



**HAL**  
open science

## Expériences d'appariement d'images avec SIFT

Arnaud Le Bris

► **To cite this version:**

Arnaud Le Bris. Expériences d'appariement d'images avec SIFT. [Rapport Technique] IGN - Laboratoire MATIS. 2010, pp.163. hal-02370256

**HAL Id: hal-02370256**

**<https://hal.science/hal-02370256>**

Submitted on 15 May 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# *Expériences d'appariement d'images avec SIFT*

Arnaud LE BRIS

IGN - Laboratoire MATIS

— 2007-2010 —



# Expériences d'appariement d'images avec SIFT

2007-2010 — 163 pages

## Avant-propos :

Ce document rassemble des extraits de plusieurs rapports rédigés à l'issue de différentes études menées ces dernières années autour de SIFT. Ainsi, les travaux sur les données aériennes datent majoritairement de 2007-2008, tandis que ceux concernant les données terrestres sont plutôt de 2009-2010.

## Historique des modifications :

- **fin 2007** : Première version du document, incluant présentation de SIFT, de certaines implémentations et premiers tests de chantiers aériens
- **début 2008** : Ajout de la partie sur la constitution de tableaux d'assemblage (Ile-au-Cochons).
- **2008 - 2009** : Ajout de tests sur différents chantiers aériens au fur et à mesure qu'ils sont effectués. Ajout de la partie sur le recalage de données aériennes.
- **2009** : Ajout de la partie sur les données simulées : intégration d'un document déjà rédigé par ailleurs
- **fin 2009** : Ajout de la partie sur les données terrestres : intégration d'un rapport rédigé pour le projet Terra Numerica
- **2010** : Modification de la partie consacrée au recalage d'images terrestres diachroniques ainsi que de la partie consacrée aux outils.
- Corrections diverses.

## Rédigé par :

Arnaud LE BRIS ( IGN - Laboratoire MATIS ) — [arnaud.le-bris@ign.fr](mailto:arnaud.le-bris@ign.fr)





## Expériences d'appariement d'images avec SIFT

Arnaud LE BRIS

arnaud.le-bris@ign.fr

IGN - Laboratoire MATIS

2007-2010— 163 pages

**Résumé :** *SIFT* (*Scale Invariant Feature Transform*) est à la fois une méthode d'extraction de points d'intérêt et d'appariement de ces points entre deux images. Le comportement de l'image au voisinage de chaque point détecté est décrit au moyen d'un descripteur qui lui est associé, la mise en correspondance des points homologues s'effectuant par la suite dans l'espace de ces descripteurs.

SIFT est un algorithme robuste, en particulier aux variations d'échelle, aux rotations (2D), au bruit ainsi qu'à des transformations affines "légères". Il est en revanche assez sensible au phénomène de diachronisme autrement dit au fait que les images n'aient pas été acquises à la même période et présentent un aspect différent.

SIFT a donc fait l'objet de tests sur différents types d'images (aérien, satellite, terrestre) dans différents contextes (images de même résolution ou non, diachronisme) et pour différentes applications (extraction de points de liaison pour l'aérotriangulation, constitution automatique de tableaux d'assemblage ou recalage automatique d'images).

**Mots-clés :** *SIFT* (*Scale Invariant Feature Transform*) - Points de liaison - Points homologues - Mise en correspondance - Aérotriangulation - Tableau d'assemblage - Recalage



## Image matching experiments with SIFT

Arnaud LE BRIS

arnaud.le-bris@ign.fr

IGN - Laboratoire MATIS

2007-2010— 163 pages

**Abstract:** *SIFT* (*Scale Invariant Feature Transform*) is both an interest point extraction algorithm and a point matching method. A descriptor is associated to each extracted interest point : it describes the behaviour of the image in the neighbourhood of the point. Homologous points are then matched in the space of these descriptors.

SIFT algorithm is robust to scale variations, 2D rotations, noise and “light” affine transforms. On the other hand, it is sensitive to diachronism (that is to say to the fact that images have not been captured at the same time and appears too different from each other).

SIFT has been tested on different kinds of images (aerial, satellite, terrestrial) in different contexts (images at the same resolution or not, diachronism) and for different applications (extract tie points for bundle adjustment, build automatically mosaic/indexes of aerial photography or register automatically images).

**Key-words:** *SIFT* (Scale Invariant Feature Transform) - Tie points - Homologous points - Matching - Bundle adjustment - Aerotriangulation - Mosaic/index of aerial images - Registration

## Table des matières

<b>I</b>	<b>SIFT : algorithme et implémentations</b>	<b>7</b>
<b>1</b>	<b>Présentation de SIFT (Scale Invariant Feature Transform)</b>	<b>8</b>
1.1	Présentation générale . . . . .	8
1.2	Extraction de points d'intérêt et calcul des descripteurs . . . . .	9
1.3	Appariement des points d'intérêt . . . . .	15
<b>2</b>	<b>Utilisations possibles</b>	<b>17</b>
<b>3</b>	<b>Implémentations</b>	<b>18</b>
3.1	Détection des points et calcul des descripteurs . . . . .	18
3.1.1	Implémentations CPU disponibles et testées . . . . .	18
3.1.2	Implémentations GPU . . . . .	18
3.2	Mise en correspondance . . . . .	18
<b>4</b>	<b>Outils développés</b>	<b>19</b>
4.1	Keypoint : gestion des points et de leurs descripteurs . . . . .	19
4.1.1	Classe <i>Keypoint</i> . . . . .	19
4.1.2	Namespace <i>OutilsIOKeypoint</i> . . . . .	19
4.1.3	Namespace <i>OutilsKeypoint</i> . . . . .	19
4.2	Format binaire de stockage des points SIFT . . . . .	19
4.3	Extraction des points de Lowe : <i>Siftpp_tgi</i> . . . . .	20
4.3.1	Dépendances . . . . .	20
4.3.2	Modifications apportées à <i>sift++</i> . . . . .	20
4.3.3	<i>SiftDriver</i> . . . . .	20
4.3.4	Paramétrage . . . . .	20
4.3.5	Exécutable . . . . .	21
4.4	Appariement . . . . .	22
4.4.1	Dépendances . . . . .	22
4.4.2	Classe <i>AppariementDescripteurs</i> . . . . .	22
4.4.3	Exécutable . . . . .	23
<b>5</b>	<b>Filtrer les fautes</b>	<b>24</b>
5.1	Méthodes "générales" . . . . .	24
5.1.1	Filtre "signe" . . . . .	24
5.1.2	Détection des erreurs par appariement croisé, filtre aller-retour . . . . .	24
5.2	Méthodes contraintes par le type de scène imagée . . . . .	25
5.2.1	Détection des erreurs par considération du voisinage . . . . .	25
5.2.2	Filtre "transformation paramétrique entre images" . . . . .	25
5.2.3	Filtre similitude par accumulation (plan) . . . . .	26
5.3	Autres méthodes . . . . .	26
5.3.1	Filtrage par reprojection . . . . .	26
5.3.2	Filtrages "photogrammétriques" . . . . .	27
5.3.3	Filtrage par "chaînage" . . . . .	27
<b>II</b>	<b>Tests divers sur des données simulées ou des images isolées</b>	<b>29</b>
<b>6</b>	<b>Évaluation du potentiel de SIFT sur des données simulées</b>	<b>30</b>
6.1	Effet d'une compression jpeg . . . . .	30
6.2	Ajout de bruit . . . . .	32
6.3	Flou . . . . .	34
6.4	Canaux différents . . . . .	36
6.5	Radiométrie . . . . .	38
6.6	Translation . . . . .	40
6.7	Variation d'échelle . . . . .	41



6.8	Rotation (2D) . . . . .	44
6.9	Similitude (2D) . . . . .	44
6.10	Affinité . . . . .	45
6.11	Paramètres “première octave” et “nombre d’octaves” . . . . .	45
6.11.1	Paramètre “première octave” . . . . .	45
6.11.2	Paramètre “nombre d’octaves” . . . . .	46
<b>7</b>	<b>Critère d’appariement basé directement sur les distances ou sur leur ratio</b>	<b>46</b>
<b>8</b>	<b>Répartition et nombre de points</b>	<b>50</b>
8.1	Problème du nombre de points extraits en fonction du “contraste” . . . . .	50
8.2	Autres problèmes liés au réétalement... . . . . .	51
<b>9</b>	<b>Utilisation d’un prédicteur ?</b>	<b>51</b>
9.1	Quel prédicteur ? . . . . .	51
9.1.1	Prédicteur “global” grossier : zone en recouvrement . . . . .	52
9.1.2	Appariement multiple puis prédicteur . . . . .	52
9.2	Intérêt d’un tel prédicteur . . . . .	52
<b>10</b>	<b>Demi-descripteur</b>	<b>53</b>
<b>11</b>	<b>Possibilité d’utilisation de l’échelle et de l’orientation des points SIFT</b>	<b>54</b>
11.1	Introduction . . . . .	54
11.2	Sensibilité . . . . .	54
11.3	Détermination de $E$ et de $\alpha$ . . . . .	56
11.4	Bilan . . . . .	56
<b>III</b>	<b>Tests sur des images aériennes</b>	<b>57</b>
<b>12</b>	<b>Utilisation de SIFT pour l’aérotriangulation</b>	<b>58</b>
12.1	Méthode et outils . . . . .	58
12.1.1	Méthode . . . . .	58
12.1.2	Mise en œuvre opérationnelle . . . . .	61
12.2	Prises de vue départementales . . . . .	62
12.2.1	Département du Gard (FD30) . . . . .	62
12.2.2	Département de la Manche (FD50) . . . . .	63
12.2.3	Département de la Nièvre (FD58) . . . . .	65
12.2.4	Département de l’Allier (FD03) . . . . .	67
12.2.5	Département de la Dordogne (FD24) . . . . .	68
12.2.6	Département de la Sarthe . . . . .	69
12.2.7	Conclusions . . . . .	70
12.3	Prises de vue urbaine - prises de vue haute résolution . . . . .	76
12.3.1	Chantier TPM . . . . .	76
12.3.2	Chantier MUSARDE : Prise de vue sur Toulouse . . . . .	79
12.3.3	Images obliques avec fort changement de point de vue . . . . .	81
12.3.4	Marseille à 10cm . . . . .	85
12.3.5	Paris à 10 cm . . . . .	86
12.4	Extraction de points de liaison dans des cas anecdotiques . . . . .	87
12.4.1	Points de liaison entre images aériennes floues pour l’aérotriangulation . . . . .	87
12.4.2	Points de liaison entre images satellites pour la spatiotriangulation . . . . .	89
<b>13</b>	<b>Constitution automatique d’un tableau d’assemblage</b>	<b>90</b>
13.1	Présentation du problème et méthodes mises en œuvre . . . . .	90
13.1.1	Méthode 1 : “recherche exhaustive” . . . . .	90
13.1.2	Méthode 2 : “recherche dans une base de données de points SIFT” . . . . .	90
13.2	Prise de vue aérienne sur Toulouse . . . . .	92
13.2.1	Construction du graphe de voisinage . . . . .	92
13.2.2	Extraction des points de liaison et aérotriangulation . . . . .	93

13.3	Chantier de l'Île aux Cochons . . . . .	95
13.3.1	Présentation du chantier . . . . .	95
13.3.2	Méthode et résultats . . . . .	96
<b>14</b>	<b>Recalage d'images</b>	<b>104</b>
14.1	Recalage sur une couche de référence . . . . .	104
14.1.1	Méthode . . . . .	104
14.1.2	Tests effectués . . . . .	104
14.1.3	Conclusions . . . . .	121
<b>IV</b>	<b>Tests sur des images terrestres</b>	<b>123</b>
<b>15</b>	<b>Méthode</b>	<b>123</b>
15.1	Filtrage des fautes . . . . .	123
<b>16</b>	<b>Appariement entre images aériennes et terrestres</b>	<b>124</b>
16.1	Introduction . . . . .	124
16.2	Tests . . . . .	124
16.3	Conclusion . . . . .	127
<b>17</b>	<b>Appariement entre images terrestres acquises par Stéréopolis</b>	<b>129</b>
17.1	Introduction . . . . .	129
17.2	Motifs répétitifs et changement de point de vue . . . . .	129
17.3	Compensation . . . . .	131
17.4	Conclusion . . . . .	136
<b>18</b>	<b>Appariement entre images terrestres système nomade et BD infrastructure</b>	<b>137</b>
18.1	Introduction . . . . .	137
18.2	Difficultés . . . . .	137
18.3	Première méthode : extraction des points homologues avec filtrage "géométrique" des fautes . . .	137
18.3.1	Cas favorable : bons résultats directement . . . . .	138
18.3.2	Mauvais résultats par appariement direct, mais bons résultats en redressant les images . .	140
18.3.3	Résultats peu concluants, même en redressant les images . . . . .	141
18.3.4	Exemple de cas intermédiaires . . . . .	142
18.3.5	Conclusion . . . . .	145
18.4	Deuxième méthode : extraction des points homologues, filtrage des fautes en utilisant toutes les informations . . . . .	145
18.4.1	Description de la méthode . . . . .	145
18.4.2	Quelques résultats sur le jeu test précédent . . . . .	148
18.4.3	Quelques résultats sur un autre jeu test . . . . .	153
18.5	Conclusion . . . . .	157
<b>V</b>	<b>Références bibliographiques et glossaire</b>	<b>159</b>
<b>A</b>	<b>Glossaire</b>	<b>160</b>
<b>B</b>	<b>Bibliographie</b>	<b>161</b>



Première partie

# **SIFT : présentation de l'algorithme et des implémentations disponibles**

# 1 Présentation de SIFT (Scale Invariant Feature Transform)

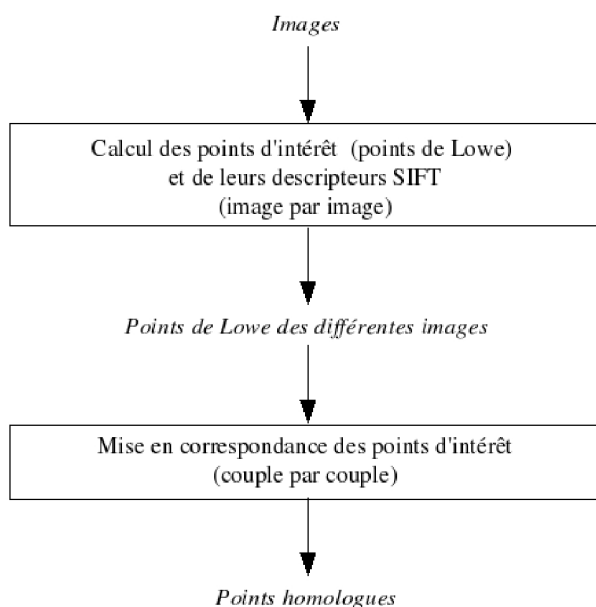
## 1.1 Présentation générale

Proposé par D. Lowe d'abord en 1999 dans [Low99] puis en 2004 dans [Low04], SIFT (*Scale Invariant Feature Transform*) est à la fois un détecteur de points d'intérêt (i.e. une méthode d'extraction de points d'intérêt) au voisinage desquels le comportement de l'image est ensuite décrit au moyen d'un descripteur nommé SIFT et une méthode de mise en correspondance de ces points (dans l'espace de leurs descripteurs).

Le détecteur de points est aussi connu sous le nom de détecteur *DoG* (*Difference of Gaussians*). Il s'agit d'un détecteur de points d'intérêt, ou plus exactement de "points clés" ou "key points" puisque les points détectés ne correspondent pas nécessairement à des détails ponctuels de l'image, mais aussi à des blobs ou de petites régions.

SIFT fonctionne donc en deux étapes bien distinctes :

1. Les points d'intérêts (points de Lowe) sont d'abord extraits de chaque image. Un descripteur leur est ensuite associé : il va décrire le comportement de l'image au voisinage de ces points. Il s'agit d'un vecteur de dimension 128.
2. Pour chaque paire d'images, l'étape de mise en correspondance revient alors à rechercher pour chaque point d'intérêt de la première image son plus proche voisin dans l'espace des descripteurs parmi les points détectés dans la seconde image.



Les grandes qualités et la robustesse des parties descripteur et appariement de SIFT ont été montrées par [MS05] et [JG10]. L'algorithme n'a été initialement conçu que pour être invariant aux variations d'échelle ([Mor08] montre d'ailleurs qu'on ne peut pas vraiment faire mieux à ce niveau), aux rotations (2D dans le plan de l'image) et aux variations d'illumination (il faut toutefois faire attention aux paramètres d'élimination des points les moins contrastés (cf section 8)), mais se comporte en pratique très bien dans le cas de déformations affines restant limitées. Mentionnons toutefois l'existence de la méthode *ASIFT* conçue par être invariante aux affinités et présentée dans [Mor09].

En pratique, SIFT fait donc preuve d'une grande robustesse notamment face aux variations d'échelle, aux rotations, au bruit et aux variations d'éclairage ainsi qu'à des transformations affines "limitées" mais est en revanche assez sensible au phénomène de diachronisme (i.e. lorsque les images n'ont pas été acquises à la même période sous les mêmes conditions d'illumination et présentent de ce fait un aspect différent).

Du fait de son caractère multi-échelle, les points homologues fournis par SIFT ne correspondent pas tous à des détails ponctuels de l'image, mais aussi à des blobs ou de petites régions. Les conséquences de ce fait sur leur précision et la manière de l'améliorer a notamment été étudiée dans [Rem06] dans le cadre d'applications photogrammétriques.

**Remarque :** Il est parfaitement possible de calculer des descripteurs SIFT pour des points extraits par d'autres détecteurs de points d'intérêt (comme ceux listés dans [TM08, MTS<sup>+</sup>04]) ou de manière dense selon une grille.

## 1.2 Extraction de points d'intérêt et calcul des descripteurs

Dans un premier temps, les points de Lowe (points d'intérêt) sont extraits de chaque image. Un descripteur leur est ensuite associé : il décrit le comportement de l'image dans le voisinage du point d'intérêt détecté et sera utilisé lors de l'étape suivante de mise en correspondance des points homologues.

Ce calcul s'effectue donc en 2 grandes étapes :

### 1. Extraction des points d'intérêt

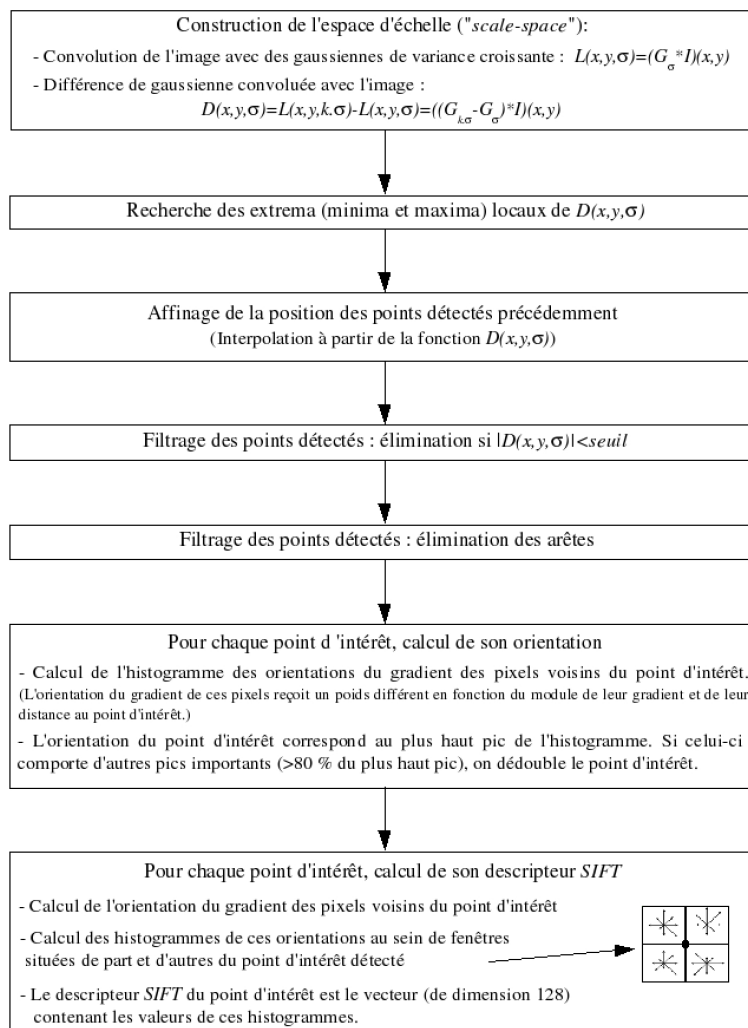
- Extraction des points d'intérêt potentiels (invariants aux changements d'échelle et à l'orientation) : ces points sont sélectionnés dans un espace d'échelles (scale-space) parmi les extrema d'une fonction différence de gaussienne (*DoG*) convoluée avec l'image initiale.
- Affinage de la localisation des points détectés à l'étape précédente par interpolation. Rejet ou conservation des points détectés en fonction de mesures de leur stabilité.

### 2. Calcul des descripteurs

- Calcul de l'orientation (ou des orientations) associée(s) à chaque point d'intérêt détecté.
- Calcul du descripteur SIFT associé à chaque point d'intérêt détecté

#### Extraction des points de Lowe et calcul de leur descripteur SIFT

Notations : On désigne par  $I$  l'image à traiter et par  $G_\sigma$  la gaussienne de variance  $\sigma$



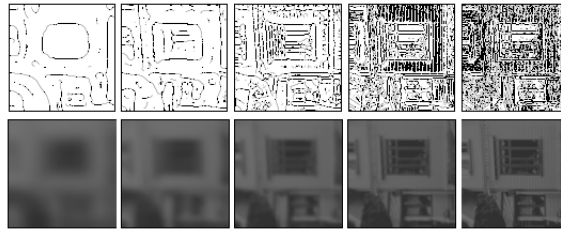
## Algorithme

### A - Extraction des points d'intérêt

0 - Sur-échantillonnage [facultatif] de l'image d'un facteur 2 (interpolation linéaire) puis convolution par une gaussienne

1 - Construction de l'espace d'échelles ("scale-space") et de la fonction *DoG* ("Difference of Gaussians")

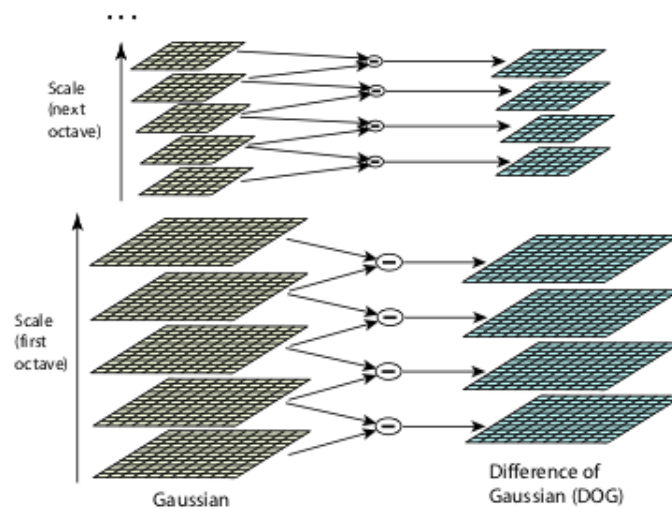
Le rôle d'un espace d'échelles est de s'intéresser à différents niveaux de détails, à différentes échelles (comme dans l'illustration suivante).



### Notations :

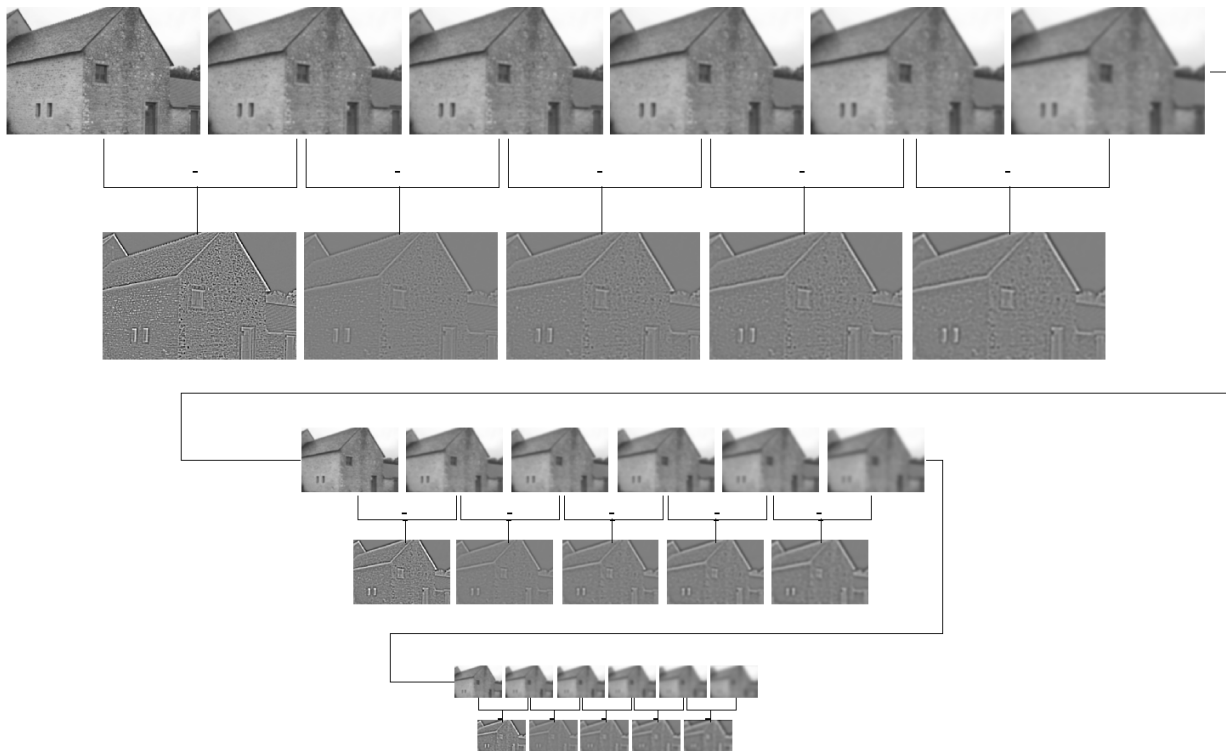
On désigne par  $I$  l'image à traiter et par  $G_\sigma$  la gaussienne de variance  $\sigma$ .

Chaque octave de l'espace d'échelles est divisé en  $N_i$  intervalles. Soit  $k = 2^{\frac{1}{N_i}}$ .

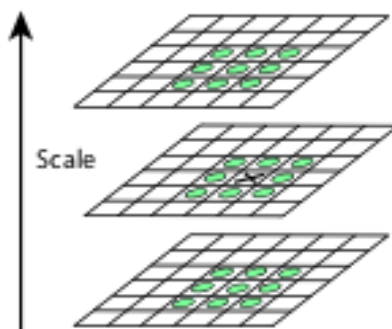


**Algorithme de construction de l'espace d'échelles** $\sigma = \sigma_0$  $I_c \leftarrow I$ **pour tout**  $o$  de 0 à  $N_0$  **faire** $\sigma_t = \sigma_0$ **pour tout**  $i$  de 1 à  $N_i + 3$  **faire** $L(x, y, \sigma) = L_\sigma(x, y) \equiv (G_{\sigma_t} * I_c)(x, y)$ 

Différences de gaussiennes convoluées avec l'image :

 $D(x, y, \sigma) = L_{k.\sigma}(x, y) - L_\sigma(x, y) = ((G_{k.\sigma_t} - G_{\sigma_t}) * I_c)(x, y)$  $\sigma = k.\sigma$  $\sigma_t = k.\sigma_t$ **fin pour**L'image  $L_{2^{o+1}.\sigma_0}$  est sous-échantillonnée d'un facteur 2 et devient la nouvelle image  $I_c$ .**fin pour****Définition :** La fonction  $D$  est connue sous le nom de “*DoG*” (ou *Difference of Gaussians*).**Remarque :** La fonction *DoG* constitue en fait une approximation de la mesure *LoG* (*Laplacian of Gaussian*) utilisée pour caractériser les structures de blobs.**Remarque :** Le fait de sous-échantillonner l'image à la fin de chaque octave est une approximation permettant de réduire les temps de calcul.**2 - Recherche des extrema (minima et maxima) locaux de  $D(x, y, \sigma)$  (i.e. de la fonction *DoG*)**Un tel point  $(x, y, \sigma)$  a une valeur dans le *DoG* non seulement inférieure (ou supérieure) à celles de ses 8 voisins à la même échelle mais également à celles de ses 9 voisins des échelles supérieure et inférieure.**Définition :** Le “*signe*” d’un point SIFT indiquant simplement si ce point correspond à un maximum (signe = +1) ou un minimum (signe = -1) de la fonction  $D$  (i.e. du *DoG*).**Remarque :** On peut conserver cette information sur le “*signe*” des points détectés.





### 3 - Affinage de la localisation des points détectés à l'étape précédente par interpolation à partir de la fonction $D$ : recherche de la position interpolée de l'extremum.

Soit  $P(x; y; \sigma)$  un point détecté précédemment. Soit la matrice  $X = {}^t [x \ y \ \sigma]$ .

On recherche l'extremum  $\hat{X} = X + d\hat{X}$  de  $D$  au voisinage de  $X$  :

$$\begin{bmatrix} \hat{dx} \\ \hat{dy} \\ \hat{d\sigma} \end{bmatrix} = d\hat{X} = - \left( \frac{\partial^2 D}{\partial X^2}(X) \right)^{-1} \cdot \frac{\partial D}{\partial X}(X)$$

(qui correspond au passage par 0 de la dérivée de

$$D(X + dX) = D(X) + {}^t \left( \frac{\partial D}{\partial X}(X) \right) \cdot dX + \frac{1}{2} \cdot {}^t dX \cdot \left( \frac{\partial^2 D}{\partial X^2} \right) \cdot dX$$

(développement de Taylor de la fonction  $D$  au voisinage de  $X$ .)

**si**  $\max(\hat{dx}, \hat{dy}, \hat{d\sigma}) > 0.5$  **alors**

On se trouve en fait face à un nouveau point de coordonnées  $X + d\hat{X}$  pour lequel on effectue une nouvelle interpolation.

**sinon**

Le point  $P$  reçoit les nouvelles coordonnées  $X + d\hat{X}$ .

**finsi**

### 4 - Filtrage des points détectés : élimination des points les moins contrastés

Les points les plus contrastés sont privilégiés car ce sont ceux que l'on aura vraisemblablement le plus de chance de détecter sur les différentes images. Les points les moins contrastés sont donc éliminés par la méthode suivante :

**si**  $\left| D(X) + \frac{1}{2} \cdot {}^t \left( \frac{\partial D}{\partial X}(X) \right) \cdot d\hat{X} \right| < S_{\text{contraste}}$  (avec les mêmes notations qu'à l'étape précédente) i.e.

$\left| D(X + d\hat{X}) \right| < S_{\text{contraste}}$  **alors**

Le point  $P$  est éliminé.

**sinon**

Le point  $P$  est conservé.

**finsi**

### 5 - Filtrage des points détectés : élimination des points situés sur une arête

On souhaite récupérer de véritables points (pour lesquels l'image présente une double discontinuité) et non des points situés sur une arête (et pour lesquels l'image ne présente une discontinuité que dans une seule direction). Les points situés sur une arête ne fixent en effet une contrainte que dans une seule direction.

En conséquence, ils sont éliminés par la méthode suivante :

Soit la matrice hessienne : 
$$\begin{bmatrix} \frac{\partial^2 D}{\partial x^2}(x, y, \sigma) & \frac{\partial^2 D}{\partial x \partial y}(x, y, \sigma) \\ \frac{\partial^2 D}{\partial x \partial y}(x, y, \sigma) & \frac{\partial^2 D}{\partial y^2}(x, y, \sigma) \end{bmatrix}$$
 au point  $P(x; y; \sigma)$ .

si  $\frac{Tr(H(x, y, \sigma))^2}{Det(H(x, y, \sigma))} < \frac{(S_{edge} + 1)^2}{S_{edge}}$  (avec  $Tr(M)$  et  $Det(M)$  respectivement trace et déterminant d'une matrice  $M$ ) alors

Le point  $P$  est conservé.

sinon

Le point  $P$  est rejeté.

fini

## B - Calcul de leur descripteur SIFT (*Scale Invariant Feature Transform*)

### 1 - Pour chaque point d'intérêt, calcul de son orientation

— Calcul de l'histogramme décrivant la répartition des orientations du gradient des pixels voisins du point d'intérêt  $(x, y, \sigma)$  (dans l'image  $L_\sigma$  de l'espace d'échelles correspondant à l'échelle  $\sigma$  du point d'intérêt). L'influence de l'orientation  $\theta$  du gradient d'un de ces pixels voisins  $(u, v)$  lors du calcul de cet histogramme est pondérée par un poids fonction de module  $g(\vec{u}, v)$  du gradient en  $(u, v)$  et de la distance de  $(u, v)$  au point d'intérêt  $(x, y)$ .

— L'orientation du point d'intérêt est celle qui correspond au plus haut pic de l'histogramme. Si celui-ci comporte d'autres pics importants ( $> 80\%$  du plus haut pic), on dédouble le point d'intérêt.

Tous les calculs qui suivent sont effectués relativement à l'orientation, l'échelle et les coordonnées 2D du point d'intérêt afin de garantir l'invariance par rapport à ces paramètres.

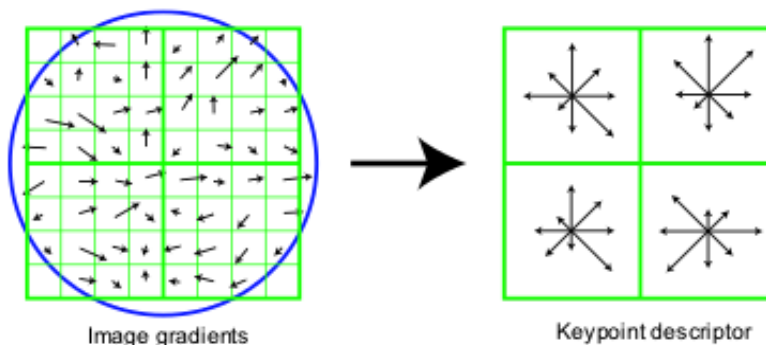
### 2 - Pour chaque point d'intérêt, calcul de son descripteur SIFT

— Calcul de l'orientation (relative à l'orientation du point d'intérêt calculée à l'étape précédente) du gradient des pixels voisins du point d'intérêt (dans l'image  $L_\sigma$  correspondant à l'échelle  $\sigma$  du point d'intérêt).

— Calcul des histogrammes de ces orientations au sein de fenêtres (plus nombreuses que les 4 fenêtres de l'illustration suivante) situées de part et d'autres du point d'intérêt détecté.

— Le descripteur SIFT du point d'intérêt est le vecteur (de dimension 128) contenant les valeurs de ces histogrammes. Ce vecteur est ensuite normalisé afin de garantir l'invariance aux variations d'éclaircissement.

Comme cela a déjà été signalé plus haut, ce descripteur constitue une mesure invariante à l'échelle, aux rotations (2D), aux variations d'éclaircissement ainsi qu'à certaines déformations locales (affinités "légères").



### Plus de détails sur le calcul de l'orientation et du descripteur SIFT

Voyons maintenant plus en détail et étape par étape la détermination de l'orientation et du descripteur d'un point (et plus particulièrement le calcul des histogrammes utilisés).

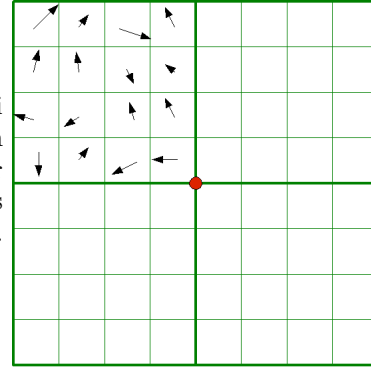
Les calculs de l'orientation et du descripteur se font à l'échelle  $\sigma$  du point d'intérêt  $(x, y, \sigma)$ , autrement dit sur l'image  $L(\sigma)$  de l'espace d'échelles calculé à partir de l'image  $I$ .

On note respectivement  $m(u, v)$  et  $\theta(u, v)$  le module et l'argument du gradient de  $L(\sigma)$  en  $(u, v)$ .

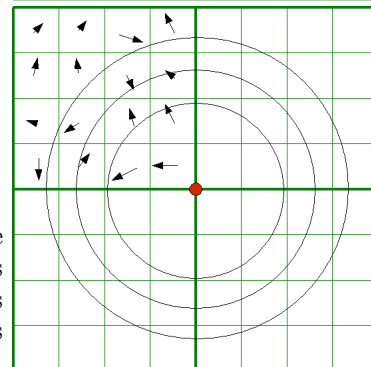
$$m(u, v) = \sqrt{[L(\sigma)(u+1, v) - (L(\sigma))(u-1, v)]^2 + [L(\sigma)(u, v+1) - (L(\sigma))(u, v-1)]^2}$$

$$\theta(u, v) = \text{atan} \left( \frac{(L(\sigma))(u, v + 1) - (L(\sigma))(u, v - 1)}{(L(\sigma))(u + 1, v) - (L(\sigma))(u - 1, v)} \right)$$

On va s'intéresser aux pixels voisins du point d'intérêt (figuré ici par un point rouge). On calcule le gradient de l'image en chacun des pixels situés dans un certain voisinage du point d'intérêt. Sur l'illustration ci-contre, on a représenté par des flèches noires ces vecteurs gradients dans la partie supérieure gauche de ce voisinage.

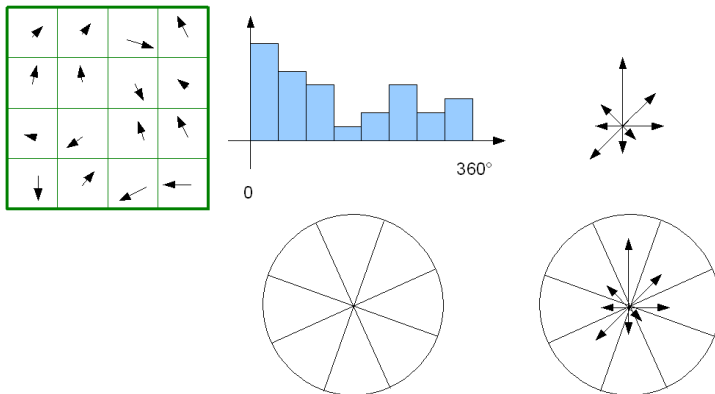
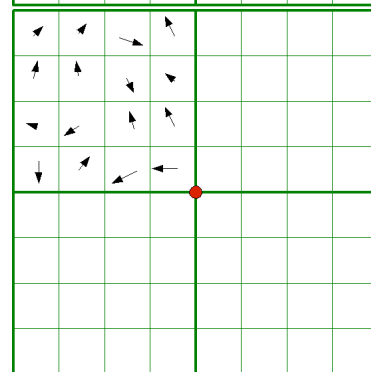


On convolue les modules de ces gradients par une gaussienne centrée sur le point d'intérêt. (Cela va permettre de donner plus d'influence aux pixels les plus proches du point d'intérêt lors du calcul de l'histogramme de répartition des orientations des gradients.)



On obtient  $m_{pond}(u, v) = \text{poids}_{(x,y)}(u, v) \cdot m(u, v)$

avec :  $\text{poids}(u, v) = \frac{1}{2 \cdot \pi \cdot (1,5 \cdot \sigma)^2} e^{-\frac{(x-u)^2 + (y-v)^2}{2 \cdot (1,5 \cdot \sigma)^2}}$



On calcule, pour une discrétisation de pas  $d\theta$  donnée, l'histogramme  $h$  décrivant la distribution de l'orientation de ces gradients (pondérée par leur module et leur éloignement par rapport au point d'intérêt).  
Autrement dit :

$$h(\theta_i) = \sum_{\substack{(u,v) \in V(x,y) \\ \theta_i - \frac{d\theta}{2} < \theta(x,y) < \theta_i + \frac{d\theta}{2}}} m_{pond}(u, v)$$

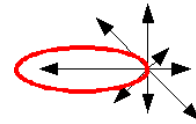
(avec  $V(x, y)$ , voisinage 2D du point de  $(x, y)$ ) soit :

$$h(\theta_i, \theta_{i+1}) = \sum_{\substack{(u,v) \in V(x,y) \\ \theta_i - \frac{d\theta}{2} < \theta(x,y) < \theta_i + \frac{d\theta}{2}}} \text{poids}_{(x,y)}(u, v) \cdot m(u, v)$$

**Dans le cas du calcul de l'orientation du point d'intérêt :**

Dans le cas du calcul de l'orientation du point d'intérêt, on calcule un seul histogramme pour tout le voisinage du point d'intérêt. On en détermine ensuite le maximum : il correspondra à l'orientation qui va être attribuée au point d'intérêt.

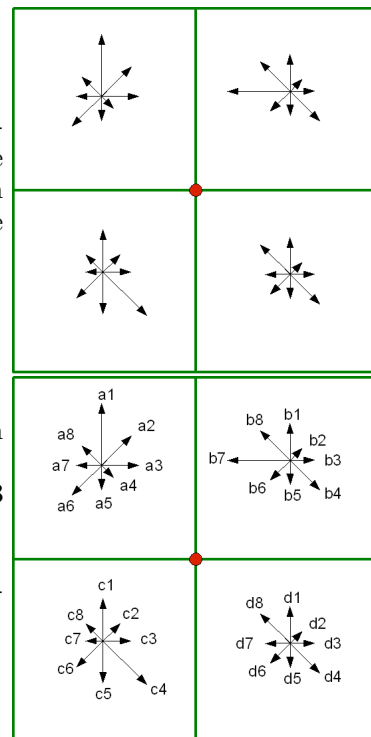
$$\theta_{SIFT} = \arg \max_t h(\theta_t)$$



Au cas où l'histogramme présente un autre pic important ( $> 80\%$  du pic correspondant à  $\theta_{SIFT}$ ), on dédouble le point.

**Dans le cas du calcul du descripteur :**

Dans le cas du calcul du descripteur (représenté ci-contre), on découpe le voisinage du point d'intérêt en plusieurs cases (situés de part et d'autres du point). (En pratique, la grille comporte bien entendu plus de cases que les 4 représentées ci-contre.) On calcule un histogramme au sein de chacune de ces cases.



Les valeurs de ces histogrammes sont ensuite concaténées en un vecteur :

$$V = [ a1 a2 a3 a4 a5 a6 a7 a8 b1 b2 b3 b4 b5 b6 b7 b8 c1 c2 c3 c4 c5 c6 c7 c8 d1 d2 d3 d4 d5 d6 d7 d8 ]$$

On normalise ensuite ce vecteur et l'on obtient alors le descripteur SIFT :

$$V_{SIFT} = \frac{V}{\|V\|}$$

**1.3 Appariement des points d'intérêt**

**Notation** :  $d_{SIFT}(P_a, P_b)$  désigne la distance euclidienne entre les descripteurs SIFT des points  $P_a$  et  $P_b$ .

La deuxième étape consiste donc, pour chaque couple d'image, à mettre en correspondance les points d'intérêt homologues. Il s'agit donc de rechercher pour chaque point d'intérêt extrait de la première image son homologue (s'il existe) parmi les points d'intérêt extraits de la seconde image. Lors de l'étape précédente, un descripteur a été associé chaque point d'intérêt détecté : l'appariement va s'effectuer dans l'espace de ces descripteurs (d'où le terme de transformation dans « *Scale Invariant Feature Transform* »).

La mesure utilisée pour ces mises en correspondance n'est donc pas un coefficient de corrélation, mais la distance euclidienne entre les descripteurs SIFT des deux points. Pour chaque point d'intérêt d'une seconde image (« *image requête* »), on recherche donc parmi les points d'intérêt d'une première image (« *image référence* ») son plus proche voisin au sens de la distance euclidienne entre leurs descripteurs SIFT.

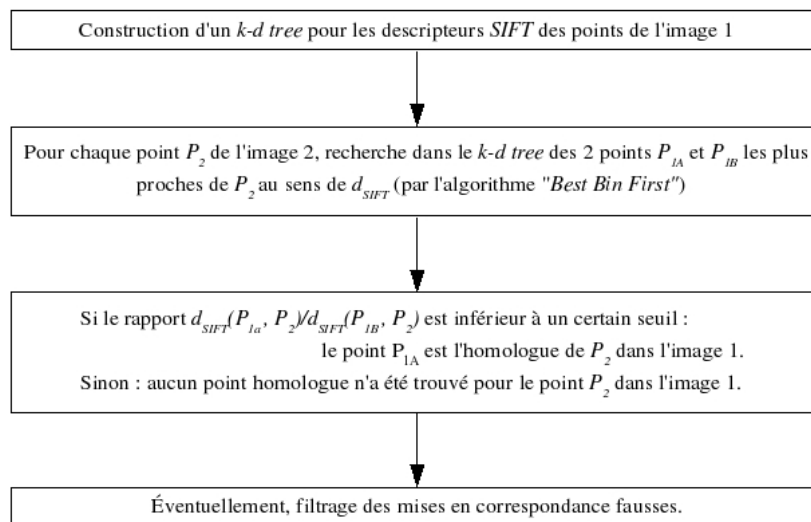
Par ailleurs, afin d'éviter des erreurs de mises en correspondance (notamment dans le cas où le point de l'image requête ne compte pas d'homologue parmi les points d'intérêts extraits de l'image référence), une sécurité est prise en s'intéressant, pour chaque point  $P$  d'intérêt de l'image requête, à ses plus proches voisins  $P_1$  et  $P_2$  parmi les points d'intérêt de l'image référence : si le plus proche voisin  $P_1$  n'est pas un « bien plus

proche voisin” (et donc un bien meilleur candidat) que le second plus proche voisin  $P_2$ , c’est-à-dire si le ratio  $\frac{d_{SIFT}(P, P_1)}{d_{SIFT}(P, P_2)}$  n’est pas inférieur à un certain seuil, alors la mise en correspondance est considérée comme trop ambiguë et est rejetée. Dans le cas contraire, la mise en correspondance est acceptée.

Par ailleurs, du fait du nombre important de points d’intérêts détectés et de la dimension du descripteur SIFT (128), une recherche “naïve” des plus proches voisins nécessite un temps considérable. Cette recherche peut néanmoins être effectuée en un temps raisonnable en utilisant un algorithme de recherche approchée de plus proches voisins basé sur les k-d-tree. Ainsi, la phase de mise en correspondance débute par la construction d’un k-d-tree (dans l’espace SIFT) dans lequel les points d’intérêt de l’image référence sont rangés en fonction de leur descripteur SIFT. Ensuite, pour chaque point d’intérêt de l’image requête, on visite les éléments de ce k-d-tree dans l’ordre de leur plus proche distance (au sens de la distance euclidienne dans l’espace SIFT) à ce point de l’image requête dont on recherche les plus proches voisins. Par ailleurs, on considère que les deux plus proches voisins recherchés font partie des *maxPtsAVisiter* (en pratique, *maxPtsAVisiter* = 200) premiers points visités : on ne parcourt donc pas l’ensemble du k-d-tree, mais on se limite au *maxPtsAVisiter* premiers points visités (d’où le terme de recherche approchée).

### Mise en correspondance des points de Lowe

Notation :  $d_{SIFT}(P_1, P_2)$  désigne la distance euclidienne entre les descripteurs SIFT des points  $P_1$  et  $P_2$ .



### Algorithme

1 - Construction d’un k-d tree pour les descripteurs SIFT des points de l’image de référence (image 1)

2 - Pour chaque point d’intérêt de l’image requête (image 2), recherche de son homologue parmi ceux de l’image référence (image 1)

---

**pour tout** point d'intérêt  $P$  de l'image requête (image 2) **faire**

Recherche par l'algorithme best-bin-first dans le k-d-tree des deux points  $P_1$  et  $P_2$  de l'image référence (image 1) les plus proches de  $P$  au sens de la distance euclidienne dans l'espace SIFT (128 dimensions) avec  $d_{SIFT}(P_1, P) < d_{SIFT}(P_2, P)$ .

(L'algorithme de recherche des plus proches voisins utilisé utilise un ordre de visite du k-d tree modifié (visite des éléments du k-d-tree dans l'ordre de leur plus proche distance au point dont on recherche les plus proches voisins). Il s'agit en outre d'une recherche approchée puisqu'on ne parcourt pas l'ensemble du k-d-tree mais que l'on considère que les deux plus proches voisins font partie des *maxPtsAVisiter* (auquel on attribue en pratique la valeur 200) premiers points visités.)

**si**  $\frac{d_{SIFT}(P, P_1)}{d_{SIFT}(P, P_2)} < \text{seuil}_{\text{appariement\_ambigu}}$  (en pratique :  $\text{seuil}_{\text{appariement\_ambigu}} = 0.8$ ) **alors**

$P_1$  est l'homologue de  $P$  dans l'image de référence.

**sinon**

$P$  n'a pas d'homologue parmi les points d'intérêt de l'image de référence (image 1).

**finsi**

**fin pour**

---

[ 3 - Éventuellement filtrage direct des mises en correspondance fausses par l'une des méthodes de la section 5 ]

## 2 Utilisations possibles

SIFT a d'ores et déjà été utilisé pour différentes applications :

- **Reconnaissance d'objet** : Cette application est d'ailleurs celle qui était proposée dans l'article de référence [Low04] présentant l'ensemble de la méthode *SIFT*.
- **Extraction de points homologues** entre clichés utilisés ensuite pour différentes applications :
  - Constitution automatique de tableau d'assemblage à partir d'un ensemble de clichés désordonnés (ou du moins savoir quels clichés se recouvrent). (voir partie 13)
  - Calcul de points de liaison pour tout calcul d'orientation 3D d'images et notamment d'aérotriangulation. (voir [LF06] et partie 12)
  - Calcul de points homologues en vue d'un recalage. (voir partie 14)
  - Réalisation automatique de panoramiques. [BL07]
- **Modélisation de la texture** : Utilisation des descripteurs comme un canal de texture pour des applications de classification. [YN08]
- **Indexation d'images, sac de mots (*bag of words*)...** : Les descripteurs SIFT extraits d'une image sont utilisés pour indexer des images dans une base de données (calcul d'un descripteur "global" par image à partir de tous ses descripteurs SIFT) [GD05], décrire des types d'images ou des parties d'images [Lie09].
- ...

## 3 Implémentations

### 3.1 Détection des points et calcul des descripteurs

#### 3.1.1 Implémentations CPU disponibles et testées

Plusieurs implémentations sont disponibles. Parmi celles qui ont été testées, on citera notamment :

- l'exécutable fourni par D. Lowe [Low05]. Cette implémentation est celle qui donne généralement les meilleurs résultats. Il s'agit malheureusement d'une boîte noire (pas d'accès aux codes, pas de possibilité de jouer sur les paramètres, taille des images limitée).
- *Autopano-sift* / *libsift* : il s'agit d'outils écrits en langage C# [Now05]
- une implémentation en C utilisant *OpenCV* [Hes09]
- *sift++* : Il s'agit de l'implémentation en langage C++ d'une version légèrement modifiée de l'algorithme SIFT ([Ved07b], [Ved07a]). [BSP07] présente les résultats d'une comparaison de cette implémentation à celle de Lowe.
- *vlfeat* : Développée par le même auteur que *sift++*, *vlfeat* est une librairie en langage C++ offrant un certain nombre d'outils de vision par ordinateur parmi lesquels SIFT. *sift++* n'est en fait plus maintenu et a été remplacé par *vlfeat*. [VF08]

Ces trois dernières implémentations présentent l'avantage d'être écrites en C++ et d'être modifiables (codes sources disponibles). Par ailleurs, elles offrent la possibilité de jouer sur bon nombre de paramètres.

L'implémentation *sift++* ainsi que celle basée sur *OpenCV* sont celles qui ont fait l'objet de davantage de tests. Elles ont été modifiées de manière à pouvoir traiter des images de taille quelconque (au moyen d'un traitement par dalle). Des fuites de mémoires ont également été corrigées dans *sift++*. Dans la suite, c'est *sift++* qui a été utilisée, mais ces deux implémentations donnaient des résultats généralement assez semblables.

#### 3.1.2 Implémentations GPU

Il existe également des implémentations exploitant la puissance de calcul des GPU (voir par exemple [SFPG06]). Les temps de calcul sont alors fortement réduits. Parmi les implémentations GPU disponibles, on pourra citer :

- *SiftGPU* Cette implémentation, dont le code source est disponible offre la possibilité de jouer sur bon nombre de paramètres. [Cha07]
- *CudaSIFT* (code source disponible, implémentation non testée) [Bjö08]

Remarque : *SiftGPU* a été modifié (traitement des images par dalle pour pouvoir gérer des images de grande taille, au lieu du comportement par défaut consistant à sous-échantillonner l'image) et testé. Néanmoins, pour des raisons matérielles (nécessité de disposer d'une machine avec une carte graphique compatible), ces tests sont restés moins nombreux que pour les versions CPU.

### 3.2 Mise en correspondance

Pour la mise en correspondance, la librairie ANN a été utilisée (voir [MA09]). Il s'agit d'une librairie (écrite en C++) qui comporte des structures de données (k-d tree, ...) et des algorithmes permettant la recherche de plus proches voisins exacts ou approchés dans des espaces de grandes dimensions. Le calcul des plus proches voisins exacts dans des espace de dimension supérieure à 8 est en effet une tâche difficile pour laquelle peu de méthodes semblent significativement meilleures que le calcul de toutes les distance... Cependant, il est apparu que le calcul de plus proches voisins approchés est possible avec des temps de calcul significativement meilleurs et le plus souvent des erreurs relativement faibles (voir [BL97, AM93, AMN<sup>+</sup>98]). ANN offre la possibilité de paramétrer cette recherche approchée (en spécifiant une limite maximale à l'erreur d'approximation et donc en choisissant un compromis entre précision et temps de calcul). Dans le cas présent, le fait de se limiter aux 200 premiers points visités lors de la recherche de plus proches voisins (ainsi que le recommande [Low04]) permet de considérablement réduire le temps de calcul. Les temps de calcul passent alors en  $O(n \cdot \log(n))$  au lieu de  $O(n^2)$ . Dans le cas d'images aériennes 4096x4096, on passe ainsi de plusieurs heures à quelques minutes.

**Remarque :** Dans notre cas, afin de pouvoir gérer en mémoire de grandes quantités de descripteurs, le fichier ANN.h a été modifié afin de choisir des *short* comme type de coordonnées. (`typedef short ANNcoord;` au lieu de `typedef double ANNcoord;` par défaut.

## 4 Outils développés

### 4.1 KeyPoint : gestion des points et de leurs descripteurs

#### 4.1.1 Classe *Keypoint*

La classe **Keypoint** gère les points d'intérêt ainsi que le descripteur qui leur est associé.

Ses données membres sont :

- les coordonnées (image) 2D du point (`_x`, `_y`)
- l'échelle caractéristique du point (`_sigma`)
- l'orientation caractéristique du point (`_orientation`)
- le signe du point (`_signe`), c'est-à-dire si le point correspond à un maximum ou un minimum du DoG pour SIFT, ou le signe du Laplacien pour SURF
- le descripteur (`_desc`)

La classe fournit également des fonctions de lecture/écriture des points d'intérêt dans différents formats de fichier. On citera notamment les fonctions :

- **lireBin** et **ecrireBin** : format binaire simple développé au MATIS (et décrit en 4.2)
- **lireKeyLowe** et **ecrireKeyLowe** : format texte des exécutable de D. Lowe.

D'autres outils sont également disponibles (distance entre les descripteurs, itérateur pour accéder aux valeurs du descripteur).

**Keypoint::Keypoints** est un `typedef` de `std::vector<Keypoint>`.

#### 4.1.2 Namespace *OutilsIOKeypoint*

Ce namespace contient différentes fonctions pour la lecture/écriture des différents formats de fichiers contenant les objets **Keypoint**.

On citera notamment les fonctions :

- **ChargerKeyBin** et **EcrireKeyBin** : format binaire simple développé au MATIS
- **ChargerKeyLowe** et **EcrireKeyLowe** : format texte des exécutable de D. Lowe.

Ce namespace contient également la classe abstraite **FluxSortieKeypoint** et ses classes filles.

Ces classes permettent d'envoyer les objets **Keypoint** vers un flux de sortie. Elles comportent au minimum une fonction **size()** qui renvoie le nombre de points (déjà envoyés dans le flux) et un opérateur `<<` permettant d'ajouter un nouveau point (i.e. l'envoyer dans le flux).

La classe **FluxSortieKeypoint\_vector** permet ainsi d'envoyer les objets **Keypoint** dans un `std::vector<Keypoint>`. La classe **FluxSortieKeypoint\_fichierbinaire** permet d'écrire directement les objets **Keypoint** dans un fichier binaire (format binaire simple développé au MATIS décrit dans la partie 4.2). Cette classe comporte également deux méthode **commencer()** et **finir()**, chargées de s'occuper de l'entête du fichier.

#### 4.1.3 Namespace *OutilsKeypoint*

Ce namespace contient quelques outils pour manipuler les objets **Keypoint**.

### 4.2 Format binaire de stockage des points SIFT

Pour des raisons de volume de données, il s'est avéré indispensable de stocker les points extraits accompagnés de leur descripteur dans des fichiers binaires. Le format suivant a été retenu (il peut être utilisé aussi bien pour des points *SIFT* que pour des points *SURF*) :

**En-tête :**

- (unsigned int) : nombre de points contenus dans le fichier
- (unsigned int) : dimension du descripteur associé à chaque point (128 pour *SIFT*)

**Pour chaque point :**

- (float) : x
- (float) : y
- (float) : sigma dont le signe indique si le point a été détecté en tant que maximum ou minimum du DoG et dont la valeur absolue donne l'échelle (sigma) à laquelle il a été détecté
- (float) : orientation
- (unsigned char) .... (unsigned char) descripteur associé à ce point (i.e. vecteur dont la dimension a été donnée dans l'en-tête)



### 4.3 Extraction des points de Lowe : *Siftpp\_tgi*

#### 4.3.1 Dépendances

Cet outil a été développé à partir de l'implémentation open-source *sift++* écrite par A. Vedaldi [Ved07b] et disponible à l'URL suivante : <http://www.vlfeat.org/~vedaldi/code/siftpp.html>

Remarque : *sift++* n'est plus maintenu par son auteur. Il a en effet été remplacé par une nouvelle librairie intitulée *VLFeat*.

#### 4.3.2 Modifications apportées à *sift++*

- Prise en compte du “signe” (minima ou maxima du DoG). Cette information peut en effet être utile pour filtrer des fautes ou réduire les temps de calcul (en choisissant de ne détecter que les maxima ou les minima du *DoG*.)
- Correction de problèmes de “fuites mémoires”
- Développement d'un enrobage autour de *sift++*, afin de traiter de grandes images (traitement par dalle), et de gérer différents paramètres.
- ( Introduction de marges (notion de zone utile) )
- Ajout de quelques options supplémentaires

#### 4.3.3 *SiftDriver*

La classe **SiftDriver** est la classe qui “encapsule” les outils de *sift++*. Elle se paramètre à l'aide d'un objet de la classe **ParametrageSiftDriver**. Ses fonctions **traiter\_image(...)** permettent de traiter l'image dans son intégralité, ou d'en traiter seulement certaines parties (fenêtres rectangulaires). Dans les deux cas, le traitement s'effectue par dalle (pour des raisons de gestion de la mémoire).

1. La classe **Dallage** permet de calculer les paramètres du découpage de l'image en dalles, de manière à obtenir le découpage le plus régulier possible en tenant compte à la fois de la taille de l'image et des paramètres du dallage souhaité spécifié par un objet de la classe **ParametrageDallage**.
2. La classe **PreparationImage** propose les outils permettant de préparer la dalle qui va être traitée :
  - (a) crop de la dalle à traiter
  - (b) prétraitements (sélection du canal à traiter, calcul éventuel du gradient, réétalement)
  - (c) transformation en un objet de type **VL::BufferImage**
3. Le traitement SIFT proprement dit est ensuite mis en œuvre par la fonction **traiter\_dalle(...)**

Il est possible de décider ce que l'on souhaite récupérer à l'aide d'une classe fille de la classe **FluxSortieKeypoint**. Ainsi, on peut choisir de récupérer un **std::vector<Keypoint>** (en utilisant la classe **FluxSortieKeypoint\_vector**), ou de les enregistrer directement dans un fichier binaire (à l'aide de la classe **FluxSortieKeypoint\_fichierbinaire**). (Ces classes comportent au minimum une fonction **size()** qui renvoie le nombre de points et un opérateur **<<** permettant d'ajouter un nouveau point.)

#### 4.3.4 Paramétrage

Le paramétrage est géré à l'aide des classes de paramétrages suivantes :

- **ParametrageDallage** : paramètres liés au dallage des images
- **ParametragePretraitement** : paramètres liés a divers prétraitements (choix du canal, gradient ? , réétalement)
- **ParametrageSift** : paramètres liés directement à *SIFT*.
- **ParametrageSiftDriver** : regroupement des paramètres précédents

Les différents paramètres sont les suivants :

##### Paramètres liés directement à SIFT (*ParametrageSift*) :

- *threshold* : seuil de contraste
- *edgeThreshold* : seuil de distinction arête/coin
- *firstoctave* : choix de la première octave (par défaut -1)
- *nboctaves* : nombre d'octaves
- *nlevels* : nombre de niveaux par octave
- *signe* : détection des minima (-1) ou des maxima (1) du DoG, ou des deux (0)
- *noorient* : ne pas calculer l'orientation caractéristique des points d'intérêt

- *nodescr* : ne pas calculer de descripteur pour les points d'intérêt détectés
- *demidescripteur* : permet de calculer un "demi descripteur". Les histogrammes des orientations des gradients sont alors calculés pour des valeurs modulo  $\pi$  au lieu de modulo  $2\pi$ . (Autrement dit, on s'intéresse uniquement à la direction du gradient, et non à son sens.)
- *siftdense* : pas d'échantillonnage pour une extraction SIFT dense. Dans ce cas, les points d'intérêt ne sont pas détectés comme extrema de la fonction *DoG*, mais comme les nœuds d'une grille régulière (au niveau le plus bas de l'espace d'échelles).

#### Paramètres de prétraitements des images (*ParametragePretraitement*) :

- *maxim* : borne supérieure pour le réétalement de l'image (par défaut 0) *sift++* traite en effet des images dont les valeurs sont comprises entre 0 et 1. *maxim* permet de définir la borne supérieure. Dans le cas contraire, (si  $\text{maxim} \leq 0$ ), on réétalement entre 0 et la valeur maximale de la dalle (ce qui peut être dangereux en cas de présence de valeurs aberrantes.)
- [ *minim* : borne inférieure pour le réétalement de l'image (par défaut 0)
- *canal* : choix du canal à traiter (par défaut 0). Si valeur aberrante (négative ou supérieure au nombre de canaux de l'image), on traite l'intensité.
- *usegradient* : utilisation d'une image de gradient (norme du gradient)

#### Paramètres du dallage souhaité (*ParametrageDallage*) :

- *taille\_souhaitee\_dalle* : taille souhaitée pour chaque dalle
- *taille\_minimale\_dalle* : taille minimale de chaque dalle
- *marge* : marge à prendre en compte (recouvrement entre les dalles)

**Exemple d'utilisation :** Extraction des points d'intérêt d'une image située à l'adresse *adresse\_image*. Les paramètres sont spécifiés via l'objet *parametres*. Ces points sont stockés dans le `std::vector<KeyPoint> vkp`.

```
void Exemple_vector(const std::string & adresse_image,
                   const ParametrageSiftDriver & parametres,
                   std::vector<KeyPoint> & vkp) {
    //Création du flux de sortie
    OutilsIOKeyPoint::FluxSortieKeypoint_vector fvkp(vkp);
    //Traitement SIFT de l'image
    SiftDriver siftdriver(parametres);
    siftdriver.traiter_image(adresse_image, fvkp);
    std::cout<<fvkp.size()<<" points"<<std::endl;
}
```

#### 4.3.5 Exécutable

L'exécutable se lance de la manière suivante :

```
./exe [--verbose | -v] [--help | -h] [--output | -o NAME] [--prefix | -p PREFIX] [--no-descriptors]
[--no-orientations] [--levels | -S NUMBER] [--octaves | -O NUMBER] [--first-octave | -f NUMBER]
[--threshold | -t NUMBER] [--edge-threshold | -e NUMBER] [--canal NUMBER] [--gradient] [--maxim
NUMBER] [--siftdense NUMBER] [--signe NUMBER] [--demidescripteur] IMAGE [IMAGE2 ...]
```

Les options proposées par l'exécutable sont les suivantes :

--verbose	Active la "verbosité"
--help	Affiche le message d'aide
--output=NAME	Sauvegarde des points détectés dans le fichier situé à l'adresse NAME (utilisable uniquement si on lance sur une seule image)
--prefix=PREFIX	Sauvegarde des points détectés à partir d'une image "./im.ext" dans un fichier nommé "PREFIX/im.key"
--octaves=O	Nombre d'octaves
--levels=S	Nombre de niveaux par octave
--first-octave=MINO	Indice de la première octave (par défaut -1)
--threshold=THR	Seuil de contraste ("Keypoint strength threshold")
--edge-threshold=THR	Seuil Point VS Arête ("On-edge threshold")
--canal=C	Canal à traiter (par défaut, 0) (si C<0 ou C>NbCanaux, calcul sur l'intensité)
--maxim=VAL	Vraie valeur maximale (calculée) ou prédéfinie
--signe=S	Détection des minima(-1) ou maxima(1) du DoG ou des deux(0)
--no-descriptors	Pas de calcul des descripteurs
--no-orientations	Pas de calcul des orientations
--demidescripteur	Calcul d'un "demi-descripteur". Les histogrammes des orientations des gradients sont alors calculés pour des valeurs modulo $\pi$ au lieu de $2\pi$ . (Autrement dit, on s'intéresse uniquement à la direction du gradient, et non à son sens.)
--siftdense=PAS	Pas d'échantillonnage pour une extraction SIFT dense. Dans ce cas, les points d'intérêt ne sont pas détectés comme extrema de la fonction <i>DoG</i> , mais comme les nœuds d'une grille régulière (au niveau le plus bas de l'espace d'échelles). On extrait alors un point tous les PAS pixels.

## 4.4 Appariement

### 4.4.1 Dépendances

Les outils d'appariement utilisent la bibliothèque *ANN* (Approximate Nearest Neighbour) de D. M. Mount et S. Arya disponible à l'URL suivante : <http://www.cs.umd.edu/mount/ANN/>

Une nouvelle librairie nommée *FLANN* est parue depuis et semble également intéressante (et apparemment plus rapide).

### 4.4.2 Classe *AppariementDescripteurs*

C'est cette classe qui contient les fonctions d'appariement dans l'espace des descripteurs.

**Exemple d'utilisation :** Appariement de deux ensembles de points d'intérêt (accompagnés de leurs descripteurs) Ces points sont stockés dans les `std::vector<KeyPoint> kpdata` et `kpquery`. Le résultat est stocké dans l'objet `ResultMEC<T2D<double>> mec`. Le paramètre `usefiltragevoisinage` permet de décider si l'on effectue un filtrage des fautes par considération du voisinage ou non.

```
void Exemple_appariement(const std::vector<KeyPoint> & kpdata,
                        const std::vector<KeyPoint> & kpquery,
                        const bool usefiltragevoisinage,
                        ResultMEC<T2D<double>> & mec) {
    //Appariement
    AppariementDescripteurs appdesc;
    std::vector<unsigned int> iddata_app, idquery_app;
    //iddata_app et idquery_app pour les indices des points appariés
    appdesc.apparie(kpdata,kpquery,iddata_app,idquery_app);

    //On récupère uniquement les points 2D
    mec.pts1().resize(iddata_app.size());
    mec.pts2().resize(idquery_app.size());
    for(int n=0;n<mec.size();n++){
        mec.pt1(n)=kpdata[iddata_app[n]].xy();
        mec.pt2(n)=kpquery[idquery_app[n]].xy();
    }
}
```

```

    }
    if(verbose)std::cout<<mec.card()<<" points apparies."<<std::endl;

    //FiltrageVoisinage (au cas ou)
    if(usefiltragevoisinage){
        FiltreVoisinage filtrevoisinage;
        filtrevoisinage.filtre(mec.pts1(),mec.pts2());
    }
    std::cout<<mec.card()<<" points apparies."<<std::endl;
}

```

#### 4.4.3 Exécutable

Il s'utilise de la manière suivante :

```
./ann_samplekey_generique --kp1 adresse_data --kp2 adresse_query [--result adresse_result ] [--filtrevo
(par défaut inactif)] [--filtresigne (par défaut inactif)] [--emprises adresse_emprises] [--verbose]
```

Les arguments et options proposées par l'exécutable sont les suivantes :

--kp1=ADRESSE1	Adresse du fichier des points+descripteurs de l'image 1 (image "référence")
--kp2=ADRESSE2	Adresse du fichier des points+descripteurs de l'image 2 (image "requête")
--result=RESULT	Adresse du fichier contenant le résultat de la mise en correspondance (par défaut, calcul automatique à partir de ADRESSE1 et ADRESSE2)
--emprises=EMP	Adresse du fichier contenant l'emprise de l'image 1 dans l'image 2 ainsi que celle de l'image 2 dans l'image 1. Ces emprises sont utilisées comme prédicteur. (par défaut, pas d'emprises et pas de prédicteur)
--filtrevoisinage	Active le filtrage des fautes par considération du voisinage (par défaut inactif)
--filtresigne	Active le filtrage des fautes en fonction du "signe" des KeyPoint (par défaut inactif)
--verbose	Active la "verbosité"

## 5 Filtrer les fautes

Plusieurs méthodes sont possibles pour filtrer les fautes.

- Filtre aller-retour
- Filtre voisinage (2,5D)
- Filtre affinité / homographie (plan)
- Filtre photogrammétrique...
- Filtre signe
- Filtre similitude par accumulation (plan)
- Suppression des doublons

### 5.1 Méthodes “générales”

Ces méthodes peuvent s’appliquer à tous les cas de figure (aucune contrainte géométrique sur les données nécessaires, ni sur la scène imagée).

#### 5.1.1 Filtre “signe”

Rappel : Les points SIFT sont les extrema (c’est-à-dire à la fois les minima et les maxima locaux) du DoG. Le “signe” d’un point SIFT indique simplement si ce point correspond à un maximum (signe = +1) ou un minimum (signe = -1) du DoG. (Dans le cas d’un point SURF, il correspond au signe du Laplacien.)

Ce filtrage peut être utilisé dans les cas où les images ne présentent pas d’inversion de contraste. Il consiste à vérifier que les points SIFT mis en correspondance ont bien le même “signe”.

---

```

pour tout ( $P_1, P_2$ ) paire de points appariés faire
  si  $P_1.signe = P_2.signe$  alors
    L’appariement est conservé.
  sinon
    L’appariement est rejeté.
  finsi
fin pour

```

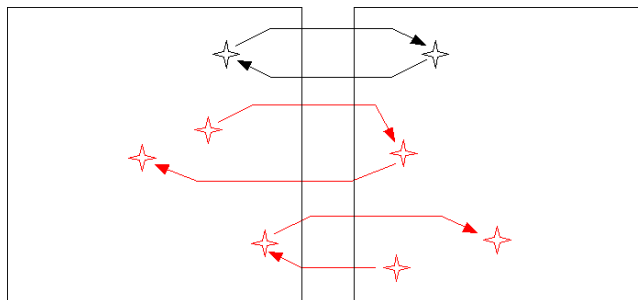
---

En pratique, lors des tests effectués avec SIFT, il est apparu qu’il était vraiment très rare que des points de signe différent soient mis en correspondance, rendant ce filtrage le plus souvent fort peu utile.

#### 5.1.2 Détection des erreurs par appariement croisé, filtre aller-retour

Le principe de l’appariement croisé est de vérifier que l’on obtient bien les mêmes paires de points homologues, que l’on recherche les homologues dans l’image 2 des points de l’image 1, ou à l’inverse les homologues dans l’image 1 des points de l’image 2.

FIGURE 1 – Appariement croisé : seul le premier appariement est conservé.



---

```

pour tout  $P_1$  de l'image 1 faire
  Recherche de son homologue  $P_2$  dans l'image 2.
  Recherche de l'homologue  $P_{1bis}$  de  $P_2$  dans l'image 1.
  si  $P_1 = P_{1bis}$  alors
    L'appariement  $(P_1, P_2)$  est conservé.
  sinon
    L'appariement est rejeté.
  finsi
fin pour

```

---

## 5.2 Méthodes contraintes par le type de scène imagée

### 5.2.1 Détection des erreurs par considération du voisinage

Ce filtrage très simple (illustré par la figure 2) vise à éliminer des erreurs grossières. Il peut être utilisé dans les cas où la scène imagée est une surface continue en "2,5D" (et non en vraie 3D) aux variations généralement pas trop brutales, comme par exemple dans le cas de l'imagerie aérienne.

Le principe en est simple : on vérifie qu'un point a majoritairement les mêmes voisins dans les deux images, i.e. que les homologues dans l'image 2 des  $N$  plus proches voisins d'un point  $P$  de l'image 1 se retrouvent majoritairement parmi les  $N$  plus proches voisins de l'homologue de  $P$  dans l'image 2.

(**Remarque** : Il s'agit ici de plus proches voisins dans l'espace image 2D et non plus dans l'espace des descripteurs.)

L'algorithme est donc le suivant :

---

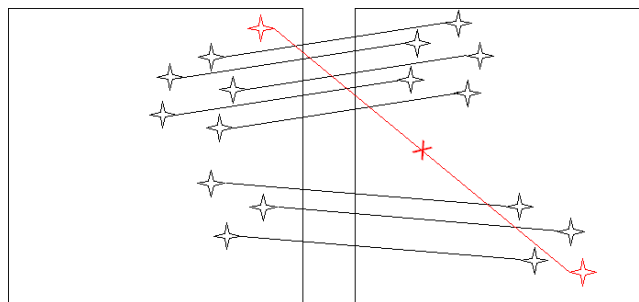
```

pour tout point apparié  $P$  de l'image 1 faire
   $S1 = \{N \text{ plus proches voisins de } P \text{ parmi les points appariés de l'image 1} \}$ 
   $S2 = \{N \text{ plus proches voisins de l'homologue de } P \text{ parmi les points appariés de l'image 2} \}$ 
   $nbvc = \text{nombre de points de } S1 \text{ dont l'homologue est dans } S2$ 
  si  $\frac{nbvc}{N} < \text{seuil}$  alors
    Le point  $P$  et son homologue sont rejetés.
  sinon
    Le point  $P$  et son homologue sont conservés.
  finsi
fin pour

```

---

FIGURE 2 – Détection des fautes par considération du voisinage : l'appariement marqué par un trait rouge (et une croix) est éliminé car son "comportement" est différent de celui des points voisins.



**Remarque** : Ce filtrage est destiné aux scènes "2,5D", ce qui correspond (en étant assez large sur la notion de "2,5D...") à la majeure partie des données que nous avons à traiter.

### 5.2.2 Filtre "transformation paramétrique entre images" : similitude / affinité / homographie

Cette méthode est adaptée au cas où le type de transformation entre les deux images à apparier est connu. Dans le cas de scènes planes, cette transformation peut se modéliser (au moins de manière approximative)

par une affinité ou une homographie. La méthode de filtrage consiste donc à estimer les paramètres de cette transformation (par RANSAC ou non) à partir des coordonnées des points homologues détectés puis à éliminer les points considéré comme des outliers à l'issue de ce calcul.

### 5.2.3 Filtre similitude par accumulation (plan)

Il s'agit d'une "variante" de la méthode précédente adaptée au cas particulier où la transformation entre les deux images est une similitude 2D (comme c'est notamment le cas entre deux images de façades redressées, ou entre images aériennes ortho-rectifiées de manière approximative).

Cette méthode va exploiter toutes les informations associées aux points SIFT mis en correspondance. A chaque point SIFT extrait d'une image sont en effet associées les informations suivantes :

- ses coordonnées image (2D)  $(x; y)$
- son échelle caractéristique  $\sigma$
- son orientation caractéristique  $\theta$
- son descripteur  $V_{SIFT}$

Dans les méthodes d'élimination des fautes précédentes, seules les coordonnées image des points ont été utilisées. Néanmoins, les informations fournies par les échelles  $\sigma$  et les orientations  $\theta$  caractéristiques associées aux points appariés peuvent être utiles dans le cas d'une similitude (2D) entre les images (ou d'un de ses cas particuliers : simple application d'un facteur d'échelle ou simple rotation). En effet, si l'on prend en compte ces informations associées aux points SIFT mis en correspondance, chaque appariement fournit une hypothèse de similitude. Dans la suite, on note  $P_i(x_i, y_i, \sigma_i, \theta_i)$  un point SIFT de l'image  $i$  avec  $(x_i; y_i), \sigma_i$  et  $\theta_i$  désignant respectivement les coordonnées image, l'échelle et l'orientation associées à  $P_i$ .

Soit la similitude 2D  $S : S(x; y) = E \cdot \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix}$

Les paramètres de cette transformation peuvent être estimés en deux étapes à partir **d'un seul appariement (correct)**  $(P_1, P_2)$ ,  $P_2 = S(P_1)$  :

1. Dans un premier temps, le facteur d'échelle  $E$  et l'angle de rotation  $\alpha$  de  $S$  peuvent être estimés indépendamment l'un de l'autre à partir des informations d'échelle et d'orientation associées à ces deux points.  $E = \frac{\sigma_2}{\sigma_1}$  et  $\alpha = \theta_2 - \theta_1$ .
2. Dans un deuxième temps, la translation  $(T_x, T_y)$  peut être estimée.

$$\begin{pmatrix} T_x \\ T_y \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - E \cdot \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

Dans le cas où l'on a plusieurs appariements parmi lesquelles des fautes, on va considérer que chaque appariement apporte une hypothèse sur les paramètres de  $S$ , le "meilleur" jeu de paramètre étant obtenu par accumulation et vote.

On peut ensuite détecter les outliers et les éliminer.

L'intérêt de cette méthode est qu'elle fonctionne même pour très peu d'appariements. Sa faisabilité est vérifiée en 11.

## 5.3 Autres méthodes

### 5.3.1 Filtrage par reprojection

Ce filtrage est plus particulièrement adapté aux cas d'extraction de points de liaison pour des calculs d'aérottriangulation. Il nécessite en effet de disposer d'un bon géoréférencement approché des deux images ainsi que d'un modèle 3D ("2,5D") même approché de la surface imagée (autrement dit d'un MNT). La méthode consiste pour chaque paire de points homologues à re-projeter le point de l'image 1 dans l'image 2, afin de calculer la distance entre cette position théorique et la position réelle de son homologue. On en déduit la distribution de ces écarts ainsi que des indicateurs statistiques classiques (moyenne, médiane, écart-type...) la décrivant. Les outliers par rapport à ce modèle sont alors éliminés...

On peut effectuer plusieurs itérations :

---

**tantque** l'état (valide/non valide) des points évolue **faire**

**pour tout** point apparié P de l'image 1 considéré comme valide (à la première itération, tous les points sont considérés comme valides) **faire**

Calcul de la distance  $d(P)$  entre les coordonnées théoriques (calculées à partir des orientations approchées des clichés 1 et 2 et d'un MNT de la zone) et mesurées de son homologue dans l'image 2

**fin pour**

Calcul de l'écart-type  $\sigma$  et de la moyenne  $\bar{d}$  des  $d(P)$

**pour tout** point apparié P de l'image 1 **faire**

**si**  $|d(P) - \bar{d}| > coef \cdot \sigma$  **alors**

Le point P et son homologue ne sont pas considérés comme valides.

**sinon**

Le point P et son homologue sont considérés comme valides.

**finsi**

**fin pour**

**fin tantque**

Seuls les points considérés comme valides sont conservés.

---

### 5.3.2 Filtrages “photogrammétriques”

- Calcul de la matrice fondamentale et élimination des points trop éloignés de la ligne épipolaire
- Orientation relative, puis compensation
- Elimination au cours d'une compensation par faisceau
- Autres variantes...

### 5.3.3 Filtrage par “chaînage”

Cette méthode s'applique aux cas de multi-recouvrement (plusieurs images en recouvrement sur une même zone) et s'applique notamment si l'on recherche des points de liaison multiple. Il est alors possible d'exploiter cette situation pour détecter des fautes, même si l'appariement s'est déroulé par paire d'images par paire d'images.

La figure 3 en montre un exemple dans un cas où l'on a trois images 1, 2 et 3 en recouvrement :

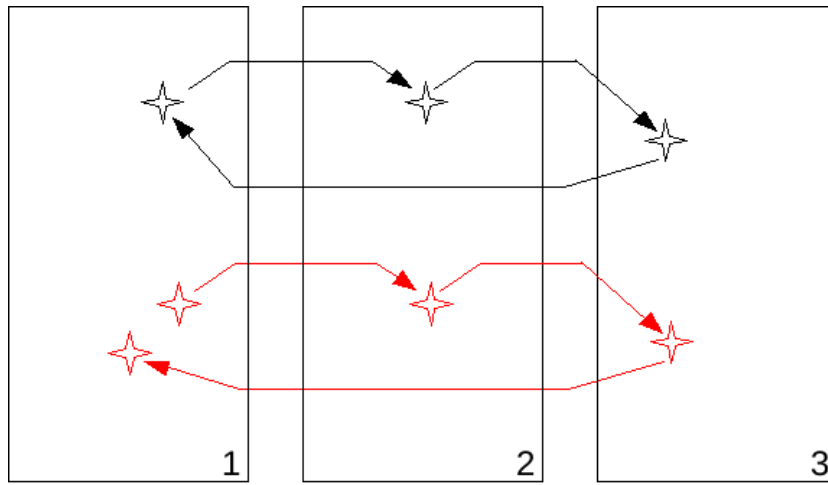
- un point  $P_1$  de l'image 1 a été mis en correspondance avec un point  $P_2$  de l'image 2 lors de la recherche d'homologues entre les images 1 et 2.
- le point  $P_2$  de l'image 2 a été mis en correspondance avec un point  $P_3$  de l'image 3 lors de la recherche d'homologues entre les images 2 et 3.
- le point  $P_3$  de l'image 3 a été mis en correspondance avec un point  $P'_1$  de l'image 1 lors de la recherche d'homologues entre les images 1 et 3.

Les points  $P_1$ ,  $P_2$ ,  $P_3$  et  $P'_1$  sont donc sensés correspondre à un même point “physique”. Or il y a donc une incohérence si  $P_1 \neq P'_1$ , et les différents appariements précédents doivent donc être rejetés.

Ce chaînage des différents appariements obtenus par paire d'images par paire d'images peut donc permettre de détecter des fautes lorsque l'on s'aperçoit que deux points distincts d'une même image sont sensés correspondre à un même point physique...



FIGURE 3 – Détection des fautes par chaînage : les appariements en rouge sont éliminés.



Deuxième partie

**Tests divers sur des données simulées ou des images isolées**

## 6 Évaluation du potentiel de SIFT sur des données simulées

Les tests qui suivent ont eu pour but d'évaluer le potentiel de SIFT pour l'extraction de points de liaison entre deux images, et de régler certains paramètres. Il s'agissait donc de considérer les critères suivants :

- **Répétabilité** : SIFT parvient-il effectivement à détecter des détails homologues suffisamment bien répartis ?
- **Précision** : Quelle est la précision des points de liaison détectés ? SIFT est en effet un détecteur multi-échelle, et les points extraits aux échelles les plus basses pourraient ne pas être très précis. Il s'agissait donc également d'estimer l'influence du paramétrage de SIFT (et plus particulièrement le choix de la première octave et du nombre d'octaves) sur la précision des points homologues obtenus.

Des expériences ont donc été menées sur des paires d'images artificiellement transformées (ce qui permet de maîtriser les paramètres de la transformation entre les deux clichés) afin de confronter SIFT à différents cas de figures. Ces images ont donc subi des transformations géométriques (facteur d'échelle, rotation 2D, similitude 2D, homographie 2D) et radiométriques (réétalement, égalisation...). Les conséquences d'une compression jpeg et de l'ajout de bruit, ainsi qu'un appariement entre canaux différents ont également été testés.

Les tests ont porté à la fois sur un extrait d'image aérienne (résolution sol 50 cm, issue d'une prise de vue départementale IGN) et sur une image terrestre (acquise par le véhicule Stéréopolis).

### 6.1 Effet d'une compression jpeg

On s'intéresse ici à l'incidence d'une compression jpeg sur la qualité (en terme de localisation) et le nombre des points extraits.

Il va donc s'agir simplement de rechercher des points homologues entre des images compressées (à l'aide de GIMP, pour différents niveaux de qualité) de l'image originale (voir les figures 4 et 6). Les résultats obtenus sont résumés par les figures 5 et 7. Les erreurs restent majoritairement inférieures au pixel dans les différents cas (même le cas le moins réaliste pour lequel l'image est assez fortement dégradée), et même au demi pixel pour les deux compressions les plus faibles.

FIGURE 4 – Jeu de données “aérien” : De gauche à droite, extrait de l'image originale et des images compressées de qualité 85%, 50%, 25% et 10%



FIGURE 5 – Jeu de données “aérien” : À gauche : histogramme cumulé de l’erreur de localisation des points homologues extraits, avec en ordonnée la proportion de points SIFT appariés pour lesquels la distance entre les coordonnées théoriques et observées de l’homologue (dans la seconde image) est inférieure à un seuil donné (en abscisse) À droite : nombre de points homologues extraits par octave

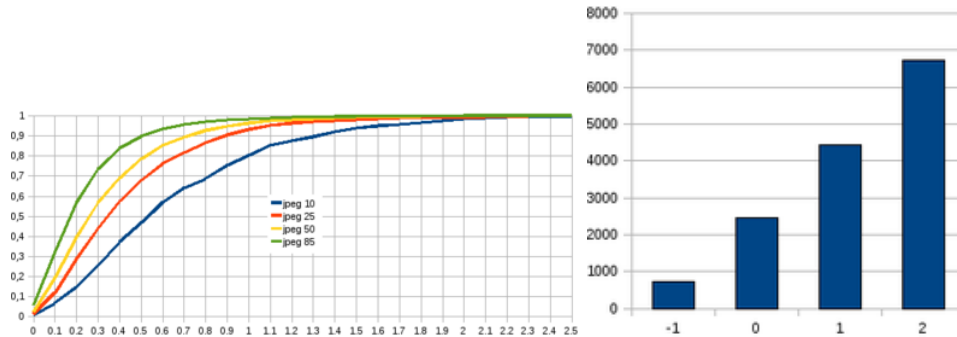


FIGURE 6 – Jeu de données “terrestre” : De gauche à droite, extrait de l’image originale et des images compressées de qualité 85%, 50%, 25% et 10%

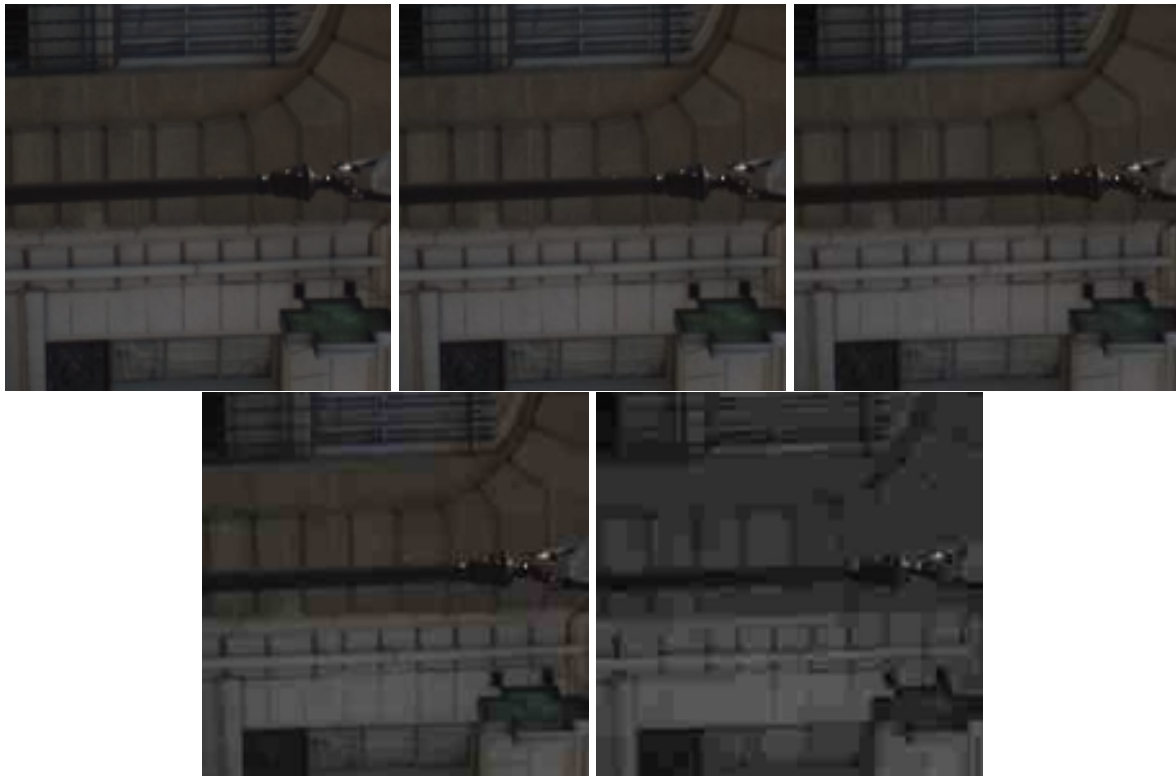
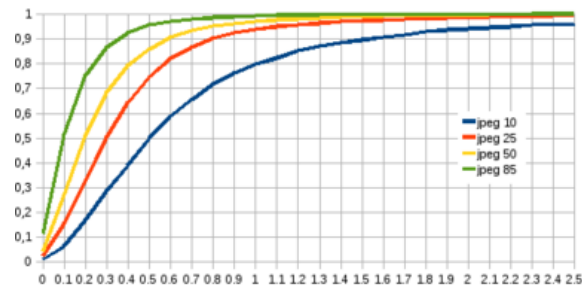


FIGURE 7 – Jeu de données “terrestre” : À gauche : histogramme cumulé de l’erreur de localisation des points homologues extraits, avec en ordonnée la proportion de points SIFT appariés pour lesquels la distance entre les coordonnées théoriques et observées de l’homologue (dans la seconde image) est inférieure à un seuil donné (en abscisse) À droite : nombre de points homologues extraits par octave



## 6.2 Ajout de bruit

Il s’agit de tester la robustesse de SIFT au bruit. Dans ce cas, comme on peut le constater sur les figures 8 et 10, un fort bruit a ici été ajouté (à l’aide du filtre “Bruit > Brouillage RVB” de GIMP) à l’image originale. Les résultats obtenus (présentés figures 9 et 11) sont assez conformes à ce que l’on pouvait espérer étant donné le niveau de bruit ajouté à l’image.

FIGURE 8 – Jeu de données “aérien” : Ajout de bruit



FIGURE 9 – Jeu de données “aérien” : histogramme cumulé de l'erreur de localisation des points homologues extraits (pour 10 puis 1 octaves) et nombre de points homologues par octave

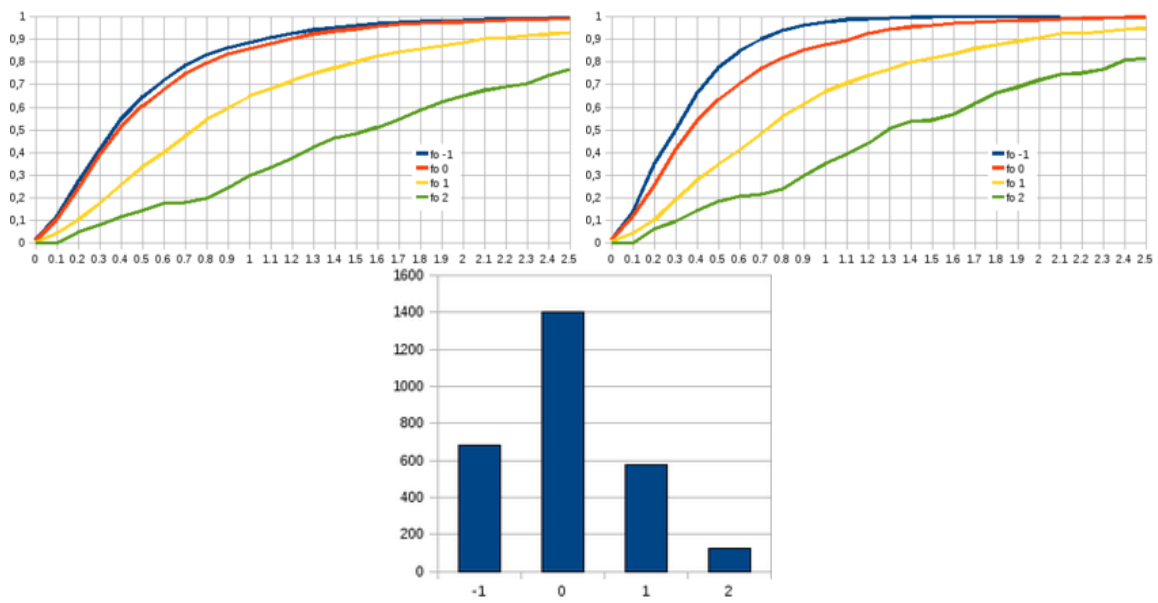
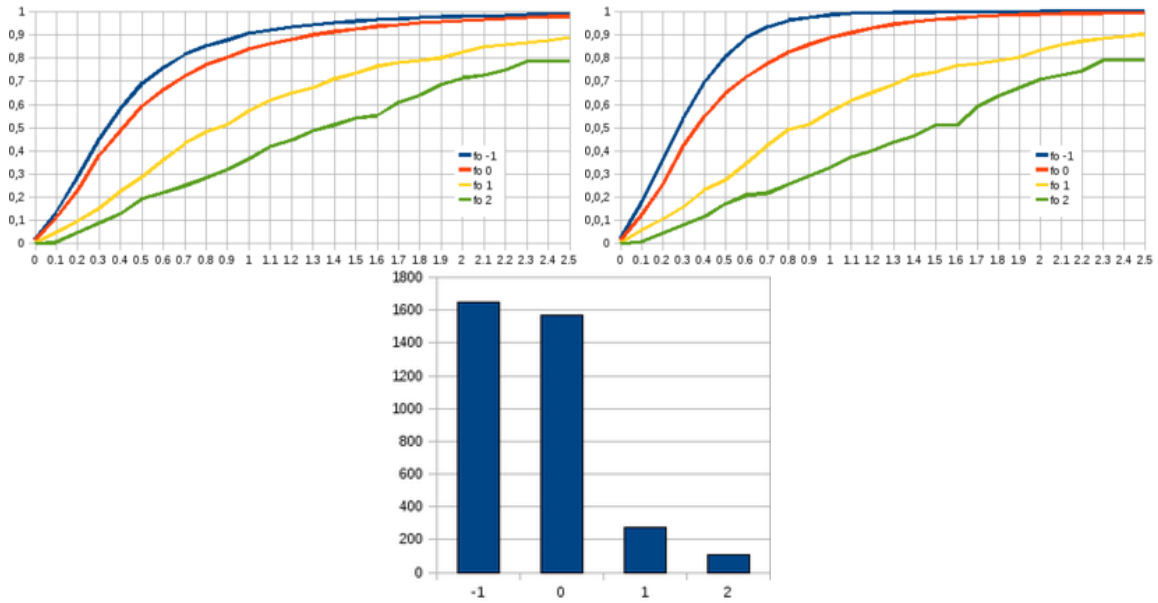


FIGURE 10 – Jeu de données “terrestre” : Ajout de bruit



FIGURE 11 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation des points homologues extraits (pour 10 puis 1 octaves) et nombre de points homologues par octave



### 6.3 Flou

SIFT est un détecteur multi-échelle et est donc sensé être robuste au flou. (Cette propriété sera d’ailleurs exploitée pour le travail présenté dans la partie 12.4.1). Dans les expériences réalisées ici, l’image originale est floutée par application d’un filtre gaussien (voir fig. 12 et 14). Les résultats résumés par les figures 13 et 15 ont été obtenus en utilisant SIFT sur 10 octaves à partir de l’octave -1, afin d’exploiter pleinement son caractère multi-échelle. Ces résultats sont assez conformes à ce à quoi l’on pouvait s’attendre.

FIGURE 12 – Jeu de données “aérien” : De gauche à droite, images floues obtenues par application à l'image originale d'un filtre gaussien pour  $\sigma \in \{0,5; 1; 2; 3\}$

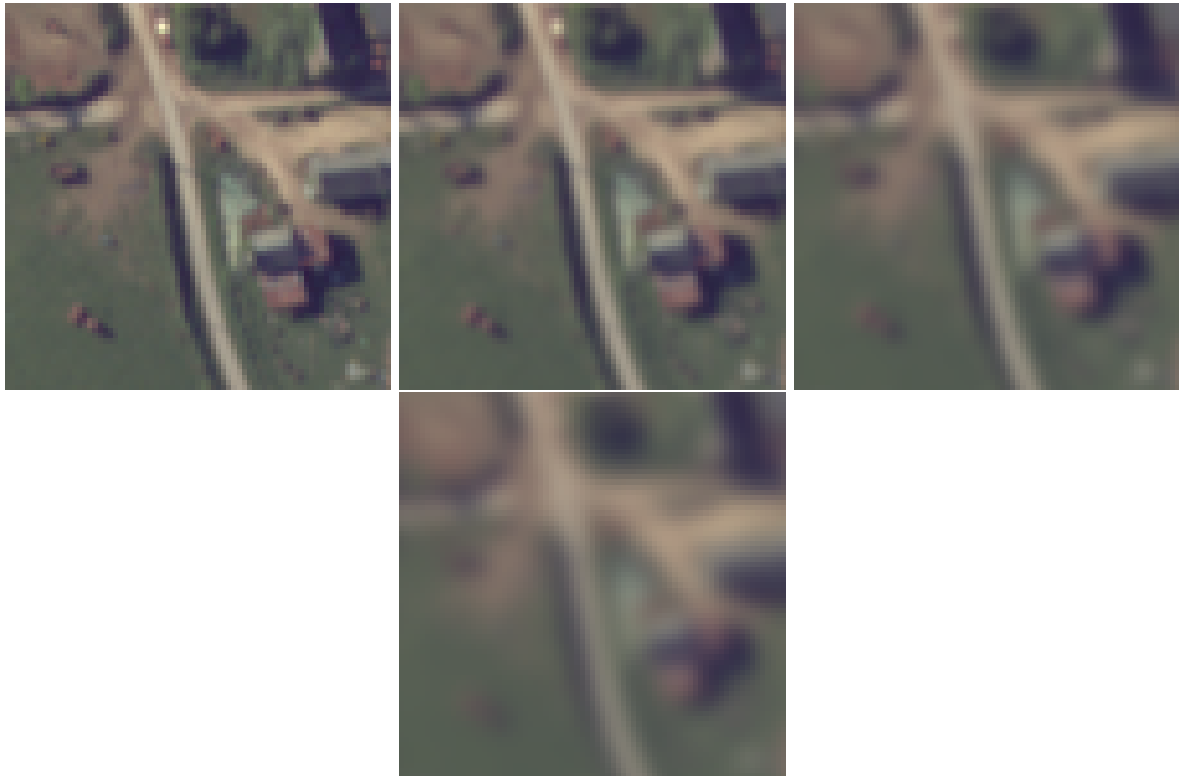


FIGURE 13 – Jeu de données “aérien” : histogramme cumulé de l'erreur de localisation des points homologues extraits (pour 10 octaves à partir de l'octave -1)

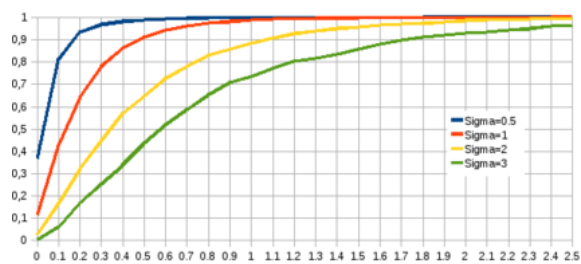




FIGURE 14 – Jeu de données “terrestre” : De gauche à droite, images floues obtenues par application à l’image originale d’un filtre gaussien pour  $\sigma \in \{0,5; 1; 2; 3\}$

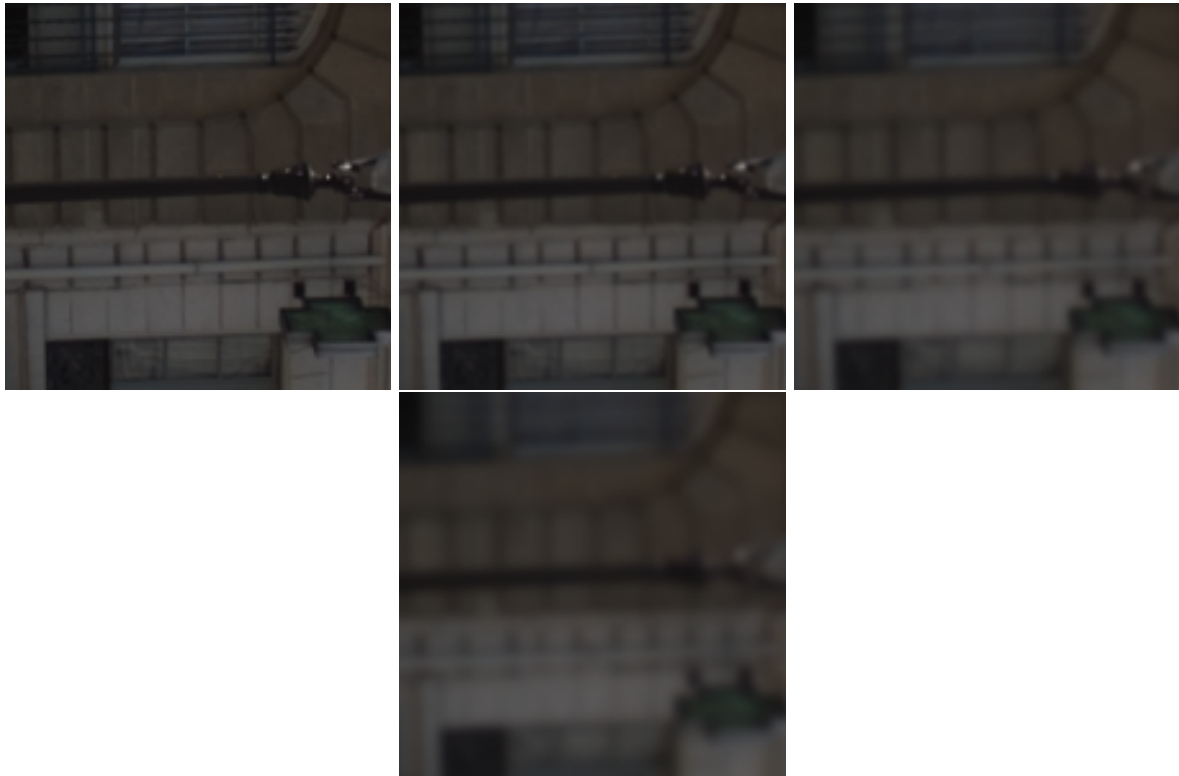
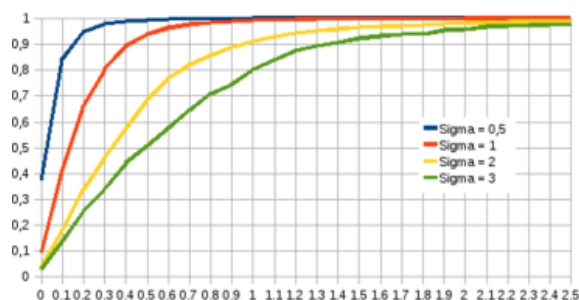


FIGURE 15 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation des points homologues extraits (pour 10 octaves à partir de l’octave -1)



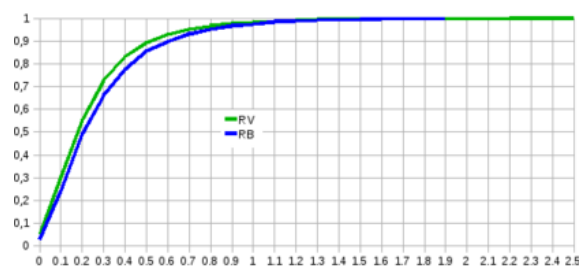
## 6.4 Canaux différents

Il est intéressant de savoir comment se comporte SIFT dans le cas d’appariement d’images représentant une même scène mais pour une radiométrie différente. C’est notamment le cas d’images acquises dans différentes bandes spectrales. Le test suivant porte plus particulièrement sur les canaux rouge, vert et bleu d’une même image (voir figures 16 et 18). Les résultats (figures 17 et 19) montrent que la majorité des points homologues extraits ont une précision inférieure à 0,5 pixel.

FIGURE 16 – Jeu de données “aérien” : De gauche à droite, canaux rouge, vert et bleu



FIGURE 17 – Jeu de données “aérien” : histogramme cumulé de l’erreur de localisation des points homologues extraits entre les canaux rouge et vert (courbe verte) ainsi qu’entre les canaux rouge et bleu (courbe bleue), en utilisant SIFT à partir de l’octave -1 pour 2 octaves

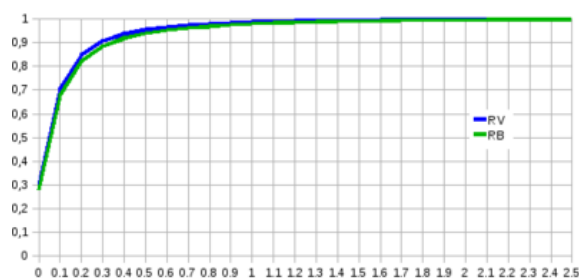


**Remarque :** L’image utilisée pour ces tests ne présentait pas de canal proche infra-rouge. Néanmoins d’autres tests ont été menés sur de telles images. Du fait de son comportement assez différent par rapport aux canaux visibles (notamment dans les zones de végétation), il n’a pas toujours été possible d’extraire des points homologues entre le canal proche-infrarouge et ces derniers (même en ne travaillant pas directement sur les images, mais sur la norme de leur gradient, ou en utilisant les demi-descripteurs). Cela a dépendu des cas...

FIGURE 18 – Jeu de données “terrestre” : De gauche à droite, canaux rouge, vert et bleu



FIGURE 19 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation des points homologues extraits entre les canaux rouge et vert (courbe verte) ainsi qu’entre les canaux rouge et bleu (courbe bleue), en utilisant SIFT à partir de l’octave -1 pour 2 octaves



## 6.5 Radiométrie

Les expériences qui suivent avaient pour but de vérifier la robustesse de SIFT aux variations d’éclairément. L’image originale a subi différents réételements, ainsi qu’une égalisation (comme on peut le voir figures 20 et 22). Les résultats présentés figures 21 et 23 montrent que les points homologues obtenus ont majoritairement une précision inférieure à 0,5 pixel. On constate également que les points extraits entre l’image égalisée et l’image originale sont généralement moins bons, ce qui s’explique par le fait que l’égali- sation a modifié le comportement radiométrique de l’image davantage qu’un simple réételement.

FIGURE 20 – Jeu de données “aérien” : De gauche à droite et de haut en bas : image originale, images obtenues respectivement suite à des réételements entre 0 et 100, entre 0 et 150 et entre 100 et 255, ainsi que suite à une égalisation de la radiométrie

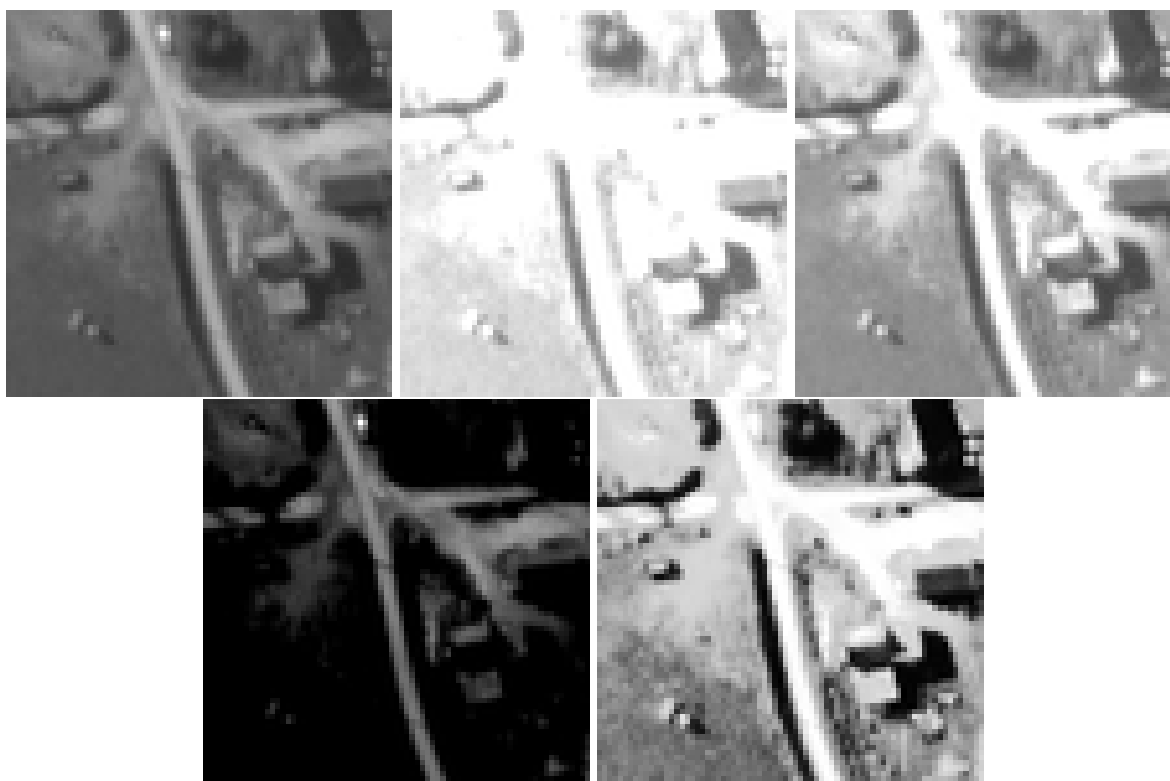


FIGURE 21 – Jeu de données “aérien” : histogramme cumulé de l’erreur de localisation des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves

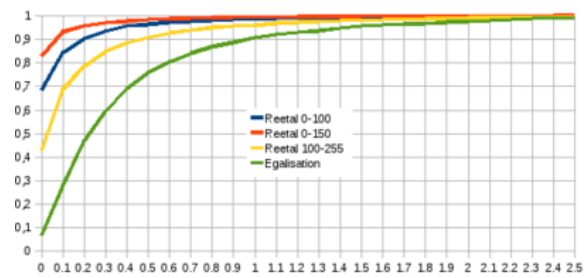


FIGURE 22 – Jeu de données “terrestre” : De gauche à droite et de haut en bas : image originale, images obtenues respectivement suite à des réételements entre 0 et 100, entre 0 et 150 et entre 100 et 255, ainsi que suite à une égalisation de la radiométrie

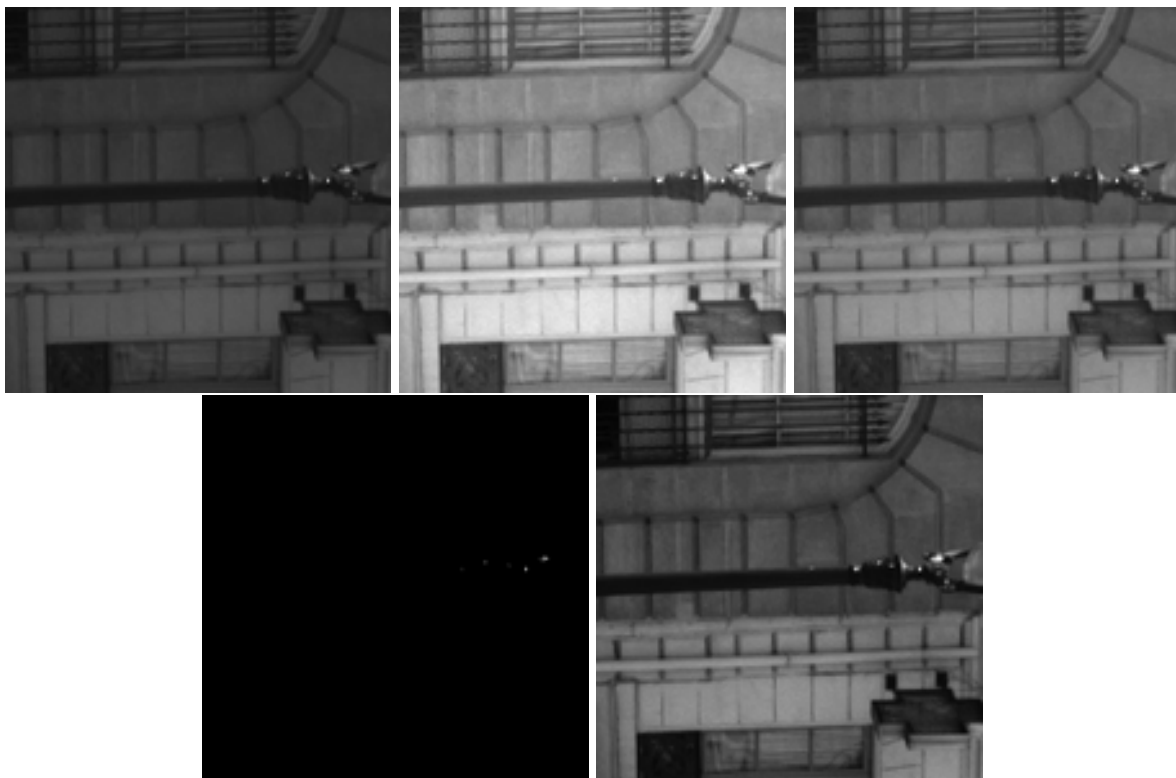
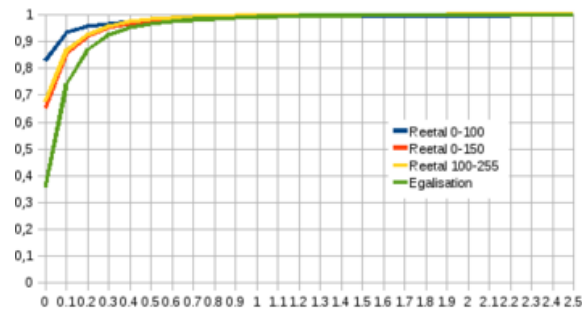


FIGURE 23 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves



## 6.6 Translation

Cette expérience consistant à appliquer une translation de 0,5 pixel est plus liée à un test sur les effets d’un échantillonnage légèrement différent (et donc lié à la qualité de l’image) qu’aux expérimentations qui suivront sur la robustesse de SIFT à des transformations géométriques.

On constate que les points homologues extraits ont une assez bonne précision (erreur  $< 0,2$  pixel pour presque tous) pour les 2 premières octaves, et que les résultats obtenus pour les octaves 1 et 2 restent conformes à ce que l’on pouvait espérer.

FIGURE 24 – Jeu de données “aérien” : histogramme cumulé de l’erreur de localisation des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves (en haut à gauche) et à partir des octaves -1,0,1 et 2 pour 1 octave (en haut, à droite), et nombre de points homologues extraits pour chaque octave (en bas)

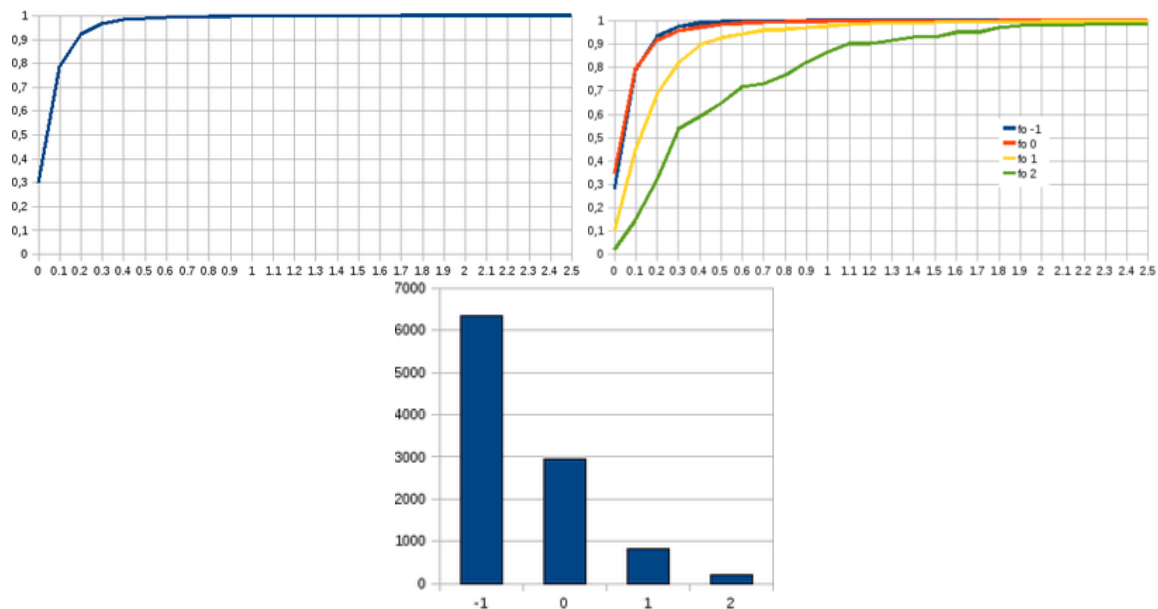
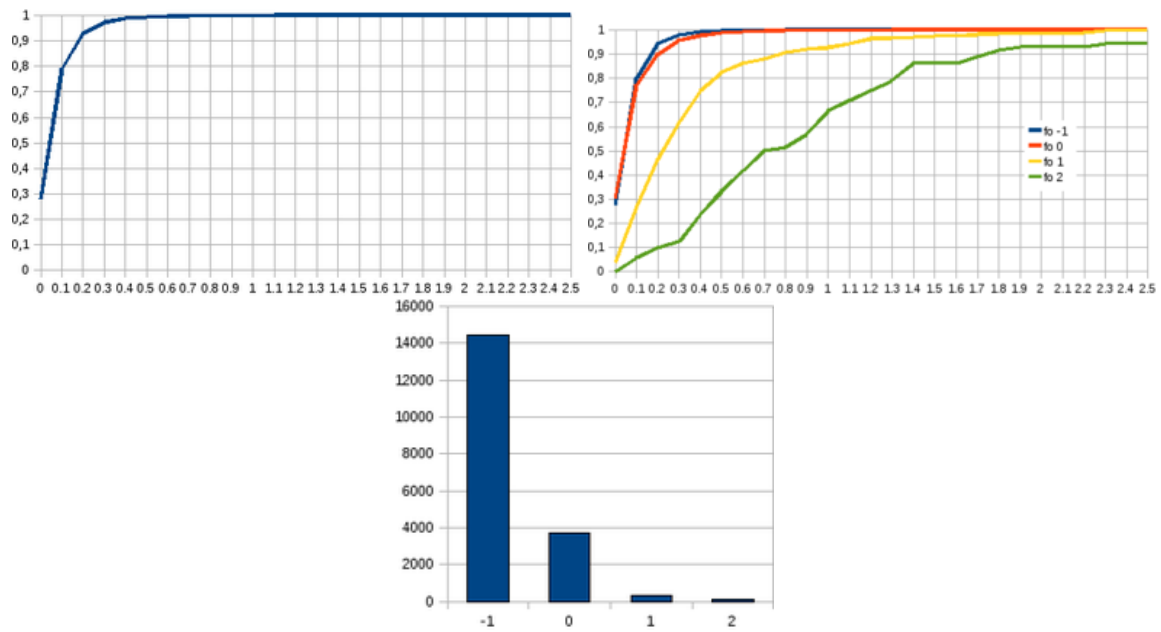


FIGURE 25 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves (en haut à gauche) et à partir des octaves -1,0,1 et 2 pour 1 octave (en haut à droite), et nombre de points homologues extraits pour chaque octave (en bas)



## 6.7 Variation d'échelle

Il s'agissait cette fois de s'intéresser à la précision des points détectés face à une variation d'échelle entre deux images. Différentes simulations consistant à simplement appliquer un facteur d'échelle à l'image ont donc été effectuées. Les résultats sont présentés dans les figures 26 et 27. Contrairement aux tests précédents, on s'intéresse ici aux erreurs de localisation dans les 2 images (originale et transformée). En effet, si dans les cas précédents, les deux images restaient à la même échelle, ce n'est plus le cas ici où les erreurs de localisation seront logiquement plus fortes dans l'image la plus “dilatée”.

FIGURE 26 – Jeu de données “aérien” : Sur chaque ligne, histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir des octaves -1,0 et 1 pour 10 octaves

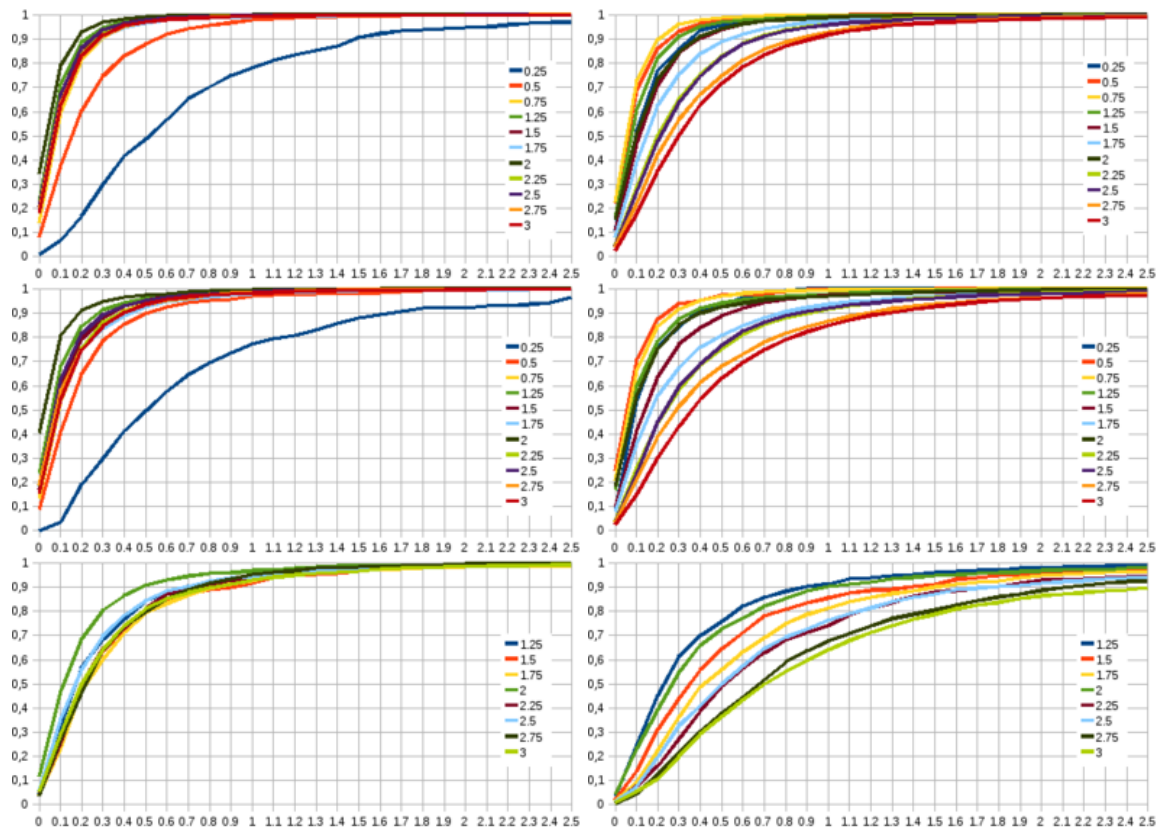
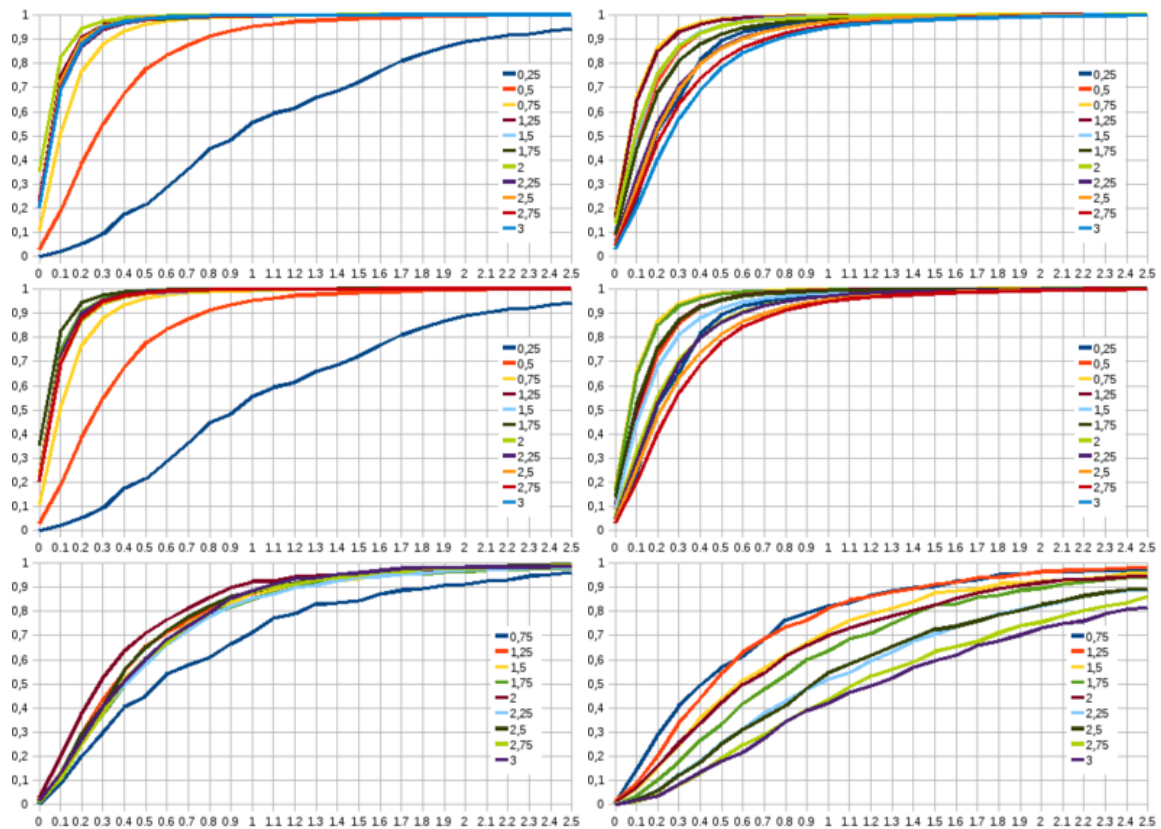


FIGURE 27 – Jeu de données “terrestre” : Sur chaque ligne, histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir des octaves -1,0 et 1 pour 10 octaves





## 6.8 Rotation (2D)

On s'intéresse ici à la précision des points détectés face à une rotation 2D entre deux images. Différentes simulations consistant à simplement appliquer une rotation 2D à l'image de référence ont donc été effectuées. Les résultats sont présentés dans les figures 28 et 29.

FIGURE 28 – Jeu de données “aérien” : histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir des octaves -1 et 0 pour 10 octaves

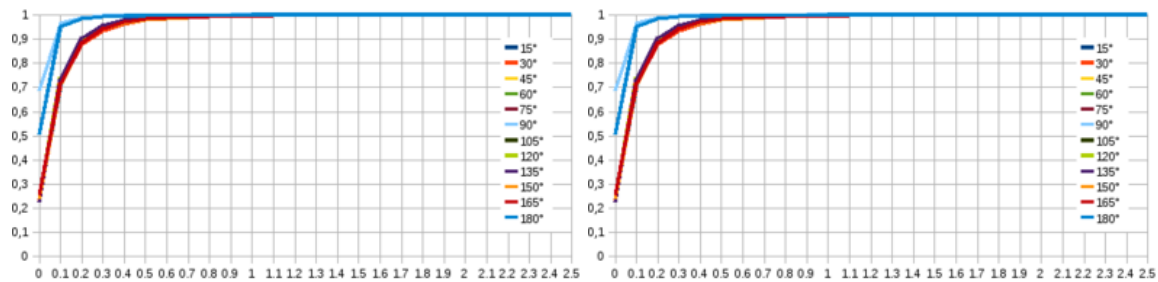
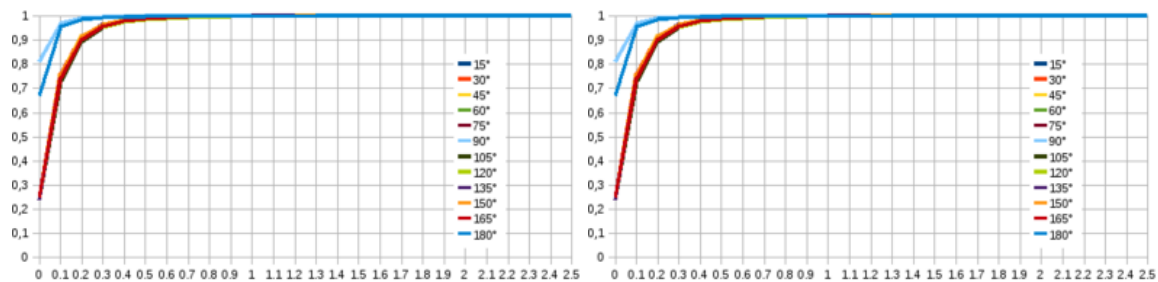


FIGURE 29 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir des octaves -1 et 0 pour 10 octaves



## 6.9 Similitude (2D)

Il s’agit de s’intéresser à la précision des points détectés face à une similitude 2D entre deux images. Différentes simulations ont été appliquées à l’image de référence. Les résultats sont présentés dans les figures ?? et 31.

FIGURE 30 – Jeu de données “aérien” : histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves

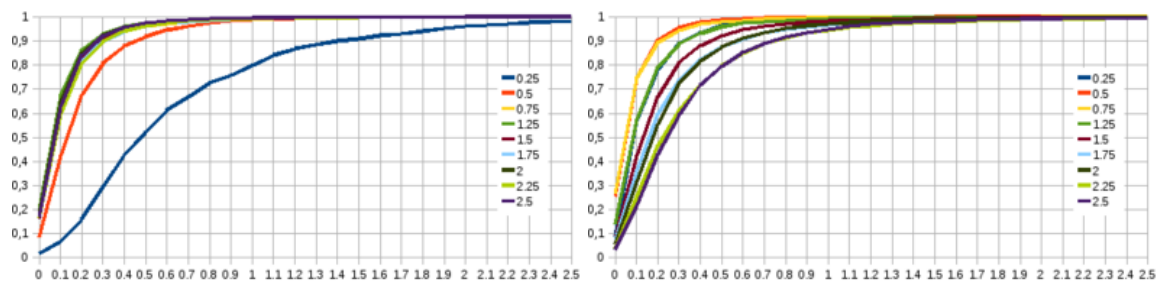
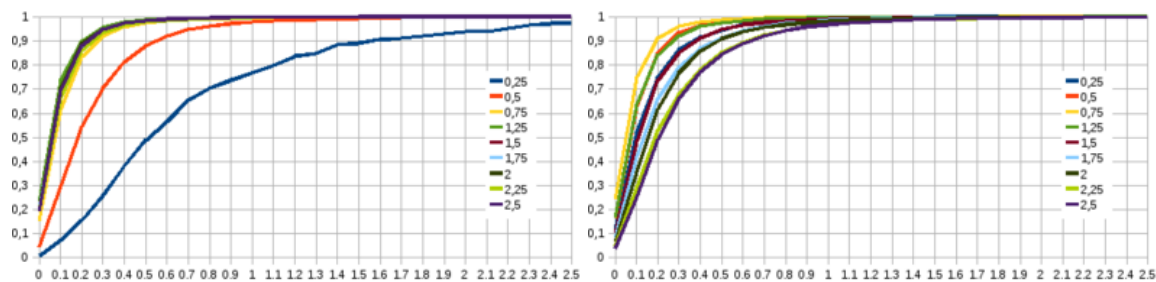


FIGURE 31 – Jeu de données “terrestre” : histogramme cumulé de l’erreur de localisation dans l’image originale (à gauche) et sa transformée (à droite) des points homologues extraits en utilisant SIFT à partir de l’octave -1 pour 10 octaves



## 6.10 Affinité

On considère que toutes les déformations peuvent localement être modélisées par des affinités. Des tests de mise en correspondance ont été menés en simulant différentes affinités. En général, la majorité des points appariés ont une précision inférieure au pixel. Néanmoins, en fonction des déformations causées, la distribution des points peut ne pas être très régulière, avec de beaucoup de points détectés (et d’excellente précision) là où les déformations sont faibles, et moins de points (en outre moins précis) là où elles sont plus fortes (et plus particulièrement là où l’image est “dilatée”).

## 6.11 Paramètres “première octave” et “nombre d’octaves” lors de l’extraction des points DoG

Cette question revêt une certaine importance. En effet, les points extraits aux octaves les plus hautes risquent d’être moins précis que ceux extraits aux octaves les plus basses. Ceci peut avoir deux conséquences sur l’utilisation pratique de SIFT :

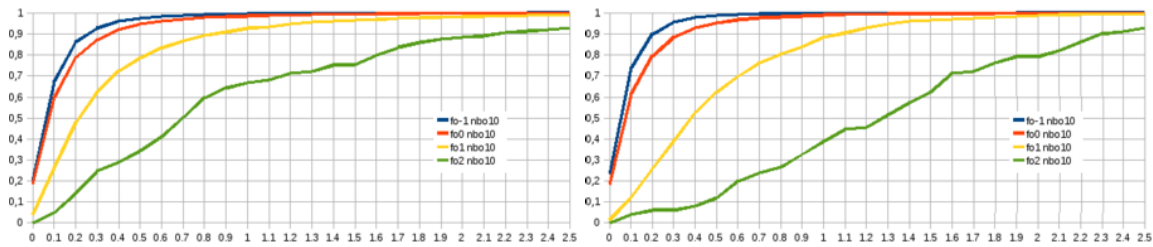
- Plus on s’élève dans la pyramide d’images et moins l’on détecte de points. Par conséquent, ces points un peu flous risquent d’être noyés dans la masse. Néanmoins, il pourrait être plus prudent de fixer une octave maximale, notamment dans le cas d’une extraction de points de liaison pour un calcul d’aéro-triangulation d’un chantier aérien “classique”, pour lequel les variations d’échelle entre clichés ne sont pas très importantes.
- Comme on le verra par la suite, les temps d’extraction de points de liaison avec SIFT peuvent être assez importants en particulier dans le cas des chantiers de prise de vue aériennes départementales. Par défaut, le calcul commence à partir de l’octave -1, c’est-à-dire en suréchantillonnant l’image d’un facteur 2. Ceci permet d’améliorer la précision des points fournis (localisation subpixelaire). Or, le fait de commencer l’extraction des points à l’octave 0 au lieu de l’octave -1 réduit fortement les temps de calcul (quasiment de moitié), et semble fournir des points de qualité suffisante.

Les tests qui suivent ont été réalisés en appliquant à l’image de référence une similitude de rotation  $30^\circ$  et de facteur d’échelle 1,25. Les résultats qui suivent sont donnés dans l’image transformée, c’est-à-dire celle où le facteur d’échelle (et donc les erreurs de localisation) sont les plus importantes

### 6.11.1 Paramètre “première octave”

La figure 33 montre que les points extraits pour les octaves -1 et 0 sont bons : la grande majorité d’entre eux a une erreur de localisation inférieure à 0,3 pixel. Les points extraits pour l’octave 1 ont une erreur très majoritairement inférieure à 0,5 pixel. La précision commence à se dégrader à partir de l’octave 2.

FIGURE 32 – Histogramme cumulé des erreurs de localisation des points dans l'image transformée en fonction du paramètre première octave



### 6.11.2 Paramètre “nombre d’octaves”

FIGURE 33 – Histogramme cumulé de localisation des points dans l'image transformée en fonction du paramètre nombre d’octaves

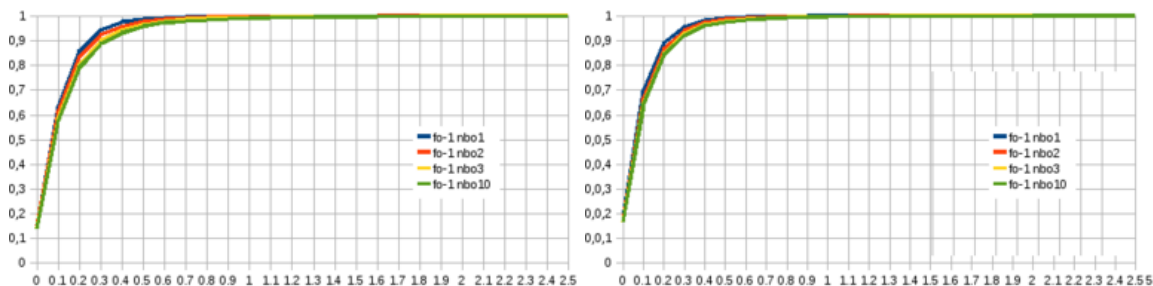
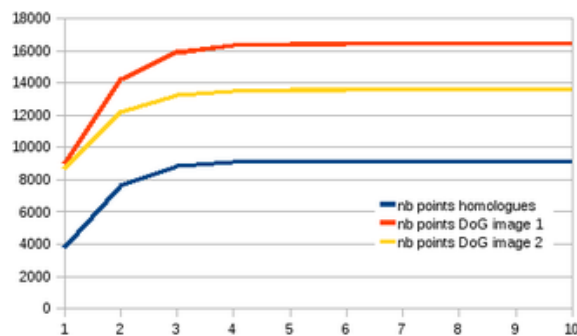


FIGURE 34 – Nombre de points obtenus (à la fois extraction directe des points de Lowe des images, et points homologues) en fonction du nombre d’octaves traitées (première octave = -1). L'écrasante majorité des points se trouve dans les 2-3 premières octaves.



## 7 Critère d'appariement basé directement sur la distance au plus proche voisin, ou sur le ratio des distances aux deux plus proches voisins

La méthode d'appariement proposée par [Low04] et décrite dans la partie 1.3 consiste à rechercher pour chaque point  $P$  de la première image ses deux plus proches voisins  $P_1$  et  $P_2$  parmi les points de Lowe de la seconde image au sens de la distance euclidienne entre leurs descripteurs SIFT.

Si  $\frac{d_{SIFT}(P, P_1)}{d_{SIFT}(P, P_2)} < \text{seuil\_ratio\_appariement}$  (en pratique,  $\text{seuil\_ratio\_appariement} = 0,8$ ), alors l'appariement est considéré comme non ambigu et  $P_1$  est considéré comme l'homologue de  $P$ . Dans le cas contraire, on considère que  $P$  ne possède pas d'homologue.

Ce critère permet d'éliminer d'emblée les appariements ambigus, mais empêche également les appariements multiples, qui pourraient être tolérés dans certains cas où l'ambiguïté pourrait ensuite être levée par un critère géométrique. Il est intéressant de savoir quels résultats peuvent être obtenus en utilisant à la place de ce critère un simple critère sur la distance  $d_{SIFT}(P, P_1)$  :

si  $d_{SIFT}(P, P_1) < \text{seuil\_distance\_appariement}$ , alors  $P_1$  est considéré comme l'homologue de  $P$ .

Deux images issues d'une même bande d'un chantier aérien ont fait l'objet d'un appariement suivant ces deux critères. Les résultats bruts (i.e. sans aucun filtrage des fautes) sont visibles sur les illustrations de la figure 35. On peut constater que davantage d'appariements fournis en utilisant le simple critère sur la distance au plus proche voisin sont erronés (même si ces fautes peuvent ici être ensuite détectées et éliminées assez facilement). La figure 37 quant à elle montre localement la différence entre ces deux appariements dans le cas d'une liaison interbande sur le même chantier aérien : les cercles bleus entourent des points de liaison extraits en utilisant simplement le critère "distance" mais pas lorsque l'on utilise le critère "ratio des distances aux deux plus proches voisins". Or ces points sont de mauvais points de liaison : ils correspondent à un même type de détail sur l'image - une limite d'ombre - mais pas sur le terrain puisque les ombres ne sont plus les mêmes entre ces images acquises à des heures différentes... (On verra ce cas de figure et l'avantage qu'apporte alors SIFT dans les parties 12.2.2 et 12.2.3.)

Mieux vaudra donc utiliser le critère basé sur le ratio des distances aux deux plus proches voisins, afin de garantir la robustesse de l'appariement.

FIGURE 35 – Appariement selon le critère "ratio des distances aux deux plus proches voisins" (en haut) et selon le critère "distance maximale" (en bas) entre deux clichés aériens d'une même bande

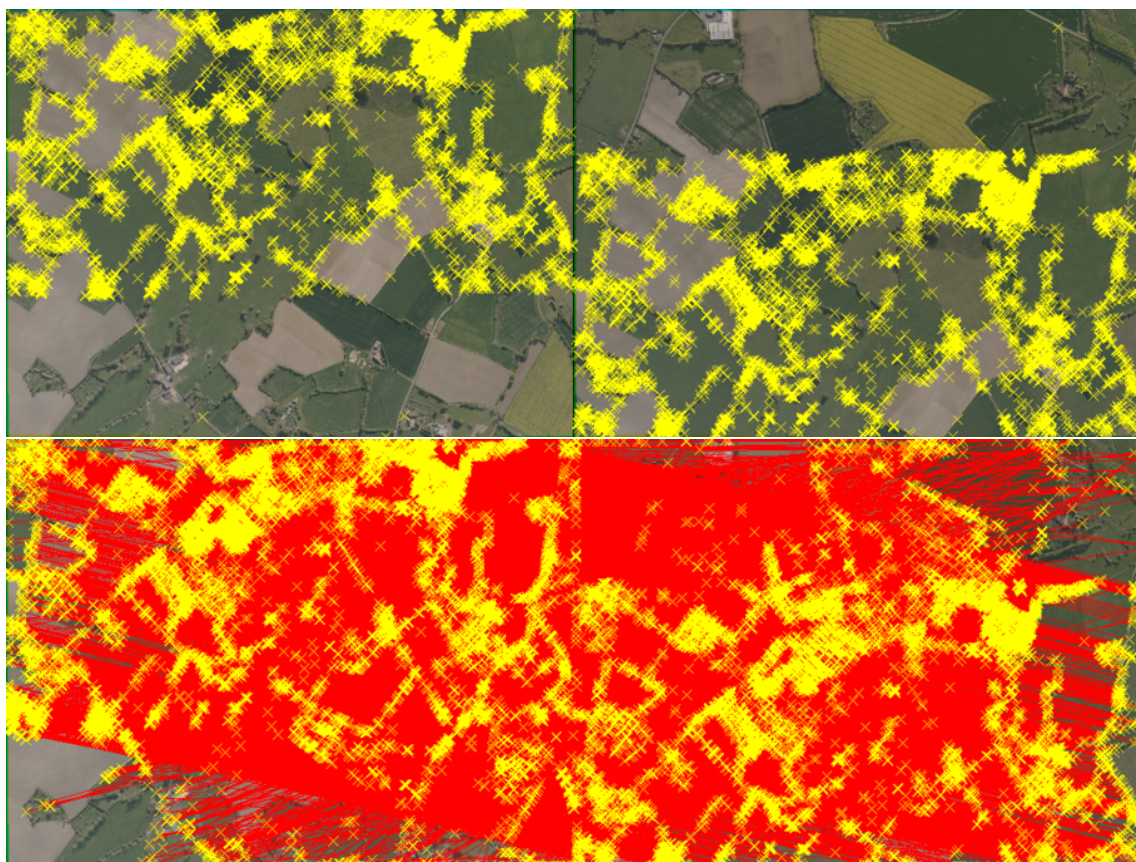


FIGURE 36 – Parmi les points mis en correspondance entre les images aériennes de la figure 35 (par simple recherche du plus proche voisin) en utilisant ou non le critère “ratio” , nombre d’appariements en fonction de la distance euclidienne entre leurs descripteurs.

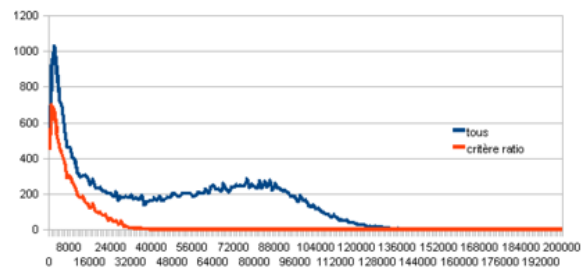


FIGURE 37 – Dans le cas d’une liaison interbande, différence entre les résultats obtenus suivant les 2 critères : les cercles bleus entourent des points de liaison extraits en utilisant simplement le critère “distance” mais pas lorsque l’on utilise le critère “ratio”



FIGURE 38 – Parmi les points mis en correspondance entre deux images aériennes “simulées”(sur le même jeu de données que dans les parties précédentes) (par simple recherche du plus proche voisin) en utilisant ou non le critère “ratio” , nombre de points en fonction de la distance euclidienne entre leurs descripteurs. À gauche, vue globale. À droite, zoom sur la partie encadrée en vert sur la figure de gauche.

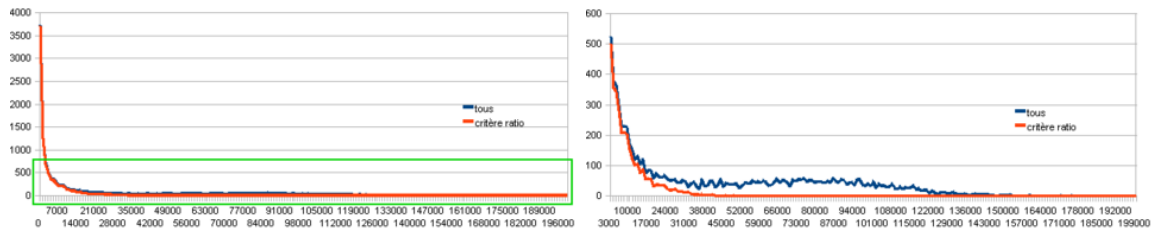


FIGURE 39 – Parmi les points appariés entre les deux images aériennes en interbande de la figure ?? (par simple recherche du plus proche voisin) en utilisant ou non le critère “ratio” , nombre de points en fonction de la distance euclidienne entre leurs descripteurs.

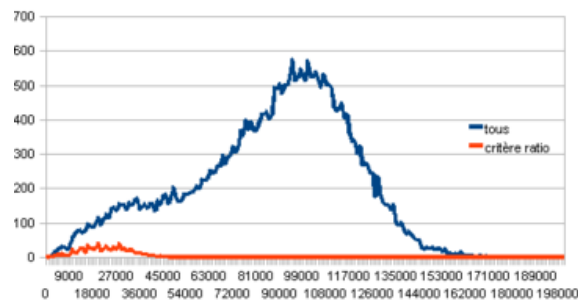


FIGURE 40 – Parmi les points mis en correspondance entre deux images terrestres “simulées” (même jeu de données simulées que dans les parties précédentes) par simple recherche du plus proche voisin en utilisant ou non le critère “ratio”, nombre d'appariements en fonction de la distance euclidienne entre leurs descripteurs. À gauche, vue globale. À droite, zoom sur la partie encadrée en vert sur la figure de gauche.

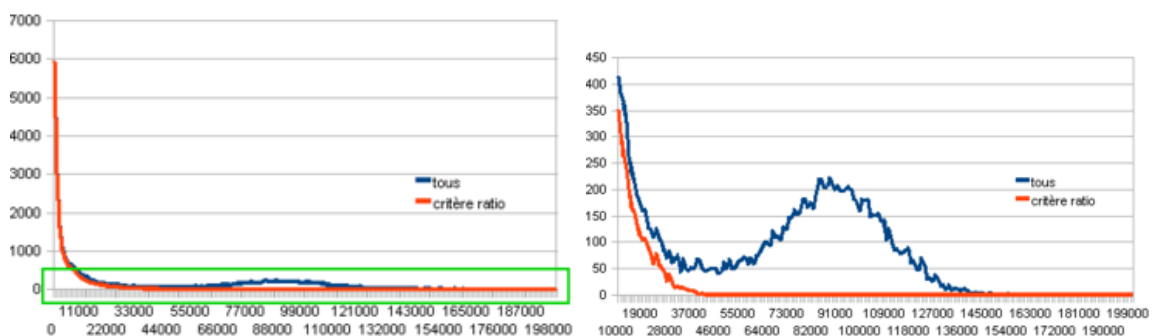
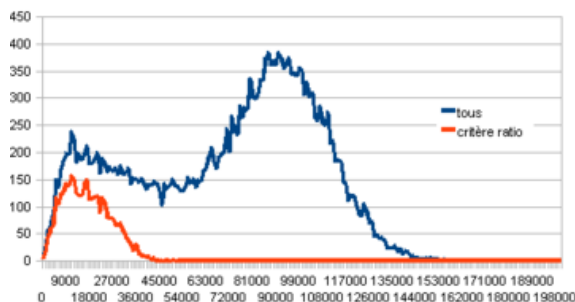


FIGURE 41 – Parmi les points mis en correspondance (par simple recherche du plus proche voisin) entre images terrestres en utilisant ou non le critère “ratio”, nombre d’appariements en fonction de la distance euclidienne entre leurs descripteurs.



## 8 Répartition et nombre de points

### 8.1 Problème du nombre de points extraits en fonction du “contraste”

Ainsi que l’illustre la figure 42, un nombre moindre de points d’intérêt est généralement extrait des zones les plus sombres des images lors de l’étape de détection de points d’intérêt (image par image). Ceci s’explique par le fait que l’image présente bien souvent un contraste moindre et une texture également moindre dans ces zones. La plupart des points initialement détectés comme extrema de la fonction *DoG* dans ces zones ne sont par conséquent pas conservés à l’issue de l’application du critère de contraste. Il est donc possible d’y conserver plus de points en jouant sur le seuil de contraste (nommé *threshold* dans la partie 4.3), mais cela a également pour effet de conserver davantage de points dans les autres zones où l’on en avait déjà suffisamment.

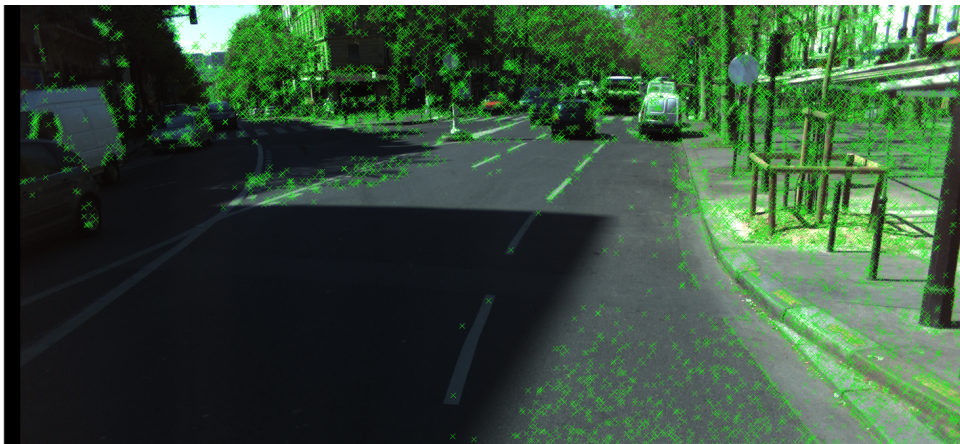
Dans la littérature, ce problème a été décrit de manière plus générale (comment récupérer plus de points dans les zones peu texturées) par [LMN09] qui propose de faire du paramètre *threshold* un paramètre auto-adaptatif destiné à varier au sein de l’image et présente une méthode permettant d’en déterminer localement la valeur de manière automatique en fonction de l’image (et plus précisément de sa texture).

Pour ce qui nous concerne, ce problème ne se pose pas vraiment dans le cas de la plupart des chantiers aériens : les zones à l’ombre y restent limitées et les régions peu texturées (champs, ...) sont généralement séparées par des zones plus texturées dans lesquelles on détecte des points. Or SIFT fournit déjà beaucoup de points dans ces zones, impliquant un filtrage spatial : on ne conserve pas tous les points, mais on se donne une grille et on ne conserve qu’un point homologue par case de cette grille, ceci afin d’en obtenir une répartition à peu près régulière. Il n’est donc pas forcément nécessaire dans ce cas d’essayer d’extraire davantage de points d’intérêts.

Dans le cas de chantiers terrestres, le problème de zones importantes dans l’ombre (tout un côté de la rue...) peut se poser. Par ailleurs, les façades de bâtiments peuvent comporter de grandes zones très peu texturées. La solution retenue dans les expériences menées ici a donc simplement consisté à fixer un seuil de contraste assez bas, de manière à extraire beaucoup de points.

En outre, les séries d’images que nous avons à traiter ont généralement été acquises pour un réglage donné dans des conditions assez semblables (cf chantiers de prises de vue aérienne, ou chantiers terrestres Stéréopolis). Un même réglage convient donc en général à l’ensemble du chantier.

FIGURE 42 – Moins de points dans les zones sombres (moins contrastées)



## 8.2 Autres problèmes liés au rééchantillonnage...

Les algorithmes de *sift++* attendent des images dont les valeurs sont comprises entre 0 et 1. C'est notamment par rapport à ces valeurs que le seuil de contraste *threshold* a un sens. Il va donc falloir rééchantillonner la radiométrie de nos images.

Dans le cas de chantiers aériens ou de chantiers terrestres Stéréopolis, les images à traiter peuvent être tantôt fournies en 8 bits, tantôt en 16 bits. Dans le cas d'images 8 bits, ces clichés ont déjà fait l'objet d'un rééchantillonnage "intelligent", et on peut donc considérer que la dynamique de l'image s'étale bien sur les 256 niveaux possibles. Par contre, dans le cas d'images 16 bits, la dynamique de l'image ne s'étale pas sur l'ensemble des niveaux possibles. Il faut donc dans ce cas déterminer la borne supérieure de l'intervalle à partir duquel on va rééchantillonner les valeurs de l'image. Deux solutions sont possibles :

- calcul automatique : la valeur maximale de l'image est considérée comme la borne supérieure de cet intervalle. Cette méthode peut néanmoins poser problème dans deux cas de figures (rencontrés au cours des différents tests menés). Il arrive que quelques pixels de l'image aient des valeurs très fortes (ou du moins bien plus fortes que les autres). Aussi, à l'issue du rééchantillonnage par rapport à la valeur maximale, la majorité des pixels voient leur valeur proche de 0. Dans ces conditions, très peu de points sont détectés. (Il serait toutefois possible de traiter ce cas de figure en excluant du calcul de la valeur maximale une certaine proportion des pixels ayant les valeurs les plus fortes.) A l'inverse, certaines images sont parfois très sombres (océan, forêt, ...). La valeur maximale est alors très faible, le rééchantillonnage fait alors exploser le bruit, et un nombre considérable de points d'intérêt est détecté.
- paramétrage par l'utilisateur : un paramètre permettant à l'utilisateur de définir la borne supérieure de l'intervalle de cet intervalle a été créé. Il s'agit du paramètre *maxim*. Comme les séries d'images à traiter ont généralement été acquises pour un réglage donné dans des conditions assez semblables (cf chantiers de prises de vue aérienne, ou chantiers terrestres Stéréopolis), il est possible de fixer ce paramètre à une même valeur sur l'ensemble du chantier.

Par ailleurs, le simple fait de rééchantillonner la radiométrie permet souvent d'améliorer la situation décrite en 8.1.

## 9 Utilisation d'un prédicteur ?

Au vu des différents tests menés, il est apparu que, dans la majorité des cas, SIFT est une méthode suffisamment robuste pour se passer de prédicteur lors de l'appariement. Néanmoins, l'utilisation d'un prédicteur peut malgré tout parfois s'avérer utile.

### 9.1 Quel prédicteur ?

L'utilisation d'un prédicteur donnant pour chaque point SIFT de l'image 1 une zone de recherche très restreinte dans l'image 2 pourrait s'avérer dangereux en favorisant l'apparition de faux appariements. En effet, la force de SIFT vient en grande partie du critère d'appariement basé sur le ratio des distances aux deux plus proches voisins dans l'espace SIFT. Or, un prédicteur trop fin (i.e. une recherche trop contrainte) limiterait



la recherche à moins de points parmi lesquels ces cas d'ambiguïté se produiraient moins souvent. On pourrait alors se trouver dans des cas, où un point donné de l'image 1 n'aurait en fait pas d'homologue dans la l'image 2, mais serait tout de même mis en correspondance avec un point "beaucoup" plus proche de lui dans l'espace SIFT que les quelques autres points présents dans la zone restreinte définie par le prédicteur. En revanche, si la recherche s'était faite sur une zone plus importante, on aurait eu davantage de chances de trouver un autre point assez proche dans l'espace SIFT ; l'appariement aurait alors été considéré comme ambigu et rejeté... Dans la suite, deux méthodes de prise en compte d'un prédicteur ont été testés.

### 9.1.1 Prédicteur "global" grossier : zone en recouvrement

Ainsi, pour profiter de ce qui fait la robustesse de SIFT, il vaudrait donc mieux utiliser des prédicteurs restant assez grossiers. Dans les expériences menées par la suite, et plus particulièrement dans le cas des chantiers aériens, le prédicteur utilisé consistait simplement à restreindre le nombre de points mis en jeu dans la procédure d'appariement en exploitant la connaissance approximative pour chaque paire de clichés 1 et 2 de la zone  $R_{2dans1}$  du cliché 1 en recouvrement avec le cliché 2, et de la zone  $R_{1dans2}$  du cliché 2 est en recouvrement avec le cliché 1 : on cherche alors uniquement à apparier les points du cliché 1 appartenant à  $R_{2dans1}$  aux points du clichés 2 appartenant à  $R_{1dans2}$ . On réduit ainsi le nombre de requêtes et donc les temps de calcul (en particulier pour des paires d'images présentant un faible recouvrement), et on évite d'obtenir des appariements erronés mettant en jeu des points situés en dehors de ces zones.

Un tel prédicteur est donc un prédicteur "global" portant sur les ensembles de points à apparier et non un prédicteur "local" définissant une fenêtre de recherche d'homologue propre à chaque point à apparier et centrée sur la position approchée de son homologue calculée d'après l'orientation approchée des clichés. De cette manière, on utilise un certain prédicteur tout en continuant à bénéficier de ce qui fait la robustesse de SIFT.

### 9.1.2 Appariement multiple puis prédicteur

Dans le cas où l'on dispose d'un prédicteur capable de fournir pour chaque point  $P$  de l'image 1 la position approchée  $pred_{1 \rightarrow 2}(P)$  de son homologue dans l'image 2, une autre façon d'utiliser un tel prédicteur consiste à procéder de la manière suivante :

Pour chaque point  $P_1$  de l'image 1 :

1. Recherche parmi les points extraits de l'image 2 de l'ensemble  $S_{1 \rightarrow 2}(P_1)$  de ses *nbvois* plus proches voisins dans l'espace des descripteurs.
2. Parmi ces points, recherche de  $P_2$  plus proche voisin (dans l'espace image) de la position approchée  $pred_{1 \rightarrow 2}(P_1)$  de l'homologue de  $P_1$  dans l'image 2.
3. Si  $P_2$  est trop éloigné de la position approchée de l'homologue de  $P_1$  dans l'image 2, alors l'appariement est rejeté : pas d'homologue pour  $P_1$  dans l'image 2. Dans le cas contraire,  $P_2$  est l'homologue de  $P_1$ .

On a donc l'algorithme suivant :

---

```

pour tout point  $P_1$  de l'image 1 faire
  Recherche parmi les points extraits de l'image 2 de l'ensemble  $S_{1 \rightarrow 2}(P_1)$  de ses nbvois plus proches voisins
  dans l'espace des descripteurs
   $P_2 \leftarrow \operatorname{argmin}_{p \in S_{1 \rightarrow 2}(P_1)} d(p, pred_{1 \rightarrow 2}(P_1))$ 
  si  $d(P_2, pred_{1 \rightarrow 2}(P_1)) > \text{seuil}$  alors
    l'appariement est rejeté : pas d'homologue pour  $P_1$  dans l'image 2.
  sinon
     $P_2$  est l'homologue de  $P_1$ 
  finsi
fin pour

```

---

## 9.2 Intérêt d'un tel prédicteur

L'intérêt d'un tel prédicteur est à la fois d'éviter certaines erreurs grossières et de permettre de réduire les temps de calcul (puisque l'on a alors de fait moins de points à mettre en correspondance).

Par ailleurs, dans le cas de très grandes images (images satellite) pour lesquelles on a quasiment trop de points pour tous les charger en mémoire, un prédicteur peut également s'avérer utile pour diviser l'appariement en plusieurs sous problèmes. (Dans ce type de cas de figure, on peut d'ailleurs effectuer une première détection de

points homologues à basse résolution, puis d'en déduire quelle zone de la seconde image correspond à une région de la première image.)

L'utilisation d'un prédicteur peut aussi permettre de détecter un peu plus de points homologues. Ceci peut s'avérer utile dans le cas du recalage d'images sur une couche de référence (comme dans la partie 14.1) ou dans le cas du processus d'aérottriangulation afin d'espérer obtenir plus de points multiples.

## 10 Demi-descripteur

Ainsi qu'on l'a vu précédemment, le descripteur SIFT associé à un point est la concaténation d'histogrammes décrivant la distribution de l'orientation du gradient de l'image dans un certain voisinage autour de ce point. Or, dans certains cas, il pourrait être plus intéressant de ne s'intéresser qu'à la direction du gradient, et non à son orientation. Autrement dit, il s'agirait de prendre en compte la valeur de son orientation modulo  $\pi$ , et non modulo  $2\pi$ .

Cela pourrait présenter un intérêt dans le cas où l'on souhaite extraire des points de liaison entre images de modalité différente (comme par exemple une image acquise dans le visible avec une image acquise dans le proche infrarouge) pouvant présenter des inversions de contraste.

Par ailleurs, dans ce cas, la dimension du descripteur est alors divisée par 2 (passant de 128 à 64). Ceci peut également présenter un avantage en terme **d'occupation de la mémoire**.

### Validité des demi-descripteurs

Voyons si ces "demi-descripteurs" sont suffisants pour permettre une bonne détection de points homologues entre les images.

Des tests de calcul de points homologues ont donc été effectués sur quelques paires d'images aériennes en utilisant d'une part des descripteurs SIFT normaux (de dimension 128) et d'autre part des demi-descripteurs.

- Un test a porté sur une paire d'images aériennes d'une même bande issue d'une prise de vue départementale. Dans ce cas, on obtient (après élimination des fautes) 41513 points homologues contre 47943 avec le descripteur complet.
- Un second test a porté sur une paire d'images aériennes en interbande (le recouvrement entre ces clichés est assez faible). On obtient (après élimination des fautes) 82 points homologues avec les demi-descripteurs au lieu de 95 points avec les descripteurs complets.
- Un autre test a porté sur une paire d'images acquises par Stéréopolis. Dans ce cas, 326 points homologues sont obtenus (après filtrage des fautes) avec les demi-descripteurs contre 406 avec les descripteurs normaux.

Lors de ces tests, on obtient donc généralement un peu moins d'appariements lorsque l'on utilise les demi-descripteurs que lorsque l'on utilise les descripteurs SIFT classiques. Il semble également que le demi-descripteur soit un peu moins discriminant que le descripteur complet ; il y a en effet un peu plus d'erreurs d'appariement. Néanmoins, il permet malgré tout d'obtenir des résultats satisfaisants.

### Apparier des images de modalité différente

De telles images peuvent présenter des inversions de contraste. Deux solutions sont possibles :

- On peut travailler sur le module du gradient de l'image au lieu de travailler directement sur l'image originale
- On peut utiliser un demi-descripteur.

Néanmoins, pour que l'appariement réussisse, il faut dès le départ avoir extrait sur chaque image des points correspondant aux mêmes détails de l'image, ce qui n'est malheureusement pas toujours le cas.

Intéressons nous au cas "extrême" où l'on essaie d'apparier une image avec son négatif, on n'obtient aucun point homologue en utilisant les descripteurs SIFT normaux, tandis qu'un nombre très raisonnable de points est détecté si l'on utilise les demi-descripteurs.

## 11 Possibilité d'utilisation de l'échelle et de l'orientation des points SIFT

### 11.1 Introduction

On se place ici dans le cas de figure où la transformation entre les deux images est une similitude 2D. Dans ce cas, il est possible d'obtenir un jeu de paramètres (approchés) de cette similitude à partir d'une seule paire de points SIFT homologues. À chaque point SIFT extrait d'une image sont en effet associées les informations suivantes :

- ses coordonnées image (2D)  $(x; y)$
- son échelle caractéristique  $\sigma$
- son orientation caractéristique  $\theta$
- son descripteur  $V_{SIFT}$

Lors du calcul de points homologues entre deux images, les descripteurs sont bien entendu utilisés lors de l'appariement entre points SIFT.

Les coordonnées 2D  $(x, y)$  des points appariés peuvent ensuite être utilisées pour calculer une "transformation" géométrique entre ces deux images (et éventuellement filtrer les fautes).

Pour le moment, on s'est donc principalement intéressé à la précision des coordonnées 2D  $(x, y)$ . Néanmoins, les informations fournies par les échelles  $\sigma$  et les orientations  $\theta$  caractéristiques associées aux points appariés pourraient également s'avérer utiles dans certains cas, et plus particulièrement lorsque la transformation entre les images est une similitude (2D) (ou un de ses cas particuliers : simple application d'un facteur d'échelle ou simple rotation). Dans ce cas, un seul appariement permettrait de calculer les paramètres de la similitude :

Dans la suite, on note  $P_i(x_i, y_i, \sigma_i, \theta_i)$  un point SIFT de l'image  $i$  avec  $(x_i; y_i), \sigma_i$  et  $\theta_i$  désignant respectivement les coordonnées image, l'échelle et l'orientation associées à  $P_i$ .

Soit la similitude 2D  $S : S(x; y) = E \cdot \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix}$

Les paramètres de cette transformation peuvent être estimés en deux étapes à partir **d'un seul appariement (correct)**  $(P_1, P_2)$ ,  $P_2 = S(P_1)$  :

1. Dans un premier temps, le facteur d'échelle  $E$  et l'angle de rotation  $\alpha$  de  $S$  peuvent être estimés indépendamment l'une de l'autre à partir des informations d'échelle et d'orientation associées à ces deux points.  $E = \frac{\sigma_2}{\sigma_1}$  et  $\alpha = \theta_2 - \theta_1$ .
2. Dans un deuxième temps, la translation  $(T_x, T_y)$  peut être estimée.

$$\begin{pmatrix} T_x \\ T_y \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - E \cdot \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

Dans le cas où l'on a plusieurs appariements parmi lesquelles des fautes, on va considérer que chaque appariement apporte une hypothèse sur les paramètres de  $S$ , le "meilleur" jeu de paramètres étant obtenu par accumulation et vote.

On déduit alors de cette méthode la technique de filtrage des erreurs d'appariement décrite en 5.2.3.

Il est donc important du fait de leur intérêt potentiel de s'intéresser à la précision de ces ratios d'échelles caractéristiques et à ces différences d'orientations caractéristiques. Dans les tests qui suivent, les images utilisées sont les mêmes que dans la partie 6 précédente.

### 11.2 Sensibilité

Dans un premier temps, on fait varier les paramètres  $E$  et  $\alpha$  indépendamment. Ainsi, dans un premier temps (figures 43 et 44),  $E$  évolue tandis que  $\alpha = 0$ , puis dans un deuxième temps (figures 45 et 46), on fait varier l'angle de la rotation  $\alpha$  pour  $E = 1$ .

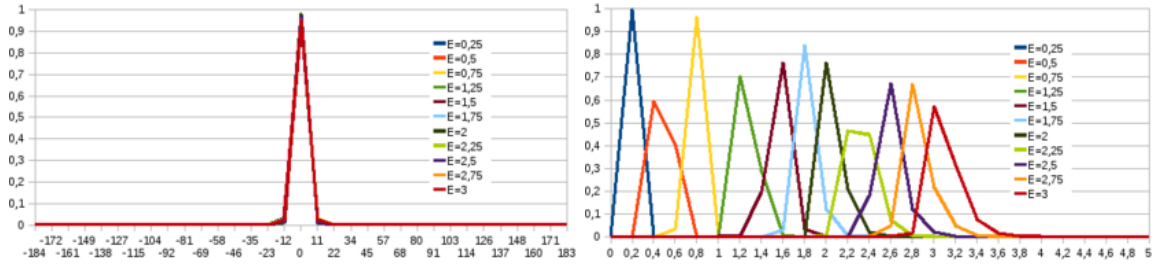


FIGURE 43 – Tests sur les images aériennes de la partie 6 pour l'application de différents facteurs d'échelle  $E$  ( $\alpha = 0$ ). À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

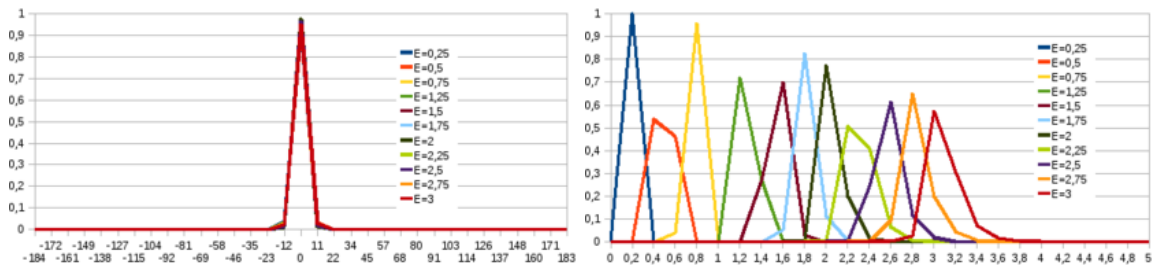


FIGURE 44 – Tests sur les images terrestres de la partie 6 pour l'application de différents facteurs d'échelle  $E$  ( $\alpha = 0$ ). À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

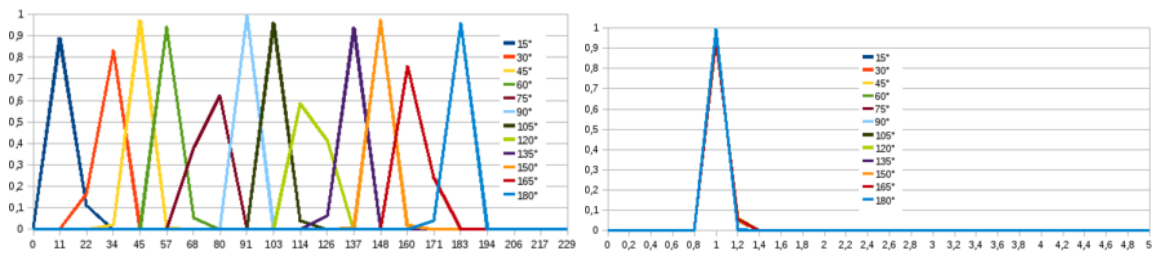


FIGURE 45 – Tests sur les images aériennes de la partie 6 pour l'application de différentes rotations 2D ( $E=1$ ). À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

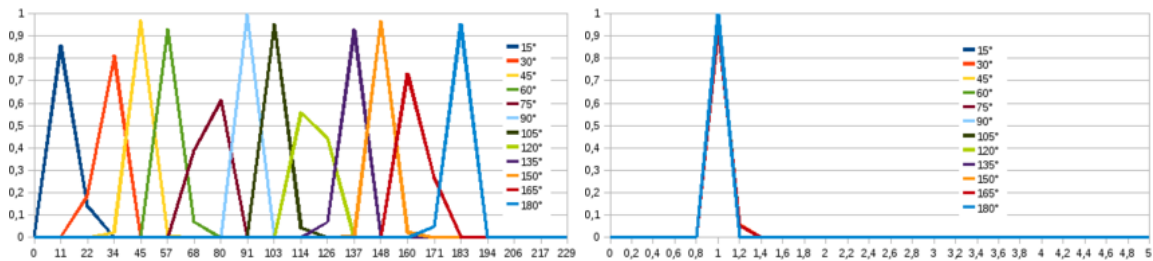


FIGURE 46 – Test sur les images terrestres de la partie 6 pour l'application de différentes rotations 2D ( $E=1$ ). À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

À l'issue de ces différents tests, on constate que le pic de l'histogramme est toujours proche de la valeur recherchée et regroupe généralement une écrasante majorité des appariements. (D'ailleurs, les situations pour lesquelles ce n'est pas le cas sont liées à la discrétisation des bins de l'histogramme et correspondent en fait à une valeur recherchée à la limite entre deux bins consécutifs, comme pour  $E = 750$  sur les figures 45 et 46.)

### 11.3 Détermination de $E$ et de $\alpha$

On essaie maintenant de réellement déterminer les paramètres  $E$  et  $\alpha$  directement à partir des ratios  $\frac{\sigma_2}{\sigma_1}$  et des différences  $\theta_2 - \theta_1$ . La méthode consiste à :

1. Calculer les histogrammes respectifs  $H_E$  et  $H_\alpha$  des  $\frac{\sigma_2}{\sigma_1}$  et des  $\theta_2 - \theta_1$
2. Déterminer les pics de ces histogrammes :  $\hat{E} = \operatorname{argmax}(H_E)$  et  $\hat{\alpha} = \operatorname{argmax}(H_\alpha)$ .
3. Affiner les valeurs de  $\hat{E}$  et  $\hat{\alpha}$ .  
 $\hat{E}$  = moyenne des valeurs situées au voisinage du pic de  $H_E$   
 $\hat{\alpha}$  = moyenne des valeurs situées au voisinage du pic de  $H_\alpha$

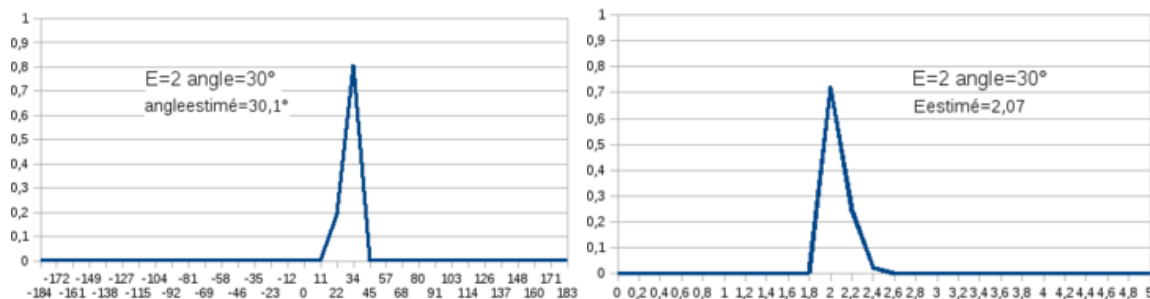


FIGURE 47 – Test sur les images aériennes de la partie 6 pour une similitude d'échelle  $E = 2$  et d'angle  $\alpha = 30^\circ$ . À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

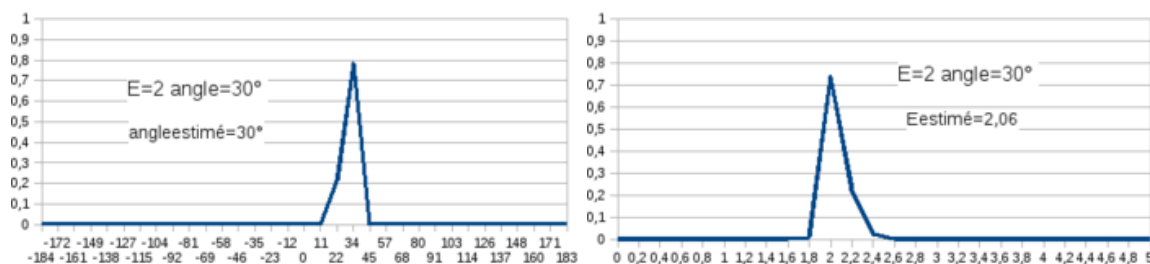


FIGURE 48 – Test sur les images terrestres de la partie 6 pour une similitude d'échelle  $E = 2$  et d'angle  $\alpha = 30^\circ$ . À gauche, répartition des différences  $\theta_2 - \theta_1$ . À droite, répartition des ratios  $\frac{\sigma_2}{\sigma_1}$ .

On constate que les résultats obtenus sont proches de la vraie valeur.

### 11.4 Bilan

En conclusion, il est possible d'estimer les paramètres (approchés) d'une similitude 2D entre les deux images en utilisant toutes les informations associées aux points SIFT mis en correspondance.

## Troisième partie

# Tests sur des images aériennes

Les différents tests menés ici ont montré que la mise en correspondance de points de Lowe par la méthode décrite donne généralement de bons résultats (et ceci sans aucune connaissance géométrique a priori (prédicteur)). Elle est notamment peu sensible aux rotations et aux variations d'échelles ainsi qu'au bruit, aux variations d'éclairement, aux changements de points de vue et à certaines transformations affines (modérées). SIFT est néanmoins sensible au diachronisme (i.e. au fait que les images n'ont pas été acquises au même moment et ne se ressemblent pas).

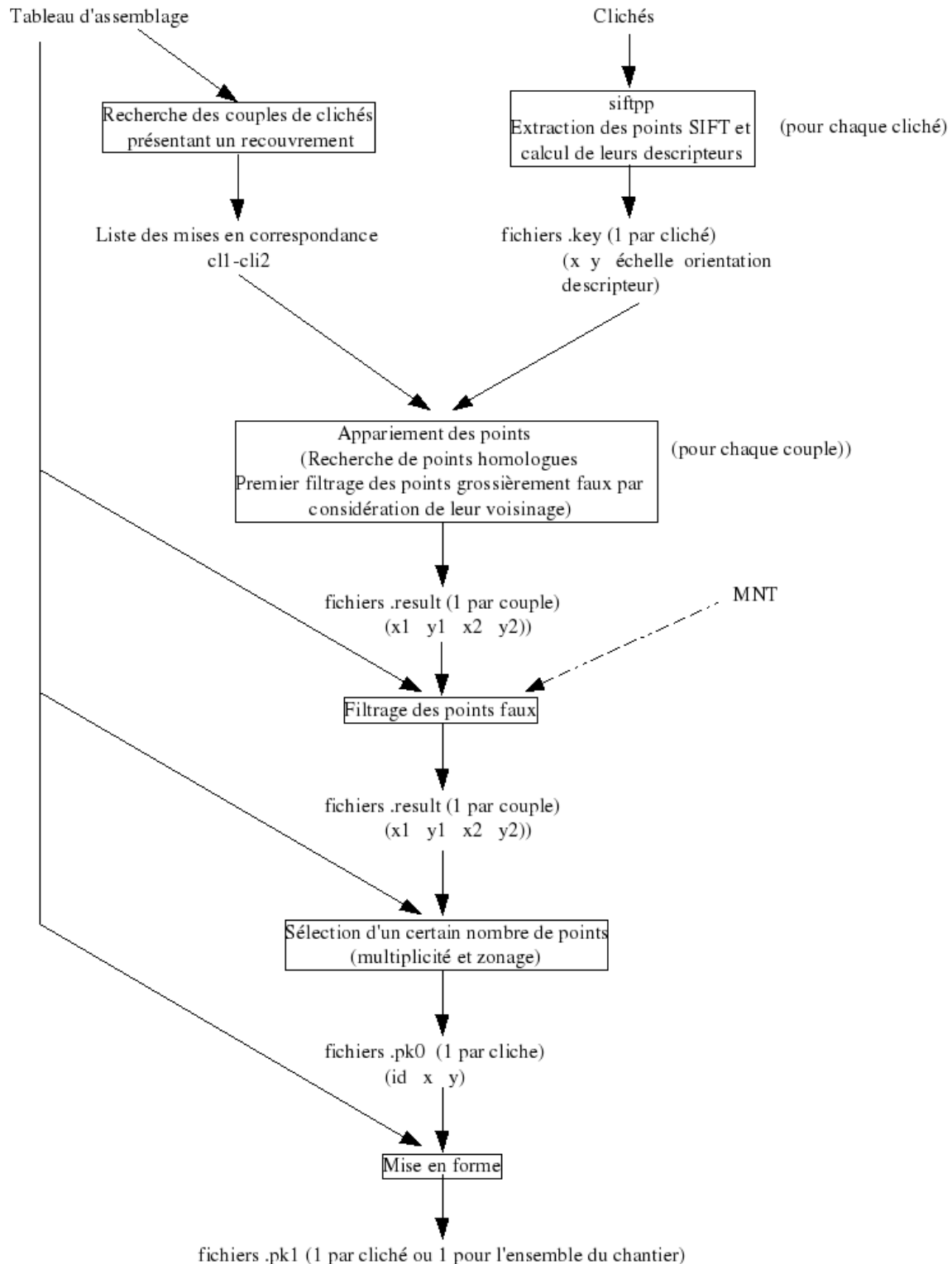
## 12 Utilisation de SIFT pour l'aérotriangulation : calcul de points de liaison entre images aériennes

Les tests qui suivent visent à évaluer l'aptitude de SIFT à fournir des points de liaison destinés aux calculs d'aérotriangulation de chantiers aériens.

### 12.1 Méthode et outils

#### 12.1.1 Méthode

La méthode mise en œuvre pour l'extraction de points de liaison utilisables lors d'un calcul d'aérotriangulation est résumée dans le schéma suivant.



**Étape 1 : Extraction des points d'intérêt (cliché par cliché)** Les points d'intérêt et leurs descripteurs sont calculés pour chaque cliché indépendamment des autres clichés. D'un point de vue pratique, cette étape est donc parfaitement parallélisable (à l'aide d'un système de calcul distribué *condor* [TTL05] sur un pool de machines, ou du système "*make.exe -f makefile -j nb\_proc*" sur une machine unique, ...). Un fichier contenant les coordonnées des points d'intérêt ainsi que leur descripteur est créé pour chaque cliché.



**Étape 2 : Détermination des paires d'images voisines** Disposant du tableau d'assemblage (et de l'orientation approchée des clichés), il est possible de savoir quels sont les voisins d'un cliché donné, et donc les paires de clichés entre lesquels il va être *a priori* possible de trouver des points homologues. L'appariement des points d'intérêt ne sera donc effectué que pour ces paires d'images.

Remarque : Au cas où l'on ne dispose pas de tableau d'assemblage (ou du moins de liste des paires de clichés en recouvrement) il est possible d'en créer un de manière automatique à l'aide de SIFT, comme cela sera le cas dans la partie 13.)

**Étape 3 : Appariement des points d'intérêt (paire par paire)** Les points homologues sont recherchés (parmi les points d'intérêt détectés à l'issue de l'étape 1) pour chaque paire de clichés sélectionnée à l'étape 2. D'un point de vue pratique, cette étape est donc parallélisable. Pour chaque paire de clichés à appairer, un fichier contenant les points homologues mis en correspondance est créé.

#### Utilisation d'un prédicteur ?

Au cours des tests effectués, aucun prédicteur n'a été utilisé dans la plupart des cas.

Il est en effet apparu que SIFT fonctionnait très bien sans prédicteur, fournissant des appariements très majoritairement corrects, même dans des cas où le recouvrement entre images est faible et où les images ne sont pas orientées de la même manière. On peut voir quelques exemples d'appariements sans prédicteur sur les figures 58,59 et 60.

Néanmoins, si l'on dispose d'un tableau d'assemblage et d'un géoréférencement approché des clichés, l'utilisation d'un "prédicteur" très approché est possible et peut permettre de bien réduire les calculs et d'éviter certaines fautes grossières. Ainsi qu'on l'a vu en 9, disposant du géoréférencement approché des clichés que l'on souhaite appairer (ainsi qu'un MNT même très grossier), il est en effet possible de calculer approximativement pour chaque paire de clichés 1 et 2 quelle est la zone  $R_{2dans1}$  du cliché 1 en recouvrement avec le cliché 2, et inversement quelle zone  $R_{1dans2}$  du cliché 2 est en recouvrement avec le cliché 1. On peut dès lors restreindre le nombre de points mis en jeu dans la procédure d'appariement : on cherche alors uniquement à appairer les points du cliché 1 appartenant à  $R_{2dans1}$  aux points du clichés 2 appartenant à  $R_{1dans2}$ . On réduit ainsi le nombre de requêtes et donc les temps de calcul (en particulier pour des paires d'images présentant un faible recouvrement), et on évite d'obtenir des appariements erronés mettant en jeu des points situés en dehors de ces zones.

Le prédicteur utilisé lors de ces procédures d'appariement de points SIFT est donc un prédicteur global portant sur les ensembles de points à appairer et non un prédicteur "local" définissant une fenêtre de recherche d'homologue propre à chaque point à appairer et centrée sur la position approchée de son homologue calculée d'après l'orientation approchée des clichés.

**Étape 4 : Filtrage par considération du voisinage** Il s'agit d'un premier filtrage simple visant à éliminer les erreurs grossières. Ce filtrage fonctionne bien dans le cas de l'aérien (Le sol est une surface continue en "2,5D" (et non en vrai 3D) aux variations généralement pas trop brutales.) Le principe déjà décrit en 5.2.1 en est simple : on vérifie qu'un point a majoritairement les mêmes voisins dans les deux images, i.e. que les homologues dans l'image 2 des N plus proches voisins d'un point P de l'image 1 se retrouvent majoritairement parmi les N plus proches voisins de l'homologue de P dans l'image 2. L'algorithme est donc le suivant :

---

**pour tout** point apparié P de l'image 1 **faire**

$S1 = \{N \text{ plus proches voisins de P parmi les points appariés de l'image 1} \}$

$S2 = \{N \text{ plus proches voisins de l'homologue de P parmi les points appariés de l'image 2} \}$

$nbvc = \text{nombre de points de } S1 \text{ dont l'homologue est dans } S2$

**si**  $\frac{nbvc}{N} < \text{seuil}$  **alors**

Le point P et son homologue sont rejetés.

**sinon**

Le point P et son homologue sont conservés.

**finsi**

**fin pour**

---

Cette étape de filtrage est parallélisable puisqu'elle s'effectue paire de clichés par paire de clichés. En pratique, elle est directement intégrée dans l'outil d'appariement déjà utilisé à l'étape 3 : pour une paire d'image donnée, l'appariement de points d'intérêts est donc directement suivi de ce filtrage.

**Étape 5 : Filtrage “photogrammétrique”** Le but de ce filtrage est d'éliminer des mises en correspondance fausses en utilisant l'orientation approchée des clichés fournie par le tableau d'assemblage. Il s'agit ici d'éliminer les erreurs de mise en correspondance les plus grossières. Un autre filtrage des fautes sera de toutes manière effectué par la suite lors du calcul d'aérotriangulation.

La méthode retenue est la suivante :

---

**tantque** l'état (valide/non valide) des points évolue **faire**

**pour tout** point apparié P de l'image 1 considéré comme valide (à la première itération, tous les points sont considérés comme valides) **faire**

Calcul de la distance  $d(P)$  entre les coordonnées théoriques (calculées à partir des orientations approchées des clichés 1 et 2 et d'un MNT de la zone) et mesurées de son homologue dans l'image 2

**fin pour**

Calcul de l'écart-type  $\sigma$  et de la moyenne  $\bar{d}$  des  $d(P)$

**pour tout** point apparié P de l'image 1 **faire**

**si**  $|d(P) - \bar{d}| > coef \cdot \sigma$  **alors**

Le point P et son homologue ne sont pas considérés comme valides.

**sinon**

Le point P et son homologue sont considérés comme valides.

**finsi**

**fin pour**

**fin tantque**

Seuls les points considérés comme valides sont conservés.

---

**Étape 5 : Sélection de points bien répartis** Tous les points homologues issus de l'appariement n'ont pas vocation à être conservés : ils sont trop nombreux pour le calcul d'aérotriangulation et ne sont pas répartis de manière homogène. Par ailleurs, les points de multiplicité supérieure ou égale à 3 sont plus intéressants que les points vus uniquement sur deux clichés.

La méthode mise en œuvre est donc la suivante :

1. Chargement de tous les points homologues (jusqu'ici connus uniquement paire d'image par paire d'image)
2. Numérotation (un identifiant unique par point “physique”) et calcul de la multiplicité des points de liaison

Remarque : des fautes peuvent être détectées à l'issue de ce chaînage des appariements en s'apercevant que deux points distincts d'une même image ont reçu le même identifiant et sont donc sensés correspondre donc à un même point physique...

3. Pour chaque image, mise en place d'une grille. Un seul point de liaison est conservé par case de cette grille. Le critère de sélection actuellement utilisé pour cette sélection est la multiplicité : on ne conserve qu'un des points de plus forte multiplicité au sein de chaque case. Il serait néanmoins sans doute souhaitable d'ajouter une contrainte sur l'appartenance à une ou plusieurs bandes.

Dans les différents tests effectués, la grille était régulière et couvrait toute l'image, mais il est parfaitement possible d'adopter une stratégie proche de celle de PTLiaison en ne recherchant des points que dans une zone prédéfinie de l'image (zone de nadir).

**Étape 6 : Mise en forme pour le logiciel d'aérotriangulation** Il s'agit ici simplement de préparer les fichiers contenant les points de liaison au format (format du fichier, mais également passage du repère image au repère photogrammétrique et correction de la distorsion) attendu par le logiciel d'aérotriangulation (en l'occurrence ici le format pk1 attendu par TopAero).

### 12.1.2 Mise en œuvre opérationnelle

**Temps de calcul** Rappelons ici que la phase d'extraction des points SIFT (et de calcul de leurs descripteurs) avec *SIFTTP* prend généralement de 3 à 5 minutes par image de la caméra numérique IGN V1. L'appariement s'effectue également en 3 à 5 minutes par paire d'images.

Remarque : L'utilisation d'une implémentation de SIFT utilisant la puissance de calcul des GPU permet de considérablement réduire les temps de calcul (notamment pour l'extraction). Néanmoins, de telles implémentations n'étaient pas disponibles lors des premiers tests, et même ensuite, il restait des restrictions sur le type de carte graphique installée sur les PCs.

**Utilisation du calcul distribué** Étant donnés les temps de calcul et le volume de ces chantiers, ainsi que le fait que la plupart des calculs sont hautement parallélisables (travail cliché par cliché ou paire de clichés par paire de clichés), le système de calcul distribué *condor* ([TTL05], [Con09b], [Con09a]) sur un pool de machines a été utilisé afin de traiter les chantiers en un temps raisonnable.

Par ailleurs, les tests les plus récents (cf section 12.2.6) ont été plus simplement effectués sur une seule machine plus récente autorisant jusqu'à 16 calculs en parallèle. (On a alors utilisé le système "*make.exe -f makefile -j nb<sub>proc</sub>*" pour lancer ces tâches.) Le traitement des différents chantiers se faisait alors en un temps raisonnable.

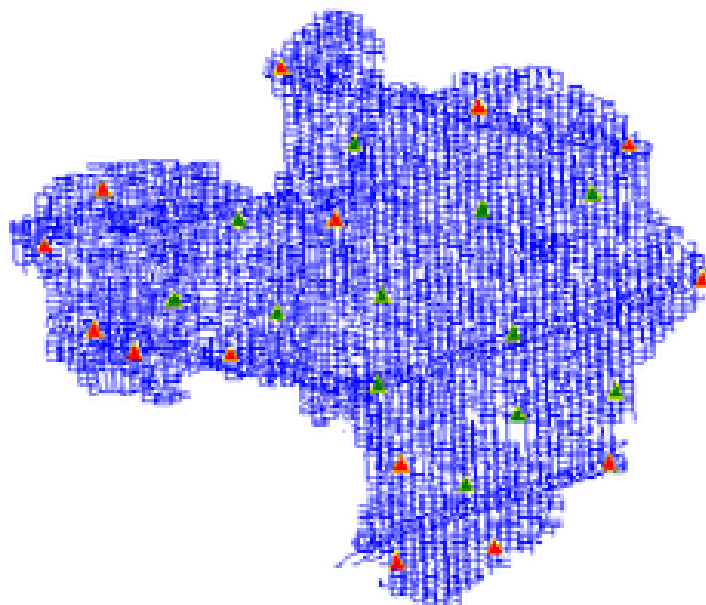
## 12.2 Prises de vue départementales

Il s'agit ici de traiter des chantiers complets de prise de vue départementale. On se place donc dans le contexte de la production. Dans ce cas de figure, on dispose d'un tableau d'assemblage et d'une orientation approchée des images. Cette connaissance *a priori* peut être utilisée pour :

- établir *a priori* la liste des clichés susceptibles de se recouvrir, c'est-à-dire la liste des paires d'images pour lesquels un appariement de points homologues doit être effectué
- éliminer de mauvaises mises en correspondance à l'issue de l'étape d'appariement des points homologues

### 12.2.1 Département du Gard (FD30)

FIGURE 49 – Tableau d'assemblage du chantier sous TopAero



Le traitement de ce chantier de prise de vue départementale (fig. 49) comportant 3949 clichés acquis par la caméra numérique IGN V1 s'était plutôt bien passé avec les outils utilisés en production. Le but de ce premier test était donc de savoir quels résultats SIFT pouvait permettre d'obtenir (mieux, de même qualité ou moins bons qu'avec les outils de production) ainsi que de développer les outils susceptibles de traiter un chantier de cette taille. Par ailleurs, ce chantier comporte une zone littorale (et donc des clichés comportant des zones d'eau très importantes) susceptible de poser des problèmes.

Les points de liaison fournis par SIFT ont une répartition à peu près homogène sur l'ensemble du chantier. Aucun problème spécifique ne se produit pour les clichés de la zone littorale. Les résultats obtenus à l'issue de l'aérotriangulation s'avèrent assez semblables ceux obtenus précédemment en production, avec les outils habituels. Ils sont résumés dans le tableau 1.

TABLE 1 – EMQ de l'aérotriangulation

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)		
	X	Y	XY
tous	0.34	0.33	0.47
inter-bandes	0.35	0.33	0.48

EMQ Points terrain (en m)	EMQ Points terrain (en m)				
	X	Y	Z	XY	XYZ
Appui	0.70	0.76	0.91	1.03	1.38
Controle	0.74	0.69	0.64	1.01	1.20
GPS	0.00	0.00	0.01	0.00	0.01

### 12.2.2 Département de la Manche (FD50)

Ce chantier de 3440 clichés acquis par la caméra numérique IGN V1 posait des problèmes en production. Il était donc intéressant d'y tester SIFT afin de savoir s'il pouvait permettre une amélioration. La non-détection par SIFT de points de liaison entre deux bandes acquises à près d'un mois d'intervalle a alors été constatée (ainsi qu'on peut le voir sur la figure 50).

FIGURE 50 – Absence de points de liaison entre deux bandes

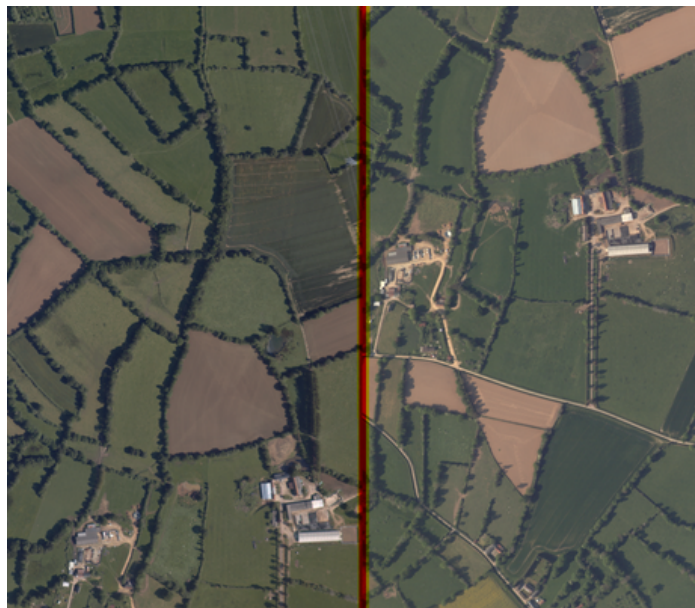


Cette absence de points de liaison sur un raccord interbande est due à un phénomène de diachronisme rendant les images trop différentes pour permettre des appariements (voir figure 51). Dans le cas présent, deux difficultés se combinaient :

- Les images ont été acquises à près d'un mois d'intervalle avec les changements au niveau de la végétation qui en résultent.
- Une bande a été acquise le matin et l'autre l'après-midi, ce qui a pour conséquence une "inversion" des ombres.

Les points détectés et mis en correspondance entre ces deux bandes par les outils actuellement utilisés en production (recherche d'homologue par corrélation avec prédicteur centré sur la position estimée de l'homologue) sont quant à eux très majoritairement faux : ils correspondent en effet généralement à des extrémités de zones d'ombre.

FIGURE 51 – Absence de points homologues à cause du diachronisme : les images sont trop différentes



Afin d'achever le calcul d'aérotriangulation de ce chantier, de nouveaux points de liaison ont été calculés sur la liaison interbande posant problème. Ces points ont été obtenus à l'aide des outils utilisés en production mais pour un paramétrage différent permettant d'obtenir cette fois des points corrects.

Deux calculs d'aérotriangulation ont été effectués :

- un calcul utilisant uniquement des points fournis par PTLiaison (avec un paramétrage différent pour le raccord interbande posant problème) (voir table 2)
- un calcul utilisant les points calculés par PTLiaison pour la liaison interbande posant problème et les points fournis par SIFT pour le reste du chantier (voir table 3)

TABLE 2 – EMQ de l'aérotriangulation pour des points PTLiaison

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)		
	X	Y	XY
tous	0.19	0.22	0.29
inter-bandes	0.28	0.28	0.39

	EMQ Points terrain (en m)				
	X	Y	Z	XY	XYZ
Appui	0.29	0.33	0.71	0.44	0.83
Controle	0.26	0.35	0.79	0.43	0.90
GPS	0.00	0.00	0.01	0.00	0.01

TABLE 3 – EMQ de l'aérotriangulation pour des points SIFT

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)		
	X	Y	XY
tous	0.19	0.19	0.26
inter-bandes	0.20	0.18	0.27

	EMQ Points terrain (en m)				
	X	Y	Z	XY	XYZ
Appui	0.32	0.26	0.67	0.41	0.78
Controle	0.29	0.34	0.79	0.45	0.90
GPS	0.00	0.00	0.02	0.00	0.02

Sur ce chantier, l'utilisation de SIFT a donc permis de diagnostiquer l'origine du problème, sans pour autant le résoudre. Dans un tel cas de fort diachronisme, plutôt que de donner des points de liaison erronés, SIFT ne fournit pas ou très peu de points de liaison et lorsqu'il en donne, il ne s'agit que de quelques points qui sont généralement suffisamment faux pour qu'il soit possible de s'en rendre compte.

À l'inverse, les outils actuellement utilisés en production fournissaient quant à eux toujours des points de liaison, mais, dans le cas présent, ces points étaient majoritairement faux tout en restant cohérents entre eux...

### 12.2.3 Département de la Nièvre (FD58)

Ce chantier comportant 3688 clichés acquis par la caméra numérique IGN V1 posait également des difficultés en production : les résidus de certains points terrain demeuraient en effet anormalement forts. Par ailleurs, afin d'obtenir des résidus acceptables pour les spécifications BDOortho, le systématisme cliché "déformation cylindrique" proposé par TopAero avait été calculé bande par bande (et non de manière unique pour l'ensemble des bandes du chantier).

Cette fois encore, SIFT n'a pas détecté de points de liaison sur certaines liaisons interbandes (voir figure 52), avec deux cas de figure différents :

- les bandes ont été acquises à un mois d'intervalle. (voir figure 55)
- les images sont quasi-entièrement couvertes par de la forêt et ont été acquises à quelques jours (1 ou 2) d'intervalle à des heures différentes : les ombres sont donc alors bien différentes. (voir figure 54)

FIGURE 52 – Absence de points de liaison entre certaines bandes voisines

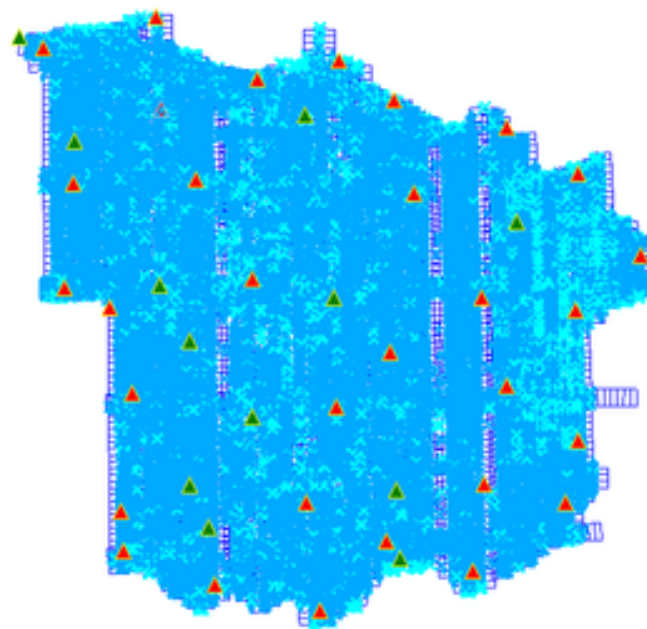
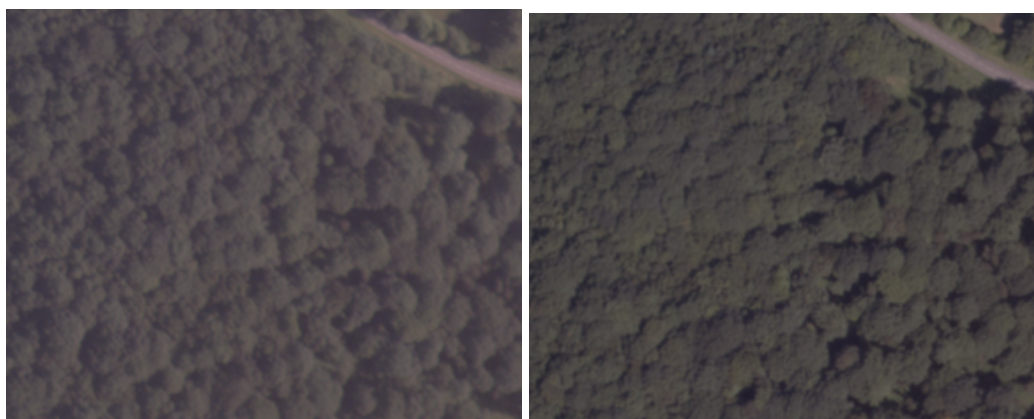


FIGURE 53 – Absence de points de liaison entre images voisines acquises à un mois d'intervalle



FIGURE 54 – Absence de points de liaison entre images voisines quasiment entièrement couvertes par de la forêt et acquises à quelques jours (1 ou 2) d'intervalle à des heures différentes



Le chantier a ensuite fait l'objet d'un calcul d'aérotriangulation en utilisant les points de liaison fournis par SIFT complétés par des points de liaison saisis par un opérateur sur les liaisons interbandes pour lesquelles SIFT avait "échoué". La saisie manuelle de ces points de liaison s'est d'ailleurs avérée difficile sur certaines parties de ces liaisons interbandes du fait des fortes différences entre les images... Les résultats du calcul (avec cette fois un seul systématisme cliché "Déformation cylindrique" pour l'ensemble du chantier) sont présentés dans le tableau 4.

TABLE 4 – EMQ de l'aérotriangulation pour des points SIFT complétés des points de liaison saisis par un opérateur

	EMQ Mesures image (en pixels)			EMQ Points terrain (en m)					
	X	Y	XY	X	Y	Z	XY	XYZ	
tous	0.21	0.19	0.28	Appui	0.31	0.26	0.39	0.41	0.56
inter-bandes	0.23	0.19	0.30	Contrôle	0.43	0.27	0.53	0.51	0.74
				GPS	0.00	0.00	0.02	0.00	0.02

On a observé que l'utilisation des points de liaison fournis par SIFT sur ce chantier apportait une amélioration sur l'EMQ à la fois en planimétrie et en altimétrie sur les points d'appui, et en planimétrie sur les points de contrôle. Par ailleurs, des améliorations locales significatives sur les résidus de certains des points de contrôle ont également été constatées, sans que des dégradations locales ne soient observées.

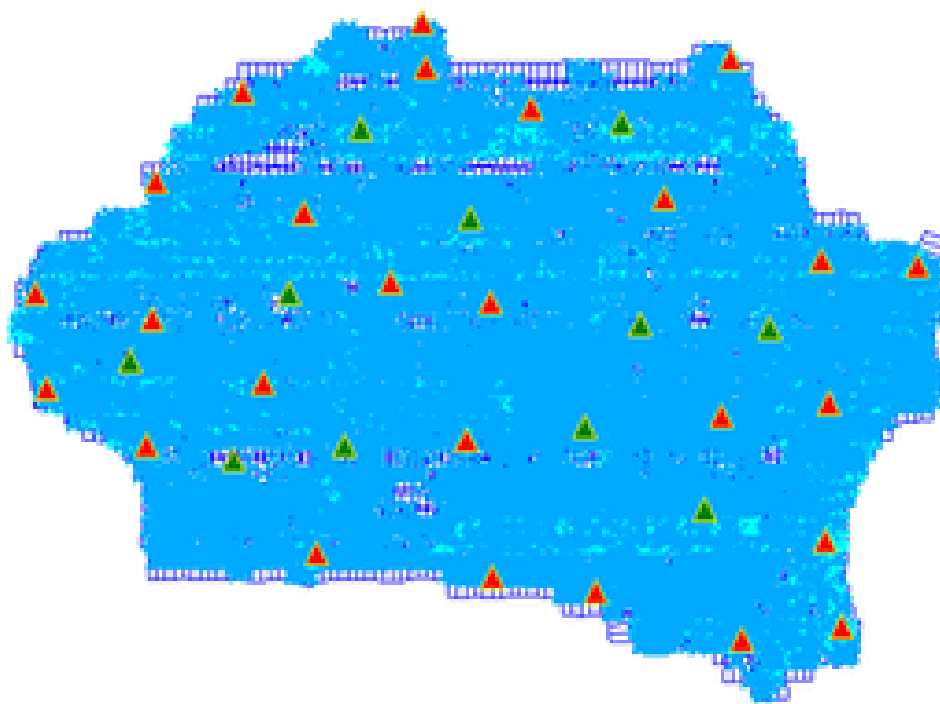
#### 12.2.4 Département de l'Allier (FD03)

Ce chantier (acquis par la caméra numérique IGN V1) avait également posé quelques difficultés en production. Néanmoins, des zones de forêt denses étaient susceptibles de poser quelques difficultés ; en effet, de nombreux points de liaison ont dû être saisis manuellement en interbande et sur les bandes transversales, à la fois sur les images sous-échantillonnées (ce qui correspond à la première phase du calcul d'aérotriangulation en production) et sur les images à pleine résolution. Il était donc cette fois encore intéressant de tester l'éventuelle utilité de SIFT dans ce cas de figure. Une expérience a alors été menée en partenariat avec l'équipe Produit BDOOrtho du Service des Bases de Données Image dans le cadre d'une étude préalable à la mise en place d'un nouvel outil de mesure de points d'appui et de liaison.

Les points calculés avec SIFT présentent les caractéristiques suivantes :

- Leur répartition est homogène sur les parties d'image en recouvrement.
- Les points multiples sont suffisamment nombreux et bien répartis à quelques exceptions près.
- Les bandes transversales sont bien équipées.
- Par contre, la densité de points en interbande est plus faible entre des bandes acquises à des dates différentes (15 jours). Généralement, cette densité est suffisante pour le calcul d'aérotriangulation, mais elle peut ici être nulle dans le cas lorsque le paysage correspond à de la forêt.

FIGURE 55 – Quelques trous peuvent être observés dans l'équipement en points de liaison fournis par SIFT



Deux calculs d'aérotriangulation ont été effectués :

- calcul à partir des points de liaison fournis par PTLiaison et la saisie opérateur
- calcul à partir des points de liaison fournis par SIFT uniquement (pas de points complémentaires).

Dans les deux cas, le paramétrage était le même, avec notamment l'utilisation du systématisme cliché "Déformation cylindrique" proposé par TopAéro. Les résultats sont présentés dans le tableau 5.



TABLE 5 – EMQ Points terrain (en m)

	points de liaison SIFT						points de liaison PTLiaison + points opérateur				
	X	Y	Z	XY	XYZ		X	Y	Z	XY	XYZ
Appui	0.28	0.31	0.24	0.41	0.48	Appui	0.27	0.29	0.41	0.40	0.57
Controle	0.44	0.32	0.58	0.54	0.80	Controle	0.44	0.54	0.92	0.70	1.16

Les résidus planimétriques sont quasiment équivalents tandis que les résidus altimétriques sont quasiment réduits de moitié avec le calcul utilisant les points de liaison fournis par SIFT. (On rappellera au passage que la trajectographie GPS disponible pour ce chantier avait été calculée par la méthode PPP, ce qui peut avoir des conséquences sur le fait que le calcul avec les points de liaison fournis par SIFT reste bon malgré les quelques trous, vu que la contrainte sur la position des sommets de prise de vue est alors plus forte et que ceux-ci sont mieux positionnés.)

Par ailleurs, le taux de points filtrés par TopAero est relativement faible (un peu moins de 2%) par rapport à ceux habituellement observés dans les mêmes conditions dans le cadre de la chaîne actuelle de production (filtrage pouvant aller jusqu'à 20% des points).

Par ailleurs, ce chantier a fait l'objet d'un test complémentaire. Les temps de calcul pour la détermination de points homologues par SIFT sont importants à pleine résolution, or par défaut SIFT débute la phase d'extraction à l'octave -1, c'est-à-dire en suréchantillonnant l'image d'un facteur 2. Il était donc intéressant de voir ce qui se passerait en choisissant cette fois l'octave 0 comme premier octave lors de l'extraction (cf partie 6.11). Les temps de calcul sont alors en effet fortement réduits. Les résultats de l'aérotriangulation utilisant ces points de liaison sont présentés dans le tableau 6. Ils sont moins bons que ceux obtenus avec le mode par défaut ("first-octave -1"), mais restent acceptables.

TABLE 6 – EMQ Points terrain (en m) pour des points de liaison SIFT obtenu en travaillant à partir de l'octave 0

	X	Y	Z	XY	XYZ
Appui	0.48	0.48	0.51	0.68	0.85
Controle	0.53	0.39	0.68	0.65	0.94

### 12.2.5 Département de la Dordogne (FD24)

Ce chantier a fait l'objet d'une acquisition avec la caméra UltraCamXP, avec une résolution de 40 cm. Seul un sous-bloc du chantier a fait l'objet d'un test d'aérotriangulation avec SIFT. Ce sous-bloc a été choisi dans une zone de collines relativement boisées. Par ailleurs, il est encadré par des bandes transverses et est suffisamment bien équipé en points d'appui pour obtenir une aérotriangulation valable (voir figure 56).

Tout comme lors du test précédent, l'extraction des points SIFT a été effectuée avec 2 valeurs différentes (-1 ou 0) pour le paramètre "first-octave" concernant le choix de la première octave.

Cette fois encore, il semble au vu des résultats d'aérotriangulation présentés dans les tableaux 7 et 8 que le mode "first-octave 0" soit suffisant.

FIGURE 56 – Tableau d'assemblage du chantier sous TopAero

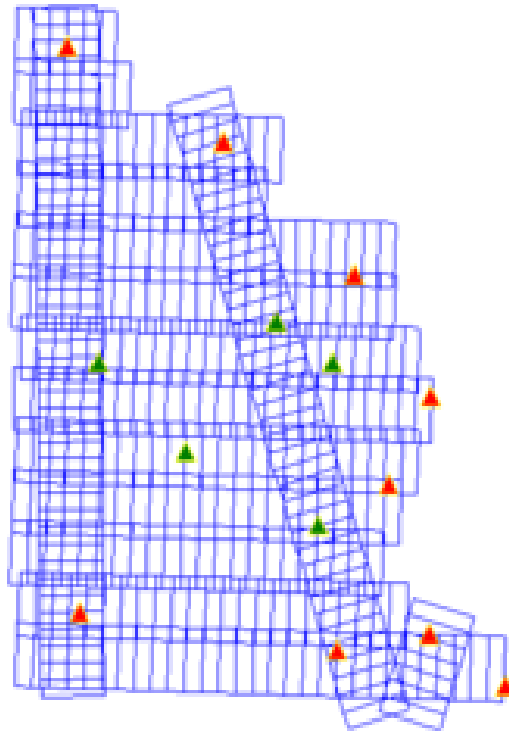


TABLE 7 – EMQ de l'aérotriangulation calculée à partir de points de liaison SIFT extraits en mode "first-octave 0"

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)								
	X	Y	XY	X	Y	Z	XY	XYZ	
tous	0.35	0.36	0.50	Appui	0.12	0.15	0.25	0.19	0.31
inter-bandes	0.43	0.36	0.56	Contrôle	0.15	0.15	0.28	0.22	0.36
				GPS	0.08	0.09	0.06	0.12	0.13

TABLE 8 – EMQ de l'aérotriangulation calculée à partir de points de liaison SIFT extraits en mode "first-octave -1"

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)								
	X	Y	XY	X	Y	Z	XY	XYZ	
tous	0.23	0.23	0.33	Appui	0.07	0.16	0.27	0.18	0.32
inter-bandes	0.29	0.24	0.37	Contrôle	0.14	0.14	0.24	0.20	0.31
				GPS	0.07	0.08	0.06	0.11	0.12

### 12.2.6 Département de la Sarthe

Ce chantier départemental a été acquis par la caméra numérique V2 de l'IGN, avec un pixel sol de 25 cm. Des problèmes ont été rencontrés lors du traitement de ce chantier en production. Un test avec SIFT a donc été effectué, mais n'a pas permis de résoudre ce problème (sans doute un souci de calibration géométrique de la caméra?). Aussi les résultats des aérotriangulations ne seront pas présentés ici.

Néanmoins, d'un point de vue opérationnel, ce chantier a été traité différemment des chantiers précédents, avec cette fois l'utilisation d'une seule machine sur laquelle 16 calculs peuvent se faire en parallèle (à l'aide du système "`make.exe -f makefile -j nb_proc`" sur une machine unique), et non plus du système de calcul distribué "condor" sur un pool de machines. Par ailleurs, tout comme pour le test de FD24 (cf 12.2.5), les points de liaison ont été

extraits pour deux valeurs différentes du paramètre de sélection de la première octave (-1 et 0). Les temps de calcul obtenus sont présentés dans le tableau 9 : on peut constater qu'ils sont raisonnables (chantier traité en 3 jours) et sont bien réduits en mode "first-octave 0".

TABLE 9 – Comparaison des temps de calcul en mode "first-octave -1" et "first-octave 0"

	"first-octave -1"	"first-octave 0"
Estraction (en parallèle x7)	64,25 h	20 h
Appariement sans emprises (en parallèle x15)	12,75 h	6,75 h
Appariement avec emprises (en parallèle x15)	3,25 h	1,25 h

**Statistiques sur la multiplicité des points de liaison (first-octave 0) :** Pour ce chantier, des statistiques sur la multiplicité des points de liaison extraits ont également été calculées :

2 : 16.74 % - 3 : 62.35 % - 4 : 8.42 % - 5 : 5.40 % - 6 : 6.42 % - 7 : 0.33 % - 8 : 0.18 % - 9 : 0.13 % - 10 : 0.01 % - 11 : 0.007 % - 12 : 0.004 % - 13 : 0.002 % - 14 : 0.001 %

### 12.2.7 Conclusions

SIFT a été testé sur plusieurs chantiers de prises de vue départementales dont certains avaient posé des soucis aux outils actuellement utilisés en production. Dans ces cas particuliers (sections 12.2.2 et 12.2.3), SIFT a permis de diagnostiquer les causes des problèmes sans pour autant totalement les résoudre.

En cas de fort diachronisme, si les images sont trop différentes, SIFT présente en effet l'avantage de ne pas (ou peu) fournir de points de liaison (plutôt que de donner des points nombreux et faux) et lorsqu'il donne quelques points faux, ceux-ci soient généralement suffisamment faux pour qu'il soit possible de s'en apercevoir (voir figure 61).

En dehors de ces cas problématiques, la répartition des points fournis par SIFT est à peu près homogène (voir figures 58, 59 et 60).

Lors des calculs d'aérotriangulation effectués, les résultats étaient généralement du même ordre que ceux obtenus avec des points de liaison fournis par PTLiaison. Par contre, le taux de points de liaison éliminés lors de l'aérotriangulation est nettement inférieur avec SIFT qu'avec PTLiaison.

En revanche, la multiplicité des points de liaison extraits par SIFT est globalement inférieure à celle des points calculés par PTLiaison : les points fournis par SIFT sont en effet massivement des points triples. Ce phénomène est lié à la différence entre les deux algorithmes :

- avec SIFT, on recherche en effet des points homologues paire par paire avant d'identifier, parmi tous ces appariements, ceux qui correspondent à un même point 3D
- avec PTLiaison, on utilise un prédicteur pour rechercher pour un point extrait d'un cliché donné ses homologues dans tous les clichés en recouvrement, ce qui garantit un nombre élevé de points de multiplicité supérieure à 4.

FIGURE 57 – Exemples d'appariements entre deux images consécutives

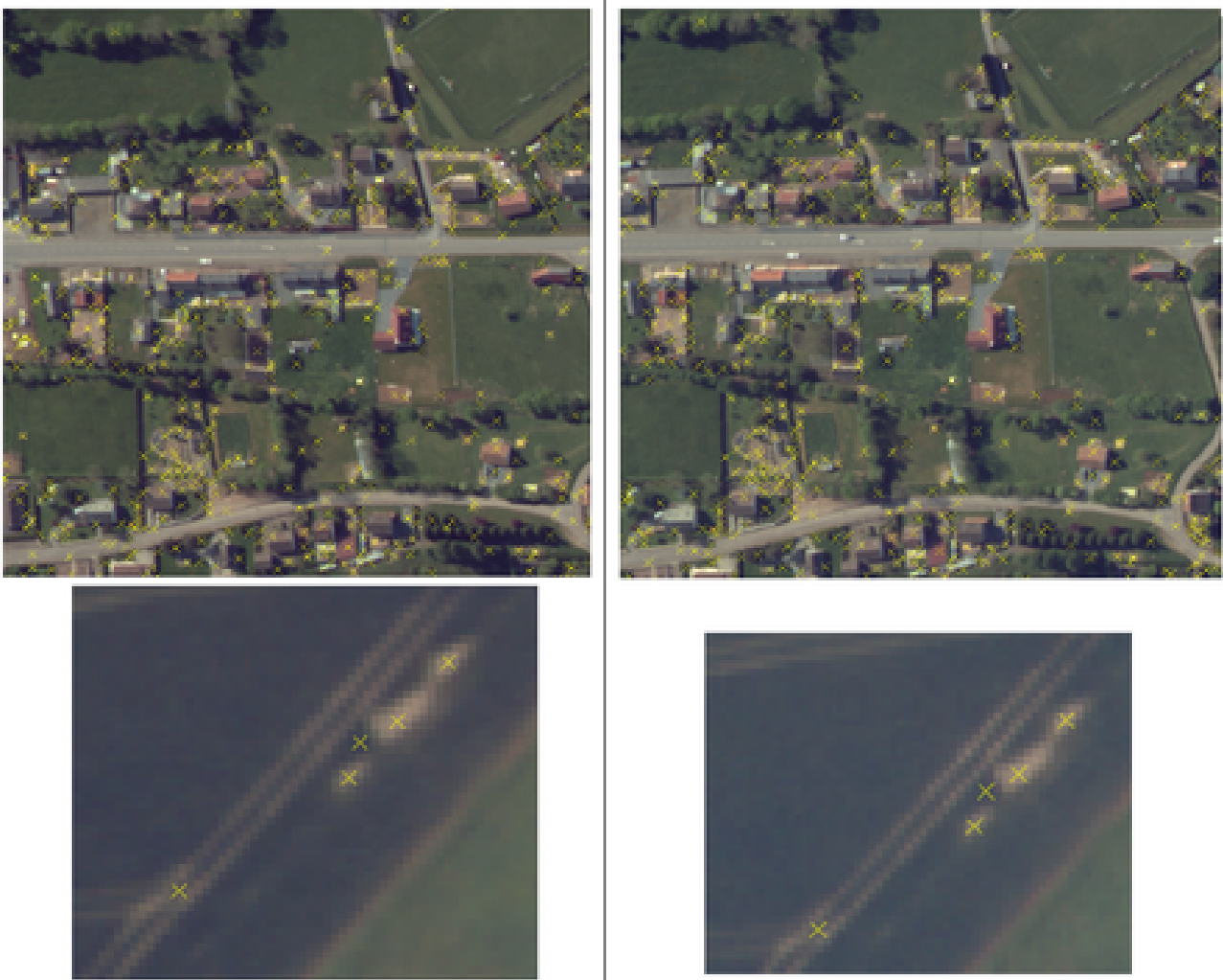


FIGURE 58 – Exemple d'appariements entre deux images consécutives en intrabande (avant et après sélection d'un nombre limité de points pour l'aérottriangulation)

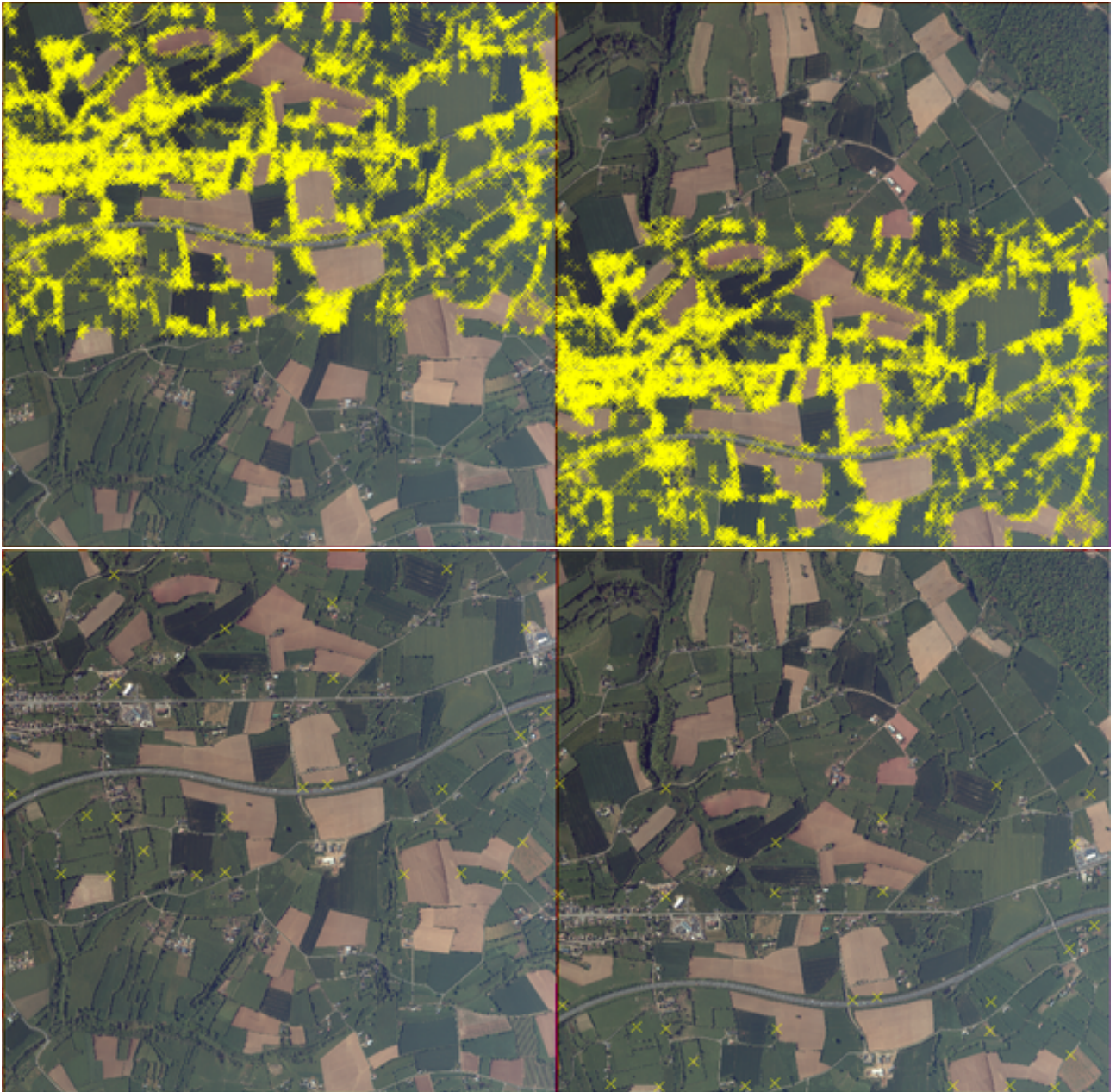


FIGURE 59 – Exemple d'appariements entre deux images en interbande (avant et après sélection d'un nombre limité de points pour l'aérotriangulation)



FIGURE 60 – Exemple d'appariements entre deux images en interbande pour un faible recouvrement



FIGURE 61 – Exemple d'appariements obtenus entre deux images en interbande dans un cas de fort diachronisme : les fautes sont suffisamment importantes pour qu'il soit possible de s'en rendre compte



**Temps de calcul :** Les temps de calcul pour la détermination de points de liaison avec SIFT sont importants. Pour ces différents tests, les calculs ont généralement duré entre 3 et 4 jours, en utilisant un système de calcul distribué sur un pool de machines (cf 12.1.2) ou une seule machine puissante permettant 16 calculs en parallèle. Précisons toutefois que ces temps varient en fonction de la taille et de la configuration du chantier, des paysages (textures comportant plus ou moins de points...) et du paramétrage de SIFT. En effet, le fait de jouer sur certains seuils (plus précisément le seuil de "contraste" et le rééchantillonnage de l'image par l'intermédiaire du paramètre "*maxim*") permet de réduire le nombre de points extraits et donc les temps de calcul.

Par ailleurs, le fait de commencer l'extraction des points à l'octave 0 au lieu de l'octave -1 réduit également les temps de calcul, mais dégrade la qualité de l'aérotriangulation obtenue, qui semble toutefois demeurer acceptable.

En outre, l'utilisation lors de l'appariement d'un prédicteur sur les parties des images en recouvrement (de la manière décrite en 12.1) permet également de réduire le nombre de requêtes et donc les temps de calcul, en particulier pour les paires d'images présentant un faible recouvrement. Les temps de calcul de l'appariements en sont donc réduits (en pratique, à peu près divisés par deux sur un chantier complet).

**Remarque sur le paramétrage de l'extraction de points SIFT :** Dans le cas de ces prises de vue départementales (acquises avec une visée nadir et sans brusques variations de la hauteur de vol), les variations d'échelle entre images en recouvrement sont généralement assez faibles. Par conséquent, des points homoplages auront sensiblement la même échelle.

Il n'est donc pas nécessaire ici d'exploiter complètement le caractère multi-échelle de SIFT et il est possible de limiter l'extraction de points à quelques octaves. Ainsi, lors de l'extraction de points d'intérêt, on pourra donc fixer le paramètre *octaves* à 1 (ou 2) en travaillant à partir des octaves -1 ou 0. Ceci permet de réduire le nombre de points extraits (et donc les temps de calcul à la fois pour l'extraction de points d'intérêts et l'appariement des points homologues), mais aussi d'obtenir des points de meilleure qualité géométrique, en évitant les points extraits à des échelles trop basses, qui auraient tendance à avoir une localisation moins précise (avec un léger "flottement" de leur position d'une image à l'autre).



## 12.3 Prises de vue urbaine - prises de vue haute résolution

Ces prises de vue urbaines ont généralement une résolution de 10 ou 20 cm. Les difficultés qui leur sont spécifiques concernent le bâti : il s'agit en effet de déformations causées par les forts dévers sur le bâti, d'importantes parties cachées ainsi que d'importantes ombres portées.

### 12.3.1 Chantier TPM

Cette prise de vue a été réalisée par plusieurs caméras numériques IGN V1 disposées dans une configuration particulière comprenant une caméra dirigée vers le nadir et deux à visée oblique vers la gauche et la droite. Du fait de cette configuration, dévers et parties cachées pouvaient être très importants.

Quelques paires d'images de ce chantier ont fait l'objet de tests avec SIFT, ne révélant pas de fautes grossières. La répartition des points y est homogène. Des calculs d'orientation ont également été effectués uniquement en relatif sur quelques images.

Les illustrations suivantes (fig. 62, 63 et 64) montrent quelques exemples d'appariements obtenus .

FIGURE 62 – La répartition des points reste homogène.  
en haut, appariement (après filtrage des erreurs) entre une image oblique et une image nadir.  
en bas, appariement entre deux images nadir consécutives après sélection des points (cf 12.1.1)

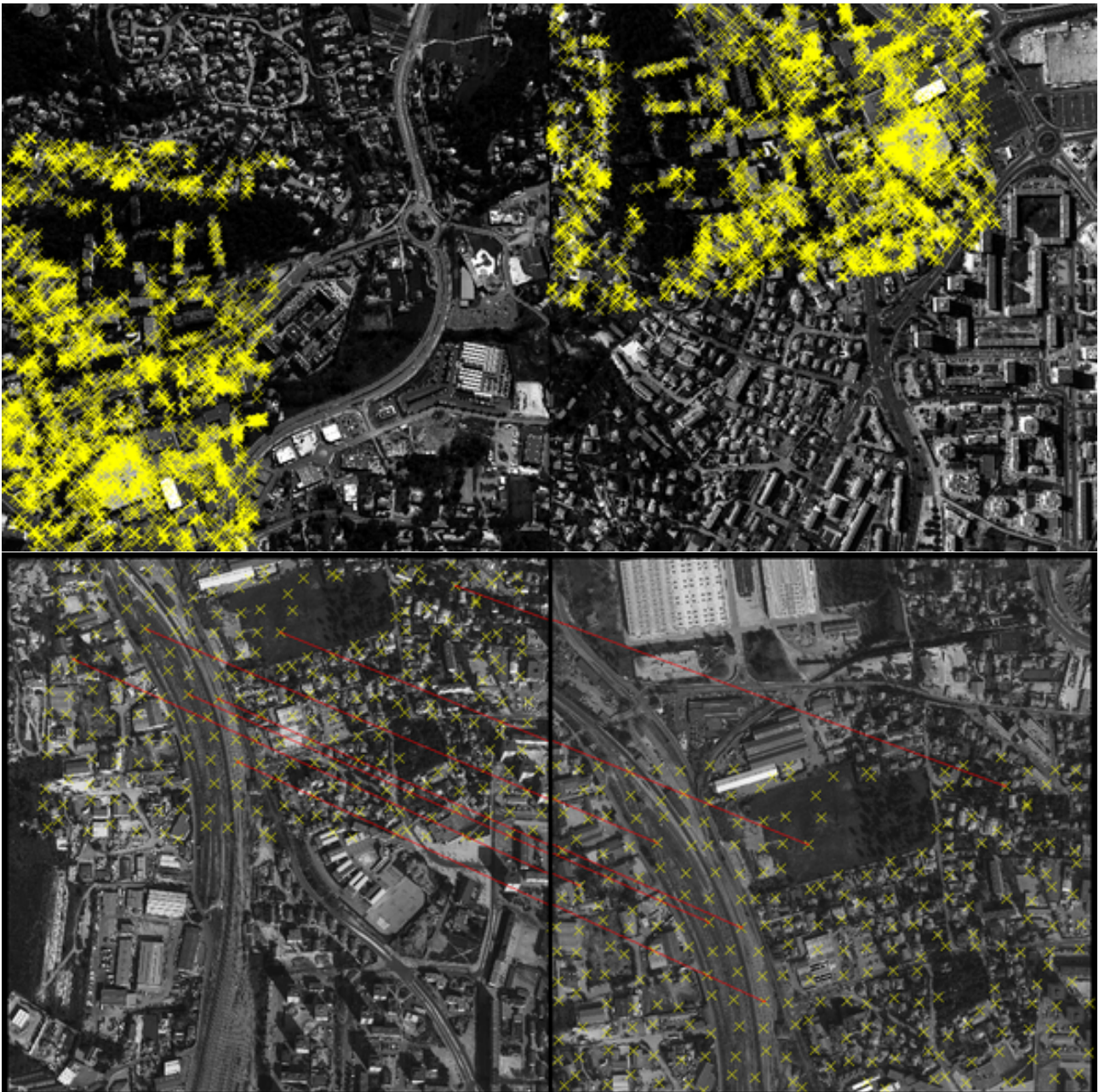


FIGURE 63 – Exemples d'appariements obtenus entre une image nadir et une image oblique

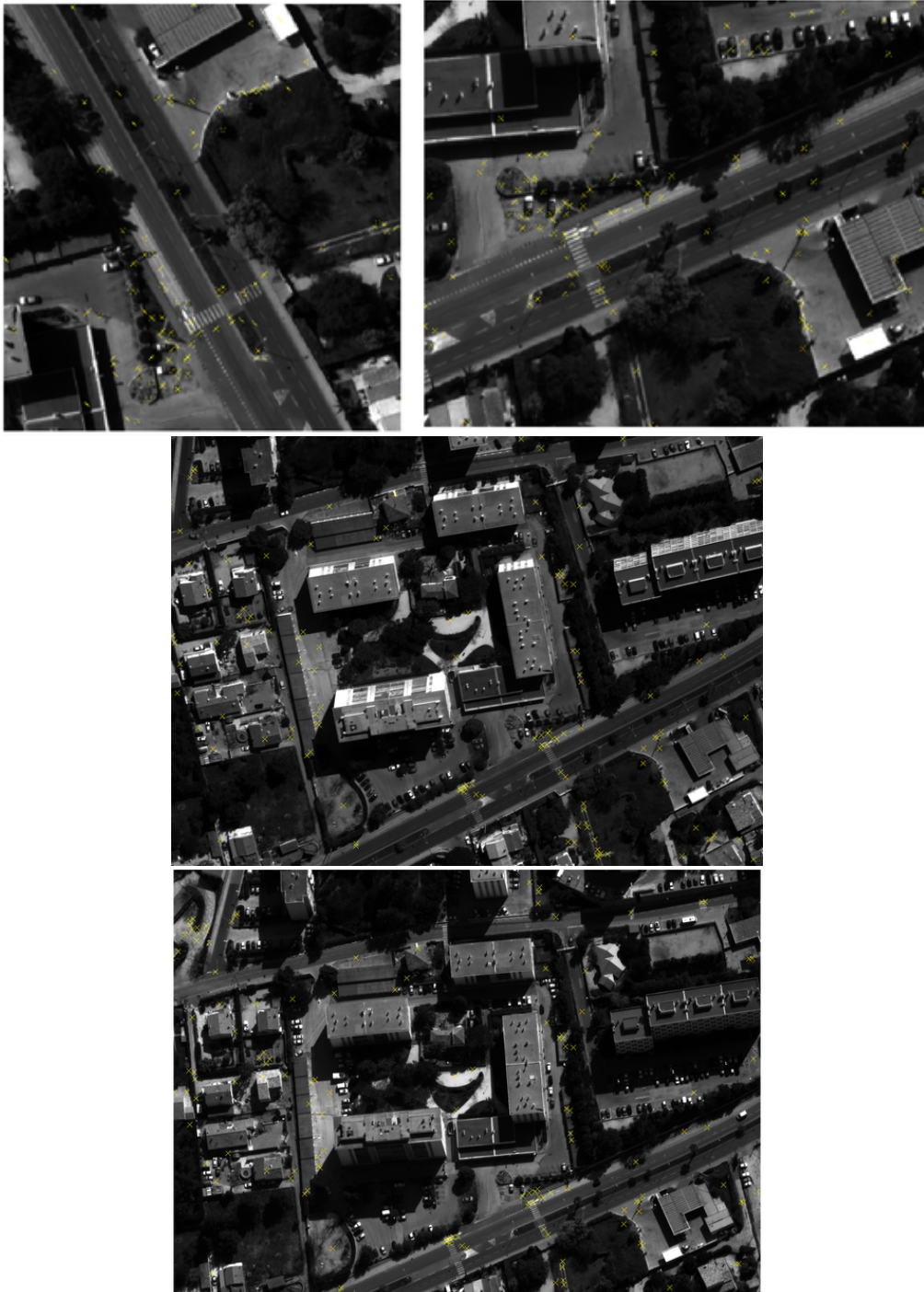
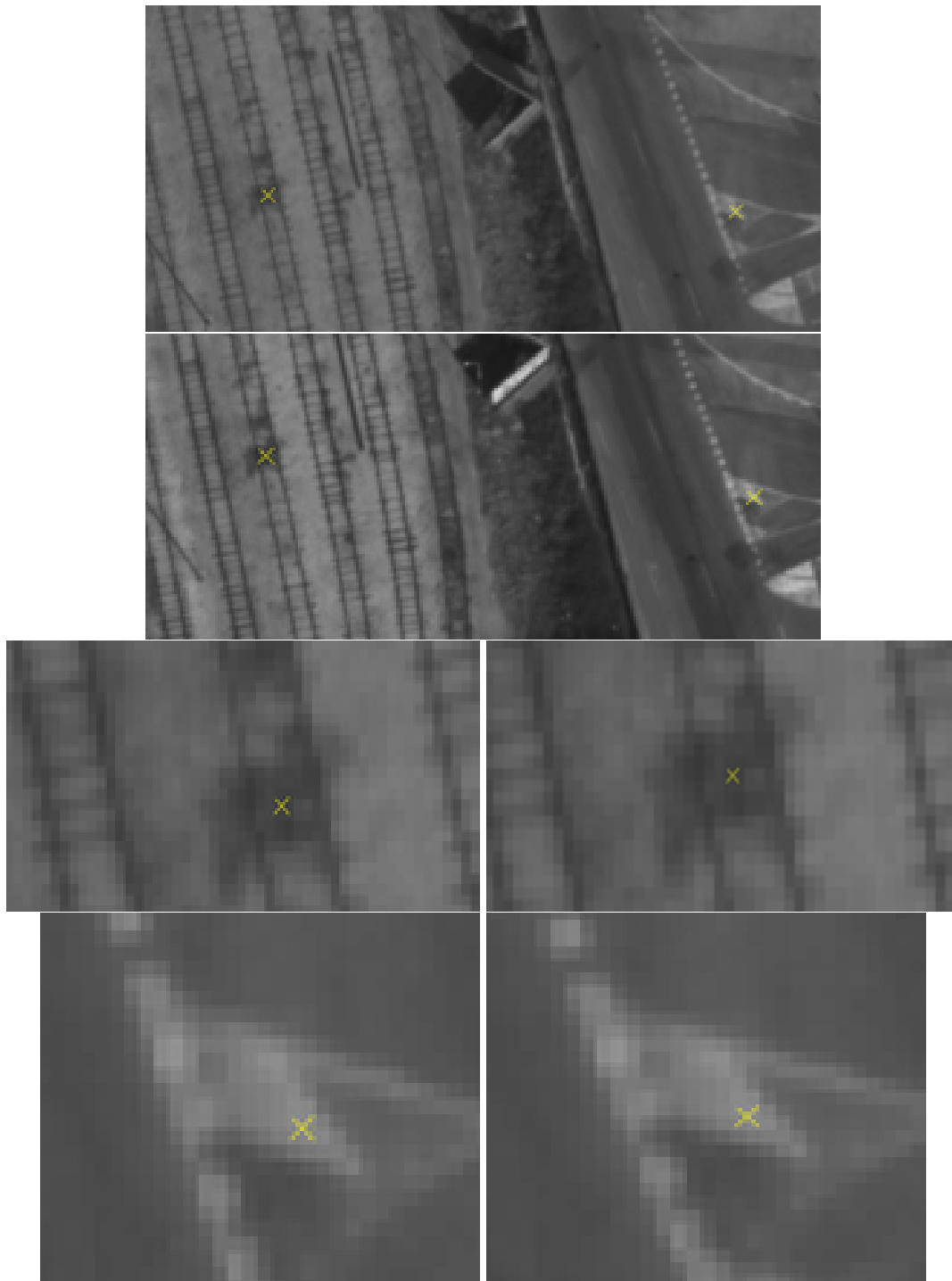


FIGURE 64 – Exemples d'appariements obtenus entre deux images nadirs consécutives



### 12.3.2 Chantier MUSARDE : Prise de vue sur Toulouse

Cette prise de vue urbaine de l'ONERA comporte des clichés de résolution 20cm acquis par deux caméra numérique IGN V1 dans une configuration particulière comprenant une caméra dirigée vers le nadir et une autre à visée oblique vers l'avant. Ce chantier est constitué de plusieurs bandes disposées en "étoile" autour du groupe de bâtiments à modéliser : on ne se trouve donc pas dans la configuration habituelle (bandes parallèles et quelques bandes transverses) des chantiers de prise de vue aérienne.

La détermination de points de liaison entre les images nadirs, ainsi qu'entre images nadir et obliques d'une

même bande se passe plutôt bien.

C'est parfois plus difficile entre images obliques ou entre images nadir et obliques de bandes différentes, en fonction de l'importance du changement de point de vue et du diachronisme (acquisition sur deux journées), ainsi qu'on peut le voir sur les figures 65 et 66. Dans les cas de figures difficiles, les points détectés se situent dans les zones les moins affectées par le changement de point de vue et suffisamment importantes pour permettre l'extraction du point et le calcul d'un descripteur pas trop concerné par le changement de point de vue, comme par exemple des zones de voiries bien dégagées ou de grands toits plutôt plats.

FIGURE 65 – Appariements obtenus entre deux images obliques pour un changement de point de vue limité

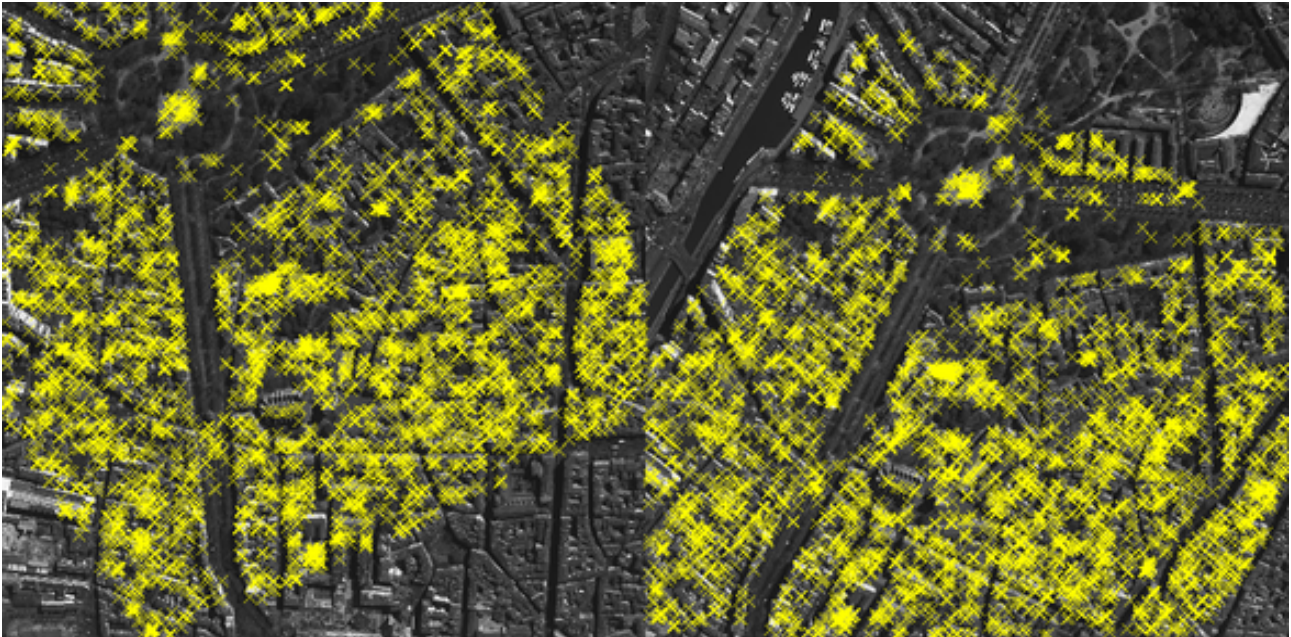
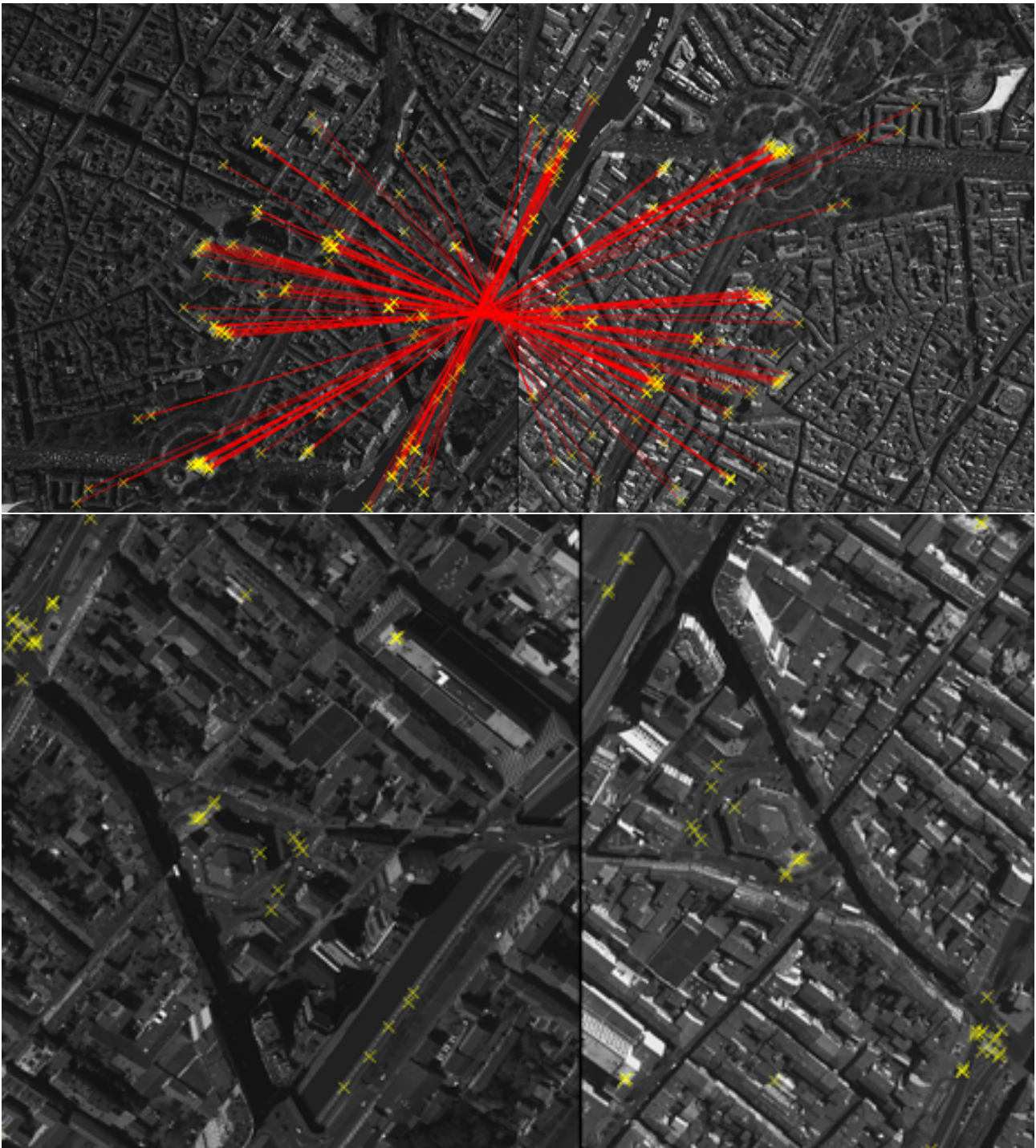


FIGURE 66 – Exemples d'appariements obtenus entre deux images obliques pour un changement de point de vue plus important et un peu de diachronisme



### 12.3.3 Images obliques avec fort changement de point de vue

Seules quelques paires d'images de ce chantier ont fait l'objet de tests. Plusieurs difficultés se cumulaient dans ce cas :

- Forts changements de point de vue
- Variations d'illumination (à la fois liées au diachronisme et à des traitements radiométriques)
- (Bruit plus ou moins important)

Les points homologues obtenus se trouvent très majoritairement au sol lorsqu'il est visible dans les zones où le bâti est le moins dense, c'est-à-dire dans les zones les moins affectées par le changement de point de vue et suffisamment importantes pour permettre l'extraction du point et le calcul d'un descripteur peu concerné par le changement de point de vue. Ainsi que l'illustrent les figures 67, 68 et 69, les résultats obtenus sont variables avec un cas qui se passe relativement bien, un cas où l'appariement échoue et un cas intermédiaire.

FIGURE 67 – Exemples d'appariements obtenus entre deux images : les résultats sont meilleurs dans les zones où le bâti est le moins dense (et où les différences causées à la fois par le diachronisme et le changement de point de vue sont moindres)



FIGURE 68 – Exemples d'appariements : les résultats sont meilleurs dans les zones où le bâti est le moins dense (et où les différences causées à la fois par le diachronisme et le changement de point de vue sont moindres)

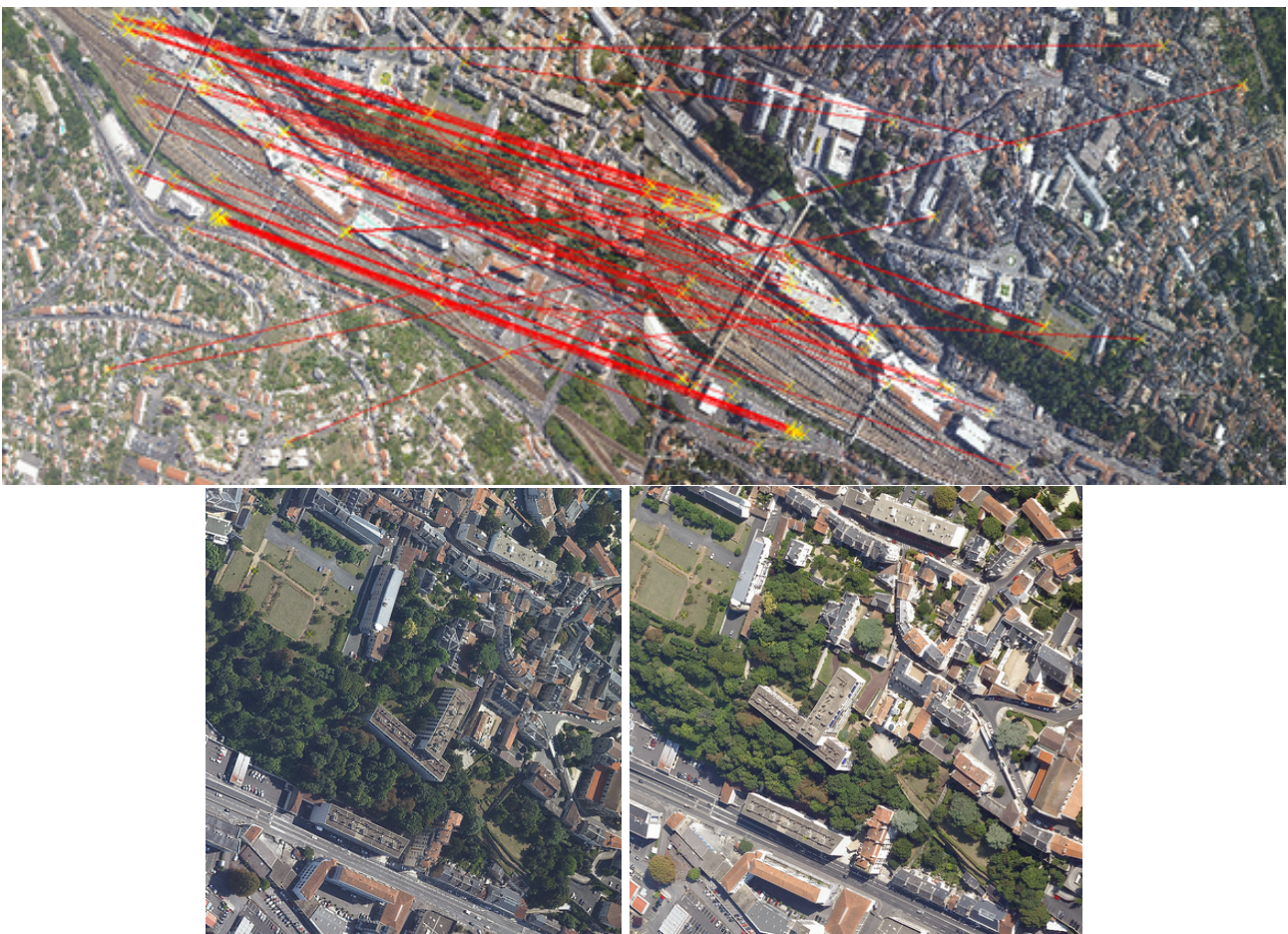




FIGURE 69 – Aucun appariement correct entre ces deux images : le changement de point de vue “3D” est trop important



### 12.3.4 Marseille à 10cm

Une seule paire d'images de cette prise de vue à 10cm effectuée par la caméra Zeiss DMC a fait l'objet d'un test au SBI. Il s'agissait notamment ici de voir le comportement de SIFT sur ces images de grande dimension (7680x13824 pixels) acquises avec une caméra à large fauchée, et par conséquent présentant des parties cachées importantes en zone urbaine dense.

Tout comme précédemment, la répartition des points détectés est à peu près homogène.

Par ailleurs, les mises en correspondance semblent globalement correctes. Sur ce chantier, près de 2/3 des points de liaison fournis par TopAero avaient été éliminés au cours de l'aérotriangulation ; ils correspondaient en effet à des parties cachées, des ombres et des détails répétitifs comme par exemple la texture d'un toit.

FIGURE 70 – Exemples de mises en correspondance.

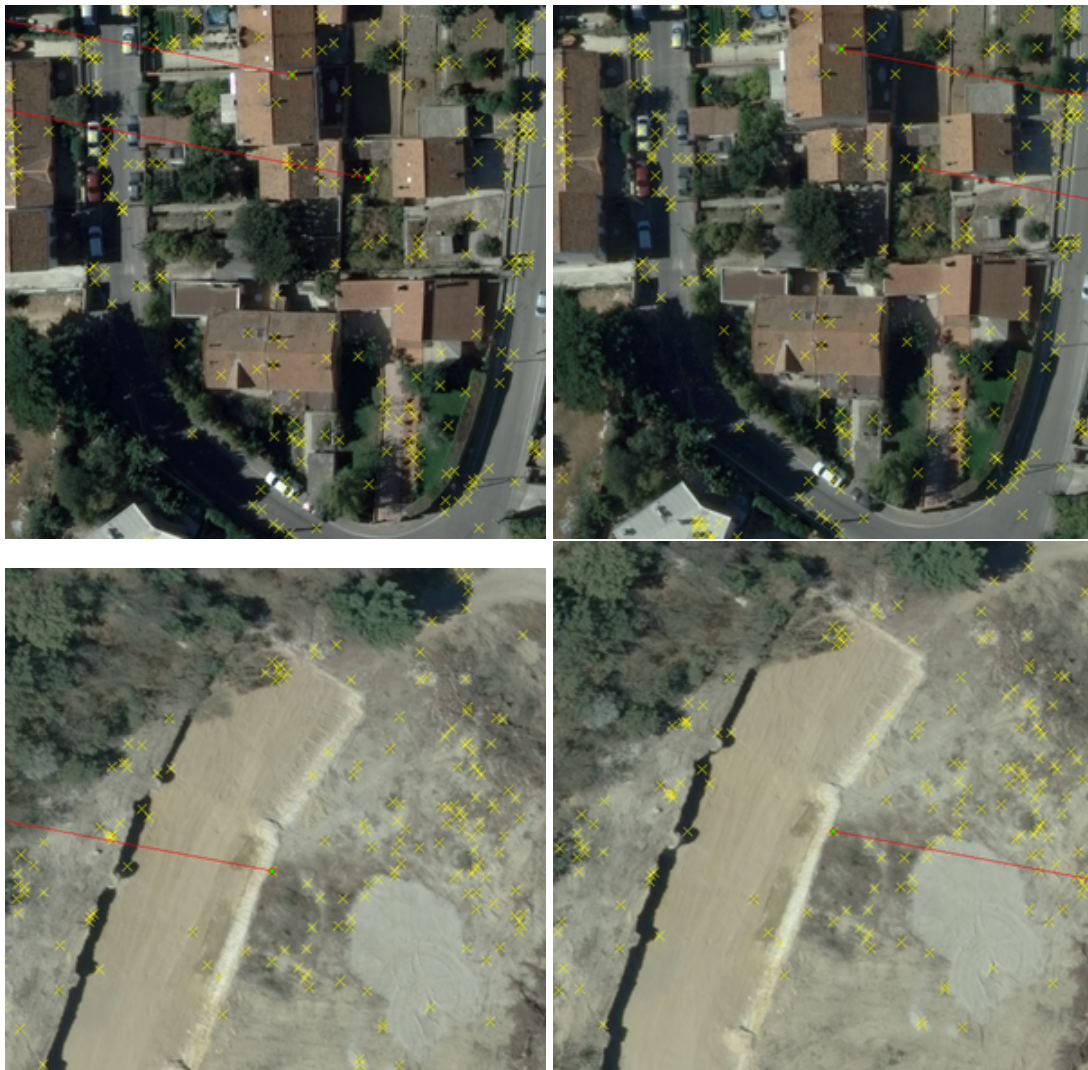
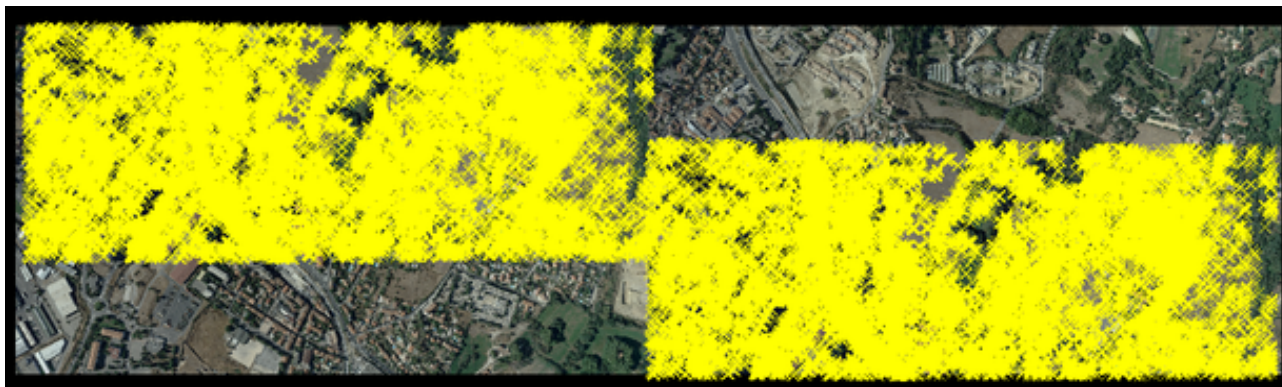
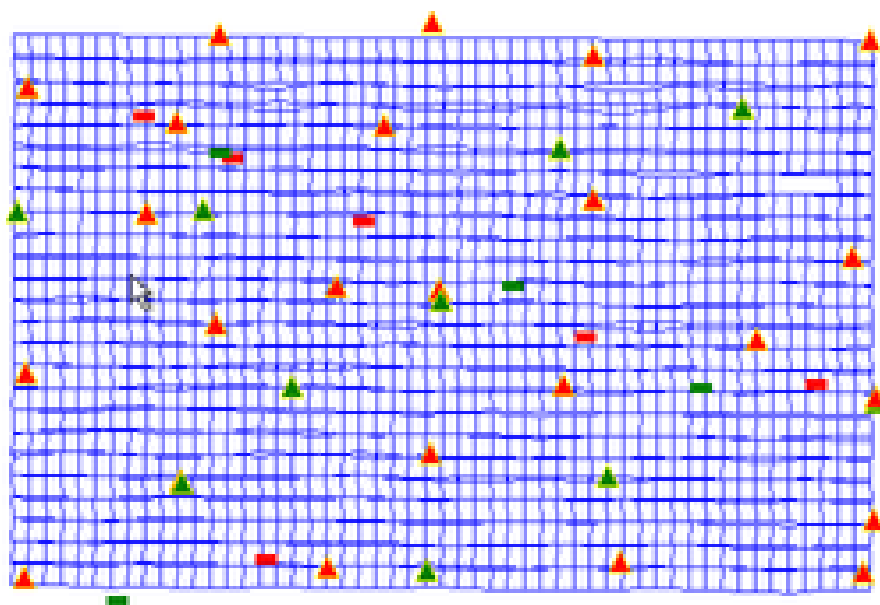


FIGURE 71 – La répartition des points reste homogène.



### 12.3.5 Paris à 10 cm



Il s'agissait tout comme au 12.3.4 d'une prise de vue à 10cm réalisée par la caméra Zeiss DMC. Cette fois, des points SIFT ont été extraits sur l'ensemble du chantier et une aérotriangulation a été calculée. Un fort résidu sur la coordonnée X des mesures GPS du sommet de prise de vue est observé (de l'ordre de 20 cm). Les résidus sur les points terrain restent eux-aussi initialement anormalement élevés en Z mais sont bien réduits en activant le systématisme cliché "Résidus radiaux" proposé par TopAéro. (voir tab. 10 et 11).

**Remarque :** En relâchant certaines contraintes lors de l'aérotriangulation, cette tendance est également observée en utilisant des points de liaison extraits par les outils utilisés en production.

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)								
	X	Y	XY						
tous	0.31	0.25	0.40	Appui	0.07	0.07	0.28	0.10	0.30
inter-bandes	0.33	0.25	0.41	Contrôle	0.06	0.06	0.15	0.09	0.17
				GPS	0.20	0.07	0.06	0.21	0.22

TABLE 11 – EMQ de l'aérotriangulation pour des points de liaison fournis par SIFT avec systématisme cliché "Résidus radiaux"

	EMQ Mesures image (en pixels)			EMQ Points terrain (en m)					
	X	Y	XY		X	Y	Z	XY	XYZ
tous	0.30	0.24	0.39	Appui	0.08	0.08	0.14	0.11	0.18
inter-bandes	0.31	0.24	0.39	Controle	0.07	0.05	0.11	0.09	0.14
				GPS	0.20	0.06	0.05	0.21	0.20

## 12.4 Extraction de points de liaison dans des cas anecdotiques

### 12.4.1 Calcul de points de liaison entre images aériennes floues pour l'aérotriangulation

Il s'agissait ici d'un chantier aérien (prise de vue à 20 cm acquise par la caméra Applanix DSS322) dans lequel les images présentaient des problèmes de flou. Ce flou variait au sein de l'image : il était en effet fort en haut à gauche de l'image puis diminuait progressivement jusqu'à disparaître avant le centre de l'image.

Si les résultats du calcul d'aérotriangulation effectué sur ce chantier en utilisant PtLiaison étaient acceptables (respect des spécifications atteint), la restitution stéréoscopique s'avérait très difficile voire impossible du fait de l'existence de sauts altimétriques entre bandes (ces sauts pouvant atteindre jusqu'à 1m).

Il était intéressant d'exploiter la robustesse et le caractère de détecteur (et de descripteur invariant) multi-échelle de SIFT dans un tel cas de figure. Un test a donc été effectué sur sous-bloc comportant 102 clichés. Des exemples de points de liaison obtenus sont visibles fig. 72.

Les EMQ du calcul d'aérotriangulation ne sont pas à première vue pas meilleures avec les points de liaison fournis par SIFT qu'avec ceux fournis par PTLiaison. Néanmoins, d'un point de vue plus qualitatif, on constate une amélioration : la restitution stéréoscopique devient en effet possible en utilisant les orientations des clichés fournies par ce nouveau calcul, les sauts altimétriques en interbande ayant considérablement diminué voire disparu.

Par ailleurs, par la suite, l'application du systématisme cliché "Affinité" de TopAéro a permis de considérablement réduire les résidus de l'aérotriangulation à la fois pour le calcul utilisant des points de liaison fournis par SIFT et pour celui utilisant des points de liaison fournis par PTLiaison. Les résultats obtenus en utilisant ce systématisme autorisaient également dans les deux cas la restitution stéréoscopique (disparition des sauts altimétriques entre bandes).

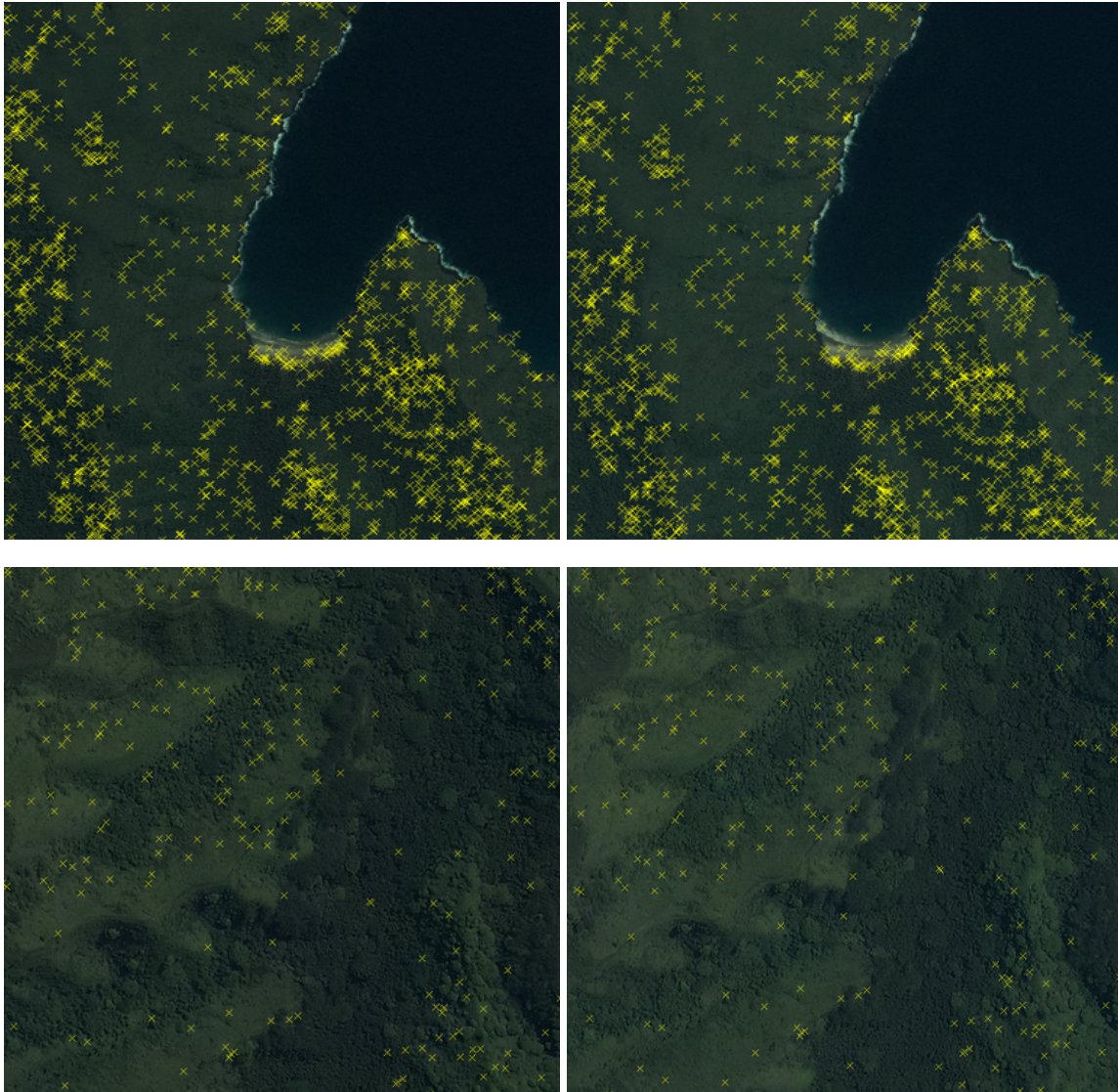
FIGURE 72 – Deux exemples de points de liaison entre images nette (en haut) et floue (en bas)



### 12.4.2 Calcul de points de liaison entre images satellites pour la spatiotriangulation

Un test a été mené de manière un peu anecdotique à ce sujet. Il a porté sur des images QuickBird acquises sensiblement au même moment afin d'éviter les problèmes de diachronisme déjà rencontrés avec les chantiers aériens classiques. Des exemples de résultats obtenus sont donnés dans la fig. 73.

FIGURE 73 – Exemples de points de liaison entre images du satellite QuickBird



## 13 Constitution automatique d'un tableau d'assemblage

### 13.1 Présentation du problème et méthodes mises en œuvre

On se trouve ici dans le cas où l'on ne dispose *a priori* ni d'un tableau d'assemblage, ni d'une orientation approchée des différentes images d'un chantier.

Dans les deux chantiers traités dans cette partie, l'objectif final est d'obtenir des points de liaison destinés à un calcul d'aérotriangulation. Or, sans tableau d'assemblage approché des clichés, il n'est pas possible de savoir quelles sont les images voisines et lancer la mise en correspondance de points SIFT sur toutes les paires possibles serait très long et surtout inutile.

La solution retenue consiste donc à dans un premier temps reconstituer le tableau d'assemblage (du moins le graphe de voisinage des clichés, c'est-à-dire savoir quel cliché est voisin de quel autre). Lors de cette première étape, le travail peut s'effectuer à basse résolution afin de réduire fortement les temps de calcul. Deux méthodes distinctes ont été mises en œuvre.

#### 13.1.1 Méthode 1 : “recherche exhaustive”

Cette première méthode est la suivante :

##### 1. Construction du graphe de voisinage

- (a) Extraction des points d'intérêt et de leur descripteur dans des images sous-échantillonnées. D'un point de vue pratique, ceci peut se faire soit en sous-échantillonnant l'image puis en lançant l'extraction des points d'intérêt à pleine résolution sur l'image sous-échantillonnée obtenue, soit en travaillant directement sur l'image originale mais en commençant le calcul des points d'intérêt uniquement à partir d'une certaine octave (Les deux méthodes sont à peu près équivalentes.)
- (b) Pour tous les couples possibles, appariement des points homologues parmi les points extraits à l'étape précédente.
- (c) On considère que deux clichés sont en recouvrement s'ils possèdent plus d'un certain nombre  $n_{min}$  de points homologues. On obtient ainsi une liste des paires d'images pour lesquels on va ensuite rechercher des points de liaison nécessaires au calcul d'aérotriangulation. On dispose dès lors d'un tableau d'assemblage (approché), ou du moins d'un graphe de voisinage du chantier.

2. **Calcul de points de liaison** : On lance à pleine résolution le calcul des points de liaison comme pour les chantiers aériens classiques uniquement pour les couples identifiés à l'étape précédente. On peut ensuite calculer l'aérotriangulation.

#### 13.1.2 Méthode 2 : “recherche dans une base de données de points SIFT”

Cette seconde méthode (plus rapide que la précédente) est la suivante :

##### 1. Construction du graphe de voisinage

- (a) Extraction des points d'intérêt et de leur descripteur pour ces images sous-échantillonnées. D'un point de vue pratique, ceci peut se faire soit en sous-échantillonnant l'image puis en lançant l'extraction des points d'intérêt à pleine résolution sur l'image sous-échantillonnée obtenue, soit en travaillant directement sur l'image originale mais en commençant le calcul des points d'intérêt uniquement à partir d'une certaine octave (Les deux méthodes sont à peu près équivalentes.)
- (b) On recherche alors pour chaque cliché quels sont les  $nb_{cliches\ homologues}$  clichés les plus susceptibles d'être ses voisins. L'algorithme mis en œuvre est le suivant :

---

**Notations :**

On note  $\mathcal{I}$  l'ensemble des images du chantier.

Soit  $d$  un point SIFT extrait d'une image.

$d.sift$  désigne le descripteur SIFT associé à  $d$ .

$d.image$  désigne l'image dont  $d$  a été extrait.

On note  $\mathcal{D}(i) = \{d/d.image = i\}$  l'ensemble des points SIFT extraits de l'image  $i$ .

On note  $\mathcal{D} = \cup_{i \in \mathcal{I}} \mathcal{D}(i)$  l'ensemble des points SIFT du chantier.

**Algorithme :**

**Rangement des éléments de  $\mathcal{D}$  dans un k-d-tree construit dans l'espace des descripteurs SIFT pour tout  $i \in \mathcal{I}$  faire**

  //Initialisation du score de "recouvrement" de  $i$  avec les autres images du chantiers.

**pour tout**  $j \in \mathcal{I}$  **faire**

$score[j] \leftarrow 0$

**fin pour**

  //Calcul du score de "recouvrement" de  $i$  avec les autres images du chantiers.

**pour tout**  $d \in \mathcal{D}(i)$  **faire**

    Recherche des  $nppv$  plus proches voisins de  $d$  dans  $\mathcal{D}$  (au sens de la distance euclidienne dans l'espace des descripteurs SIFT). Cette recherche s'effectue de manière efficace à l'aide du k-d-tree préalablement construit. La méthode est la même que pour la procédure d'appariement standard des points SIFT décrite en section 1.3 (i.e. en s'arrêtant au bout d'un certain nombre de points testés.)

    Notons  $ppv(d)$  l'ensemble des  $nppv$  plus proche voisin de  $d$

**pour tout**  $s \in ppv(d)$  **faire**

$score[s.image] \leftarrow score[s.image] + 1$

**fin pour**

**fin pour**

  Les images correspondant aux  $nb_{clicheshomologues}$  meilleurs scores sont considérées comme étant en recouvrement avec l'image  $i$ . Les autres, non.

**fin pour**

---



- (c) On dispose d'un tableau d'assemblage (approché), ou du moins d'un graphe de voisinage du chantier.
2. **Calcul de points de liaison** : On lance à pleine résolution le calcul des points de liaison comme pour les chantiers aériens classiques uniquement pour les couples identifiés à l'étape précédente. On peut ensuite calculer l'aérotiangulation.

## 13.2 Prise de vue aérienne sur Toulouse

On souhaite calculer l'aérotiangulation d'un chantier aérien "traditionnel" (prise de vue "nadir") pour lequel on ne dispose pas de tableau d'assemblage. Le chantier comporte 480 clichés (de dimension 4106x4096 et de résolution 20cm) acquis par la caméra numérique V1 de l'IGN.

### 13.2.1 Construction du graphe de voisinage

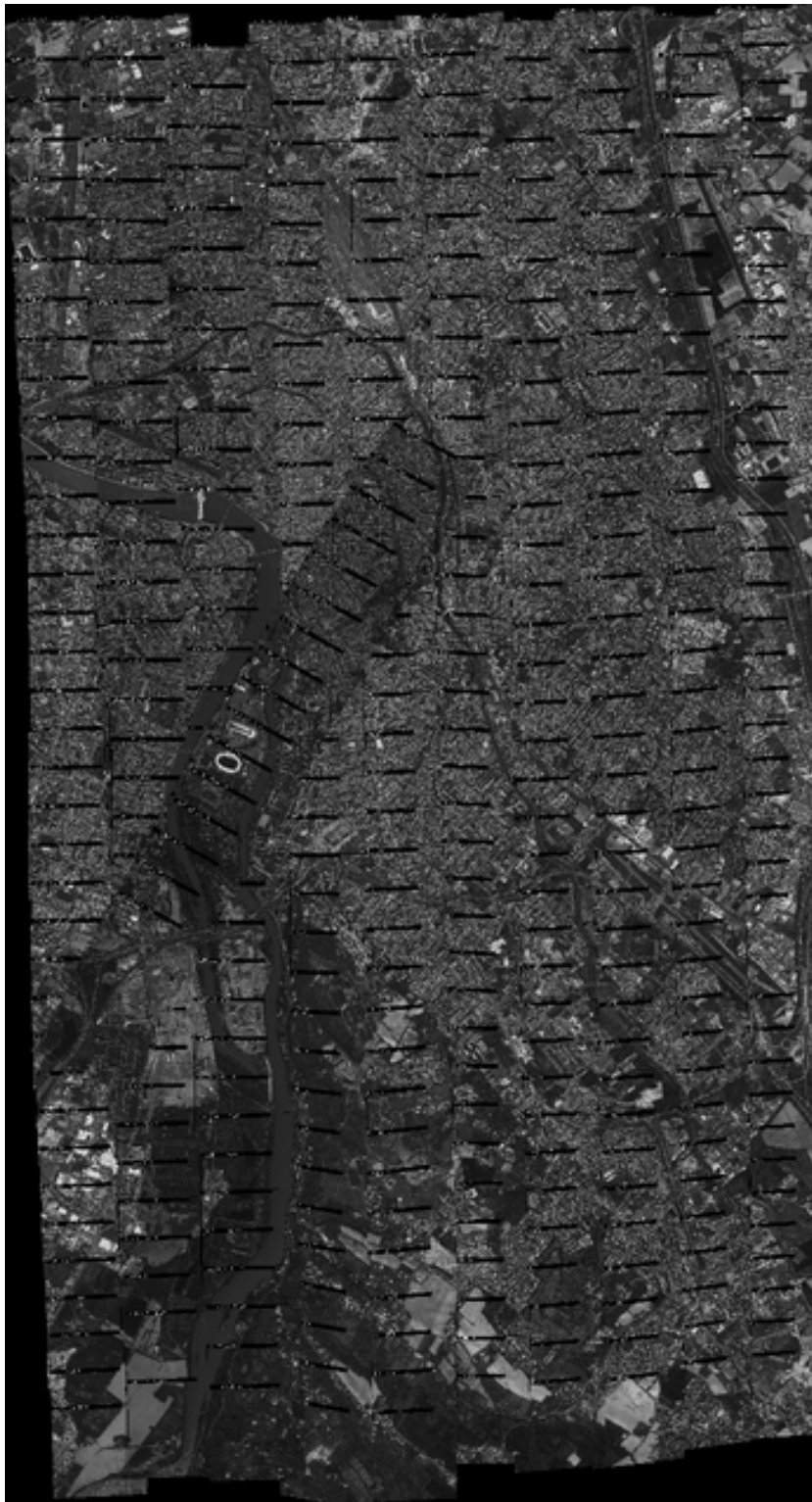
Il s'agit ici de déterminer quels couples d'image présentent un recouvrement. La première méthode ("Recherche exhaustive") est celle qui a été mise en œuvre dans ce premier cas : comme expliqué précédemment, ce travail s'effectue sur des images sous-échantillonnées (ici d'un facteur 8). À l'issue de cette étape, on connaît les paires de clichés pour lesquelles il va être possible d'extraire des points de liaison.

Afin d'illustrer ce résultat, des "pseudo-orthoimages" sont visibles sur les figures 74 et 75. Chaque image y est géoréférencée par une similitude 2D (par rapport à un cliché de référence) calculées à partir des points homologues extraits.

FIGURE 74 – Extrait d'une "pseudo-ortho" obtenue directement à partir du tableau d'assemblage calculé



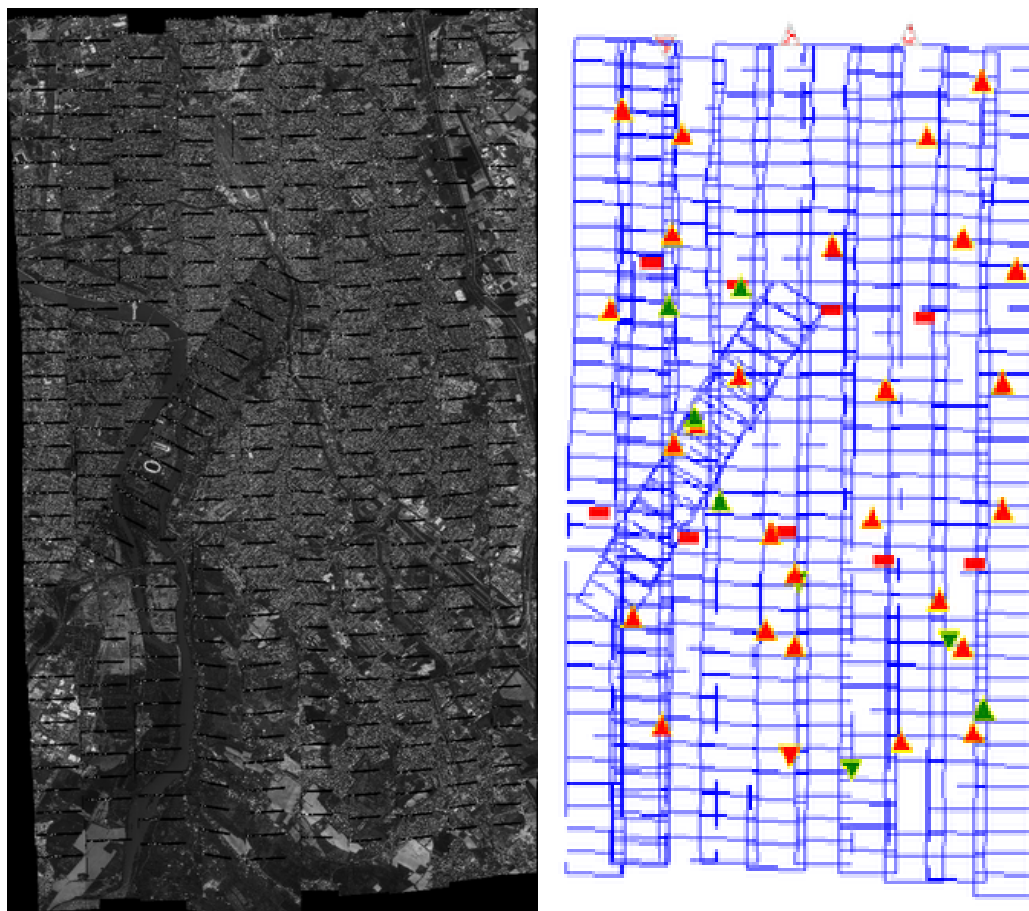
FIGURE 75 – “Pseudo-ortho” obtenue directement à partir du tableau d'assemblage calculé



### 13.2.2 Extraction des points de liaison et aérotriangulation

Aucune difficulté n'a ensuite été rencontrée pour l'extraction des points de liaison (chantier favorable, diachronisme très faible). Le calcul d'aérotriangulation comportant à la fois une mise en place approchée des clichés (en l'absence d'orientations approchées) et une compensation uniquement à partir des points de liaison détectés

FIGURE 76 – Equipement du chantier d'aérotriangulation en points d'appui



et de points d'appui (pas de mesures GPS des sommets de prise de vue) s'est également déroulé sans problème. (La répartition des points terrain est visible sur la figure 76.)

Les EMQ obtenues (environ 1m en X, 1m en Y et 2m en Z) peuvent à première vue sembler un peu fortes. Il faut toutefois préciser que l'étalonnage de la caméra utilisée n'était pas disponible : le calcul a donc été effectué en supposant que le point principal d'autocollimation et le point principal de symétrie se trouvent au centre de l'image et que la distorsion radiale est nulle. On peut d'ailleurs observer que les EMQ diminuent fortement (0.5m, 0.4m en X et en Y, et 0.5m en Z) si la compensation est effectuée avec le systématisme cliché "Résidus radiaux + Déformation du film" proposé par TopAéro.

Les résultats de ces calculs d'aérotriangulations sont visibles dans les tableaux 12 et 13.

TABLE 12 – EMQ de l'aérotriangulation

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)								
	X	Y	XY	X	Y	Z	XY	XYZ	
tous	0.23	0.25	0.34	Appui	0.93	1.26	2.18	1.56	2.69
inter-bandes	0.25	0.25	0.36	Contrôle	0.83	0.99	1.81	1.29	2.22

TABLE 13 – EMQ de l'aérotriangulation avec systématismes

EMQ Mesures image (en pixels)	EMQ Points terrain (en m)		
	X	Y	XY
tous	0.20	0.22	0.30
inter-bandes	0.23	0.22	0.32

	EMQ Points terrain (en m)				
	X	Y	Z	XY	XYZ
Appui	0.55	0.34	0.51	0.65	0.82
Controle	0.34	0.42	0.54	0.54	0.77

### 13.3 Chantier de l'Île aux Cochons

#### 13.3.1 Présentation du chantier

FIGURE 77 – Tableau d'assemblage papier du chantier



Cette prise de vue “historique” date de 1982 : les images à traiter sont donc cette fois des clichés argentiques scannés.

On notera au passage l'anecdote suivante : la carte de l'Île aux Cochons comprend des zones non cartographiées comportant la seule mention “nuages”. En effet, en raison de la fréquence importante de nuages orographiques sur les hauteurs, il s'est avéré impossible de réaliser des photographies aériennes permettant de voir le sol de l'ensemble de l'île dans le temps imparti à la mission.

Ce petit chantier réunit en outre de nombreuses difficultés :

- Il n'y a pas de tableau d'assemblage numérique (comportant au moins le géoréférencement approché des différentes images). Les seuls documents disponibles sur ce chantier sont un tableau d'assemblage papier (voir figure 77) accompagné d'un carnet de stéréopréparation ; le géoréférencement des clichés (calculé à l'époque de la prise de vue) n'est plus disponible.
- Les images ont été acquises depuis un hélicoptère, avec pour conséquence des variations d'échelle et des rotations plus importantes que dans les chantiers aériens classiques (avec prise de vue depuis un avion).
- Lors de leur numérisation, les clichés (mal posés sur le scanner) ont souvent subi une rotation (2D) supplémentaire.

- Le relief de cette île volcanique est assez fort : on part en effet du niveau de la mer pour atteindre une altitude de 770m au niveau du Mont Richard-Foy (point culminant de l'île), avec pour conséquence des variations d'échelle importantes sur les images.
- De nombreuses images sont en grande partie couvertes par la mer (zones d'écumes, vagues...).
- Des zones de nuages sont présentes sur bon nombre d'images (au point de les couvrir presque totalement pour certaines d'entre elles).
- De grandes taches blanches apparaissent sur certaines des images. (Ce phénomène est dû à un problème optique lors de la prise de vue).
- Les images sont bruitées.
- Peu de détails auxquels il serait possible de se raccrocher apparaissent (même pour un opérateur).

### 13.3.2 Méthode et résultats

La méthode mise en œuvre est la même que dans le cas précédent (cf 13.2), à savoir extraction de points de Lowe à basse résolution, mise en correspondance de ces points de liaison pour toutes les paires de clichés possibles et construction du graphe de voisinage, suivies d'un calcul des points de liaison à pleine résolution.

Au terme des calculs, on constate que SIFT parvient malgré tout à fournir un certain nombre de points de liaison entre les clichés, même dans des conditions difficiles (voir figures 78, 79 et 80).

Néanmoins, le nombre et la répartition des points de liaison extraits varient fortement d'un couple à l'autre (pour un recouvrement équivalent). Ainsi, sur certaines zones d'une image, on va obtenir une bonne densité de points, tandis que dans d'autres parties, aucun point ne sera détecté.

On remarque que peu de points sont détectés dans les zones de nuages et qu'aucun appariement correct n'y est extrait. (Les appariements erronés qui ont pu s'y produire peuvent alors être détectés et éliminés.)

Une "pseudo-orthoimage" (figures 81 et 82) a ensuite été calculée à partir du tableau d'assemblage approché obtenu. Le "décrochage" qu'on peut y voir à droite provient principalement de la méthode de calcul "expédiée" utilisée pour la générer qui conduit à une accumulation d'erreurs sur l'orientation (qui n'est elle-même qu'une simple similitude 2D par rapport à un cliché de référence) des clichés ; il n'y a pas eu de compensation globale du chantier. En outre, le modèle de similitude 2D n'est pas nécessairement le plus adapté ici.

Étant donnée la disposition des points obtenus, le calcul d'aérottriangulation s'avère difficile : quelques clichés (sur lesquels le sol est en grande partie masqué par la couverture nuageuse) ne sont "rattachés" au reste de la prise de vue que par un nombre insuffisant de points (voir figures 81 et 80). Il serait bon de saisir manuellement quelques points complémentaires.

FIGURE 78 – Exemples de points homologues détectés (vue d'ensemble et détails)

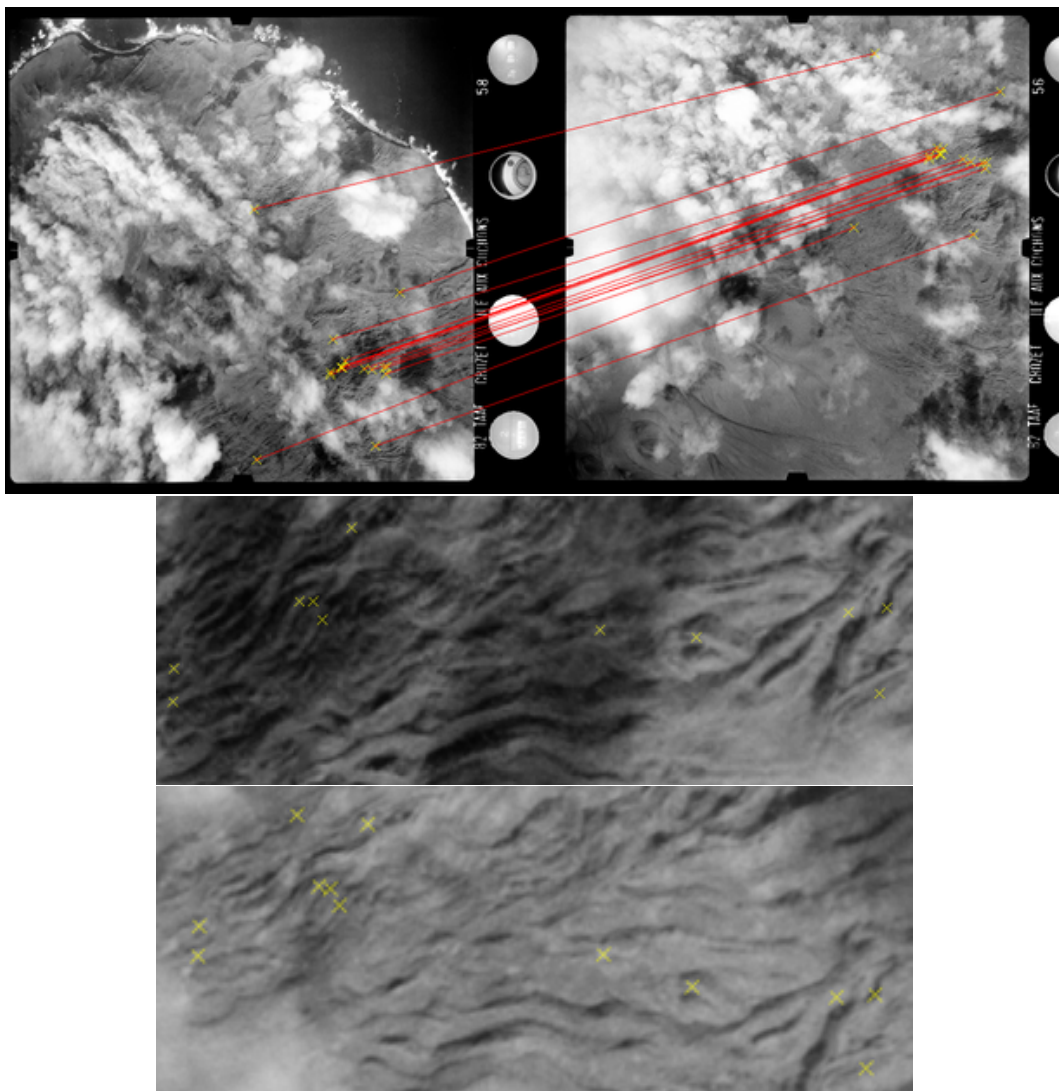


FIGURE 79 – Exemples de points homologues détectés

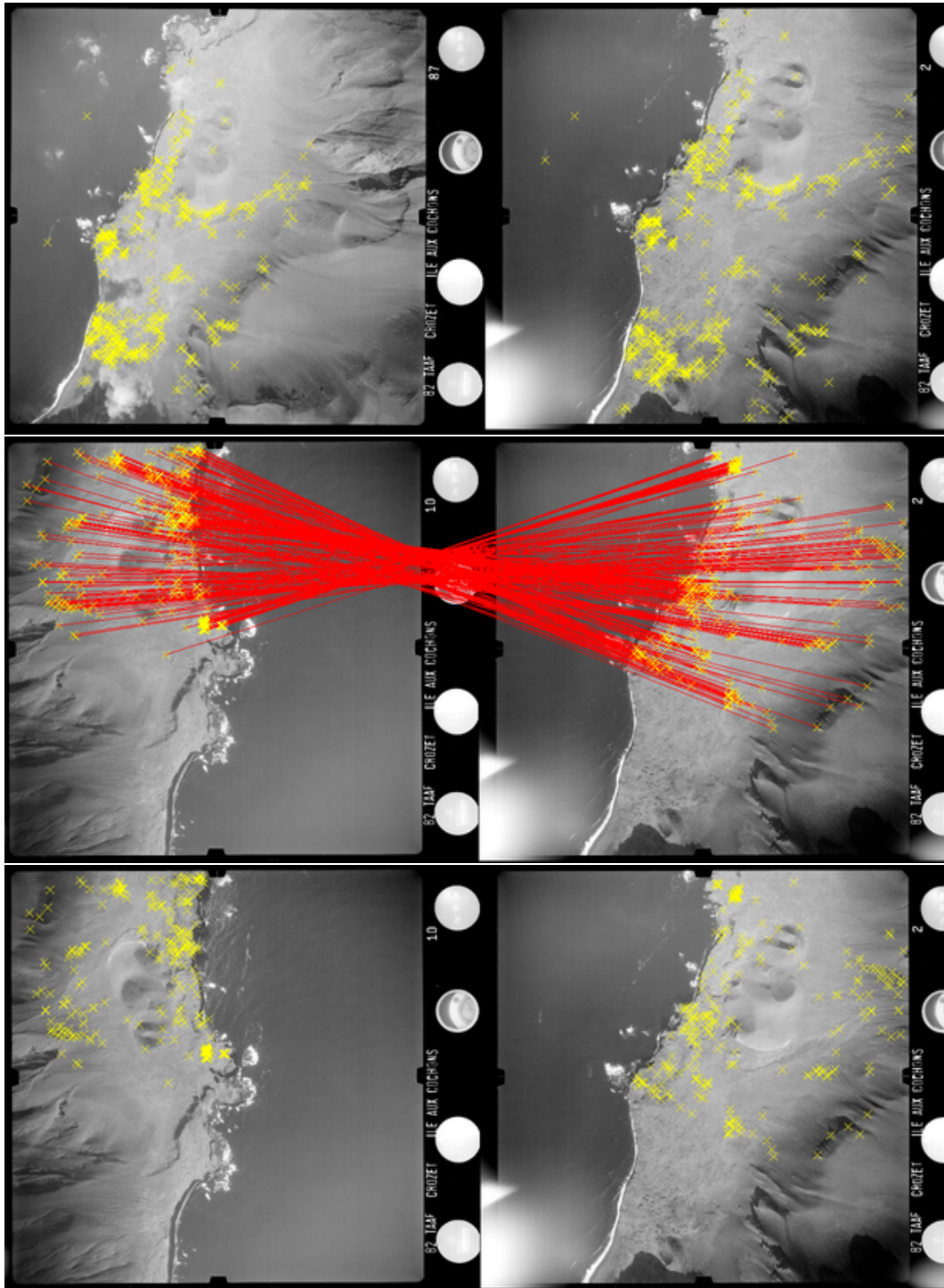


FIGURE 80 – Exemples de points homologues détectés (malgré les nuages)

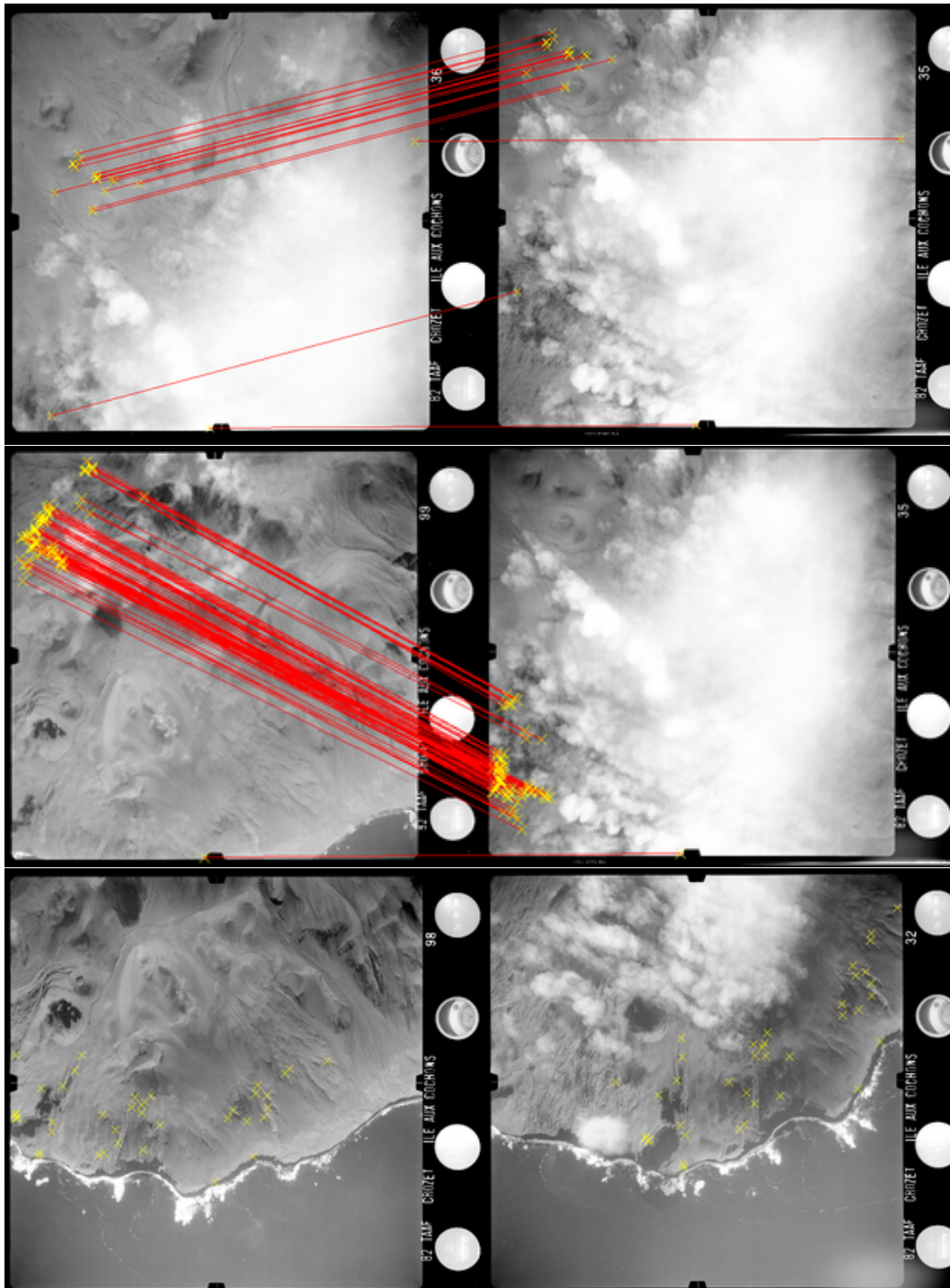




FIGURE 81 – “Pseudo-orthoimage”

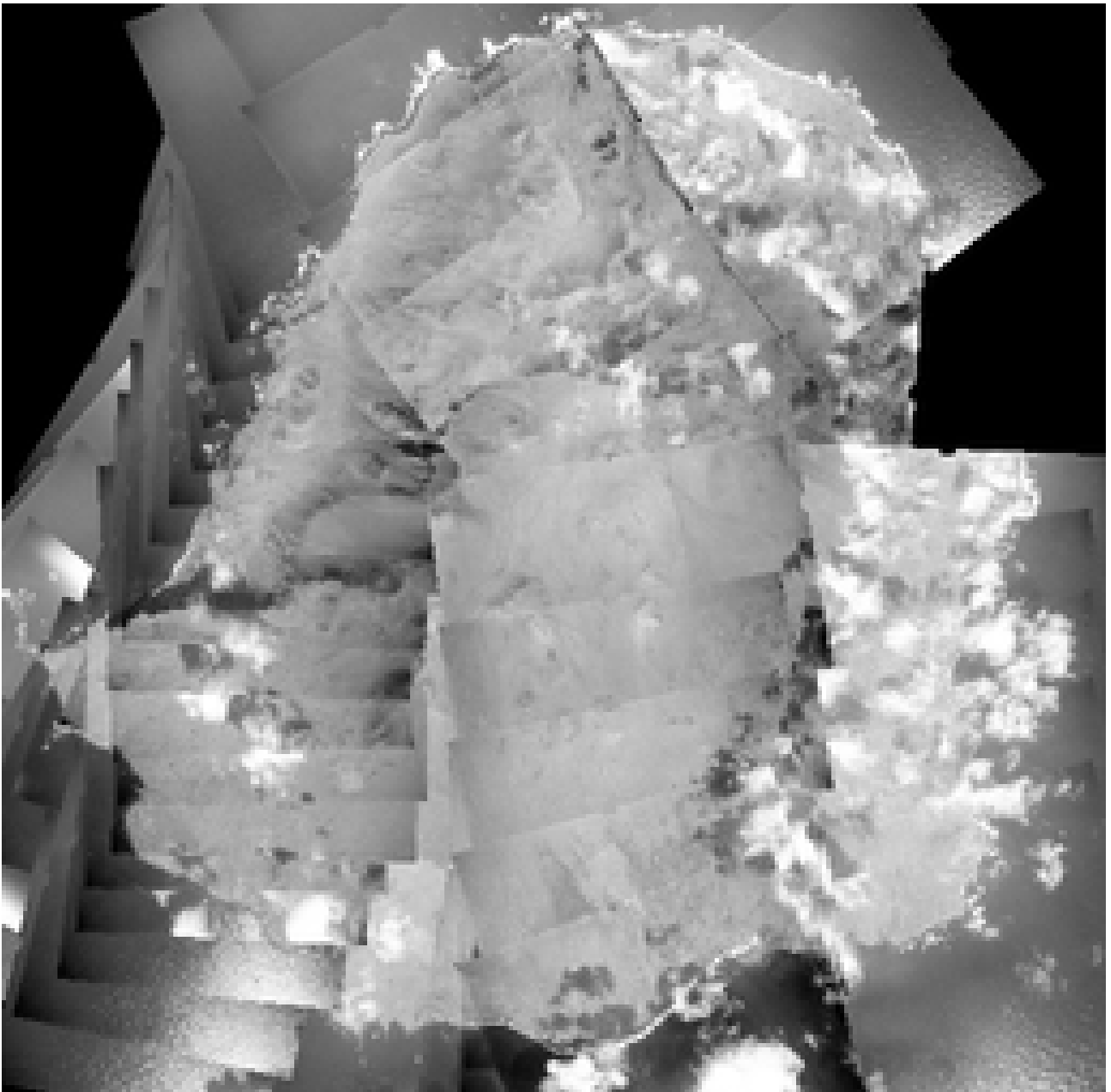
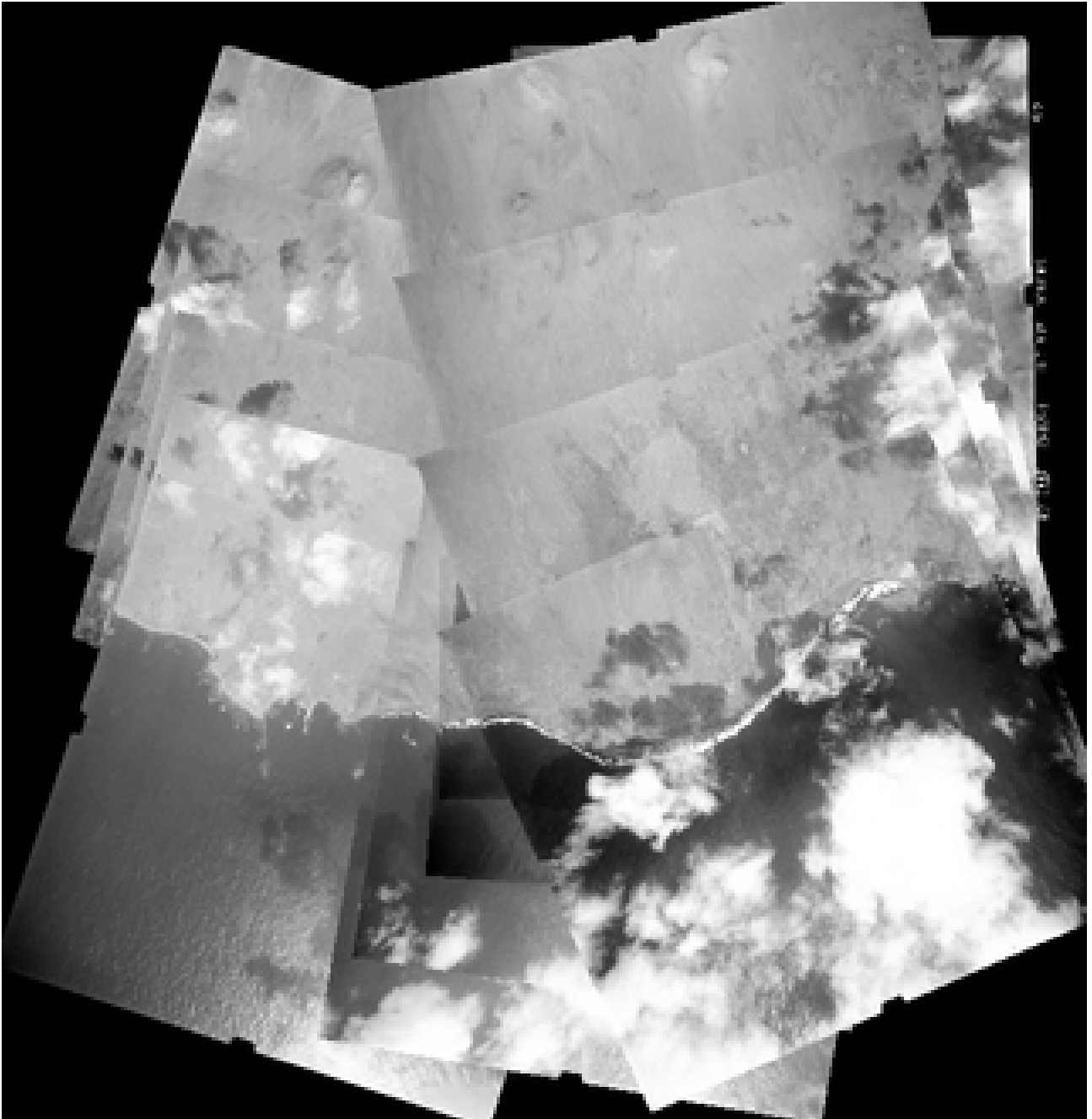
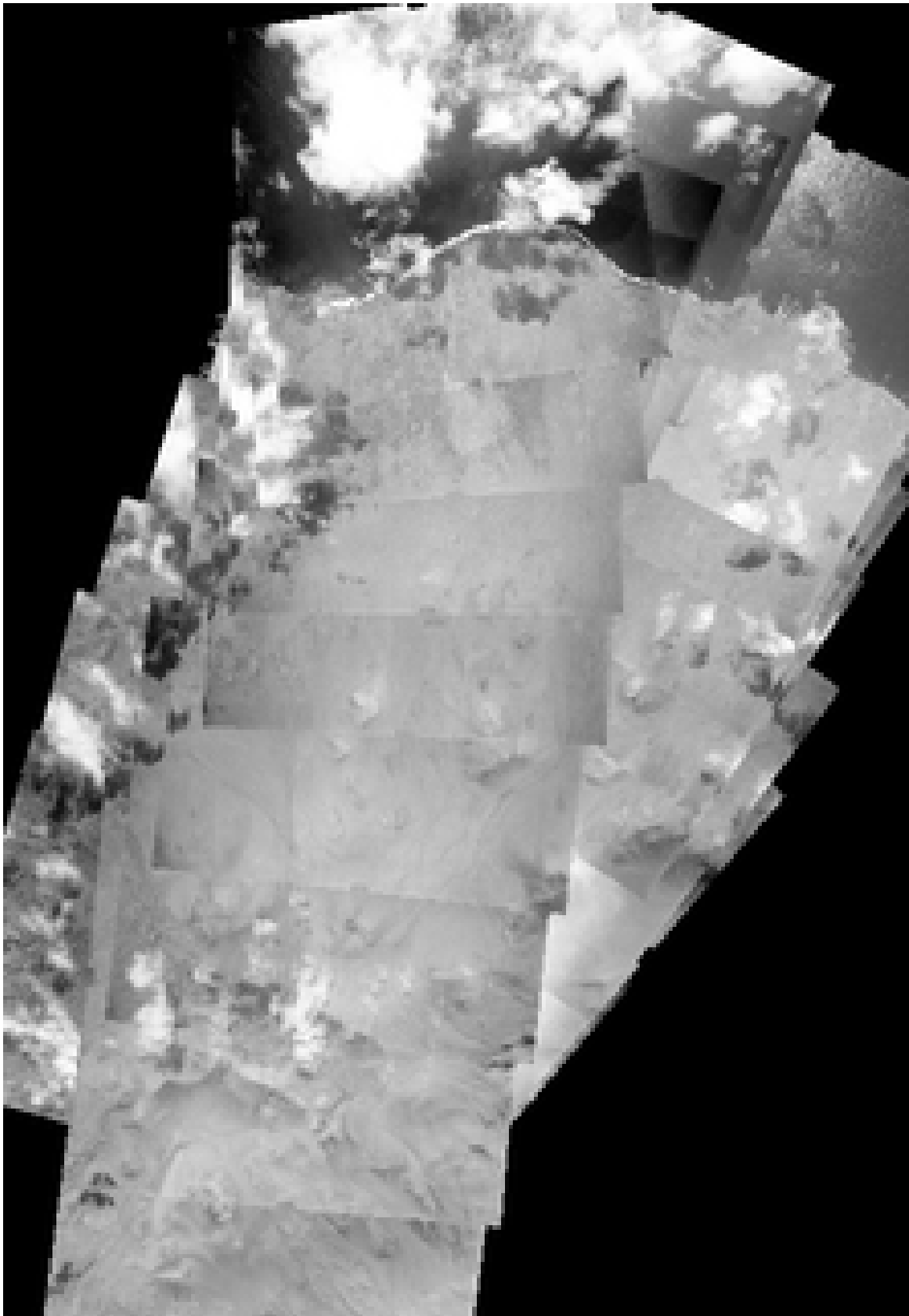
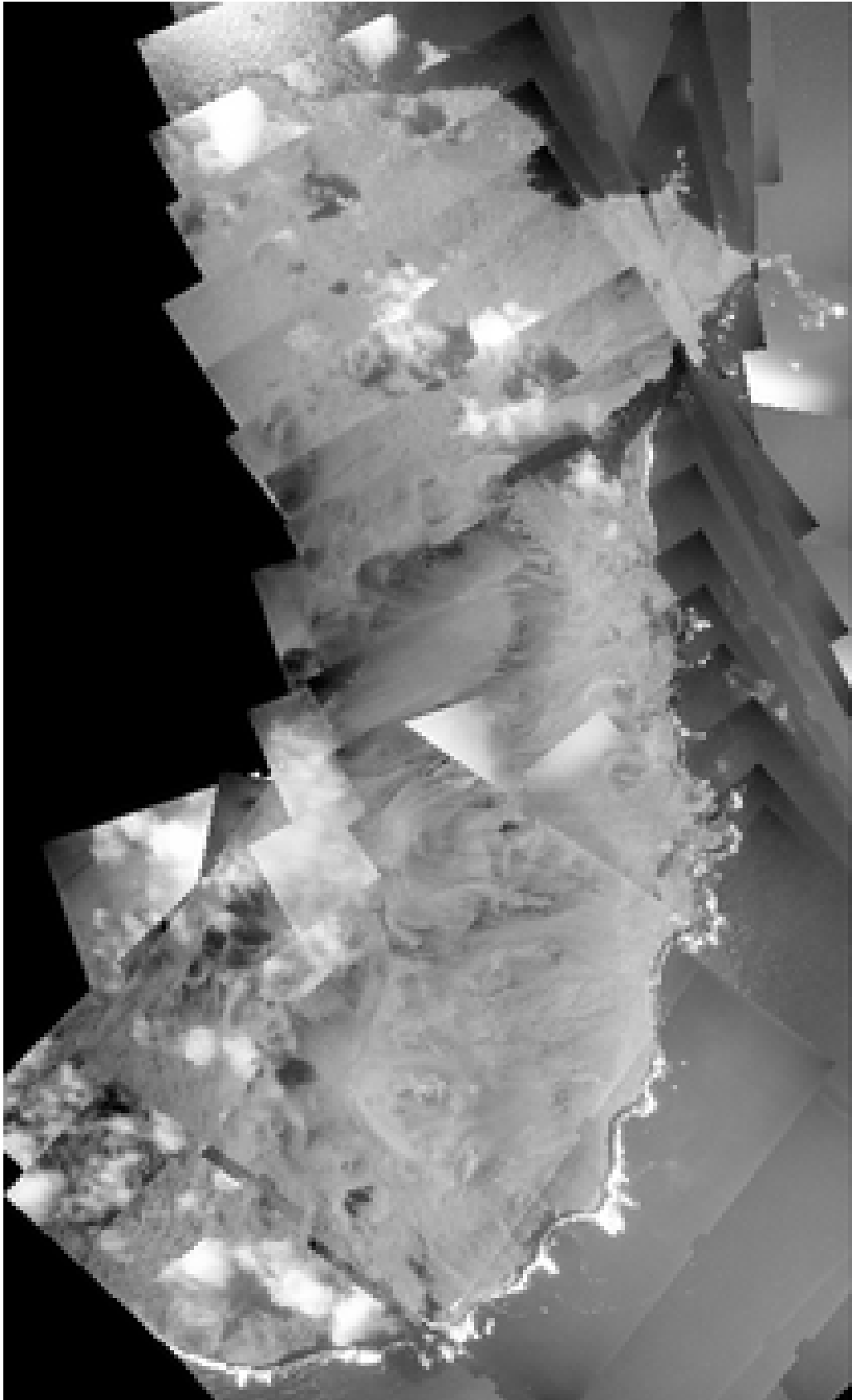


FIGURE 82 – Extraits de la “Pseudo-orthoimage”







## 14 Recalage d'images

### 14.1 Recalage sur une couche de référence

#### 14.1.1 Méthode

Dans les exemples qui suivent, le travail porte sur des images aériennes de différentes résolutions et acquises par différents capteurs optiques que l'on souhaite recalculer sur une ortho-image de référence (à savoir ici l'ortho Référence 3D / GéoBase de résolution 5m). L'objectif est donc de déformer ces images de manière à les rendre superposables (à la résolution près) à l'ortho-image de référence.

La méthode proposée est la suivante :

1. On utilise SIFT afin d'obtenir des points homologues entre l'image à recalculer et la référence. On considère que l'on dispose d'un géoréférencement (plus ou moins) approché de l'image à recalculer. Cette information permet de connaître directement quelle zone de l'ortho-image de référence lui correspond (et donc où y effectuer la recherche de points homologues).
2. Un premier modèle de géoréférencement est calculé à partir de ces points. Dans le cas présent, comme un MNT est associé à l'ortho-image de référence, ces points peuvent être considérés comme des points d'appui, leurs coordonnées terrain 3D étant donc connues.

Le modèle de géoréférencement utilisé, composé de deux fonctions rationnelles, est le suivant :

$$x_N = \frac{a_1 + a_2 \cdot X_N + a_3 \cdot Y_N}{1 + c_1 \cdot Z_N} \quad \text{et} \quad y_N = \frac{b_1 + b_2 \cdot X_N + b_3 \cdot Y_N}{1 + c_1 \cdot Z_N}$$

avec d'une part les coordonnées normalisées du point dans l'image à recalculer :

$$x_N = \frac{x - x_{centre}}{x_{centre}} \quad \text{et} \quad y_N = \frac{y - y_{centre}}{y_{centre}}$$

où  $(x; y)$  et  $(x_{centre}; y_{centre})$  désignent respectivement les coordonnées image du point courant et du centre du cliché.

et d'autre part les coordonnées terrain normalisée du point :

$$X_N = \frac{X - X_{centre}}{X_{coeff}} \quad Y_N = \frac{Y - Y_{centre}}{Y_{coeff}} \quad \text{et} \quad Z_N = \frac{Z - Z_{centre}}{Z_{coeff}}$$

$$X_{centre} = \frac{X_{max} + X_{min}}{2} \quad \text{et} \quad X_{coeff} = \frac{X_{max} - X_{min}}{2}$$

$$Y_{centre} = \frac{Y_{max} + Y_{min}}{2} \quad \text{et} \quad Y_{coeff} = \frac{Y_{max} - Y_{min}}{2}$$

$$Z_{centre} = Z_{MNT}(X_{centre}; Y_{centre}) \quad (\text{altitude au centre du cliché}) \quad \text{et} \quad Z_{coeff} = 500$$

où  $(X; Y; Z)$  désigne les coordonnées terrain du point courant tandis que  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$  et  $Y_{max}$  sont les coordonnées terrain de l'emprise du cliché à recalculer.

3. L'image à recalculer est "orthorectifiée" selon ce modèle. On obtient une nouvelle image à recalculer.
4. Cette nouvelle image est recalculée finement sur l'image de référence (corrélacion, méthodes de recalculer rigide ou déformant de la librairie Insight ToolKit [ITK09, ISNC05])

Des traitements peuvent être appliqués préalablement à l'image la mieux résolue, de manière à la flouter (et donc à la rendre plus ressemblante à l'image de référence).

#### 14.1.2 Tests effectués

**Toulouse - basse résolution** L'image à recalculer est ici une image aérienne argentique scannée dont la résolution n'est pas très bonne. Dans ce cas où les images se ressemblent suffisamment, tout se passe bien sans traitements complémentaires : un nombre suffisant de points homologues corrects assez bien répartis est obtenu d'emblée (voir fig. 84 et 83).

FIGURE 83 – Points homologues détectés entre une image aérienne scannée (à gauche) et l'ortho-image de référence (à droite)

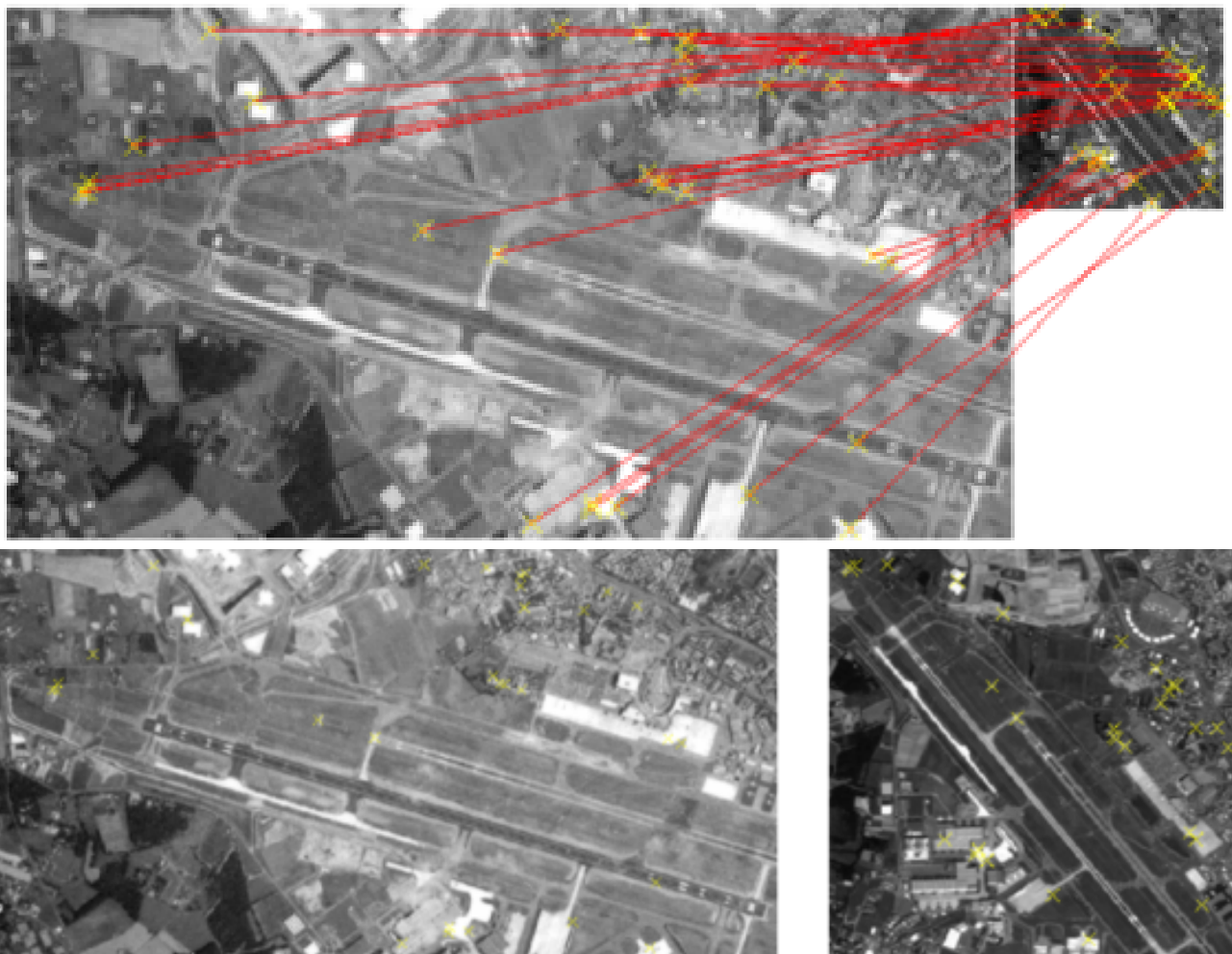
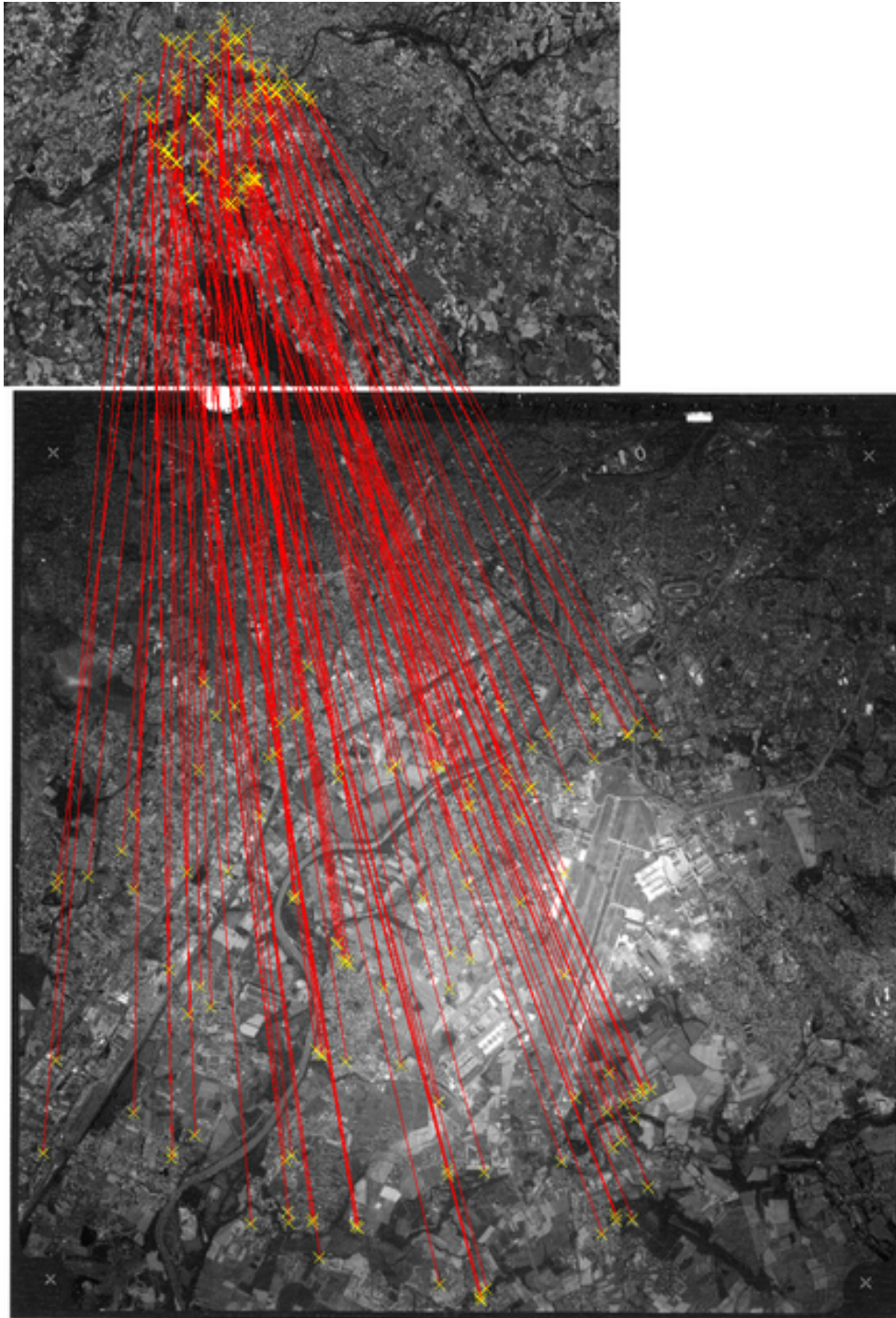


FIGURE 84 – Points homologues détectés entre une image aérienne scannée (en bas) et l’ortho-image de référence (en haut)



**Toulouse - haute résolution** L'image à recaler est ici une image aérienne de résolution 70 cm acquise par la caméra numérique de l'IGN. Dans ce cas, la variation d'échelle est trop importante et un calcul directement à pleine résolution ne fournit qu'un seul appariement. En sous-échantillonnant l'image la mieux résolue (i.e. l'image à recaler) d'un facteur 4, un nombre suffisant de points homologues corrects assez bien répartis est obtenu (voir fig. 85).

Le recalage obtenu en utilisant ces points est d'ailleurs meilleur que celui obtenu par simple orthorectification de l'image aérienne (dont le géoréférencement connu avait été calculé par aérotriangulation). (voir les images en damier de l'image recalée et de l'image de référence des figures 86 et 87).

D'autres tests consistant à recaler une image aérienne de résolution 20 cm sur la même orthoimage de référence ont été effectués. Cette fois, la différence d'échelle était vraiment trop importante et la mise en correspondance a échoué.

FIGURE 85 – Points homologues détectés

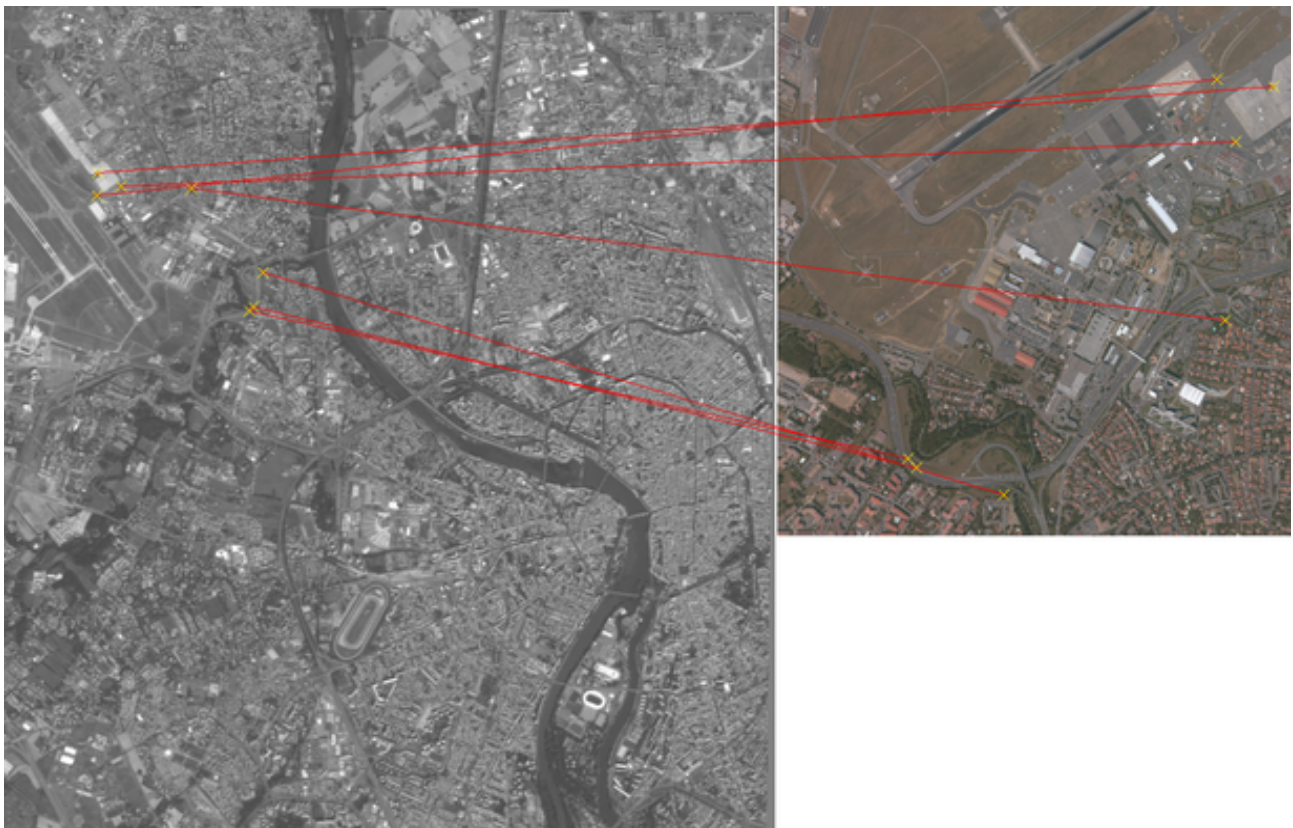




FIGURE 86 – Recalage avec un modèle calculé à partir des points fournis par SIFT

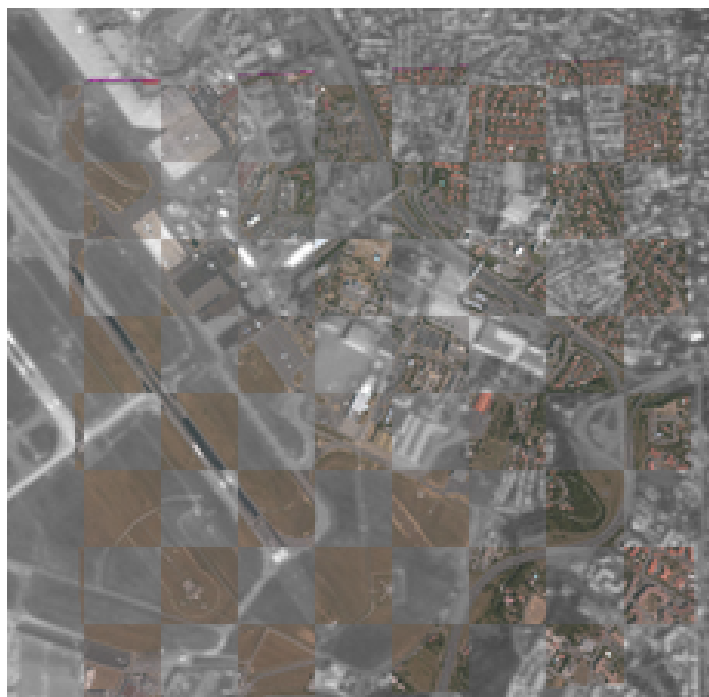
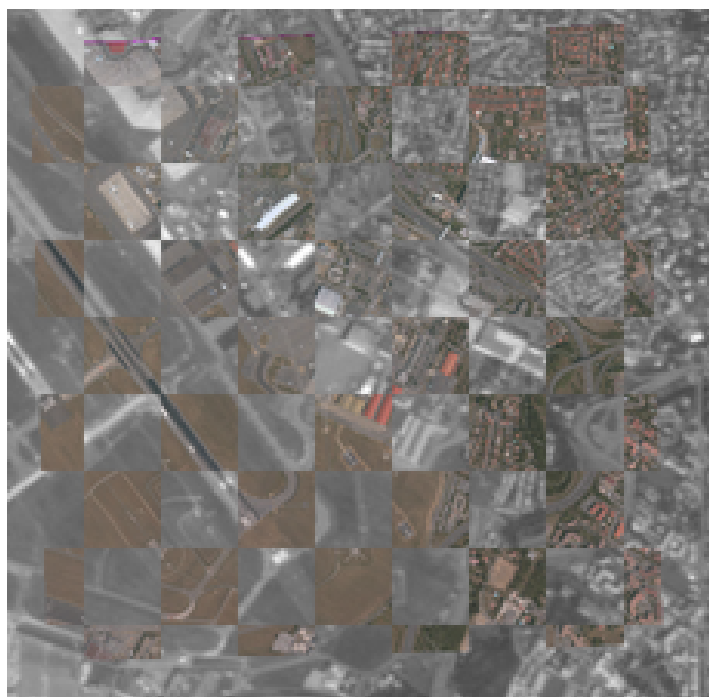


FIGURE 87 – Orthorectification utilisant le géoréférencement (fourni) des images aériennes



**Dunkerque** L'image à recaler est ici une image aérienne de résolution 70 cm acquise par la caméra numérique de l'IGN. Dans ces deux cas également, la variation d'échelle est trop importante pour un calcul direct à pleine résolution. En appliquant un filtre moyenne (de taille 4x4) à l'image la mieux résolue (i.e. l'image à recaler), un nombre suffisant de points homologues corrects assez bien répartis est obtenu (voir fig. 89 et 90). Le recalage obtenu dans le premier cas en utilisant les points calculés est présenté figure 88.

FIGURE 88 – Recalage avec un modèle calculé à partir des points fournis par SIFT

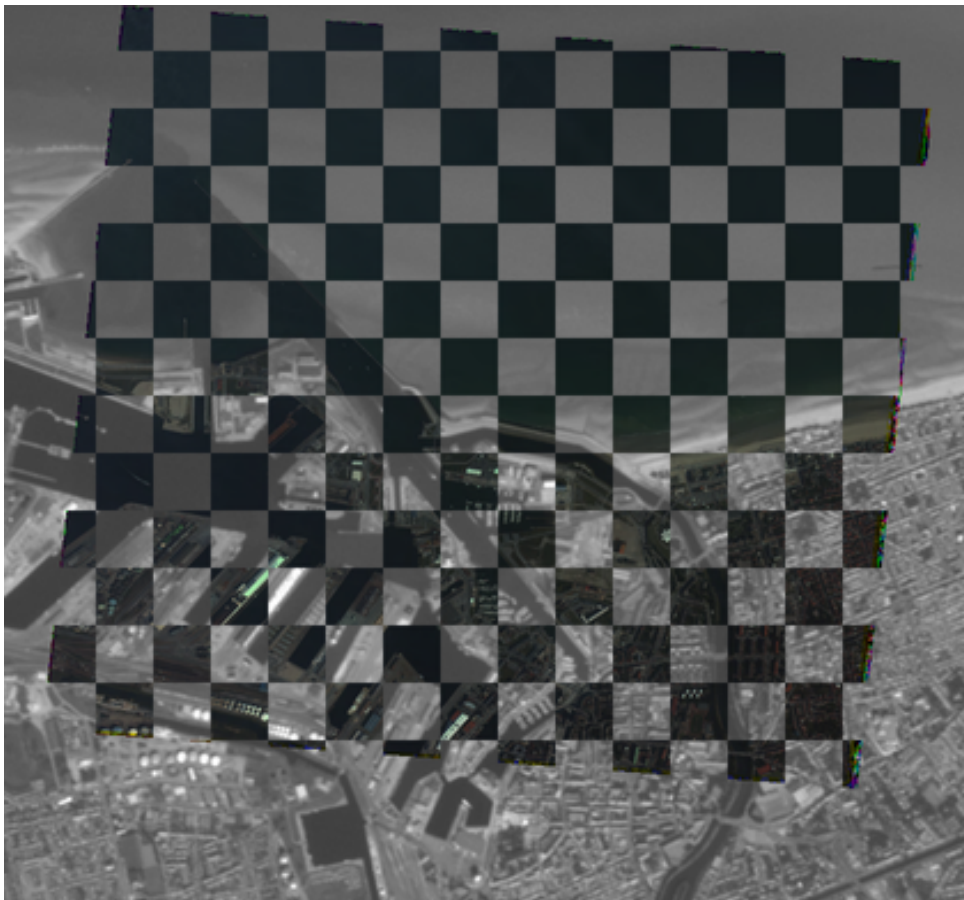


FIGURE 89 – Points homologues détectés entre une image aérienne de résolution 70cm (à droite) et l'orthoimage de référence de résolution 5m (à gauche)

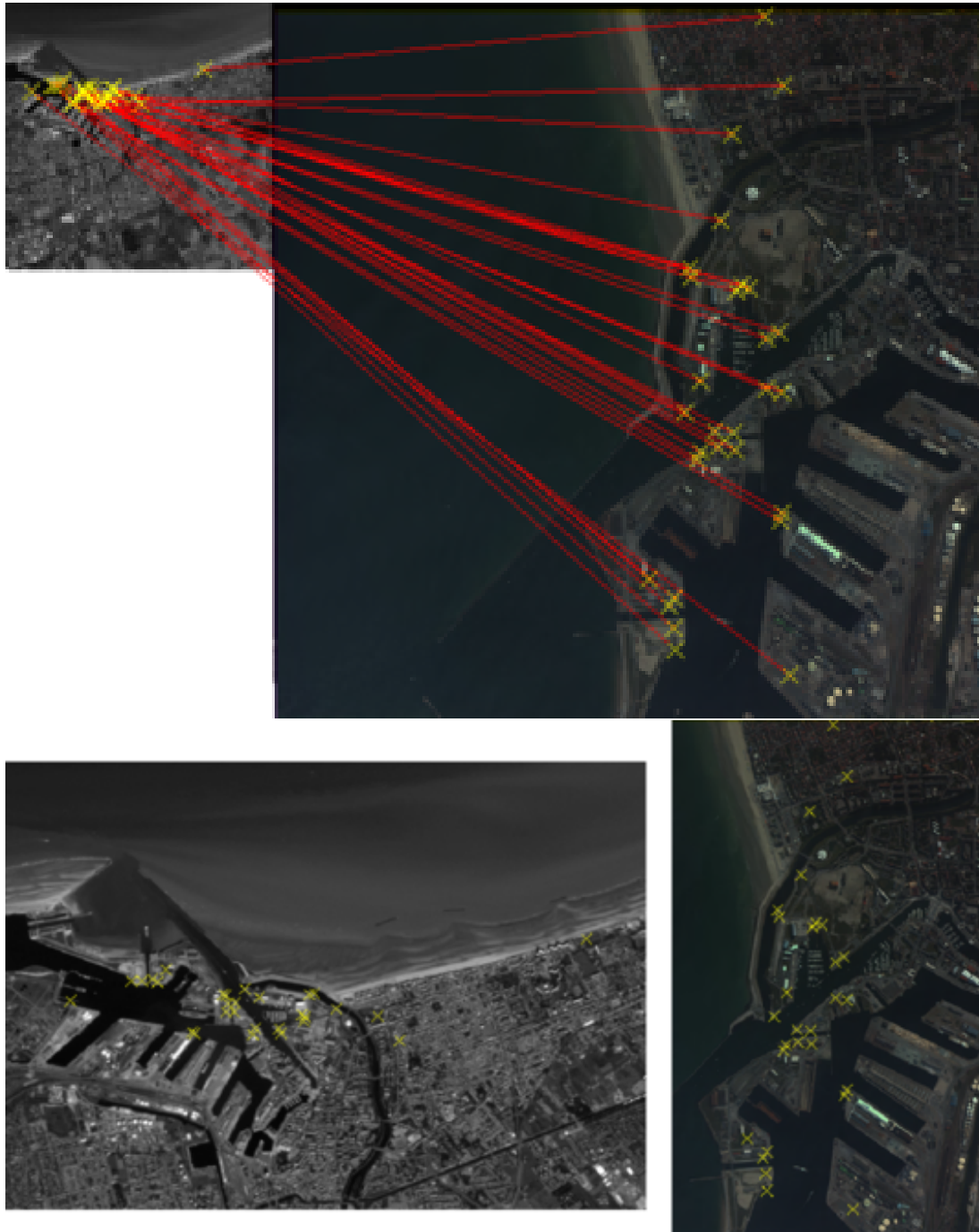
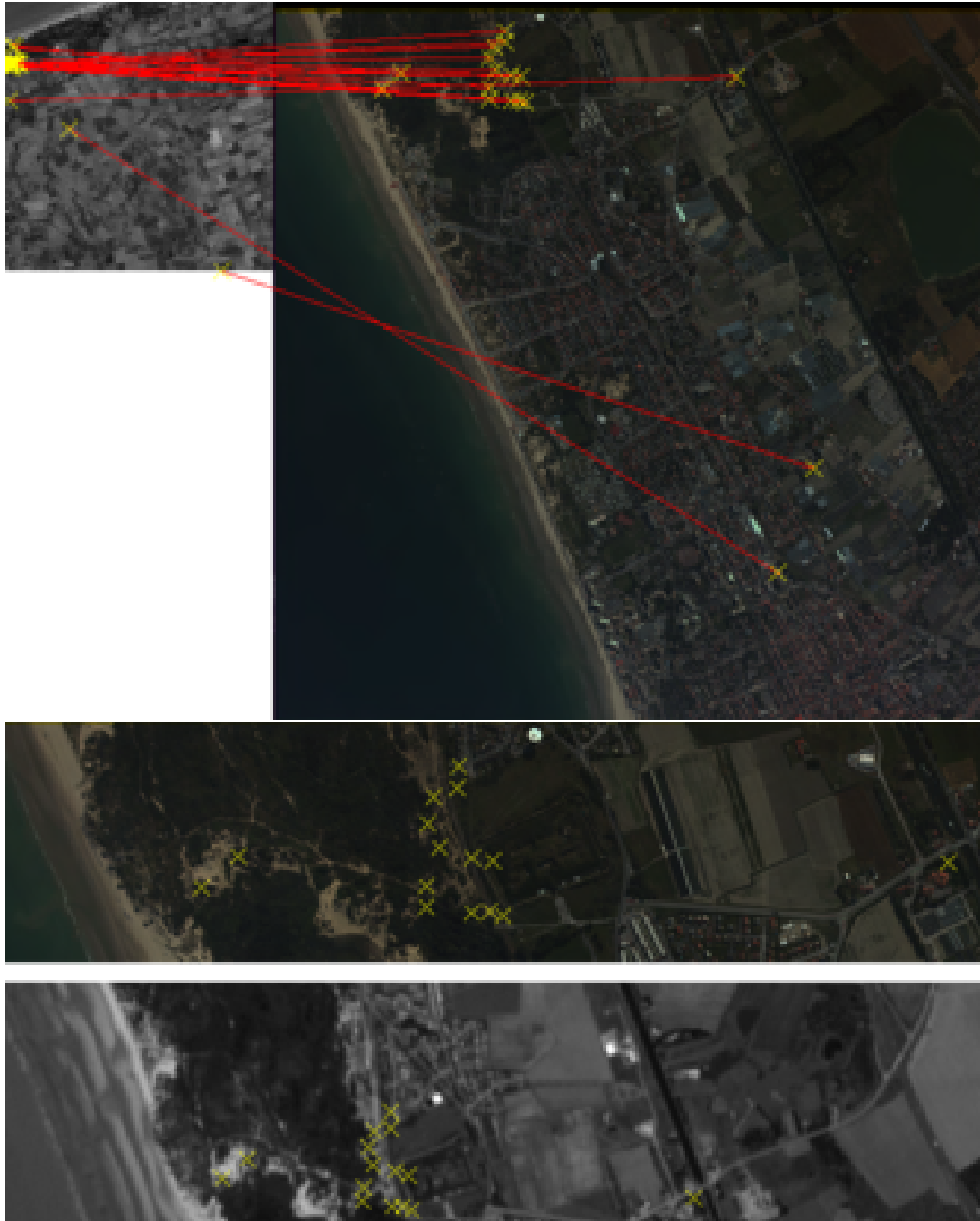
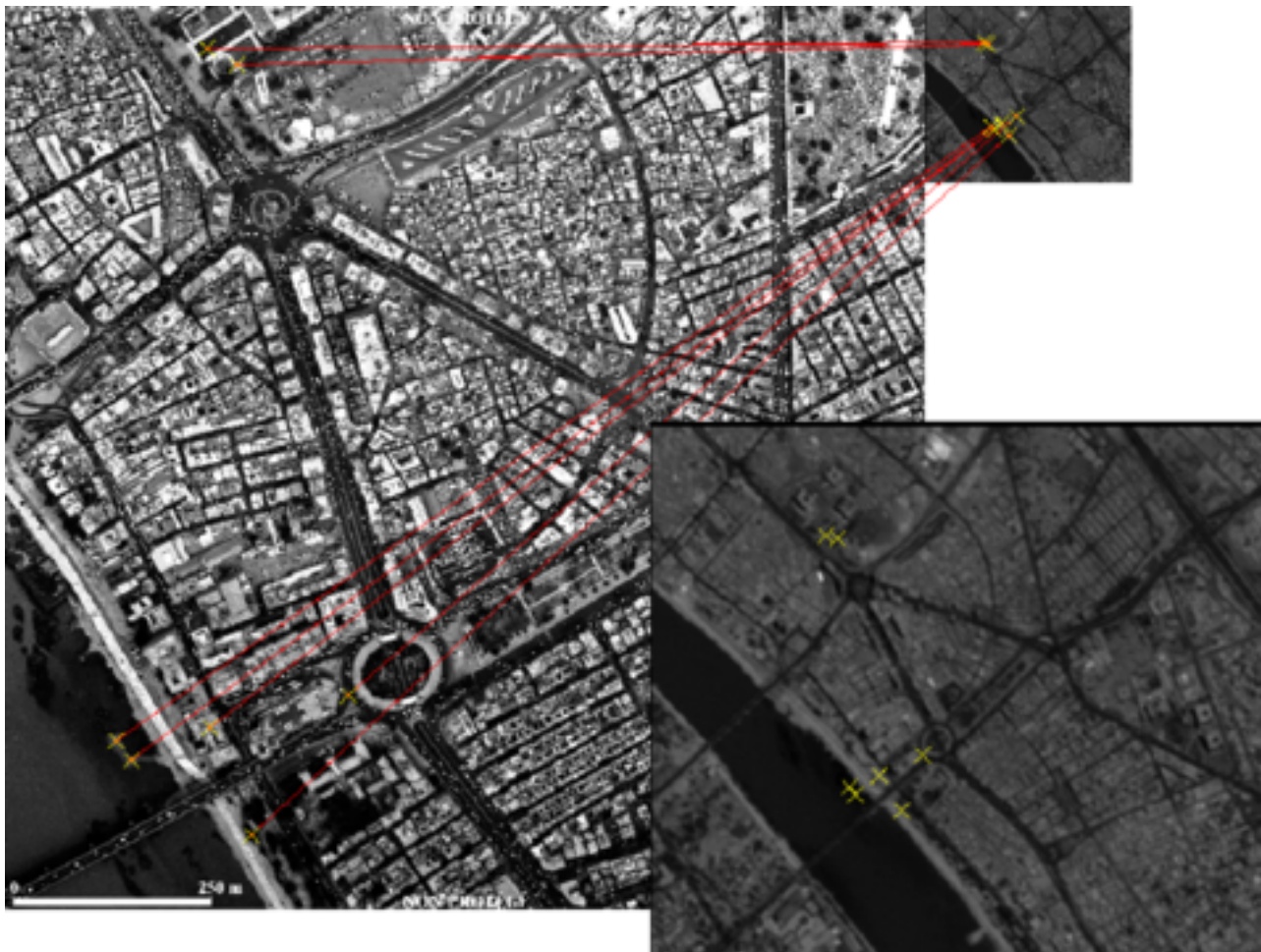


FIGURE 90 – Points homologues détectés (avant élimination des fautes) entre une image aérienne de résolution 70cm (à droite) et l'orthoimage de référence de résolution 5m (à gauche)



**Bagdad** Il est possible de déterminer quelques points de liaison entre ces deux images de résolution bien différente (voir fig. 91). L'image la mieux résolue a préalablement été convoluée par un filtre moyenne de taille  $10 \times 10$ . Des résultats analogues sont obtenus en la convoluant avec une gaussienne de  $\sigma = 2$ .

FIGURE 91 – Points homologues détectés (à gauche, image à recaler ; à droite, ortho-image de référence ; en bas, détail de l'ortho-image de référence)



**Chamonix - cliché argentique scanné** L'image à recaler est un cliché argentique scanné issu d'une prise de vue départementale de vocation de l'IGN effectuée en 1998. Le calcul des points homologues se passe bien (sur une image à recaler convoluée avec un filtre moyenne  $4 \times 4$ ) malgré la présence de nuages sur l'image de référence. (voir figures 92 et 93)

FIGURE 92 – Points homologues détectés entre l'image à recaler (à droite) et l'ortho-image de référence (à gauche)

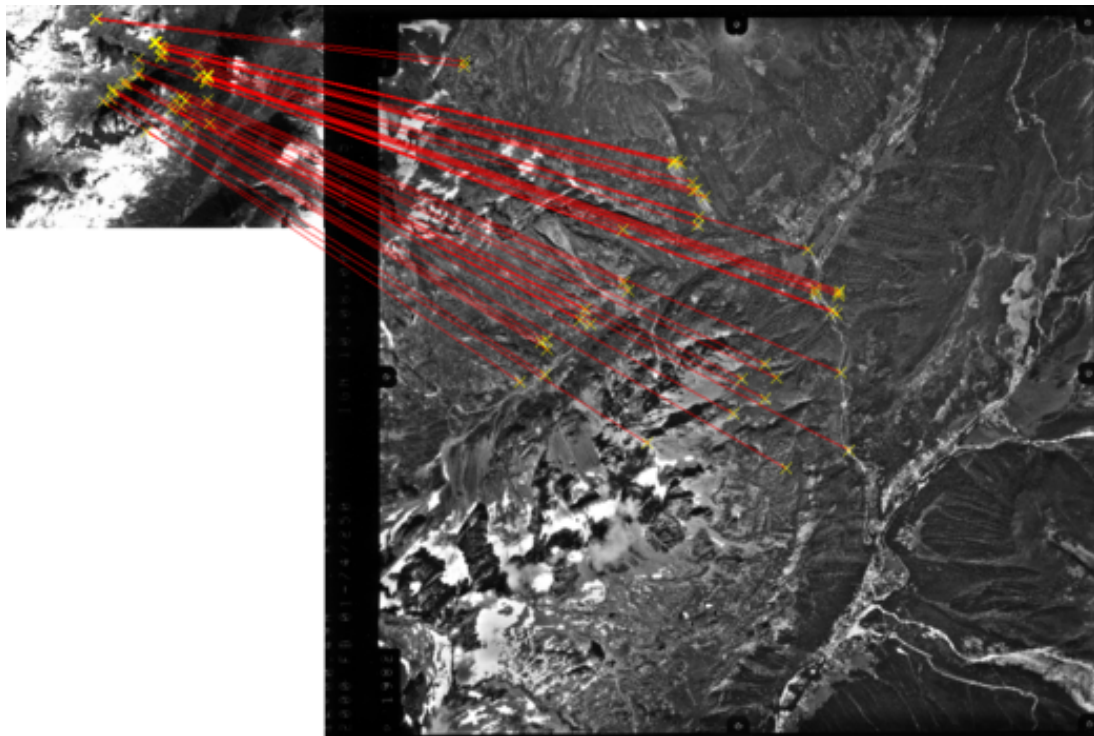
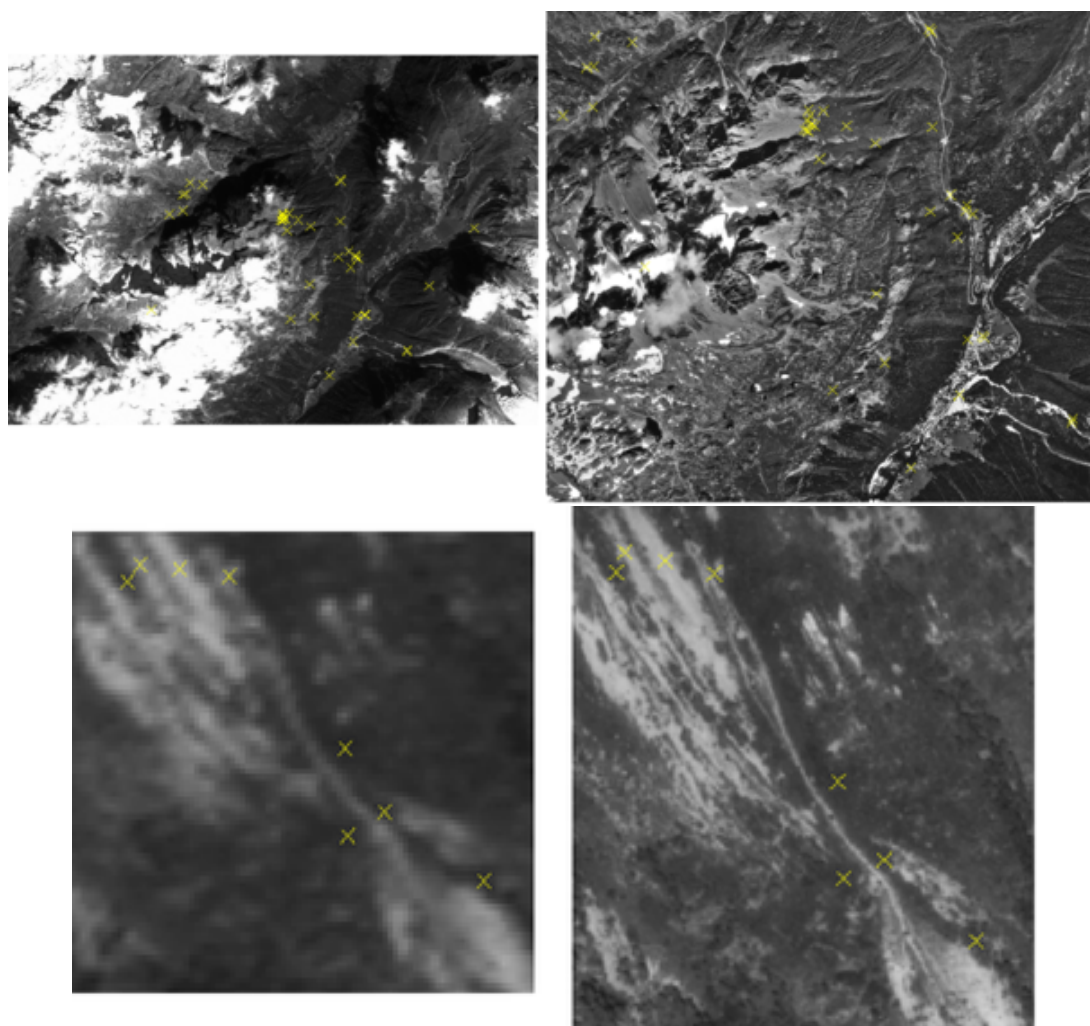


FIGURE 93 – Points homologues détectés entre l'image à recaler (à droite) et l'ortho-image de référence (à gauche)



**Chamonix - cliché caméra numérique** L'image à recaler est un cliché orthorectifié de résolution 70cm acquis par la caméra numérique de l'IGN. Ce cliché a préalablement subi une orthorectification (insuffisante pour le recalage), son géoréférencement étant connu (issu d'une aérotriangulation).

Le calcul des points homologues se passe bien avec l'image à recaler préalablement convoluée avec un filtre moyenne 4x4 : des points homologues sont détectés en nombre suffisant avec une bonne répartition ainsi qu'on peut le voir sur la figure 94.

Le recalage obtenu en utilisant ces points est d'ailleurs meilleur que celui obtenu par simple orthorectification de l'image aérienne. (voir les images en damier de l'image recalée et de l'image de référence des figures 95)

Un second exemple de mise en correspondance donnant de bons résultats est présenté figure 96

FIGURE 94 – Points homologues détectés entre l'image à recaler (à droite) et l'ortho-image de référence (à gauche)

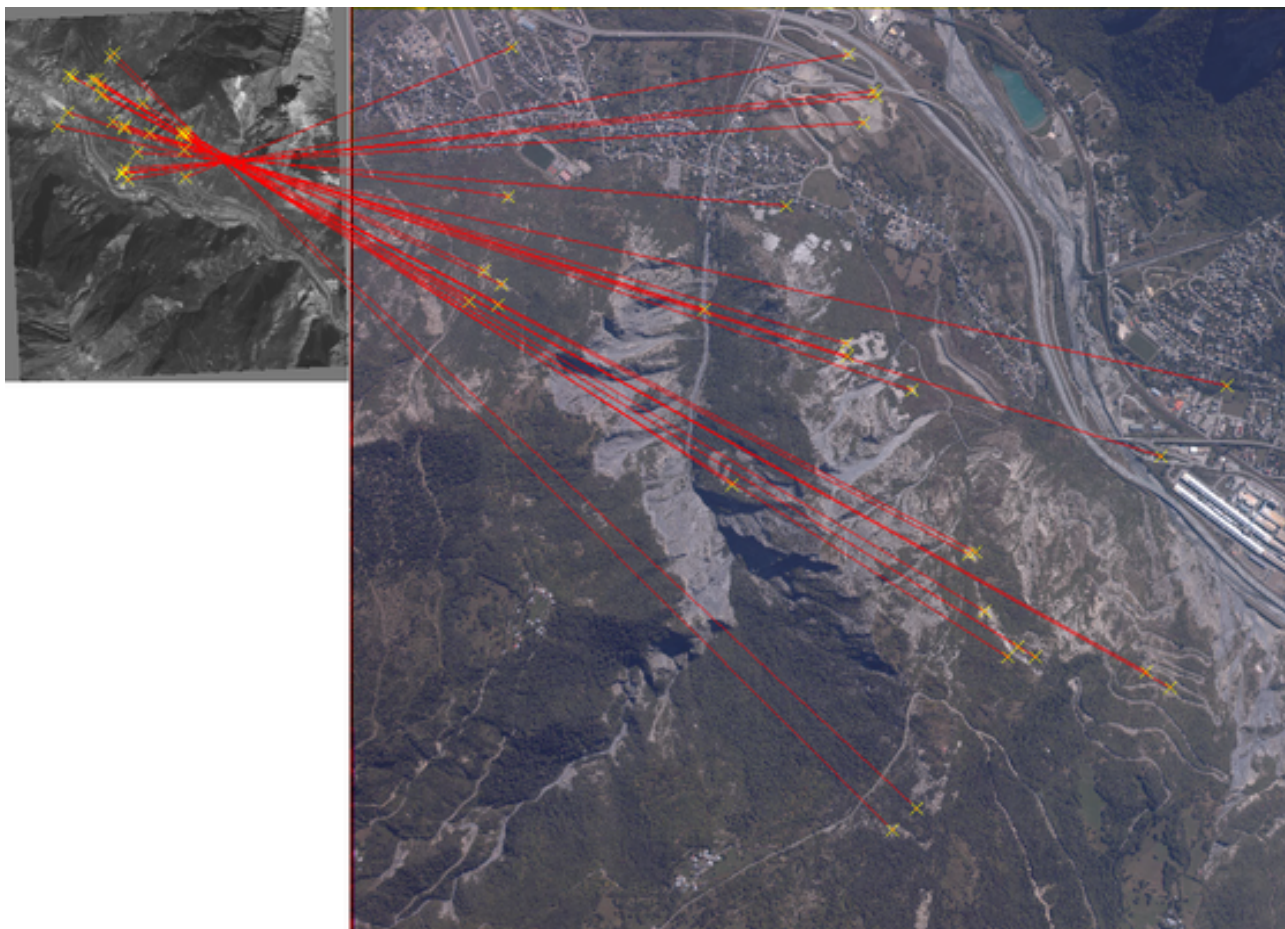




FIGURE 95 – Recalage avec un modèle calculé à partir des points fournis par SIFT (en haut) et orthorectification utilisant le géoréférencement (fourni) des images aériennes (en bas)

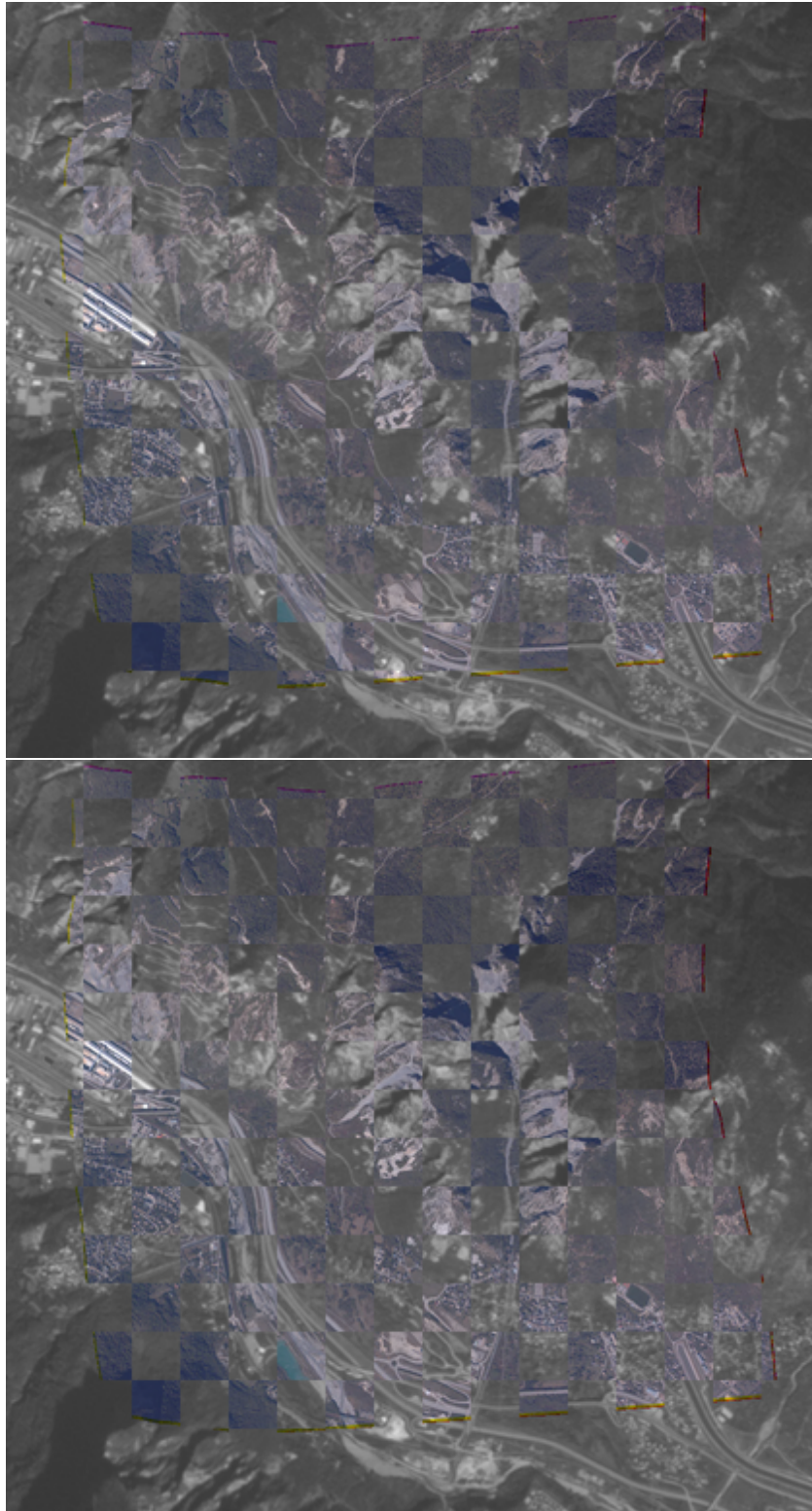


FIGURE 96 – Points homologues détectés entre l'image à recaler (en haut) et l'ortho-image de référence (en bas)

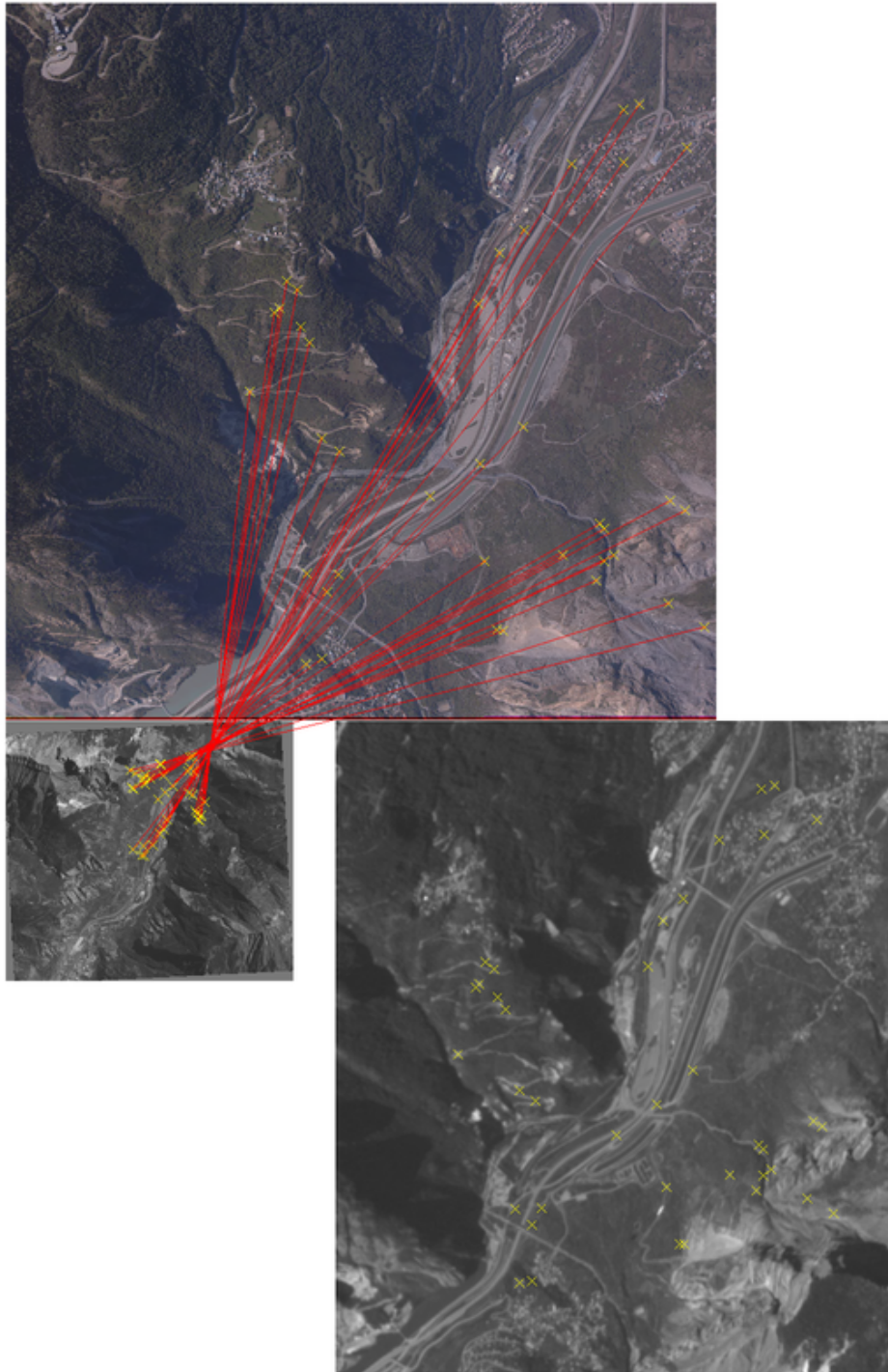
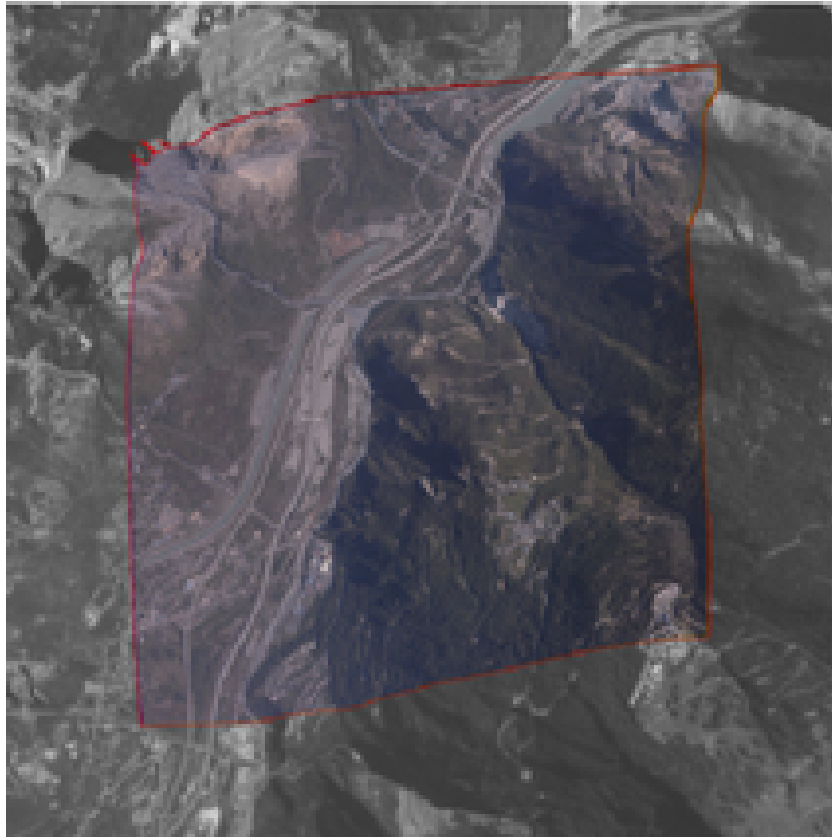
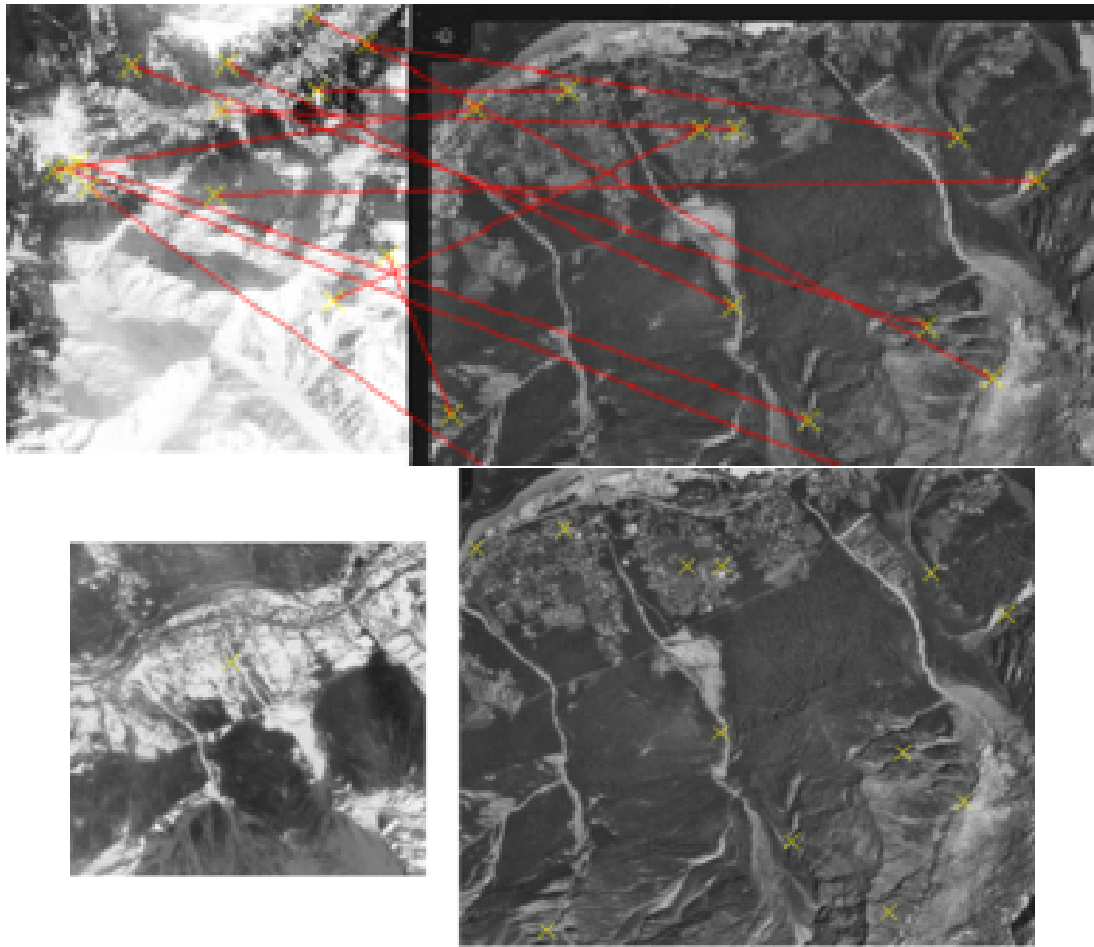


FIGURE 97 – Image à recaler orthorectifiée (à partir des points d'appui fournis par SIFT) superposée à l'ortho-image de référence



**Chamonix - cliché argentique scanné - diachronisme fort** L'image à recaler est un cliché argentique scanné issu d'une prise de vue départementale de vocation de l'IGN. Dans ce cas, la détection de point homologue a échoué du fait du diachronisme important entre les images, ainsi qu'il est possible de le constater figure 99.

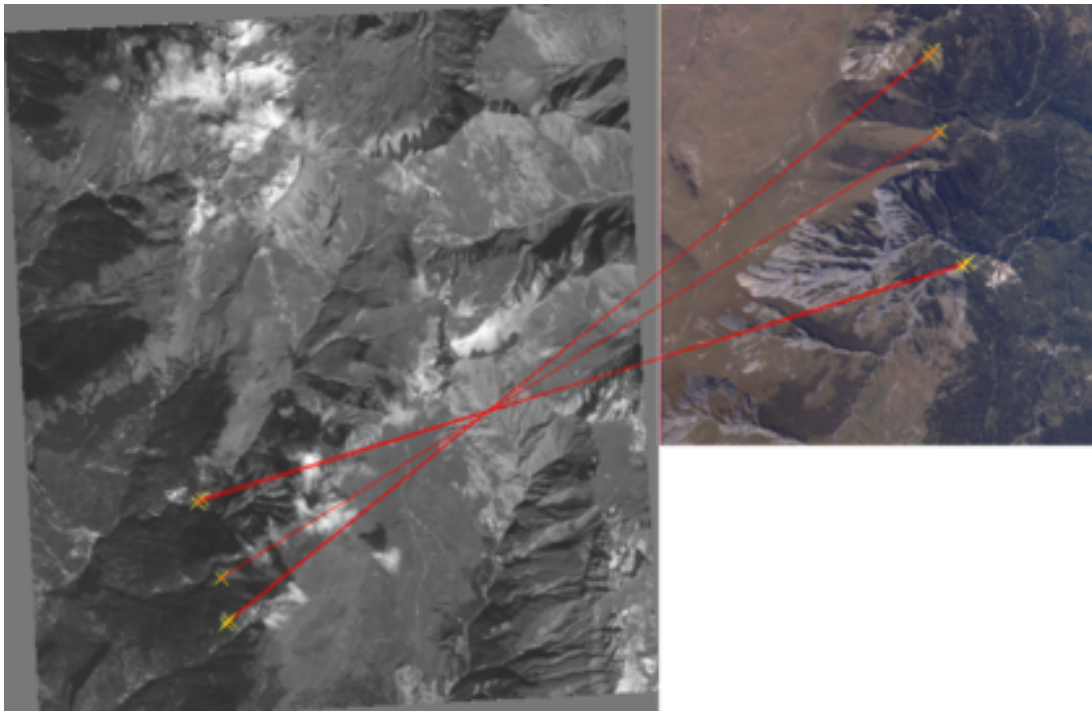
FIGURE 98 – Echet lors de la détection de points homologues



**Chamonix - cliché caméra numérique - échec** L'image à recaler est un cliché orthorectifié de résolution 70 cm acquis par la caméra numérique de l'IGN. La détection de points homologues a échoué, à la fois à cause du diachronisme entre les images se manifestant notamment par la présence de nuage sur l'une d'elles et de la position différente des ombres sur les deux images, mais aussi en raison du fait que très peu de détails apparaissent sur certaines zones de l'image. (voir figure 99).

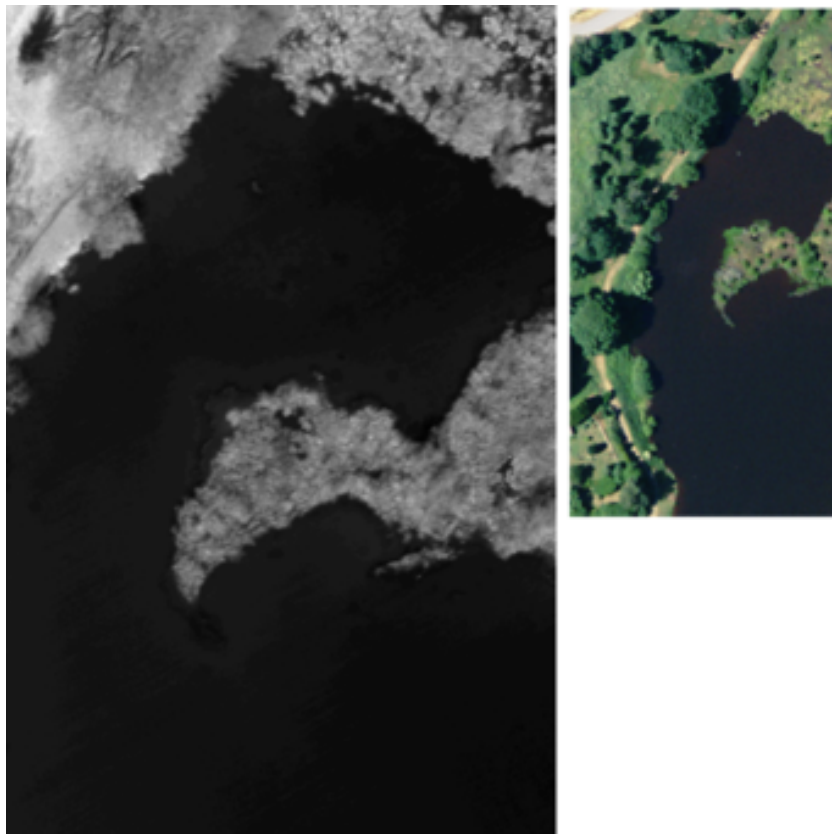
En fait, on parvient à extraire 5 points homologues entre l'image à recaler et l'image de référence, mais ces points sont quasiment alignés. Or une telle disposition est très mauvaise pour calculer le géoréférencement de l'image à recaler. Par conséquent, ce calcul échoue.

FIGURE 99 – Echec lors de la détection de points homologues entre ces deux images



**Diachronisme** Un diachronisme trop important associé à une forte variation d'échelle empêche là-encore la détection de points homologues : l'une des images a été acquise en été, l'autre en hiver.

FIGURE 100 –



### 14.1.3 Conclusions

En pratique, le fait de “flouter” l’image la mieux résolue (en lui appliquant les filtres gaussien ou moyenne) ou de la sous-échantillonner permet généralement assez logiquement d’améliorer les résultats (plus de points corrects mieux répartis). Cela permet en effet de réduire le facteur d’échelle par rapport à l’image basse résolution : ainsi, les deux images se ressemblent alors davantage.

La limitation même grossière de la zone de recherche des points homologues sur l’image de référence permet également généralement d’améliorer les résultats, principalement en évitant certaines mises en correspondance erronées.

Des tests ont été menés sur plusieurs types d’images (photographies aériennes argentiques ou numériques, résolutions différentes) et dans des contextes différents (zones urbaines, zones rurales, zones littorales, zones de montagne).

Les résultats obtenus sont assez variables (nombres de mises en correspondance, erreurs) et souvent difficiles à prévoir : en effet, si SIFT est généralement robuste aux variations d’échelle (que l’on a ici atténuées en jouant sur la résolution des images), aux rotations (2D), aux variations d’éclairement, aux changements de points et autres transformations géométriques, il montre une grande sensibilité au diachronisme, ou plus généralement au fait que les images soient trop différentes.

On peut aussi remarquer que le fait d’obtenir des résultats le plus souvent acceptables lors des différents tests vient aussi du type d’images utilisées, et notamment du fait que l’image de référence était une ortho-image GéoBase de résolution 5m. Une telle résolution gomme en effet un certain nombre de détails qui auraient été susceptibles de varier au cours du temps et auraient eu pour effet de perturber la recherche de points homologues. La “basse” résolution atténue les effets du diachronisme.



## Quatrième partie

# Tests sur des images terrestres

Les tests qui suivent ont principalement porté sur des images acquises à l'extérieur, dans des rues en milieu urbain dense (centre ville). Il s'agit donc surtout d'images de façades.

## 15 Méthode

### 15.1 Filtrage des fautes

Bien qu'ils soient généralement de bonne qualité, les appariements comprennent également des fautes pouvant perturber les calculs d'orientation réalisés à partir de ces mesures. Plusieurs méthodes de filtrage sont possibles mais toutes ne peuvent pas être utilisées suivant les différents cas de figure (orientation approchée ou non) :

- Notamment dans le cas d'images fronto-parallèles aux façades, on se trouve dans un cas de figure "2,5D" assez analogue à l'aérien. Le **filtrage par considération du voisinage** (dont le principe a déjà été expliqué en 5.2.1 et en 12.1.1) peut donc être utilisé, du moins en première approximation quand les surfaces sont fronto-parallèles, car on a tout de même le plus souvent des objets situés au premier plan (véhicules en stationnement, arbres, mobilier urbain) ainsi que des objets en mouvement.
- Un **filtrage aller-retour** (voir partie 5.1.2) permet d'éliminer des fautes grossières, notamment les points qui n'ont pas d'homologue naturellement en raison de la présence de parties cachées (cf vérification ou non de l'hypothèse d'existence d'homologues). Il s'agit de rechercher les homologues des points de l'image 1 dans l'image 2, puis les homologues des points de l'image 2 dans l'image 1. Seules les paires trouvées lors de ces deux appariements sont conservées.
- La majorité des points se trouve sur des façades, i.e. des surfaces que l'on va assimiler à des plans. Une méthode de filtrage peut donc consister à calculer une **homographie** ou même une simple **affinité** entre les deux images et à éliminer les outliers. Les paramètres initiaux de la transformation seront déterminés soit directement, soit en utilisant un algorithme de type RANSAC. Ils seront ensuite affinés au cours de quelques itérations.
- Méthode plus générale utilisant les écarts des points par rapport aux lignes épipolaires (connues soit d'après le géoréférencement approché, soit en calculant la matrice fondamentale à partir des points homologues par exemple à l'aide des outils de (Kalantari, 2009)) sur lesquelles ils devraient se situer.
- Filtrage photogrammétrique : les points ayant des résidus trop forts à l'issue de la compensation sont rejetés. Ce filtrage a été utilisé ici dans le cadre du calcul d'aérotriangulation des images du véhicule, lorsque les erreurs aberrantes avaient déjà été éliminées. Il pourrait également être utilisé en effectuant d'abord une mise en place relative des clichés à l'aide des outils de (Kalantari, 2009) par une méthode RANSAC.



## 16 Appariement entre images aériennes et terrestres : application à l'amélioration du géoréférencement d'un véhicule infrastructure

### 16.1 Introduction

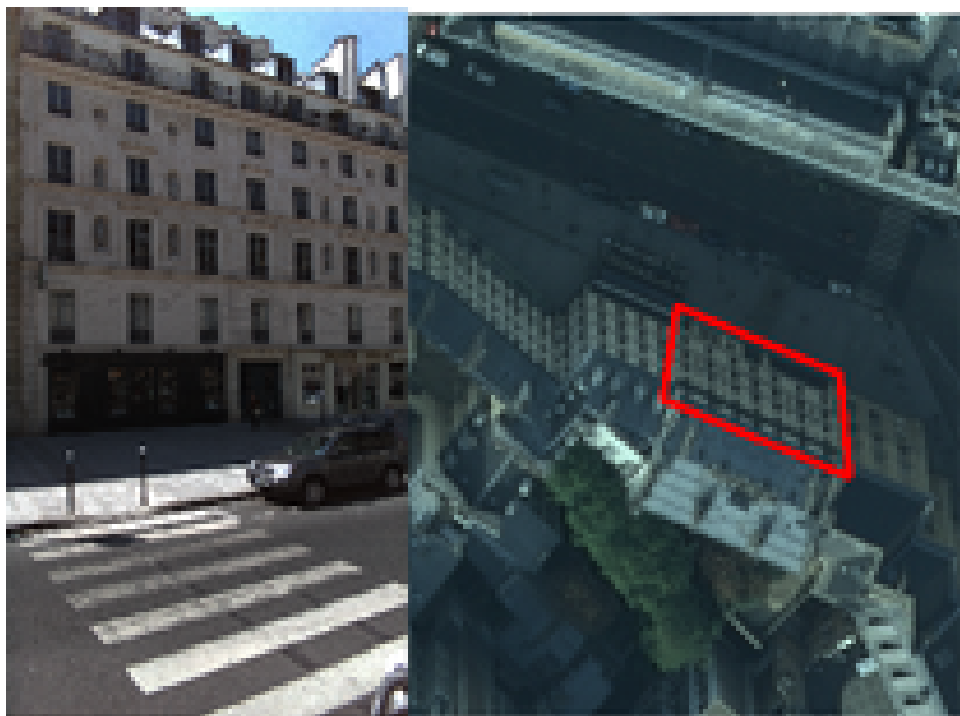
Il s'agit d'essayer d'obtenir des points de liaison entre des clichés aériens et des images terrestres acquises au sol par le véhicule Stéréopolis. L'objectif final visé ici est l'amélioration du géoréférencement des images du véhicule cartographe. Avec de tels points de liaison, il deviendrait en effet possible de déterminer au cours d'un même calcul d'aérotriangulation les orientations des images du véhicule ainsi que celles des clichés aériens ; la cohérence entre les deux types de données en serait améliorée.

Plus précisément, les tests qui suivent visent à évaluer la faisabilité de la détection de points homologues entre images aériennes et terrestres en particulier au niveau des façades de bâtiments.

### 16.2 Tests

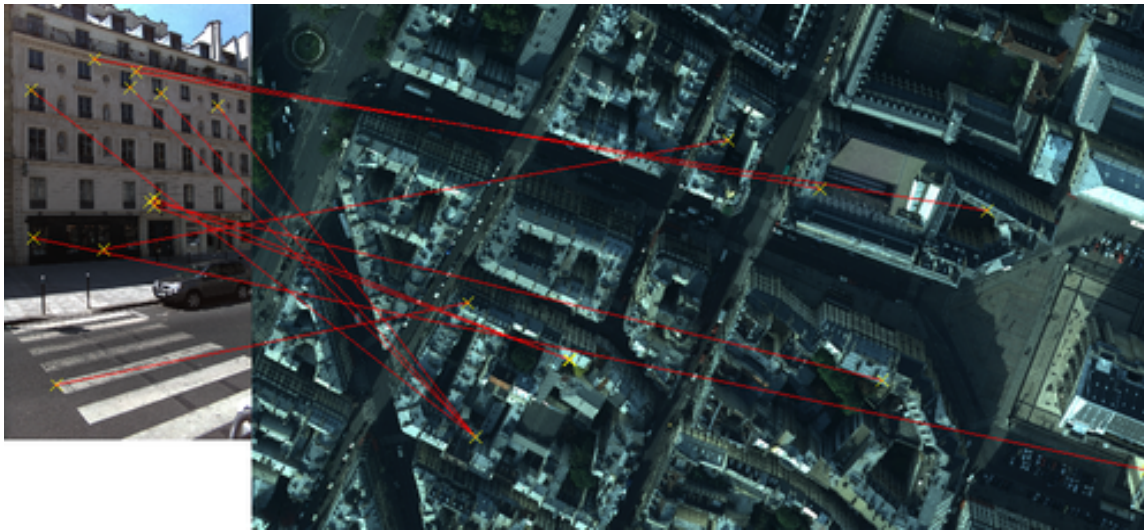
Les tests ont d'abord porté sur la paire d'images ci-dessous (fig. 101) :

FIGURE 101 – L'emprise approximative de l'image terrestre sur l'image aérienne est indiquée par le tracé rouge.



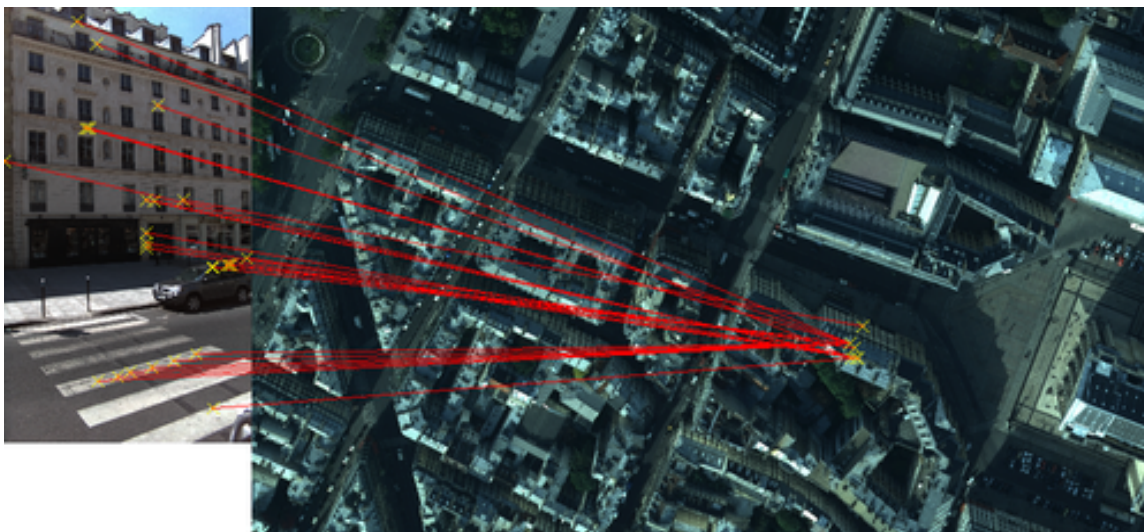
Le premier test a consisté naïvement à effectuer directement à pleine résolution une extraction de points d'intérêt suivie d'un appariement sans prédicteur : on espère alors que le caractère multi-échelle de SIFT sera suffisant pour trouver des points homologues entre les deux images. Le résultat obtenu (cf figure 102 ) est assez logiquement mauvais : aucun point n'est correctement apparié.

FIGURE 102 – Appariement brut sans prédicteur de points extraits à pleine résolution



Un second test (voir figure 103) à pleine résolution a ensuite été effectué en restreignant cette fois la zone de recherche de points homologues dans le cliché aérien. Le résultat obtenu reste mauvais : aucun appariement n'est correct.

FIGURE 103 – Appariement avec restriction de la zone de recherche de points extraits à pleine résolution



Les deux images sont donc ici trop différentes pour obtenir directement des appariements corrects : il faut en effet compter à la fois avec :

- une forte variation d'échelle
- un changement de point de vue important (vue depuis le sol / vue depuis le ciel).
- des radiométries différentes

Ceci a pour conséquence :

- des points d'intérêt extraits le plus souvent différents d'une image à l'autre
- des descripteurs différents (encore accentué par le fait que le toit est visible sur l'image aérienne et risque donc d'être pris en compte dans le descripteur de certains points alors qu'il ne le sera pas pour les clichés terrestres)

Afin de se placer dans un cas de figure plus favorable, les deux images subissent un redressement (dont les paramètres sont connus) de manière à avoir une résolution semblable. Dans le cas présent, le redressement a été effectué manuellement, mais il serait tout à fait possible de l'automatiser ; en effet, d'une part, la texture de la façade dans l'image aérienne est naturellement redressée lors de la texturation du modèle Bati3D. On dispose donc pour chaque face du modèle de l'ortho-image associée provenant de chaque image aérienne. D'autre part, l'image terrestre est redressée en utilisant les points de fuite obtenus par l'approche de [KJPG08]. On considère ensuite que l'appariement est focalisé sur la même façade en supposant que l'on dispose d'une pose approchée du cliché terrestre à recalcr (comme c'est le cas d'images acquise par Stéréopolis). Ainsi que le montre l'illustration 105, SIFT parvient alors à fournir deux points homologues corrects.

Quelques appariements supplémentaires (voir figure 105) sont obtenus en convoluant par une gaussienne l'image terrestre (i.e. l'image initialement la mieux résolue) redressée, mais aussi en jouant sur la radiométrie des images (car si le descripteur SIFT est invariant aux variations d'illumination, le détecteur ne l'est pas complètement du fait du seuil d'élimination des points les moins contrastés. Une autre solution peut consister à désactiver ce filtrage.). En effet, on a alors deux images qui se ressemblent davantage et on a ainsi obtenu 8 appariements corrects contre 2 faux, qu'il serait donc possible d'éliminer (surtout si l'on dispose d'une connaissance approchée sur l'orientation des images).

FIGURE 104 – Appariements obtenus après redressement des images



FIGURE 105 – Appariements obtenus après redressement des images et en floutant l'image la mieux résolue



Néanmoins, la même méthode appliquée sur d'autres paires d'image ne permet pas forcément d'obtenir de points de liaison corrects. Par exemple, dans l'exemple ci-dessous (figure 106), les images ont un aspect trop différent pour permettre de trouver des points homologues, même après avoir subi un redressement (figure 107). On remarquera notamment dans ce cas particulier des inversions de contraste entre les fenêtres et les murs. La texture "3D" des fenêtres et des balcons perturbe également la mise en correspondance.

FIGURE 106 – Le tracé rouge indique l'emprise approximative de l'image terrestre sur l'image aérienne.



FIGURE 107 – Aucun appariement obtenu même après redressement des images et en floutant l'image la mieux résolue : l'inversion des contrastes et la texture 3D des fenêtres et des balcons perturbent l'appariement



### 16.3 Conclusion

Il n'a pas été possible d'obtenir de manière directe des points de liaison entre images aériennes et terrestres. Néanmoins, il est possible dans des conditions favorables d'obtenir quelques points homologues au niveau des façades en redressant les deux images (à la même résolution) et en les rendant les plus ressemblantes possibles ("floutage" de l'image la mieux résolue, action sur la radiométrie). Cela n'est toutefois pas toujours suffisant, et, si les images restent trop différentes, l'appariement échoue.

Des tests complémentaires effectués dans les mêmes conditions avec d'autres détecteurs de points d'intérêt (MSER [MCUP02], Harris-Affine et Hessian-Affine [MS05]) n'ont pas donné de meilleurs résultats.

Deux autres solutions seraient néanmoins envisageables pour l'application visée :

- Il est en effet toutefois possible d'extraire des points objets de liaison entre clichés aériens et images terrestres acquises par Stéréopolis à l'aide des outils décrits dans [Tou07] et [Soh08], qui présentent en effet des algorithmes de détection et de reconstruction tridimensionnelle de marquages routiers à partir d'images aériennes haute résolution pour [Tou07] ou d'images de bancs stéréoscopiques axiales de

- Séréopolis pour [Soh08]. Ces points, liés aux marquages routiers, se situent non plus sur les façades des bâtiments, mais au niveau du sol et présentent beaucoup moins de problèmes de déformation.
- Une autre solution consiste à ne plus apparier les points indépendamment. En effet, chaque point subit des déformations locales importantes du fait des textures 3D de la façade liées à la forte différence de points de vue. Il est donc possible d'apparier directement les textures [Ben06] et d'estimer la translation qui permettra ensuite de pouvoir apparier (avec intégration de cette solution initiale) les points d'intérêt de manière plus focalisée.

## 17 Appariement entre images terrestres acquises par Stéréopolis dans le but d'améliorer leur géoréférencement

### 17.1 Introduction

Il s'agit d'essayer cette fois de déterminer des points de liaison entre images terrestres acquises par le véhicule Stéréopolis (figure 108). Ces points vont ensuite être utilisés par une compensation par faisceaux afin d'améliorer le géoréférencement de ces images lors de la constitution de la base de données infrastructure.

FIGURE 108 – À gauche, appariement brut. À droite, appariement après filtrage par calcul d'une homographie.



### 17.2 Motifs répétitifs et changement de point de vue

Entre deux images consécutives (clichés  $n$  et  $n + 1$ ), l'appariement se passe généralement bien ainsi que l'illustre l'exemple ci-dessous (figure 109). Le changement de point de vue entre les deux images n'est pas suffisamment important pour réellement poser problème.

FIGURE 109 – Entre deux images consécutives (clichés  $n$  et  $n + 1$ ), l'appariement se passe généralement bien.



FIGURE 110 – Dès l'appariement entre deux images  $n$  et  $n + 2$ , des problèmes peuvent se poser.



Par contre, l'appariement des points entre les images  $n$  et  $n + 2$  commence déjà bien souvent à poser problèmes comme dans le cas suivant (figure 110).

Ces erreurs d'appariement viennent du fait que les façades de bâtiments sont généralement composées d'éléments plus ou moins répétitifs (fenêtres, balcons, motifs ornementaux).

Tant que le changement de point de vue entre les images à apparier n'est pas trop important, les déformations causées par la perspective restent limitées et la mise en correspondance des points homologues se passe donc

bien. À l'inverse, lorsque ce changement de point de vue devient plus important (à partir de  $n-n+2$ ), ces déformations deviennent suffisantes pour que, lors de l'appariement, un point de l'image 1 correspondant à un détail répétitif soit apparié, non pas avec son véritable homologue (qui apparaît alors de manière trop déformée), mais avec un détail identique apparaissant du fait du changement de point de vue dans l'image 2 avec quasiment le même aspect que celui de notre point dans l'image 1.

L'utilisation d'un prédicteur permet néanmoins d'obtenir de nombreux appariements corrects comme le montre la copie d'écran ci-dessous (figure 111). Le prédicteur a été défini "à la main" dans cet exemple, mais il aurait parfaitement pu être obtenu à partir du géoréférencement approché des images ou par calcul en utilisant les appariements entre les images  $n$  et  $n+1$ , et entre les images  $n+1$  et  $n+2$ .

FIGURE 111 – Restriction de la zone de recherche.



**Remarque :** Travailler sur des images redressées peut ici aussi permettre d'améliorer les résultats.

**Remarque :** Une alternative possible consisterait à réaliser dans un premier temps des appariements multiples (par exemple par une approche comme celle de [RDG08]), puis d'y éliminer (par un critère géométrique) les faux points homologues. [Cl9] présente une telle méthode basée sur des appariements multiples : il ne s'agit alors plus de déterminer pour un point de l'image 1 son homologue dans l'image 2, mais de rechercher tous ses homologues raisonnables dans l'image 2 (autrement dit, les points de l'image 2 avec lesquels la distance des descripteurs est relativement proche de celle obtenue pour le meilleur candidat). Le nombre d'appariements corrects obtenus augmente alors, mais celui des appariements erronés également. Les mises en correspondances les plus improbables peuvent toutefois être rejetées d'emblée en utilisant des informations comme la distance entre les descripteurs, le nombre de candidats homologues trouvés ainsi que le rapport à la distance entre le point à appairer et le meilleur candidat. Là aussi, l'utilisation d'un prédicteur permet d'éliminer la plupart de ces erreurs.

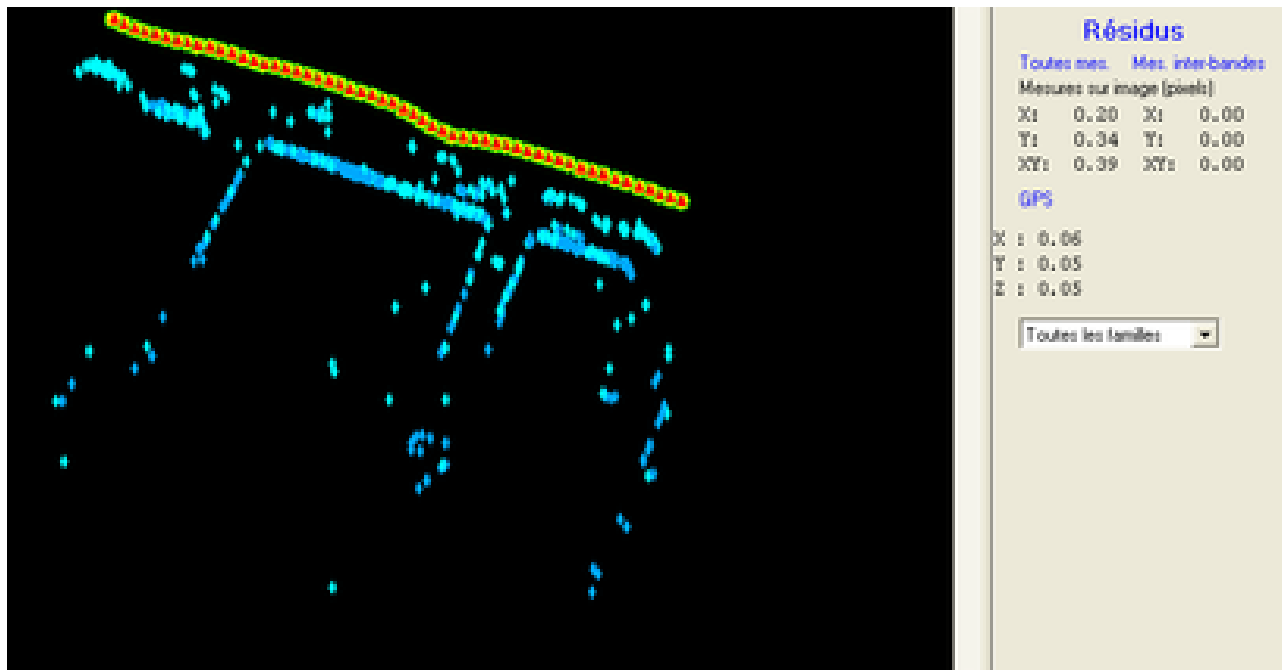
### 17.3 Compensation

Un calcul a été effectué sur les images de Stéréopolis fronto-parallèles aux façades dans la rue Soufflot. Ce cas de figure (notion de bande avec des rotations très proches de l'identité entre clichés consécutifs) est analogue à celui habituellement rencontré en aérien. Le calcul a d'ailleurs été effectué à l'aide du logiciel TopAéro



(compensation par faisceaux par moindres carrés) de l'IGN utilisé en production pour ce type de chantiers (voir copie d'écran de la figure 112). Une solution approchée est fournie par le géoréférencement direct du véhicule (GPS+INS), les coordonnées GPS des sommets de prise de vue sont d'ailleurs également considérées comme des observations lors de la compensation.

FIGURE 112 – Compensation dans TopAero



Pour ce calcul, seuls des appariements entre images consécutives ont été effectués. À l'issue de l'étape d'appariement et de filtrage des erreurs, tous les points homologues ne sont pas conservés. Ces points ne sont en effet pas répartis de manière homogène et sont souvent trop nombreux. La méthode de sélection des points de liaison destinés à être utilisés lors de la compensation est ici la même que celle utilisée précédemment dans le cas de l'aérien. Elle consiste d'abord à charger l'ensemble des appariements (connus jusqu'ici paire d'images par paire d'images) puis à attribuer à chaque point ("objet") de liaison un identifiant et à calculer sa multiplicité (nombre d'images sur lequel il est vu). (Remarque : des fautes peuvent également être détectées à l'issue de ce chaînage des appariements en s'apercevant qu'au final deux points distincts d'une même image ont reçu le même identifiant et sont donc supposés correspondre à un même point physique.)

On se donne ensuite sur chaque image une grille régulière dans chaque case de laquelle un seul point est conservé : un des points de plus forte multiplicité. On notera que les images (non clefs) acquises lors d'un arrêt du véhicule ont été exclues du calcul.

Les illustrations suivantes (figures 114, 115 et 115) montrent les résultats de la projection dans une image aérienne (préalablement orientée de manière indépendante des images terrestres) des points (objet) 3D calculés à partir de tous les points de liaison SIFT entre les images Stéréopolis fronto-parallèles aux façades en utilisant les orientations de ces images obtenues à l'issue de la compensation.

FIGURE 113 – Reconstruction (3D) des points de liaison extraits entre les images Stéréopolis : on reconnaît bien la structure de la rue

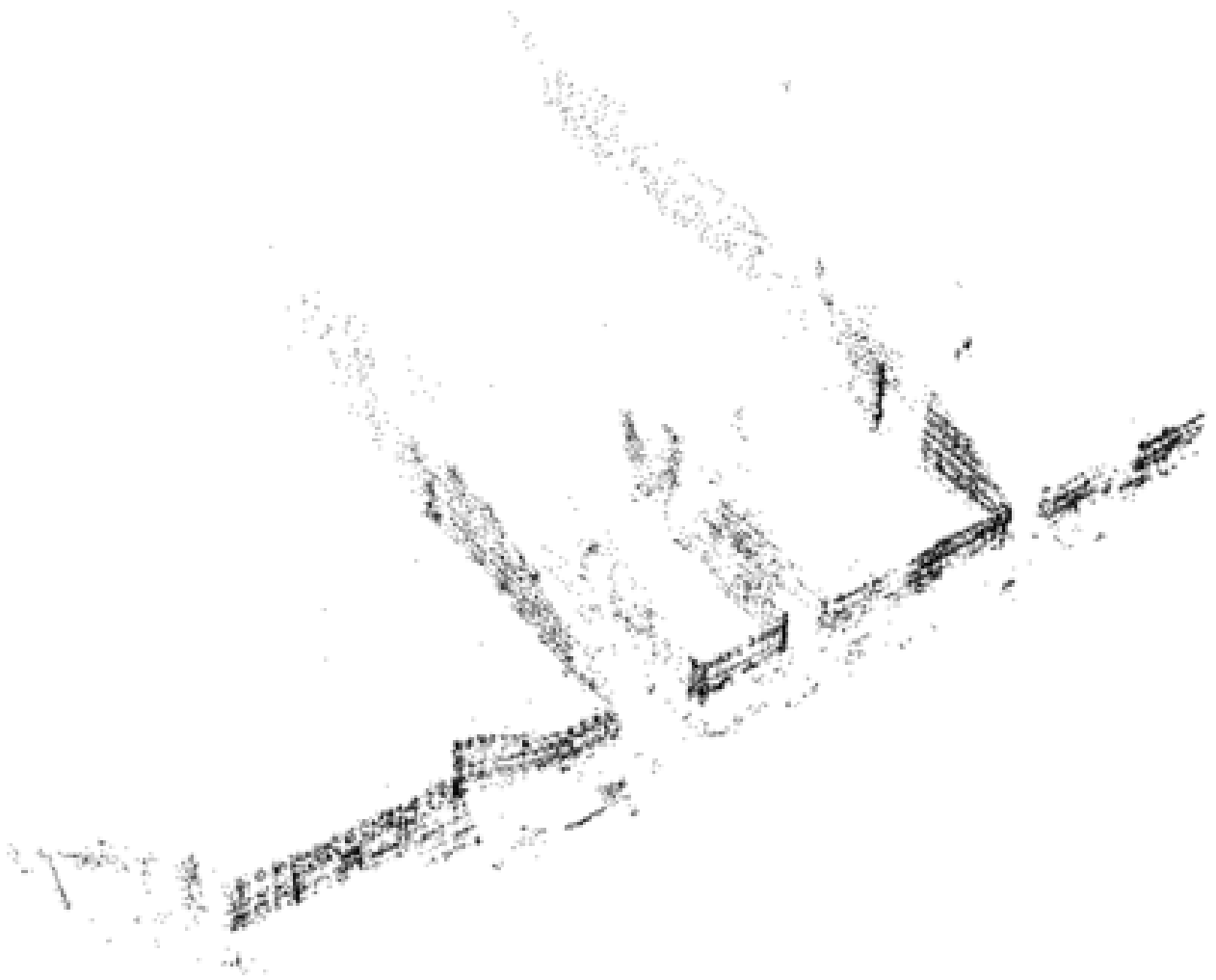


FIGURE 114 – Projection dans l'image aérienne de points objet 3D calculés à partir des points de liaison entre images Stéréopolis

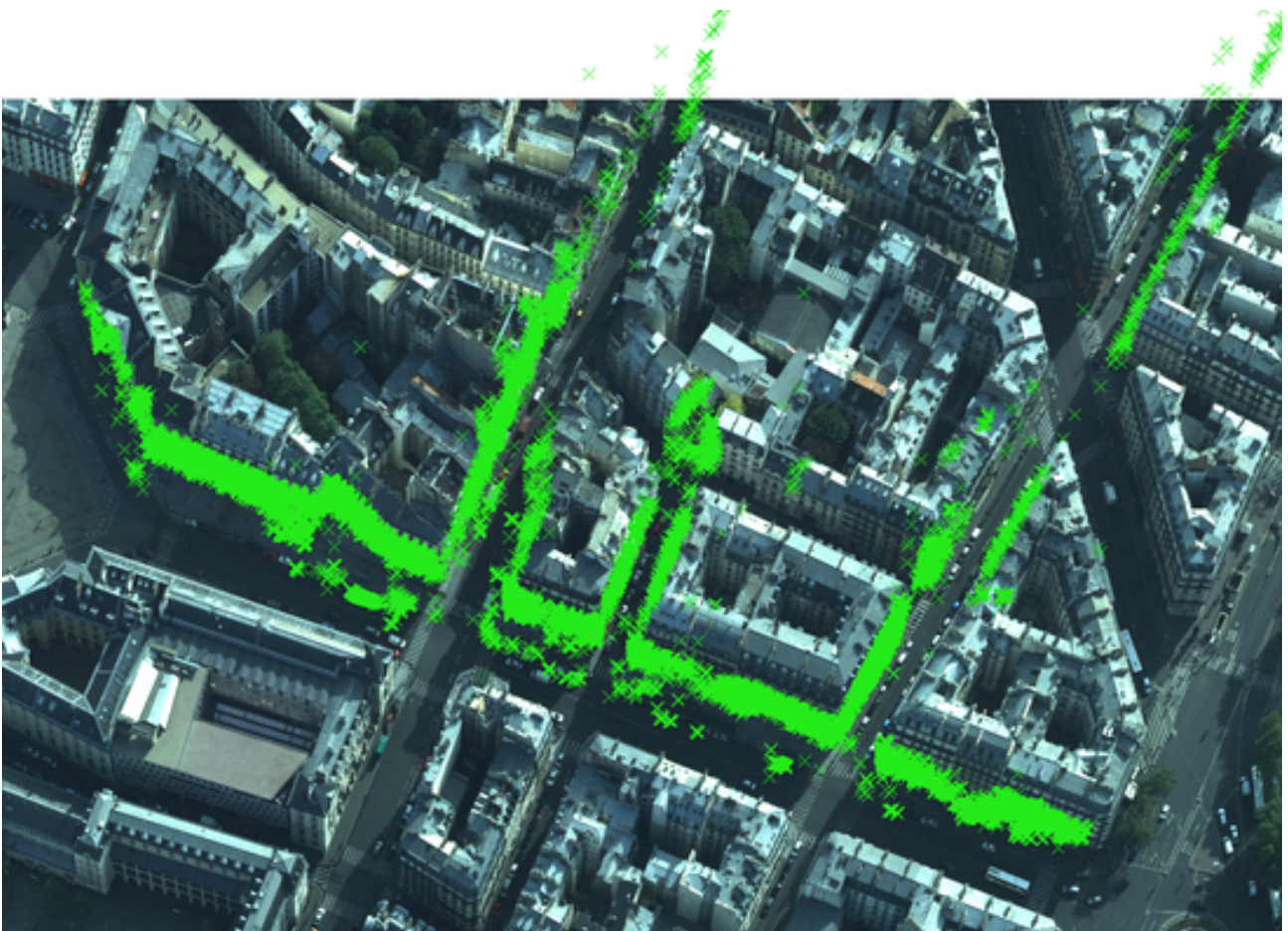


FIGURE 115 – Projection dans l'image aérienne de points objet 3D calculés à partir des points de liaison entre images Stéréopolis

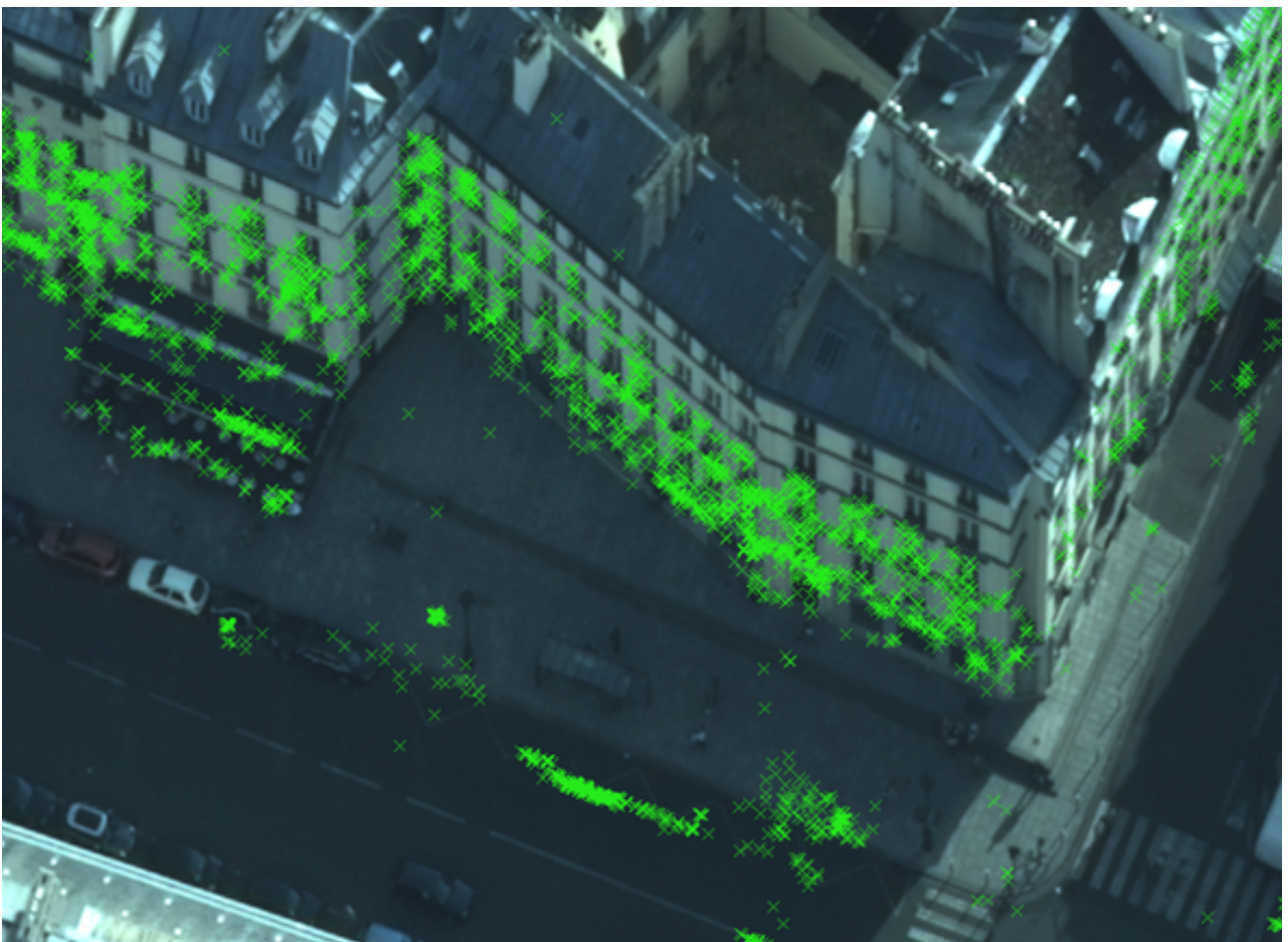


FIGURE 116 –

**À gauche** : Projection dans l'image aérienne de points 3D calculés à partir des points de liaison entre images Stéréopolis en utilisant le géoréférencement direct (GPS,INS) de ces images.

On observe un décalage.

**Au centre** : Points de liaison sur une des images du véhicule

**À droite** : Projection dans l'image aérienne de points 3D calculés à partir des points de liaison entre images Stéréopolis en utilisant le géoréférencement de ces images obtenu à l'issue de la compensation.

Le décalage est réduit.



## 17.4 Conclusion

La détermination de points de liaison entre images terrestres acquises par le véhicule Stéréopolis et fronto-parallèles aux façades des bâtiments se déroule plutôt bien. On se trouve en effet cette fois, malgré l'existence de motifs répétitifs, dans une configuration plutôt favorable puisque les images à mettre en correspondance sont du même type et ont été acquises au même moment (il n'y a pas de diachronisme).

Entre images consécutives ( $n-n+1$ ), le changement de point de vue n'est pas très important et l'appariement brut donne directement de bons résultats. A l'inverse, lorsque le changement de point de vue devient plus important (à partir de  $n-n+2$ ), les déformations de la texture de la façade associées à l'existence de motifs répétitifs deviennent suffisantes pour causer des erreurs d'appariement en nombre important. Des solutions présentées ci-dessus (prédicteur, appariement multiple...) peuvent toutefois permettre de régler ces problèmes.

On observe une amélioration du géoréférencement des images à l'issue de la compensation par faisceaux effectuée en utilisant les points de liaison fournis par SIFT. Cette amélioration concerne ici vraisemblablement davantage la calibration du véhicule (c'est-à-dire la position et l'orientation de la caméra dans le référentiel véhicule) que son géoréférencement absolu (position et orientation du véhicule dans le référentiel "objet").

## 18 Appariement d'images terrestres d'un système nomade avec des images d'une BD infrastructure terrestre.

### 18.1 Introduction

Il s'agit dans cette partie d'apparier des clichés acquis depuis le sol par un système nomade avec des images d'une base de données infrastructure d'images terrestres, acquises dans notre cas par le véhicule Stéréopolis. Ce système nomade pourrait être un appareil photo grand public ou un smartphone utilisé par un piéton. Aussi, le système nomade est-il dans la suite parfois dénommé "piéton". L'objectif des expériences qui suivent est donc de déterminer des points homologues permettant de recaler ces images acquises par un système nomade (images "requêtes") sur des images de la base de données.

On considère dans les tests suivants que l'on dispose déjà d'une information permettant de savoir quelles sont les images de la base de données sur lesquelles on va recaler l'image "requête" acquise par le système nomade. Cette information pourrait provenir d'une position approchée de l'image "requête" soit indiquée "manuellement" (par exemple par sélection interactive sur une carte de la zone acquise), soit fournie par le système nomade si celui-ci dispose d'outils de positionnement, c'est-à-dire d'un GPS et si possible de capteurs donnant une orientation absolue approchée, comme c'est désormais notamment le cas des smartphones les plus modernes. Par ailleurs, des outils d'indexation d'images existent et pourraient être utilisés pour permettre la recherche dans la base de données des meilleures images homologues possibles.

**Quelle utilité pour de tels appariements ?** Il pourrait être intéressant de pouvoir géoréférencer des images acquises par un système nomade de manière cohérente par rapport aux images de la base de données infrastructure pour plusieurs raisons :

- pour enrichir cette base de données (ou une base de données complémentaire en covisualisation), avec plus d'images de meilleure résolution en certains points (édifices remarquables, ...). Dans ce cas, ces nouvelles images doivent avoir un géoréférencement cohérent avec celui des images déjà présentes dans la base de données.
- pour mettre à jour ponctuellement cette base de données. L'utilisation d'un système nomade permettrait de n'acquérir de nouvelles images que là il y a eu un changement. On évite alors de devoir faire revenir le véhicule.
- pour se localiser à partir des images de la base de données...

### 18.2 Difficultés

Ces appariements vont se trouver face à plusieurs difficultés :

- **Diachronisme** : les deux groupes d'image n'ont pas été acquis au même moment (ni le même jour, ni à la même heure). L'aspect des images peut donc être très différent avec un éclairage, un contraste (parfois inversé) et des ombres (causées par des bâtiments et des motifs architecturaux des façades) différents, mais aussi avec la présence d'autres changements comme des fenêtres ouvertes puis fermées, des objets déplacés et des éléments mobiles (piétons, voitures, échafaudages, ...), mais aussi de "véritables" changements (changements d'enseignes pour un commerce, ...).
- Les **points de vue** (i.e. les angles et sommets de prise de vue) sont **différents**. Or on a déjà vu précédemment que l'association de la **présence d'éléments répétitifs** sur les façades et les changements de point de vue conduisent souvent à des erreurs d'appariement.
- **Variation d'échelle** : les deux types d'images n'ont pas la même résolution. Ce phénomène accentue encore le phénomène précédent.

La détermination de l'orientation des images est elle aussi difficile : on risque en effet de se trouver le plus souvent dans un cas dégénéré avec des points de liaison possibles se trouvant très majoritairement sur un même plan (celui de la façade des bâtiments de la rue). Toutefois, les algorithmes d'orientation relative de [Kal09] peuvent gérer ce type de configuration.

### 18.3 Première méthode : extraction des points homologues avec filtrage "géométrique" des fautes

La méthode mise en œuvre ici reste classique : extraction de points d'intérêt SIFT pour chaque image, recherche de points homologues par paire d'images, élimination des fautes comme outliers du calcul d'une trans-

formation (affinité, homographie vue que l'on se trouve le plus souvent avec une scène plane : la façade d'un bâtiment) entre les images.

Les résultats obtenus sont assez variables pour le jeu test étudié (constitué, rappelons-le, d'images acquises par un appareil photo grand public par un piéton que l'on souhaite recalcr sur des images Stéréopolis).

**Remarque** : Comme cela a déjà mentionné plus haut, il faut se méfier du seuil d'élimination des points les moins contrastés du détecteur SIFT. En effet, dans notre cas, les images Stéréopolis sont plus sombres et moins contrastées que celles acquises par le piéton. Dans les expériences qui suivent, ce filtrage est alors quasiment désactivé.

### 18.3.1 Cas favorable : bons résultats directement

Dans les deux exemples qui suivent (figures 117, 118, 119 et 120), l'appariement se passe bien : des points homologues majoritairement corrects et assez bien répartis sont détectés en nombre suffisamment important pour éliminer les fautes.

FIGURE 117 – Appariements bruts



FIGURE 118 – Recalage (à gauche) par un modèle d'homographie 2D (calculé à partir des points homologues détectés) de l'image acquise par le piéton sur l'image de la base de donnée (à droite)

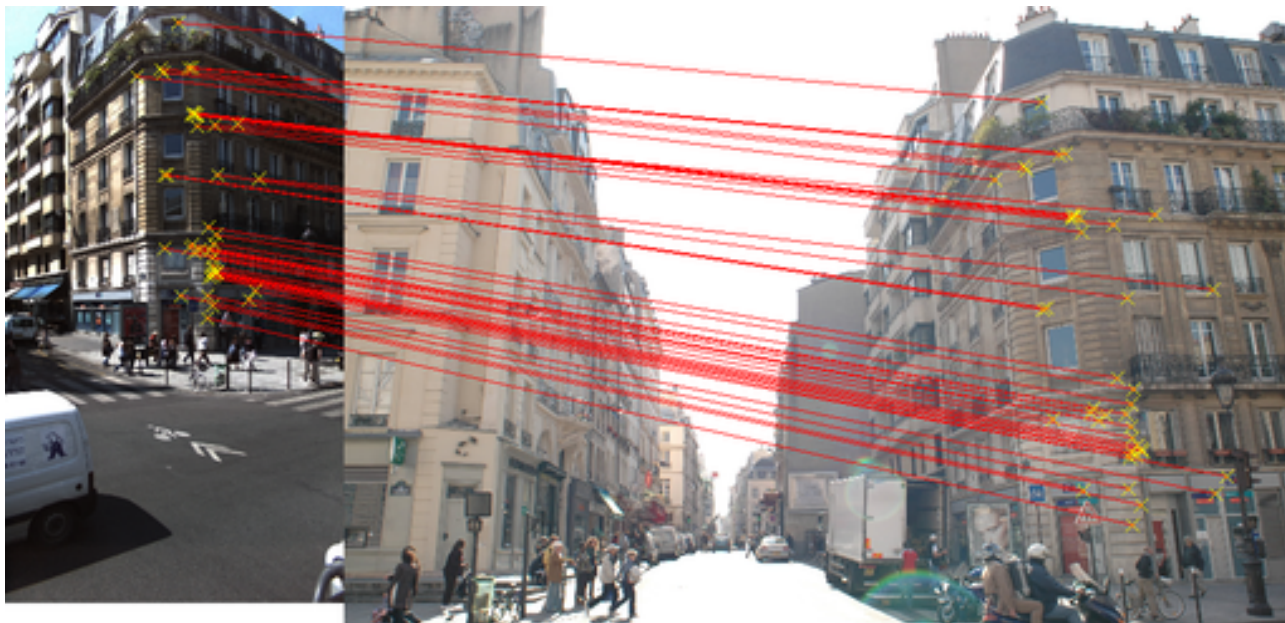


FIGURE 119 – Appariements bruts





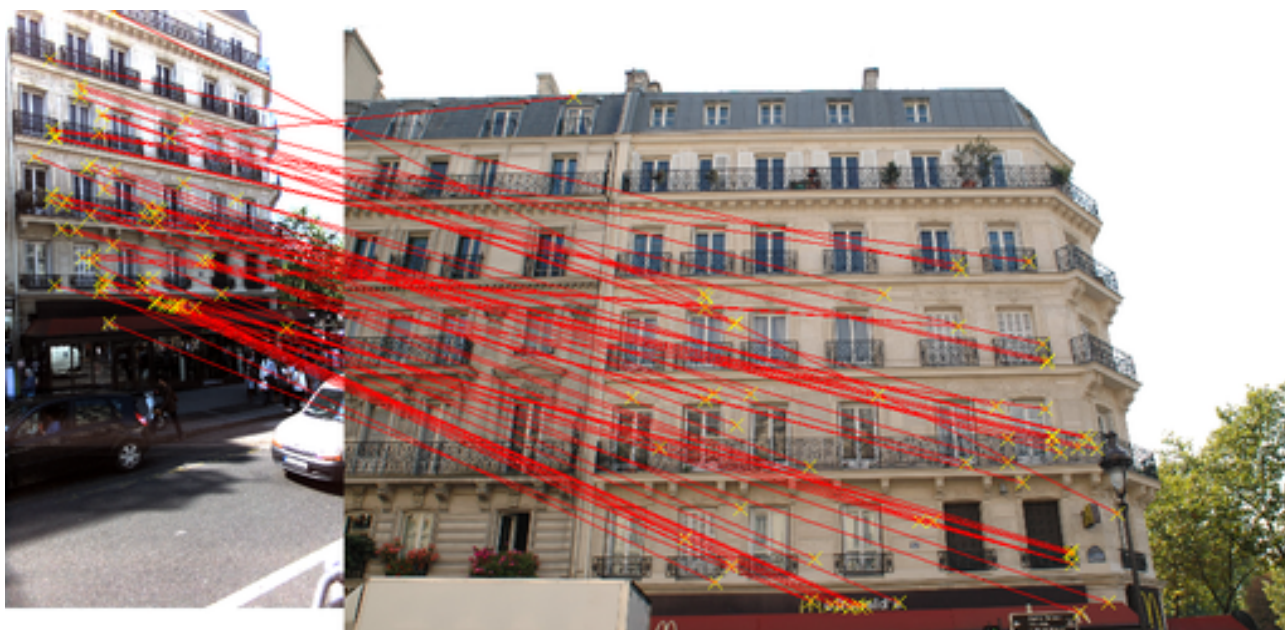
FIGURE 120 – Appariements après filtrage des fautes



### 18.3.2 Mauvais résultats par appariement direct, mais bons résultats en redressant les images

Dans l'exemple qui suit (figure 121), l'appariement direct ne donne pas de bons résultats. Ainsi que l'illustre la figure ci-dessous, les appariements obtenus sont généralement erronés. Précisons que de meilleurs résultats sont obtenus en utilisant une autre image du véhicule présentant un changement de point de vue moins important. Il semble donc que dans ce cas, ce soit l'association d'un fort changement de point de vue entre les images associé à la présence de motifs répétitifs qui soit la principale cause des mauvais résultats obtenus en ne prenant aucune précaution.

FIGURE 121 – Appariements bruts (à gauche : véhicule / à droite : piéton)



Par conséquent, il serait intéressant d'obtenir des images présentant un point de vue plus semblable. La solution retenue a donc consisté à redresser les images (ainsi qu'on l'avait déjà fait dans le cas des appariements aérien-terrestre). Le redressement a cette fois encore été effectué manuellement, mais il peut être effectué à l'aide des outils décrits dans [KJPG08] et [Kal09] qui utilisent les points de fuite de l'image. Un nombre satisfaisant de points de liaison (présentant par ailleurs une bonne répartition) est alors détecté entre les deux images redressées, ainsi qu'il est possible de le constater sur les deux illustrations suivantes (figure 122).

FIGURE 122 – Appariements bruts entre images redressées (à gauche : véhicule / à droite : piéton)



### 18.3.3 Résultats peu concluants, même en redressant les images

Dans l'exemple suivant (figure 123), les résultats de l'appariement restent mauvais même en redressant les images.

Ceci est probablement là encore causé par la présence d'éléments répétitifs, mais aussi par l'existence de changements radiométriques sur la façades (et même d'inversion de contraste au niveau de certaines fenêtres).

FIGURE 123 – les résultats de l'appariement restent mauvais même en redressant les images. (à gauche : véhicule / à droite : piéton)



#### 18.3.4 Exemple de cas intermédiaires

Il existe bien entendu des cas intermédiaires par rapport aux trois cas présentés précédemment. On peut vérifier dans ces différents cas que l'on obtient généralement de meilleurs résultats (lors d'un appariement direct) avec l'image de la base de données acquises avec le point de vue le plus proche de celui du cliché à recaler.

On constate également que l'on détecte souvent quelques point homologues corrects, mais mal répartis ou insuffisamment nombreux pour filtrer les outliers.

Dans certains cas, on a également pu constater que même les points d'intérêt détectés (avant l'appariement) sont dès le départ différents d'une image à l'autre.

FIGURE 124 – Exemples de cas intermédiaires. (à gauche : véhicule / à droite : piéton)

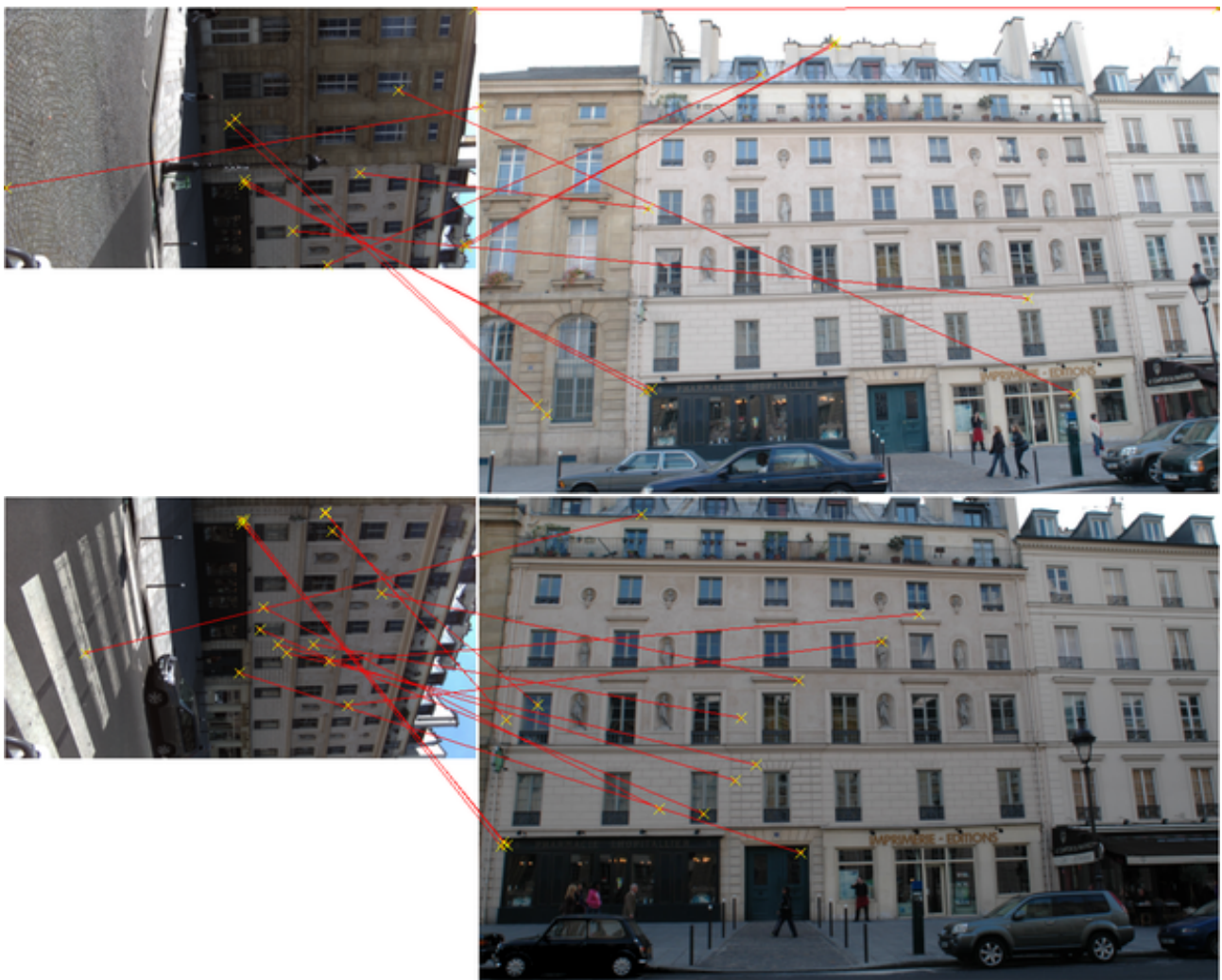
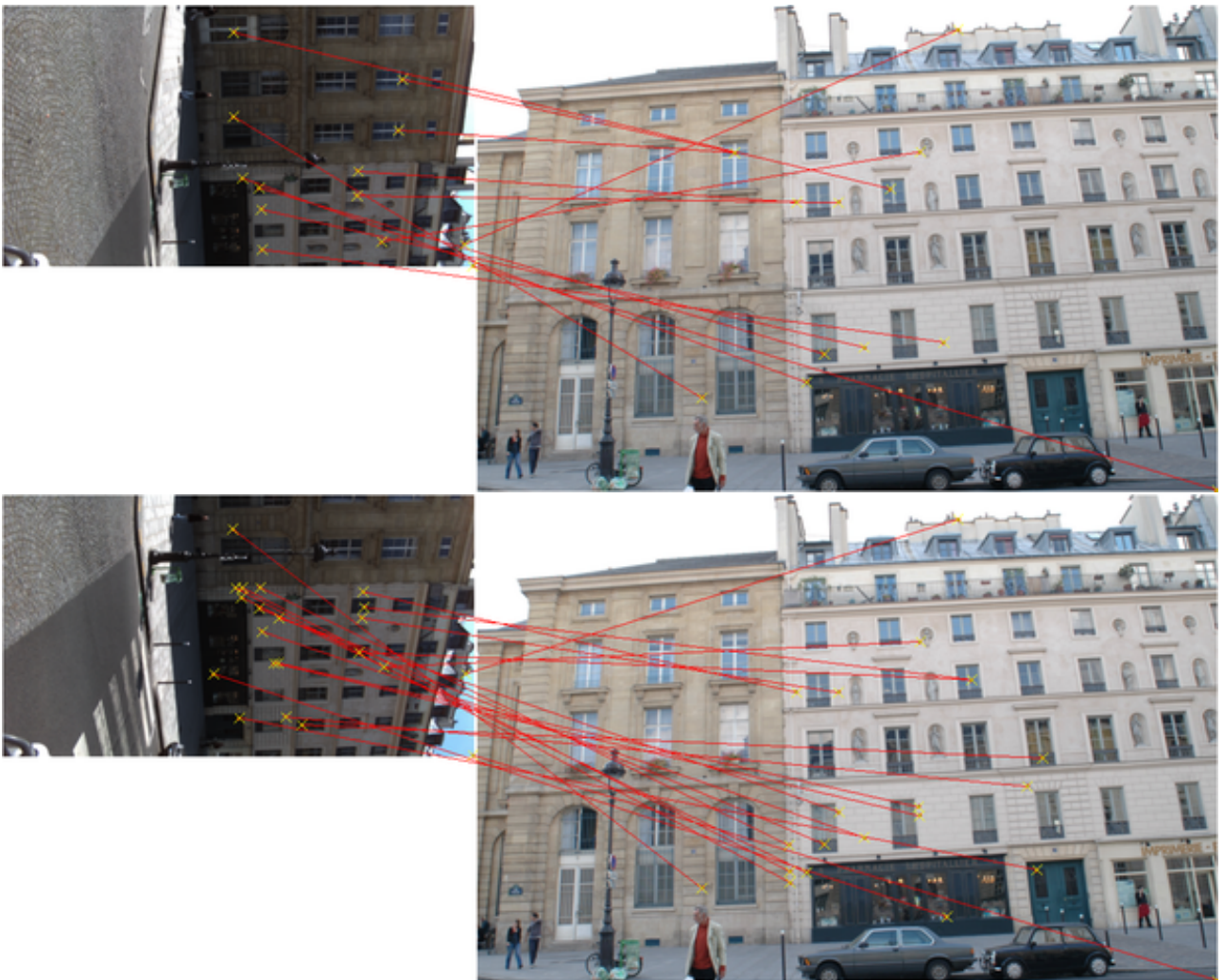


FIGURE 125 – Exemples de cas intermédiaires. (à gauche : véhicule / à droite : piéton)



### 18.3.5 Conclusion

Dans ce cas de figure d'un appariement entre images d'un système nomade avec celles d'une base de données infrastructure, du fait de nombreuses difficultés se cumulant (diachronisme, changement de point de vue, variations d'échelle, éléments répétitifs), il n'est pas forcément possible d'obtenir directement de bons points homologues, bien répartis et en nombre suffisant pour éliminer les outliers (et orienter l'image).

Néanmoins, comme pour l'appariement aérien-terrestre, dans plusieurs cas, le simple fait de redresser les images (et d'obtenir ainsi deux images acquises avec des "points de vue semblables") permet souvent d'améliorer les choses.

Cette solution ne fonctionne toutefois pas toujours. Par ailleurs, dans certains cas, on peut constater que même les points d'intérêt détectés sont dès le début différents d'une image à l'autre.

Des tests complémentaires effectués dans les mêmes conditions avec d'autres détecteurs de points d'intérêt (MSER, Harris-Affine, Hessian-Affine...) n'ont pas donné de meilleurs résultats.

Plusieurs alternatives seraient possibles :

- Une alternative possible consisterait à détecter dans un premier temps des appariements multiples, puis d'y éliminer (par un critère géométrique) les faux points homologues.
- [Cl9] présente une telle méthode basée sur des appariements multiples : il ne s'agit alors plus de déterminer pour un point de l'image 1 son homologue dans l'image 2, mais de rechercher tous ses homologues raisonnables dans l'image 2 (autrement dit, les points de l'image 2 avec lesquels la distance des descripteurs est relativement proche de celle obtenue pour le meilleur candidat). Le nombre d'appariements corrects obtenus augmente alors, mais celui des appariements erronés également. Les mises en correspondances les plus improbables peuvent toutefois être rejetées d'emblée en utilisant des informations comme la distance entre les descripteurs, le nombre de candidats homologues trouvés ainsi que le rapport à la distance entre le point à apparier et le meilleur candidat. Les faux points homologues extraits lors des appariements multiples sont éliminés à l'aide de variante de l'algorithme RANSAC. Là aussi, l'utilisation d'un prédicteur permet d'éliminer la plupart de ces erreurs.
- des méthodes d'appariement *a contrario* autorisent les appariements multiples en fonction d'un critère statistique plus élaboré, comme dans [RDG08]. Mais qu'en serait-il des temps de calcul ?
- ASIFT [Mor09], ce qui permettrait de régler les problèmes liés aux changements de point de vue, aux déformations dues à la perspective. Néanmoins, cette méthode est plus lente. Or il est possible de faire plus simple ici, en redressant simplement les images dans le plan de la façade.

## 18.4 Deuxième méthode : extraction des points homologues, filtrage des fautes en utilisant toutes les informations

### 18.4.1 Description de la méthode

Jusqu'à présent :

- on a constaté que le fait de travailler sur des images redressées dans le plan de la façade améliorerait la détection de points homologues. Jusqu'à présent, ce redressement s'est fait à la main, mais il est parfaitement possible de l'automatiser en passant par une étape préalable de détection des points de fuite de l'image.

Néanmoins, dans certains cas, même en travaillant sur des images redressées, le nombre d'inliers détectés était insuffisant pour permettre de détecter les erreurs et de les éliminer...

- on a exploité uniquement les coordonnées images (2D) des points détectés pour le filtrage des fautes. Or, d'autres informations sont associées à ces points : leur échelle caractéristique et leur orientation, que l'on va pouvoir utiliser ici. En effet, si les images ont été au préalable redressées par rapport à un même plan, la transformation pour passer de l'une à l'autre est une similitude 2D. Or chaque appariement de points SIFT fournit une hypothèse de similitude si l'on exploite à la fois leurs coordonnées image  $(x, y)$ , mais aussi leur échelle et leur orientation caractéristique.

On en déduit donc la méthode suivante :

1. Redressement des images dans le plan de la façade
  - (a) Détection des points de fuite correspondant aux deux directions principales de la façade. Une méthode possible est celle proposée par [Kal09], mais il en existe bien d'autres.  
On en déduit la rotation entre le plan de la façade et le plan image.

- (b) Redressement - ortho-rectification de l'image dans le plan de la façade.
- 2. Recherche de points homologues entre les deux images redressées
  - (a) Extraction des points SIFT pour chacune des images redressées
  - (b) Appariement de ces points SIFT
  - (c) Elimination des faux appariements par la méthode décrite en 5.2.3 :
    - i. Estimation des paramètres d'une similitude 2D à partir des points mis en correspondance (méthode "accumulation")
    - ii. Elimination des outliers par rapport à ce modèle.
- 3. Projection des coordonnées des points mis en correspondance dans les images originales.

FIGURE 126 – Méthode mise en œuvre pour la détection de points d'intérêt

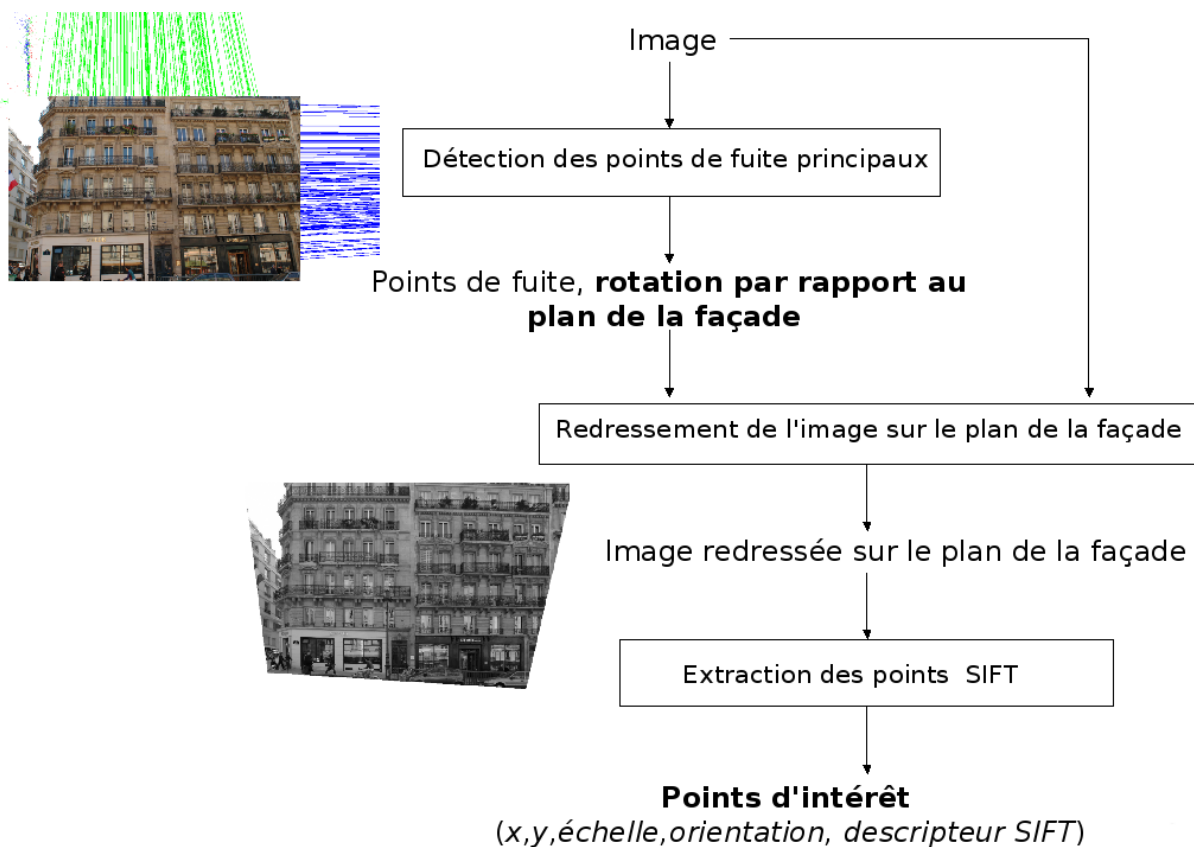
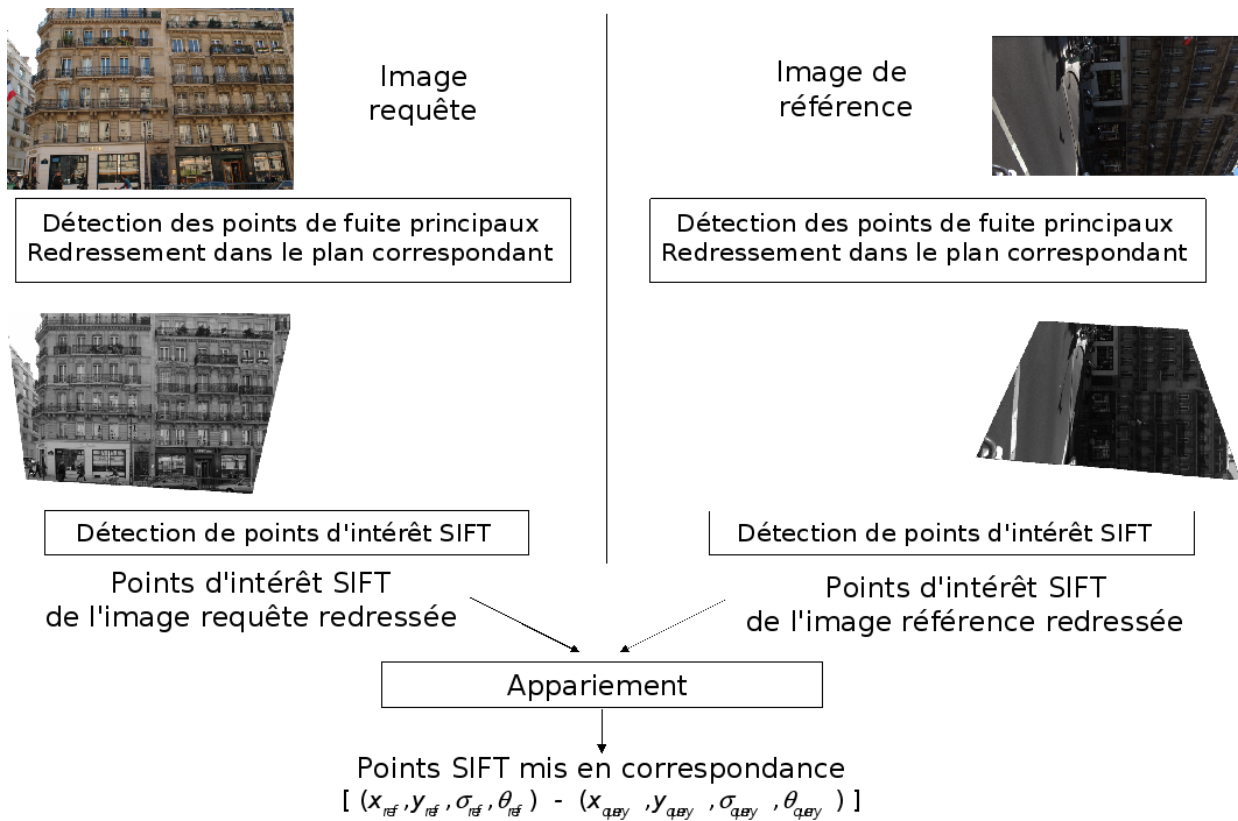


FIGURE 127 – Méthode mise en œuvre



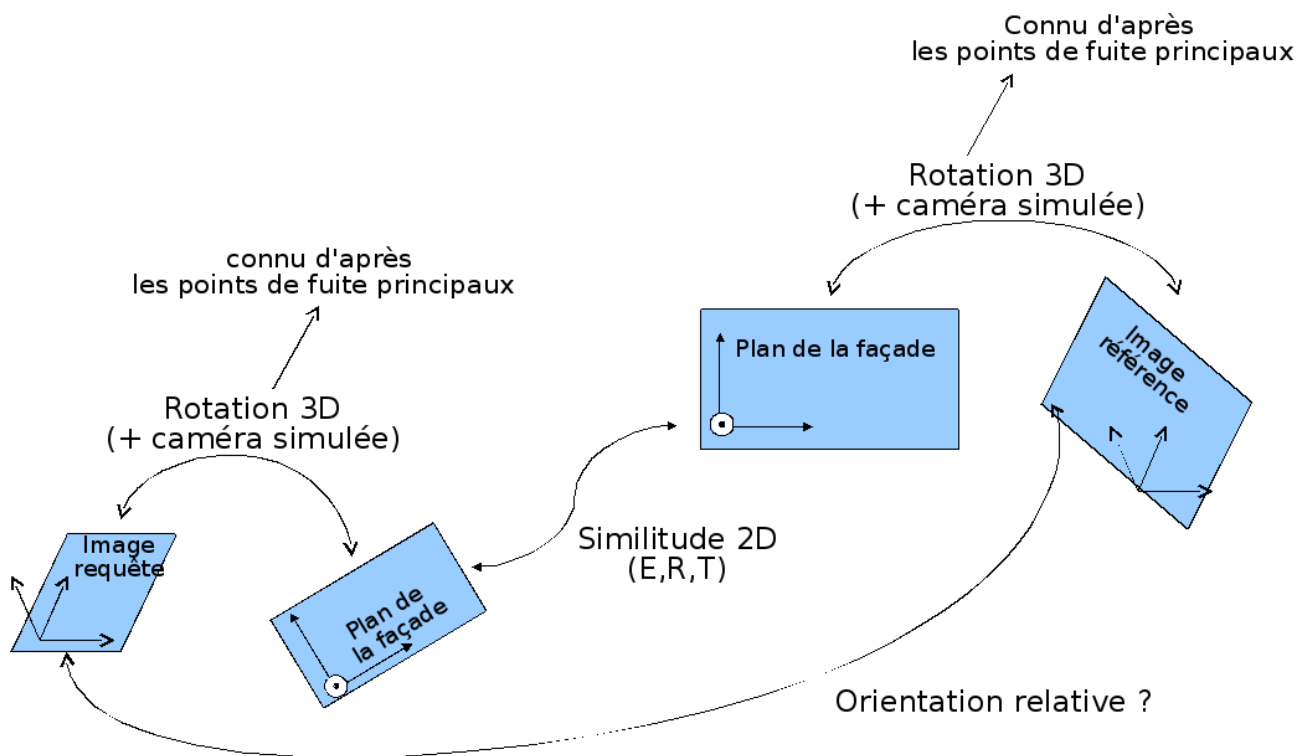
**Remarque 1 : ASIFT** Le fait de redresser les images permet d'avoir deux images qui se ressemblent davantage. L'algorithme ASIFT [Mor09] applique à basse résolution différentes affinités aux images et recherche ensuite des points de liaison entre ces images "déformées". Il identifie alors quelles sont les paires d'affinités permettant d'obtenir le plus de points homologues et relance alors la détection à pleine résolution uniquement pour ces quelques paires d'affinités.

Dans notre cas, on se trouve dans un cas particulier où l'on n'a pas besoin de tester toutes ces affinités, puisque l'on sait qu'il est possible de redresser les images dans le plan de la façade, et que cette information peut être calculée à partir des points de fuite principaux des images.

**Remarque 2 : Orientation relative** Cette méthode peut également permettre d'estimer une orientation relative approchée, ainsi que l'explique le schéma de la figure 128.

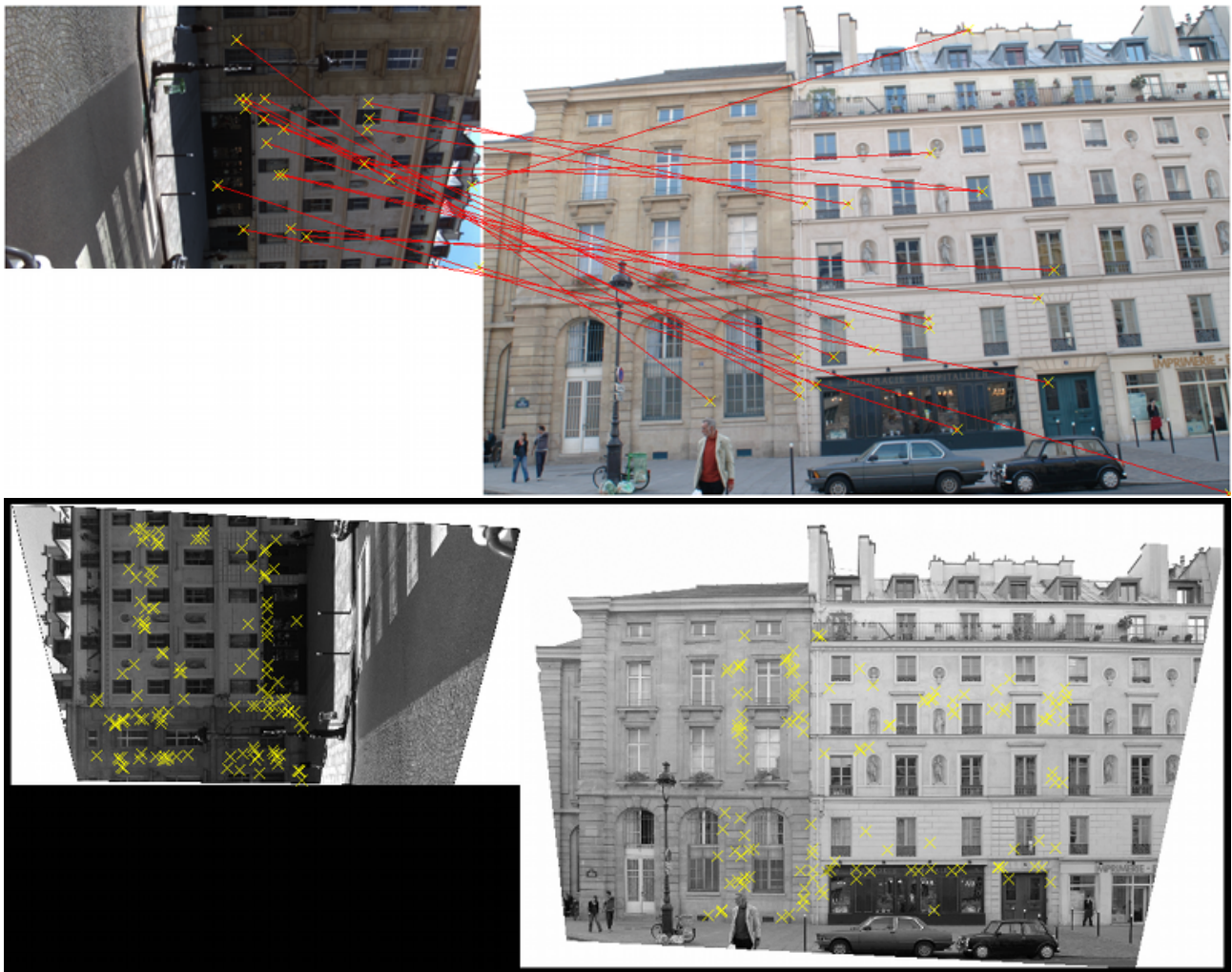


FIGURE 128 – Orientation relative



#### 18.4.2 Quelques résultats sur le jeu test précédent

FIGURE 129 – Points homologues détectés à partir des images originales (en haut) ou des images redressées (en bas)



Les illustrations suivantes (figures 130, 131, 132 and 133) montrent quelques exemples de résultats obtenus par la méthode proposée. A chaque fois, on a :

- en haut à droite, l'image “piéton” requête que l’on souhaite recalculer.
- en haut à gauche, deux images Stéréopolis (acquises par des caméras différentes) correspondant à l’image “piéton”.

Ces images “piéton” et Stéréopolis sont ici affichées à la même “résolution”. (Aucune n’a été agrandie ou réduite.)

- en bas, une image “fausse couleur” représentant la superposition de ces différentes images redressées dans le plan de la façade et recalculées, avec l’image “piéton” en rouge et les images Stéréopolis en bleu et en vert.

FIGURE 130 – Exemple de résultat obtenu



FIGURE 131 – Exemple de résultat obtenu

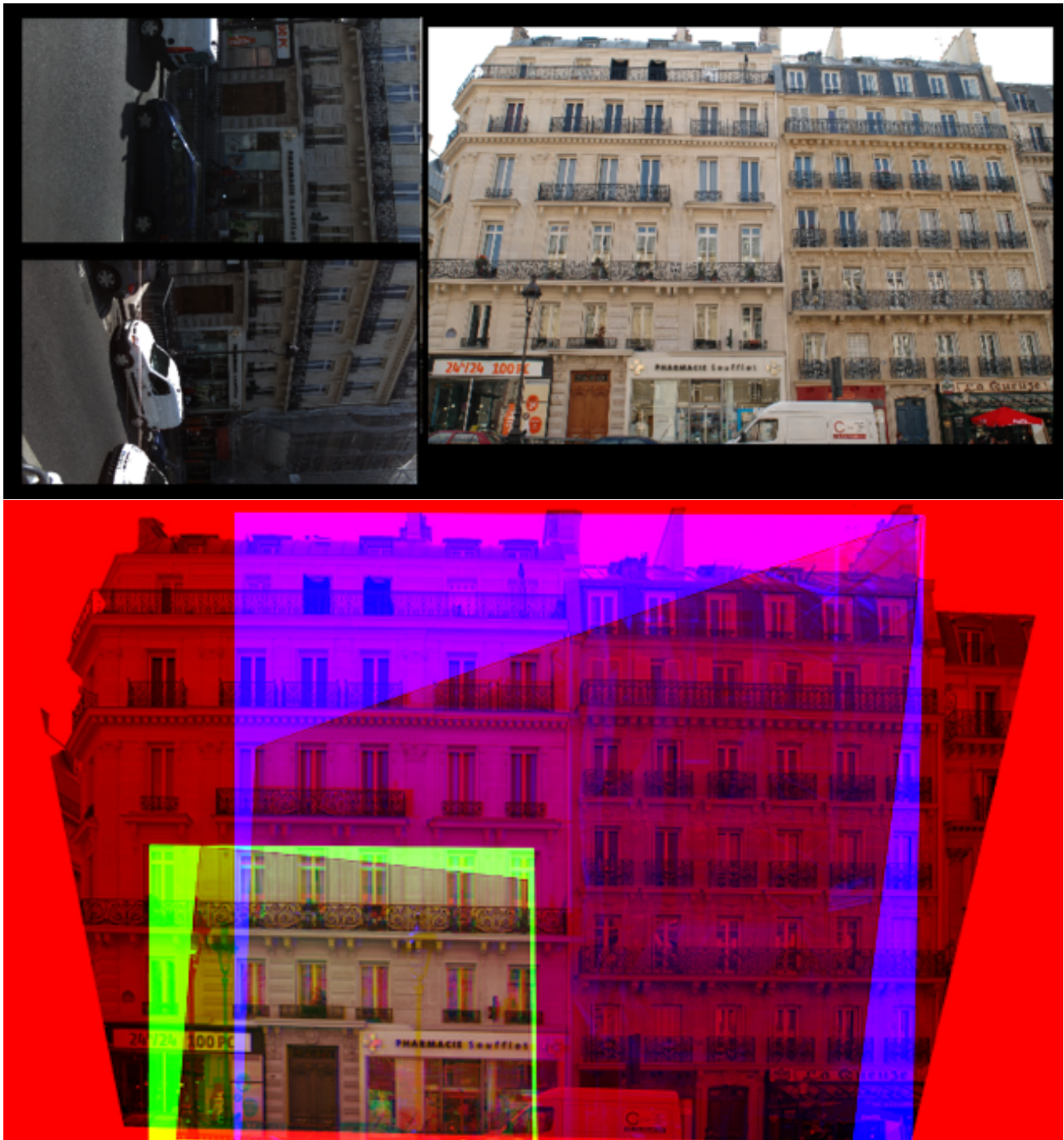


FIGURE 132 – Exemple de résultat obtenu

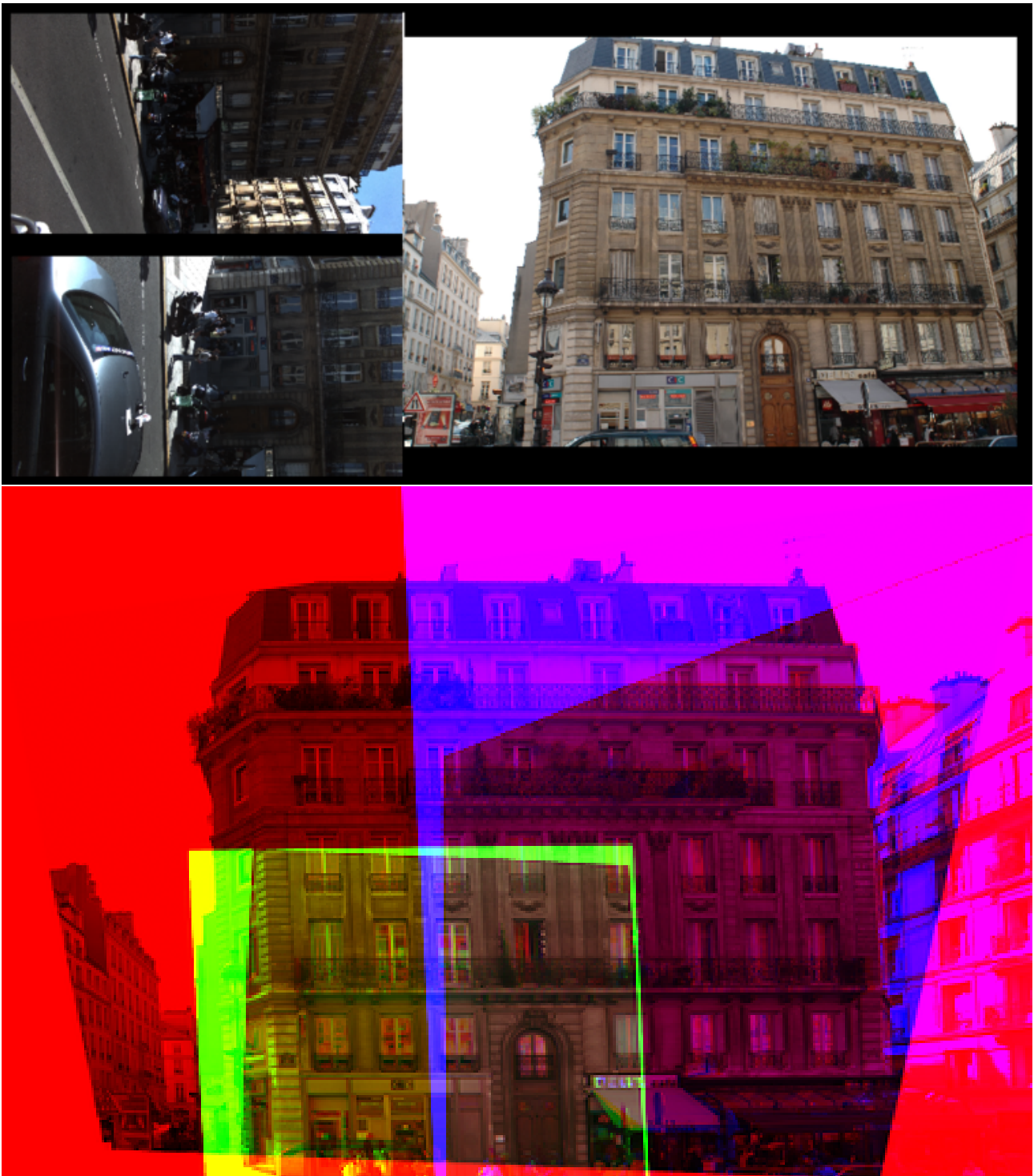
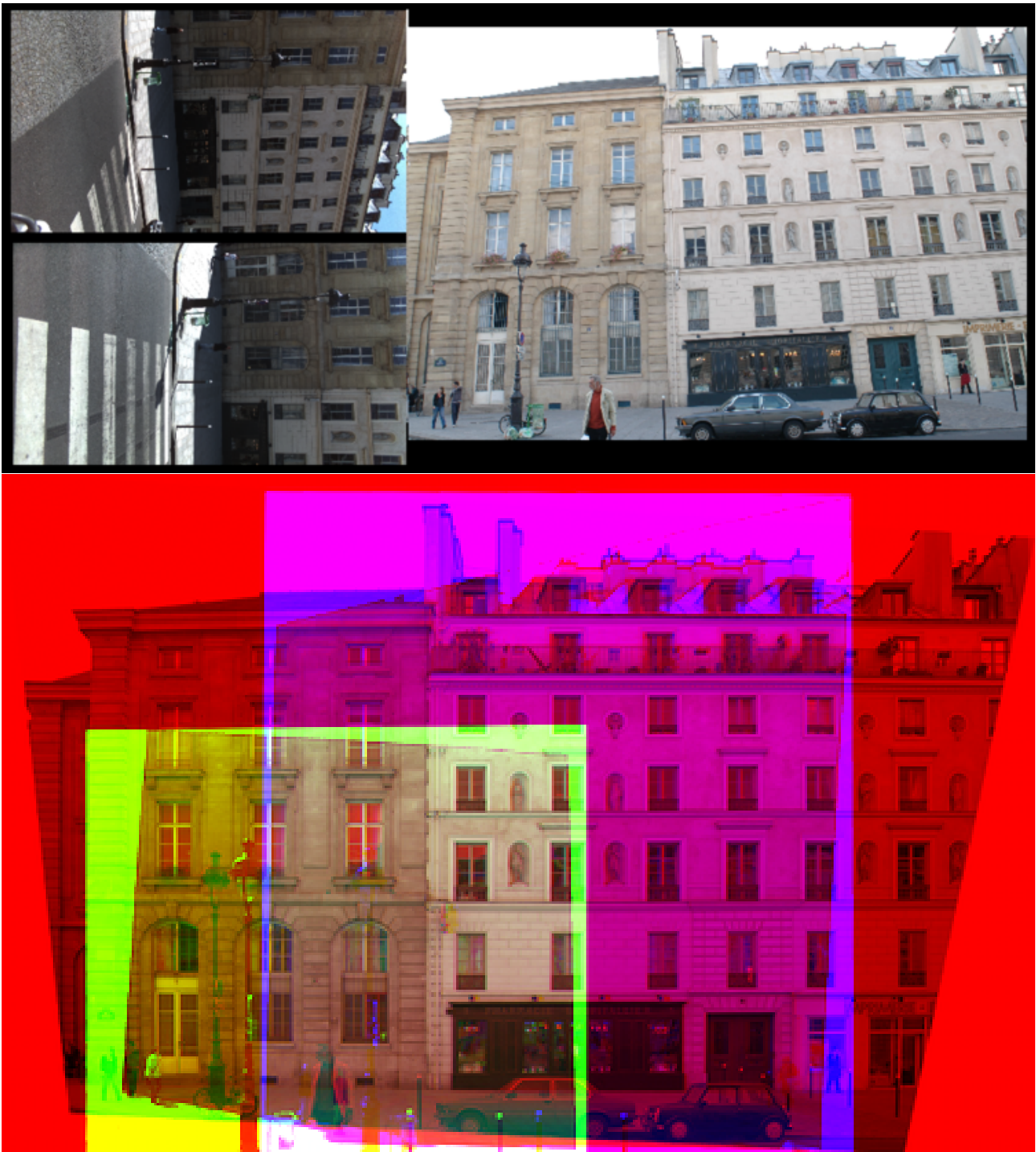


FIGURE 133 – Exemple de résultat obtenu



On constate que dans ces différents cas de figure, et même dans ceux qui posaient précédemment problème, on a réussi à détecter un certain nombre de points de liaison, à détecter les outliers et à détecter une transformation (approchée) entre les images.

#### 18.4.3 Quelques résultats sur un autre jeu test

D'autres essais ont été effectués sur un autre jeu de données test. Cette fois, les clichés à recaler ont été acquis par un smartphone. La méthode proposée précédemment donne de bons résultats dans certains cas (comme on

peut le voir sur la figure 134), mais échoue dans d'autres. On se trouve en effet dans un cas plus difficile, du fait de l'importante variation de résolution entre les 2 capteurs (comme on peut le constater sur la figure 135). Par ailleurs, dans ce jeu test, les images Stéréopolis avaient été acquises assez près des façades, il y a moins de recul, contrairement à certaines des images du smartphone, acquise depuis l'autre côté de la rue. On se trouve donc face à un facteur d'échelle important, qui se combinant avec les soucis déjà évoqués précédemment (diachronisme, objets répétitifs, ...) explique la difficulté dans certains cas à trouver des détails communs entre les deux images.

FIGURE 134 – Exemple de recalages obtenus (à droite) entre une image smartphone (à gauche, agrandie) et une image Stéréopolis (au centre)





FIGURE 135 – En haut : Exemple d'une image smartphone (à gauche) et d'une image Stéréopolis (au centre). Cette fois, l'image smartphone n'a pas été agrandie. En bas : à gauche - image Stéréopolis , à droite - image smartphone recalée sur l'image Stéréopolis



## 18.5 Conclusion

La seconde méthode proposée permet de bien améliorer les résultats, notamment dans le cas du premier jeu test. Dans le cas du second jeu test, de bons résultats sont également obtenus pour un certain nombre d'images tandis que dans d'autres cas, le processus échoue.

Comme dans le cas de l'appariement aérien-terrestre, travailler avec l'ensemble de la texture pourrait présenter des avantages : on n'apparierait alors plus des points indépendamment mais directement les textures comme dans [Ben06] afin d'estimer la transformation qui permettra ensuite de pouvoir apparier (avec intégration de cette solution initiale) des points d'intérêt de manière plus focalisée.

Lorsque l'on dispose de suffisamment de points de liaison bien répartis, il devient possible d'orienter les images "requêtes" par rapport aux images de la base de données notamment en utilisant les outils robustes présentés dans [Kal09], et finalement d'en déduire leur géoréférencement.



## Cinquième partie

# Références bibliographiques et glossaire

## A Glossaire

**SIFT** : Scale Invariant Feature Transform

**MSER** : Maximum Stable Extrema Region [MCUP02]

MSER est un détecteur de régions d'intérêts. Le barycentre d'une région détectée est ensuite considéré comme un point d'intérêt pour lequel un descripteur (SIFT ou autre) est calculé. La région détectée est modélisée par une ellipse et le descripteur sur une image déformée localement de manière à transformer cette ellipse en cercle. De ce fait, MSER est plus invariant aux affinités que SIFT. Par contre, MSER fournit généralement beaucoup moins de points.

**SURF** : Speeded-Up Robust Features [BETVG08]

Il s'agit d'une variante alternative à SIFT, comportant à la fois un détecteur de points d'intérêts et un descripteur. Ce détecteur de points d'intérêt est très rapide, du fait de son utilisation d'images intégrales. En revanche, il fournit moins de points que SIFT, ce qui pourrait peut-être se révéler gênant pour le calcul de points de liaison multiples en interbande sur des chantiers aériens (à vérifier...). Quelques tests ponctuels de SURF (non mentionnés dans ce document) ont par ailleurs été menés en utilisant l'implémentation libre OpenSURF [Eva09, Eva]. Il semblerait au vu de ces expériences que SURF soit un peu moins robuste que SIFT aux rotations (2D) de l'image, confirmant ainsi [JG10].

**ASIFT** : Affine SIFT [Mor09].

Il s'agit d'une adaptation de SIFT afin de le rendre invariant aux affinités même importantes. Contrairement à SIFT, ASIFT travaille directement sur une paire d'image. ASIFT applique à basse résolution différentes affinités aux deux images. Une modélisation particulière des affinités est utilisée, pour permettre une inspection intelligente des différents paramétrages possibles. ASIFT recherche ensuite des points de liaison entre ces images "déformées", en utilisant la méthode SIFT standard. Il identifie alors quelles sont les paires d'affinités permettant d'obtenir le plus de points et relance alors la détection à pleine résolution uniquement pour ces quelques paires d'affinités.

**PCA-SIFT** : une alternative au descripteur SIFT [KS04]

**TopAéro** : TopAéro est le logiciel d'aérotriangulation utilisé à l'IGN.

## B Bibliographie

### Références

- [AM93] S. ARYA et D. M. MOUNT : Approximate nearest neighbor queries in fixed dimensions. *In Proc. of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 271–280, Austin, USA, janvier 1993.
- [AMN<sup>+</sup>98] S. ARYA, D. M. MOUNT, N. S. NETANYAHU, R. SILVERMAN et A. Y. WU : An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
- [Ben06] O. BENTRAH : *Reconstruction de la géométrie d'acquisition de séquences d'images acquises par un véhicule en milieu urbain*. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, décembre 2006.
- [BETVG08] H. BAY, A. ESS, T. TUYTELAARS et L. VAN GOOL : SURF : Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.
- [Bjö08] M. BJÖRKMAN : A cuda implementation of sift. <http://www.csc.kth.se/~celle/>, 2008.
- [BL97] J. S. BEIS et D. G. LOWE : Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, Puerto Rico, juin 1997.
- [BL07] M. BROWN et D. G. LOWE : Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.
- [BSP07] J. BAUER, N. SÜNDERHAUF et P. PROTZEL : Comparing several implementations of two recently published feature detectors. *In Proc. of the International Conference on Intelligent and Autonomous Systems*, Toulouse, France, 2007.
- [Cha07] W. CHANGCHANG : SiftGPU : A GPU implementation of scale invariant feature transform (SIFT). <http://cs.unc.edu/~ccwu/siftgpu>, 2007.
- [Cl9] I. CLÉRY : Test d'algorithmes d'appariement robustes entre des images ambiguës et prises dans des conditions différentes. Mémoire de maîtrise, École Nationale des Sciences Géographiques, 2009.
- [Con09a] CONDOR TEAM : Condor, high throughput computing. <http://www.cs.wisc.edu/condor/>, septembre 2009.
- [Con09b] CONDOR TEAM : *Condor Version 7.2.4 Manual*. University of Wisconsin-Madison, août 2009.
- [Eva] C. EVANS : OpenSURF. <http://www.chrisevansdev.com>.
- [Eva09] C. EVANS : Notes on the opensurf library. Rapport technique CSTR-09-001, University of Bristol, janvier 2009.
- [GD05] K. GRAUMAN et T. DARREL : The pyramid match kernel : Discriminative classification with sets of image features. *In Proc. of the IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, octobre 2005.
- [Hes09] R. HESS : Sift feature detector. <http://web.engr.oregonstate.edu/~hess/index.html>, 2009.
- [ISNC05] L. IBANEZ, W. SCHROEDER, L. NG et J. CATES : *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, <http://www.itk.org/ItkSoftwareGuide.pdf>, second édition, 2005.
- [ITK09] The insight segmentation and registration toolkit, last visited on the 26th of March 2009. <http://www.itk.org>.
- [JG10] L. JUAN et O. GWUN : A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2010.
- [Kal09] M. KALANTARI : *Approche directe de l'estimation automatique de l'orientation 3D d'images*. Thèse de doctorat, Université de Nantes, France, septembre 2009.
- [KJPG08] M. KALANTARI, F. JUNG, N. PAPANODITIS et J.P. GUÉDON : Robust and automatic vanishing points detection with their uncertainties from a single uncalibrated image, by planes extraction on the unit sphere. *In International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS)*, volume XXXVII(3A), pages 203–208, Beijing, China, juillet 2008.
- [KS04] Y. KE et R. SUKTHANKAR : PCA-SIFT : a more distinctive representation for local image descriptors. *In Proc. of the conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

- [LF06] T. LÄBE et W. FÖRSTNER : Automatic relative orientation of images. *In Proc. of the 5th Turkish-German Joint Geodetic Days*, pages 1218–1225, Berlin, Germany, mars 2006.
- [Lie09] M. LIENOU : *Apprentissage automatique des classes d'occupation du sol et représentation en mots des images satellitaires*. Thèse de doctorat, Télécom ParisTech, Paris, France, mars 2009.
- [LMN09] A. LINGUA, D. MARENCHINO et F. NEX : Performance analysis of the SIFT operator for automatic feature extraction and matching in photogrammetric applications. *Sensors*, 9(5):3745–3766, 2009.
- [Low99] D. G. LOWE : Object recognition from local scale-invariant features. *In Proc. of the IEEE International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, septembre 1999. IEEE.
- [Low04] D. G. LOWE : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [Low05] D. G. LOWE : Demo software : SIFT keypoint detector. <http://www.cs.ubc.ca/~lowe/keypoints/siftDemoV4.zip>, juillet 2005.
- [MA09] D. M. MOUNT et S. ARYA : ANN : a library for Approximate Nearest Neighbor searching, last visited on the 22nd of March 2009. <http://www.cs.umd.edu/mount/ANN/>.
- [MCUP02] J. MATAS, O. CHUM, M. URBAN et T. PAJDLA : Robust wide baseline stereo from maximally stable extremal regions. *In Actes de British Machine Vision Conference (BMVC)*, volume XXXVI(3), pages 384–393, 2002.
- [Mor08] G. MOREL, J.M. et Yu : On the consistency of the SIFT method. Rapport technique CMLA 2008-26, Centre de Mathématique et de Leurs Applications (CMLA) UMR 8536, ENS Cachan, Cachan, France, septembre 2008.
- [Mor09] G. MOREL, J.M. et Yu : ASIFT : A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.
- [MS05] K. MIKOLAJCZYK et C. SCHMID : A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005.
- [MTS<sup>+</sup>04] K. MIKOLAJCZYK, T. TUYTELAARS, C. SCHMID, A. ZISSERMAN, J. MATAS, F. SCHAFFALITZKY, T. KADIR et L. VAN GOOL : A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2004.
- [Now05] S. NOWOZIN : libsift - scale-invariant feature transform implementation. <http://user.cs.tu-berlin.de/~nowozin/libsift/>, last visited on the 31st of October 2005.
- [RDG08] J. RABIN, J. DELON et Y. GOUSSEAU : A contrario matching of SIFT-like descriptors. *In ICPR*, pages 1–4, Tampa, Florida, USA, 2008.
- [Rem06] F. REMONDINO : Detectors and descriptors for photogrammetric applications. *In International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (IAPRS)*, volume XXXVI(3), pages 49–54, Bonn, Allemagne, septembre 2006.
- [SFPG06] S. N. SINHA, J.-M. FRAHM, M. POLLEFEYS et Y. GENÇ : GPU-based video feature tracking and matching. *In Proc. of the workshop on Edge Computing Using New Commodity Architectures*, Chapel Hill, USA, mai 2006.
- [Soh08] B. SOHEILIAN : *Roadmark reconstruction from stereo-images of a mobile mapping system*. Thèse de doctorat, Université Paris-Est, avril 2008.
- [TM08] T. TUYTELAARS et K. MIKOLAJCZYK : Local invariant feature detectors : a survey. *Foundations and Trends in Computer Graphics and Vision*, pages 177–280, 2008.
- [Tou07] O. TOURNAIRE : *Reconstruction 3D d'objets du réseau routier en milieu urbain à partir d'images haute résolution*. Thèse de doctorat, Université Paris-Est, décembre 2007.
- [TTL05] D. THAIN, T. TANNENBAUM et M. LIVNY : Distributed computing in practice : the Condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.
- [Ved07a] A. VEDALDI : An open implementation of the SIFT detector and descriptor. Rapport technique 070012, UCLA CSD, 2007.
- [Ved07b] A. VEDALDI : SIFT++ : a lightweight C++ implementation of SIFT detector and descriptor. <http://www.vlfeat.org/~vedaldi/code/siftpp.html>, 2007.
- [VF08] A. VEDALDI et B. FULKERSON : VLFeat : An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.

- [YN08] Y. YANG et S. NEWSAM : Comparing SIFT descriptors and gabor texture features for classification of remote sensed imagery. *In Proc. of the 15th IEEE International Conference on Image Processing*, pages 1852–1855, San Diego, USA, octobre 2008.





Laboratoire **MATIS**

**M**éthodes d'**A**nalyse et de **T**raitement d'**I**mages pour la **S**téréorestitution

Institut Géographique National,

73 Avenue de Paris, 94165 Saint-Mandé Cedex (France)

<http://recherche.ign.fr/labs/matis>

---