

STÉRÉOVISION.

1• OBJECTIF DE CE TRAVAIL.

Il s'agit de mettre en œuvre plusieurs variantes d'identification de la matrice fondamentale associée à une paire d'images stéréoscopiques. Les résultats sont visualisés au moyen des droites épipolaires.

2• RAPPELS CONCERNANT LA MATRICE FONDAMENTALE.

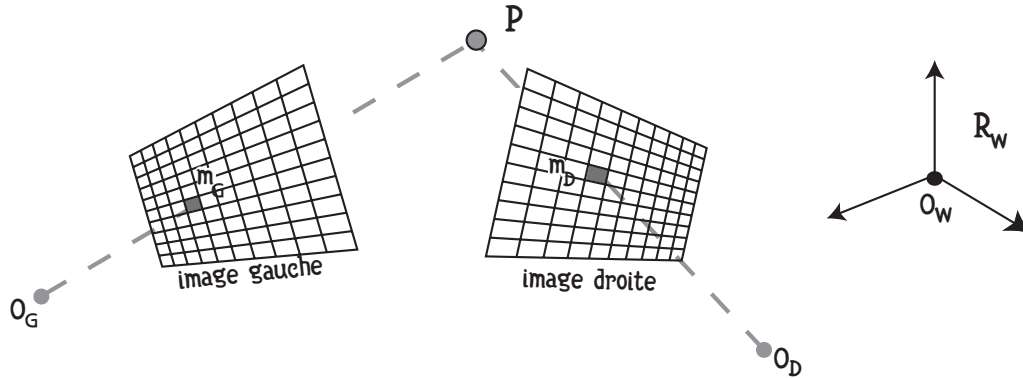


Figure 1 : Principe de la stéréovision.

Quand on dispose de deux images stéréoscopiques acquises par des caméras projectives, les projections m_G et m_D d'un même point P sur les images gauche et droite sont reliées par la relation épipolaire faisant intervenir la matrice fondamentale F :

$$m_G^T F m_D = 0 \quad (1)$$

$$\text{avec } m_G = \begin{bmatrix} x_G \\ y_G \\ 1 \end{bmatrix}, m_D = \begin{bmatrix} x_D \\ y_D \\ 1 \end{bmatrix} \text{ et } F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}.$$

Cette équation montre clairement que la matrice F est définie à un facteur d'échelle près. En effet si $m_G^T F m_D = 0$, alors $m_G^T F' m_D = 0$, avec $F' = \alpha F$, α étant un réel quelconque.

L'équation fondamentale (1) est une des équations épipolaires. Elle permet de définir, pour la projection m_G d'un point 3D P sur l'image gauche, l'ensemble des points de l'image droite pouvant être projection de P . Cet ensemble de points se trouve sur une droite appelée **droite épipolaire** dont les coefficients sont donnés par le vecteur L_D :

$$L_D = F^T m_G = [l_{D1}, l_{D2}, l_{D3}]^T, \quad (2)$$

menant à l'équation de la droite épipolaire dans l'image droite :

$$[l_{D1}, l_{D2}, l_{D3}] m_D = l_{D1} x_D + l_{D2} y_D + l_{D3} = 0 \quad (3)$$

Tout point de l'image droite ne vérifiant pas l'équation (3) ne peut être projection sur l'image droite de P, le point 3D dont m_G est la projection sur l'image gauche.

Le même raisonnement peut être tenu pour un point de l'image droite et sa droite épipolaire correspondante dans l'image gauche.

Question 1 : écrivez ce raisonnement.

En développant l'équation (1) on obtient :

$$x_G x_D f_{11} + x_G y_D f_{12} + x_G f_{13} + y_G x_D f_{21} + y_G y_D f_{22} + y_G f_{23} + x_D f_{31} + y_D f_{32} + f_{33} = 0, \quad (4)$$

En réécrivant cette équation sous forme matricielle on obtient :

$$\begin{bmatrix} x_G x_D & x_G y_D & x_G & y_G x_D & y_G y_D & y_G & x_D & y_D & 1 \end{bmatrix} f = 0, \quad (5)$$

où f est le vecteur $\begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{21} & f_{22} & f_{23} & f_{31} & f_{32} & f_{33} \end{bmatrix}^T$.

Les contraintes épipolaires sont telles que la matrice F est de rang 2, ce qui veut dire qu'une de ses valeurs propres est nulle. Cette propriété est importante et son exploitation permettra d'améliorer l'identification de F .

3• AIDE POUR LA PROGRAMMATION.

3.1• Sélection de paires de points.

Pour vous aider dans la suite du programme, voici le morceau de code vous permettant de sélectionner des paires de points sous matlab et de tracer une droite d'équation

$ax + by + c = 0$.

```
% Nom de l'image que l'on va charger
% NomGenerique = 'Turtle' ;
% NomGenerique = 'Baixa' ;
% NomGenerique = 'Dracula' ;
% NomGenerique = 'Rock' ;
NomGenerique = 'Stereo' ;
% et son suffixe
Suffixe = '.tif' ;
% Suffixe = '.jpg' ;
% Composition des noms des images a charger et chargement de ces images
Nom = sprintf('%sG%s', NomGenerique, Suffixe) ;
ImageGauche = double(imread(Nom)) ;
[Nlin, Ncol, Nplan] = size(ImageGauche) ;
figure(1) ; hold off ; image(uint8(ImageGauche)) ; hold on ; axis image ;
title('Image gauche') ; if(Nplan<3) colormap(gray(256)) ; end ; drawnow ;
Nom = sprintf('%sD%s', NomGenerique, Suffixe) ;
ImageDroite = double(imread(Nom)) ;
figure(2) ; hold off ; image(uint8(ImageDroite)) ; hold on ; axis image ;
title('Image droite') ; if(Nplan<3) colormap(gray(256)) ; end ; drawnow ;
saisie=0;
% Chargement des points appariés entre les deux images
% il vous faut définir le nombre de points a saisir
% ou créer une boucle while avec un arrêt relié à l'appui sur une touche
% par exemple
NombreDePoints = 16 ;
'
```

```

% Le drapeau suivant sert a eviter de re-cliquer a chaque fois
% sur les points de calibrage
saisir_les_points = 1 ;

if saisir_les_points
    PointDroite = ones(NombreDePoints,3) ;
    PointGauche = ones(NombreDePoints,3) ;
    for n=1:NombreDePoints
        figure(1) ; drawnow ;
        [x,y] = ginput(1) ;
        plot(x,y,'bo') ;
        PointGauche(n,1:2) = [x,y] ;
        figure(2) ; drawnow ;
        [x,y] = ginput(1) ;
        plot(x,y,'ro') ;
        PointDroite(n,1:2) = [x,y] ;
    end
    save PointsDeCalibrage PointDroite PointGauche ;
else
    load PointsDeCalibrage ;
end

NombreDePoints = length(PointGauche) ;

XG=PointGauche(:,1);
YG=PointGauche(:,2);
XD=PointDroite(:,1);
YD=PointDroite(:,2);

% methode 1 moindres carres simples
% methode 2 norme de F = 1
methode=1

if methode==1
% methode 1 - methode des moindres carres avec contrainte unitaire
% sur une des coordonnees de f (question 4.1)
% calculer A B et f
end;

if methode==2
%methode 2 - methode des moindres carres avec contrainte unitaire
%sur la norme de f (question 4.2)
%calculer C D V et f
end;

% cliquez sur
for n=1:NombreDePoints
    h=gcf ; correct = 0 ;

    if(h==1) correct = 1 ; couleur = 'r' ; end
    if(h==2) correct = 1 ; couleur = 'b' ; end

    if correct
        [xs,ys] = ginput(1) ;
        m=[xs; ys; 1];
        if h==1
            plot(xs,ys,'ro') ;
            % calcul de L = [a b c] ;
        end
    end
end

```

```

else
    plot(xs,ys,'bo') ;
    % calcul de L = [a b c] ;
end

x_min=1;
y_min=1;
if(h==1)
    x_max=size(ImageDroite,2);
    y_max=size(ImageDroite,1);
else
    x_max=size(ImageGauche,2);
    y_max=size(ImageGauche,1);
end
y_x_min = -1 ; y_x_max = -1 ;
x_y_min = -1 ; x_y_max = -1 ;

k=0 ;
if( ( y_x_min >= y_min ) && ( y_x_min <= y_max ) )
    k=k+1 ;
    Point(k,1) = x_min ;
    Point(k,2) = y_x_min ;
end
if( ( y_x_max >= y_min ) && ( y_x_max <= y_max ) )
    k=k+1 ;
    Point(k,1) = x_max ;
    Point(k,2) = y_x_max ;
end
if( ( x_y_min >= x_min ) && ( x_y_min <= x_max ) )
    k=k+1 ;
    Point(k,1) = x_y_min ;
    Point(k,2) = y_min ;
end
if( ( x_y_max >= x_min ) && ( x_y_max <= x_max ) )
    k=k+1 ;
    Point(k,1) = x_y_max ;
    Point(k,2) = y_max ;
end

if(h==1) figure(2) ; else figure(1) ; end ;
line( Point(:,1), Point(:,2), 'color', couleur ) ;
end

```

3.2• Affichage d'une droite dont on connaît le vecteur normal.

Soit la droite d'équation $ax + by + c = 0$ représentée par le vecteur $L = [a, b, c]^T$. Pour afficher cette droite sur une image on utilise la fonction `line` qui trace une ligne entre deux points. Pour calculer ces deux points, le plus simple est de calculer l'intersection de cette droite avec le cadre de l'image. Vous avez cette solution dans le programme ci-dessus.

3.3• Selection de points aléatoires.

Pour sélectionner p points aléatoirement parmi n points, il faut créer un vecteur d'indice de p valeurs parmi n . En matlab cette procédure peut être facilement réalisée par :

```
indice = floor(rand(1,p)*n) + 1 ;
```

3.4• Décomposition en valeurs singulières.

Dans ce TP vous aurez besoin de faire des décompositions en valeur singulières de matrices. La fonction `svd` (pour **s**ingular **v**alue **d**ecomposition) vous permet de facilement réaliser cette décomposition. Soit la matrice D à décomposer en valeur singulière :

$$[U, S, V] = \text{svd}(D) ;$$

produit trois matrices. S est la matrice des valeurs propres de D triées par ordre décroissant et V est la matrice des vecteur propres de D . La $k^{\text{ième}}$ colonne de V est le vecteur propre associé à la $k^{\text{ième}}$ valeur propre de D . Si D est symétrique définie positive (ce qui sera le cas pour vous) alors $U=V$.

3.5• Statistiques, médianes et quartiles.

On décrit généralement la statistique d'un ensemble de valeurs par sa moyenne et son écart-type. La médiane et les quartiles est une autre façon de décrire cette statistique.

Soit n valeurs $\{\varepsilon_1, \dots, \varepsilon_n\}$ et un opérateur de tri noté $(.)$, tel que $\varepsilon_{(1)} \leq \dots \leq \varepsilon_{(n)}$,

la médiane μ et l'écart interquartile σ sont donnés par :

$$\mu = \varepsilon_{(n/2)}, \quad \sigma = \varepsilon_{(3n/4)} - \varepsilon_{(n/4)}. \quad (6)$$

4• IDENTIFICATION DE LA MATRICE F .

Pour identifier la matrice F il faut et suffit d'identifier le vecteur f . Pour cela, on a besoin de connaître un certain nombre d'appariements droite-gauche. Evidemment, plus il y a de points, plus l'estimation est fiable. Dans la suite du TP, on utilisera environ 12 points.

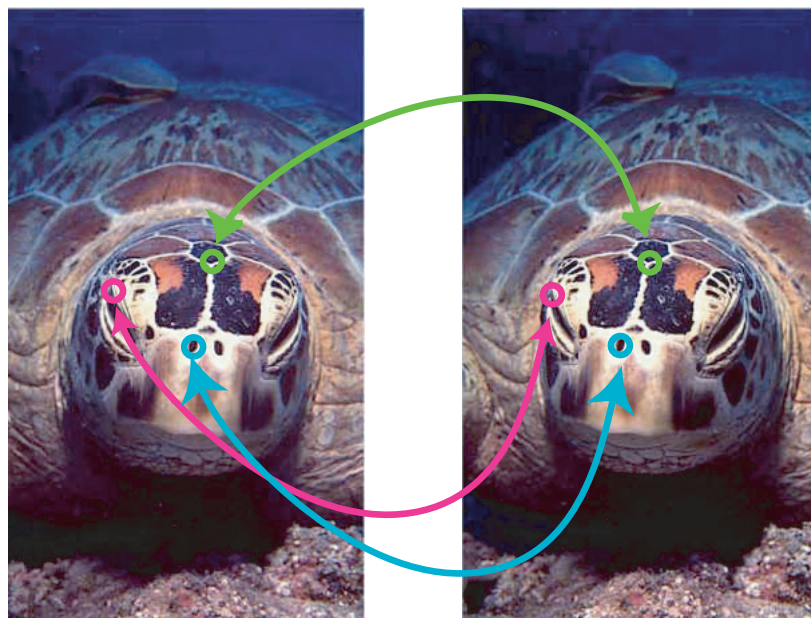


Figure 2 : Appariements de point droite-gauche sur images stéréoscopique.

Question 2 : quel est le nombre minimal de points appariés qu'il faut utiliser pour identifier la matrice F ?

Si on possède n couples de points, on peut donc écrire N équations de type (5), ce qui

peut se résumer matriciellement par :

$$\begin{bmatrix} x_{G1}x_{D1} & x_{G1}y_{D1} & x_{G1} & y_{G1}x_{D1} & y_{G1}y_{D1} & y_{G1} & x_{D1} & y_{D1} & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{Gn}x_{Dn} & x_{Gn}y_{Dn} & x_{Gn} & y_{Gn}x_{Dn} & y_{Gn}y_{Dn} & y_{Gn} & x_{Dn} & y_{Dn} & 1 \end{bmatrix} \mathbf{f} = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix}. \quad (7)$$

Comme la matrice F est définie à un facteur d'échelle près, le vecteur \mathbf{f} est aussi défini à un facteur d'échelle près. Pour l'estimer, il faut imposer une contrainte à l'identificateur utilisé.

4.1• Méthode des moindres carrés avec contrainte unitaire sur une des coordonnées de \mathbf{f} .

Cette méthode est la plus simple. Elle consiste à supposer qu'un des éléments de \mathbf{f} vaut une valeur fixe. On prend généralement $f_{33}=1$. Dans ce cas (7) devient :

$$\begin{bmatrix} x_{G1}x_{D1} & x_{G1}y_{D1} & x_{G1} & y_{G1}x_{D1} & y_{G1}y_{D1} & y_{G1} & x_{D1} & y_{D1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{Gn}x_{Dn} & x_{Gn}y_{Dn} & x_{Gn} & y_{Gn}x_{Dn} & y_{Gn}y_{Dn} & y_{Gn} & x_{Dn} & y_{Dn} \end{bmatrix} \mathbf{f} = \begin{bmatrix} -1 \\ \dots \\ -1 \end{bmatrix}. \quad (8)$$

Posons $A = \begin{bmatrix} x_{G1}x_{D1} & x_{G1}y_{D1} & x_{G1} & y_{G1}x_{D1} & y_{G1}y_{D1} & y_{G1} & x_{D1} & y_{D1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{Gn}x_{Dn} & x_{Gn}y_{Dn} & x_{Gn} & y_{Gn}x_{Dn} & y_{Gn}y_{Dn} & y_{Gn} & x_{Dn} & y_{Dn} \end{bmatrix}$ et $B = \begin{bmatrix} -1 \\ \dots \\ -1 \end{bmatrix}$, la résolu-

tion moindres carrés de l'équation (8) est donnée par :

$$\hat{\mathbf{f}} = (A^T A)^{-1} A^T B. \quad (9)$$

Plusieurs inconvénients sont connus à cette méthode. On en dénombre 4 principales :

- si $f_{33}=0$, cet identificateur ne peut pas converger, de plus le fait d'imposer une contrainte de normalisation seulement sur **un** des éléments de \mathbf{f} a tendance à très mal répartir l'erreur d'identification (a),
- la matrice F obtenue n'est que très rarement de rang 2 (b),
- le repère dans lequel les points sont exprimés a une importance fondamentale dans l'identification, on peut remarquer que, de façon surprenante, changer de repère change la nature de l'identification (c),
- la présence de mauvais appariements fait diverger cet identificateur (d).

Chacun de ces inconvénients trouve une solution.

Travail à effectuer 4.1 : Implantez cet algorithme. Vérifiez son fonctionnement en créant un petit programme qui trace, sur une des images, la ligne épipolaire correspondant à un point sélectionné sur l'autre image.

Question 3 : les droites épipolaires s'intersectent elles ? doivent-elles s'intersecter ? Si oui, quel est le point d'intersection ?

4.2• Méthode des moindres carrés avec contrainte unitaire sur la norme de \mathbf{f} .

Cette méthode règle le point (a). En présence de bruit (c'est à dire d'erreur de mesure ou de modèle) la relation (7) ne peut être exacte et doit être remplacée par :

$$\begin{bmatrix} x_{G1}x_{D1} & x_{G1}y_{D1} & x_{G1} & y_{G1}x_{D1} & y_{G1}y_{D1} & y_{G1} & x_{D1} & y_{D1} & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{Gn}x_{Dn} & x_{Gn}y_{Dn} & x_{Gn} & y_{Gn}x_{Dn} & y_{Gn}y_{Dn} & y_{Gn} & x_{Dn} & y_{Dn} & 1 \end{bmatrix} \mathbf{f} = \begin{bmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_n \end{bmatrix}, \quad (10)$$

où le vecteur $\mathbf{E} = [\varepsilon_1 \dots \varepsilon_n]^T$ est le vecteur d'erreur d'identification.

Question 4 : donnez toutes les sources d'erreur que vous pouvez recenser.

L'objet de l'identification de \mathbf{f} (et donc de \mathbf{F}) est de trouver un vecteur $\hat{\mathbf{f}}$ qui minimise la norme du vecteur \mathbf{E} . Ecrivons cette norme :

$$\|\mathbf{E}\|^2 = \mathbf{E}^T \mathbf{E} = (\mathbf{C}\mathbf{f})^T \mathbf{C}\mathbf{f} = \mathbf{f}^T \mathbf{C}^T \mathbf{C}\mathbf{f} = \mathbf{f}^T \mathbf{D}\mathbf{f}, \quad (11)$$

$$\text{avec } \mathbf{C} = \begin{bmatrix} x_{G1}x_{D1} & x_{G1}y_{D1} & x_{G1} & y_{G1}x_{D1} & y_{G1}y_{D1} & y_{G1} & x_{D1} & y_{D1} & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{Gn}x_{Dn} & x_{Gn}y_{Dn} & x_{Gn} & y_{Gn}x_{Dn} & y_{Gn}y_{Dn} & y_{Gn} & x_{Dn} & y_{Dn} & 1 \end{bmatrix} \text{ et } \mathbf{D} = \mathbf{C}^T \mathbf{C}.$$

Si on écrit l'équation (11) en imposant, par la méthode des multiplicateurs de Lagrange, au vecteur \mathbf{f} d'être unitaire (ce qu'on peut toujours faire puisqu'il est défini à un facteur d'échelle près), alors il en résulte que le vecteur $\hat{\mathbf{f}}$ unitaire minimisant la norme du vecteur \mathbf{E} n'est autre que le vecteur propre unitaire associé à la plus petite valeur propre de la matrice \mathbf{D} .

Démonstration :

Le critère à minimiser par les multiplicateurs de Lagrange est :

$$\mathbf{C}_r = \|\mathbf{E}\|^2 + \lambda(1 - \|\mathbf{f}\|^2). \text{ Ce qui veut dire qu'on minimise } \|\mathbf{E}\|^2 \text{ en imposant } 1 - \|\mathbf{f}\|^2 = 0.$$

Comme ce critère est quadratique, il sera minimal par rapport à \mathbf{f} si

$$\frac{\partial \mathbf{C}_r}{\partial \mathbf{f}} = \mathbf{0}. \quad (12)$$

Le développement de ce calcul est le suivant :

$$\mathbf{C}_r = \|\mathbf{E}\|^2 + \lambda(1 - \|\mathbf{f}\|^2) = \mathbf{f}^T \mathbf{C}^T \mathbf{C}\mathbf{f} + \lambda(1 - \mathbf{f}^T \mathbf{f}) \quad (13)$$

$$\text{, donc } \frac{\partial \mathbf{C}_r}{\partial \mathbf{f}} = 2\mathbf{C}^T \mathbf{C}\mathbf{f} - 2\lambda \mathbf{f} = \mathbf{0}. \text{ Ce qui entraîne : } \mathbf{C}^T \mathbf{C}\mathbf{f} = \mathbf{D}\mathbf{f} = \lambda \mathbf{f}, \text{ c'est à dire que}$$

tout vecteur propre de \mathbf{D} est solution. Si on remplace $\mathbf{D}\mathbf{f}$ par $\lambda \mathbf{f}$ dans l'équation (13), on obtient : $\mathbf{C}_r = \mathbf{f}^T \mathbf{D}\mathbf{f} + \lambda(1 - \mathbf{f}^T \mathbf{f}) = \lambda \mathbf{f}^T \mathbf{f} + \lambda - \lambda \mathbf{f}^T \mathbf{f} = \lambda$. Ce qui montre que le critère est égal à la valeur propre associée au vecteur propre considéré. Il sera donc minimal pour le vecteur propre associé à la valeur propre minimale.

La solution à ce problème consiste donc à effectuer une SVD (single value décomposition) de la matrice D :

$$D = USV^T, \quad (14)$$

D est la matrice diagonale des valeurs propres et V la matrice des vecteurs propres unitaires. On ne retient alors, pour le vecteur \hat{f} , la dernière colonne de V (car généralement, dans les SVD, les valeurs propres sont triées par ordre décroissant).

Travail à effectuer 4.2 : Implantez ce nouvel algorithme. Comparez la qualité des résultats obtenus avec cette méthode avec ceux obtenus avec la première méthode.

4.3• Réduction du rang de la matrice F .

La matrice F identifiée par l'une ou l'autre des méthodes précédemment décrite n'est généralement pas de rang 2 (elle serait de rang 2 s'il n'y avait aucune erreur de mesure, de modèle ou de calcul). Pour obliger la matrice F identifiée à être de rang 2, on passe de nouveau par la SVD.

Soit F la matrice identifiée (de rang plein), on cherche à obtenir la matrice F' de rang 2 la plus proche de la matrice F . Pour obtenir cette propriété, il faut faire une SVD de la matrice F , forcer la plus petite des valeurs propres à être nulle, puis on recompose la matrice fondamentale :

$$F = USV^T, \text{ avec } S = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \text{ on pose } S' = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ puis } F' = US'V^T. \quad (15)$$

4.4• Modification des données initiales.

L'espace dans lequel sont exprimés les points de mesure (points appariés m_{Dk} et m_{Gk} , $k=1\dots n$) influe énormément sur l'identificateur (c). Plusieurs raisons à cela, mais principalement le mauvais conditionnement de l'équation (7). Pour minimiser cette influence, il est généralement recommandé de faire subir une transformation d'échelle aux points de mesure de façon à ce que leur moyenne soit nulle et leur variance soit égale à deux. Une telle normalisation est très simplement obtenue en créant les matrices suivantes :

$$T_G = \begin{bmatrix} \frac{\sigma_{X_G}}{\sqrt{2}} & 0 & \tilde{x}_G \\ 0 & \frac{\sigma_{Y_G}}{\sqrt{2}} & \tilde{y}_G \\ 0 & 0 & 1 \end{bmatrix}^{-1} \quad \text{et} \quad T_D = \begin{bmatrix} \frac{\sigma_{X_D}}{\sqrt{2}} & 0 & \tilde{x}_D \\ 0 & \frac{\sigma_{Y_D}}{\sqrt{2}} & \tilde{y}_D \\ 0 & 0 & 1 \end{bmatrix}^{-1} \quad (16)$$

$$\text{avec } \tilde{x}_G = \frac{1}{n} \sum_{k=1}^n x_{Gk}, \quad \tilde{y}_G = \frac{1}{n} \sum_{k=1}^n y_{Gk}, \quad \tilde{x}_D = \frac{1}{n} \sum_{k=1}^n x_{Dk}, \quad \tilde{y}_D = \frac{1}{n} \sum_{k=1}^n y_{Dk},$$

$$\sigma_{x_G} = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_{Gk} - \tilde{x}_G)^2}, \quad \sigma_{y_G} = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_{Gk} - \tilde{y}_G)^2},$$

$$\sigma_{x_D} = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_{Dk} - \tilde{x}_D)^2}, \quad \sigma_{y_D} = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_{Dk} - \tilde{y}_D)^2}.$$

Remarque : ce sont simplement les moyennes et les variances des points de mesures.

Question 5 : Montrez que, si on multiplie l'ensemble de points gauches par T_G on obtient bien un ensemble de points de moyenne nulle et de variance 2.

Pour procéder à l'identification, il vous faut transformer tous vos points m_{Dk} et m_{Gk} en m'_{Dk} et m'_{Gk} : $m'_{Dk} = T_D m_{Dk}$ et $m'_{Gk} = T_G m_{Gk}$, identifier F' avec une des méthodes ci-dessus, puis revenir à la matrice F exprimée dans l'espace initial par :

$$F = T_G^T F' T_D. \quad (17)$$

Travail à effectuer 4.4 : Implantez cette modification. Comparez la qualité des résultats obtenus avec cette méthode avec ceux obtenus avec les méthodes précédentes.

4.5• Elimination des points aberrants.

Si quelques points ont été mal sélectionnés, le mauvais conditionnement du problème mène à une très mauvaise identification de la matrice F (ce qui est fréquent) (d).

Pour résoudre ce problème, on utilise une variante **robuste** des moindres carrés. Ce terme de robuste concerne justement les points mal sélectionnés. On fait l'hypothèse que ces points sont minoritaires et on va essayer de les éliminer de façon récursive.

Le principe de cet algorithme est très simple. Il s'initialise en tirant aléatoirement plusieurs groupes de p paires de points parmi les n paires de points ($p < n$). **Attention**, p doit être supérieur ou égal au nombre que vous avez donné en réponse à la question 2. Pour chaque groupe de points, on calcule une matrice fondamentale. On choisit, comme valeur initiale de la matrice, celle obtenue par le groupe de points amenant au plus petit résidu d'identification (norme du vecteur E).

Le cœur de l'algorithme consiste alors à calculer le résidus d'identification pour les n points, estimer la médiane μ et l'interquartile σ de ces résidus (voir section 3) et de sélectionner tous les points tels que $|\varepsilon_k - \mu| < 2,4\sigma$.

En théorie, on itère cette procédure jusqu'à ce que le groupe de points sélectionnés soit stable. Dans la pratique 5 ou 6 itération suffisent largement.

Travail à effectuer 4.5 : Implantez cette modification et mettez en valeur la robustesse obtenue en sélectionnant quelques mauvais appariements.

