8, on peut donner une autre définition pour les épipôles : l'épipôle de la première image – $\mathbf{e}_1$ – est l'image du centre de projection de la deuxième image. Réciproquement, $\mathbf{e}_2$ est l'image du premier centre de projection.

Remarque. Intuitivement, la connaissance des épipôles est utile pour l'estimation du placement relatif des deux caméras : par exemple, le premier épipôle $\mathbf{e}_1$ nous indique dans quelle direction, par rapport à la première caméra, se trouve le centre de projection de la deuxième caméra.

Examinons de plus près les deux faisceaux de droites épipolaires. On peut se les représenter comme deux espaces uni-dimensionnels avec comme éléments constituants les droites contenant l'épipôle respectif. Il existe une relation *bijective* entre ces deux espaces : une droite $\mathbf{l}_1$ du premier faisceau correspond à un seul plan épipolaire qui, en revanche, correspond à une seule droite épipolaire $\mathbf{l}_2$ dans la deuxième image. Nous appelons cette relation la **transformation épipolaire**.

La **géométrie épipolaire** peut alors être définie comme consistant de la position des deux épipôles ainsi que de la transformation épipolaire. Sa connaissance nous permet d'utiliser la contrainte épipolaire pour, comme esquissé plus haut, définir l'espace de recherche pour le point correspondant à $\mathbf{q}_1$: la connaissance de $\mathbf{e}_1$ nous permet de calculer la droite épipolaire $\mathbf{l}_1$ passant par $\mathbf{q}_1$. La transformation épipolaire ainsi que $\mathbf{e}_2$ nous fournissent alors la droite épipolaire $\mathbf{l}_2$ correspondante.

Dans la section suivante, nous examinons la modélisation algébrique de la géométrie épipolaire.

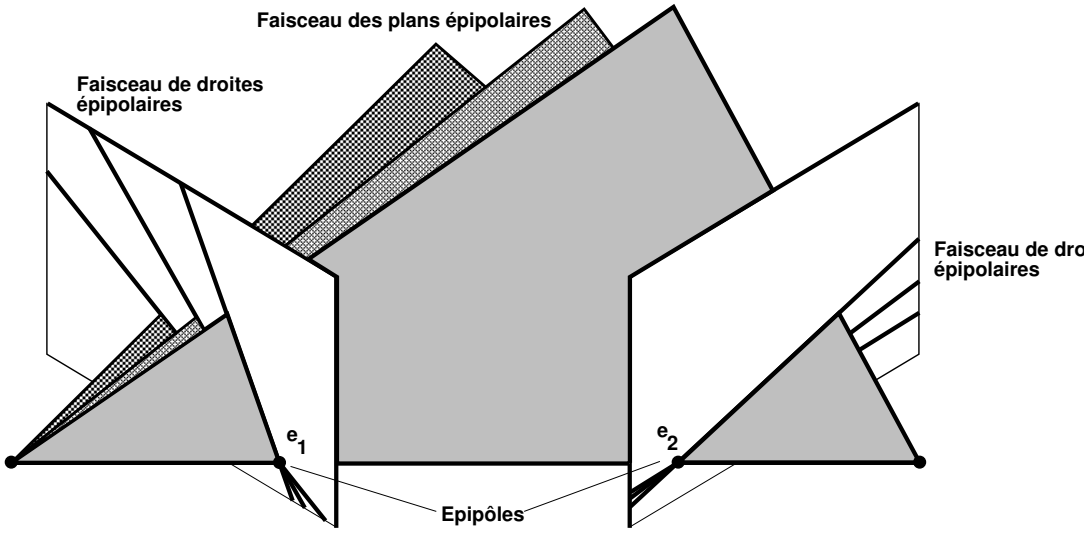


FIG. 8 – Tous les plans épipolaires contiennent les deux centres de projection. Donc, ils forment un faisceau de plans. Leurs intersections avec les plans image forment deux faisceaux de droites épipolaires. Les épipôles sont les points de base de ces deux faisceaux.

6.4 Représentation algébrique de la géométrie épipolaire – La matrice fondamentale

Soient P_1 et P_2 les matrices de projection des deux caméras :

$$\begin{aligned} \mathsf{P}_1 &\sim \mathsf{K}_1\mathsf{R}_1 \begin{pmatrix} \mathtt{I} & -\mathbf{t}_1 \end{pmatrix} \\ \mathsf{P}_2 &\sim \mathsf{K}_2\mathsf{R}_2 \begin{pmatrix} \mathtt{I} & -\mathbf{t}_2 \end{pmatrix} \end{aligned}$$

Étant donné un point $\mathbf{q}_1$ dans la première image, nous voulons calculer la droite épipolaire $\mathbf{l}_2$ associée, dans la deuxième image.

Nous procédons comme suit : d'abord, nous calculons le rayon de projection associé à $\mathbf{q}_1$; le rayon peut être représenté par deux points 3-D ; nous projetons ces points sur la deuxième image ; la droite $\mathbf{l}_2$ peut alors être donnée par le produit vectoriel des vecteurs de coordonnées des deux points image.

Le choix naturel pour le premier point sur le rayon de projection est le centre de projection de la première caméra :

$$\begin{pmatrix} \mathbf{t}_1 \\ 1 \end{pmatrix} .$$

Comme deuxième point nous choisissons de part son expression algébrique simple, le point à l'infini du rayon :

$$\begin{pmatrix} \mathsf{R}_1^{\mathsf{T}}\mathsf{K}_1^{-1}\mathbf{q}_1 \\ 0 \end{pmatrix}$$

(il est facile de vérifier que ce point est projeté sur $\mathbf{q}_1$ par P_1).

Ces deux points définissent entièrement le rayon de projection. Nous les projetons sur la deuxième image[6] :

$$\begin{aligned}\mathbf{a} &\sim \mathsf{P}_2 \begin{pmatrix}\mathbf{t}_1 \\ 1\end{pmatrix} \sim \mathsf{K}_2\mathsf{R}_2\left(\mathbf{t}_1 - \mathbf{t}_2\right) \\ \mathbf{b} &\sim \mathsf{P}_2 \begin{pmatrix}\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\mathbf{q}_1 \\ 0\end{pmatrix} \sim \mathsf{K}_2\mathsf{R}_2\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\mathbf{q}_1 \ .\end{aligned}$$

La droite épipolaire est alors donnée par le produit vectoriel $\mathbf{a} \times \mathbf{b}$. En utilisant la règle $(\mathsf{M}\mathbf{x}) \times (\mathsf{M}\mathbf{y}) \sim \mathsf{M}^{-\mathsf{T}}(\mathbf{x} \times \mathbf{y})$, nous obtenons :

$$\begin{aligned}\mathbf{l}_2 &\sim \mathbf{a} \times \mathbf{b} \\ &\sim \left\{\mathsf{K}_2\mathsf{R}_2\left(\mathbf{t}_1 - \mathbf{t}_2\right)\right\} \times \left\{\mathsf{K}_2\mathsf{R}_2\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\mathbf{q}_1\right\} \\ &\sim \left(\mathsf{K}_2\mathsf{R}_2\right)^{-\mathsf{T}}\left\{\left(\mathbf{t}_1 - \mathbf{t}_2\right) \times \left(\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\mathbf{q}_1\right)\right\} \ .\end{aligned}$$

En utilisant la notation $[\cdot]_\times$ (cf. §0.2), nous pouvons finalement écrire :

$$\begin{aligned}\mathbf{l}_2 &\sim \left(\mathsf{K}_2\mathsf{R}_2\right)^{-\mathsf{T}}\left[\mathbf{t}_1 - \mathbf{t}_2\right]_\times \left(\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\mathbf{q}_1\right) \\ &\sim \left(\mathsf{K}_2\mathsf{R}_2\right)^{-\mathsf{T}}\left[\mathbf{t}_1 - \mathbf{t}_2\right]_\times \left(\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\right)\mathbf{q}_1 \ . \qquad (22)\end{aligned}$$

L'équation (22) représente la transformation qui nous donne la droite épipolaire $\mathbf{l}_2$ associée au point $\mathbf{q}_1$, à partir des coordonnées de celui-ci. La matrice qui représente cette transformation est appelée la **matrice fondamentale** F_{12} :

$$\mathsf{F}_{12} \sim \left(\mathsf{K}_2\mathsf{R}_2\right)^{-\mathsf{T}}\left[\mathbf{t}_1 - \mathbf{t}_2\right]_\times \left(\mathsf{R}_1^\mathsf{T}\mathsf{K}_1^{-1}\right) \ . \qquad (23)$$

Elle représente entièrement la géométrie épipolaire (la transformation épipolaire et la position des épipôles peuvent en être extraites).

La matrice fondamentale est « orientée » – elle donne des droites épipolaires dans la deuxième image à partir de points dans la première image. Qu'en est-il pour le chemin inverse ? Si nous permutons les indices 1 et 2 dans l'équation (23), nous obtenons la matrice fondamentale qui donne des droites dans la première image, à partir de points dans la deuxième :

$$\mathsf{F}_{21} \sim \left(\mathsf{K}_1\mathsf{R}_1\right)^{-\mathsf{T}}\left[\mathbf{t}_2 - \mathbf{t}_1\right]_\times \left(\mathsf{R}_2^\mathsf{T}\mathsf{K}_2^{-1}\right) \ .$$

Nous pouvons observer[7] que F_{21} n'est rien d'autre que la transposée de F_{12} (éventuellement à un facteur scalaire près) :

$$\mathsf{F}_{21} \sim \mathsf{F}_{12}^\mathsf{T} \ .$$

Dans la suite, nous allons simplement écrire F, en sousentendant que la « direction » de la transformation découle du contexte.

[6] Notons que le point $\mathbf{a}$ coïncide en fait avec l'épipôle $\mathbf{e}_2$.

[7] En utilisant le fait que, pour tout $\mathbf{x}$: $[\mathbf{x}]_\times \sim \left([\mathbf{x}]_\times\right)^\mathsf{T}$.

Nous pouvons maintenant exprimer la contrainte épipolaire évoquée au §6.2. Le point $\mathbf{q}_2$ correspondant à $\mathbf{q}_1$ doit se trouver sur la droite épipolaire $\mathbf{l}_2$, c'est-à-dire que le produit scalaire de $\mathbf{q}_2$ et $\mathbf{l}_2$ est nul, ou bien $\mathbf{q}_2^\mathsf{T}\mathbf{l}_2 = 0$. En remplaçant la définition de $\mathbf{l}_2$, nous obtenons finalement la **contrainte épipolaire** :

$$\mathbf{q}_2^\mathsf{T}\mathsf{F}\mathbf{q}_1 = 0 \ . \tag{24}$$

Remarque. La connaissance de la matrice fondamentale (i.e. la géométrie épipolaire) associée à deux images, nous permet donc de formuler la contrainte épipolaire afin de simplifier la mise en correspondance de points. Réciproquement, étant donné des correspondances de points, l'équation (24) donne des contraintes permettant de calculer la matrice fondamentale (ce qui sera utile, comme on verra plus tard, pour estimer le mouvement entre les deux caméras) !

6.5 Quelques détails sur la matrice fondamentale

La matrice fondamentale est une transformation entre points et droites, ce qui est souvent appelée une *corrélation*.

La propriété algébrique principale de la matrice fondamentale est qu'elle est singulière, ce qui est équivalent au fait que son déterminant est nul ou que son rang est inférieur à 3. Il y a plusieurs manières de constater ceci – nous en expliquons deux. Premièrement, la matrice $[\mathbf{t}_1 - \mathbf{t}_2]_\times$ dans l'équation (23) est singulière (toute matrice anti-symétrique l'est) et par conséquent, la matrice fondamentale l'est aussi (le déterminant d'un produit de matrices carrées équivaut au produit des déterminants).

Le deuxième argument découle de l'observation de la figure 8. La matrice fondamentale est une fonction entre deux espaces bi-dimensionnels – l'espace des points dans la première image et l'espace des droites dans la deuxième. Pourtant, seul le faisceau de droites épipolaires – un espace uni-dimensionnel – est atteint. La fonction n'est pas bijective ce qui se manifeste par la singularité de la matrice fondamentale.

Une matrice singulière a un noyau, c'est-à-dire qu'il existe des vecteurs non-nuls pour qui le produit matrice-vecteur donne le vecteur nul. Le noyau de la matrice fondamentale n'est rien d'autre que l'épipôle $\mathbf{e}_1$, i.e. nous avons :

$$\mathsf{F}\mathbf{e}_1 = \mathbf{0} \ .$$

Ceci peut être prouvé de manière analytique, mais il y a aussi une explication intuitive : la droite épipolaire associée à l'épipôle n'est pas définie, ce qui correspond bien au fait que le vecteur nul $\mathbf{0} = \mathsf{F}\mathbf{e}_1$ n'est pas admis pour représenter des points ou des droites en coordonnées homogènes.

Pour le deuxième épipôle il existe une relation analogue à celle donnée ci-dessus :

$$\mathsf{F}^\mathsf{T}\mathbf{e}_2 = \mathbf{0} \ .$$

6.6 Géométrie épipolaire calibrée et matrice essentielle

La matrice fondamentale dépend du positionnement des deux caméras ainsi que de leurs paramètres intrinsèques, c'est-à-dire de leurs matrices de calibrage K_1 et K_2. Dans cette section, nous supposons que les caméras sont calibrées, c'est-à-dire que nous connaissons K_1 et K_2. Nous pouvons donc, à partir de la matrice fondamentale, calculer la **matrice essentielle** E (cf. l'équation (23)) :

$$\mathsf{E} \sim \mathsf{K}_2^\mathsf{T} \mathsf{F} \mathsf{K}_1 \sim \mathsf{R}_2 \left[\mathbf{t}_1 - \mathbf{t}_2\right]_\times \mathsf{R}_1^\mathsf{T} \ . \tag{25}$$

Que représente la matrice essentielle ? En effet, la matrice fondamentale, elle, représente la géométrie épipolaire de deux images si les repères utilisés pour les représenter sont les repères pixels. La matrice essentielle, quant à elle, représente également la géométrie épipolaire, mais cette fois-ci exprimée par rapport aux repères image[8]. On dit souvent que la matrice essentielle représente la *géométrie épipolaire calibrée*, puisque c'est l'information sur le calibrage des caméras qui permet de remonter aux repères image.

On constate (cf. l'équation (25)) que la matrice essentielle ne dépend que de la position et de l'orientation des caméras, d'où son utilité pour l'estimation du mouvement d'une caméra, respectivement du placement relatif de deux caméras, ce qui sera traité au §7.2.

Remarque. Au paragraphe précédent nous avons vu que les matrices fondamentales sont caractérisées par leur singularité. Ce constat s'applique également aux matrices essentielles. Outre la singularité, une matrice essentielle « valide » doit posséder une autre propriété : ses deux valeurs singulières non nulles sont identiques.

6.7 Estimation de la géométrie épipolaire – Méthode de base

Dans la section 6.4, nous avons vu comment calculer la matrice fondamentale, donc la représentation algébrique de la géométrie épipolaire, à partir du positionnement et des paramètres intrinsèques de deux caméras en question. Un des usages de la géométrie épipolaire, discuté précédemment, consiste à contraindre la mise en correspondance des images. Réciproquement, la connaissance de suffisamment de correspondances permet d'estimer la géométrie épipolaire. Ceci est très important en pratique, où l'on ne connaît souvent pas le positionnement des caméras.

Dans cette section, nous traitons de la méthode de base pour le calcul de la matrice fondamentale, à partir de correspondances de points. Dans la section suivante, une extension de la méthode est présentée, qui est conçue pour des correspondances peu fiables, ce qui est quasiment toujours le cas en pratique.

La méthode de base s'appuie sur l'équation (24) :

$$\mathbf{q}_2^\mathsf{T} \, \mathsf{F} \, \mathbf{q}_1 = 0 \ .$$

[8] Cf. §1.2 ; pour être exact, il s'agit des repères image, ayant subi un changement d'unité tel que l'unité de base équivaut à la distance focale.

En explicitant les coefficients de la matrice fondamentale F, ceci s'écrit comme :

$$\begin{aligned} & F_{11}q_{1,1}q_{2,1} + F_{12}q_{1,2}q_{2,1} + F_{13}q_{1,3}q_{2,1} \\ +& F_{21}q_{1,1}q_{2,2} + F_{22}q_{1,2}q_{2,2} + F_{23}q_{1,3}q_{2,2} \\ +& F_{31}q_{1,1}q_{2,3} + F_{32}q_{1,2}q_{2,3} + F_{33}q_{1,3}q_{2,3} \quad = \quad 0 \ . \end{aligned}$$

Cette équation est linéaire en les coefficients de F. Si l'on prend en compte n correspondances de points, chacune donne une équation du type ci-dessus. Toutes ces équations peuvent être regroupées en un système matriciel :

$$\mathsf{A}\mathbf{f} = \mathbf{0} \ . \tag{26}$$

avec un vecteur $\mathbf{f}$ contenant les coefficients de F (les inconnues) :

$$\mathbf{f} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^{\mathsf{T}} \ ,$$

et la matrice A de dimension $n \times 9$ qui est de la forme :

$$\mathsf{A} = \begin{pmatrix} q_{1,1}q_{2,1} & q_{1,2}q_{2,1} & q_{1,3}q_{2,1} & q_{1,1}q_{2,2} & q_{1,2}q_{2,2} & q_{1,3}q_{2,2} & q_{1,1}q_{2,3} & q_{1,2}q_{2,3} & q_{1,3}q_{2,3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}_{n\times 9} \ .$$

A cause du bruit dans les données, il n'y a en général pas de solution exacte pour l'équation (26), donc on résoud le système aux moindres carrés, c'est-à-dire on détermine $\hat{\mathbf{f}}$ tel que :

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \|\mathsf{A}\mathbf{f}\|^2 \quad \text{sous la contrainte} \ \ \|\mathbf{f}\| = 1 \ .$$

Pour plus de détails sur la méthode de résolution, nous renvoyons au poly du cours « Optimisation » du M2R IVR 2005/06, qui sera distribué.

Remarque. La notation $\|\mathbf{v}\|$ fait référence à la norme du vecteur $\mathbf{v}$, i.e. la racine de la somme des carrés de ses coefficients. Pour être exacte, il s'agit de la norme L_2. Il y a beaucoup d'autres normes de vecteurs, mais nous sousentendons qu'il s'agit de la norme L_2, à défaut d'une spécification dans le contexte.

6.7.1 Un petit problème...

Cette méthode néglige une contrainte sur la structure de la matrice fondamentale : dans la section 6.5, il a été mentionné que toute matrice fondamentale « valide » (c'est-à-dire qui représente effectivement la géométrie épipolaire d'une paire de caméras), doit être singulière. Or, cette contrainte (non linéaire) n'est pas prise en compte par la méthode proposée, et en présence de bruit dans les données, elle ne sera pas vérifiée par la solution.

Il y a un remède à ce problème : étant donnée une matrice F, qui n'est pas singulière, il y a un moyen d'estimer la matrice $\hat{\mathsf{F}}'$ qui est parfaitement singulière et qui approche au mieux F, c'est-à-dire qui minimise la somme des carrés des différences des coefficients :

$$\hat{\mathsf{F}}' = \arg\min_{\mathsf{F}'} \sum_{i,j=1}^{3} \left(F'_{ij} - F_{ij}\right)^2 \quad \text{sous la contrainte que} \ \ \mathsf{F}' \ \ \text{soit singulière} \ .$$

Remarque. Ce problème est résolu à l'aide de la décomposition en valeurs singulières. Supposons que l'on veut déterminer, pour une matrice A quelconque, la matrice B qui soit de rang r et qui soit le plus proche possible de A, au sens de la somme des carrés des différences de leurs coefficients. Pour ce faire, considérons la décomposition en valeurs singulières (voir le cours « Optimisation ») de $\mathsf{A} : \mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$, où les éléments de la matrice diagonale Σ sont : $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$. Alors, B est donnée par : $\mathsf{B} = \mathsf{U}\Sigma'\mathsf{V}^\mathsf{T}$, où Σ' est la matrice diagonale définie par :

$$\sigma'_i = \begin{cases} \sigma_i & \text{si } i \leq r \\ 0 & \text{si } i > r \end{cases}$$

Pour ce qui est de notre cas, la matrice fondamentale est une matrice 3×3 singulière, donc de rang 2 au plus (ce qui veut dire que $r = 2$).

Remarque. La *norme de Frobenius* est définie comme étant la racine de la somme des carrés des coefficients d'une matrice :

$$\|\mathsf{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2} \ .$$

La norme de Frobenius est pour les matrices ce qui est la norme L_2 pour les vecteurs.

Dans cette section, nous avons considéré la somme des carrés des différences des coefficients de deux matrices A et B. Il s'agit alors du carré de la norme de Frobenius de la « matrice de différence », $\mathsf{A} - \mathsf{B}$.

6.7.2 Combien de correspondances faut-il avoir ?

L'équation (26) est un système d'équations *linéaire* et *homogène* (sur le côté droit se trouve un vecteur nul). De manière générale, en présence de m inconnues, il y a une solution unique (à l'échelle près) si l'on dispose de $m - 1$ équations. Avec moins d'équations, le système est sous-contraint[9]. Si plus d'équations sont disponibles, il n'y a en général, en présence de bruit dans les données, pas de solution exacte, c'est pourquoi la méthode des moindres carrés est appliquée.

Dans notre cas alors, chaque correspondance de points fournit une seule équation, donc il nous faut a priori un minimum de 8 correspondances pour estimer la géométrie épipolaire. C'est la raison pourquoi cet algorithme est typiquement appelée la *méthode des 8 points* (8 point method) dans la littérature.

Remarque. Grâce au fait que la matrice fondamentale doit être singulière, nous pouvons en fait l'estimer avec seulement 7 correspondances de points. Avec les 7 équations linéaires et homogènes sur les 9 inconnues (qui sont définies à un facteur multiplicatif près), il existe une famille de solutions pour la matrice fondamentale qui peut être exprimée par la combinaison linéaire de deux matrices de base :

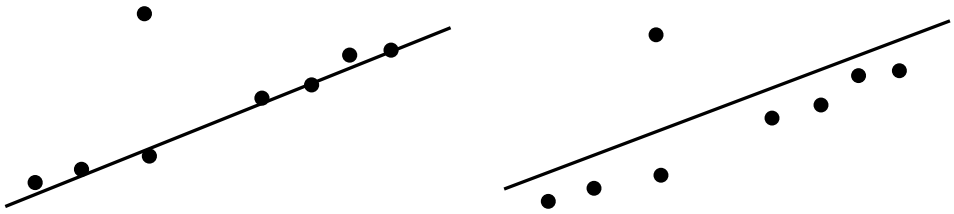
$$\mathsf{F} = \mathsf{F}_1 + \lambda F_2$$

La matrice F étant singulière se traduit par le fait que son déterminant est nul. Le déterminant de la matrice ci-dessus, est un polynôme cubique en λ. Donc, au lieu d'avoir une infinité de solutions, il n'en restent plus que 3 (correspondant aux 3 racines du polynôme).

[9] Il existe alors un espace de solutions qui est linéaire, c'est-à-dire que toutes les solutions peuvent être exprimées par des combinaisons linéaires de vecteurs de base.

6.8 Estimation robuste de la géométrie épipolaire

Dans cette section, nous rencontrons le principe d'*estimation robuste* qui est vraiment essentiel pour faire marcher une grande partie des algorithmes en vision par ordinateur en conditions réalistes. Par conditions réalistes nous entendons bien entendu la présence du bruit dans les données (e.g. localisation imprécise de points dans les images), mais aussi, et surtout, le problème des *erreurs grossières* (*outliers*). La figure 9 illustre ce problème à l'aide du problème de l'estimation d'une droite qui approche au mieux un ensemble de points. Comme on peut le voir, une seule erreur grossière peut avoir une influence « catastrophique » sur les résultats.



(a) C'est le résultat qu'on voudrait avoir : le point est correctement identifié comme étant une erreur grossière (il peut, par exemple, être le résultat d'une erreur de mesure).

(b) C'est le résultat que l'on obtient si les erreurs grossières ne sont pas identifiées et gérées.

FIG. 9 – Illustration de l'influence d'erreurs grossières. Le but est d'estimer la droite qui approche au mieux les points.

Intuitivement, il ne devrait pas être trop difficile de gérer ce petit exemple. Pourtant, on rencontre souvent des problèmes d'estimation avec beaucoup de données, dont beaucoup sont des erreurs grossières. Aussi, dans l'exemple montré, l'estimation des inconnues se passe dans un espace bi-dimensionnel (une droite dans un plan a 2 paramètres). Le problème peut devenir nettement plus difficile pour plus de dimensions ou s'il y a des relations complexes entre les inconnues.

Nous expliquons dans la suite le principe de base pour l'une des méthodes d'*estimation robuste*, c'est-à-dire d'estimation de paramètres, tout en identifiant des erreurs grossières. Elle est basée sur l'utilisation de mesures dites « robustes », pour la qualité d'une estimation courante des paramètres. Deux possibilités pour le cas illustré sur la figure 9, c'est-à-dire des mesures pour la « qualité » d'une droite hypothétique, sont :

- le nombre des points qui se trouvent à moins d'une distance donnée (un seuil) de la droite ;
- la médiane des distances des points de la droite.

La première mesure est plus simple à calculer, mais le choix du seuil peut en général être délicat. La deuxième mesure, quant à elle, ne necéssite pas de seuil. Pourtant, dans des cas où plus de 50 % des données sont des erreurs grossières, il faudrait utiliser une médiane généralisée, ce qui revient à devoir connaître approximativement le taux d'erreurs grossières. La première mesure, elle, « marchera » même en présence de plus de 50 % de données erronnées.

Remarque. Le contraire des erreurs grossières (*outliers*) sont les *inliers* (l'expression anglaise est souvent utilisée même en français). C'est-à-dire, pour l'exemple donné, les points qui sont proches de la droite.

Revenons à notre application, l'estimation de la géométrie épipolaire. Les erreurs grossières sont ici les correspondances incorrectes : si deux points sont mis en correspondance bien qu'ils ne soient pas les projections du même point 3-D, leur prise en compte rend en général la solution obtenue pour la matrice fondamentale totalement inutilisable.

Passons maintenant à comment mesurer la qualité d'une matrice fondamentale donnée. Ce qui est le plus souvent utilisé est basé sur la « distance épipolaire » : la matrice fondamentale permet de tracer les droites épipolaires associées aux points donnés. Si la correspondance de deux points $\mathbf{q}_1$ et $\mathbf{q}_2$ est correcte, alors $\mathbf{q}_2$ sera proche de la droite épipolaire associée à $\mathbf{q}_1$, et vice versa. On pourra alors, de manière analogue aux mesures évoquées ci-dessus, compter le nombre de correspondances dont la distance épipolaire est en-dessous d'un seuil donné, ou bien calculer la médiane de toutes les distances épipolaires.

Nous disposons désormais de moyens pour évaluer des hypothèses sur la matrice fondamentale. La question qui reste est comment obtenir des hypothèses. L'idée de base est très simple : il faut, selon la méthode utilisée, 7 ou 8 correspondances de points, pour estimer la matrice fondamentale (cf. §6.7.2). Tout ce que l'on fait alors est d'effectuer un certain nombre de tirages aléatoires de 7 ou 8 correspondances dans l'ensemble des correspondances données. Pour chacun des échantillons de correspondances, on estime la matrice fondamentale, puis on l'évalue à l'aide de toutes les autres correspondances, en utilisant une des mesures données ci-dessus. Si suffisamment de tirages sont effectués, on obtiendra, avec une certaine probabilité, au moins un échantillon qui ne contient que des correspondances correctes. En général, seuls les échantillons ne contenant que des correspondances correctes, recevront un bon score lors de l'évaluation. On retiendra alors l'estimation de la matrice fondamentale qui a reçu le plus grand score.

Une fois que l'on dispose d'une bonne estimation de la matrice fondmentale, on pourra l'améliorer en ne prenant en compte que les correspondances correctes.

Le processus entier est illustré sur la figure 10, pour le cas de l'estimation d'une droite.

Une dernière chose reste à préciser : combien de tirages aléatoires faut-il effectuer afin d'être sûr d'obtenir au moins un bon échantillon ? Notons tout de suite qu'une certitude absolue n'est pas possible sans effectuer un échantillonnage exhaustif, ce qui, même pour de « petits » problèmes, n'est typiquement pas applicable.

Etant donné le taux d'erreurs grossières parmi les données, des formules des statistiques élémentaires permettent de calculer la probabilité de trouver au moins un bon échantillon, si n tirages sont effectués (cette probabilité dépend aussi, et fortement, du nombre de données qui constituent un échantillon). Le taux d'erreurs grossières n'est évidemment pas connu à l'avance, mais pour beaucoup de problèmes, des valeurs empiriques sont le plus souvent utilisables. Par exemple, des méthodes de mise en correspondance (non contrainte par la géométrie épipolaire, puisque celle-ci n'est pas encore connue) fournissent typiquement jusqu'à une vingtaine ou trentaine de pourcent de mauvaises correspondances entre deux images.

Si g est la fraction supposée de « bonnes » données, s la taille des échantillons et x le nombre de tirages aléatoires, alors la probabilité de trouver au moins un bon échantillon, est

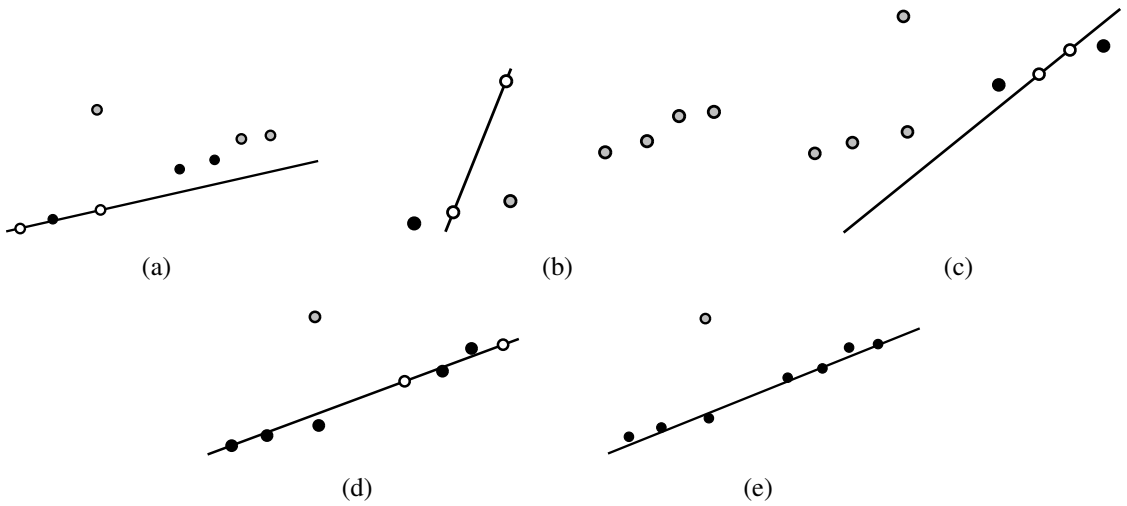


FIG. 10 – Illustration du processus d'estimation robuste. A chaque itération, deux points sont tirés au hasard (montrés en blanc), qui définissent une droite. Les erreurs grossières par rapport à la droite actuelle sont montrées en gris, les points « valides » (les *inliers*), en noir. (a) – (c) des échantillons qui ne seront pas retenus. (d) le meilleur echantillon. (e) le résultat de l'estimation de la droite, prenant en compte tous les inliers du cas (d).

approximativement :

$$1 - (1 - g^s)^x$$

Le nombre de tirages aléatoires qu'il faut effectuer pour avoir la probabilité α de trouver au moins un échantillon bon, est alors :

$$x \geq \frac{ln(1-\alpha)}{ln(1-g^s)}$$

La figure 11 montre quelques exemples de valeurs concernant l'estimation de la géométrie épipolaire (ici, $s = 8$).

Remarque. Nous avons décrit, dans cette section, le principe de base de méthodes d'estimation robuste. Ces méthodes peuvent être appliquées à tous les problèmes où les données sont affectées par des erreurs grossières. Leur concept est effectivement relativement simple ; il faut pourtant prendre en compte une croissance parfois significative en temps de calcul par rapport à des méthodes non robustes. Par contre, ces dernières échouent en général si les données sont telles que décrites.

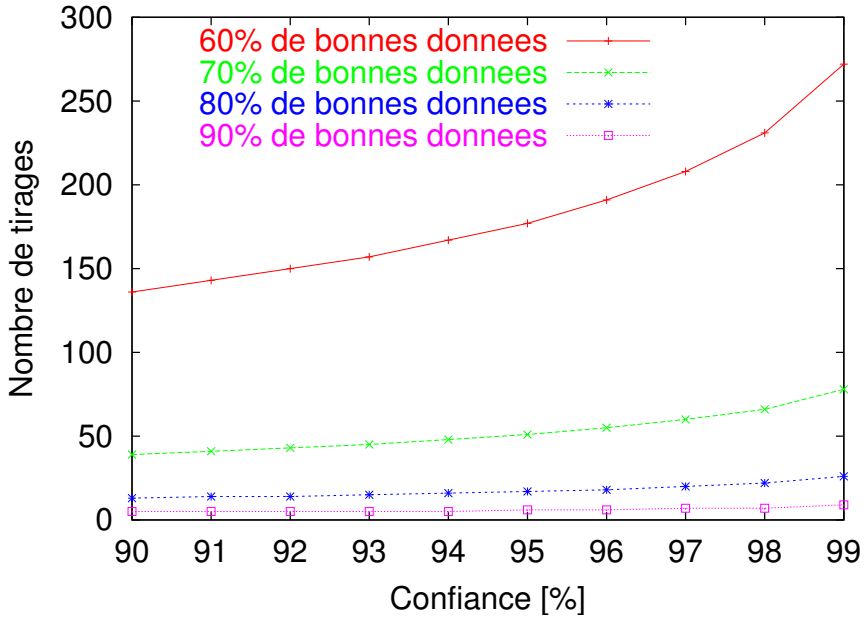


FIG. 11 – Nombre de tirages necéssaires pour atteindre la probabilité α d'avoir au moins un bon échantillon de 8 correspondances, en dépendance du taux de bonnes correspondances.

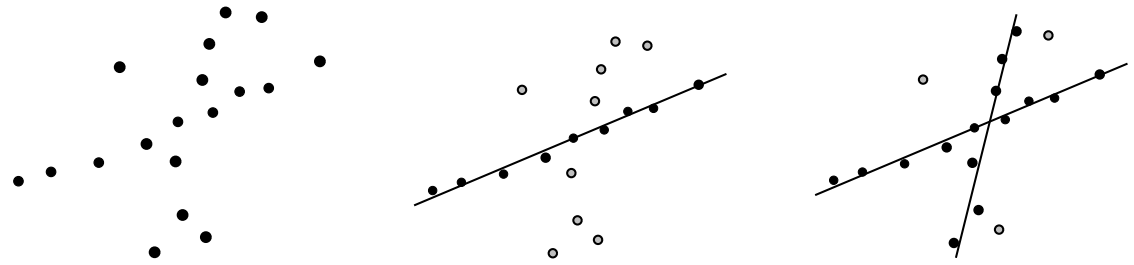
7 Estimation et segmentation de mouvements

7.1 Une méthode de base pour la segmentation de mouvements

Considérons une scène qui contient des objets qui bougent indépendamment les uns des autres (dans des directions différentes, avec des vélocités différentes, etc.). Le but de cette section est de délinéer, dans des images prises par une caméra (qui elle peut aussi bouger), les différents objets. Ceci sera fait en regroupant les points qui suivent le même mouvement (et qui seront considérés ici, un peu naïvement, comme constituant des objets). On parle alors aussi de *segmentation de mouvements*.

Nous illustrons le principe de la méthode à l'aide de l'exemple commencé plus haut, sur l'estimation d'une droite à partir de points. Considérons maintenant la figure 12 (a). L'estimation robuste de la droite donnera probablement un résultat comme celui montré sur la figure 12 (b). Il s'agit normalement de la droite « dominante ». Afin de détecter la deuxième droite, on relancera alors l'estimation robuste, mais uniquement sur les points qui sont considérés comme des erreurs grossières par rapport à la première droite. Un résultat possible est montré sur la figure 12 (c).

On pourra réitérer le processus, en ne considérant plus que les points étant des outliers pour les deux droites trouvées. Dans notre exemple pourtant, il s'agit de points quelconques et plus aucune droite recevant un « support » de la part des points, sera trouvée.



(a) Les données.

(b) Résultat de l'estimation robuste.

(c) Résultat de l'estimation robuste sur les outliers de la première estimation.

FIG. 12 – Récursion de l'estimation robuste, pour trouver plusieurs droites.

Revenons à notre application, la segmentation de mouvements. La clef de la méthode que nous allons décrire est le fait que des objets qui bougent différemment, donnent lieu à différentes géométries épipolaires entre des paires d'images. Regardons la figure 13 : à gauche est montré le cas d'une caméra immobile qui prend deux images à des instants de temps successifs, d'une scène mobile. Les mêmes images seraient créées par une caméra mobile, se déplaçant dans le sens inverse, si cette fois-ci c'est la scène qui est immobile. La deuxième interprétation correspond alors au cas de deux caméras, placées à différents endroits, et l'on peut définir leur géométrie épipolaire. Les points dans les deux images satisferont cette géométrie épipolaire (pour chaque point de la première image, le point correspondant dans la deuxième image se trouve sur la droite épipolaire associée).

Que se passe-t-il si la scène contient un deuxième objet, qui bouge différemment par rapport au premier ? La deuxième caméra (virtuelle) se déplaçant dans le sens inverse par rapport au mouvement de l'objet, se trouvera alors à une autre position, comparé à celle associée au mouvement du premier objet. Donc, une autre géométrie épipolaire y sera associée.

Remarque. La géométrie épipolaire, comme nous l'avons introduite au chapitre précédent, caractérise le positionnement relatif de deux caméras. Elle peut être estimée à partir de correspondances entre deux images. Jusqu'ici, abstraction a été faite de l'aspect temporel, c'est-à-dire il n'a pas été spécifié si les deux caméras prennent les images en même temps ou pas.

Dans cette section, nous considérons le cas d'une caméra en mouvement, ou bien de deux caméras, mais qui prennent des images à des instants différents. Si la scène a bougé entretemps, on peut alors définir plusieurs géométries épipolaires, chacune d'entre elles caractérisant en effet le positionnement relatif des deux caméras *et* d'un objet, bougeant indépendamment du reste de la scène.

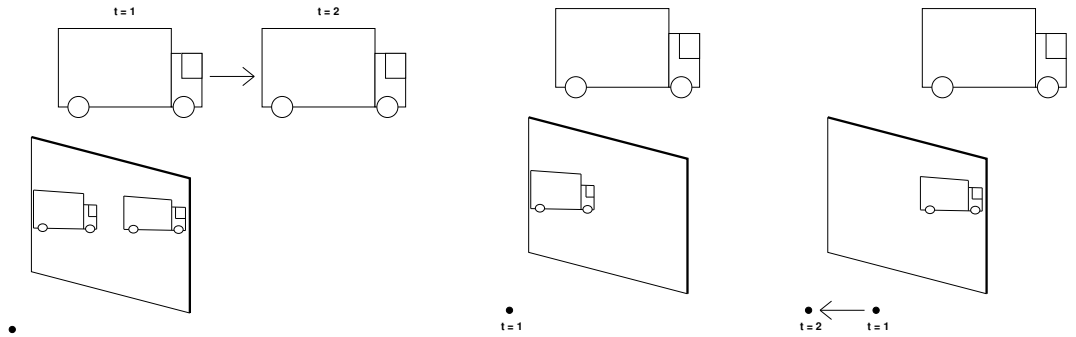


FIG. 13 – A gauche, une caméra statique qui prend deux images d'une scène dynamique. Ceci est équivalent à deux images prises par une caméra mobile, d'une scène statique.

Le fait que chaque objet bougeant indépendamment de la scène donne en général lieu à une géométrie épipolaire particulière, nous donne le moyen de concevoir une méthode de segmentation de mouvements. La méthode suit le même schéma que pour la détection de droites, décrite ci-dessus. Les données de départ sont des correspondances de points dans deux images. La méthode robuste du §6.8 donnera la géométrie épipolaire « dominante », qui correspondra normalement à l'objet pour lequel le plus de correspondances sont disponibles. L'application récursive de la méthode, sur les outliers des géométries épipolaires trouvées, permet alors en principe de détecter tous les objets (ou bien, mouvements) de la scène.

La figure 14 montre un exemple. La scène contient deux objets bougeant différemment : le camion et le reste de la scène (la caméra, elle, bouge aussi, donc le reste de la scène bouge par rapport à la caméra).

7.2 Estimation du mouvement à partir de la matrice essentielle

Nous regardons comment extraire, à partir de la matrice essentielle, les paramètres extrinsèques des deux caméras. On parle aussi d'*estimation du mouvement*, puisque l'application principale concerne une seule caméra qui prend des images pendant qu'elle se déplace (par exemple, une caméra montée sur un véhicule).

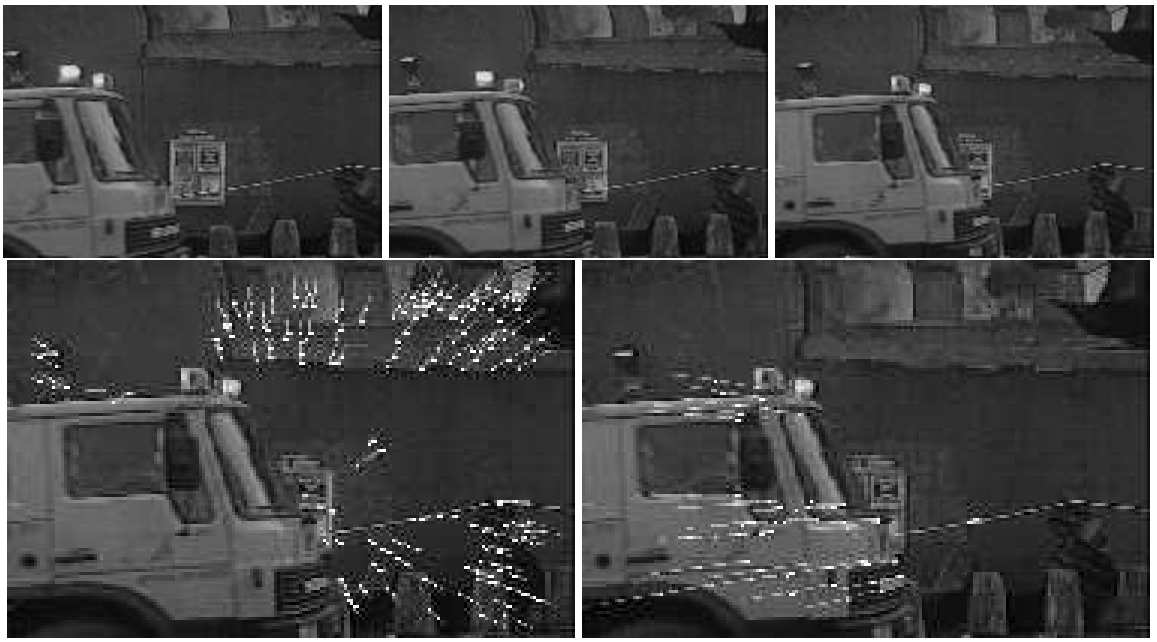


FIG. 14 – Trois images d'une séquence : le camion se dirige vers la droite, la caméra recule devant la scène. Les deux images en bas montrent les deux mouvements apparents qui ont été segmentés : fond et camion. Images prises de : P. Torr, « Motion Segmentation and Outlier Detection », Thèse de Doctorat, Université d'Oxford, 1995.

Considérons l'équation (25) définissant la matrice essentielle. Les paramètres extrinsèques $\mathsf{R}_1, \mathsf{R}_2, \mathbf{t}_1$ et $\mathbf{t}_2$ sont définis par rapport à un repère monde, c'est-à-dire qu'ils expriment le positionnement *absolu* des deux prises d'image. Pourtant, lors de l'estimation du mouvement à partir de seules deux images, tout ce qui peut être déterminé est le positionnement *relatif* (donc, le mouvement). On pourra donc supposer, sans perte de généralité, que la première caméra est en « position canonique », i.e. :

$$\begin{aligned} \mathsf{R}_1 &= \mathtt{I} \\ \mathbf{t}_1 &= \mathbf{0} \ . \end{aligned}$$

L'équation (25) devient alors :

$$\mathsf{E} \sim \mathsf{R} \left[-\mathbf{t} \right]_\times \quad ,$$

où R et $\mathbf{t}$ expriment l'orientation et la position de la deuxième caméra, relativement à la première (donc, le mouvement). En appliquant le fait que $\left[-\mathbf{v} \right]_\times = - \left[\mathbf{v} \right]_\times$ pour tout vecteur $\mathbf{v}$, l'équation ci-dessus peut encore être simplifiée un peu :

$$\mathsf{E} \sim \mathsf{R} \left[\mathbf{t} \right]_\times \quad . \tag{27}$$

A l'aide de cette équation, on trouve directement un moyen pour calculer $\mathbf{t}$. Notamment, $\mathbf{t}$ est effectivement un vecteur du noyau de E. Ceci parce que : $\left[\mathbf{t} \right]_\times \mathbf{t} \sim \mathbf{t} \times \mathbf{t} = \mathbf{0}$, donc : $\mathsf{E}\mathbf{t} = \mathbf{0}$. Une matrice essentielle étant singulière, le noyau existe et peut être calculé. Pourtant, ceci

nous donne $\mathbf{t}$ seulement à un facteur d'échelle près (si un vecteur est dans le noyau, alors tous ses « multiples » le sont aussi).

Ceci n'est pas gênant en fin de compte : l'échelle de $\mathbf{t}$, donc la distance entre les deux caméras, ne pourra de toute façon pas être calculée de manière absolue, puisque deux images d'une scène « large », prises par des caméras distantes, ne peuvent être distinguées de deux images d'une scène petite, prises par des caméras rapprochées. Tout ce que l'on peut obtenir, est donc la *direction* du déplacement, non son étendue ! Par contre, si l'estimation du mouvement est faite pour des paires d'images successives, on pourra estimer les étendues relatives des déplacements : le mouvement d'une caméra associé à toute une séquence d'images peut alors être estimé correctement, à un seul facteur d'échelle près, ce qui n'est pas si mal !

Le calcul de la matrice de rotation R est un peu plus compliqué que celui de $\mathbf{t}$, même si, ayant calculé le vecteur $\mathbf{t}$, l'équation (27) permettrait d'établir un système d'équations linéaires.

Dans la suite, nous donnons donc une méthode simple à mettre en œuvre pour calculer R et $\mathbf{t}$, basée sur la décomposition en valeurs singulières de E. Soit cette dernière donnée par :

$$\mathsf{E} \sim \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}} \quad .$$

Si E a été obtenue à partir de données bruitées, elle ne satisfera en général pas tous les critères d'une matrice essentielle (une valeur singulière nulle, les deux autres égales entre elles, cf. §6.6). Dans ce cas, la matrice essentielle « parfaite » qui est le plus proche possible de E est donnée par[10] :

$$\hat{\mathsf{E}} = \mathsf{U}\ \mathrm{diag}(1,1,0)\ \mathsf{V}^{\mathsf{T}} \quad .$$

Le vecteur $\mathbf{t}$ est alors donné par la troisième colonne de V (il est facile de vérifier que $\hat{\mathsf{E}}\mathbf{t}$ donne alors le vecteur nul). Quant à la matrice de rotation, il existe deux solutions :

$$\mathsf{R} = \mathsf{U} \begin{pmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathsf{V}^{\mathsf{T}} \quad .$$

Si $\det \mathsf{R} = -1$, multiplier R avec -1.

Il est facile de vérifier qu'il s'agit effectivement de matrices de rotation (multiplier les matrices avec leurs transposées ; le résultat doit être l'identité).

Esquisse de preuve. Dans la suite, nous esquissons comment vérifier si ces solutions correspondent effectivement à l'équation (27). Notons

$$\mathsf{V} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{pmatrix} \quad .$$

Donc, nous avons : $\mathbf{t} = \mathbf{v}_3$. L'équation suivante doit alors être vérifiée :

$$\mathsf{R}\,[\mathbf{t}]_\times \sim \hat{\mathsf{E}}$$

ou bien :

$$\mathsf{U} \begin{pmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^{\mathsf{T}} \\ \mathbf{v}_2^{\mathsf{T}} \\ \mathbf{v}_3^{\mathsf{T}} \end{pmatrix} [\mathbf{v}_3]_\times \sim \mathsf{U}\ \mathrm{diag}(1,1,0) \begin{pmatrix} \mathbf{v}_1^{\mathsf{T}} \\ \mathbf{v}_2^{\mathsf{T}} \\ \mathbf{v}_3^{\mathsf{T}} \end{pmatrix}$$

[10] La notation $\mathrm{diag}(a,b,c)$ représente la matrice diagonale dont les éléments sur la diagonale sont a, b et c.

Multipliant les deux côtés avec l'inverse de U etc. donne :

$$\begin{pmatrix} \pm\mathbf{v}_2^\mathsf{T} \\ \mp\mathbf{v}_1^\mathsf{T} \\ -\mathbf{v}_3^\mathsf{T} \end{pmatrix} [\mathbf{v}_3]_\times \sim \begin{pmatrix} \mathbf{v}_1^\mathsf{T} \\ \mathbf{v}_2^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \end{pmatrix}$$

puis (en utilisant le fait que $\mathbf{v}^\mathsf{T} [\mathbf{v}]_\times = (\mathbf{v} \times \mathbf{v})^\mathsf{T} = \mathbf{0}^\mathsf{T}$ pour tout vecteur $\mathbf{v}$) :

$$\begin{pmatrix} \pm\mathbf{v}_2^\mathsf{T} [\mathbf{v}_3]_\times \\ \mp\mathbf{v}_1^\mathsf{T} [\mathbf{v}_3]_\times \\ \mathbf{0}^\mathsf{T} \end{pmatrix} \sim \begin{pmatrix} \mathbf{v}_1^\mathsf{T} \\ \mathbf{v}_2^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \end{pmatrix}$$

Les $\pm$ peuvent être tirés en dehors de la matrice de gauche, et puis écartés de l'équation, puisque celle-ci est déterminée à un facteur multiplicatif (et donc au signe) près, donnant :

$$\begin{pmatrix} \mathbf{v}_2^\mathsf{T} [\mathbf{v}_3]_\times \\ -\mathbf{v}_1^\mathsf{T} [\mathbf{v}_3]_\times \\ \mathbf{0}^\mathsf{T} \end{pmatrix} \sim \begin{pmatrix} \mathbf{v}_1^\mathsf{T} \\ \mathbf{v}_2^\mathsf{T} \\ \mathbf{0}^\mathsf{T} \end{pmatrix}$$

La matrice V est issue de la décomposition en valeurs singulières de E ; il s'agit donc d'une matrice orthogonale. A l'aide des propriétés des matrices orthogonales, l'équation ci-dessus peut être vérifiée ...

Remarque. Toutes les deux solutions pour R sont mathématiquement valides. Pourtant, la mauvaise solution peut souvent être écartée en pratique. Notamment, si elle est utilisée pour effectuer la reconstruction 3-D, comme décrit en §4, on obtiendra souvent comme résultat des points 3-D qui se trouvent *derrière* une ou même des deux caméras ...

7.3 Résumé : estimation du mouvement d'une caméra calibrée

Dans les deux derniers chapitres, nous avons introduit des techniques qui permettent d'estimer le mouvement d'une caméra calibrée, en ayant à disposition uniquement des images d'une scène inconnue. Le squelette d'une approche pratique peut alors être résumé comme suit :

1. Mise en correspondance des images, avec une méthode non contrainte par la géométrie épipolaire (non disponible à ce stade). Le résultat sera « infesté » par des outliers.
2. Estimation robuste de la géométrie épipolaire.
3. Optionnel : mise en correspondance plus fine (sur des points image moins pertinants), en utilisant la géométrie épipolaire. Ré-estimation de la géométrie épipolaire avec toutes les correspondances trouvées.
4. Calcul de la matrice essentielle (voir l'équation (25)).
5. Extraction du mouvement, comme décrit dans la section précédente.

Si la scène contient des objets bougeant différemment, la segmentation de mouvements peut être effectuée, et le résultat final serait un ensemble de mouvements : le mouvement associé au plus grand objet trouvé sera typiquement adopté comme estimation du mouvement de la caméra ; les autres décriront alors les mouvements des autres objets par rapport à la caméra.

8 Reconstruction 3-D à partir de plusieurs images

Dans le chapitre précédent, nous avons décrit des méthodes permettant d'obtenir la position relative de deux caméras, ou bien le mouvement qu'effectue une caméra entre deux prises de vue. Ensuite, la méthode du §4 peut être appliquée afin d'obtenir une reconstruction 3-D des points à partir des deux images.

C'est déjà pas mal de pouvoir faire ceci – on peut alors regarder en avant et se poser d'autres questions :

- est-il possible d'estimer le mouvement de caméra et la structure 3-D de la scène *simultanément*, et non l'un après l'autre ?
- est-il possible d'utiliser plus de deux images à la fois ?

On peut s'attendre à ce que des solutions à ces questions permettent d'obtenir des résultats plus précis, surtout en ce qui concerne la deuxième question. La réponse aux questions est « oui, mais » ... Le premier « mais » concerne le modèle de caméra – avec le modèle de projection perspective (modèle sténopé) utilisé jusqu'ici, il est possible de travailler dans la direction indiquée par les questions. Par contre, c'est moins intuitif et plus complexe à mettre en œuvre qu'avec un modèle de caméra plus simple – la *projection affine*, que nous allons détailler dans le paragraphe suivant. En §8.2, nous développons une méthode de reconstruction « multi-images » qui s'appuie sur ce modèle de caméra et qui permet effectivement de déterminer les mouvements de caméra et la structure 3-D simultanément, et ce en utilisant plusieurs images à la fois.

8.1 Le modèle de caméra affine

La représentation algébrique du modèle de projection perspective ou modèle sténopé est une matrice de projection – une matrice de dimension 3×4 (qui est définie à un facteur scalaire près). Si nous imposons que la matrice de projection ait la forme :

$$\mathsf{P} \sim \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & 0 & \times \end{pmatrix} \tag{28}$$

on obtient une matrice de *projection affine*.

Pourquoi cette dénomination ? Considérons les projections de deux droites parallèles. Deux droites qui sont parallèles ont un point d'intersection qui est un point à l'infini. Soit

$$\mathbf{Q} \sim \begin{pmatrix} \bar{\mathbf{Q}} \\ 0 \end{pmatrix}$$

le vecteur des coordonnées homogènes de ce point. Les projections des deux droites sont des droites dans l'image qui contiennent la projection de $\mathbf{Q}$, qui est donnée par (pour P comme dans l'équation (28)) :

$$\mathbf{q} \sim \mathsf{P}\mathbf{Q} \sim \begin{pmatrix} \times \\ \times \\ 0 \end{pmatrix}$$

On note que $\mathbf{q}$ est un point à l'infini dans le plan image. Par conséquent, les droites dans l'image sont parallèles, tout comme c'est le cas pour les droites originales en 3-D. On dit que la projection préserve le parallélisme de droites.

Plus généralement, une projection de la forme (28) préserve le parallélisme et des rapports de distances le long d'une droite. Puisque ces propriétés caractérisent les transformations affines, on parle alors de *projection affine*.

Il est intéressant de déterminer le centre de projection d'une projection affine. Puisque son image n'est pas définie, le centre de projection

$$\mathbf{C} \sim \begin{pmatrix} X \\ Y \\ Z \\ t \end{pmatrix}$$

doit vérifier :

$$\mathsf{P}\mathbf{C} = \mathbf{0}$$

Ceci donne :

$$\begin{pmatrix} \times \\ \times \\ P_{34}t \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Le coefficient P_{34} est en général non nul (sinon, la projection serait dégénérée, contenant toute une ligne de zéros). Par conséquent, $t = 0$, ce qui veut dire rien d'autre que le centre de projection est un point à l'infini ! Les rayons de projection contiennent le centre de projection et sont donc tous parallèles entre eux, c'est pourquoi on parle aussi de *projection parallèle*.

Le modèle de projection affine (on parlera parfois aussi de *caméra affine*) est moins riche que celui de la projection perspective et il décrit donc en général moins bien ce qui se passe dans une caméra réelle. Pourtant, le modèle affine est une bonne approximation par exemple dans les situations suivantes :

- On utilise un objectif avec un très grand zoom ou bien une mise au point à l'infini.
- La profondeur de la scène est petite par rapport à la distance entre la caméra et la scène.

L'avantage du modèle affine par rapport au modèle perspectif réside dans le fait qu'on peut éviter la prise en compte parfois embarassante des facteurs scalaires imposés par l'utilisation des coordonnées homogènes. Concrètement, on peut fixer l'échelle des matrices de projection telle que les matrices sont de la forme :

$$\mathsf{P} = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aussi bien, si l'on fait de la reconstruction 3-D, on peut imposer que les vecteurs-4 représentant les points 3-D aient un 1 comme dernière coordonnée.

Alors, on peut écrire l'équation de projection entre points 3-D et points 2-D dans l'image,

avec une égalité *exacte* entre vecteurs (il n'y a plus de « $\sim$ ») :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \times \\ \times \\ 1 \end{pmatrix}$$

Dans la suite, nous utilisons les définitions suivantes :

$$\mathbf{q} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{q}} \\ 1 \end{pmatrix} \qquad \mathbf{Q} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{Q}} \\ 1 \end{pmatrix} \qquad \mathsf{P} = \begin{pmatrix} \mathsf{A} & \mathbf{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}$$

avec A de dimension 2×3 et $\mathbf{b}$ de longueur 2. Ainsi, l'équation de projection ci-dessus s'écrit comme :

$$\begin{pmatrix} \bar{\mathbf{q}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{A} & \mathbf{b} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{Q}} \\ 1 \end{pmatrix} \tag{29}$$

8.2 Estimation du mouvement et reconstruction 3-D multi-images par factorisation

8.2.1 Formulation du problème

Considérons maintenant la situation suivante. On dispose de m images d'une scène statique et on a réussi à extraire et mettre en correspondance les projections de n points de la scène. Soit

$$\mathbf{q}_{ij} \quad i = 1, \ldots, m; j = 1, \ldots, n$$

la projection du j^{e}point sur la i^{e}image. Les $\mathbf{q}_{ij}$ sont nos seules données. Regardons d'où viennent ces points image : il y a m matrices de projection P_i et n points 3-D $\mathbf{Q}_j$ tels que :

$$\mathsf{P}_i \mathbf{Q}_j = \mathbf{q}_{ij} \quad \forall i, j$$

ou bien (d'après l'équation (29)) :

$$\begin{pmatrix} \mathsf{A}_i & \mathbf{b}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{Q}}_j \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{q}}_{ij} \\ 1 \end{pmatrix} \quad \forall i, j \tag{30}$$

Le but est de reconstruire les P_i et les $\mathbf{Q}_j$. Nous avons la liberté du choix pour le repère de coordonnées dans lequel la reconstruction sera exprimée – liberté de choix dont on pourra

profiter pour simplifier le problème à résoudre. Nous pouvons par exemple « imposer » que l'un des points 3-D reconstruits se trouve à l'origine du repère. Sans perte de généralité :

$$\mathbf{Q}_1 = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$$

Considérons alors les m projections de ce point :

$$\begin{pmatrix} \bar{\mathbf{q}}_{i1} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{A}_i & \mathbf{b}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_i \\ 1 \end{pmatrix} \quad \forall i$$

Ces équations nous donnent directement la dernière colonne $\mathbf{b}_i$ pour chacune des matrices de projection : $\mathbf{b}_i = \bar{\mathbf{q}}_{i1}$. En remplaçant ceci dans l'équation (30), on obtient alors :

$$\begin{pmatrix} \mathsf{A}_i & \bar{\mathbf{q}}_{i1} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{Q}}_j \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{q}}_{ij} \\ 1 \end{pmatrix} \quad \forall i, j$$

ou bien :

$$\begin{pmatrix} \mathsf{A}_i \bar{\mathbf{Q}}_j + \bar{\mathbf{q}}_{i1} \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{q}}_{ij} \\ 1 \end{pmatrix} \quad \forall i, j$$

Seulement les deux premiers coefficients de ces vecteurs-3 sont intéressants, et après soustraction du vecteur $\bar{\mathbf{q}}_{i1}$ on obtient :

$$\mathsf{A}_i \bar{\mathbf{Q}}_j = \bar{\mathbf{q}}_{ij} - \bar{\mathbf{q}}_{i1} \quad \forall i, j$$

Comment peut-on interpréter le côté droit de cette équation ? Il correspond en effet à un changement de repère dans le plan image ; plus concrètement, une translation qui ramène les points $\bar{\mathbf{q}}_{i1}$ à l'origine (respectivement pour chaque i). Soient les $\mathbf{v}_{ij}$ les coordonnées des points image dans les nouveaux repères[11] :

$$\mathbf{v}_{ij} = \bar{\mathbf{q}}_{ij} - \bar{\mathbf{q}}_{i1} \quad \forall i, j$$

On a donc mn équations de la forme :

$$\mathsf{A}_i \bar{\mathbf{Q}}_j = \mathbf{v}_{ij}$$

où les inconnues sont les matrices A_i de dimension 2×3 et les vecteurs $\bar{\mathbf{Q}}_j$ de longueur 3.

8.2.2 Méthode de factorisation

L'ensemble des mn équations peut être écrit sous forme d'un système matriciel :

$$\underbrace{\begin{bmatrix} \mathsf{A}_1 \\ \mathsf{A}_2 \\ \vdots \\ \mathsf{A}_m \end{bmatrix}}_{\mathsf{A}} \underbrace{\begin{bmatrix} \bar{\mathbf{Q}}_1 & \bar{\mathbf{Q}}_2 & \cdots & \bar{\mathbf{Q}}_n \end{bmatrix}}_{\mathsf{Q}} = \underbrace{\begin{bmatrix} \mathbf{v}_{11} & \mathbf{v}_{12} & \cdots & \mathbf{v}_{1n} \\ \mathbf{v}_{21} & \mathbf{v}_{22} & \cdots & \mathbf{v}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_{m1} & \mathbf{v}_{m2} & \cdots & \mathbf{v}_{mn} \end{bmatrix}}_{\mathsf{V}}$$

[11] Naturellement, on a $\mathbf{v}_{i1} = \mathbf{0}$.

Les dimensions de ces matrices sont :

$$\mathsf{A}_{2m\times 3}\mathsf{Q}_{3\times n} = \mathsf{V}_{2m\times n} \tag{31}$$

Faisons quelques observations :

- On dispose de $2mn$ données (les coordonnées de tous les points dans toutes les images) – les coefficients de la matrice V. Pourtant, ces valeurs ne sont pas indépendantes les unes des autres, puisqu'elles peuvent être reproduites à partir de seulement $6m+3n$ paramètres ($6m$ pour les matrices de projection et $3n$ pour les points 3-D). Implicitement, c'est cette redondance dans les données qui permettra d'obtenir la reconstruction 3-D.
- La matrice A n'a que 3 colonnes ; elle est donc de rang 3 au plus. Idem pour la matrice Q, qui n'a que 3 lignes. Le rang d'un produit de matrices est toujours inférieur ou égal au minimum des rangs des matrices individuelles. Par conséquent, la matrice V est de rang 3 au plus (c'est une manifestation de la redondance dans les données, mentionnée ci-dessus). Cette observation est la clef de l'algorithme de reconstruction énoncé dans la suite.

Si l'on effectue la décomposition en valeurs singulières de V, on obtient :

$$\mathsf{V}_{2m\times n} = \mathsf{U}_{2m\times n}\Sigma_{n\times n}\mathsf{X}_{n\times n} \tag{32}$$

Le fait que le rang de V soit 3 au plus implique qu'au plus 3 valeurs singulières sont non nulles. La matrice diagonale Σ est alors de la forme suivante :

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

Soit $\mathbf{u}_i$ la i^{e}colonne de U et $\mathbf{x}_j^\mathsf{T}$ la j^{e}ligne de X. L'équation (32) peut alors s'écrire :

$$\begin{aligned}
\mathsf{V} &= \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 & \cdots & \mathbf{u}_n\end{pmatrix}
\begin{pmatrix}\sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0\end{pmatrix}
\begin{pmatrix}\mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T} \\ \mathbf{x}_4^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T}\end{pmatrix} \\
&= \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \mathbf{u}_4 & \cdots & \mathbf{u}_n\end{pmatrix}
\begin{pmatrix}\sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0\end{pmatrix}
\begin{pmatrix}\mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T}\end{pmatrix} \\
&= \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{pmatrix}
\begin{pmatrix}\sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3\end{pmatrix}
\begin{pmatrix}\mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T}\end{pmatrix}
\end{aligned}$$

Si nous introduisons les notations

$$\begin{aligned}
\mathsf{A}' &= \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{pmatrix}
\begin{pmatrix}\sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3\end{pmatrix} \qquad (33) \\
\mathsf{Q}' &= \begin{pmatrix}\mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T}\end{pmatrix}
\end{aligned}$$

on obtient alors une équation matricielle avec les dimensions suivantes :

$$\mathsf{V}_{2m\times n} = \mathsf{A}'_{2m\times 3}\mathsf{Q}'_{3\times n} \ .$$

On note qu'on a construit des matrices A' et Q' qui « reproduisent » les données V et qui ont les mêmes dimensions que les matrices A et Q dans l'équation (31). Si l'on extrait les sous-matrices A'_i de dimension 2×3 de A' et les colonnes $\bar{\mathbf{Q}}'_j$ (des vecteurs-3) de Q', on obtient une estimation du mouvement et une reconstruction 3-D correctes : si l'on re-projette les points 3-D (représentés par les $\bar{\mathbf{Q}}'_j$) par les matrices de projection (les A'_i), on obtient les points image $\mathbf{v}_{ij}$ mesurés, nos données de départ.

8.2.3 Concernant l'unicité de la reconstruction

La méthode décrite ci-dessus permet donc d'obtenir une reconstruction 3-D, pourtant, il n'y a pas de solution unique. Afin de passer de la décomposition en valeurs singulières à deux matrices A' et Q' avec les bonnes dimensions, nous avons fait le choix arbitraire d'« absorber

» la matrice diagonale des valeurs singulières dans la matrice de gauche (voir l'équation (33)).

Nous aurions aussi bien pu choisir par exemple :

$$\begin{aligned} \mathsf{A}' &= \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{pmatrix} \\ \mathsf{Q}' &= \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T} \end{pmatrix} \end{aligned}$$

ou bien

$$\begin{aligned} \mathsf{A}' &= \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{pmatrix} \begin{pmatrix} \sqrt{\sigma_1} & 0 & 0 \\ 0 & \sqrt{\sigma_2} & 0 \\ 0 & 0 & \sqrt{\sigma_3} \end{pmatrix} \\ \mathsf{Q}' &= \begin{pmatrix} \sqrt{\sigma_1} & 0 & 0 \\ 0 & \sqrt{\sigma_2} & 0 \\ 0 & 0 & \sqrt{\sigma_3} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \mathbf{x}_3^\mathsf{T} \end{pmatrix} \end{aligned}$$

Plus généralement, ayant obtenu une solution pour A' et Q', les matrices définies par :

$$\mathsf{A}'' = \mathsf{A}'\mathsf{T} \qquad \mathsf{Q}'' = \mathsf{T}^{-1}\mathsf{Q}'$$

sont aussi une solution possible, pour toute matrice 3×3 inversible T, puisque :

$$\mathsf{A}''\mathsf{Q}'' = \mathsf{A}'\mathsf{Q}' = \mathsf{V}$$

Il existe donc effectivement une « famille de solutions » de 9 degrés de liberté (les 3×3 coefficients de T). Comment interpréter ce résultat ?

- Premièrement, il faut souligner qu'on a réduit le nombre d'inconnues de $6m + 3n$ à 9.
- La matrice T représente une transformation affine de l'espace 3-D. Puisque c'est exactement cette matrice qui reste indéfinie, on dit alors qu'on a obtenu une reconstruction 3-D *à une transformation affine près* (ou bien une *reconstruction affine*). Ayant choisi n'importe laquelle des solutions possibles pour A' et Q', on sait alors que cette reconstruction approche la « réalité » à seulement une transformation affine près : des points qui sont co-planaires dans la scène le seront dans la reconstruction, et aussi bien le parallélisme de droites et les rapports de longueurs le long d'une droite sont correctement retrouvés dans la reconstruction.

Il y a plusieurs moyens de déterminer, parmi la famille de solutions, la bonne solution (qui ne sera plus seulement une reconstruction affine, mais une reconstruction métrique ou Euclidienne) :

- Des connaissances sur par exemple des distances entres des points dans la scène donnent des contraintes sur la transformation affine T. Suffisamment de contraintes permettront de trouver une solution unique pour T et donc pour la reconstruction.
- Des connaissances sur les caméras (sur les A_i) pourront être utilisées de la même manière.

8.2.4 Quelques remarques

Au §8.2.1, nous avons fixé l'origine du repère 3-D de la reconstruction sur un point particulier de la scène. Cette méthode ne doit pas être appliquée directement comme telle en pratique, puisque la qualité de la reconstruction entière dépendra fortement de la qualité de la mise en correspondance concernant le seul point choisi.

En présence de bruit dans les données, la décomposition en valeurs singulières de V ne donnera pas seulement trois valeurs singulières non nulles. Pourtant, en pratique, il y aura trois valeurs singulières qui seront très grandes par rapport aux autres, en on fait comme si ces autres valeurs étaient égales à zéro.

8.3 Bibliographie

- C. Tomasi et T. Kanade, *Shape and Motion from Image Streams under Orthography : A Factorization Method,* International Journal on Computer Vision, Vol. 9, No. 2, pp. 137-154, 1992.
- P. Sturm et B. Triggs, *A factorization based algorithm for multi-image projective structure and motion,* European Conference on Computer Vision, pp. 709-720, 1996.
- B. Triggs, *Factorization Methods for Projective Structure and Motion,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 845-851, 1996.

9 Bibliographie supplémentaire

Voici quelques références pour une lecture approfondie. Pour les thèmes non citées ci-dessous, se référer aux livres donnés dans la bibliographie générale, surtout celui de Hartley et Zisserman.

Calibrage de caméra

- R.I. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- P. Sturm et S.J. Maybank, *On Plane-Based Camera Calibration : A General Algorithm, Singularities, Applications,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 432-437, 1999.

Mosaïques d'images

- H.-Y. Shum et R. Szeliski, *Panoramic Image Mosaics,* Technical Report MSR-TR-97-23, Microsoft Research, 1997.
- M. Jethwa, A. Zisserman et A. Fitzgibbon, *Real-time Panoramic Mosaics and Augmented Reality,* British Machine Vision Conference, pp. 852-862, 1998.

Calibrage en ligne / auto-calibrage

- R.I. Hartley, *An Algorithm for Self Calibration from Several Views,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 908-912, 1994.
- B. Triggs, *Autocalibration and the Absolute Quadric,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 609-614, 1997.
- M. Pollefeys, R. Koch et L. Van Gool, *Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters*, International Conference on Computer Vision, pp. 90-95, 1998.
- R. Horaud et G. Csurka, *Self-Calibration and Euclidean Reconstruction Using Motions of A Stereo Rig,* International Conference on Computer Vision, pp. 96-103, 1998.
- R.I. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- P. Sturm, *Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1100-1105, 1997.

Reconstruction 3-D à partir de deux images (calibrées ou non)

- R. Hartley et P. Sturm, *Triangulation*, Computer Vision and Image Understanding, Vol. 68, No. 2, pp. 146-157, 1997.

Détermination de la pose d'un objet

- R.J. Holt et A.N. Netravali, *Camera Calibration Problem : Some New Results,* CVGIP - Computer Vision, Graphics and Image Processing, Vol. 54, No. 3, pp. 368-383, 1991.

- R.M. Haralick, C. Lee, K. Ottenberg et M. Nölle, *Analysis and Solutions of the Three Point Perspective Pose Estimation Problem,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 592-598, 1991.
- D. Dementhon et L.S. Davis, *Model-Based Object Pose in 25 Lines of Code,* International Journal on Computer Vision, Vol. 15, No. 1/2, pp. 123-141, 1995.
- P. Sturm, *Algorithms for Plane-Based Pose Estimation,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 706-711, 2000.

Reconstruction 3-D à partir de plusieurs images

- C. Tomasi et T. Kanade, *Shape and Motion from Image Streams under Orthography : A Factorization Method,* International Journal on Computer Vision, Vol. 9, No. 2, pp. 137-154, 1992.
- P. Sturm et B. Triggs, *A factorization based algorithm for multi-image projective structure and motion,* European Conference on Computer Vision, pp. 709-720, 1996.
- B. Triggs, *Factorization Methods for Projective Structure and Motion,* IEEE International Conference on Computer Vision and Pattern Recognition, pp. 845-851, 1996.