

## Rapport PAIR 4ème année

« Reconstruction 3D à partir d'une paire  
d'images stéréoscopiques non calibrées »

16 mai 2008

Ludovic Arnold 4B21  
Eric Benhaim 4B22  
Arnaud Deschavanne 4B21  
Gilles Lepennec 4B21

suiveurs :

Laurent Baudoin, Antoine Gademer, Benoît Petitpas

## Notations

Sauf mention contraire, les notations suivantes sont utilisées dans la suite du document.

Les matrices sont notées en lettres majuscules (i.e.  $A$ ). Les vecteurs ou points sont représentés en minuscules et en gras (i.e.  $\mathbf{a}$ ). les scalaires seront simplement représentés par une lettre minuscule (i.e.  $a$ ). La  $i^{\text{me}}$  ligne d'une matrice ou d'un vecteur sera représentée par un indice (i.e.  $\mathbf{a}_1, A_1$ ). la numérotation de ces indices commence à 1 et non pas 0 comme dans un célèbre langage de programmation.

Un point  $\mathbf{m}$  de l'image 1 correspondra à un point  $\mathbf{m}'$  dans la seconde.

$F$	matrice fondamentale liant deux images
$\mathbf{E}$	ensemble de points
$\mathbf{m}$	point d'une image
$l$	droite ou demi-droite
$H$	Homographie. matrice 3x3 représentant une application bijective projective
$d_E$	distance euclidienne
$A^t$	matrice transposée de $A$
$A^{-1}$	inverse de $A$ vérifiant $AA^{-1} = A^{-1}A = I$

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Présentation générale</b>	<b>5</b>
2.1	Choix techniques . . . . .	5
2.2	Méthode de développement . . . . .	6
2.3	Géométrie épipolaire . . . . .	7
<b>3</b>	<b>Recherche des points caractéristiques</b>	<b>8</b>
3.1	Enjeux . . . . .	8
3.2	Filtre de Harris : . . . . .	9
<b>4</b>	<b>Mise en relation de points</b>	<b>10</b>
4.1	Enjeux . . . . .	10
4.2	Critère ZNSSD . . . . .	11
4.2.1	Présentation . . . . .	11
4.2.2	Optimisation avec la distance $d_{max}$ . . . . .	13
4.2.3	Optimisation avec pré-calcul des moyennes . . . . .	13
4.2.4	Optimisation avec multithread . . . . .	13
4.2.5	Sélection des meilleurs appariements avec le seuil $t$ . . . . .	14
<b>5</b>	<b>Estimation de la matrice fondamentale</b>	<b>14</b>
5.1	Algorithmes de calcul de la matrice fondamentale . . . . .	14
5.1.1	$F$ est le lien algébrique entre les deux images de notre système stéréoscopique . . . . .	14
5.1.2	Rappel sur la décomposition en valeurs singulières . . . . .	15
5.1.3	Le “8-point algorithm” non-normalisé . . . . .	15
5.1.4	Le “8-point algorithm” normalisé . . . . .	16
5.1.5	Le “7-point algorithm” . . . . .	17
5.1.6	Comparaisons et commentaires . . . . .	17
5.2	RANSAC . . . . .	17
5.2.1	Cas de la régression linéaire . . . . .	17
5.2.2	Généralisation . . . . .	18
5.2.3	Nombre optimal d’itérations . . . . .	18
5.2.4	Estimation de la déviation standard pour le calcul de $F$ . . . . .	20
5.3	LMEDS . . . . .	21
5.3.1	Cas de la régression linéaire . . . . .	21
5.3.2	Généralisation . . . . .	21
5.4	Critères pour RANSAC et LMEDS . . . . .	23
5.4.1	Distance euclidienne $d_E$ . . . . .	23
5.4.2	Erreurs sur les $\mathbf{m}'F\mathbf{m}$ . . . . .	24
5.5	Partition spatiale des points . . . . .	24

<b>6</b>	<b>Rectification linéaire d'une paire d'images non calibrées</b>	<b>25</b>
6.1	Enjeux de la rectification . . . . .	25
6.2	Une rectification à partir de la géométrie épipolaire . . . . .	25
6.2.1	Projection des images dans le plan de rectification et épipôles à l'infini . . . . .	25
6.2.2	La matrice fondamentale d'une paire d'images rectifiées est une matrice antisymétrique . . . . .	25
6.2.3	Avantage de cette méthode : minimiser les distorsions . . . . .	26
6.3	Étapes de la rectification . . . . .	26
6.3.1	Calcul d'un jeu de matrices de rectification compatible avec $F$ . . . . .	26
6.3.2	Déduction de l'ensemble des matrices de rectification avec conservation des disparités . . . . .	27
<b>7</b>	<b>Rectification pour une géométrie épipolaire quelconque.</b>	<b>29</b>
7.1	Principe de fonctionnement . . . . .	29
7.2	Transformation linéaire . . . . .	29
7.2.1	Homographies compatibles . . . . .	29
7.2.2	Moindres carrés . . . . .	30
7.2.3	Minimisation de la distorsion géométrique . . . . .	31
7.3	Transformation non linéaire . . . . .	32
7.3.1	Intervalle angulaire commun . . . . .	32
7.3.2	Choix des angles . . . . .	33
7.3.3	passage en coordonnées polaires . . . . .	33
<b>8</b>	<b>Conclusion générale</b>	<b>35</b>
<b>9</b>	<b>Conclusions personnelles</b>	<b>35</b>
9.1	Arnaud Deschavanne . . . . .	35
9.2	Ludovic Arnold . . . . .	36
9.3	Eric Benhaim . . . . .	36
9.4	Gilles Lepennec . . . . .	37
	<b>Références</b>	<b>39</b>

# 1 Introduction

L'objectif de ce projet est de modéliser une scène ou un objet en 3D à partir de deux images. Les techniques mises en oeuvre font encore l'objet de recherches aujourd'hui, ainsi il est important de noter qu'il n'existe pas de solution finale au problème. Il s'agit d'un problème moderne et complexe au sens mathématique qui nécessite d'estimer un grand nombre de résultats qui ne peuvent pas être raisonnablement calculés.

Nous expliquons dans ce rapport les techniques ou algorithmes utilisés et tentons dans la mesure du possible d'éclairer le lecteur sur les choix effectués.

La réalisation se décompose naturellement en quatre parties distinctes présentées ci-dessous :

1. La *recherche des points caractéristiques* qui consiste à filtrer les deux images afin de ne garder que les points qui ont un attrait dans la caractérisation de l'image.
2. L'*appariement* – également appelé *mise en correspondance* – consiste à coupler des points de la première image avec des points de la seconde.
3. L'*estimation de la matrice fondamentale* permet d'établir une relation matricielle entre les deux images. Lors de la recherche du correspondant d'un point de la première image, Cette relation permet de réduire l'espace de recherche à une droite dans la seconde.
4. La *rectification* a pour objectif de déformer les deux images afin de les mettre dans le même plan. cette étape permet ensuite de faire une triangulation et de retrouver les coordonnées dans l'espace des points qui disposent d'un correspondant.

L'objectif fixé pour ce projet est d'obtenir une librairie permettant de faire de la reconstruction 3D de manière réaliste en utilisant du matériel actuel. Cette librairie est distribuée sous la licence CECILL v2 pour être utilisée et améliorée par les parties intéressées.

## 2 Présentation générale

### 2.1 Choix techniques

Le code est écrit en C++ pour le système d'exploitation GNU/LINUX. Le C++ est retenu pour sa capacité à supporter la programmation de haut niveau tout en maintenant une rapidité comparable au C – quoique légèrement inférieure – à l'exécution.

Sont utilisées La librairie standard C++ connue sous le nom STANDARD TEMPLATE LIBRARY ou plus simplement STL en conjonction avec la librairie BOOST qui regroupe les fonctions les plus courantes, notamment pour la gestion du multithreading et des dates ; la librairie ORFEO TOOLBOX (i.e. OTB ) pour ses fonctions de traitement d'images et pour l'algèbre linéaire ; la librairie

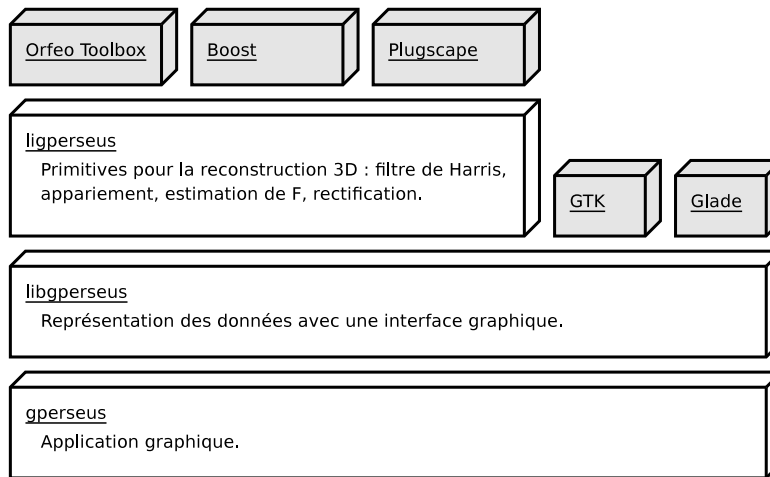


FIG. 1 – Architecture du projet

PLUGSCAPE pour la gestion de la mémoire et le benchmarking; THE GIMP TOOLKIT (i.e. GTK) pour l’interface graphique.

Le projet est configuré avec l’utilitaire CMAKE puis compilé avec un Makefile classique pour générer les binaires suivants :

- une librairie appelée LIBPERSEUS qui fournit les primitives essentielles à la reconstruction 3D.
- une librairie appelée LIBGPERSSEUS laquelle permet l’affichage de données au sein d’une interface GLADE+GTK.
- un exécutable appelé simplement GPERSEUS qui permet graphiquement de travailler sur la reconstruction en 3D d’une paire d’images.

L’application est donc décomposée suivant une architecture en couches comme illustrée en figure 1.

## 2.2 Méthode de développement

La méthode de développement retenue est la méthode dite *en spirale* illustrée par la figure 2 qui suit un processus itératif en quatre phases :

1. Analyse et conception : déterminer ce qu’il faut faire.
2. Implémentation : écriture du code.
3. Vérification : vérifier que le code se conforme à sa spécification.
4. Validation : vérifier que le résultat obtenu est bien celui voulu par l’utilisateur.

Cette méthode de développement est particulièrement adaptée dans un projet comme celui-ci où les enjeux ne sont pas connus à l’avance.

En effet, un développement en V nécessiterait de faire une étape d’analyse et de conception en début de projet. Lorsqu’il s’agit du développement d’un

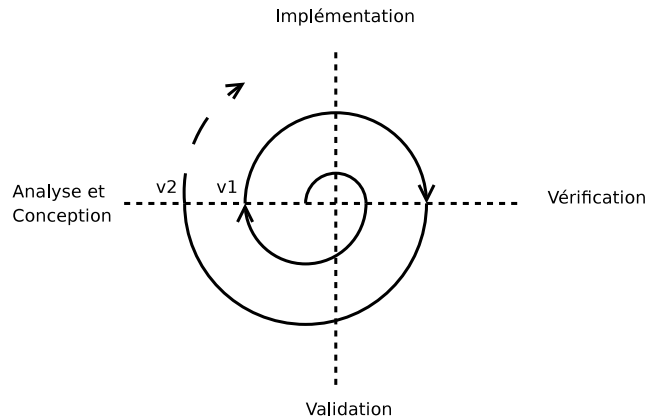


FIG. 2 – Développement en spirale

site web ou d'un logiciel de gestion comptable, les besoins peuvent être définis clairement dès le commencement. De ces besoins découlent immédiatement une analyse et une conception qui seront gardées jusqu'à la fin du projet.

Dans le cas présent, il est impossible de prévoir quels seront les problèmes rencontrés. Il s'agit encore d'un sujet de recherche et pour chaque étape, les possibilités d'implémentation sont multiples, chacune avec ses avantages et défauts particuliers.

### 2.3 Géométrie épipolaire

L'objectif de notre projet est de reconstituer à partir d'images stéréoscopiques une scène en trois dimensions. Afin d'atteindre cet objectif, la géométrie épipolaire est incontournable et constitue un outil idéal en terme de vision stéréoscopique.

La stéréoscopie est basée sur l'existence de deux clichés décrivant une scène commune. Le décalage entre ces prises de vue doit être suffisamment important pour pouvoir calculer une différence entre les deux images mais assez léger pour que les deux images aient une partie importante de la scène en commun.

Ces clichés peuvent se faire au même instant ou non, tant que la mise en relation de la paire d'images stéréoscopiques est possible.

Les notions fondamentales de géométrie épipolaire sont illustrées en figure 3. Dans ce schéma les caméras sont projectives. Pour plus de détails sur la géométrie épipolaire, le lecteur pourra se référer à [Dev97] et [FP07].

Soit  $M$  un point dans l'espace dont on recherche les coordonnées.  $M$  se projette dans l'image 1 en  $\mathbf{m}_1$  et en  $\mathbf{m}_2$  dans l'image 2.  $C_1$  et  $C_2$  sont les centres optiques des images.

La droite  $C_1C_2$  est appelée *ligne de base*. Les *épipôles*  $\mathbf{e}_1$  et  $\mathbf{e}_2$  sont les intersections de la ligne de base avec les plans des images. Ainsi  $l_{m_2}$  est l'intersection du plan  $\Pi$  passant par  $m_1, m_2, C_1$  et  $C_2$  avec l'image 2 et  $l_{m_1}$  est l'intersection

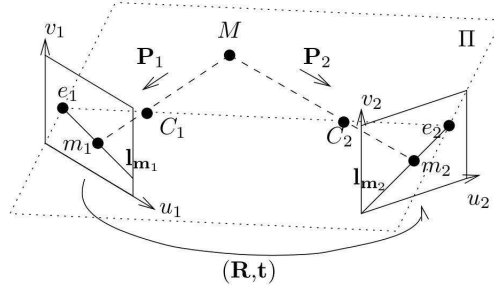


FIG. 3 – La géométrie épipolaire de deux caméras. source [Dev97]

du plan  $\Pi$  avec l'image 1. Les droites  $l_{m_1}$  et  $l_{m_2}$  sont appelées *lignes épipolaires*. Pour n'importe quel point  $M$  de l'espace les droites épipolaires des deux images se croisent en un épipôle. l'ensemble des lignes épipolaires est connu sous le nom de *faisceau épipolaire*.

$(R, t)$  correspond aux valeurs de rotations et translation entre les deux images. Ces valeurs nous sont inconnues car nous avons fait le choix de travailler avec des images non calibrées pour plus de flexibilité.

Nous ne visons donc pas à une précision de reconstruction centimétrique, mais à une reconstruction vrai à 3 facteurs d'échelle près sur les 3 axes.

Cela dit nous possédons les valeurs de focales soit la distance des centres optique aux centres des images, grâce aux informations Exif des images issue d'appareil photos numérique, ce qui nous permettra de pré ajuster les facteurs d'échelles précédemment cités.

### 3 Recherche des points caractéristiques

#### 3.1 Enjeux

**Objectif** définir un ensemble de points caractéristiques pour chaque image.

Dans notre projet nous cherchons les points les plus significatifs des images pour pouvoir par la suite les mettre en correspondance. Les points les plus intéressants se situent sur les contours et en particulier aux coins des objets représentés. En effet, durant la phase suivante d'appariement, il sera important de pouvoir différencier localement un point caractéristique d'un autre pour avoir un traitement suffisamment rapide.

Plusieurs filtres permettent de récupérer des points caractéristiques :

- Le filtre de Canny pour les contours des formes dans l'image, dont un exemple est donné en figure 4.



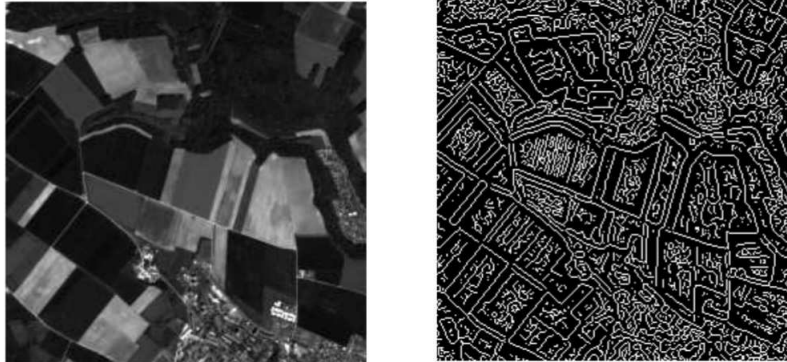


FIG. 4 – Exemple d’application du filtre de Canny.



FIG. 5 – Exemple d’application du filtre de Harris, les points caractéristiques étant indiqués en bleu.

- Le filtre de Harris pour les coins des formes dans l’image, dont un exemple est donné en figure 5.

Nous devons détecter les coins des deux images stéréoscopiques, ce qui revient à appliquer l’un de ces filtres deux fois afin de mettre les points désignés comme des coins en correspondance par la suite.

### 3.2 Filtre de Harris :

Nous avons choisi d’utiliser le filtre de Harris qui permet de limiter le nombre de points sélectionnés. Le filtre de Harris doit être appliqué à chaque image et nous avons pu multithreader cette partie du code afin de diviser par presque deux le temps d’exécution sur des machines double-coeur.

Le filtre de Harris utilisé est fourni par la librairie OTB et requiert trois paramètres en entrée :

$\sigma_D$             facteur de dérivation

$\sigma_I$	facteur d'intégration effectuant un lissage de l'image
$\alpha$	facteur servant à limiter le bruit du filtre
$t$	seuil

L'équation utilisé par OTB est donnée par

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) \star \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y^2(\mathbf{x}, \sigma_D) \\ L_x L_y^2(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}$$

La sortie du filtre est  $det(\mu) - \alpha trace^2(\mu)$ . Les points de Harris sont ensuite extraits avec le seuil  $t$ .

Les paramètres sont choisis expérimentalement et la meilleure combinaison est différente pour chaque image. Nous avons néanmoins déterminé un ensemble de paramètres qui donne en général de bons résultats sur la plupart des images.

$$\sigma_D = 1, 2 \quad \sigma_I = 6 \quad \alpha = 1 \quad t = 220$$

Pour une mauvaise combinaison de paramètres il est possible que de trop nombreux points soient sélectionnés, ou au contraire qu'aucun point ne passe au dessus du seuil. Dans certains cas, et en particulier pour les valeurs de  $\alpha$  élevées, les points détectés s'éloignent de part et d'autre des coins. Pour déterminer ces paramètres nous nous sommes inspirés de la documentation ORFEO et de l'équation précisée ci-dessus et avons terminé par une analyse empirique.

Une fois de bons points sélectionnés, ils doivent être mis en relation pour établir un lien entre les deux images.

## 4 Mise en relation de points

### 4.1 Enjeux

**Objectif** mettre en correspondance les points d'une image à l'autre.

L'objectif étant d'atteindre la 3D, nous devons mettre en correspondance les images, c'est-à-dire repérer quel coin de l'image 1 correspond un coin donné de l'image 2. Nous avons choisi les critères suivants pour mettre en relation les points des deux images.

- Ressemblance sur un voisinage du pixel.
- Distance entre un point et son point associé.

Afin de noter la ressemblance du voisinage de deux pixels nous comparons une fenêtre carrée de côté  $2d + 1$  centrée sur les pixels à comparer. Cette fenêtre est appelée *fenêtre de corrélation* est représentée en figure 6. Partant d'un point de l'image 1, seuls les points se situant à une distance inférieure à une distance passée en paramètre  $d_{max}$  sont étudiés. Pour chaque point caractéristique dans la première image, tous ses correspondants potentiels sont notés. Le correspondant potentiel disposant de la meilleure note est gardé si, de surcroît, sa note est meilleure qu'un seuil donné. Ces informations sont récapitulées dans l'algorithme

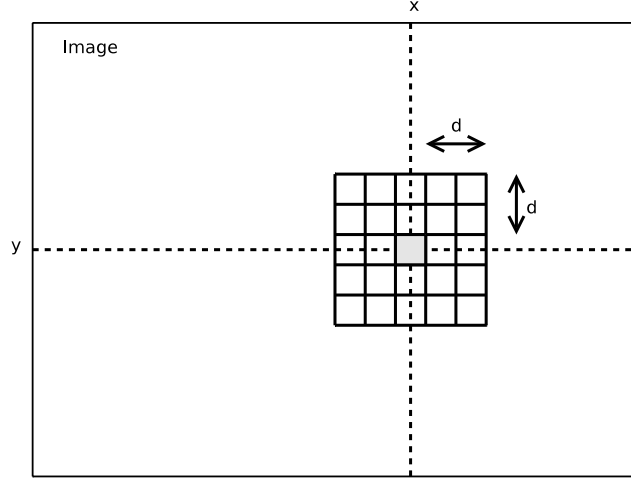


FIG. 6 – Fenêtre de corrélation de paramètre  $d$  pour un pixel  $(x, y)$

1 qui pour un nombre de points appariés  $n$  a une complexité en  $O(n^2)$ . Avec une telle complexité, la mise en correspondance est le point le plus critique de la chaîne de traitement au niveau du temps de calcul.

Il existe de nombreux critères tels que ZNCC le ZNSSD inspirés du CC et SSD. En voici les équations :

$$CC_d(x_1, y_1, x_2, y_2) = \sum_{i,j} I_1(x_1 + i, y_1 + j) I_2(x_2 + i, y_2 + j)$$

$$SSD_d(x_1, y_1, x_2, y_2) = \sum_{i,j} (I_1(x_1 + i, y_1 + j) - I_2(x_2 + i, y_2 + j))^2$$

$$ZNCC_d(x_1, y_1, x_2, y_2) = \frac{\sum_{i,j} (I_1(x_1 + i, y_1 + j) - \bar{I}_1)(I_2(x_2 + i, y_2 + j) - \bar{I}_2)}{\sqrt{\sum_{i,j} (I_1(x_1 + i, y_1 + j) - \bar{I}_1)^2} \sqrt{\sum_{i,j} (I_2(x_2 + i, y_2 + j) - \bar{I}_2)^2}}$$

$$ZNSSD_d(x_1, y_1, x_2, y_2) = \frac{-\sum_{i,j} ((I_1(x_1 + i, y_1 + j) - \bar{I}_1) - (I_2(x_2 + i, y_2 + j) - \bar{I}_2))}{\sqrt{\sum_{i,j} (I_1(x_1 + i, y_1 + j) - \bar{I}_1)^2} \sqrt{\sum_{i,j} (I_2(x_2 + i, y_2 + j) - \bar{I}_2)^2}}$$

## 4.2 Critère ZNSSD

### 4.2.1 Présentation

Le critère ZNSSD – ou Zero mean sum of squared difference –, que nous avons retenu pour l'appariement, décrit la somme des différences des intensités,

---

**Algorithm 1** Mise en correspondance des points caractéristiques

---

**Entrées**

$\mathbf{E}_1$  ensemble de points caractéristiques sélectionnés dans l'image 1  
 $\mathbf{E}_2$  ensemble de points caractéristiques sélectionnés dans l'image 2  
 $d_{max}$  distance maximum entre deux points appariés  
 $t$  seuil de sélection

**Sorties**

$\mathbf{S}$  ensemble de couples de points caractéristiques appariés

**Début**

$S \leftarrow \emptyset$

**Pour**  $\mathbf{a} \in \mathbf{E}_1$  :

**Pour**  $\mathbf{b} \in \mathbf{E}_2$  :

$note_{best} \leftarrow 0$

**Si**  $d_E(\mathbf{a}, \mathbf{b}) < d_{max}$  **alors**

            calculer  $note(\mathbf{a}, \mathbf{b})$

**Si**  $note(\mathbf{a}, \mathbf{b}) > note_{best}$  **alors**

$note_{best} \leftarrow note(\mathbf{a}, \mathbf{b})$

$\mathbf{b}_{best} \leftarrow \mathbf{b}$

**Si**  $note_{best} > t$  **alors**

$\mathbf{S} \leftarrow \mathbf{S} \cup \{(\mathbf{a}, \mathbf{b}_{best})\}$

**Fin**

---

filtrées par la moyenne sur la fenêtre. Le tout est normalisé par la variance des intensités. De ce fait, le critère ZNSSD représente une différence entre des voisinages de pixels et on cherche donc des couples de points caractéristiques dont la note est la plus proche de 0.

Afin de ne garder que les meilleurs points, nous fixons un seuil modifiable dans le logiciel fixé à 0.1. Ceci permet de faire un premier tri, et de ce fait, on observe en général que seulement 40% des coins sont appariés.

Nous aurions pu utiliser les critères ZNCC ou ZSSD cependant ZNSSD est normalisé. Ici la normalisation est capitale car nous voulons travailler sur toute sortes d'images, plus ou moins bruitées. le défaut du critère ZNSSD est d'être coûteux en temps de calcul en raison des racines carrés et puissances calculées dans la variance. Nous avons donc dû optimiser son déroulement.

#### 4.2.2 Optimisation avec la distance $d_{max}$

Afin d'optimiser ce processus, plutôt que de chercher la correspondance d'un point caractéristique avec tous les autres points caractéristiques de l'image, nous réduisons l'espace de recherche à un disque de rayon  $d_{max}$ . Cette zone est variable d'une image à l'autre.

Dans le cas où le déplacement d'une image à l'autre est faible, une petite zone de recherche convient. Chercher avec une distance trop grande serait inutile et pourrait introduire des erreurs dans la définition des matches.

Au contraire, si l'utilisateur effectue des photos avec un fort zoom ou une translation importante, le déplacement sera d'autant plus accentué. Dans ce cas, il faudra augmenter la zone de recherche. De même, selon la résolution de l'image, la zone de recherche varie dans des proportions non négligeables. Ainsi, sur des images de dimensions 640x480, une zone convenable pourrait être de 200 pixels, mais sur des photographies 3000x2000 Il faut compter sur une zone de 400 pixels au moins. Ce choix est donc pour le moment laissé à l'utilisateur et fixé par défaut à 400 pixels.

#### 4.2.3 Optimisation avec pré-calcul des moyennes

De plus nous observons ici que pour chaque point à appairier, une moyenne doit être calculée. Nous pré-calculons donc toutes les moyennes avant l'algorithme principal de corrélation afin d'accélérer le calcul.

#### 4.2.4 Optimisation avec multithread

Pour tirer parti des capacités actuelles et en particulier des processeurs multicœurs, la mise en correspondance est faite en parallèle. L'ensemble des points à appairier  $\mathbf{E}_1$  de la première image est réparti dans quatre sous ensembles constituant une partition de  $\mathbf{E}_1$ . Pour chacun des ces sous ensembles, la mise en correspondance est faite dans un thread particulier. Avec ces méthodes d'optimisation et en utilisant l'optimisation du compilateur, les temps de calculs sont grandement réduits pour tomber à une durée de l'ordre de la dizaine de secondes alors qu'ils peuvent durer plus d'une minute sans optimisation.

### 4.2.5 Sélection des meilleurs appariements avec le seuil $t$

Nous n'acceptons les correspondants que si leur note est en dessous du seuil  $t$ . De ce fait, nous ne gardons qu'un certain pourcentage de couples. Le matching de point est capital dans le déroulement du projet car il définit les points correspondants qui nous serviront à calculer la matrice fondamentale mettant en relation les deux images.

Afin de calculer cette matrice, nous n'avons besoin que de huit points appariés, l'information doit donc être filtrée au maximum. Malgré cette sélection des meilleurs couples de points, beaucoup de résultats sont mauvais et sont appelés *outliers*. Les bons couples en revanche sont appelés *inliers*.

A la sortie du ZNSSD on considère avoir 50% d'inliers et 50% d'outliers. Ceci constitue une erreur importante malgré nos efforts visant à limiter le nombre d'outliers : normalisation du ZNSSD, distance de recherche limitée à  $d_{max}$ , application d'un seuil sur le ZNSSD, taille de fenêtre de corrélation adaptée.

Ce qui singularise une zone autour d'un pixel, sont des valeurs comme la moyenne de l'intensité de couleur, la variance sur la fenêtre. Plus la fenêtre est petite, moins la zone est singularisée et plus elle est grande, plus celle-ci est caractéristique du pixel en question jusqu'à un certain point. Par défaut, nous avons fixé le paramètre  $d$  à 4 ce qui correspond à une fenêtre de 9 pixels de côté. Si nous avons mis une taille de 50 pixels alors la moyenne aurait été noyé sous l'information et non plus caractéristique de ce pixel, mais aussi de tous ceux dans son proche voisinage. Le choix de la taille de fenêtre de corrélation est donc très important dans le calcul des matches.

## 5 Estimation de la matrice fondamentale

### 5.1 Algorithmes de calcul de la matrice fondamentale

#### 5.1.1 $F$ est le lien algébrique entre les deux images de notre système stéréoscopique

Afin de lier une paire d'images stéréoscopique dans le cadre d'une géométrie épipolaire, nous utilisons une matrice fondamentale appelée usuellement  $F$ . Cette matrice de taille  $3 * 3$  doit être connue pour effectuer le processus de rectification.

Afin de remplir cette matrice nous devons obtenir des appariements valides.  $F$  relie les images par les propriétés suivantes :

$$\mathbf{m}'^t F \mathbf{m} = 0$$

où  $\mathbf{m} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$  et  $\mathbf{m}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$

avec  $\mathbf{m}$  et  $\mathbf{m}'$  qui sont respectivement des points de l'image 1 et de l'image 2 correspondants.

Cette équation est vraie théoriquement pour toutes bonnes paires de points correspondant.

D'autre part, la géométrie épipolaire nous enseigne que :  $F\mathbf{e}_1 = 0$  et  $\mathbf{e}'_2 F = 0$  où  $\mathbf{e}_1$  et  $\mathbf{e}_2$  sont les épipôles respectifs des deux images.

Passons maintenant aux différents calculs de  $F$ .

### 5.1.2 Rappel sur la décomposition en valeurs singulières

La décomposition en valeurs singulières (i.e. Singular Value Decomposition notée *SVD*) est une généralisation de la décomposition en éléments propres qui peut être utilisée pour analyser des matrices rectangulaires (voir [Str03]).

La décomposition en valeurs singulières fut développée à l'origine par les mathématiciens étudiant la géométrie différentielle, qui désiraient déterminer si une forme bilinéaire réelle pouvait en égaler une autre par des transformations orthogonales indépendantes des deux espaces concernés.

Le but principal de la SVD est de décomposer une matrice rectangulaire en deux matrices orthogonales et une matrice diagonale.

$$svd(A) = P\Delta Q^t$$

- $P$  : vecteurs propres (normalisés) de la matrice  $AA^t$  i.e.  $P^tP = Id$  Les colonnes de  $P$  sont appelées les vecteurs singuliers gauches de  $A$
- $Q$  : vecteurs propres (normalisés) de la matrice  $AA^t$  i.e.  $Q^tQ = Id$  Les colonnes de  $Q$  sont appelées les vecteurs singuliers droits de  $A$
- $\Delta$  : la matrice diagonale formée des valeurs singulières,  $\Delta = \Lambda^{1/2}$  avec  $\Lambda$  : la matrice diagonale formée des valeurs propres de la matrice  $AA^t$  ou  $A^tA$ .

Un réel positif  $\sigma$  est appelé valeur singulière de  $M$  si et seulement s'il existe un vecteur unitaire  $u$  dans  $K^m$  et un vecteur unitaire  $v$  dans  $K^n$  tel que :  $Mv = \sigma u$  et  $M^t u = \sigma v$ .

Les vecteurs  $u$  et  $v$  sont appelés vecteurs singuliers

### 5.1.3 Le "8-point algorithm" non-normalisé

Comme son nom l'indique, cet algorithme nécessite huit couples de points appariés, soit huit points sur chaque image.

Nous avons précédemment vu que  $\mathbf{m}'^t F \mathbf{m} = 0$ . Avec

$$F = \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{33} & F_{33} \end{pmatrix} \text{ avec } \mathbf{m} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ et } \mathbf{m}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

On a donc :

$$x'x F_{11} + x'y F_{12} + x' F_{13} + y'x F_{21} + y'y F_{22} + y' F_{23} + x F_{31} + y F_{32} + F_{33} = 0$$

Nous définissons la matrice  $A$  comme suit

$$A = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & x_1 & y_1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & x_n & Y_n & 1 \end{bmatrix}$$

où  $x_i$ , et  $y_i$  sont les coordonnées dans l'image des huit points matchés de l'image 1 et  $x'_i$  et  $y'_i$  sont les coordonnées dans l'image des huit points matchés de l'image 2 avec ceux de l'image 1.

Nous voulons obtenir  $Af = 0$ .

On pose  $SVD(A) = U * \Delta * Q^t$ . Nous remplissons une matrice  $F_2$  avec la 8<sup>me</sup> colonne de  $Q$ .  $F_2$  est donc de taille  $3 * 3$  car  $Q$  est de taille  $9 * 8$  tout comme  $A$ .

On pose  $SVD(F_2) = U * \Delta * Q^t$ , avec  $\Delta$  de taille  $3 * 3$ . Afin d'obtenir une matrice fondamentale de rang 2, nous forçons le rang en posant  $\Delta(3, 3) = 0$  dans la  $SVD$  de  $F_2$ .

Cette méthode offre de bons résultats, que nous comparons par la suite.

#### 5.1.4 Le "8-point algorithm" normalisé

La normalisation de cet algorithme s'effectue en transformant les coordonnées des matchs employés.

Ainsi on calcul deux matrices de normalisations  $T_1$  et  $T_2$  de taille  $3 * 3$ .

$T_1$  et  $T_2$  sont remplies avec les variances et moyennes des matchs de l'image 1 et de l'image 2 respectivement.

Une matrice de normalisation s'écrit donc ainsi :

$$T = \begin{bmatrix} \frac{\sqrt{2}}{var} & 0 & \frac{-mean_x}{var} \\ 0 & \frac{\sqrt{2}}{var} & \frac{-mean_y}{var} \\ 0 & 0 & 1 \end{bmatrix}$$

où  $mean_x$  et  $mean_y$  sont les moyennes des coordonnées  $x$  et  $y$  des matchs.

La variance  $var$  est ici calculée de la manière suivante

$$var = \sum (\sqrt{(x - mean_x)^2 + (y - mean_y)^2})$$

Enfin nous stockons les points dans une matrice  $U$  de taille  $3 * 8$ .

Les points normalisés sont alors égaux à  $TU$ .

Nous appellerons les deux matrices de points normalisés  $V_1$  et  $V_2$ .

Ensuite, de la même manière que dans le "8-point algorithm" non normalisé, nous remplissons la matrice  $A$  mais avec les coordonnées des points normalisés.

Au final du calcul de  $F_2$ , nous appliquons la transformation :  $F = T_2^t * F_2 * T_1$  qui a pour effet de dé-normaliser les coordonnées.

Les résultats sont bien meilleurs avec cette méthode. En effet, la normalisation a pour conséquence de replacer les points autour d'un centroïde, ce qui clarifie l'information.



### 5.1.5 Le “7-point algorithm”

Une autre méthode existe : c’est celle du “7-point algorithm”. Elle est très précise mais trop sensible aux bruits et erreurs vis à vis des matchs. Voici son fonctionnement :

On remplit une matrice  $A$  en utilisant la même méthode que le “8-point”, ainsi la matrice  $A$  est de taille  $8 * 9$ .

On a bien sûr la relation  $SVD(A) = U * \Delta * Q^t$ .

En utilisant les deux dernières colonnes de  $Q$ , on cherche à résoudre  $\det(F_1 + \lambda F_2) = 0$  où  $F_1$  et  $F_2$  sont des matrices  $3 * 3$  dans lesquelles on range les vecteurs colonnes précédemment cités.

Ceci nous amène à résoudre une équation du  $3^{me}$  degré avec laquelle  $\lambda$  est déterminé.

Ainsi  $F(i, j) = F_1(i, j) + \lambda * F_2(i, j)$ .

### 5.1.6 Comparaisons et commentaires

Nous avons développé ces trois méthodes car nous cherchions à obtenir une matrice  $F$  aussi précise que possible.

Cela se juge sur la minimisation de  $\mathbf{m}^t F \mathbf{m}$ , la distance des matchs à leurs lignes épipolaires et un bon calcul de position des épipôles.

Nous observons une précision bien plus nette sur le critère  $\mathbf{m}^t F \mathbf{m}$  avec le “7-point”. Le “8-point” normalisé est quant à lui meilleur que le non normalisé : la normalisation est donc bien choisie.

On pourrait donc considérer que le “7-point” est meilleur, cependant celui-ci est extrêmement sensible et les lignes épipolaires qui sont censées se croiser ne le font que rarement, bien qu’elles approchent de très près les matchs.

Pour plus de stabilité, nous choisissons donc le “8-point normalized”.

## 5.2 RANSAC

le RANSAC ou encore RANDOM SAMPLE CONSENSUS est un algorithme publié en 1981 par Fischler et Bolles. C’est une méthode itérative pour estimer des paramètres d’un modèle mathématique d’après une observation de données qui contient des outliers. on part du principe que les inliers sont explicables par des paramètres du modèle et que les outliers sont des points qui ne vérifient pas le modèle. De plus le lot de donnée peut être bruité. Les outliers peuvent être des valeurs extrêmes du bruit, des mesures erronées ou encore une mauvaise interprétation des données.

le RANSAC est capable avec un lot de quelques inliers de montrer qu’il existe une méthode qui permet d’estimer les paramètres du modèle qui correspondent aux données globales de façon optimale.

### 5.2.1 Cas de la régression linéaire

Le RANSAC est couramment utilisé en régression linéaire. L’ensemble de données considéré contient des inliers et des outliers, c’est-à-dire respectivement des

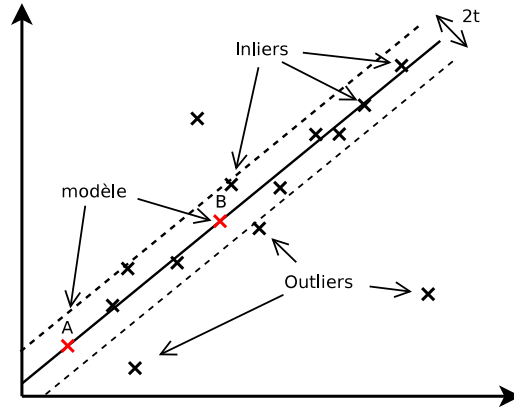


FIG. 7 – Application du RANSAC à un cas de régression linéaire

données qui pourront passer par la droite et des données qui ne pourront pas passer par la droite. Le RANSAC estime la meilleure droite qui représente les inliers, en se basant sur le fait que probabilité d'avoir des inliers dans l'échantillon de données est suffisamment grande. Une représentation graphique du problème de régression linéaire résolu avec le RANSAC est donné en figure 7.

### 5.2.2 Généralisation

Le RANSAC a besoin de trois paramètres pour fonctionner : un ensemble de données, un seuil  $t$  et un modèle paramétrique qui correspond aux données observées.

L'algorithme fonctionne de manière itérative pour  $n$  itérations. On sélectionne d'abord un sous échantillon aléatoire des données. Cet échantillon constitue une base pour calculer un modèle de l'ensemble de données. Une fois ce modèle calculé, tous les résidus  $r_i$  (i.e. l'écart d'une donnée  $i$  au modèle) de l'ensemble de données sont calculés avec ce modèle. C'est ici que rentre en jeu le seuil  $t$  donné. Le nombre d'inliers c'est-à-dire le nombre de données pour lesquelles le résidu est inférieur à  $t$ , est calculé. Finalement, le modèle pour lequel le nombre d'inliers est maximal est gardé comme étant la meilleure estimation. L'algorithme 2 donne les détails de l'implémentation du RANSAC.

### 5.2.3 Nombre optimal d'itérations

Le nombre d'itérations  $n$  peut être déterminé à partir d'un résultat théorique. Étant donnée  $p$  la probabilité voulue d'avoir un modèle basé uniquement sur des inliers, et  $q$  le nombre de données qui composent un échantillon (i.e.  $q = 2$  pour une régression linéaire,  $q = 8$  pour une estimation de  $F$  avec le 8-point-algorithm) et  $r$  le ratio d'inliers dans l'ensemble de données, on peut calculer le nombre d'itérations.

---

**Algorithm 2** RANSAC généralisé

---

**Entrées**

- E** l'ensemble des données.  
 $n$  le nombre d'itérations nécessaires pour obtenir notre estimation  
 $q$  le nombre de données par échantillon  
 $t$  une valeur seuil qui détermine si un point est un inlier

**Sorties**

- $m$  le meilleur modèle  
 $\mathbf{E}_{modele}$  les données permettant de calculer le modèle  $m$   
 $\mathbf{E}_{consensus}$  l'ensemble des inliers pour le modèle  $m$

**Début**

$w_{modele} \leftarrow 0$

1. Pour  $k$  de 1 à  $n$  :
  - (a)  $\mathbf{E}_{sample} \leftarrow n$  valeurs sélectionnées aléatoirement dans  $\mathbf{E}$ .
  - (b) calculer un modèle à partir des données de  $\mathbf{E}_{sample}$
  - (c)  $w \leftarrow 0$
  - (d) Pour chaque donnée  $i$  de  $\mathbf{E}$ 
    - i. calculer  $r_i$  le résidu de la donnée  $i$  par rapport au modèle calculé en (b)
    - ii. Si  $r_i < t$  alors  $w \leftarrow w + 1$
  - (e) Si  $w > w_{modele}$  alors  $\mathbf{E}_{modle} \leftarrow \mathbf{E}_{sample}$
2. calculer le meilleur modèle  $m$  à partir de  $\mathbf{E}_{modle}$
3.  $E_{consensus} \leftarrow \emptyset$
4. Pour  $i$  une donnée de  $\mathbf{E}$ 
  - (a) Calculer  $r_i$  le résidu de la donnée  $i$  par rapport au modèle  $m$ .
  - (b) Si  $r_i < t$  alors  $E_{consensus} \leftarrow E_{consensus} \cup \{i\}$

---

**Fin**

---

La valeur  $n$  recherchée correspond donc au nombre minimum d'échantillons aléatoires de taille  $q$  requis pour obtenir au moins un des échantillons exempt d'outliers.

si  $r$  est la probabilité d'obtenir un inlier, alors  $r^q$  est la probabilité qu'un échantillon de taille  $q$  soit exempt d'outliers. On en déduit la probabilité qu'au moins une des  $q$  données soit un outlier égale à  $1 - r^q$ . Cette probabilité élevée à la puissance  $n$  correspond à la probabilité qu'après  $n$  échantillons, aucun de ces échantillons ne soit exempt d'outlier. On a immédiatement  $1 - p = (1 - r^q)$  et finalement l'équation tant convoitée

$$n = \frac{\log(1 - p)}{\log(1 - r^q)}$$

N'oublions pas que ce résultat part du principe que tous les échantillons sont sélectionnés indépendamment, c'est-à-dire que l'on est dans le cas d'un tirage aléatoire avec remise.

L'avantage du Ransac est qu'il est capable d'effectuer une estimation robuste des paramètres du modèle, c'est-à-dire que ses résultats sont bon même avec un taux élevé d'outliers.

#### 5.2.4 Estimation de la déviation standard pour le calcul de $F$

Lorsqu'on veut automatiser au maximum le fonctionnement du Ransac, il devient intéressant de pouvoir déterminer le seuil  $t$  à partir des données. Une technique (voir [HZ04]) consiste à utiliser l'écart type des données pour fixer le seuil. Dans le cas du calcul de la matrice fondamentale, un seuil correspondant à ce qu'il y ait 5% de chances qu'un inlier soit inutilement rejeté, est donné par  $t^2 = 3.84\sigma^2$ .

Malheureusement nous sommes en train d'estimer les paramètres de notre modèle et l'écart type ne nous est pas donné. Nous utilisons donc une estimation robuste de l'écart type basée sur  $MAD$  (i.e. Median Absolute Deviation, voir [Wik]), la valeur absolue des écarts à la médiane définie par  $MAD = median_i(|X_i - median_j(X_j)|)$  et définie dans notre cas de calcul de la matrice fondamentale par  $MAD = median_i(r_i)$ .

$$\hat{\sigma} = \Phi^{-1}(0.75) median_i(r_i)$$

avec  $\Phi^{-1}$  la fonction de répartition inverse d'une loi normale et  $\Phi^{-1}(0.75) = 1.4826$ . De plus, comme décrit dans [RL87], il est possible d'améliorer l'estimation de l'écart type lorsque la taille de l'ensemble de données est petite en ajoutant un facteur  $(1 + \frac{5}{n - q})$ . On obtient l'estimation finale de l'écart type

$$\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n - q}\right) median(r_i)$$

On remarque néanmoins que fixer un seuil  $t^2 = 3.84\hat{\sigma}^2$  n'est pas une solution efficace. En effet, avec un plus grand nombre d'outliers, la déviation standard

estimée devient plus grande et le seuil moins efficace. Il est donc nécessaire de réduire cet impact sur le seuil.

Ces éléments étant connus, nous avons choisi la fonction racine carré pour limiter l'impact de l'augmentation de l'écart type estimé. On fixe donc notre seuil avec  $t^2 = 3.84\sqrt{\hat{\sigma}^2}$ , soit  $t = \sqrt{3.84}\hat{\sigma}$ . La fonction racine carré nous garantit que pour de grands écarts types,  $t$  sera relativement petit, alors que pour  $\sigma$  proche de 1, on aura bien  $t^2 \approx 3.84\hat{\sigma}^2$ .

### 5.3 LMEDS

Un des principaux désavantages du RANSAC est sa dépendance au seuil  $t$ , qui détermine à quelle distance doit être un point du modèle estimé pour être considéré comme un inlier.

Pour palier à cette faiblesse, le RANSAC a été modifié afin de ne plus avoir de dépendance au seuil. Deux principaux algorithmes sont basés sur ce RANSAC modifié : le LTS ou LEAST TRIMMED SQUARES (voir [Far05]), que nous n'avons pas étudié, et le LMEDS ou LEAST MEDIAN OF SQUARES. Ces algorithmes fonctionnent globalement de la même façon que le RANSAC, malheureusement comme nous le verrons par la suite, LMEDS ne fonctionne que s'il y a au moins 50% d'inliers. L'algorithme LMEDS est décrit dans [Zha].

#### 5.3.1 Cas de la régression linéaire

Tout comme dans l'exemple de régression linéaire avec le RANSAC, LMEDS a pour objectif d'estimer une droite à partir d'un certain nombre de points dont certains sont représentatifs du modèle et d'autres non. Le LMEDS trouve la droite qui minimise la médiane des résidus au carré. Ceci explique pourquoi LMEDS ne permet pas de faire une régression efficace avec plus de 50% d'outliers. S'il y a plus de 50% d'outliers, la médiane des résidus correspond à un outlier et on juge donc la qualité de notre estimation sur un critère qui n'a pas de sens. La régression linéaire avec l'algorithme LMEDS est illustrée en figure 8.

#### 5.3.2 Généralisation

Comme dans le cas du RANSAC en 5.2.2, nous sélectionnons  $n$  échantillons de  $q$  données sur un ensemble de données  $\mathbf{E}$  avec un ratio minimum d'inliers  $r$ . le nombre  $n$  d'itérations est calculé avec la même équation que pour le RANSAC (voir 5.2.3) c'est à dire  $n = \log(1 - p) / \log(1 - r^q)$ .

Pour chaque itération un modèle est calculé à partir de l'échantillon aléatoire. Il s'agit ensuite de calculer tous les résidus  $r_i$  et la médiane qui donne son nom à l'algorithme :  $median_i(r_i^2)$ . Une fois toutes les médianes calculées, le modèle retenu est celui pour lequel la médiane des résidus au carré est la plus petite. Il est intéressant de noter que dans notre cas d'utilisation de LMEDS, les résidus sont positifs et qu'il est par conséquent totalement inutile d'en prendre la racine carré.

---

**Algorithm 3** LMEDS généralisé

---

**Entrées**

- E** l'ensemble des données.  
 $n$  le nombre d'itérations nécessaires pour obtenir notre estimation  
 $q$  le nombre de données par échantillon

**Sorties**

- $m$  le meilleur modèle  
 $\mathbf{E}_{modele}$  les données permettant de calculer le modèle  $m$   
 $\mathbf{E}_{inliers}$  l'ensemble des inliers pour le modèle  $m$

**Début** $mediane_{best} \leftarrow \infty$ 

1. Pour  $k$  de 1 à  $n$  :
  - (a)  $\mathbf{E}_{sample} \leftarrow n$  valeurs sélectionnées aléatoirement dans  $\mathbf{E}$ .
  - (b) calculer un modèle à partir des données de  $\mathbf{E}_{sample}$
  - (c) Pour chaque donnée  $i$  de  $\mathbf{E}$ 
    - i. calculer  $r_i$  le résidu de la donnée  $i$  par rapport au modèle calculé en (b)
  - (d) Calculer  $mediane(r_i^2)$
  - (e) Si  $mediane(r_i^2) < mediane_{best}$  alors
    - i.  $\mathbf{E}_{modele} \leftarrow \mathbf{E}_{sample}$
    - ii.  $mediane_{best} \leftarrow mediane(r_i^2)$
2. calculer le meilleur modèle  $m$  à partir de  $\mathbf{E}_{modele}$
3.  $E_{inliers} \leftarrow \emptyset$
4.  $\hat{\sigma} \leftarrow 1.4826 \left(1 + \frac{5}{n-q}\right) \sqrt{mediane_{best}}$
5.  $t \leftarrow \hat{\sigma}$
6. Pour  $i$  une donnée de  $\mathbf{E}$ 
  - (a) Calculer  $r_i$  le résidu de la donnée  $i$  par rapport au modèle  $m$ .
  - (b) Si  $r_i < t$  alors  $E_{inliers} \leftarrow E_{inliers} \cup \{i\}$

**Fin**

---

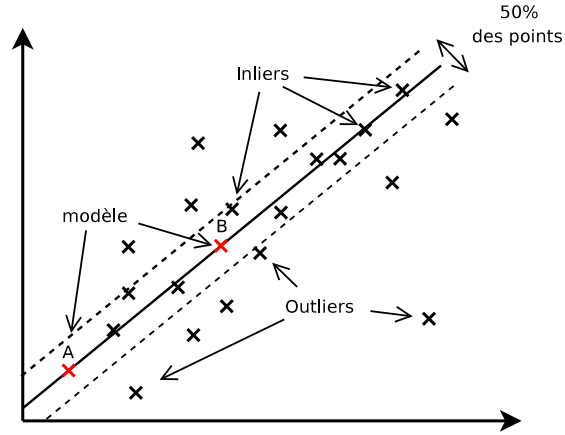


FIG. 8 – Régression linéaire avec LMEDS

## 5.4 Critères pour RANSAC et LMEDS

RANSAC et LMEDS sont basés sur un calcul des résidus, c'est-à-dire la distance d'une donnée au modèle. Nous expliquons ici comment sont calculés ses résidus pour l'estimation de la matrice fondamentale  $F$ .

### 5.4.1 Distance euclidienne $d_E$

Le premier calcul que nous utilisons est la distance euclidienne. Comme nous avons pu le voir en 5.1.1, la relation qui caractérise la matrice fondamentale est

$$\mathbf{m}^t F \mathbf{m} = 0$$

Pour un point  $\mathbf{m}$  de l'image 1, cette relation définit un ensemble de points qui correspond à une ligne épipolaire dans la deuxième image. En effet, nous

avons pour  $\mathbf{m}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$

$$[x' \ y' \ 1] F \mathbf{m} = 0$$

En calculant  $\begin{bmatrix} a \\ b \\ c \end{bmatrix} = F \mathbf{m}$  nous obtenons immédiatement

$$[x' \ y' \ 1] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = ax' + by' + c = 0$$

Cette équation correspond à l'équation de la droite épipolaire recherchée. Il nous suffit maintenant de calculer la distance euclidienne de  $\mathbf{m}'$  à  $F \mathbf{m}$  puis celle de  $\mathbf{m}$  à  $F^t \mathbf{m}'$  à cette droite pour obtenir le résidu souhaité.

$$r_{aux}(F, \mathbf{m}, \mathbf{m}') = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}}$$

$$r_{d_E}(F, \mathbf{m}, \mathbf{m}') = r_{aux}(F, \mathbf{m}, \mathbf{m}') + r_{aux}(F^t, \mathbf{m}, \mathbf{m}')$$

Ce critère est très efficace et permet d'estimer  $F$  de manière robuste.

#### 5.4.2 Erreurs sur les $\mathbf{m}'F\mathbf{m}$

Comme second critère, nous utilisons la valeur de  $\mathbf{m}'F\mathbf{m}$ . Cette valeur doit tendre vers 0 pour avoir une bonne estimation de  $F$ . le résidu est trivialement calculé avec la formule :

$$r_{aux}(F, \mathbf{m}, \mathbf{m}') = \mathbf{m}'F\mathbf{m}$$

$$r_{\mathbf{m}'F\mathbf{m}}(F, \mathbf{m}, \mathbf{m}') = r_{aux}(F, \mathbf{m}, \mathbf{m}') + r_{aux}(F^t, \mathbf{m}, \mathbf{m}')$$

Ce critère est très efficace pour obtenir une erreur  $\mathbf{m}'F\mathbf{m}$  très basse, ce qui est utile pour l'étape suivante. toutefois, ce critère à une nette tendance à favoriser à mauvais escient des calculs de  $F$  pour lesquels l'épipôle se situe au centre de l'image. Il est donc utile d'employer ce critère pour réduire une première fois l'ensemble de points appariés à des couples de points pour lesquels la valeur  $\mathbf{m}'F\mathbf{m}$  est basse pour ensuite utiliser le critère plus classique mettant en jeu la distance euclidienne.

### 5.5 Partition spatiale des points

Pour obtenir une plus grande qualité de l'estimation de  $F$ , il est nécessaire de choisir huit points dans des régions suffisamment distantes. En effet, dans le cas de la régression linéaire, on voit immédiatement pourquoi deux points proches conduisent rarement à une estimation correcte de la droite recherchée.

Les images sont donc réparties en huit zones, et lors de la sélection d'une échantillon par le RANSAC ou LMEDS, chacun des huit points est sélectionné dans une zone différente.

Afin de garantir une certaine équité des zones, celles-ci sont calculées avec un algorithme dichotomique. L'image est d'abord séparée en deux verticalement afin qu'autant de points se situent à gauche et à droite de cette séparation. L'opération est répétée pour chacune des zones obtenues avec une division horizontale. À la troisième itération, on vérifie bien que le nombre de zones est égal à  $2^3 = 8$ .



## 6 Rectification linéaire d'une paire d'images non calibrées

### 6.1 Enjeux de la rectification

La rectification est une étape primordiale dans la stéréoscopie par corrélation : Elle permet de passer d'une géométrie tridimensionnelle complexe du système à une géométrie épipolaire simple dans laquelle les droites épipolaires sont parallèles aux lignes des images.

### 6.2 Une rectification à partir de la géométrie épipolaire

Dans notre projet, la calibration des caméras n'étant pas effectuée, nous n'avons pas les coordonnées de celles-ci : il s'agit des paramètres intrinsèques et extrinsèques des caméras qui donnent les matrices de projection sur le plan des deux images. La rectification devient donc une étape difficile. En suivant les travaux de Frédéric Devernay [Dev97] nous utilisons une technique de rectification mettant en relation à la fois les acquis de la géométrie épipolaire ainsi que les propriétés de la SVD (Singular Value Decomposition, voir 5.1.2) de la matrice fondamentale, seule information que nous avons sur le système des deux images. [AF97], [Har97] et [BM95] pourront être consultés pour plus d'informations sur cette méthode de rectification.

#### 6.2.1 Projection des images dans le plan de rectification et épipôles à l'infini.

La technique la plus simple de rectification dans le cas d'une paire d'images non calibrées est de projeter les deux images sur un même plan dit « plan de rectification » parallèle à la droite joignant les deux centres optiques. Par cette transformation, on voit que les épipôles de chacune des images sont projetés à l'infini dans les images rectifiées. On choisit donc cette direction et les droites épipolaires deviennent donc horizontales. Une paire d'images rectifiées est illustrée en figure 9.

#### 6.2.2 La matrice fondamentale d'une paire d'images rectifiées est une matrice antisymétrique

Soit  $F$  la matrice fondamentale associée à une paire d'images. Une condition nécessaire pour que deux points  $\mathbf{m}$  et  $\mathbf{m}'$  correspondent est qu'ils satisfassent la « contrainte épipolaire » :

$$\mathbf{m}'^t F \mathbf{m} = 0$$

Lorsque la paire d'images est rectifiée, les lignes épipolaires sont parallèles et deux points qui se correspondent ont la même ordonnée donc on en déduit

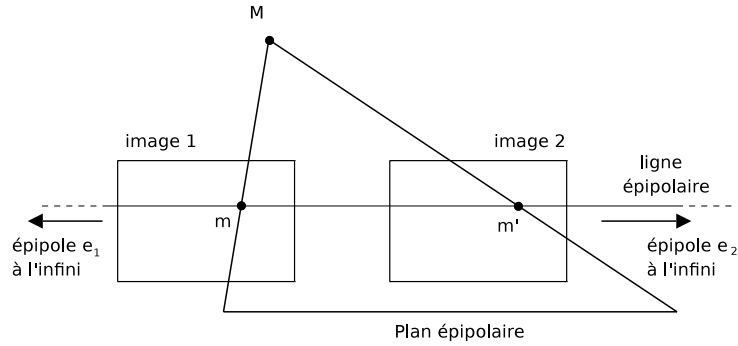


FIG. 9 – Paire d’images rectifiées.

que la matrice fondamentale d’une paire d’images rectifiées est une matrice antisymétrique de la forme :

$$F_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

### 6.2.3 Avantage de cette méthode : minimiser les distorsions

L’avantage de cette méthode dite linéaire est qu’elle permet de projeter les images dans le plan de « rectification » et non projeter une image dans le plan de l’autre image. Les distorsions sont ainsi fortement minimisées, la géométrie des objets est conservée et les deux images sont traitées symétriquement.

## 6.3 Étapes de la rectification

### 6.3.1 Calcul d’un jeu de matrices de rectification compatible avec $F$

Commençons par trouver une paire particulière de matrice de rectification ( $H_{01}, H_{02}$ ) telle que  $F = H_{01}^t F_0 H_{02}$

Soit  $F$  la matrice fondamentale associée à notre système de deux images. Cette matrice est de rang 2 (voir les techniques robustes d’estimation de la matrice fondamentale comme le “8-point algorithm normalized” en 5.1) et sa décomposition en valeur singulière s’écrit donc sous la forme suivante après une simple transformation par changement de base :

$$F = \begin{bmatrix} \mathbf{e}' & \mathbf{u}_1 & \mathbf{u}_2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_1 & 0 \\ 0 & 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{e}^t \\ \mathbf{v}_1^t \\ \mathbf{v}_2^t \end{bmatrix} = P \Delta Q^t$$

où  $\sigma_1$  et  $\sigma_2$  sont deux nombres positifs appelés valeurs singulières de  $F$ ,  $\Delta$  est une matrice diagonale et,  $P$  et  $Q$  sont deux matrices orthogonales. Notons que les vecteurs  $\mathbf{e}'$  et  $\mathbf{e}$  sont les épipôles associés à chacune des images. Ils correspondent à la valeur singulière nulle.

Nous voulons maintenant écrire cette décomposition avec  $F_0$  afin de déterminer notre paire de matrices de rectification. Posons  $\sigma_1 = 1$  et  $\sigma_2 = \sigma > 0$  car nous travaillons à un facteur d'échelle près.

On réécrit la décomposition en symétrisant l'expression ainsi :

$$F = \begin{bmatrix} \mathbf{e}' & \mathbf{u}_1 & \sqrt{\sigma}\mathbf{u}_2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}^t \\ \sqrt{\sigma}\mathbf{v}_2^t \\ -\mathbf{v}_1 \end{bmatrix}$$

Nous obtenons donc notre paire  $(H_{01}, H_{02})$  de matrices de rectification :

$$H_{02} \cong \begin{bmatrix} \mathbf{e}^t \\ \sqrt{\sigma}\mathbf{v}_2^t \\ -\mathbf{v}_1 \end{bmatrix}$$

$$H_{01} \cong \begin{bmatrix} \mathbf{e}'^t \\ \mathbf{u}_1^t \\ \sqrt{\sigma}\mathbf{u}_2^t \end{bmatrix}$$

### 6.3.2 Dédution de l'ensemble des matrices de rectification avec conservation des disparités

Devernay montre dans [Dev97] que l'ensemble des paires de matrices de rectification se déduisent de  $(H_{01}, H_{02})$  par les relations :

$$H_1 = T_1 H_{01} \text{ et } H_2 = T_2 H_{02}$$

$$\text{avec } T_1 = \begin{pmatrix} a & b & c \\ 0 & e & f \\ 0 & h & 1 \end{pmatrix} \text{ et } T_2 = \begin{pmatrix} a' & b' & c' \\ 0 & e & f \\ 0 & h & 1 \end{pmatrix}$$

Notre objectif est donc de déterminer les inconnus  $a, b, c, a', b', c', e, f$  et  $h$ .

Nous allons les déterminer tout en gardant en tête que nous devons minimiser les distorsions lors de la projection dans le plan de rectification mais aussi conserver les disparités entre les deux images. Expliquons tout d'abord comment on définit cette disparité :

- Dans le cas d'images rectifiées, c'est la différence  $d'$  de position selon les colonnes de deux points correspondants.
- Dans le cas d'images non rectifiées, c'est la valeur  $d = d_E(\mathbf{m}_1, \mathbf{m}_2) = \sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2}$  en considérant  $\mathbf{m}_1(u_1, v_1)$  et  $\mathbf{m}_2(u_2, v_2)$  deux points respectivement de l'image 1 et de l'image 2, correspondants.

La contrainte de conservation des disparités des images originales s'écrit donc :  $d = d' = u'_2 - u'_1$  avec  $u'_2$  et  $u'_1$  les abscisses respectives des points  $m_2$  et  $m_1$  dans les images rectifiées.

On exprime alors :

$$u'_1 = \frac{a_1 K_1 + b_1 L_1 + c_1 N_1}{h L_1 + N_1} \text{ et } u'_2 = \frac{a_2 K_2 + b_2 L_2 + c_2 N_2}{h L_2 + N_2}$$

avec  $K_i(u_i, v_i) = H_{0i}^1 P_i^t$ ,  $L(u_i, v_i) = H_{0i}^2 p_i^t$  et  $N_i = (u_i, v_i) = H_{0i}^3 p_i^t$

où  $H_{0i}^j$  est la  $j^{me}$  ligne de la matrice  $H_{0i}$  et  $p_i = (u_i, v_i, 1)$  un pixel de l'image  $i$ .

En utilisant les expressions précédentes et le fait que :

$$\frac{H_{01}^2 p_1^t}{H_{01}^3 p_1^t} = \frac{L_1}{N_1} = \frac{H_{02}^2 p_2^t}{H_{02}^3 p_2^t} = \frac{L_2}{N_2} = \mu$$

La contrainte de conservation de la disparité entre les deux images devient :

$$d = d' = u'_2 - u'_1 = -a_1 \frac{K_1}{N_1} - b_1 \mu - c_1 + a_2 \frac{K_2}{N_2} + b_2 \mu + c_2 - h d \mu$$

La résolution de ce système linéaire de huit équations (nous travaillons essentiellement avec huit correspondances) n'est cependant pas suffisant pour déterminer nos inconnus. La contrainte  $d' = d$  ne suffit pas car les abscisses respectives des points dans les images rectifiées restent indéterminées. On peut en effet aboutir à des images rectifiées trop grandes ou trop déformées.

On calcul donc tout d'abord la taille de l'image rectifiée ( $Lig', Col'$ ) :

$Col' = Col * \cos(\alpha_c) + Lig * \sin(\alpha_c)$  et  $Lig' = Lig * \cos(\alpha_c) + Col * \sin(\alpha_c)$  avec  $\alpha_c$  l'angle que fait la droite épipolaire reliant  $e'$  au coin de l'image le plus éloigné, avec l'horizontale.

On en déduit la position des coins de l'image d'origine dans l'image rectifiée :

$$\begin{aligned} u'_c &= 0, v'_c = Lig * \cos(\alpha_c) \\ u'_a &= Col * \cos(\alpha_a), v'_a = 0 \\ u'_d &= Col * \cos(\alpha_d), v'_d = Lig' \\ u'_b &= Col, v'_b = Col * \sin(\alpha_b) \end{aligned}$$

Ces quatre équations permettent donc de déterminer les paramètres  $a_i, b_i, c_i$ , et  $h$ , tout en minimisant la distorsion.

Il reste alors à déterminer les derniers coefficients manquant des matrices  $T_1$  et  $T_2$ .

Pour cela, il suffit de résoudre le système suivant, exprimant les ordonnées rectifiées des coins de l'image 1 en fonction de  $e$  et  $f$ .

$$\begin{cases} eL_1(0, 0) + fN_1(0, 0) = v_A \\ eL_1(Col, 0) + fN_1(Col, 0) = v_b(hL_1(Col, 0) + N_1(Col, 0)) \\ eL_1(0, Lig) + fN_1(0, Lig) = v_c(hL_1(0, Lig) + N_1(0, Lig)) \\ eL_1(Col, Lig) + fN_1(Col, Lig) = v_d(hL_1(Col, Lig) + N_1(Col, Lig)) \end{cases}$$

Une fois ces coefficients calculés, on déduit les matrices  $H_1$  et  $H_2$  par  $H_1 = T_1 H_{01}$  et  $H_2 = T_2 H_{02}$  et on peut appliquer la rectification aux deux images.

## 7 Rectification pour une géométrie épipolaire quelconque.

### 7.1 Principe de fonctionnement

Cette méthode de rectification basée sur le travail de Daniel Oram (i.e. [Ora01]) permet de rectifier une paire d'images de géométrie épipolaire quelconque en limitant la déformation des images.

La plupart des méthodes de rectification actuelles opèrent en utilisant une simple transformation linéaire qui aboutit souvent à des images rectifiées très étirées. En contrepartie, le risque d'une transformation non linéaire est de déformer l'image localement et de rendre difficile l'appariement des points caractéristiques.

On utilise ici successivement une transformation linéaire dont l'objectif est d'obtenir un épipôle unique pour les deux images et des lignes épipolaires correspondantes puis une transformation non linéaire qui rectifie l'image en utilisant des coordonnées polaires centrées sur l'épipôle.

### 7.2 Transformation linéaire

On cherche ici à déterminer une homographie  $H$  (une application projective bijective) qui permette de déplacer le premier épipôle à l'emplacement du second et de faire correspondre les lignes épipolaires.

#### 7.2.1 Homographies compatibles

On cherche  $H$  qui transforme  $\mathbf{e}$  en  $\mathbf{e}'$ , ce qui correspond à la relation

$$H\mathbf{e} = \mathbf{e}'$$

De plus, les lignes épipolaires doivent correspondre : on veut que pour un couple de points correspondants  $(\mathbf{m}, \mathbf{m}')$ , sachant par définition que  $H\mathbf{m}'$  se situe sur la ligne épipolaire  $F\mathbf{m}$ , avoir  $H\mathbf{m}$  sur cette même ligne épipolaire  $F\mathbf{m}$ .

Il suffirait d'imposer la contrainte supplémentaire qui est de rendre les lignes épipolaires parallèles pour aboutir à un cas de rectification totalement linéaire, ceci avec l'inconvénient d'obtenir des images très étirées ou pire, sans limites.

Nous cherchons donc une homographie qui permette de déplacer l'épipôle  $\mathbf{e}$  à l'emplacement de l'épipôle  $\mathbf{e}'$ , en faisant coïncider les lignes épipolaires mais sans les rendre nécessairement parallèles. Une telle homographie est appelée une *homographie compatible*.

L'ensemble des homographies compatibles est donné par la relation suivante (voir [Har97, HG93]) :

$$H = [\mathbf{e}']_{\times} F - \mathbf{e}' \mathbf{a}^t$$

avec  $[\mathbf{e}']_{\times}$  la matrice antisymétrique définie par :

$$[\mathbf{e}']_{\times} = \begin{bmatrix} 0 & -\mathbf{e}'_3 & \mathbf{e}'_2 \\ \mathbf{e}'_3 & 0 & -\mathbf{e}'_1 \\ -\mathbf{e}'_2 & \mathbf{e}'_1 & 0 \end{bmatrix}$$

et  $\mathbf{a}$  un vecteur de taille 3 quelconque tel que  $\det(H) \neq 0$ .

### 7.2.2 Moindres carrés

La méthode des moindres carrés (i.e. Least Squares) est couramment utilisée en régression linéaire. Contrairement aux méthodes statistiques comme Ransac ou LMedS, la méthode des moindres carrés ne tient pas compte d'inliers ou d'outliers mais sera utilisée dans le cadre de cette rectification pour estimer l'homographie  $H$ .

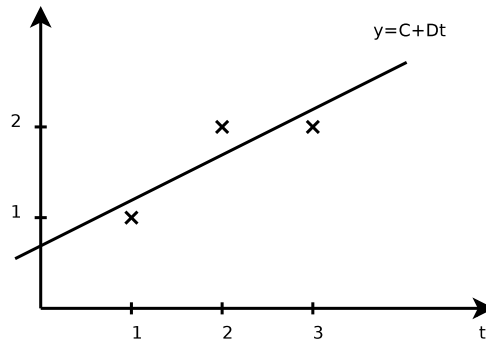


FIG. 10 – Régression linéaire avec la méthode des moindres carrés

Comme pour les algorithmes d'estimation précédents, nous illustrons cette méthode avec le cas de la régression linéaire. Nous disposons de trois points par lesquels doit passer une droite d'équation  $y = C + Dt$ , mais aucune droite ne passe par tous ces points. On cherche donc à trouver la meilleure droite qui minimise le vecteur d'erreurs.

Dans l'exemple donné en figure 10, on voudrait des points qui vérifient le système d'équations suivant :

$$\begin{cases} C + D = 0 \\ C + 2D = 2 \\ C + 3D = 2 \end{cases}$$

ou autrement posé :

$$\mathbf{Ax} = \mathbf{b} \text{ avec } A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} C \\ D \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}$$

Bien entendu, il n'existe aucune solution  $\mathbf{x}$  à ce système, mais il existe une meilleure solution  $\hat{\mathbf{x}}$ . Celle-ci pourrait être trouvée par le calcul en cherchant le minimum de la fonction  $f(C, D) = (C + D - 1)^2 + (C + 2D - 2)^2 + (C + 3D - 2)^2$ , en calculant ses dérivées partielles et en trouvant les valeurs pour lesquelles elles s'annulent.

La solution est trouvée en résolvant l'équation linéaire  $A^t A \hat{\mathbf{x}} = A^t \mathbf{b}$  soit

$$\begin{aligned} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} &= \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 3 & 6 \\ 6 & -14 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \\ &\Leftrightarrow \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \end{aligned}$$

La droite trouvée correspond bien à ce que l'on cherchait :  $y = \frac{2}{3} + \frac{1}{2}t$ .

### 7.2.3 Minimisation de la distorsion géométrique

Le choix du vecteur  $\mathbf{a}$  va maintenant être fait de manière à réduire les effets de distorsion dus à la projection  $H$ . Cette minimisation de la distorsion est basée sur un ensemble de points appariés  $(\mathbf{m}_i, \mathbf{m}'_i)$  donnés en entrée et consiste à minimiser la distance euclidienne entre un point  $H\mathbf{m}_i$  et son correspondant dans la deuxième image  $\mathbf{m}'_i$ . Le critère à minimiser est le suivant :

$$\min \sum_{i=1}^n d_E^2(\mathbf{m}_i, ([\mathbf{e}']_X F - \mathbf{e}' \mathbf{a}^t) \mathbf{m}'_i)$$

En posant  $h = [\mathbf{e}']_X F$  on obtient deux équations par couple de points appariés

$$k \in \{1, 2\}, (\mathbf{m}'_k \mathbf{e}'_3 - \mathbf{e}'_k) \mathbf{m}^t \mathbf{a} = (\mathbf{m}'_k h_3^t - h_k) \mathbf{m}$$

Le critère précédent est d'abord minimisé en utilisant la méthode des moindres carrés (voir [Str]) pour une explication détaillée).

Soit  $A$  une matrice  $m * n$  et  $\mathbf{b}$  un vecteur de taille  $m$ , la méthode des moindres carrés permet de trouver  $\hat{\mathbf{x}}$  tel que  $\|A\hat{\mathbf{x}} - \mathbf{b}\|^2 = \|\mathbf{e}\|^2$  soit minimal. La solution est donnée par l'une des plus importantes équations de statistiques et d'estimation :

$$A^t A \hat{\mathbf{x}} = A^t \mathbf{b}$$

Appliquée à notre critère, cette équation nous permet de trouver une homographie minimisant la distorsion géométrique. Il nous suffit de construire une matrice de contrainte  $A$  et un vecteur  $\mathbf{b}$  pour minimiser ce critère sur les points donnés.

$$\underbrace{(\mathbf{m}'_k \mathbf{e}'_3 - \mathbf{e}'_k) \mathbf{m}^t}_{A_i} \mathbf{a} = \underbrace{(\mathbf{m}'_k h_3^t - h_k) \mathbf{m}}_{\mathbf{b}_i}$$

Malheureusement, cette approche a l'inconvénient de prendre en compte tous les points dans le calcul de notre homographie et il est donc nécessaire de doubler cette minimisation d'un procédé d'estimation robuste qui nous permette de ne pas tenir compte des outliers.

Un algorithme tel que LMEDS ou RANSAC déjà décrit précédemment permet de rendre la minimisation plus robuste.

### 7.3 Transformation non linéaire

On dispose maintenant de deux images pré-rectifiées pour lesquelles les deux épipôles  $\mathbf{e}$  et  $\mathbf{e}'$  sont confondus et dont les lignes épipolaires correspondent. Il suffit maintenant de se placer en coordonnées polaires centrées sur l'épipôle  $\mathbf{e}$  et de parcourir l'image avec de petits angles pour reconstruire une image rectifiée. Une ligne épipolaire  $l_i$  correspondra à la  $i^{\text{me}}$  ligne de l'image rectifiée comme montré en figure 11. Ce procédé rend les lignes épipolaires parallèles et place l'épipôle à l'infini.

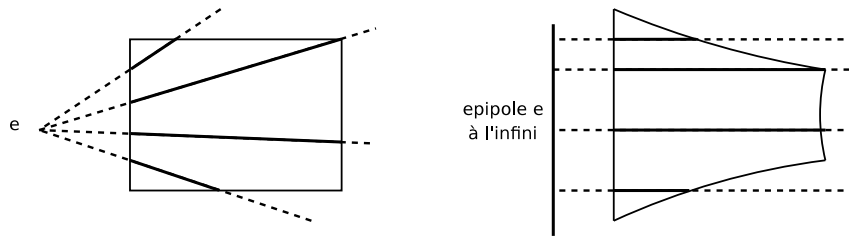


FIG. 11 – correspondance des lignes épipolaires et lignes rectifiées

#### 7.3.1 Intervalle angulaire commun

Grâce à notre homographie compatible  $H$ , nous disposons de lignes épipolaires correspondantes. Cependant, les lignes épipolaires d'une image n'ont pas nécessairement de correspondante dans l'autre image. On se limite maintenant à un intervalle d'angles par rapport à l'épipôle  $I = [\alpha_{min}, \alpha_{max}]$  commun aux deux images.

Le principe de l'algorithme utilisé est de calculer l'intervalle  $[\alpha_{mink}, \alpha_{maxk}]$  pour chaque image puis de calculer l'intersection de ces deux intervalles  $I = [\alpha_{min1}, \alpha_{max1}] \cap [\alpha_{min2}, \alpha_{max2}]$ .

Si l'épipôle se situe à l'intérieur d'une image, on a immédiatement un intervalle d'angle  $[\alpha_{mink}, \alpha_{maxk}] = [0, 2\pi]$ . Dans le cas où l'épipôle est à l'extérieur de l'image, ces angles sont nécessairement des angles de coins de l'image par rapport à l'épipôle. Soit  $\theta = \langle \mathbf{i}, \mathbf{ce} \rangle$  l'angle entre le vecteur unitaire  $\mathbf{i}$  et le vecteur  $\mathbf{ce}$  du centre de l'image à l'épipôle. On effectue une rotation d'angle  $-\theta$  centrée sur l'épipôle qui nous permet d'avoir l'épipôle à droite de l'image comme décrit en figure 12.

Une fois cette rotation effectuée, les angles  $\langle \mathbf{i}, \mathbf{k} \rangle$  pour  $\mathbf{k}$  un coin de l'image sont ordonnés de telle sorte que le plus petit  $\alpha_i$  corresponde à l'angle minimum



et le plus grand  $\alpha_j$  à l'angle maximum.

On obtient finalement  $[\alpha_{min}, \alpha_{max}] = [\alpha_i + \theta, \alpha_j + \theta]$ .

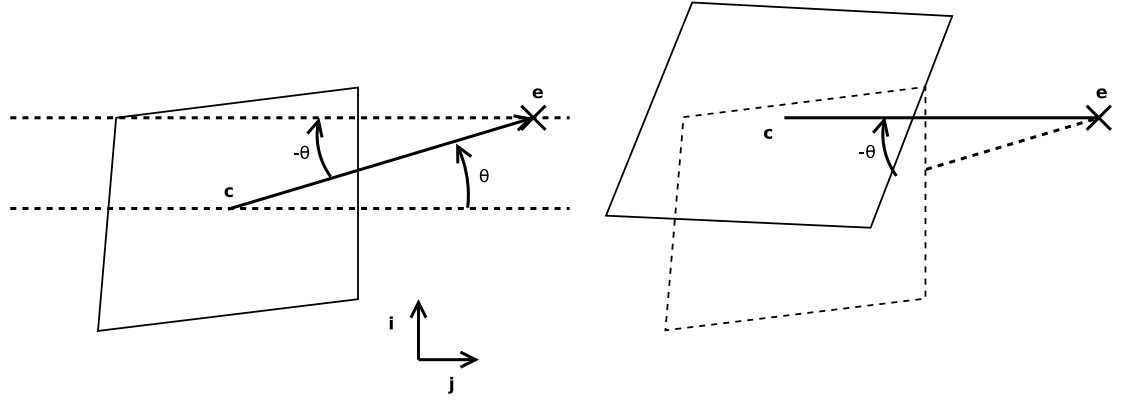


FIG. 12 – Rotation d'angle  $-\theta$

### 7.3.2 Choix des angles

Nous savons désormais que les lignes épipolaires de l'image seront parcourues entre les angles  $\alpha_{min}$  et  $\alpha_{max}$ . Nous allons maintenant établir une correspondance entre un angle  $\alpha_i$  tel que  $\alpha_{min} \leq \alpha_i \leq \alpha_{max}$  et la ligne  $i$  dans l'image rectifiée.

Soit  $d$  la distance maximale entre l'épipôle et un pixel sur la  $i^{me}$  ligne épipolaire. La relation inductive suivante permet de définir tous les angles  $\alpha_i$  avec la garantie que l'écart maximum entre un pixel de la ligne  $i$  et un pixel de la ligne  $j$  soit au pire égal à 1. Cette relation est décrite par la figure 13.

$$\begin{cases} \alpha_0 = \alpha_{min} \\ \alpha_{i+1} = \alpha_i + \arctan(\frac{1}{d}) \end{cases}$$

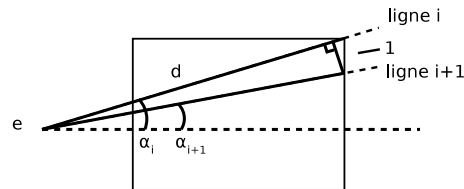


FIG. 13 – incrément d'angle minimum pour éviter la perte d'information

### 7.3.3 passage en coordonnées polaires

Une fois connus les angles  $\alpha_i$ , le passage en coordonnées polaires se fait tout naturellement. l'image rectifiée est échantillonnée selon l'algorithme 4.

---

**Algorithm 4** échantillonnage de l'image rectifiée

---

**Entrées**

- $img_{orig}$  Une image à rectifier.  
 $H$  L'homographie compatible sélectionnée.  
 $T$  Une table mettant en correspondance un index de ligne épipolaire rectifiée avec l'angle correspondant.  
 $d_{min}$  La distance minimum de l'épipôle à un point de l'image transformée par  $H$ .  
 $d_{max}$  La distance maximum de l'épipôle à un point de l'image transformée par  $H$ .  
 $\mathbf{e}$  L'épipôle commun aux deux images transformées par  $H$ .

**Sorties**

- $img_{dest}$  La nouvelle image rectifiée de taille  $(d_{min} - d_{max}, size(T))$ .

**Début**

Pour  $y$  un index de ligne épipolaire rectifiée :

1. Soit  $l$  la demi-droite épipolaire d'angle  $\alpha_i = T[y]$ . Déterminer les intersections de  $l$  avec le contour de l'image transformée par  $H$ .
2. Trouver  $\mathbf{a}$  l'intersection la plus proche et  $\mathbf{b}$  l'intersection la plus éloignée.
3. Poser  $\mathbf{u} = H^{-1}(\mathbf{a})$  et  $\mathbf{v} = H^{-1}(\mathbf{b})$  les points correspondants à  $\mathbf{a}$  et  $\mathbf{b}$  dans l'image d'origine.
4. Construire un vecteur  $\mathbf{w}$  tel que  $\mathbf{u} + d_E(\mathbf{a}, \mathbf{b})\mathbf{w} = \mathbf{v}$ .
5. Pour  $j$  entier allant de 0 à  $d_E(\mathbf{a}, \mathbf{b})$  :
  - (a) poser  $\mathbf{p}_{dest}$  le point de coordonnées  $(d_E(\mathbf{e}, \mathbf{a}) - d_{min} + j, y)$ .
  - (b) poser  $\mathbf{p}_{orig} = \mathbf{u} + j\mathbf{w}$ .
  - (c)  $img_{dest}(\mathbf{p}_{dest}) := B(img_{orig}, \mathbf{p}_{orig})$ , avec  $B(img, \mathbf{p})$  l'interpolation bilinéaire de  $\mathbf{p}$  dans l'image  $img$ .

**Fin**

---

## 8 Conclusion générale

Ce projet est pour nous une réussite. Effectivement, nous étions relativement incertains quant au fait d’obtenir des résultats satisfaisants, bien que divers chercheurs aient réussi à réaliser une reconstruction 3D, la plupart de nos sources sont des thèses portant sur des images calibrées. Nous sommes donc fiers de pouvoir présenter des images non calibrées rectifiées, et il est à noter que nos travaux seront continués par la suite, afin d’aboutir à la modélisation 3D complète. Ce projet fut commencé en stage technique par Gilles Lepenec avec pour but d’être porté en projet temps réel pour être embarqué sur un drone. Pour le moment, il nous semble tout à fait improbable de pouvoir réaliser une telle prouesse, même en optimisant de manière drastique le code, et en implémentant de nouvelles heuristiques. Un point intéressant serait de tenter de paralléliser certaines parties du code, pour l’exécuter sur des dsp dans le cas d’une application temps réel ou embarqué, ou pour l’exécuter sur les puces des cartes graphiques, particulièrement efficaces dans le calcul parallèle, ce qui n’existe pas à l’heure où nous rédigeons ces lignes.

Ce projet nous a permis de nous rapprocher du monde réel de l’ingénieur, et ce à la fois humainement et scientifiquement : établir un cahier des charges, faire des recherches et des tests, gérer un projet. Nous avons eu là un aperçu fort réaliste de ce qui nous attend dans moins d’un an. la modélisation 3D automatisé est un enjeu important dans des domaines divers, que ce soit scientifique, militaire ou grand public. Cela nous fait croire qu’il y a sûrement de nombreux débouchés, et il ne serait pas impossible que notre stage de fin d’étude ait un rapport avec ce sujet. Nous tenons bien entendu à remercier M. Benoît Petitpas, notre suiveur, qui nous a accompagnés et encouragés tout au long de ce défi ambitieux.

## 9 Conclusions personnelles

### 9.1 Arnaud Deschavanne

Travailler dans cette équipe a été pour moi une expérience fort enrichissante à plusieurs points de vue, d’une part scientifique, puisque j’ai pu découvrir un peu plus concrètement ce qui touchait au monde de l’imagerie et de la vision par ordinateur, au travers des nombreux documents que nous avons pu lire sur le sujet, également sur les nombreux aspects mathématiques qui sont abordés dans ce projet. Que ce soit des probabilités, de l’estimation de donnée des notions relative au matrices, ou encore du traitement du signal, nous avons su trouver un sujet pluridisciplinaire, qui nous a ouvert à la réalité du métier d’ingénieur. Mais c’est aussi sur le côté humain que j’ai énormément appris .

Concernant l’équipe, je pense qu’un leader s’est naturellement dégagé du groupe, Ludovic Arnold, et je tenais à souligner la façon dont il a su gérer et faire rebondir l’équipe tout au long du projet lorsque nous étions dans une impasse, et pour cela je voulais le remercier. Le partage des tâches initialement prévu,

n'a pas pu être respecté, car il s'agissait avant tout d'un projet de recherche, où nous avons principalement mis en place divers algorithmes issus de nombreuses thèses, parfois relativement complexes et difficiles à interpréter. nous avons donc travaillé ensemble afin de mieux comprendre et assimiler les différentes notions du projet. Je suis très heureux d'avoir pu participer à ce projet avec ces personnes que j'ai pu apprendre à connaître au fil de ce semestre, et je pense qu'il serait important de continuer ce projet qui pourrait être une base d'un projet plus conséquent, avec pourquoi pas des enjeux commerciaux à la clé. Nous ne nous sommes pas suffisamment intéressés à ce qui se faisait actuellement sur le marché pour pouvoir répondre à cette question. je tenais également à remercier Gilles Lepennec pour nous avoir proposé de reprendre ce projet qui est la suite de son stage technique, il a donc pu nous apporter de nombreux éléments de réponses à nos diverses questions en début de parcours, et Eric Benhaim pour ses explications mathématiques toujours claires, synthétiques et concises.

## 9.2 Ludovic Arnold

Ce projet aura été pour moi un grand moment de ma scolarité à l'ESIEA. C'est avant tout une formidable expérience humaine et scientifique qui m'a permis d'apprendre et de progresser. Je tiens bien sûr à remercier Gilles Lepennec initiateur de ce projet sans quoi rien n'aurait été possible, et tous les membres de l'équipe, qui ont su montrer leur motivation pour réussir. Il s'agissait d'un sujet difficile mais c'est le prix à payer pour faire un travail intéressant et moderne.

Nous avons pu découvrir des applications concrètes de ce que nous avons appris en cours cette année et acquérir de nouvelles connaissances dans des domaines variés. Je ne citerai que la minimisation et la régression pour lesquelles nous avons exploré de nombreuses alternatives. Ce projet a été pour nous l'occasion d'améliorer nos compétences mathématiques, informatiques et rédactionnelles et j'espère que ce rapport reflète correctement la passion qui nous a guidé pendant ces derniers mois. Bien sûr ce projet ne s'arrête pas là ! Le programme développé doit être distribué sous licence CECiLL d'ici peu et sera de ce fait en perpétuelle évolution.

## 9.3 Eric Benhaim

Le projet de travailler dans le traitement d'image est un projet qui me tient à coeur depuis longtemps et un projet aussi ambitieux que la reconstruction 3D a tout de suite attiré ma curiosité et mon envie d'atteindre le but final.

J'aime relier les domaines de connaissance entre eux et ce projet a mis à rude épreuve :

- mon esprit analytique pour aborder un problème inconnu jusqu'à lors, envisager les différentes possibilités et choisir la meilleure,
- mon esprit mathématique pour la résolution de nombreux problèmes liés à l'algèbre et l'interprétation d'une svd, à la géométrie tridimensionnelle et épipolaire, aux calculs différentiels et aux probabilités,

- mon esprit logique pour la modélisation d’algorithmes et l’interprétation de nos résultats.
- mes savoirs en programmation c++

La stéréoscopie étant un domaine en plein essor où de nombreux chercheurs apportent chaque année leur contribution. Il était très intéressant de découvrir l’évolution des techniques de rectification par exemple, ou d’estimation de la matrice fondamentale avec la normalisation du “8-point algorithm”. Le choix d’utiliser une succession d’estimateurs dans un certain ordre ou de préférer un critère de corrélation (ZNSSD ou ZNCC) pour le matching de point est l’exemple même de l’étendue des possibilités et des connaissances dans lesquelles nous nous sommes plongés et, ce fut très instructif et passionnant.

L’équipe était active et motivée, ce qui a rendu le travail plus attrayant car les résultats ne sont pas arrivés tout de suite : Chaque passage à une étape supérieure dans ce projet difficile fut très motivant, et excitant. Je pense particulièrement aux nombreux mauvais résultats des premières rectifications épipolaires.

Ce projet m’a donc permis de rentrer pleinement dans l’univers de la stéréoscopie et d’entrevoir toutes les possibilités que l’on peut en faire. C’est tout à fait un domaine d’avenir et fait clairement partie de mes ambitions futures car ses domaines d’applications comme l’imagerie médicale m’attirent beaucoup.

## 9.4 Gilles Lepennec

J’ai commencé à travailler sur la reconstruction 3D en stage au laboratoire ATIS durant l’été 2007. Ce stage m’avait passionné et je désirais continuer ce travail dans le cadre d’un projet en équipe au sein de l’ESIEA si possible.

C’est pourquoi le projet PAIR était l’occasion de prolonger ce projet.

Présenter ce travail aux étudiants et les attirer sur ce projet ne fut pas chose facile, tant le champs de connaissance est large.

Cependant la perspective d’obtenir une reconstruction 3D d’une paire d’image était assez motivante.

Au début de ce projet nous avons du faire de la recherche documentaire, comprendre les algorithmes. Essentiellement nous devons avoir une vue d’ensemble de tout le processus soit du filtre de Harris à la rectification.

Notre objectif était d’aboutir à la rectification épipolaire et si possible obtenir un nuage de points dans l’espace. C’est pourquoi nous avons travaillé sur le filtre de Harris jusqu’au ZNSSD ensemble afin de concentrer nos efforts. En effet la divisions des tâches fut difficile au début car il n’est pas possible de travailler sur des algorithmes tels que le RANSAC sans comprendre le ZNSSD ou encore le filtre de Harris. Toutes les parties de ce projet sont liées entre elles.

Nous avons donc concrétisé notre apprentissage théorique en apportant chacun un savoir acquis sur le même algorithme et ce jusqu’au ZNSSD.

Ensuite les méthodes de trie de points type RANSAC, LMEDS ont été abordés.

En raison de la complexité et de la multitude de papiers existant à ce sujet chacun a pu apporter des éléments nouveaux aux bases posées par le stage.

Une fois que nous maîtrisons le calcul de la matrice fondamentale grâce à un tri approprié des matches, nous sommes passés à l'étude de la rectification.

Deux approches ont été trouvées et étudiées parallèlement.

Ce projet fut passionnant pour moi et l'ensemble du groupe car il a mis nos compétences mathématiques, informatique et algorithmique à rude épreuve du début à la fin, et chaque résultat ou progression a été ressenti comme une victoire au sein du groupe.

Ce projet nous a également permis de lire énormément de documentation scientifique anglophone ou non, ce qui fut enrichissant et nécessaire.

Ainsi nous avons acquis des méthodes d'analyse, d'estimation assez utilisés dans le traitement de l'image comme le Ransac, Lmeds et d'autres méthodes de corrélation.

De plus travailler en groupe sur un même article scientifique est assez intéressant et est une activité que nous aurons probablement à réaliser en tant qu'ingénieur.

Notre projet a donc progressé et aboutit à une chaîne de traitement, chacun des membres s'est impliqué et approprié le projet ce qui était mon objectif.

En conclusion ce travail exigeant de compréhension et de réalisation nous apporté à tous un sérieux éclairage sur le traitement d'image et l'analyse de documentation scientifique, et avoir atteint une rectification fonctionnelle est pour moi une grande satisfaction.

Je remercie donc chacun des membres de ce projet et monsieur Beaudoin, Gademer et Petitpas pour leur soutien.

## Références

- [AF97] et A. Verri A. Fusiello, E. Trucco. Rectification with unconstrained stereo geometry. 1997.
- [BM95] Boubakeur-Seddik Boufama and Roger Mohr. Epipole and fundamental matrix estimation using virtual parallax. In *ICCV*, pages 1030–1036, 1995.
- [Dev97] F. Devernay. *Vision stéréoscopique et propriétés différentielles des surfaces*. PhD thesis, Ecole Polytechnique, l’Institut National de Recherche en Informatique et en Automatique, Chesnay Cedex, France, 1997.
- [Far05] Dirk Sven Farin. *Automatic video segmentation employing object camera modeling*. PhD thesis, Technische Universiteit Eindhoven, 2005.
- [FP07] David A. Forsyth and Jean Ponce. *Computer Vision : A Modern Approach*. Prentice Hall, August 2007.
- [Har97] R. Hartley. Theory and practice of projective rectification. 1997.
- [HG93] R. Hartley and R. Gupta. Computing matched-epipolar projections. 1993.
- [HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521540518, second edition, 2004.
- [Ora01] Daniel Oram. Advanced rectification for any epipolar geometry. 2001.
- [RL87] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.
- [Str] G. Strang. Massachusetts institute of technology, 18.06 linear algebra, fall 2002, video lecture 16 : Projection matrices and least squares. <http://dspace.mit.edu/handle/1721.1/35861>.
- [Str03] G. Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press, 2003.
- [Wik] Wikipedia, the free encyclopedia - median absolute deviation. [http://en.wikipedia.org/wiki/Median\\_absolute\\_deviation](http://en.wikipedia.org/wiki/Median_absolute_deviation).
- [Zha] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty : A review. Technical Report 2927.