

Master 2 recherche Informatique et Télécommunication
Parcours Audio, Vidéo, Image
Responsable : Véronique GAILDRAT
2007–2008

Segmentation en plans par estimations simultanées des homographies inter-images

par Antoine LETOUZEY

Directeur de recherche : Patrice DALLE
Responsables de stage : Pierre GURDJOS et Alain CROUZIL

Mots-clés : homographie, analyse généralisée en composantes principales, frontière, vision par ordinateur.

Keywords : homography, generalized principal component analysis, computer vision.

Résumé

Ce travail s'inscrit dans le cadre de la vision par ordinateur et plus précisément de la détection de surfaces planes dans une scène vue à travers un couple stéréoscopique d'images. Nous nous sommes intéressés particulièrement aux méthodes permettant la détection simultanée de l'ensemble des plans. Nous avons tout d'abord étudié une technique existante, basée sur une méthode mathématique appelée "analyse généralisée en composantes principales". Ensuite nous avons élaboré une approche nouvelle au problème, particulièrement adaptée aux scènes urbaines, et utilisant le calcul des positions dans les images des frontières entre les différents plans de la scène.

Remerciements

Je remercie Monsieur Patrice Dalle, Professeur à l'Université Paul Sabatier, de m'avoir accueilli dans son équipe.

Je remercie sincèrement Messieurs Pierre Gurdjos, Ingénieur d'études à l'INPT et Alain Crouzil, Maître de conférence à l'Université Paul Sabatier de m'avoir encadré tout au long de ce stage et d'avoir toujours été là pour répondre à mes questions.

Je remercie Benoît Bocquillon, Doctorant, pour son aide précieuse malgré sa surcharge de travail.

Je remercie Guillaume Gales, Doctorant, pour son soutien et ses aides au transport.

Je remercie également Jean-Denis Durou, Maître de conférence à l'Université Paul Sabatier pour son soutien tout au long de ce stage.

Pour finir, je remercie l'ensemble des membres de l'équipe TCI qui m'ont enseigné, durant mes deux années passées à l'Université Paul Sabatier, tout ce que je sais en vision par ordinateur ainsi qu'en analyse d'images.

Table des matières

Introduction	1
1 Vision par ordinateur	3
1.1 Modélisation géométrique de la caméra	4
1.2 Calibrage de la caméra	4
1.3 Modèle géométrique du capteur stéréoscopique binoculaire	5
1.4 Géométrie épipolaire	5
1.5 Détection et mise en correspondance de primitives	7
1.6 Homographies	7
1.6.1 Estimation d'une homographie	8
1.6.2 Compatibilité avec la géométrie épipolaire	9
2 Méthodes de segmentation en plans	11
2.1 Analyse généralisée en composantes principales	11
2.2 Stratégie de type RANSAC	13
2.3 Méthodes semi-automatiques	14
2.4 Flux optique	16
2.5 Zones uniformes	17
2.6 Couples points–droites	20
2.7 Correspondances de droites dans une séquence d'images	20
2.8 Conclusion	22
3 Analyse Généralisée en composantes principales	25
3.1 Présentation et espérances	25
3.1.1 Principe de l'analyse généralisée en composantes principales	25
3.1.2 Exemple	27
3.2 Mise en œuvre	28
3.2.1 Adaptation aux droites	29
3.2.2 Calcul du nombre de plans	31
3.2.3 Segmentation des données	32
3.3 Expérimentations	32
3.3.1 Données de synthèses	32
3.3.1.1 Droites	32
3.3.1.2 Points	35
3.3.2 Données réelles	39
3.3.3 Nombre de plans	39
3.4 Conclusion	41

4	Une nouvelle approche : la segmentation en utilisant les frontières	43
4.1	Introduction	43
4.2	Présentation de la méthode	43
4.2.1	Projection de l'intersection de deux plans de la scène dans les images	43
4.2.2	Segmentation des données	44
4.3	Présentation de l'algorithme	44
4.3.1	Calcul de la matrice fondamentale	45
4.3.2	Recherche des homographies	45
4.3.3	Segmentation et affinement des résultats	46
4.4	Évaluations	48
4.4.1	Tests	48
4.4.1.1	Images de synthèse	48
4.4.1.2	Images réelles	48
4.5	Conclusion et perspectives	48
	Conclusion	53
	Bibliographie	55

Table des figures

1.1	Modèle de caméra sténopé.	5
1.2	Modèle de capteur stéréoscopique binoculaire.	6
1.3	Illustration de la géométrie épipolaire.	6
1.4	Homographie induite par un plan π	8
1.5	Compatibilité homographie – matrice fondamentale.	10
2.1	Représentation de la scène de synthèse contenant deux plans.	12
2.2	Représentation de la scène de synthèse contenant trois plans.	13
2.3	Élimination des “faux” plans.	14
2.4	Reconstruction d’un bâtiment.	15
2.5	Segmentation semi-automatique.	16
2.6	Modèle du robot et de son environnement.	17
2.7	Segmentation en plans en utilisant le flux optique.	18
2.8	Détection de plans en utilisant des zones de couleur uniforme.	19
2.9	Extractions des calques et reconstruction d’une image de la séquence vidéo.	19
2.10	Détection de plans en utilisant des points et des droites.	21
2.11	Critères de fusion des plans.	23
2.12	Critère de création de nouveaux segments.	23
3.1	Figure d’exemple.	27
3.2	Images de synthèse : droites.	33
3.3	Droites parfaites.	34
3.4	Bruit sur $d_3 : 0,001$	35
3.5	Bruit sur $d_3 : 0,01$	36
3.6	Points parfaits.	36
3.7	Bruitage des points.	37
3.8	Fiabilité de la segmentation.	38
3.9	Images réelles.	39
3.10	Résultat de la classification des droites sur un couple d’images réelles.	40
3.11	Etude de la fiabilité du calcul du nombres de plans.	41
4.1	Exemple de projection de l’intersection entre deux plans.	44
4.2	Vérification par la frontière.	47
4.3	Résultat de la segmentation initiale.	49
4.4	Performances sur les images de synthèse.	50
4.5	Segmentation erronée.	50
4.6	Résultat final.	51

Liste des tableaux

1.1	Conventions de notation.	3
2.1	Récapitulatif des méthodes de segmentation en plans par ordre de présentation dans ce document.	24
3.1	Correspondance entre nombre de plans et nombre minimum de données.	31

Introduction

La vision par ordinateur est une discipline qui a pour but de concevoir des outils permettant d'extraire de manière automatique des informations sur la géométrie 3D d'une scène vue au travers d'une ou de plusieurs images. La stéréovision binoculaire est une technique de vision par ordinateur utilisant deux images de la même scène prise par des capteurs placés à des positions différentes. Elle est utilisée de très nombreuses façons pour retrouver le relief d'une scène. Après une phase de mise en correspondance d'éléments (pixels, régions, ...) entre les deux images, il est possible de retrouver certaines propriétés géométriques de la scène. La détection des plans d'une scène à partir de couples d'images est un problème qui a été abondamment traité. Plusieurs méthodes ont été proposées chacune essayant d'utiliser une approche différentes (RANSAC, flux optique, pré-segmentation manuelle, ...). Chacune des approches a ses avantages et ses inconvénients suivant le type de scène.

Ce stage s'intègre dans le projet TSIGANE¹ dont le but est, d'une part, de permettre la localisation par vision embarquée au sein d'un système d'information géographique (SIG 3D) et, d'autre part, de compléter le SIG à partir des images acquises. La détection des plans permettrait de se repérer dans le monde à partir des informations du SIG en détectant les murs des bâtiments par exemple. Un premier positionnement serait donné par un GPS puis la mise en correspondance du modèle avec la reconstruction de la scène vue par les capteurs pourrait offrir plus de précision tout en incorporant des textures au SIG.

Ce rapport de stage présente tout d'abord un état de l'art sur les méthodes de détection de plans puis nous verrons ensuite deux approches pour l'estimation simultanée des homographies inter-images. La première méthode est basée sur l'utilisation de l'analyse généralisée en composantes principales tandis que la seconde fait l'hypothèse de scènes urbaines et se sert des frontières entre les plans pour segmenter les primitives mises en correspondance entre les images. Le travail effectué consiste en une étude approfondie de la première méthode ainsi que son adaptation à un autre type de données, les correspondances de droites ; mais il concerne aussi la mise au point de la seconde méthode qui aborde le problème d'une façon nouvelle. Nous détaillerons le fonctionnement de chaque approche puis présenterons les résultats obtenus ainsi que leur évaluation.

1. <http://www.irit.fr/wiki/doku.php?id=tsiganes>

Chapitre 1

Vision par ordinateur

Avant toutes choses et afin de faciliter la lecture de ce document, nous présentons dans le tableau suivant les conventions de notations utilisées tout au long de ce rapport.

a, α	: scalaires
\mathbf{p}, \mathbf{P}	: points
$\mathbf{v}, \overrightarrow{AB}$: vecteurs colonnes, bipoints
\mathbf{M}	: matrice
\mathbf{M}_i	: i ème ligne de la matrice \mathbf{M}
t	: opérateur de transposition d'un vecteur ou d'une matrice: \mathbf{M}^t
-1	: opérateur d'inversion d'une matrice: \mathbf{M}^{-1}
$-t$: $\mathbf{M}^{-t} = (\mathbf{M}^t)^{-1}$
\cdot	: produit scalaire de deux vecteurs: $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^t \mathbf{y}$
\sim	: égalité à un facteur multiplicatif près: $\mathbf{x} \sim \mathbf{y}$
$*$: transposée conjuguée: \mathbf{M}^*
\otimes	: produit tensoriel, ou produit de Kronecker: $\mathbf{v} \otimes \mathbf{u}$

TAB. 1.1 – Conventions de notation.

La vision par ordinateur a pour but d'extraire automatiquement des informations sur une scène à partir d'images. Un des domaines les plus importants est la recherche du relief de la scène: la stéréovision. Suivant le nombre d'images utilisées, les techniques pour retrouver le relief varient. Dans le cas où l'on ne dispose que d'une image, on parle alors de stéréovision monoculaire et on pourra utiliser un des critères suivants:

- le flou de mise au point
- les ombres propres (*shape from shading*)
- les contours (*shape from contour*)
- la texture
- ...

Même s'il existe de multiples techniques de restitution du relief à partir d'une seule image, la plupart des travaux de stéréovision se font à partir de plusieurs images; il s'agit alors de stéréovision multi-oculaire. Elle comprend les techniques suivantes:

- la photostéréométrie, qui consiste à changer la position de la source lumineuse pour chaque image;
- la stéréovision multi-vues où le capteur se déplace dans une scène fixe;

- les capteurs actifs qui utilisent un laser ou un éclairage structuré dont on étudie la déformation dans les images ;
- la stéréovision binoculaire (ou trinoculaire), qui nécessite deux (ou trois) images de la scène prises au même moment depuis des positions différentes.

Dans ce rapport, nous nous plaçons dans le cadre de la stéréovision binoculaire dans le but de détecter les plans présents dans la scène. Avant d’en arriver là, il faut détecter et mettre en correspondance des points d’intérêt dans les deux images avant d’estimer les homographies induites par les plans. Nous allons voir succinctement comment se déroulent ces étapes.

1.1 Modélisation géométrique de la caméra

Le modèle utilisé pour représenter la caméra est appelé le modèle sténopé (ou “trou d’aiguille”). Il se caractérise par un centre de projection, le centre optique \mathbf{F} , et un plan de projection, appelé le plan image. Ce modèle, illustré par la figure 1.1 permet de projeter un point \mathbf{P} du repère de la scène en un point \mathbf{p} dans le plan image par l’équation suivante :

$$\lambda(u \ v \ 1)^t = \mathbf{M}(X \ Y \ Z \ 1)^t \quad (1.1)$$

La matrice \mathbf{M} , de taille (3×4) , est appelée matrice de projection perspective ; elle est définie à un coefficient multiplicatif près. Comme on peut le voir sur la figure 1.1, la projection d’un point de la scène en un point dans l’image se décompose en deux étapes. La première est le passage du repère 3D de la scène vers le repère 3D de la caméra, qui consiste en une translation et une rotation. Ce changement de repère est décrit par une matrice (4×4) \mathbf{A} appelée matrice des paramètres extrinsèques. La seconde est une projection perspective qui permet d’obtenir les coordonnées 2D dans le repère image de la projection d’un point 3D dans le repère de la caméra ; elle est décrite par une matrice (3×3) \mathbf{C} appelée matrice des paramètres intrinsèques. On a la relation suivante :

$$\mathbf{M} = [\mathbf{C} \ \mathbf{0}_{3 \times 1}] \mathbf{A} \quad (1.2)$$

où

$$\mathbf{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \mathbf{C} = \begin{pmatrix} -k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

La matrice \mathbf{A} contient six paramètres indépendants, trois pour la rotation et trois pour la translation (t_x , t_y et t_z). La matrice \mathbf{C} en contient quatre : les coordonnées du point principal, (u_0, v_0) , la distance focale exprimée en hauteur de pixels et la distance focale exprimée en largeur de pixels. On peut éventuellement compléter ces paramètres intrinsèques en ajoutant un paramètre dépendant de l’angle formé par les axes u et v du repère image. Ce paramètre est appelé le “*skew factor*” (ou “coefficient de cisaillement horizontal”). Néanmoins, nous considérerons ici que ce dernier paramètre vaut 0.

1.2 Calibrage de la caméra

L’étape de calibrage de la caméra consiste à calculer la matrice \mathbf{M} , généralement à partir de correspondances entre des points de la scène et des points de l’image. Le calibrage à partir de points nécessite de connaître la position 3D dans le repère de la scène des points utilisés ; on utilise pour cela des mires de calibrage [27]. Si cette étape est nécessaire à la reconstruction euclidienne de la scène, elle n’est pas indispensable pour obtenir des informations sur sa structure géométrique (détecter des plans, ...).

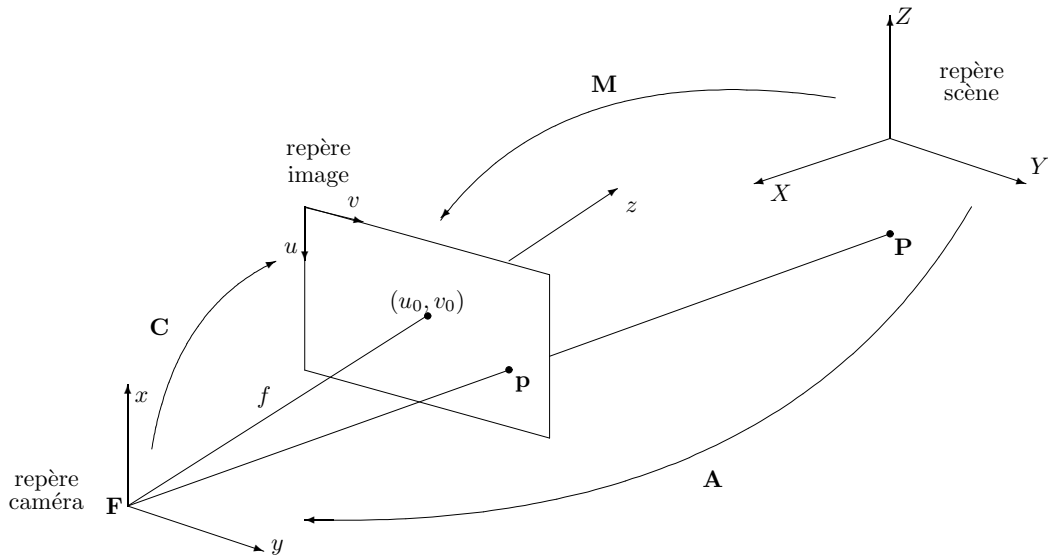


FIG. 1.1 – Modèle de caméra sténopé.

1.3 Modèle géométrique du capteur stéréoscopique binoculaire

Un capteur stéréoscopique binoculaire est constitué de deux caméras. Une condition nécessaire pour que le problème de reconstruction 3D (projection euclidienne) soit “bien posé” consiste en les deux contraintes suivantes sur la position relative des caméras. La première est que la partie de la scène qui nous intéresse doit être visible dans les deux images et la seconde est que la position de la seconde caméra ne soit pas obtenue par une rotation de la première caméra autour d’un axe passant par son centre optique (“rotation pure”). Le passage du repère de la caméra de gauche à celui de la caméra de droite est défini par une rotation et une translation et peut être écrit sous la forme d’une matrice (4×4) $\mathbf{A}_{g \rightarrow d}$ telle que :

$$\mathbf{A}_{g \rightarrow d} = \mathbf{A}_d \mathbf{A}_g^{-1} \quad (1.3)$$

Il est possible de reconstruire les points de la scène dont on connaît la projection dans les images gauche et droite si on dispose des paramètres intrinsèques des deux caméras et de la matrice $\mathbf{A}_{g \rightarrow d}$. La figure 1.2 montre le modèle d’un capteur stéréoscopique binoculaire.

1.4 Géométrie épipolaire

Les images fournies par le capteur binoculaire respectent certaines propriétés géométriques. Parmi celles-ci, la contrainte épipolaire, illustrée dans la figure 1.3, est très utilisée. Elle permet de limiter l’espace de recherche du correspondant d’un pixel d’une image dans l’autre image à une droite, appelée “droite épipolaire”. Pour obtenir l’équation de la droite épipolaire associée à un pixel on utilise la matrice fondamentale, notée \mathbf{F} . Il s’agit d’une matrice (3×3) de rang 2 et qui est définie par l’équation 1.4 dans laquelle \mathbf{m}_g et \mathbf{m}_d sont les coordonnées homogènes des projections dans les images de gauche et de droite du même point de la scène :

$$\mathbf{m}_d^t \mathbf{F} \mathbf{m}_g = 0 \quad (1.4)$$

Les équations des droites épipolaires gauche et droite sont données respectivement par $\mathbf{F}^t \mathbf{m}_d$ et $\mathbf{F} \mathbf{m}_g$. La matrice fondamentale peut être obtenue à partir de points mis en correspondance entre les deux images ; il en faut au moins 7.

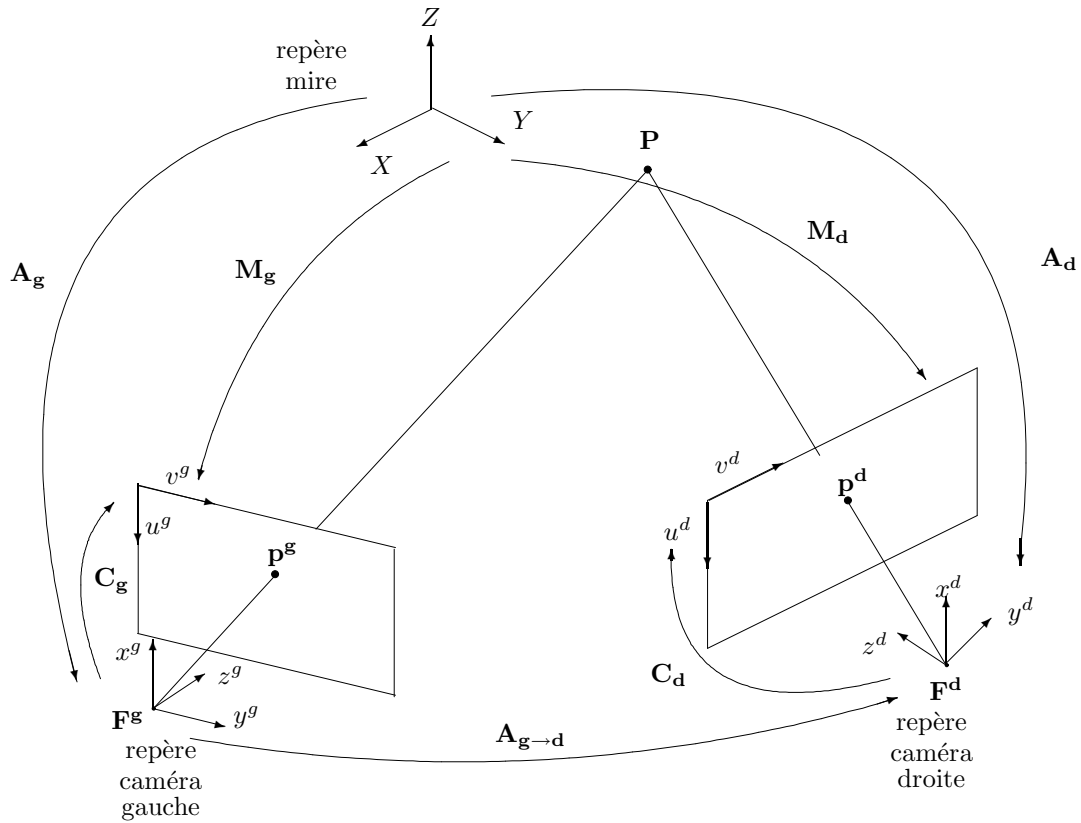


FIG. 1.2 – *Modèle de capteur stéréoscopique binoculaire.*

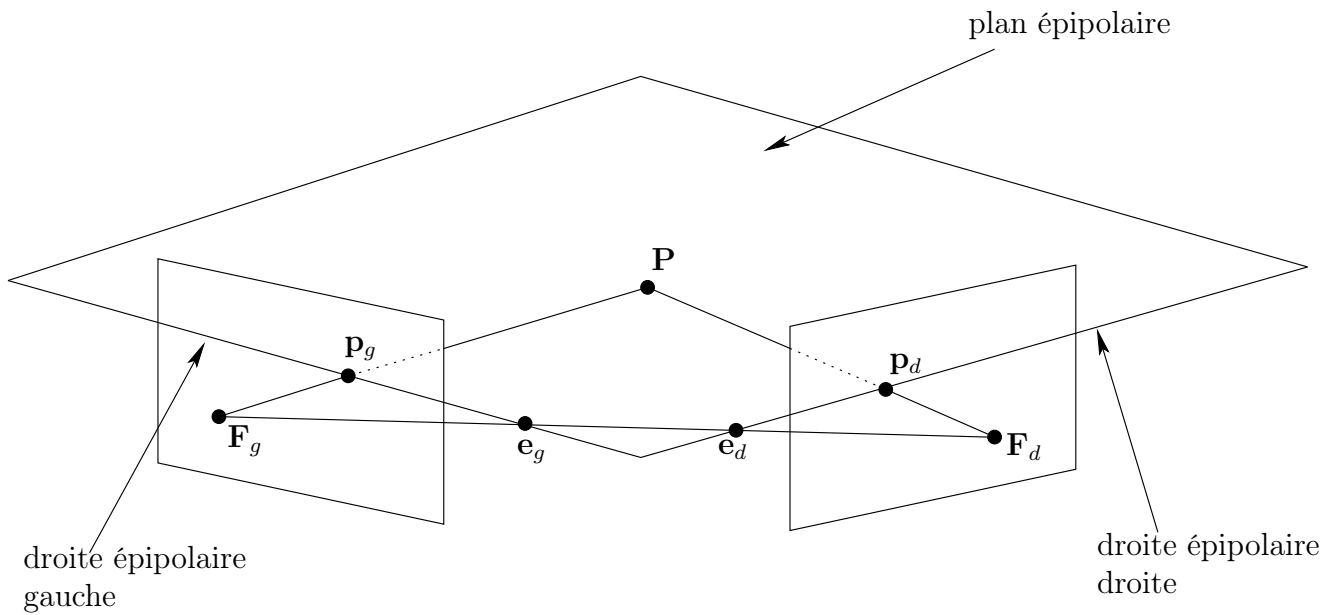


FIG. 1.3 – *Illustration de la géométrie épipolaire.*

1.5 Détection et mise en correspondance de primitives

La première étape pour obtenir des informations sur la scène est la mise en correspondance de primitives entre les images. Il peut s’agir de points, de droites, de zones de couleurs, ... La plupart des méthodes existantes se déroulent en deux phases. La première consiste à détecter les primitives dans une image puis la deuxième a pour but de chercher leurs correspondants dans les autres images. Parmi les détecteurs de points d’intérêt, on peut citer SIFT [22] ou Harris [15]. La détection de droites (ou d’autres primitives géométriques) se fait plutôt en utilisant une transformée de Hough [10].

Pour l’étape de mise en correspondance, il existe des critères permettant d’évaluer si deux éléments correspondent à la même entité de la scène. Ces critères diffèrent suivant le type de primitives. Pour la mise en correspondance de points, les contraintes suivantes sont parmi les plus utilisées.

Contrainte épipolaire Comme nous l’avons signalé dans la section 1.4, cette contrainte permet de restreindre la zone de recherche du correspondant d’un point à une droite dans la seconde image.

Contrainte de similarité Elle consiste à vérifier que le voisinage du point ressemble à celui de son correspondant potentiel.

Contrainte d’unicité Chaque point a au plus un correspondant.

Contrainte d’ordre Les points sont dans le même ordre le long des droites épipolaires dans les deux images.

Il est possible de confirmer les appariements trouvés dans le sens image gauche \rightarrow image droite en recommençant la procédure dans le sens inverse (image droite \rightarrow image gauche) et en ne conservant que les couples de primitives dont les appariements sont cohérents.

1.6 Homographies

Une homographie 2D est une transformation projective d’un plan projectif sur lui-même, qui a la propriété d’être linéaire en coordonnées homogènes. En particulier, étant donné un ensemble de points 3D coplanaires, la transformation des coordonnées homogènes de leurs projections dans les images de gauche et de droite est une homographie, dite induite par le plan de support de ces points. Deux cas se présentent :

- Les centres optiques des deux caméras sont confondus. On peut considérer que les antécédents de tous les points homologues sont situés sur un plan “à l’infini” ; ainsi il existe une homographie, induite par le plan de l’infini, liant *tous* les points homologues des deux images.
- Les antécédents d’un certain nombre de points homologues sont situés sur un plan “fini” (c.-à-d., affine), ne contenant pas les centres optiques. Il existe une homographie, induite par ce plan, liant ces points homologues des deux images.

C’est ce deuxième cas qui nous intéresse ici puisqu’il donne des informations utiles à la détection des plans de la scène à partir de correspondances de points.

Les coordonnées “tangentes” d’un plan 3D sont données par tout vecteur non nul proportionnel à $\boldsymbol{\pi} = (n_1, n_2, n_3, -d)$. Ainsi un point en coordonnées homogène $\mathbf{P} = (X, Y, Z, 1)^t$ situé sur ce plan vérifie :

$$\boldsymbol{\pi}^t \mathbf{P} = 0 \iff \frac{1}{d}(n_1, n_2, n_3)(X, Y, Z)^t = 1 \quad (1.5)$$

Vu par deux images, ce plan induit une homographie \mathcal{H} entre les deux vues. Cette relation est illustrée par la figure 1.4 [16, chapitre 13]. Un point de l'image de gauche est lié à son correspondant

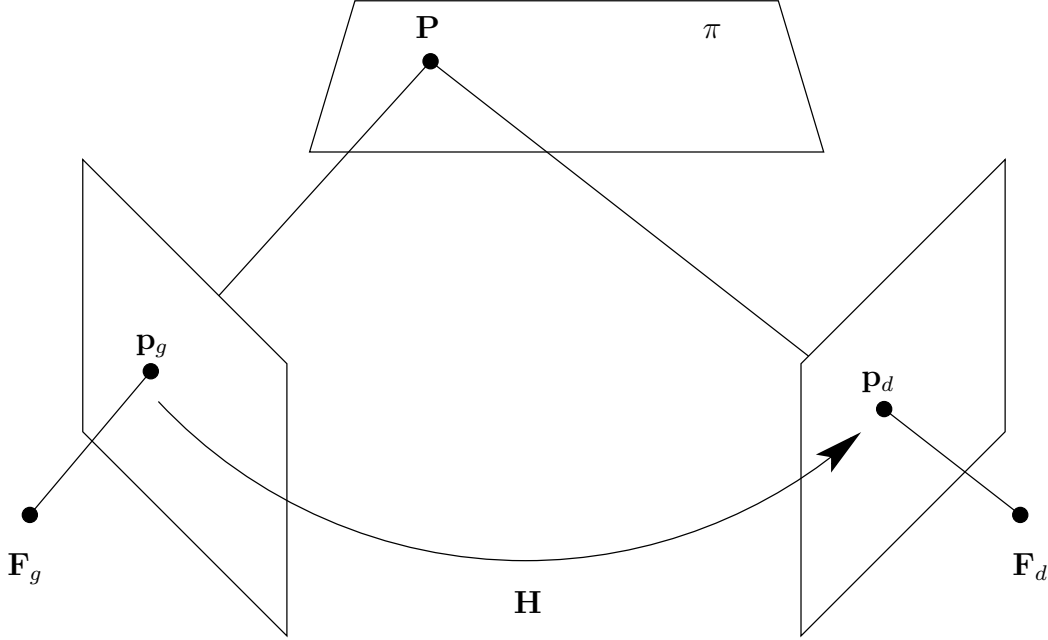


FIG. 1.4 – Homographie induite par un plan π .

par l'équation 1.6 où \mathbf{H} est une matrice (3×3) définie à un facteur multiplicatif près représentant l'homographie \mathcal{H} (par la suite nous utiliserons \mathbf{H} ou \mathcal{H} pour désigner une homographie).

$$\mathbf{m}_d \sim \mathbf{H}\mathbf{m}_g \quad (1.6)$$

Supposons que le repère de la scène soit confondu avec celui de la caméra gauche ; on a :

$$\mathbf{A}_g = [\mathbf{I}_{3 \times 3} | \mathbf{0}_3] \quad \text{et} \quad \mathbf{A}_d = [\mathbf{R} | \mathbf{t}].$$

L'homographie induite par le plan π entre l'image de gauche et celle de droite est définie par :

$$\mathbf{H} = \mathbf{C}_d \left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^t}{d} \right) \mathbf{C}_g^{-1} \quad (1.7)$$

1.6.1 Estimation d'une homographie

Estimation à partir de correspondances de points Une homographie \mathcal{H} est une application de \mathbb{R}^2 dans \mathbb{R}^2 définie par [8]:

$$\mathcal{H} : (u, v) \mapsto (u', v') \quad \begin{cases} u' = \frac{au + bv + c}{gu + hv + i} \\ v' = \frac{du + ev + f}{gu + hv + i} \end{cases} \quad (1.8)$$

$$\Leftrightarrow \lambda \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad \forall \lambda \neq 0 \quad (1.9)$$

$$\Leftrightarrow \mathbf{m}' \sim \mathbf{H}\mathbf{m} \quad (1.10)$$

avec $\mathbf{H} \in \mathbb{R}^{3 \times 3}$ et $gu + hv + i \neq 0$.

Un couple de points mis en correspondance fournit deux équations linéairement indépendantes en les inconnues. Or comme il n'y a que 8 paramètres indépendants sur les 9 éléments de la matrice \mathbf{H} , il suffit d'avoir 4 couples de points non alignés pour obtenir une solution exacte. Dans la pratique, où les données sont souvent bruitées, nous utiliserons plus de couples de points, le système est alors surdéterminé et peut être résolu par une méthode d'estimation de paramètres, éventuellement par une méthode robuste.

Estimation à partir de correspondances de droites L'homographie est une transformation qui s'applique aussi aux projections des droites dans les images. Soient \mathbf{l}_g et \mathbf{l}_d deux vecteurs appartenant à \mathbb{R}^3 définissant les paramètres des projections d'une droite L de la scène dans les deux images. Si cette droite repose sur un plan induisant dans les images une homographie \mathbf{H} , alors la relation entre les deux droites dans les images est définie par [16, chapitre 4 - page 92]:

$$\mathbf{l}_d \sim \mathbf{H}^{-t} \mathbf{l}_g \quad (1.11)$$

où \mathbf{l}_g et \mathbf{l}_d sont de la forme $(l_1 \ l_2 \ l_3)^t$ avec $l_1^2 + l_2^2 = 1$ pour les droites affines. Cette équation, homologue à l'équation 1.10, permet d'estimer la matrice \mathbf{H} à partir de correspondances de droites de la même manière qu'avec des correspondances de points.

1.6.2 Compatibilité avec la géométrie épipolaire

Soit \mathbf{H} une matrice représentant l'homographie induite entre les deux images par un plan π de la scène ne passant pas par les centres optiques des caméras et \mathbf{P} un point 3D. Le point \mathbf{P} se projette en \mathbf{p}_g sur l'image gauche et en \mathbf{p}_d sur l'image de droite. La droite passant par le centre optique de la caméra gauche et par le point \mathbf{P} coupe le plan π en un point 3D \mathbf{P}_π qui se projette lui aussi sur \mathbf{p}_g dans l'image de gauche et en un point $\mathbf{p}_{\pi d} = \mathbf{H}\mathbf{p}_g$ dans l'image de droite. Ceci est illustré par la figure 1.5. Nous avons vu dans le paragraphe 1.4 que chaque couple de points mis en correspondance vérifiait la contrainte épipolaire. Nous avons donc les deux équations suivantes:

$$\mathbf{p}_{\pi d}^t \mathbf{F} \mathbf{p}_g = 0 \quad (1.12)$$

$$\mathbf{p}_{\pi d} = \mathbf{H} \mathbf{p}_g \quad (1.13)$$

Des équations 1.12 et 1.13 nous pouvons déduire:

$$(\mathbf{H} \mathbf{p}_g)^t \mathbf{F} \mathbf{p}_g = \mathbf{p}_g^t \mathbf{H}^t \mathbf{F} \mathbf{p}_g = 0 \quad \forall \mathbf{p}_g. \quad (1.14)$$

Cette équation est vraie pour chaque point de l'image de gauche qu'il soit ou non l'image d'un point de la scène se trouvant sur le plan π . Si cette contrainte n'est pas vérifiée, cela signifie que l'homographie \mathbf{H} n'est pas induite par un plan de la scène. Une autre relation entre \mathbf{H} et \mathbf{F} permettant de vérifier que \mathbf{H} est bien induite par un plan est la suivante [16, chapitre 13]:

$$\mathbf{H}^t \mathbf{F} + \mathbf{F}^t \mathbf{H} = 0 \quad (1.15)$$

Ce qui revient à dire que $\mathbf{H}^t \mathbf{F}$ est antisymétrique.

Démonstration

Posons:

$$\mathbf{H}^t \mathbf{F} = \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

L'équation 1.14 devient $\mathbf{x}^t \mathbf{A} \mathbf{x} = 0 \quad \forall \mathbf{x}$ de la forme $(x, y, 1)^t$ avec x et $y \in \mathbb{R}$. Donc :

$$\begin{aligned}
 0 &= \mathbf{x}^t \mathbf{A} \mathbf{x} \\
 &= a_{11}x^2 + a_{22}y^2 + (a_{21} + a_{12})xy + (a_{31} + a_{13})x + (a_{32} + a_{23})y + a_{33} \\
 \Leftrightarrow &\begin{cases} a_{11} = a_{22} = a_{33} = 0 \\ a_{21} + a_{12} = 0 \\ a_{31} + a_{13} = 0 \\ a_{32} + a_{23} = 0 \end{cases} \\
 \Leftrightarrow &\mathbf{A} \text{ est antisymétrique} \quad \square
 \end{aligned}$$

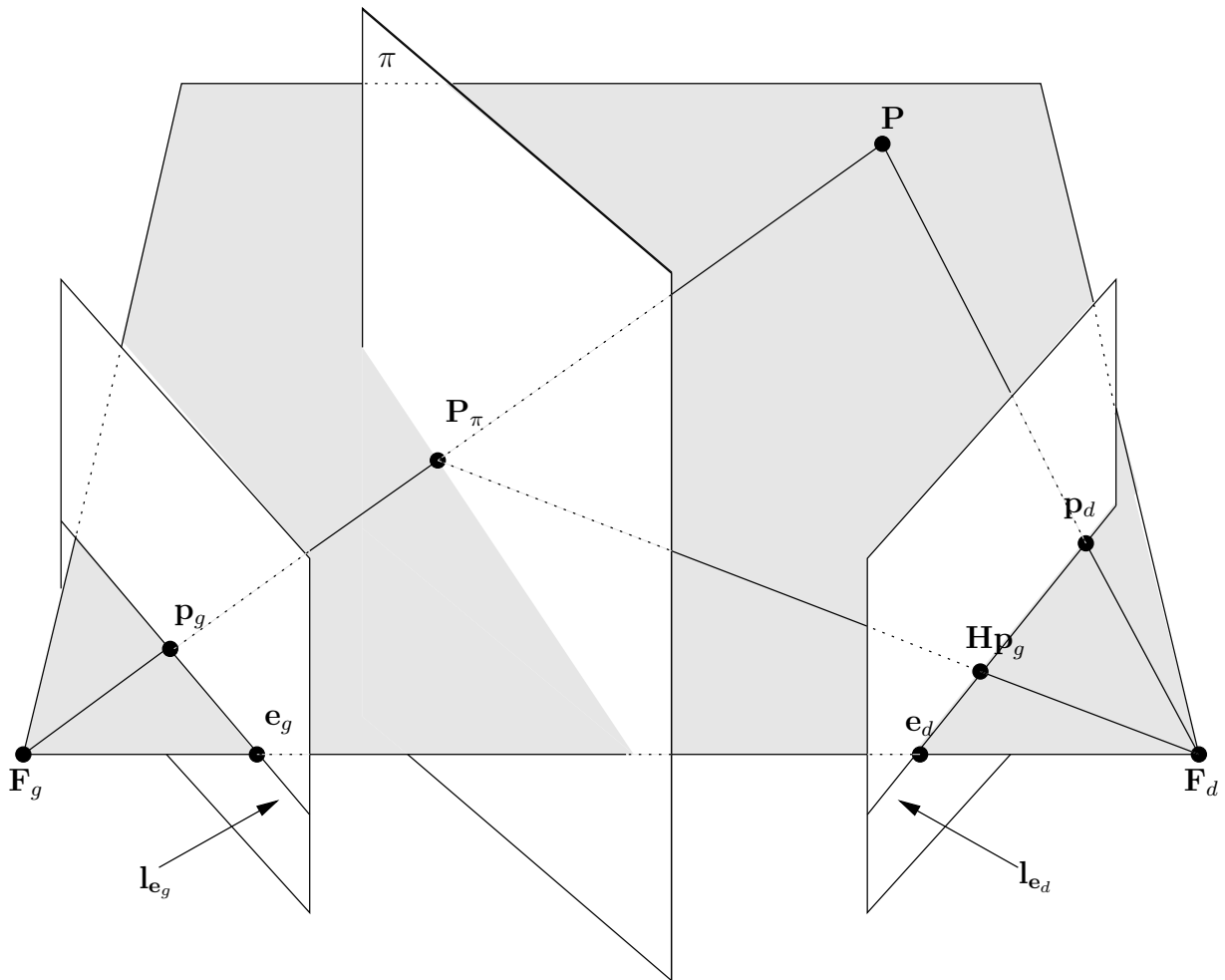


FIG. 1.5 – *Compatibilité homographie – matrice fondamentale.* l_{e_g} et l_{e_d} sont respectivement les droites épipolaires associées au point P dans les images gauche et droite.

Chapitre 2

Méthodes de segmentation en plans

Détecter les plans dans une scène 3D à partir de plusieurs images permet d'obtenir des informations importantes sur sa structure. Ces informations peuvent ensuite être utilisées, par exemple, pour reconstruire un modèle de la scène en calculant les paramètres des équations des plans 3D. Une autre application est l'estimation des homographies inter-images induites par les plans qui, comme nous l'avons vu dans le paragraphe 1.6, permet d'ajouter une contrainte géométrique forte entre un pixel et son correspondant dans un couple d'images. Généralement la détection de plans se déroule en deux phases. La première consiste à extraire puis à mettre en correspondance entre les images des primitives (points, droites, ...); la seconde concerne la segmentation en classes de ces primitives et le calcul des paramètres des plans. Une classe regroupe les primitives qui sont les projections d'entités appartenant au même plan de la scène. Les deux étapes de la seconde phase sont intimement liées dans la mesure où elles se heurtent généralement au problème de type “*chicken and egg*” suivant :

- si les paramètres des plans étaient connus, alors il serait facile de segmenter les données ;
- si les données étaient correctement segmentées, alors il serait facile d'obtenir les paramètres des plans.

C'est un obstacle d'autant plus grand que le nombre de classes est généralement inconnu. La plupart des techniques tentent de résoudre ce problème de manière itérative. Certaines estiment les classes une à une, d'autres proposent un modèle global puis l'adaptent aux données.

L'extraction de plans dans des scènes 3D est un problème qui a été abondamment traité dans la littérature. Nous allons présenter ici un état de l'art des différentes méthodes proposées. La présentation n'a pas pour but d'être exhaustive mais plutôt d'être la plus variée possible. Chacune des techniques présentées diffère par la forme des données utilisées en entrée ou par les hypothèses plus ou moins fortes faites sur la structure de la scène.

La grande majorité des méthodes présentées se basent sur l'utilisation de couples d'images, néanmoins certaines, notamment celles basées sur le flux optique, utilisent des séquences d'images. Dans une séquence d'images, on peut considérer que deux images consécutives forment un couple stéréoscopique dans la mesure où la caméra effectue un léger déplacement entre deux vues.

2.1 Analyse généralisée en composantes principales

Dans [30], [29] ainsi que [33], René Vidal et Yi Ma présentent une technique, nommée *Generalized Principal Component Analysis*, de segmentation de données en un nombre N de sous-espaces de dimensions M_j , $j \leq j \leq N$, où N et les M_j sont inconnus, basée sur l'utilisation de polynômes.

Chaque sous-espace (classe) est représenté par un polynôme homogène dont le degré est égal au nombre de classes. Leurs travaux se basent sur le principe suivant énoncé dans [30].

Une union de n sous-espaces de \mathbb{R}^D peut être représentée par un ensemble de polynômes homogènes de degré n à D variables. Avec suffisamment de données ces polynômes peuvent être estimés de manière linéaire. Une base des compléments de chacun de ces sous-espaces peut être calculée à partir des dérivées de ces polynômes en un point de chacun des sous-espaces. Ces points peuvent être sélectionnés récursivement par division polynomiale. Le problème de segmentation en sous-espaces peut donc être résolu par des divisions, des dérivations et des ajustements de polynômes homogènes.

Cette méthode n'est pas spécifique à la détection de plans ; les auteurs s'en servent d'ailleurs dans plusieurs contextes allant de la segmentation en plans d'une séquence vidéo d'un journal TV à la reconnaissance de visages en passant par la détection de mouvements. En 2005, dans [33], l'analyse généralisée en composantes principales est utilisée pour détecter les plans d'une scène à partir de couples de points mis en correspondance dans un couple stéréoscopique d'images. Il s'agit d'un cas particulier de leur méthode puisqu'ici, les dimensions des classes sont connues et toutes égales à deux. Les principes mathématiques de cette méthode seront détaillés plus longuement dans la section 3.4.

Deux étapes de résolution

- le calcul du nombre de plans de la scène ;
- la segmentation des points dans les différentes classes.

La méthode est expérimentée uniquement sur les scènes de synthèse présentées sur les figures 2.1 et 2.2 extraites de [30].

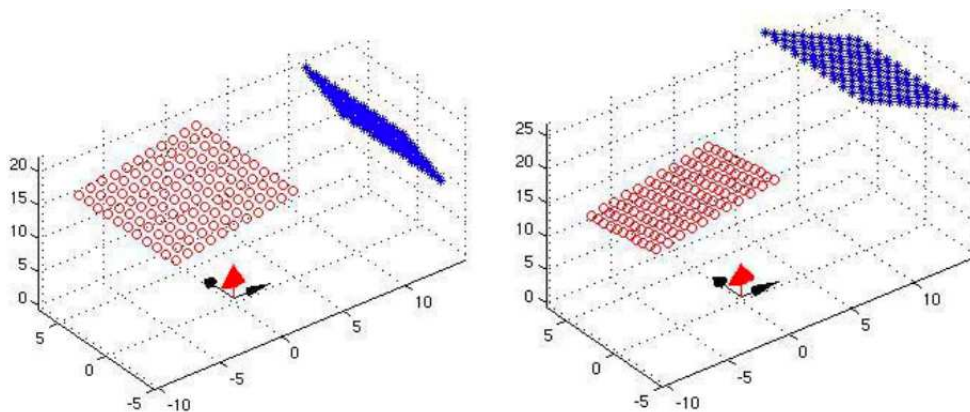


FIG. 2.1 – Représentation de la scène de synthèse contenant deux plans.

Avantages et inconvénients : L'avantage principal de cette méthode est qu'il n'est pas nécessaire de faire des hypothèses sur le nombre de plans ou sur la structure de la scène. Néanmoins, comme nous le verrons plus loin, elle demande beaucoup de correspondances de points. Elle est aussi assez peu résistante au bruit et est très dépendante du calcul du nombre de plans. Si celui-ci est faux, tous les calculs qui suivent, dont la segmentation des points, sont faux.

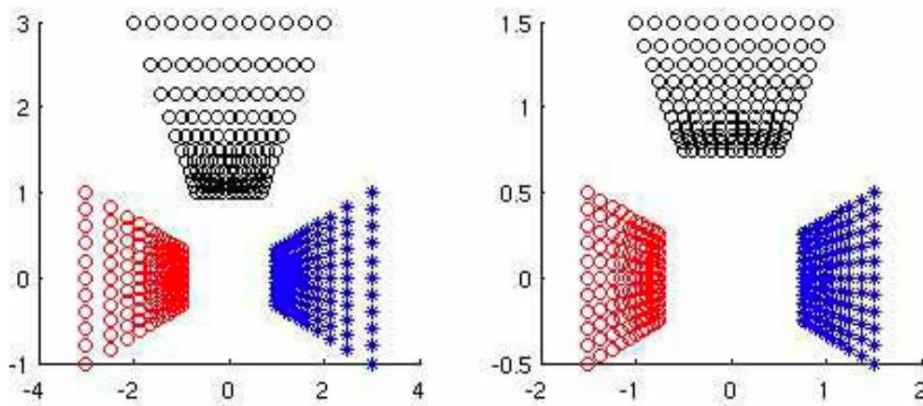


FIG. 2.2 – Représentation de la scène de synthèse contenant trois plans.

2.2 Stratégie de type RANSAC

En 1981 Fischler et Boles présentent une méthode d'estimation robuste de paramètres très populaire dans la communauté de vision par ordinateur : RANSAC (*RAN*d*om SA*m*ple Conc*en*sus*) [13]. Son principe est simple : il consiste à sélectionner le nombre minimum de données au hasard pour calculer les paramètres d'une classe, puis à identifier, parmi toutes les données, celles qui ont des chances raisonnables d'appartenir à cette classe. Si ce sous-ensemble de données est assez conséquent, les paramètres de la classe sont réévalués de manière à correspondre le mieux possible aux données choisies. Le processus est itératif et s'arrête lorsque l'on arrive à un état stable ou à une oscillation. À ce moment, les données appartenant à la classe calculée sont retirées et l'algorithme reprend au début pour trouver de nouvelles classes. L'algorithme s'interrompt lorsqu'il n'y a plus de données à classer ou lorsque le calcul de nouvelles classes ne permet plus d'obtenir de classes pouvant correspondre à suffisamment de données. Le principal avantage de cette méthode est qu'elle résiste très bien à la présence de données aberrantes (*outliers*) en supportant jusqu'à 50% de grosses erreurs. Il existe aujourd'hui de nombreuses variantes de cette méthode et [16, chap 4, page 118] propose un algorithme pour implémenter sa version de base.

C'est sur ce principe que beaucoup de méthodes de segmentation opèrent. Par exemple, en 2007 dans [4], Adrien Bartoli propose une méthode basée sur RANSAC pour segmenter des couples de points dans des images stéréoscopiques et reconstruire les plans 3D de la scène correspondant aux classes de points trouvées. Les modifications apportées concernent deux points de l'algorithme présenté plus haut.

L'auteur fait l'hypothèse que certains couples de points peuvent appartenir à plusieurs plans de la scène. Cette hypothèse est plausible dans la mesure où les frontières entre les plans dans des scènes réelles sont souvent des zones susceptibles de contenir des points d'intérêt (changements nets de couleur, de texture, ...). Donc, une fois qu'un plan est détecté, les points qui lui appartiennent ne sont pas retirés pour la suite de l'algorithme. Pour éviter de calculer plusieurs fois le même plan, l'auteur propose un critère de dissimilarité que doivent vérifier les nouveaux plans avec ceux déjà trouvés. Cette mesure se base sur la proportion de points qu'ont en commun deux plans et est donnée par :

$$\mathcal{D}(i,j) = \frac{2 * \#(\Pi_i \cap \Pi_j)}{\#(\Pi_i) + \#(\Pi_j)} \quad (2.1)$$

où Π_i est l'ensemble des points appartenant au $i^{\text{ème}}$ plan et $\#(E)$ désigne le nombre d'éléments de l'ensemble E . D'après l'auteur, deux plans i et j peuvent être considérés comme différents si $\mathcal{D}(i,j) < 0.5$.

Il se peut, comme illustré dans la figure 2.3, que certains des plans calculés ne correspondent pas à de vrais plans de la scène. Pour pallier ce problème, l'auteur propose d'appliquer à chaque plan la suite d'opérations suivante :

1. sélection des images dans lesquelles le plan apparaît en entier ;
2. calcul de l'enveloppe convexe formée par les points d'intérêt associés au plan ;
3. maillage (avec des triangles) du plan à partir des points d'intérêt qu'il contient ;
4. vérification de la consistance photométrique de chaque triangle et suppression des triangles non consistants.

La cohérence photométrique utilisée au point 4 est une contrainte inspirée de [20] qui permet d'écartier les triangles dont certains pixels ne satisfont pas la contrainte géométrique de l'homographie associée au plan contenant le triangle. Pour cela, l'algorithme vérifie que le correspondant de chaque pixel \mathbf{p} de chaque triangle se trouve bien dans un espace délimité dans les autres images par un disque de rayon r centré en $\mathbf{H}\mathbf{p}$ où \mathbf{H} est l'homographie induite par le plan entre les deux images considérées. En faisant varier r , il est possible d'obtenir une contrainte plus ou moins forte. Cette technique permet d'obtenir de manière précise le contour des facettes planes de la scène mais aussi de supprimer les faux plans.

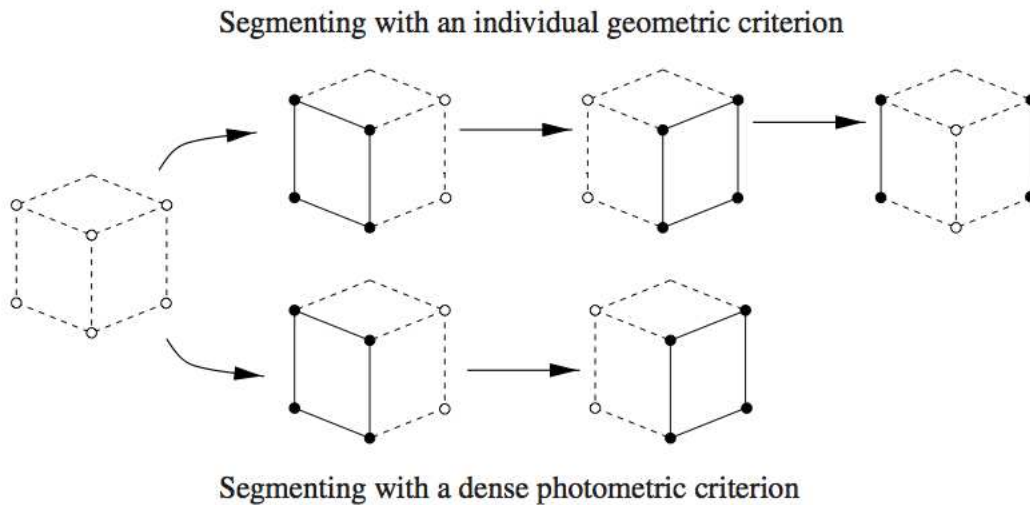


FIG. 2.3 – Le critère de consistance photométrique permet d'éliminer le plan qui ne correspond pas à un plan de la scène (figure extraite de [4]).

Les résultats obtenus sont assez bons et les expériences ont été menées sur des images de synthèse et des images réelles. La figure 2.4 montre un exemple de reconstruction en utilisant la méthode.

2.3 Méthodes semi-automatiques

Les méthodes semi-automatiques utilisent une initialisation manuelle afin de détecter les plans. Généralement ces méthodes sont utilisées pour obtenir des résultats de grande qualité. La méthode présentée dans [5] et [6] propose une segmentation semi-automatique en plans pour obtenir une

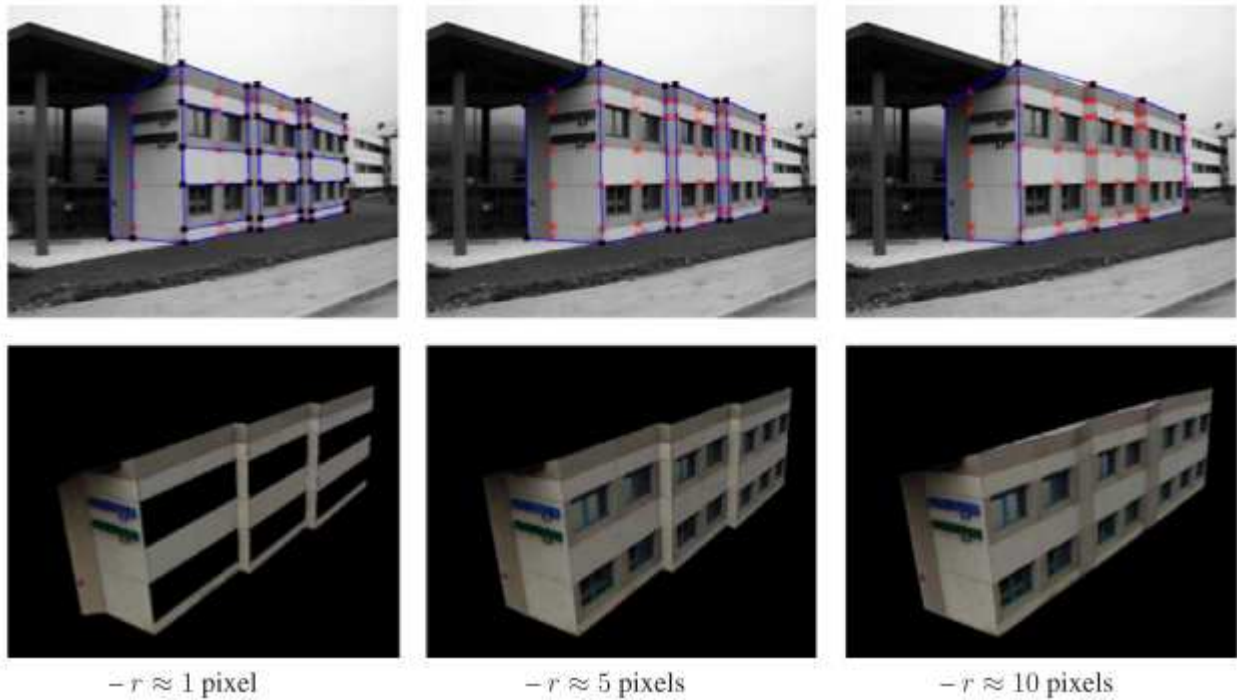


FIG. 2.4 – Reconstruction en 3D des plans d'un bâtiment. Suivant l'écart photométrique toléré la reconstruction du modèle 3D du bâtiment est plus ou moins précise. Avec une valeur faible, les fenêtres sont considérées comme n'appartenant pas à la façade, tandis qu'avec une valeur grande, la façade est assimilée à un unique plan alors qu'elle a plutôt une forme en escalier (figure extraite de [4]).

mise en correspondance dense de deux images de manière très précise en passant par une estimation robuste des homographies.

La méthode est utilisée avec des scènes composées de facettes planes à contours polygonaux. Elle se divise en deux étapes principales. Tout d'abord l'utilisateur doit donner au système une segmentation initiale proche de la réalité (quelques pixels d'erreur par segment du polygone). La seconde étape est automatique ; il s'agit d'une phase d'optimisation des contours. Elle est effectuée par un modèle de contours actifs adapté au problème de détection de polygones. Cette phase revient à maximiser la fonction d'énergie suivante :

$$E(P) = \frac{1}{N_P} \sum_{\mathbf{p} \in P} (\nabla \mathbf{I}(\mathbf{p}) \cdot \mathbf{n}_C)^2 \quad (2.2)$$

où $E(P)$ est l'énergie du polygone P , $\mathbf{p} \in P$ représente tous les pixels de P , N_P est le nombre de pixels du polygone, $\nabla \mathbf{I}(\mathbf{p})$ est le vecteur gradient de l'image I au point \mathbf{p} et \mathbf{n}_C est la normale au côté courant du polygone. La maximisation de cette énergie est faite en cherchant les positions idéales des sommets du polygone. Pour cela, une zone de recherche est définie autour des emplacements initiaux de chaque sommet. Suivant la taille de la zone, l'espace des polygones peut être très (voire trop) grand, ce qui interdit un parcours exhaustif des possibilités. La méthode présentée propose une segmentation des zones de recherche en sous-zones pour réduire les temps de calcul. Cette manière de faire donne le meilleur polygone au pixel près ; ce qui n'est pas toujours assez précis. Il

est possible d'obtenir une meilleure précision des positions des sommets en augmentant la résolution de l'image autour des cotés des polygones. La même méthode est appliquée de manière successive à tous les polygones de l'image. Les homographies des différents plans sont calculées avec l'estimateur robuste RANSAC à partir de correspondances de points d'intérêt sélectionnés avec le détecteur de Harris. Une fois les homographies de chaque plan calculées, l'appariement dense des points entre les deux images est possible car l'équation 1.6 permet de mettre en relation les points des deux images de manière bi-univoque. La figure 2.5 illustre la qualité de la segmentation manuelle nécessaire afin d'obtenir de bons résultats.

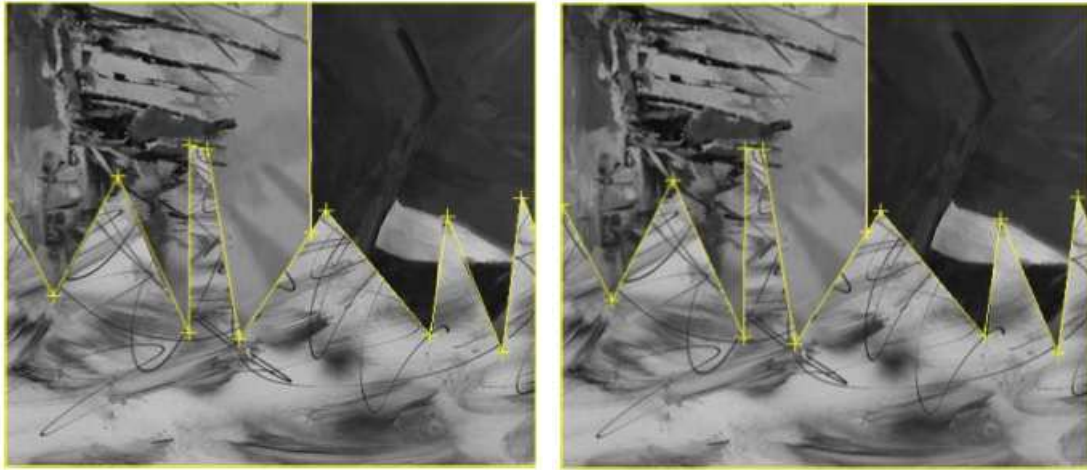


FIG. 2.5 – *Segmentation manuelle (à gauche) puis affinée automatiquement (à droite) pour le calcul précis des homographies (figure extraite de [5]).*

La méthode présentée a pour avantage de fournir de très bons résultats mais elle nécessite beaucoup d'hypothèses assez fortes sur la structure de la scène. Il faut en effet que la scène soit entièrement composée de facettes planes et que les contours des facettes soient des polygones. De plus, elle nécessite une pré-segmentation manuelle de qualité.

2.4 Flux optique

En 2007, Naoya Ohnishi et Atsushi Imiya proposent dans [24] et [25] une méthode de segmentation en plans d'une scène 3D dans le but de diriger un robot mobile. Leur approche utilise une séquence d'images pour évaluer le flux optique et fait l'hypothèse de la présence d'un plan dominant. Le plan dominant est associé au sol et permet de savoir dans quelle zone de la scène le robot peut se diriger (voir la figure 2.6). Le flux optique correspond aux vitesses de défilement d'objets présents dans l'environnement perçues par la caméra. La première étape de l'algorithme consiste à calculer le flux optique $(\dot{x}, \dot{y})^t$ en chaque point $(x, y)^t$ entre deux images successives. Pour cela les auteurs font l'hypothèse que le flux optique de chaque pixel est constant dans son voisinage puis ils utilisent la méthode multi-résolution présentée dans [7] pour estimer ce flux optique en chaque point de l'image. La seconde étape permet de déterminer les paramètres du plan dominant de la scène. Pour cela, l'hypothèse est faite que le déplacement du robot entre deux prises de vue est petit. Cela permet d'approcher l'homographie liant deux pixels correspondant au même point de la scène vu dans deux images successives par une transformation affine. Ainsi deux images $\mathbf{p} = (x, y, 1)^t$ et $\mathbf{p}' = (x', y', 1)^t$

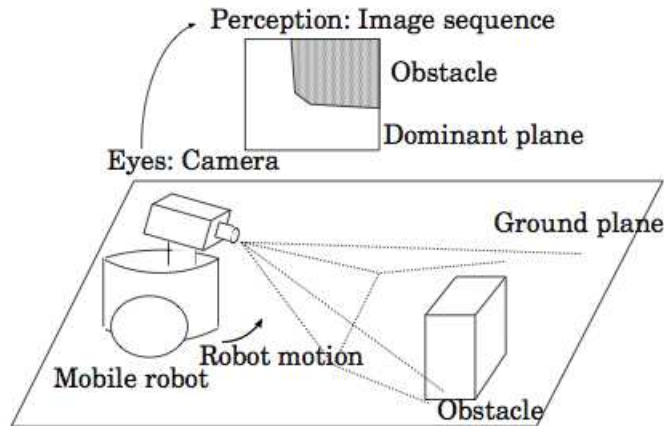


FIG. 2.6 – Modèle du robot et de son environnement ; le plan occupant la plus grande zone sur l'image est associé au sol et définit les zones accessibles au robot (figure extraite de [25]).

d'un point de la scène \mathbf{P} dans deux images successives sont liés par l'équation :

$$\mathbf{p}' = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0}^t & 1 \end{bmatrix} \mathbf{p} \quad (2.3)$$

où \mathbf{A} est une matrice (2×2) et \mathbf{b} est un vecteur à deux éléments. \mathbf{A} et \mathbf{b} forment une approximation de l'homographie \mathbf{H} . En supposant qu'il existe un plan dominant, \mathbf{A} et \mathbf{b} sont calculés avec un algorithme robuste de type RANSAC à partir de correspondances de points d'intérêt. Les paramètres \mathbf{A} et \mathbf{b} ainsi estimés, il est possible de calculer $(\hat{x}, \hat{y})^t$, le flux optique du plan dominant par :

$$(\hat{x}, \hat{y})^t = \mathbf{A}(x, y)^t + \mathbf{b} - (x, y)^t \quad (2.4)$$

Le correspondant d'un point $(x, y)^t$ de l'image à un instant t dans l'image à l'instant $t + 1$ est donné par le vecteur $(\dot{x}, \dot{y})^t$ du flux optique en ce point. Or, pour les points appartenant à la projection du plan dominant, cette transformation est aussi donnée par le vecteur $(\hat{x}, \hat{y})^t$. La recherche de ces points particuliers revient donc à trouver parmi tous les points $(x, y)^t$ de l'image, ceux dont la différence entre les deux vecteurs $(\dot{x}, \dot{y})^t$ et $(\hat{x}, \hat{y})^t$ est inférieur à un seuil, c'est-à-dire :

$$\text{Plan dominant} = \{(x, y)^t \text{ tq } \|(\dot{x}, \dot{y})^t - (\hat{x}, \hat{y})^t\| < \epsilon\}$$

En appliquant de nouveau la méthode sans prendre en compte les points affectés au plan dominant, il est possible d'obtenir de manière itérative les différents plans de la scène. Dans le contexte de guidage d'un robot, tous les plans détectés à la suite du plan principal sont considérés comme des obstacles. La figure 2.7 présente deux résultats de cet algorithme de segmentation sur des images réelles.

2.5 Zones uniformes

En 2001 et 2002, Qifa Ke et Takeo Kanade ont présenté dans [18] et [19] une méthode de détection des plans d'une scène appliquée à la compression vidéo. À chaque plan de la scène est associé un calque (un morceau d'image). Ainsi, chaque image de la vidéo peut être reconstruite à partir de ces calques en spécifiant leurs positions respectives.

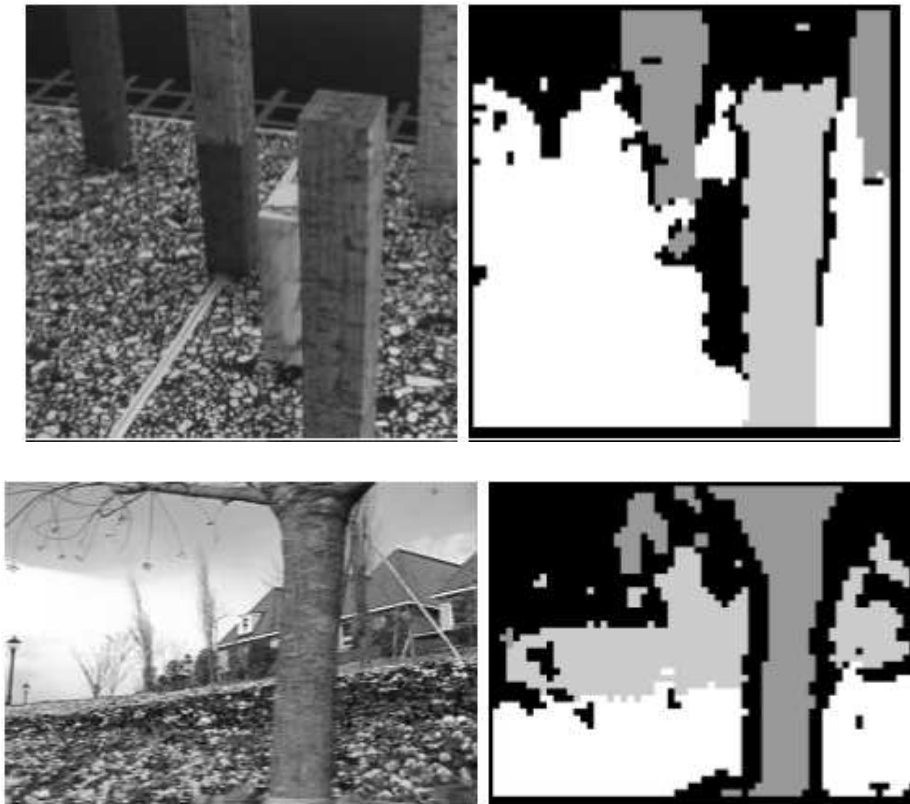


FIG. 2.7 – Exemples de segmentation itérative. Le plan dominant est blanc, les autres couleurs représentent des obstacles (figure extraite de [25]).

Les auteurs font l'hypothèse que chaque petite zone de couleur uniforme dans les images correspond à un morceau de plan de la scène. La méthode utilisée pour la détection des plans est itérative et est basée sur la mise en correspondance dans un couple d'images de régions de couleur. Dans un premier temps, une segmentation par couleur est appliquée aux images puis, après une phase de mise en correspondance de ces zones, les homographies reliant chaque couple de régions sont estimées. Les régions sont ensuite regroupées par agrandissement progressif. À chaque agrandissement d'une zone A , l'algorithme vérifie que la zone agrégée est bien compatible avec l'homographie associée à A . Il y a alternance entre les phases de calcul des homographies et les phases d'agrandissement des régions jusqu'à ce que les modifications deviennent minimales ou que plus aucun agrandissement ne soit possible. Les zones dont certains pixels ont une erreur de reprojection par l'homographie associée à la zone supérieure à un seuil sont considérées comme des outliers. Ces différentes étapes sont illustrées par la figure 2.8.

Une des applications de l'extraction de plans est ici de compresser une séquence vidéo. La segmentation finale donnée par l'algorithme permet de savoir à quelles zones des images correspondent les projections des plans détectés. Ainsi les auteurs proposent la création de calques correspondant aux différents plans de la scène et permettant de reconstruire les images de la vidéo uniquement à partir des positions des différents plans. Un exemple de calques et de reconstruction est donné dans la figure 2.9.

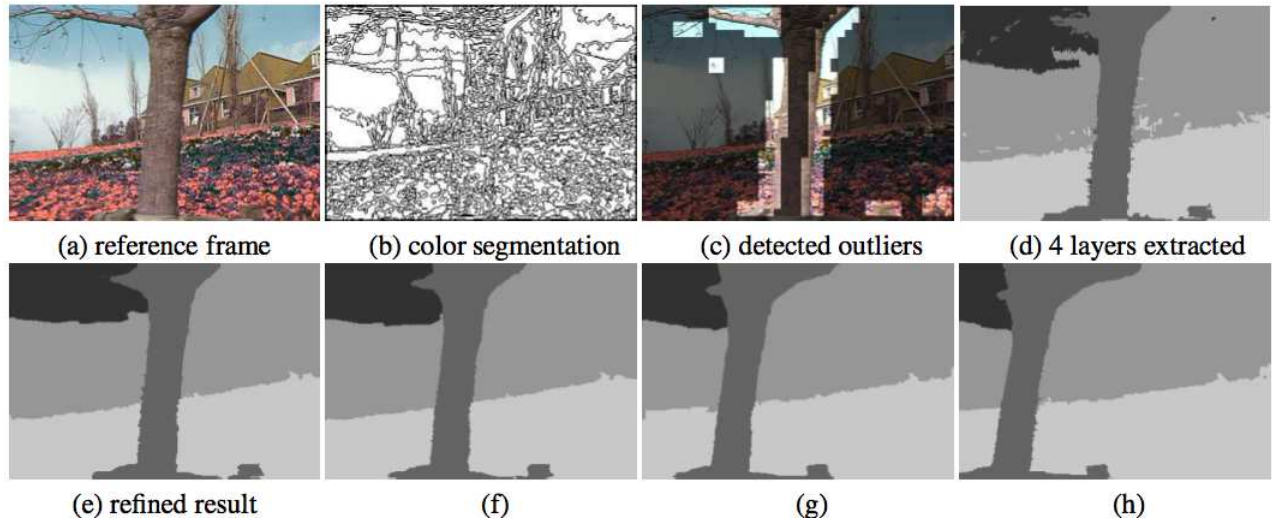


FIG. 2.8 – (a) Image de départ. (b) Segmentation en zones de couleur uniforme. (c) Détection des outliers. (d) Calques extraits. (e–h) Résultats raffinés pour l'image (a) ainsi que d'autres images de la séquence vidéo (figure extraite de [19]).

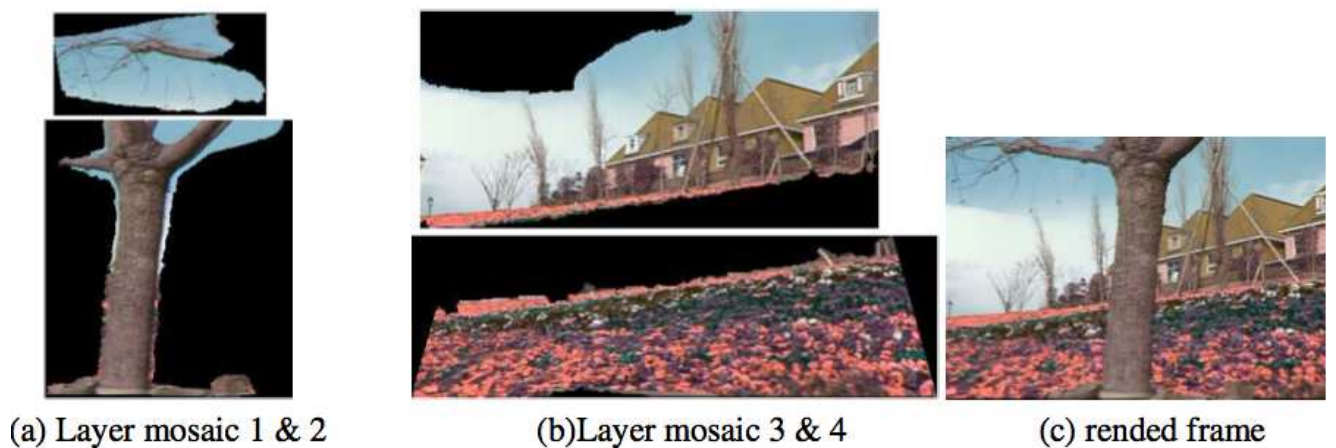


FIG. 2.9 – Extractions des calques et reconstruction d'une image de la séquence video (figure extraite de [19]).

2.6 Couples points–droites

Jusque là, nous n'avons vu que des méthodes utilisant des primitives d'un seul type (points, régions, ...). Dans [21] paru en 2002, les données en entrée sont de deux types différents. La méthode proposée pour détecter les plans d'une scène vue par deux caméras utilise des couples de points et de droites. Chaque couple point–droite représente un plan de la scène et implique donc une homographie entre les deux images. La méthode permet de faire le tri en ne conservant que les homographies correspondant à de vrais plans de la scène.

L'algorithme commence par une phase de détection de points et de droites dans les deux images. Ces points et ces droites sont ensuite mis en correspondance entre les images et la matrice fondamentale est estimée à partir des correspondances de points. Une fois la matrice fondamentale calculée, il est facile d'obtenir les épipoles gauche et droit. À partir de \mathbf{p} et \mathbf{p}' , les images d'un point \mathbf{P} de la scène, de \mathbf{l} et \mathbf{l}' , les vecteurs des paramètres des projections d'une droite \mathbf{D} de la scène, de la matrice fondamentale \mathbf{F} et des épipoles \mathbf{e} et \mathbf{e}' , il est possible de calculer l'homographie \mathbf{H} associée au plan 3D formé par \mathbf{P} et \mathbf{D} . La méthode met en évidence les plans réellement présents dans la scène de manière itérative. Dans un premier temps, toutes les homographies possibles sont calculées (chaque combinaison d'un point et d'une droite fournit une homographie). Puis une phase de vote a lieu où chaque point et chaque droite vote pour l'homographie qui minimise son erreur de reprojection (équation 1.6 pour les points et 1.11 pour les droites). Il s'agit de la distance euclidienne pour les points et d'une combinaison des erreurs sur la distance à l'origine et sur l'orientation pour les droites. L'homographie qui accumule le plus de votes est réestimée de manière robuste à partir de tous les éléments qui ont voté pour elle. Pour finir, cette homographie est utilisée pour segmenter les points et les droites en deux classes, les éléments appartenant au plan et les autres. Ceux appartenant au plan sont retirés des données et l'algorithme reprend au début jusqu'à ce qu'il n'y ait plus de point ou de droite à classer.

La figure 2.10 montre les données en entrée et le résultat obtenu sur l'exemple d'un couple d'images réelles.

Cette technique présente l'avantage de ne pas se limiter à un seul type d'informations extraites de la scène. Les points sont des primitives très souvent utilisées mais les droites, bien que très présentes dans les environnements modernes, sont assez peu exploitées. Ici, connaissant la matrice fondamentale, l'utilisation des droites permet de calculer une homographie uniquement à partir d'un point et d'une droite, alors que dans le cas de l'utilisation des points seuls il en faut au moins trois.

2.7 Correspondances de droites dans une séquence d'images

Après avoir présenté des méthodes basées sur des points puis sur des points et des droites, nous allons voir maintenant une méthode présentée en 1999 par Baillard et Zisserman dans [2] puis en 2000 dans [3] qui se base uniquement sur des correspondances de droites dans une séquence d'images. Le but de leurs travaux est de reconstruire la structure géométrique des toits de bâtiments à partir d'images aériennes.

Le principe de la méthode est de rechercher dans les images les segments de droites qui correspondent aux bords des toits puis de retrouver, s'il existe, le morceau de plan attaché à chacun de ces segments. La recherche des morceaux de plans se fait d'abord de manière indépendante pour chaque segment puis les résultats correspondant aux mêmes plans de la scène sont fusionnés. Enfin, les caméras étant calibrées, une reconstruction euclidienne de la structure des toits est faite à la fin de l'algorithme.

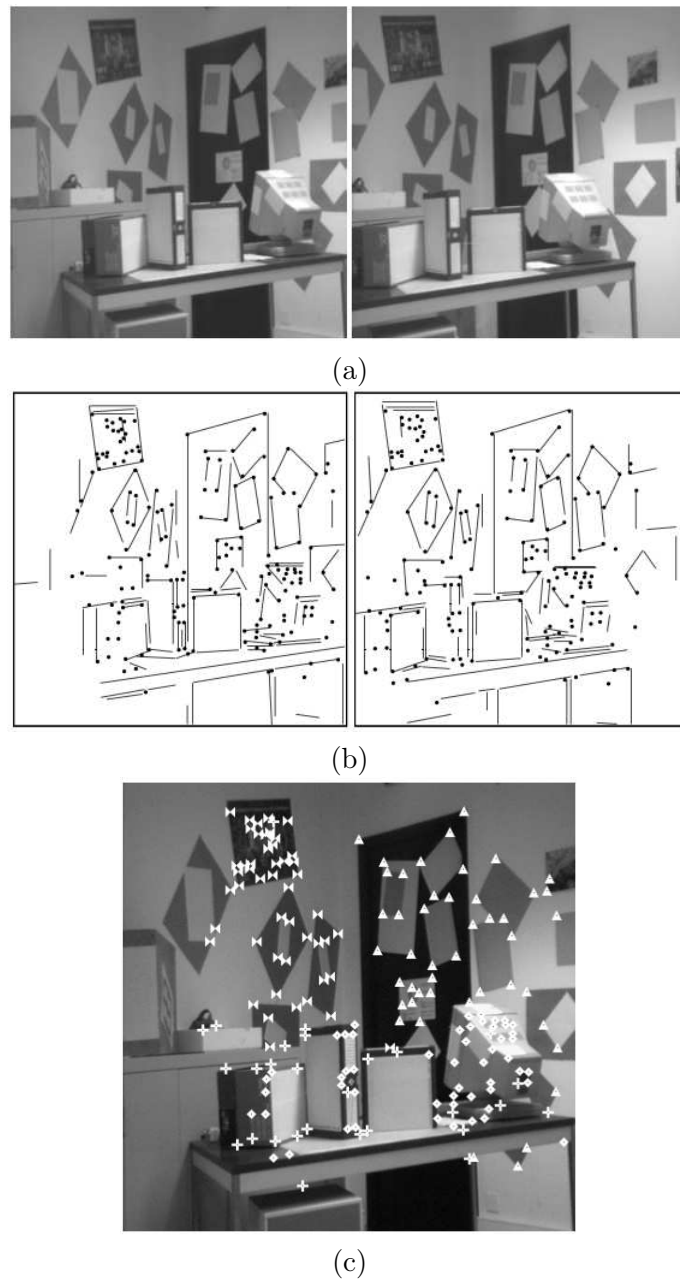


FIG. 2.10 – (a) Images gauche et droite, (b) détection des primitives (points et segments de droites), (c) classifications des points dans les différents plans calculés, chaque forme de point correspondant à un plan (figure extraite de [21]).

Les segments sont détectés dans les images à partir de points de contours mis en évidence par un filtre de Canny [9] puis, après une phase de mise en correspondance entre les images, leurs positions 3D sont calculées. À chaque segment 3D est associé un faisceau de plans $\pi(\theta)$ dont le seul paramètre est l'angle θ . Pour trouver le demi-plan associé à un segment (s'il existe, le demi-plan est un morceau de toit contenant le segment) il suffit de faire varier θ et de sélectionner la valeur qui donne le meilleur résultat vis-à-vis du critère suivant. Pour évaluer si un demi-plan correspond à un vrai plan de la scène, les auteurs proposent de vérifier que les points aux alentours du segment se projettent bien sur leur correspondant dans les autres images par l'homographie induite par le plan. Si une homographie projette correctement les points alors le demi-plan associé est considéré comme correct. Si aucune valeur de θ ne donne de résultat acceptable alors le segment est considéré comme n'appartenant à aucun plan.

Une fois les demi-plans détectés, l'algorithme procède à une étape de fusion des demi-plans similaires. Deux segments colinéaires dont les demi-plans associés sont presque identiques (angles θ proches) sont fusionnés; de même, deux segments, qui sont proches dans la scène et dont les demi-plans correspondants sont compatibles, sont fusionnés. Deux demi-plans sont compatibles s'ils peuvent correspondre à la même face d'un toit. Ces deux cas sont illustrés dans la figure 2.11. À la suite de l'étape de fusion des demi-plans, l'algorithme recherche si certains segments n'ont pas été oubliés lors de l'étape de détection. Pour cela, de nouveaux segments sont créés quand deux plans voisins s'intersectent de manière consistante, c'est-à-dire quand la frontière appartient aux deux demi-plans. Cette opération est présentée dans la figure 2.12.

Cette méthode est entièrement automatique et permet de détecter un nombre inconnu de plans. Néanmoins elle n'est applicable que dans le cas de la reconstruction de toits puisqu'elle se base sur l'hypothèse que les segments utilisés se trouvent sur le bord des demi-plans de la scène et le simple fait de manquer un segment peut empêcher de détecter le plan associé.

2.8 Conclusion

Il existe bien d'autres méthodes de détection de plans. Celles qui ont été présentées ici sont parmi les plus récentes et se basent toutes sur des données initiales (points, droites, flux optique, régions de couleur, ...) et des hypothèses sur la structure de la scène (facettes planes polygonales, demi-plans, plan dominant, ...) différentes. Le tableau 2.1 présente un récapitulatif des méthodes présentées. Elles sont toutes adaptées au contexte pour lequel elles ont été prévues et leurs résultats se dégradent très vite si les hypothèses, assez fortes pour certaines, ne sont plus respectées. L'utilisation de l'analyse généralisée en composantes principales semble pouvoir s'adapter à tous types de contraintes. C'est pourquoi nous allons en faire une étude plus approfondie dans la suite de ce rapport. Puis nous nous replacerons dans le contexte de la segmentation de scènes urbaines en émettant l'hypothèse que les différents plans présents dans la scène sont délimités par leurs intersections. Nous présenterons une nouvelle méthode de segmentation basée sur cette hypothèse.

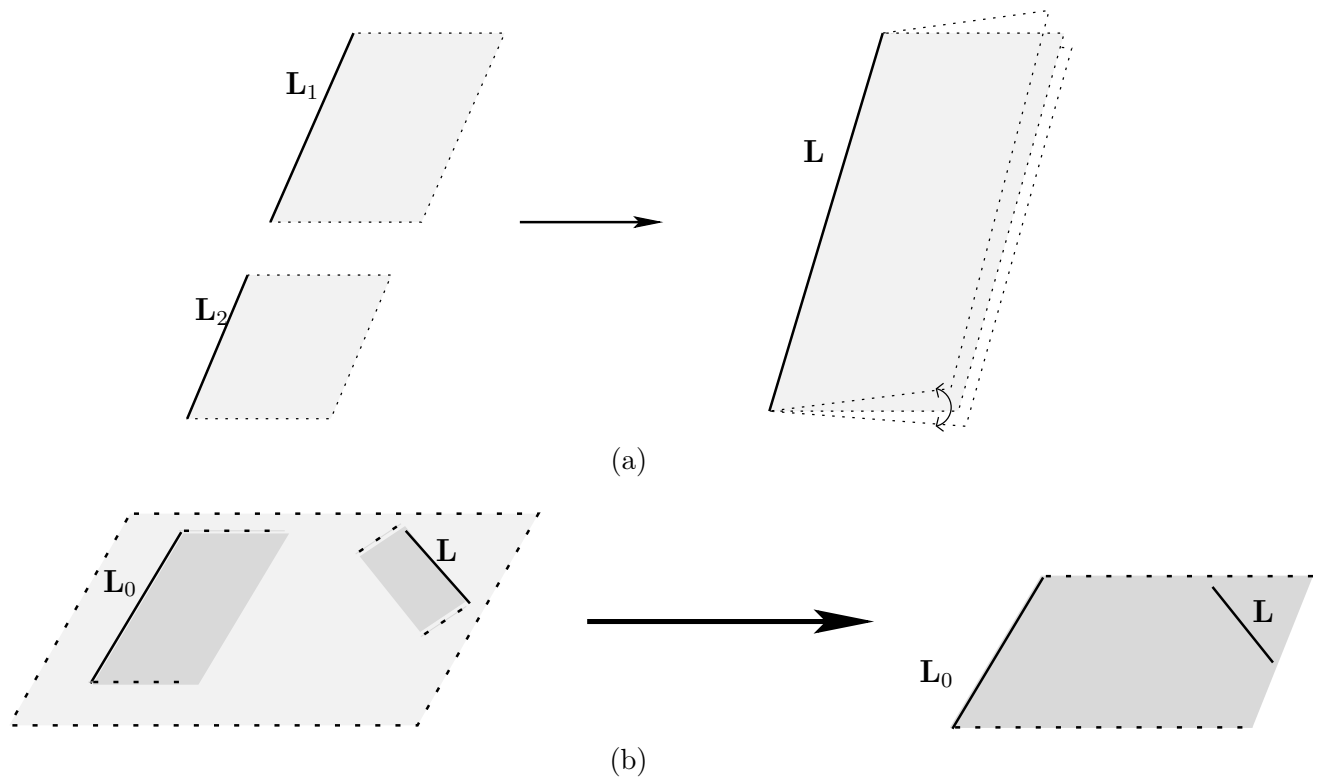


FIG. 2.11 – (a) Fusion de deux plans ayant presque le même angle il est probable que les deux segments de droite L_1 et L_2 appartiennent à une même droite qui a été coupée lors de la détection. (b) Fusion de deux demi-plans provenant de segments proches.

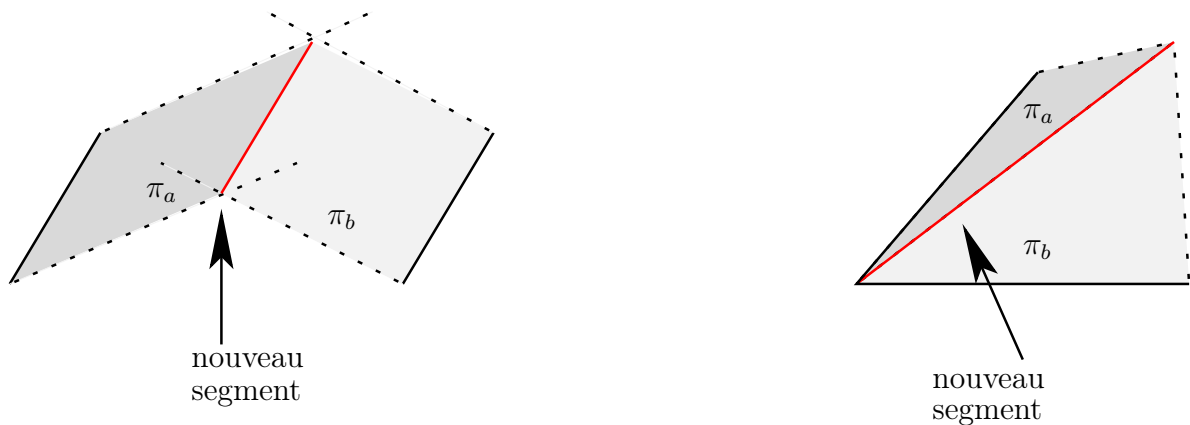
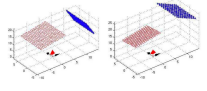

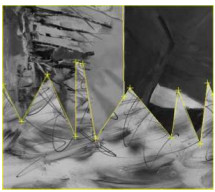



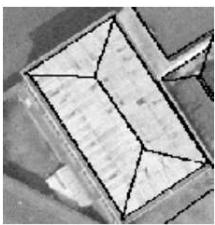


FIG. 2.12 – De nouveaux segments sont créés à l'intersection de demi-plans proches. Ces nouveaux segments sont utilisés pour augmenter la quantité de données utilisées pour calculer les homographies afin d'améliorer les estimations, mais aussi pour délimiter les plans et ainsi faciliter l'étape de reconstruction 3D. Les deux cas présentés correspondent au faîte et au coin d'un toit.

Méthode	Auteurs	Année	Données en entrée	Exemple
GPCA	René Vidal et Yi Ma	2005	Points	
Dérivée de RAN-SAC et contrainte photométrique	Adrien Bartoli	2007	Points	
Semi-automatique, première étape manuelle	Benoît Bocquillon	2004	Segmentation manuelle	
Flux optique	Naoya Ohnishi et Atsushi Imiya	2007	Points	
Zones uniformes associées à des plans puis propagation	Qifa Ke et Takeo Kanade	2001	Zones de couleur uniforme	
Couples points-droites et système de vote	Manolis I.A. Lourakis, Antonis A. Argyros et Stelios C. Orphanoudakis	2002	Points et droites	
Détection des plans à partir de droites par essais successifs	Caroline Baillard et Andrew Zisserman	1999	Droites	

TAB. 2.1 – Récapitulatif des méthodes de segmentation en plans par ordre de présentation dans ce document.

Chapitre 3

Analyse Généralisée en composantes principales

3.1 Présentation et espérances

Comme nous l'a montré l'étude bibliographique, la grande majorité des méthodes de segmentation en plans sont itératives ou se basent sur des hypothèses assez fortes. Les méthodes itératives ont l'avantage de faire moins d'hypothèses sur la structure de la scène mais elles sont relativement coûteuses en temps de calcul. À l'inverse, les méthodes qui posent des hypothèses fortes sur la scène sont souvent plus rapides et plus précises mais leurs résultats se dégradent très vite lorsqu'elles sont utilisées dans le cas de scènes inadaptées.

Nous aimerions donc disposer d'une méthode permettant de détecter un ensemble quelconque de plans de manière simultanée. C'est ce que nous avons essayé de mettre en œuvre en utilisant l'analyse généralisée en composantes principales. Il s'agit d'une méthode assez peu répandue permettant de retrouver le nombre, les dimensions et les bases de sous-espaces présents dans un ensemble de données. Dans sa thèse [28], René Vidal présente d'une manière plus formelle le problème que propose de résoudre l'analyse généralisée en composantes principales. À partir d'un ensemble $\mathbf{X} = \{\mathbf{x}^j \in \mathbb{R}^K\}$, avec $j \in [1..N]$, de points appartenant à $n > 1$ sous-espaces différents $\{S_i \subseteq \mathbb{R}^K\}$, avec $i \in [1..n]$ de dimension $k_i = \dim(S_i)$, $0 < k_i < K$, il s'agit d'identifier chaque sous-espace S_i sans connaissance préalable sur l'appartenance des points aux différents S_i .

L'identification des sous-espaces comprend :

1. le calcul du nombre n de sous-espaces ainsi que leurs dimensions respectives $\{k_i\}$, $i \in [1..n]$;
2. le calcul d'une base de chaque sous-espace (ou d'un sous-espace orthogonal) ;
3. la classification des N points dans leurs sous-espaces respectifs.

3.1.1 Principe de l'analyse généralisée en composantes principales

Rappelons qu'un polynôme à plusieurs indéterminées x_1, x_2, \dots, x_D est une somme finie de monômes $ax_1^{n_1} x_2^{n_2} \dots x_D^{n_D}$, où a est un scalaire et les n_i sont des entiers positifs ou nuls. Le *degré du monôme* est $\sum_{i=1}^D n_i$ et le *degré du polynôme* est le plus grand degré de tous les monômes qui le constituent.

Un polynôme à plusieurs indéterminées est *homogène* (de degré n) si et seulement si chacun des monômes est de même degré n .

Soit \mathbf{x} un point de \mathbb{R}^D (que nous écrirons sous la forme $\mathbf{x} = (x_1, \dots, x_D)^t$). Le principe de l'analyse généralisée en composantes principales est de représenter l'union de n sous-espaces par un équation algébrique de degré n de la forme :

$$P_n(\mathbf{x}) = 0, \tag{3.1}$$

où

$$P_n(\mathbf{x}) \equiv \prod_{i=1}^n \mathbf{b}_i^t \mathbf{x} \quad (3.2)$$

est un polynôme homogène de degré n et chaque \mathbf{b}_i est une base du complément du $i^{\text{ème}}$ sous-espace. Cette équation s'annule en tout les points de l'ensemble \mathbf{X} puisque, si un point \mathbf{x} appartient au $i^{\text{ème}}$ sous-espace, alors $\mathbf{b}_i^t \mathbf{x} = 0$.

Le polynôme obtenu n'est pas linéaire en les données mais il peut être réécrit sous la forme du produit scalaire suivant :

$$P_n(\mathbf{x}) = \mathbf{c}_n^t \boldsymbol{\nu}_n(\mathbf{x}) \quad (3.3)$$

où \mathbf{c}_n est un vecteur de $\mathbb{R}^{\binom{n+D-1}{D}}$ et $\boldsymbol{\nu}_n(\mathbf{x})$ la **carte de Véronèse**¹ de degré n de \mathbf{x} , définie par l'application de \mathbb{R}^D sur $\mathbb{R}^{\binom{n+D-1}{D}}$:

$$\boldsymbol{\nu}_n : \mathbf{x} = [x_1, \dots, x_D]^t \in \mathbb{R}^D \mapsto \boldsymbol{\nu}_n(\mathbf{x}) = [\dots, x_1^{n_1} x_2^{n_2} \dots x_D^{n_D}, \dots]^T \in \mathbb{R}^{C_n^{n+D-1}}. \quad (3.4)$$

La carte de Véronèse $\boldsymbol{\nu}_n(\mathbf{x})$ est un vecteur qui empile les monômes $x_1^{n_1} x_2^{n_2} \dots x_D^{n_D}$ de degré $n = \sum_{i=1}^D n_i$. Pour tout n et D , notons que le *coefficient binomial* C_n^{n+D-1} (aussi noté $\binom{n+D-1}{n}$) vaut :

$$C_n^{n+D-1} = \frac{(n+D-1)!}{n!(D-1)!} = C_{D-1}^{n+D-1}.$$

Par exemple, la carte de Véronèse de $\mathbf{x} = (x_1, x_2, x_3)^t$ est

$$\boldsymbol{\nu}_2(\mathbf{x}) = (x_1^2, x_1 x_2, x_1 x_3, x_2^2, x_2 x_3, x_3^2)^t.$$

Ainsi le polynôme $P_n(\mathbf{x})$ est linéaire en les éléments de \mathbf{c}_n . Il est donc possible, avec suffisamment de données, de calculer ce vecteur. Nous verrons comment à la fin de la partie 3.2.1.

Une fois le polynôme identifié, il est possible d'obtenir les bases des compléments de chaque sous-espaces (les vecteurs \mathbf{b}_i) en calculant la dérivée de P_n en un point de chaque sous-espace.

$$\mathbf{b}_i = \frac{DP_n(\mathbf{y}_i)}{\|DP_n(\mathbf{y}_i)\|}, \quad i = 1, \dots, n, \quad \text{où } DP_n(\mathbf{x}) \equiv \frac{\partial P_n(\mathbf{y}_i)}{\partial \mathbf{x}}. \quad (3.5)$$

Démonstration. La dérivée de P_n par rapport à \mathbf{x} au point \mathbf{y} est donnée par :

$$\begin{aligned} DP_n(\mathbf{y}) &= \frac{\partial P_n(\mathbf{y})}{\partial \mathbf{x}} \\ &= \frac{\partial}{\partial \mathbf{x}} \prod_{j=1}^n (\mathbf{b}_j^t \mathbf{y}) \\ &= \sum_{j=1}^n (\mathbf{b}_j) \prod_{\substack{l=1 \\ l \neq j}}^n (\mathbf{b}_l^t \mathbf{y}). \end{aligned} \quad (3.6)$$

Or, en un point $\mathbf{y}_i \in S_i$, c.-à-d. vérifiant $\mathbf{b}_i^t \mathbf{y}_i = 0$, tous les termes de l'équation 3.6 sauf le $i^{\text{ème}}$ s'annulent car $\prod_{l \neq j} (\mathbf{b}_l^t \mathbf{y}_i) = 0$ pour $j \neq i$. Nous avons donc :

$$DP_n(\mathbf{y}_i) = \lambda \mathbf{b}_i. \quad (3.7)$$

Donc l'équation 3.5 donne bien la valeur normalisée du vecteur \mathbf{b}_i . ■

1. En référence à la traduction anglaise “*Veronese map*”, nous préférons utiliser le terme “carte de Véronèse” plutôt que “plongement de Véronèse”.

Cette méthode est dite semi-supervisée car elle nécessite de connaître un point, \mathbf{y}_i , de chacun des sous-espaces. D'autre part, la dimension du sous-espace S_i est donnée par

$$\dim(S_i) = D - \text{rang}(DP_n(\mathbf{y}_i)).$$

Dans la mesure où chaque point appartient à au moins un des sous-espaces, il est possible de choisir un point au hasard. Une fois les bases de chaque sous-espace identifiées, il est très simple de classer les points de l'ensemble \mathbf{X} dans leurs sous-espaces respectifs. Nous verrons dans l'exemple ci dessous comment choisir les points dans le cas de données bruitées.

3.1.2 Exemple

Prenons l'exemple simple donné dans [30] et illustré par la figure 3.1. L'ensemble \mathbf{X} des points

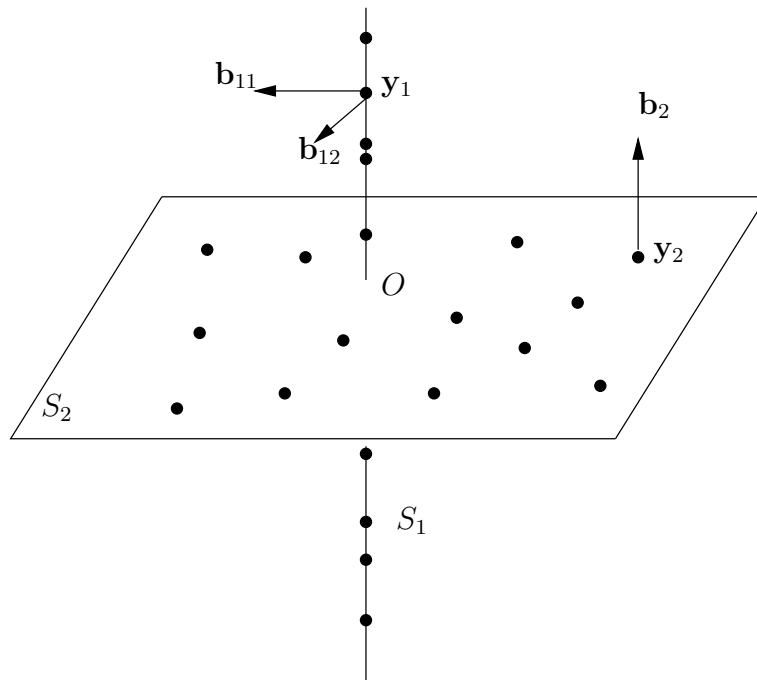


FIG. 3.1 – Les points \mathbf{x} de \mathbb{R}^3 sont répartis sur 2 sous-espaces S_1 et S_2 de dimensions $k_1 = 1$ et $k_2 = 2$.

(de la forme $(x_1, x_2, x_3)^t$) peut être divisé en deux ($n = 2$) sous-espaces : une droite $S_1 = \{\mathbf{x} \mid x_1 = x_2 = 0\}$ et un plan $S_2 = \{\mathbf{x} \mid x_3 = 0\}$. Un polynôme qui s'annule en tous points de $S_1 \cup S_2$ est par exemple $P_2(\mathbf{x}) = x_1x_3 + x_2x_3$. De manière plus générale, deux sous-espaces dans \mathbb{R}^3 peuvent être représentés par un polynôme de la forme :

$$P_2(\mathbf{x}) = c_1x_1^2 + c_2x_1x_2 + c_3x_1x_3 + c_4x_2^2 + c_5x_2x_3 + c_6x_3^2 = 0$$

Calculons maintenant la dérivée de P_2 en deux points $\mathbf{y}_1 = (0,0,1)^t \in S_1$ et $\mathbf{y}_2 = (1,1,0)^t \in S_2$.

$$\begin{aligned} P_2(\mathbf{x}) &= [x_1x_3, x_2x_3] \\ DP_2(\mathbf{x}) &= \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix} \\ \text{Donc :} & \\ DP_2(\mathbf{y}_1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ DP_2(\mathbf{y}_2) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

$DP_2(\mathbf{y}_1)$ nous donne bien les deux vecteurs \mathbf{b}_{11} et \mathbf{b}_{12} de la figure 3.1 ; il en est de même pour $DP_2(\mathbf{y}_2)$ qui nous fournit \mathbf{b}_2 . Il s'agit respectivement des bases des sous-espaces orthogonaux à S_1 et S_2 .

Ici nous avons donné directement un point de chaque sous-espace, mais dans le cas où l'on ne dispose d'aucune information concernant la classification des données et surtout dans le cas où ces données sont bruitées, il vaut mieux ne pas choisir les points au hasard. René Vidal et Yi Ma proposent dans [30] de choisir dans un premier temps le point qui minimise la distance algébrique $d_2(\mathbf{x})^2 = p_{21}(\mathbf{x})^2 + p_{22}(\mathbf{x})^2$, où $p_{21}(\mathbf{x})$ et $p_{22}(\mathbf{x})$ sont les deux (dans notre exemple) éléments de $P_2(\mathbf{x})$, à savoir $p_{21}(\mathbf{x}) = x_1x_3$ et $p_{22}(\mathbf{x}) = x_2x_3$. Le point choisi permet d'obtenir une base \mathbf{b}_i du sous-espace auquel il appartient. Pour choisir les points suivants de manière à ce qu'ils n'appartiennent pas à un sous-espace déjà calculé, il suffit de diviser le polynôme $P_2(\mathbf{x})$ par $\mathbf{b}_i^t \mathbf{x}$ et de réécrire la distance algébrique en conséquence. Dans notre exemple, en admettant que le premier point choisi appartienne à S_2 , cela donne :

$$\begin{aligned} P_1(\mathbf{x}) &= \frac{p_{21}(\mathbf{x})}{\mathbf{b}_2^t \mathbf{x}} + \frac{p_{22}(\mathbf{x})}{\mathbf{b}_2^t \mathbf{x}} \\ &= p_{11}(\mathbf{x}) + p_{12}(\mathbf{x}) \\ &= x_1 + x_2. \end{aligned}$$

La nouvelle distance algébrique à minimiser pour choisir le prochain point devient donc :

$$d_1(\mathbf{x})^2 = p_{11}(\mathbf{x})^2 + p_{12}(\mathbf{x})^2 = x_1^2 + x_2^2.$$

Cette méthode est itérative et permet de trouver toutes les bases des sous-espaces. Une fois cette étape terminée, il ne reste plus qu'à classer les données. Le sous-espace correspondant à chaque point est fourni très simplement de la manière suivante :

$$\mathcal{C}(\mathbf{x}_i) = \underset{j=1..N}{\operatorname{argmin}} \|\mathbf{b}_j^t \mathbf{x}_i\|. \quad (3.8)$$

Après cette présentation rapide du principe de l'analyse généralisée en composantes principales, voyons maintenant comment cette méthode peut être appliquée à la détections de plans.

3.2 Mise en œuvre

Dans [33] les auteurs présentent une application de l'analyse généralisé en composantes principales à la détection de plans dans des couples d'images. La méthode se base sur des couples de points mis

en correspondance entre les deux images et sur les contraintes apportées par les homographies inter-images induites par les plans de la scène. Les résultats présentés semblent prometteurs bien qu'il ne s'agisse que de tests sur des images de synthèse assez simples. Nous avons donc décidé d'évaluer l'efficacité de la méthode sur des images réelles et d'adapter la méthode pour qu'elle utilise des droites à la place des points.

3.2.1 Adaptation aux droites

Étant donnée la dualité qui existe entre les points et les droites dans un espace projectif de dimension deux, la méthode des points de [33] peut s'appliquer "dualmente" aux droites. La relation sur laquelle elle se base est la relation homographique donnée par l'équation 1.6 qui relie un point et son homologue dans l'autre image. Nous avons vu (équation 1.11) qu'il existe une équation équivalente qui s'applique aux droites.

Complexification de l'homographie. La relation $\mathbf{l}_d \sim \mathbf{H}^{-t}\mathbf{l}_g$ ne permet pas d'obtenir directement un polynôme de la même forme que celui donné dans l'équation 3.1. Il faut donc l'adapter en créant artificiellement une forme bilinéaire strictement équivalente à la relation homographique, par "complexification" de cette dernière.

Dans ce qui suit, i ne désignera plus un indice mais $\sqrt{-1}$.

En posant $\mathbf{G} = \mathbf{H}^{-t}$ nous pouvons écrire :

$$\mathbf{l}_d \sim \mathbf{G}\mathbf{l}_g \iff \begin{bmatrix} a_d \\ b_d \\ c_d \end{bmatrix} \sim \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} a_g \\ b_g \\ c_g \end{bmatrix}. \quad (3.9)$$

La complexification de l'homographie consiste à réécrire cette relation sous la forme suivante :

$$\begin{bmatrix} a_d + ib_d \\ c_d \end{bmatrix} \sim \begin{bmatrix} g_{11} + ig_{21} & g_{12} + ig_{22} & g_{13} + ig_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} a_g \\ b_g \\ c_g \end{bmatrix}. \quad (3.10)$$

Notons $\tilde{\mathbf{G}} \in \mathcal{M}(\mathbb{C}, 2, 3)$ la matrice de l'homographie complexifiée et soit le vecteur issu de la complexification de la droite \mathbf{l}_d défini par :

$$\mathbf{k}_d \equiv \begin{bmatrix} c_d \\ -a_d + ib_d \end{bmatrix} \in \mathbb{C}^2. \quad (3.11)$$

Il est facile de vérifier que :

$$\mathbf{k}_d^* \tilde{\mathbf{G}} \mathbf{l}_g = 0, \quad (3.12)$$

où \mathbf{k}_d^* est le transposé conjugué de \mathbf{k}_d . Cette équation est vraie pour tout couple de droites $(\mathbf{l}_g, \mathbf{l}_d)$ en relation homographique, c.-à-d. vérifiant (3.9).

Pôle homographique complexe Appelons *pôle homographique complexe* de \mathbf{l}_d par rapport à \mathbf{G} , le point complexe de l'image gauche défini par :

$$\mathbf{q}_g \equiv \tilde{\mathbf{G}}^* \mathbf{k}_d \in \mathbb{C}^3. \quad (3.13)$$

Il est facile de vérifier que ce point \mathbf{q}_g et son conjugué $\bar{\mathbf{q}}_g$ appartiennent à \mathbf{l}_g , la droite de l'image gauche en relation homographique avec \mathbf{l}_d , via \mathbf{G} . En fait, on peut montrer que

$$\mathbf{l}_g^t \mathbf{q}_g = 0 \iff \mathbf{l}_g^t \bar{\mathbf{q}}_g = 0,$$

ce qui est une condition strictement équivalente à

$$(\Re \mathbf{q}_g)^t \mathbf{1}_g = 0 \quad \text{et} \quad (\Im \mathbf{q}_g)^t \mathbf{1}_g = 0,$$

où \Re et \Im désignent les parties réelles et imaginaires d'un complexe.

Étant données N paires de droites homologues $\{(\mathbf{l}_g^k, \mathbf{l}_d^k)\}$, il existe, dans l'image gauche, un ensemble de N pôles homographiques complexes, que nous dénotons par $\{\mathbf{q}_g^k\}$. D'une part, la matrice $\tilde{\mathbf{G}}$ étant de rang 2, nous en déduisons qu'il existe un vecteur \mathbf{e}_g tel que

$$\tilde{\mathbf{G}} \mathbf{e}_g = \mathbf{0}.$$

D'autre part, puisque, quelque soit k ,

$$0 = \mathbf{k}_d^{k*} \tilde{\mathbf{G}} \mathbf{e}_g = \mathbf{q}_g^k \mathbf{e}_g,$$

il s'ensuit que le vecteur \mathbf{e}_g représente une droite qui est le lieu des pôles homographiques complexes \mathbf{q}_g^k des N droites de l'image droite, dans l'image gauche.

Cette remarque nous sera très utile pour la segmentation des données.

Estimation de l'homographie multi-plans En admettant que nous avons un ensemble L de couples de droites mises en correspondance appartenant à n plans de la scène, chaque couple de droites doit vérifier l'équation suivante, puisqu'au moins un des éléments du produit est nul :

$$\prod_{j=1}^n (\mathbf{k}_d^* \tilde{\mathbf{G}}_j \mathbf{l}_g) = 0. \quad (3.14)$$

En utilisant le même principe que pour l'équation 3.3, l'équation 3.14 peut être réécrite sous la forme suivante en utilisant les cartes de Véronèse de degré n des vecteurs \mathbf{k}_d et \mathbf{l}_g :

$$\nu_n(\mathbf{k}_d)^* \tilde{\mathcal{G}} \nu_n(\mathbf{l}_g) = 0. \quad (3.15)$$

La matrice $\tilde{\mathcal{G}}$ est appelée homographie multi-plans, traduction du terme “*multi-plane homography matrix*” utilisé dans [33]. Cette équation bilinéaire peut être transformée en une équation linéaire de la forme suivante :

$$(\nu_n(\mathbf{k}_d) \otimes \nu_n(\mathbf{l}_g))^* \tilde{\mathcal{G}}^S = 0, \quad (3.16)$$

où \otimes est le produit tensoriel (ou produit de Kronecker) de deux vecteurs et $\tilde{\mathcal{G}}^S$ est le vecteur formé en empilant les colonnes de la matrice $\tilde{\mathcal{G}}$; à noter que $(\nu_n(\mathbf{k}_d) \otimes \nu_n(\mathbf{l}_g)) \in \mathcal{C}^{\binom{n+2-1}{n} \binom{n+3-1}{n}}$.

Étant donné que chacun des N couples de droites $(\mathbf{l}_g^j, \mathbf{l}_d^j)$, $j \in [1..N]$ vérifie cette équation², nous pouvons construire une matrice \mathcal{L}_n de taille $\binom{n+2-1}{n} \binom{n+3-1}{n} \times N$ et qui a la forme suivante :

$$\mathcal{L}_n = [(\nu_n(\mathbf{k}_d^1) \otimes \nu_n(\mathbf{l}_g^1)), \dots, (\nu_n(\mathbf{k}_d^N) \otimes \nu_n(\mathbf{l}_g^N))],$$

et qui vérifie la relation :

$$\mathcal{L}_n^* \tilde{\mathcal{G}}^S = 0. \quad (3.17)$$

Il est donc possible d'estimer les paramètres du vecteur $\tilde{\mathcal{G}}^S$ en utilisant la technique des moindres carrés totaux étant donné que la matrice \mathcal{L}_n est connue. Lorsque $N > \binom{n+2-1}{n} \binom{n+3-1}{n}$, une solution unique est donnée par le vecteur singulier de \mathcal{L}_n^* associé à la plus petite valeur singulière.

Nombre de plans	nombre minimum de données
2	18
3	40
4	75
5	126
...	...
n	$\binom{n+2-1}{n} \binom{n+3-1}{n}$

TAB. 3.1 – Correspondance entre nombre de plans et nombre minimum de données.

Le tableau 3.1 donne un aperçu du nombre minimum de couples de droites nécessaires. Sachant que dans le cas de données bruitées il vaut toujours mieux avoir beaucoup plus de données, ce nombre peut devenir un handicap pour des valeurs élevées de n .

Nous avons supposé jusqu'ici que le nombre n de plans était connu ; nous allons voir maintenant comment le calculer à partir des données puis nous aborderons enfin le problème de la segmentation des données.

3.2.2 Calcul du nombre de plans

Le calcul du nombre de plans est une étape cruciale dans les méthodes, comme celle-ci, qui proposent de trouver simultanément l'ensemble des plans. Si le nombre n de plans est mal évalué, nous risquons non seulement de détecter trop ou trop peu de plans mais surtout, les calculs suivants étant très dépendants de n , ils ont de grandes chances d'être complètement erronés.

La méthode présentée dans [33] propose de calculer le nombre de plans en étudiant le rang de la matrice \mathcal{L}_n . En effet, \mathcal{L}_n a un noyau de dimension 1 lorsque n correspond au nombre de plans présents. Donc, idéalement n est donné par :

$$n = \underset{m}{\operatorname{argmin}} \{ \operatorname{rang}(\mathcal{L}_m) = \binom{n+2-1}{n} \binom{n+3-1}{n} - 1 \}. \quad (3.18)$$

Néanmoins pour le cas de données bruitées, les auteurs proposent d'utiliser le critère suivant supposé donner de bons résultats :

$$n = \underset{m}{\operatorname{argmin}} \left\{ \frac{\sigma_{\binom{m+2-1}{m} \binom{m+3-1}{m}}^2(\mathcal{L}_m)}{\sum_{k=1}^{\binom{m+2-1}{m} \binom{m+3-1}{m} - 1} \sigma_k^2(\mathcal{L}_m)} + \kappa \binom{m+2-1}{m} \binom{m+3-1}{m} \right\}, \quad (3.19)$$

où $\sigma_j(\mathcal{L}_m)$ est la $j^{\text{ème}}$ valeur singulière de la matrice \mathcal{L}_m et κ est une petite valeur positive. Nous l'avons fixée à $\kappa = 10^{-16}$ dans nos tests. Le principe de cette formule est de trouver la plus petite valeur de m qui fait que la dernière valeur singulière (au numérateur) de \mathcal{L}_m soit la plus proche possible de 0. Cette valeur devrait être nulle pour la bonne valeur de m , mais, en présence de bruit, cela risque de ne pas être le cas ; c'est pourquoi le score est majoré par $\kappa \binom{m+2-1}{m} \binom{m+3-1}{m}$. Cet ajout permet de favoriser les petite valeurs de m pour lesquelles la dernière valeur singulière de \mathcal{L}_m n'est pas tout à fait nulle, mais presque. En effet, plus m augmente et plus la valeur de la dernière valeur singulière décroît.

Nous pouvons voir facilement que la valeur maximale de m testée dépend de la quantité de données

2. L'indice supérieur correspond au numéro du couple (de 1 à N) et l'indice inférieur à l'image contenant la droite (g ou d).

disponibles, comme cela est montré dans le tableau 3.1. Nous présenterons l'efficacité de cette technique dans la section 3.3.

3.2.3 Segmentation des données

$$q^j \sim (\nu_n(\mathbf{k}_d^j) \otimes \frac{\partial \nu_n(\mathbf{l}_g^j)}{\partial \mathbf{l}_g})^* \tilde{\mathcal{G}}^S \in \mathcal{C}^3, \quad (3.20)$$

Le pôle homographique complexe q^j (cf §3.2.1) obtenu à l'équation précédente vérifie $\mathbf{l}_g^t q^j = 0$ pour chaque paire $(\mathbf{l}_g^j, \mathbf{k}_d^j)$.

Si il existe n plans dans la scène, chaque sous-espace est associé avec une droite $\tilde{\mathbf{e}}^j$, $j \in [1..n]$, lieu de tous les pôles homographiques complexes. Nous pouvons donc écrire la relation suivante qui est vraie pour tout q :

$$\prod_{j=1}^n (q^* \cdot \tilde{\mathbf{e}}^j) = 0 \quad (3.21)$$

Cela nous ramène à une équation de la forme de celle que se propose de résoudre l'analyse généralisée en composantes principales (voir equation 3.1). D'après ce que nous avons vu en 3.1.1, nous pouvons écrire :

$$P_n(q) = \prod_{j=1}^n (q^* \tilde{\mathbf{e}}^j) = \mathbf{a}_n^* \nu_n(q) = 0. \quad (3.22)$$

Avec suffisamment de droites, nous pouvons calculer \mathbf{a}_n^* de la même manière que $\tilde{\mathcal{G}}^S$ dans la section 3.2.1.

une fois les coefficients du polynôme estimés, nous pouvons obtenir les coefficients $\tilde{\mathbf{e}}^j$ de manière itérative en suivant la démarche présentée à la fin de la section 3.1.2. La dernière étape consiste à affecter à chaque couple de droites la classe qui lui convient le mieux, c'est-à-dire celle fournie par l'équation 3.8.

3.3 Expérimentations

Nous allons maintenant présenter les résultats obtenus par la méthode décrite sur des images de synthèse puis sur un couple d'images réelles. Les tests ont été effectués en utilisant des couples de droites ou de points connus *a priori*. Nous présenterons dans un premier temps les résultats de la phase de segmentation, c'est-à-dire en supposant que le nombre de plans a été correctement trouvé. Puis nous évaluerons à la fin de cette section le calcul du nombre de plans.

3.3.1 Données de synthèses

Dans un premier temps, nous avons testé l'algorithme sur des données de synthèse : d'abord des droites puis des points. Un avantage non négligeable de cette méthode est que du fait de la dualité point-droite, il n'y a rien à modifier dans le code Matlab lorsque le type de données en entrée change.

3.3.1.1 Droites

La scène de synthèse que nous utilisons est composée d'un cube. Sur le couple d'images générés, trois faces sont visibles. Nous plaçons de manière régulière (quadrillage) 60 droites sur chacune des faces puis nous lançons l'algorithme qui nous fournit un vecteur contenant les classes des différentes

droites.

Les paramètres des droites dans les deux images sont représentés par la matrice de Plücker [16, chap 3, page 70]. Si \mathbf{A} et \mathbf{B} sont deux points de la scène par lesquels passe la droite D , alors la matrice de Plücker de D est donnée par :

$$\mathbf{L}_D = \mathbf{AB}^t - \mathbf{BA}^t \quad (3.23)$$

et les paramètres de la droite 2D d_g dans l'image de gauche correspondent au noyau de la matrice $\mathbf{M}_g \mathbf{L}_D \mathbf{M}_g^t$ où \mathbf{M}_g est la matrice de projection perspective associée à la caméra gauche. Donc nous avons :

$$d_g = \ker(\mathbf{M}_g \mathbf{L}_D \mathbf{M}_g^t). \quad (3.24)$$

La figure 3.2 montre deux images de la scène de synthèse avec quelques droites. Dans un premier

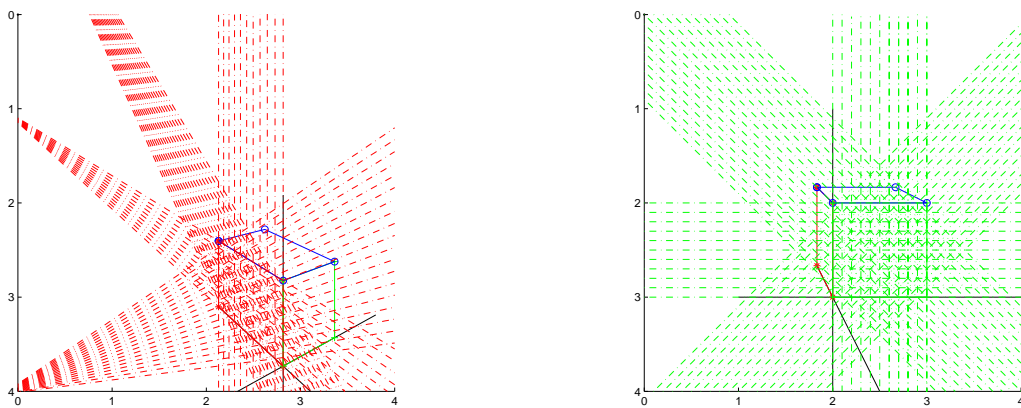
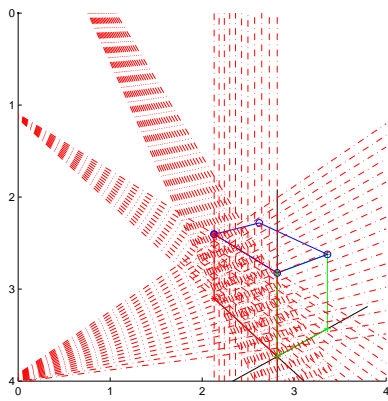


FIG. 3.2 – Images de gauche et de droite avec respectivement les droites des plans 1 (rouge) et 2 (vert).

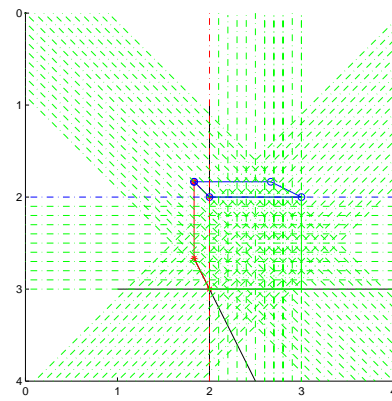
temps, les droites ne sont pas bruitées et les résultats obtenus sont corrects à près de 100%. Seules deux droites du plan 3 (sur 60) ont été mal classées. La figure 3.3 montre la classification des droites dans les images. On remarque que dans les figures des plans 2 et 3, certaines droites ne sont pas de la bonne classe. Lorsque cela se produit pour des droites se trouvant à la frontière entre deux plans, il ne s'agit pas d'une erreur. Néanmoins il reste dans la figure du plan 3 deux droites qui ont été affectées par erreur au plan 2.

Pour vérifier que l'algorithme peut être utilisé avec des images réelles, il faut étudier son comportement face à des données non parfaites. Malheureusement, perturber une droite n'est pas aussi trivial que de brüiter un point. Nous nous contenterons donc d'ajouter du bruit au paramètre de la droite correspondant à la distance à l'origine. Cette méthode n'est certes pas idéale car il aurait aussi fallu brüiter l'orientation de la droite, mais elle permet tout de même de tester l'algorithme sur des données non parfaites. Les figures 3.4 et 3.5 illustrent les résultats de la segmentation en appliquant aux droites dans les deux images des translations aléatoires, d'amplitude comprise entre 0 et respectivement 0,001 et 0,01. L'ordre de grandeur des décalages équivaut à des déplacements de 0,2 et 2 pixels si le cube était centré sur des images de 400 pixels.

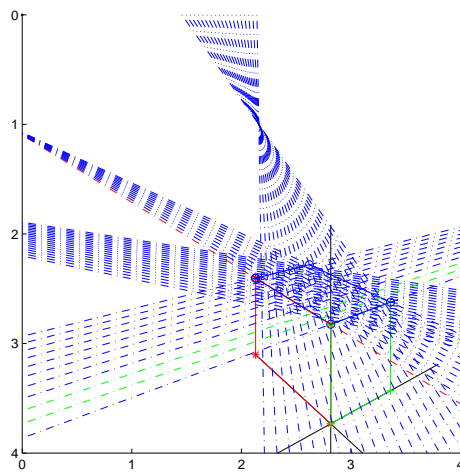
Nous pouvons voir que la qualité de la segmentation se dégrade très nettement dès que les données sont légèrement bruitées. La figure 3.4 montre que seul un des trois plans a été correctement détecté. Moins de la moitié des droites du plan 2 lui ont été affectées. Pire encore, toutes les droites du plan 3



(a) Plan 1.



(b) Plan 2.



(c) Plan 3.

FIG. 3.3 – (a) et (b) Les droites des plans 1 et 2 sont toutes correctement identifiées. (c) Nous remarquons quelques erreurs dans les droites du plan 3.

sont mal classées. Les résultats présentés dans la seconde figure sont sensiblement du même niveau. En effet, 46% des données sont correctement classées dans le premier cas et 43% dans le second cas.

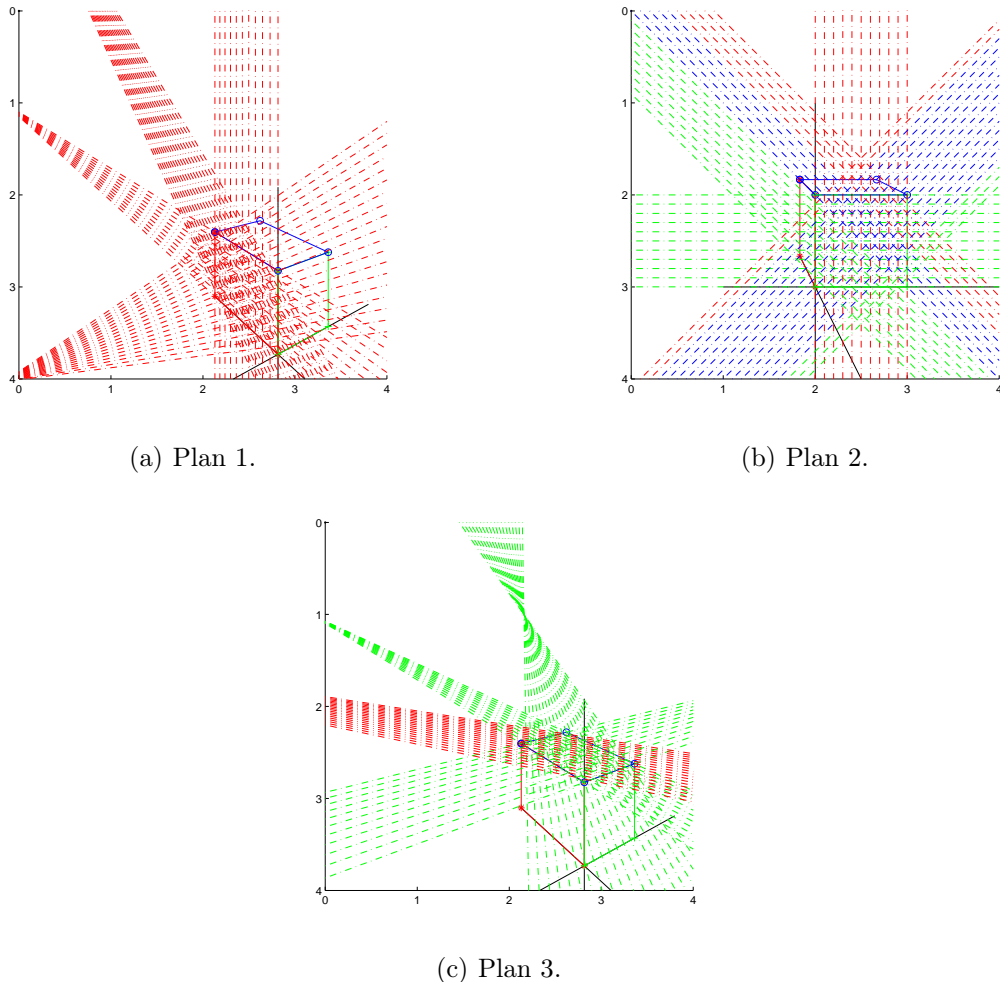


FIG. 3.4 – Résultat de la segmentation, bruit = 0,2 pixels : (a) 100% des droites du plan 1 sont correctement classées. (b) 42% pour le plan 2. (c) 0% pour le plan 3.

3.3.1.2 Points

Nous avons vu précédemment que la méthode présentée pouvait s'adapter indifféremment aux points et aux droites. Nous en avons donc profité pour étendre les tests aux points. Les points présentent l'avantage face aux droites de pouvoir être facilement bruités en perturbant aléatoirement et indépendamment leurs coordonnées. Un autre avantage des points est qu'ils peuvent être normalisés afin de rendre la méthode insensible à l'échelle des images utilisées. Pour cela nous avons utilisé la normalisation de Hartley qui consiste à centrer le nuage de points à l'origine et à appliquer un facteur d'échelle pour que la distance moyenne des points à l'origine soit égale à $\sqrt{2}$. Nous utilisons la même scène contenant le cube et prenons au hasard 360 points répartis de manière homogène sur les trois faces visibles. Les images obtenues sont présentées dans la figure 3.6.

Comme pour les droites, la phase de segmentation, pour le cas de données non bruitées, donne de très bon résultats. Nous avons ensuite voulu étudier le comportement de l'algorithme sur des

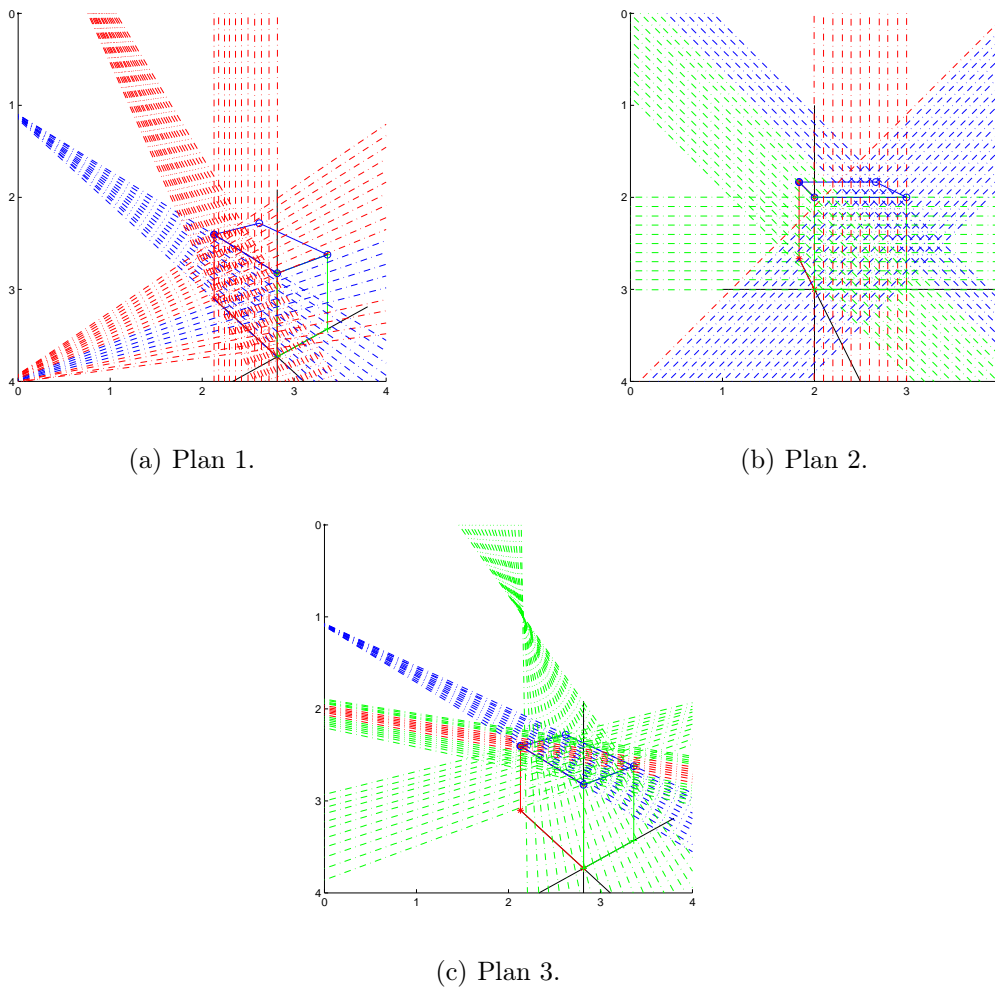


FIG. 3.5 – Résultat de la segmentation, bruit = 2 pixels : (a) 73% des droites du plan 1 sont correctement classées. (b) 40% pour le plan 2. (c) 20% pour le plan 3.

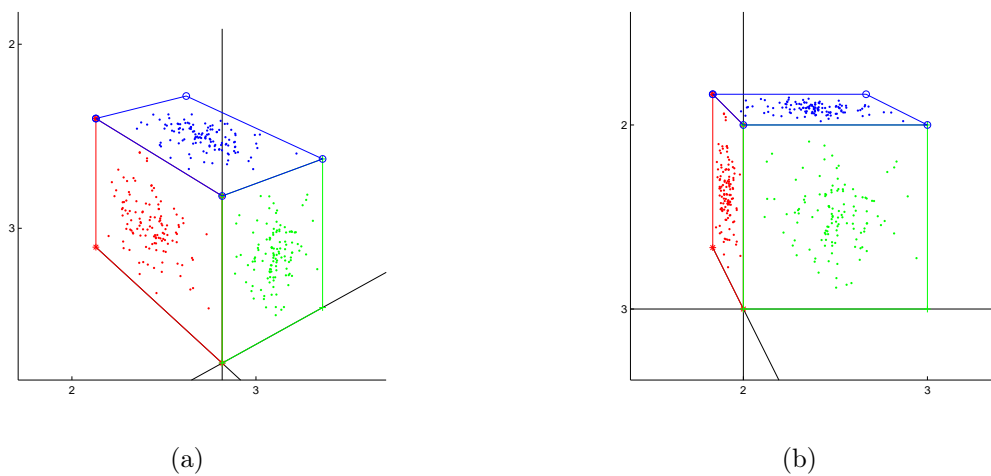


FIG. 3.6 – Classification des points sur le cube. Images de gauche (a) et de droites (b)

données non parfaites. Pour cela nous avons appliqué un bruit blanc gaussien aux points de l'image de droite. Les résultats obtenus correspondent à ce qui a été observé avec les droites, c'est-à-dire que la qualité de la segmentation décroît très rapidement tandis que le bruit augmente. La figure 3.7 présente les différentes répartitions des points dans les classes en fonction du niveau de bruit. Le graphique de la figure 3.8 rassemble les résultats obtenus pour illustrer l'évolution des erreurs de segmentation. Ce graphique indique que lorsqu'on dépasse un certain niveau de bruit, la méthode ne vaut guère mieux qu'une classification aléatoire.

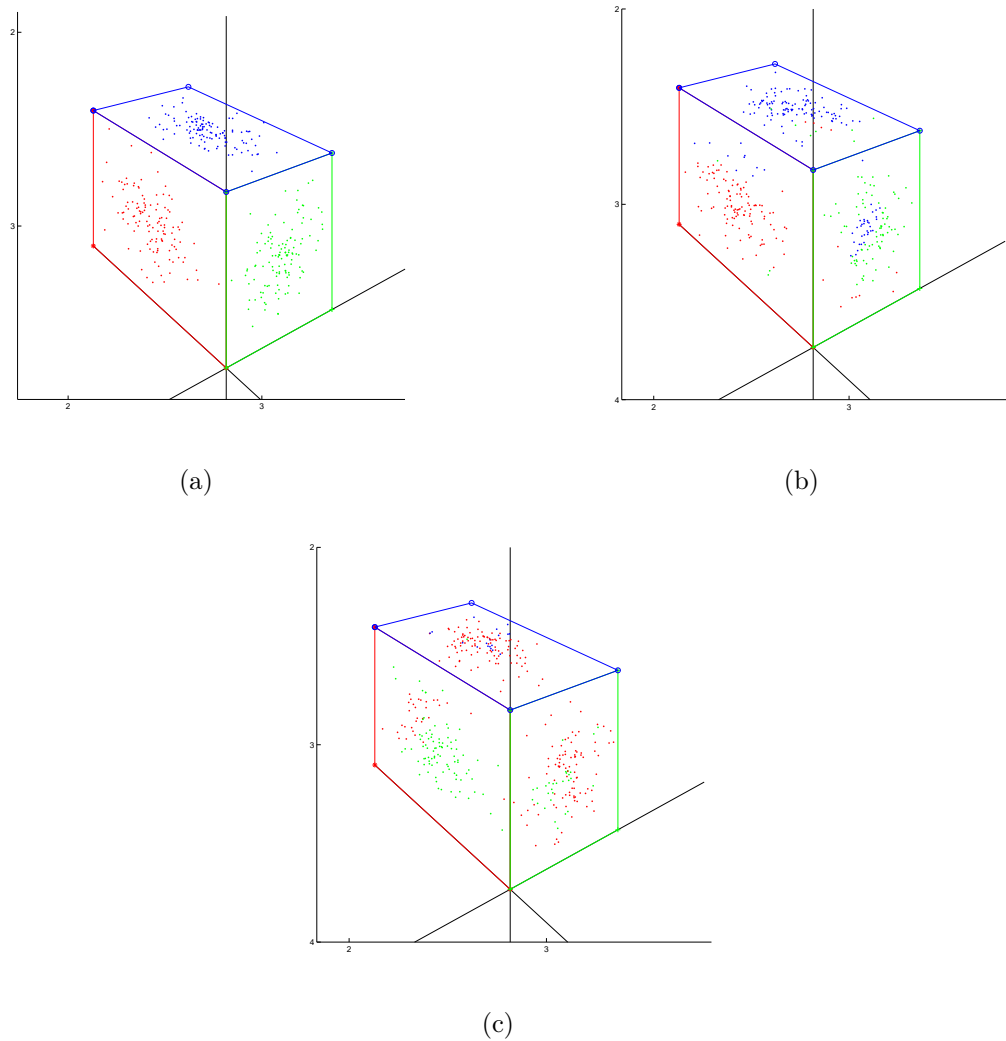


FIG. 3.7 – Classification des points sur le cube en fonction du niveau de bruit. (a) : données parfaites ; (b) bruit moyen de 1 pixel ; (c) bruit moyen de 3 pixels. Le niveau de bruit est donné pour les images ci-dessus agrandies à une taille de 400×400 pixels.

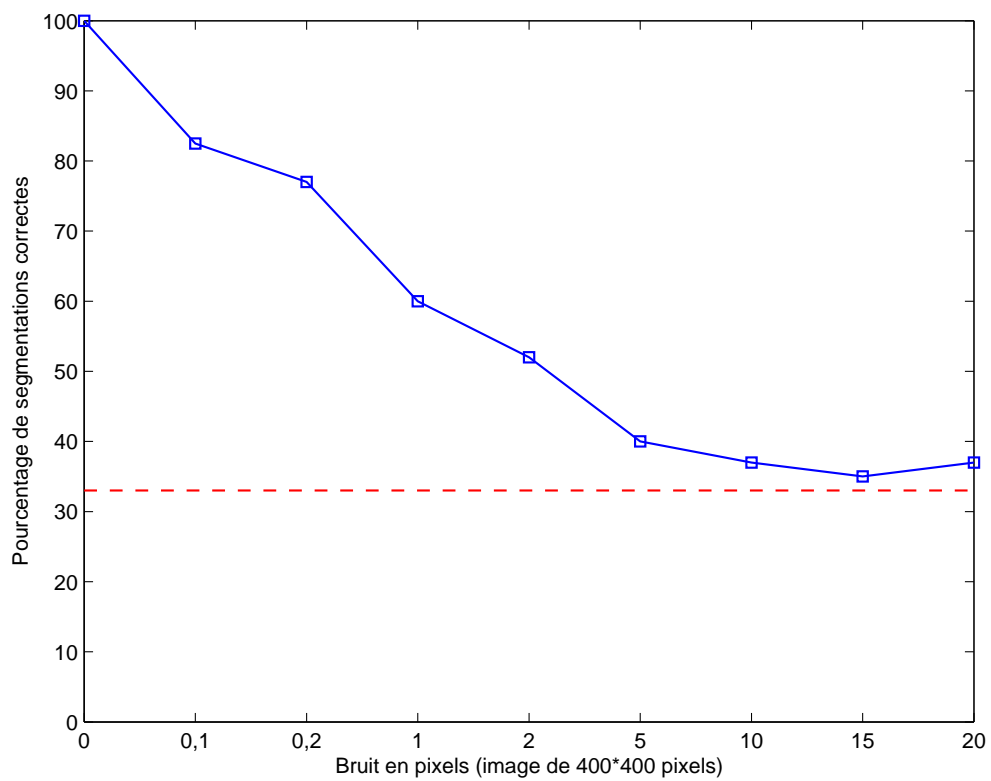


FIG. 3.8 – *Pourcentage de points correctement classés en fonction du niveau de bruit. Les valeurs sont des moyennes sur 20 tests pour chaque niveau de bruit. La ligne horizontale indique le score d'une classification aléatoire uniforme.*

3.3.2 Données réelles

Le seul moyen vraiment pertinent de tester l'efficacité d'un algorithme de vision par ordinateur est de le confronter à des données réelles. C'est ce que nous avons fait. Pour nos tests, les droites ont été identifiées et mises en correspondance dans les images à la main. Nous pouvons supposer que les erreurs commises ne dépassent pas deux ou trois pixels bien que cela soit difficile à évaluer puisqu'il faut deux points pour obtenir une droite et que, plus ces points sont distants, moins l'erreur est importante. La figure 3.9 présente les deux photos qui nous serviront pour tester la méthode.

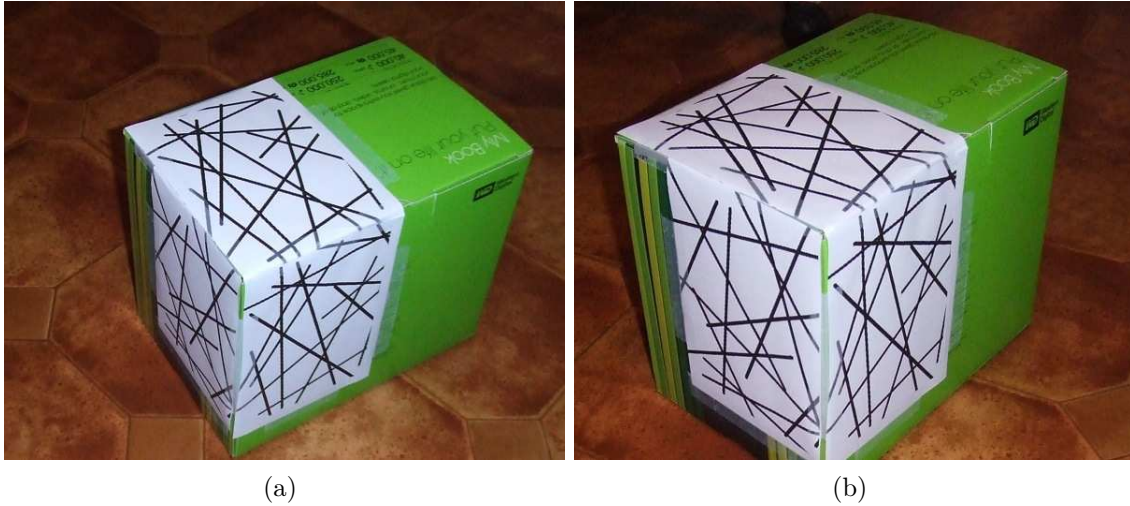


FIG. 3.9 – Images réelles gauche (a) et droite (b). Les deux images ont une taille de 600×490 pixels.

La qualité des résultats de l'algorithme ressemble très fortement à celle observée pour les données de synthèse bruitées. L'image 3.10 présente la classification des droites dans les différents plans. Seules 20 droites sur 42 ont correctement été classées, soit un taux de bons résultats de moins de 50%.

3.3.3 Nombre de plans

Jusqu'ici nous avons toujours supposé que le nombre de plans était correctement déterminé. Néanmoins, nous allons voir que la méthode présentée dans la section 3.2.2 ne permet pas toujours de trouver la valeur correcte. Le graphique de la figure 3.11 illustre la fiabilité du calcul du nombre de plans en fonction du niveau de bruit et du nombre de plans effectivement présents dans la scène. On peut même voir que dans le cas d'une scène sans bruit contenant trois plans, le calcul donne un résultat faux près d'une fois sur deux. Reprenons l'équation du calcul du nombre de plans.

$$n = \operatorname{argmin}_m \left\{ \frac{\sigma_{\binom{m+2-1}{m} \cdot \binom{m+3-1}{m}}^2(\mathcal{L}_m)}{\sum_{k=1}^{\binom{m+2-1}{m} \cdot \binom{m+3-1}{m} - 1} \sigma_k^2(\mathcal{L}_m)} + \kappa \binom{m+2-1}{m} \cdot \binom{m+3-1}{m} \right\}, \quad (3.25)$$

Rappelons que la dernière valeur singulière de \mathcal{L}_m (au numérateur) doit être égale à 0, ou être très proche dans le cas de données bruitées, si m correspond au nombre de plans n de la scène. Or dans le cas où il n'y a pas de bruit, cette valeur est proche de 0 mais inférieure à n (typiquement pour $m = n - 1$) et le membre de droite de l'addition, qui pénalise les grandes valeurs de m , incite à choisir une valeur trop petite. C'est ce qui explique les mauvais résultats du calcul pour $n = 3$ lorsqu'il y a peu de bruit. À l'inverse, lorsque les données sont moins bonnes, la dernière valeur propre de \mathcal{L}_m



FIG. 3.10 – Résultat de la classification des droites présenté sur l'image de droite. Les codes de couleurs sont les mêmes que pour les images de synthèse.

s'éloigne de 0 et le membre de droite ne suffit plus à pénaliser suffisamment les grandes valeurs de m . C'est à mon avis la cause des erreurs du calcul pour $n = 2$ lorsqu'il y a d'avantage de bruit. Ce calcul est donc assez peu fiable mais malheureusement nous nous en sommes rendus compte trop tard pour réellement nous pencher sur le problème.

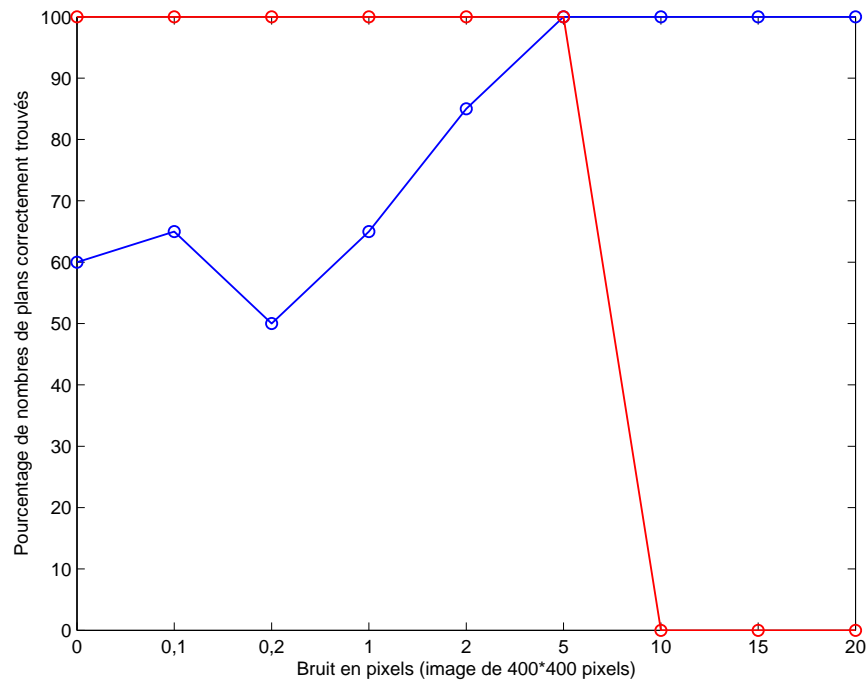


FIG. 3.11 – Fiabilité, en pourcentage de réponses correctes sur 20 tests, du calcul du nombre de plans en fonction du bruit. La courbe rouge représente une scène contenant deux plans et la courbe bleu une scène avec trois plans.

3.4 Conclusion

L'analyse généralisée en composantes principales est une méthode très peu utilisée pour la segmentation de données, et pas uniquement pour les plans. Elle repose pourtant sur une théorie mathématique assez puissante qui lui permet de s'adapter à toute sorte de problèmes de classification. D'ailleurs ses auteurs, l'utilisent pour toutes sortes de problèmes (reconnaitances de visages, détection de mouvements, ...). Néanmoins, sa complexité non négligeable ainsi que son apparente faible résistance aux données bruitées sont sans doute deux explications au fait qu'elle soit si peu répandue. Le fait que les résultats obtenus ne soient pas aussi bons que ceux auxquels nous nous attendions peut avoir plusieurs origines :

- une erreur dans notre implémentation de l'algorithme présenté qui est tout de même assez complexe à programmer et d'autant plus compliqué à déboguer que les résultats sont corrects dans le cas de données parfaites ;
- un mauvais conditionnement du système à résoudre. Ce point n'est pas abordé dans la présentation de la méthode alors qu'il s'agit de quelque chose de primordial ;

- une autre raison possible de sa sensibilité au bruit est que la méthode se base sur des polynômes d'assez hauts degrés (suivant le nombre de sous-espaces) et le fait de multiplier les données entre elles ne peut qu'accentuer les erreurs commises lors de l'acquisition.

Une nouvelle approche : la segmentation en utilisant les frontières

4.1 Introduction

Ce chapitre présente une méthode, nouvelle à notre connaissance, de segmentation en plan ; c'est-à-dire classer l'ensemble des données dans des classes représentant les différents plans présents dans la scène. La méthode présentée propose de segmenter des couples de points, mis en correspondance entre deux images, par estimation simultanée des homographies inter-images induites par les plans. Nous ferons l'hypothèse que les points d'un même plan se projettent toujours du même côté des droites projections des intersections des plans considéré avec les autres plans de la scène. Cela revient à dire que les points d'une image correspondants à un même plan se trouvent dans des espaces qu'il est facile de délimiter à partir des paramètres des plans.

Cette hypothèse assez forte mais pourtant souvent vérifiée dans le cas de scènes urbaines où les surfaces planes sont relativement nombreuses (façades, routes, trottoirs, ...). De plus, ces plans se coupent souvent en une droite présente physiquement dans la scène (un coin de mur par exemple). Cela nous semblait donc une piste intéressante à suivre dans le cadre du projet TSIGANES.

4.2 Présentation de la méthode

4.2.1 Projection de l'intersection de deux plans de la scène dans les images

Supposons, comme illustré dans la figure 4.1, que nous avons une scène contenant deux plans, \mathbf{A} et \mathbf{B} , qui s'intersectent en une droite \mathbf{L} . Cette droite se projette dans les images gauche et droite respectivement en \mathbf{l}_g et \mathbf{l}_d . Il est possible de calculer les paramètres de ces deux droites à partir des matrices d'homographies associées aux plans. La méthode présentée dans [34] permet de retrouver ces paramètres. En effet, il y est démontré que :

$$\mathbf{H}_A \mathbf{H}_B^{-1} = \mathbf{I} + \mathbf{e} \mathbf{l}_g^t \quad (4.1)$$

où \mathbf{e} est l'épipoles gauche et \mathbf{l}_g est le vecteur de la droite projection dans l'image gauche de l'intersection des deux plans. Toute matrice de la forme (4.1) est une matrice qui a la propriété d'avoir une valeur propre simple λ_1 et une valeur propre double $\lambda_2 = \lambda_3$. Notons \mathbf{v}_k le vecteur propre associé à la valeur propre λ_k , $k \in [1..3]$. D'une part il est montré que :

$$\mathbf{v}_1 \sim \mathbf{e}.$$

D'autre part, le sous-espace propre associé à $\lambda_2 = \lambda_3$ est de dimension deux, et on a :

$$\mathbf{l}_g \sim \mathbf{v}_2 \wedge \mathbf{v}_3.$$

4.2.2 Segmentation des données

Si nous disposons des paramètres de la droite formant la frontière entre les deux zones correspondant aux projections des surfaces planes, il est alors facile d'affecter les couples de points au plan qui leur correspond. Il suffit de regarder de quel "côté" de la droite ils se trouvent. Dans notre algorithme, cette première classification sert de point de départ à une méthode itérative d'affinage des paramètres des classes.

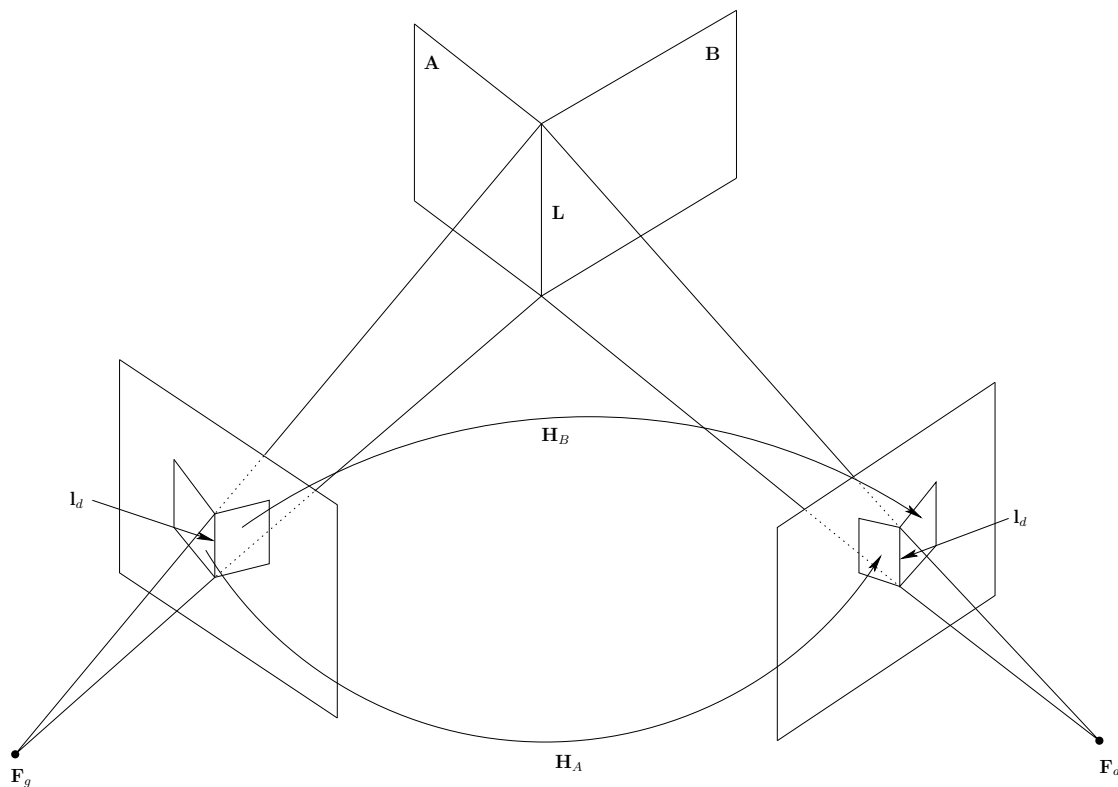


FIG. 4.1 – Exemple de projection de l'intersection entre deux plans.

4.3 Présentation de l'algorithme

Après cette présentation rapide des deux principaux aspects de notre méthode, nous allons expliquer point par point son déroulement.

Pré-requis. La méthode requiert d'avoir préalablement détecté et mis en correspondance des couples de points d'intérêt. Étant donné que nous proposons de trouver toutes les classes de points induites par les homographies en même temps, il est nécessaire de connaître à l'avance le nombre de plans présents dans la scène.

4.3.1 Calcul de la matrice fondamentale

Le calcul de la matrice fondamentale \mathbf{F} a plusieurs utilités dans notre méthode. Dans un premier temps, en utilisant une méthode robuste d’estimation de \mathbf{F} , nous pouvons écarter les données aberrantes issue de la phase de détection et de mise en correspondance de points d’intérêt. D’autre part, \mathbf{F} nous servira plus loin pour tester la validité des matrices des homographies estimées. Nous verrons de quelle manière au paragraphe 4.3.2.

Nous avons utilisé, pour le calcul de la matrice fondamentale, une fonction Matlab écrite par Peter Kovesi¹ ; ce choix a été motivé pour les raisons suivantes :

- elle utilise une méthode robuste, basée sur RANSAC, et offre donc un meilleur résultat qu’une méthode de calcul directe (moindres carrés totaux par exemple) ;
- elle effectue une normalisation telle que préconisée dans [16, chapitre 4, page 107] des points avant d’effectuer les calculs ;
- elle permet de spécifier le niveau de bruit toléré pour la détection de données aberrantes.

Le dernier point est important puisque cela nous permettra, dans l’étape décrite au paragraphe 4.3.3, d’écarter les points n’appartenant à aucun plan. De plus le niveau de bruit étant donné pour les points normalisés, cela permet de faire abstraction de la taille des images.

4.3.2 Recherche des homographies

Il s’agit là du cœur de notre algorithme. Il est inspiré de la méthode RANSAC puisque qu’il opère par tirages aléatoires successifs de points suivis de la vérification du modèle engendré par ces points.

Tirage au sort de points. Avant de pouvoir estimer les homographies induites par les n plans de la scène nous avons besoin d’au moins quatre points par classe. Il serait trop long de vouloir tirer au hasard directement un ensemble de $4 \times n$ points corrects ; c’est-à-dire un tirage contenant n quadruplets de couples de points appartenants à un même plan sans qu’un des plans ne soit représenté plusieurs fois. Étant donné que nous faisons l’hypothèse que les surfaces planes de la scène se projettent dans des zones délimitées dans les images par la projection des intersection, nous nous contentons de tirer n points au hasard puis de choisir les $3 \times n$ autres dans leurs voisinages respectifs. L’idée est que deux points proches ont plus de chances d’appartenir au même plan que deux points pris au hasard. Malheureusement, dans le cas de données bruitées, le fait de choisir quatre points proches les uns des autres conduit souvent à de très mauvaises estimations de l’homographie correspondante, si celle-ci existe. Pour tenter de résoudre ce problème, nous proposons de prendre plus de points que le minimum nécessaire.

La phase de sélection des points consiste donc à choisir au hasard n points puis à compléter la sélection en choisissant les k plus proches voisins de chacun de ces n points. Dans nos tests, nous avons posé $k = 4$; c’est-à-dire que nous utilisons cinq points pour estimer les homographies.

Cette étape boucle tant qu’il y a des doublons dans la sélection, cette contrainte permettant d’éviter de choisir les n points de départ trop proches les uns des autres en détectant les similarités dans leurs voisinages proches.

Calcul des homographies et vérification de la contrainte épipolaire. Une fois que nous avons les n ensembles de $k + 1$ points, nous pouvons estimer les homographies qu’ils sont supposés respecter. Les résultats obtenus ne correspondent pas forcément à de vraies homographies présentent entre les images. Nous devons maintenant vérifier leur validité. Dans un premier temps nous allons vérifier que chacune d’elles vérifie la contrainte liée à la matrice fondamentale présentée

1. <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/>

dans l'équation 1.14. Avec des données bruitées, le résultat ne sera pas égal à zéro même si la matrice testée correspond bien à une homographie ; il faut donc prévoir un seuil. Nous savons que les données que nous utilisons sont toutes correctes par rapport au seuil utilisé lors du calcul robuste de la matrice fondamentale. Nous pouvons donc réutiliser ce même seuil ici. Néanmoins comme les homographies sont relativement mal estimées, car à partir de peu de points, nous modifions ce seuil en le multipliant par deux. Ainsi chaque matrice d'homographie supposée doit vérifier la relation suivante :

$$\frac{1}{2N} \sum_{i=1}^N \left[d(\mathbf{x}_d^i, \mathbf{H}^t \mathbf{F}^t \mathbf{x}_d^i)^2 + d(\mathbf{x}_g^i, \mathbf{H}^{-t} \mathbf{F} \mathbf{x}_g^i)^2 \right] < 2\epsilon ; \quad (4.2)$$

où N est le nombre de total de données, c'est-à-dire les couples de points mis en correspondance, \mathbf{H} est la matrice d'homographie à tester, \mathbf{x}_g^i et \mathbf{x}_d^i sont les coordonnées des points du $i^{\text{ème}}$ couple, ϵ est le seuil utilisé pour le calcul de la matrice fondamentale, et $d(\mathbf{p}, \mathbf{l})$ est la distance orthogonale entre le point \mathbf{p} et la droite \mathbf{l} .

Si cette vérification échoue, l'algorithme tire de nouveaux points et recommence.

Calcul des frontières À partir des n matrices des homographie, nous pouvons calculer les $\binom{n}{2}$ droites correspondant aux projections, dans les images, des intersections des plans dans la scène en utilisant l'équation 4.1. Ainsi chaque plan supposé est caractérisé par un ensemble de points et un ensemble de frontières avec les autres plans. Une fois les paramètres des frontières connus, nous pouvons vérifier qu'elles séparent correctement les données. Pour cela nous calculons pour chaque paire de plans la positions des points par rapport à la frontière. Pour qu'une frontière soit considérée comme correcte, il faut qu'elle sépare les points de manière à ce que tous les points d'un même plan soient du même côté de la droite mais aussi que ce côté ne soit pas le même pour les deux plans. Bien entendu cette vérification est effectuée indépendamment dans les deux images. La figure 4.2 présente les trois cas qui peuvent se présenter lors de cette étape. Encore une fois, si cette vérification échoue, l'algorithme reprend au début et choisit de nouveaux points. En revanche, si cette étape se déroule sans erreur, nous considérons que les n homographies calculées correspondent à de bonnes estimations et qu'elles peuvent être utilisées pour classer toutes les données. C'est ce que nous allons faire au point suivant.

4.3.3 Segmentation et affinement des résultats

À cette étape de l'algorithme, nous disposons d'une approximation convenable des homographies induites par les plans de la scène. Il est donc possible d'affiner ces résultats en utilisant plus de données pour estimer les homographies. Pour cela nous effectuons une première classification des données à partir des frontières. Les frontières sont des droites qui permettent de diviser les images en différentes zones. Ainsi chaque couple de points est affecté à une classe si les deux points se trouvent dans la même zone, délimitée par les frontières, dans les deux images. La figure 4.3 montre un exemple de segmentation initiale assez mauvais induisant des incertitudes sur certains couples, c'est-à-dire des couples dont les deux points ne se trouvent pas dans la même zone dans leurs images respectives.

Une fois cette première segmentation effectuée, nous affinons les résultats par une méthode ordinaire de classification consistant à alterner entre estimation du modèle et segmentation des données. Cette étape se déroule ainsi :

1. estimation des n homographies à partir des différentes classes ;
2. segmentation des données en affectant à un couple de points l'homographie qui minimise son erreur de reprojection ;
3. si des points ont changé de classe alors retourner au point 1.

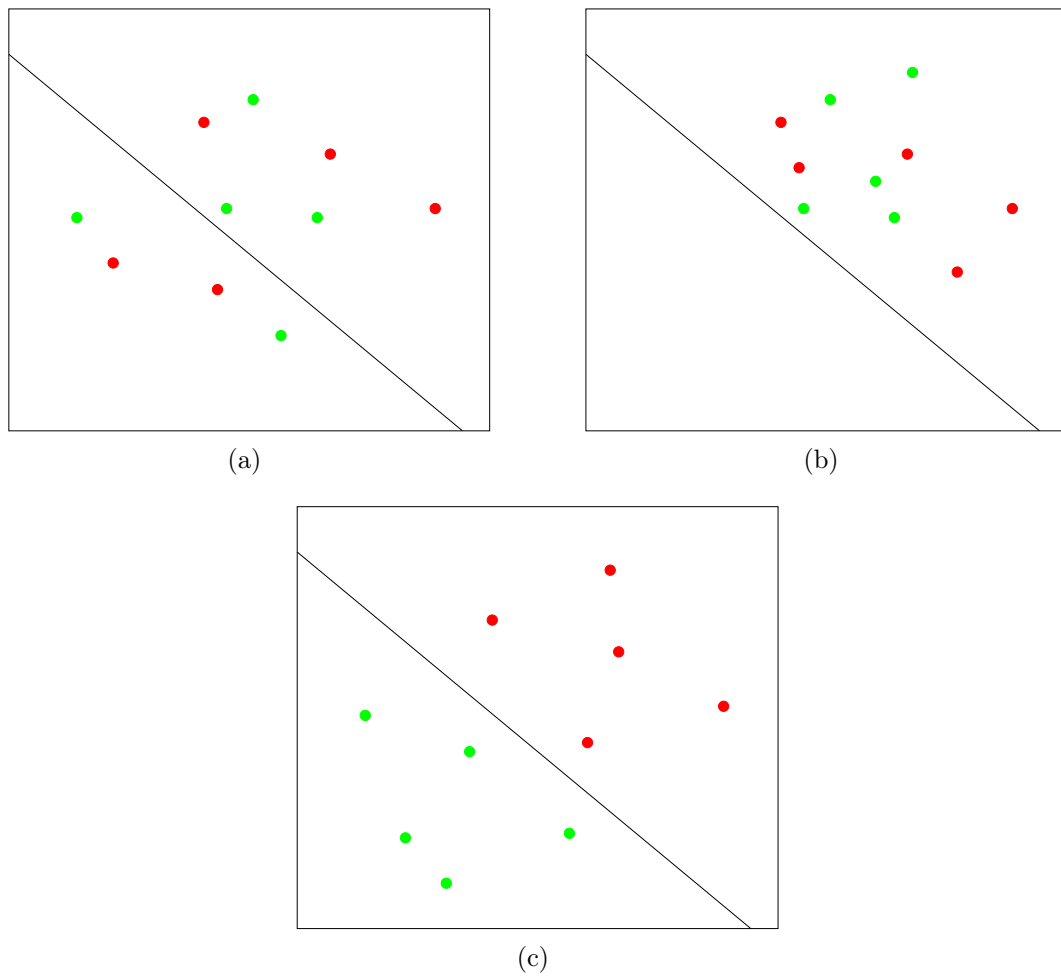


FIG. 4.2 – Différents cas possibles lors de la vérification des positions des points par rapport à la frontière : (a) les points d'un même plan ne sont pas tous du même côté ; (b) les points sont tous du même côté ; (c) positions correctes.

Il faut toutefois faire attention à ne pas tomber dans un système oscillant entre deux (ou plus) modèles ; pour cela nous avons ajouté une condition limitant le nombre d'itérations de cette boucle. Le point 2 de la boucle permet de détecter les points n'appartenant à aucun plan. Si aucune homographie ne projette un point \mathbf{p}_g sur son homologue \mathbf{p}_d avec une erreur inférieure à un seuil alors nous pouvons affirmer que le point 3D \mathbf{P} associé à ces deux points n'appartient à aucun plan de la scène. Le seuil utilisé peut encore une fois être dépendant de celui utilisé lors de l'estimation de la matrice fondamentale ; dans notre implémentation nous utilisons la même valeur.

À la fin de cette étape, nous disposons des n classes de points et des n homographies correspondantes.

4.4 Évaluations

4.4.1 Tests

Nous allons maintenant tester la méthode sur des données de synthèse puis sur des images réelles. Il s'agit des mêmes scènes que dans le chapitre précédent.

4.4.1.1 Images de synthèse

Cette fois-ci nous avons testé la méthode uniquement sur des données de synthèse de type points. Les résultats sont meilleurs que ceux obtenus avec l'analyse généralisée en composantes principales. Mais les temps de calcul sont aussi bien plus élevés et, lorsque le bruit est trop élevé (entre 2 et 4 pixels d'écart type sur une image de 400×400 pixels) l'algorithme peine à trouver un ensemble de points pouvant être utilisé pour calculer l'initialisation des matrices des homographies. Les graphes de la figure 4.4 présentent respectivement les évolutions du pourcentage de segmentations correctes et du nombre moyen de tirages nécessaires pour trouver une initialisation correcte des homographies. Contrairement à la méthode précédente, les résultats obtenus avec celle-ci ne se dégradent pas de manière progressive : soit la segmentation est juste à 100%, soit elle est complètement fautive. La figure 4.5 montre un exemple de segmentation complètement erronée.

4.4.1.2 Images réelles

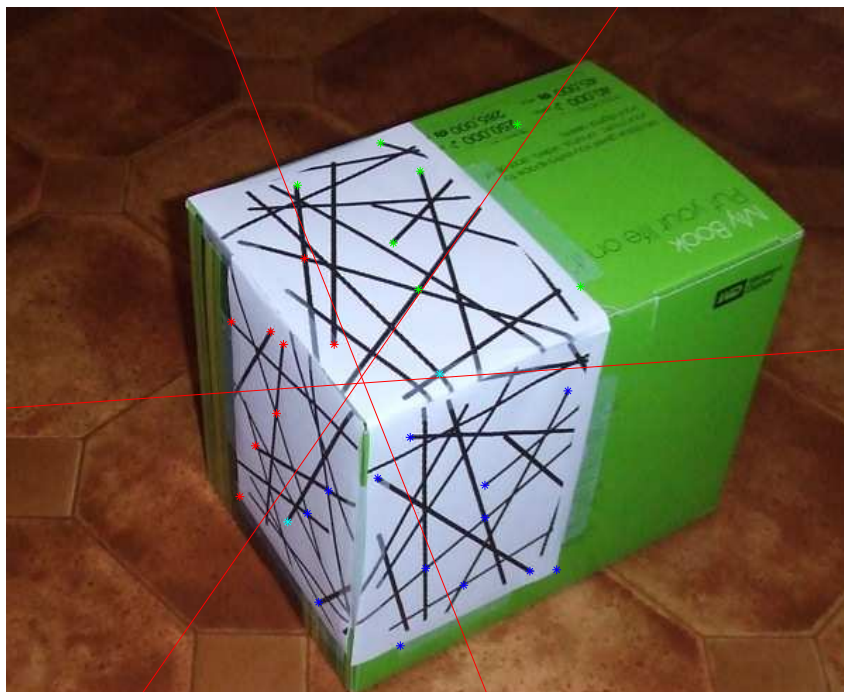
Nous avons repris le couple d'images utilisé pour l'évaluation de l'analyse généralisée en composantes principales ; mais cette fois-ci nous avons mis en correspondance, à la main, des points et non des droites. Nous avons effectué 100 tests et chacun d'eux a mené à une segmentation parfaite des données. Dans cette scène contenant trois plans et avec 30 couples de points, il faut en moyenne 81 tirages avant d'obtenir un ensemble de points satisfaisant. La figure 4.6 présente le résultat de notre méthode sur les images réelles. Nous pouvons remarquer que les frontières (droites en orange) correspondent exactement aux intersections des surfaces planes de la boîte.

4.5 Conclusion et perspectives

Les résultats obtenus sur les images de synthèse et les images réelles sont encourageants. Le fait d'obtenir une segmentation parfaite sur le couple d'images réelles, même si la localisation et la mise en correspondance des points ont été faites manuellement, permet de supposer que la méthode pourra donner de bons résultats lors d'utilisations dans un cadre entièrement automatisé. Dans le cas des images de synthèse, l'algorithme montre ses limites de résistance au bruit ; plus que la qualité des résultats, ce sont les temps de calcul qui posent problème. Comme toute méthode inspirée de



(a)



(b)

FIG. 4.3 – Résultat de la segmentation initiale. Les deux points bleu ciel correspondent aux points qui n'ont pas été affectés à un plan particulier car ils ne sont pas sur la même zone délimitée par les frontières dans l'image gauche (a) et droite (b).

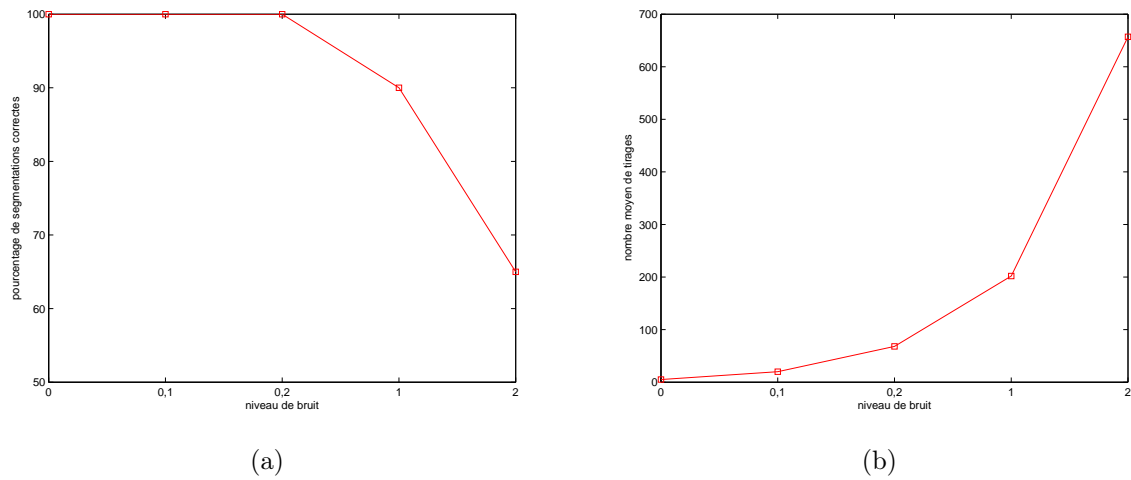


FIG. 4.4 – (a) *Pourcentage de segmentations correctes et (b) nombre moyen de tirages nécessaires, en fonction du niveau de bruit en pixels.*

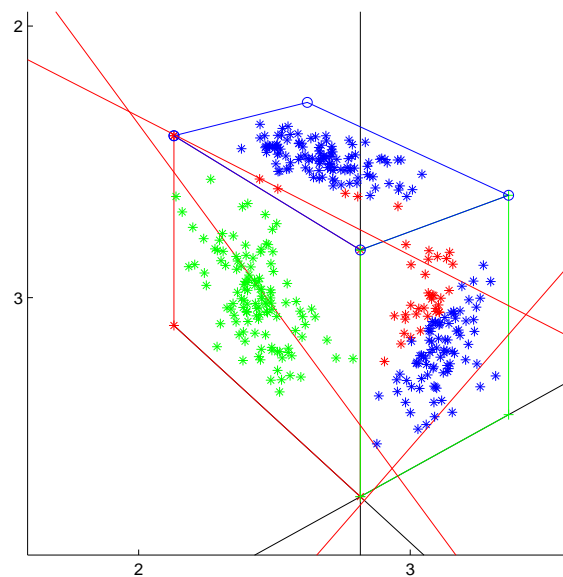


FIG. 4.5 – *Exemple de segmentation complètement erronée. Les droites rouges représentent les frontières entre les plans calculées à partir des homographies finales.*



(a)



(b)

FIG. 4.6 – Résultat final de la segmentation. Ces images sont issues de l'affinement des résultats de la figure 4.3.

RANSAC les temps de calcul peuvent devenir très importants ; il se peut même que la l’algorithme boucle à l’infini si aucun tirage ne peut satisfaire les contraintes.

Nous pensons que cette méthode peut être améliorée, par exemple en y intégrant la contrainte présentée dans [4] par Adrien Bartoli, sur la cohérence photométrique à l’intérieur des zones correspondant aux projections des surfaces planes de la scène. Cette contrainte supplémentaire pourrait nous aider à mieux détecter les zones correspondant aux frontières entre les plans.

Conclusion

Le but de ce stage était d'étudier le problème de la segmentation en plans de données issues de couples stéréoscopiques d'images. Notre travail se divise en deux parties principales : dans la première nous avons fait une étude approfondie de l'utilisation de l'une de ces méthodes, puis dans la seconde, nous avons proposé une approche nouvelle pour le cas spécifique des scènes urbaines.

Dans un premier temps, l'étude de la méthode basée sur l'analyse généralisée en composantes principales nous a permis de découvrir une technique récente, élégante mais peu intuitive car très algébrique (on travaille dans des espaces projectifs de très grandes dimensions), basée sur des outils mathématiques puissants. Malgré des résultats qui n'ont pas été tout à fait à la hauteur de nos espérances, nous pensons que cette méthode a un fort potentiel. Deux raisons possibles pour expliquer nos résultats sont :

- un problème d'implémentation : cela reste possible bien qu'il soit difficile d'expliquer pourquoi, en effet l'algorithme fonctionne parfaitement sans bruit ;
- un problème de conditionnement des données : cet aspect n'étant pas du tout discuté dans l'article, j'ai fait les choix habituels pour des algorithmes numériques.

La deuxième partie de notre travail a consisté à mettre au point une technique de segmentation en plans basée sur l'utilisation des projections des intersections des plans de la scène dans les images. La méthode que nous avons développée se base sur une contrainte qui n'a pas, à notre connaissance, été utilisée dans ce cadre ; mais qui s'adapte très bien au contexte de scènes urbaines. Nous avons manqué de temps pour permettre à cette idée de mûrir complètement mais les premiers résultats observés sont très encourageants. Il serait intéressant de voir comment cette contrainte peut être combinée avec d'autres, notamment la cohérence photométrique, et aussi d'observer comment se comporte notre algorithme face à des scènes plus complexes que celle que nous avons utilisée pour nos expérimentations. Il faudrait aussi chercher un moyen de limiter la durée de la phase de sélection des points qui peut poser problème dans le cas de données fortement bruitées ou lorsque le nombre de plans est élevé. Une idée pour ce dernier point serait, par exemple, de détecter les plans successivement par "petits groupes" de trois ou quatre plans.

Bibliographie

- [1] J. ALON et S. SCLAROFF. « Recursive Estimation of Motion and Planar Structure ». *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, 02:2550, 2000.
- [2] C. BAILLARD et A. ZISSERMAN. « Automatic Reconstruction of Piecewise Planar Models from Multiple Views ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, pages 559–565, juin 1999.
- [3] C. BAILLARD et A. ZISSERMAN. « A plane-sweep strategy for the 3D reconstruction of buildings from multiple images », 2000.
- [4] A. BARTOLI. « A Random Sampling Strategy For Piecewise Planar Scene Segmentation ». Dans *Computer Vision and Image Understanding*, volume 105, pages 42–59, janvier 2007.
- [5] B. BOCQUILLON. « Obtention de la vérité terrain pour la mise en correspondance stéréoscopique ». Rapport de stage DEA IIL, Université Paul Sabatier, Toulouse, France, juin 2004.
- [6] B. BOCQUILLON, S. CHAMBON et A. CROUZIL. « Segmentation semi-automatique en plans pour la génération de cartes denses de disparités ». Dans *actes du congrès francophone de Vision par Ordinateur, ORASIS*, page (support électronique), <http://www.lasmea.univ-bpclermont.fr>, mai 2005. LASMEA - Université de Clermont-Ferrand.
- [7] J.-Y BOUGUET. « Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm ». Jean-Yves Bouguet, 2002.
- [8] A. CROUZIL et P. GURDJOS. « *Vision par ordinateur: partie 1 vision statique* ». Master 2 professionnel IIN, Université Paul Sabatier Toulouse III, 2007 - 2008.
- [9] P. DALLE. « *Introduction à l'analyse d'image* ». Master 2 professionnel IIN, Université Paul Sabatier Toulouse III, 2007 - 2008.
- [10] R. O. DUDA et P. E. HART. « Use of the Hough transformation to detect lines and curves in pictures ». *Commun. ACM*, 15(1):11–15, 1972.
- [11] X. FAN et R. VIDAL. « The Space of Multibody Fundamental Matrices: Rank, Geometry and Projection ». Dans *WDV*, pages 1–17, 2006.
- [12] O. FAUGERAS, Q.-T. LUONG et T. PAPADOPOULOU. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene And some of Their Applications*. MIT Press, 2001.
- [13] M. A. FISCHLER et R. C. BOLLES. « Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography ». *Commun. ACM*, 24(6):381–395, 1981.
- [14] X. GANG, T. JUN-ICHI et S. HEUNG-YEUNG. « A linear algorithm for camera self-calibration, motion and structure recovery for multi-planar scenes from two perspective images ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, 2000.
- [15] C. HARRIS et M. STEPHENS. « A combined corner and edge detector ». Dans *proceedings of Alvey Vision Conference, AVC*, pages 147–151, 1988.

- [16] R. HARTLEY et A. ZISSERMAN. *Multiple View Geometry in Computer Vision*. Seconde édition. Cambridge University Press, 2003.
- [17] K. KANATANI et C. MATSUNAGA. « Estimating the Number of Independent Motions for Multibody Motion Segmentation ». Dans *proceedings of the Asian Conference on Computer Vision, ACCV*, volume 1, pages 7–12, janvier 2002.
- [18] Q. KE et T. KANADE. « A Subspace Approach to Layer Extraction ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, volume 1, page 255, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
- [19] Qifa KE et Takeo KANADE. « A Robust Subspace Approach to Layer Extraction ». Dans *IEEE Workshop on Motion and Video Computing (Motion'2002)*, décembre 2002.
- [20] K. N. KUTULAKOS. « Approximate N-View Stereo ». Dans *Conference Proceedings of European Conference on Computer Vision, ECCV*, pages 67–83. Springer-Verlag, 2000.
- [21] M. LOURAKIS, A. ARGYROS et S. ORPHANOUDAKIS. « Detecting Planes In An Uncalibrated Image Pair ». Dans *proceedings of British Machine Vision Conference, BMVC*, pages 587–596, 2002.
- [22] D. G. LOWE. « Object Recognition from Local Scale-Invariant Features ». Dans *IEEE Conference Proceedings of International Conference on Computer Vision, ICCV*, pages 1150–1157, 1999.
- [23] Y. MA, S. SOATTO, J. KOŠECKÁ et S. S. SASTRY. *An introduction to 3-D Vision, From Images to Geometric Models*, volume 26. Springer, interdisciplinary applied mathematics édition, novembre 2003.
- [24] N. OHNISHI et A. IMIYA. « Dominant plane detection from optical flow for robot navigation ». *Pattern Recogn. Lett.*, 27(9):1009–1021, 2006.
- [25] N. OHNISHI et A. IMIYA. « Model-Based Plane-Segmentation Using Optical Flow and Dominant Plane ». Dans *Computer Vision/Computer Graphics Collaboration Techniques*, volume 4418/2007 de *Lecture Notes in Computer Science*, pages 295–306, juin 2007.
- [26] K. SHINDLER. « Generalized Use of homographies For Piecewise Planar Reconstruction ». Dans *Scandinavian Conference, SCIA*, 2003.
- [27] P. F. STURM et S. J. MAYBANK. « On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, pages 432–437, 1999.
- [28] R. VIDAL. « *Generalized Principal Component Analysis (GPCA): an Algebraic Geometric Approach to Subspace Clustering and Motion Segmentation* ». PhD thesis, University of California at Berkeley, août 2003.
- [29] R. VIDAL, Y. MA et J. PIAZZI. « A New GPCA Algorithm for Clustering Subspaces by Fitting Differentiating and Dividing Polynomials ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, 2004.
- [30] R. VIDAL, Y. MA et S. SASTRY. « Generalized Principal Component Analysis (GPCA) ». Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, volume 27, décembre 2005.
- [31] T. WERNER, F. SCHAFFALITZKY et A. ZISSERMAN. « Automated Architecture Reconstruction from Close-Range Photogrammetry ». Dans *Proc. on CIPA 2001 International Symposium: Surveying and Documentation of Historic Buildings – Monuments – Sites, Traditional and Modern Methods*, septembre 2001.
- [32] M. WILCZKOWIAK, P. STURM et E. BOYER. « Using Geometric Constraints through Parallelepipeds for Calibration and 3D Modeling ». *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 27(2):194–207, 2005.

-
- [33] A. Y. YANG, S. RAO, A. WAGNER et Y. MA. « Segmentation of a Piece-Wise Planar Scene from Perspective Images ». Dans *IEEE Conference Proceedings of Computer Vision and Pattern Recognition, CVPR*, 2005.
- [34] L. ZELNIK-MANOR et M. IRANI. « Multiview Constraints on Homographies ». *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 24(2):214–223, 2002.