Towards automated, precise and validated vectorisation of disparity maps in urban satellites stereoscopy

Eric Bughin

École Normale Supérieure de Cachan

THÈSE

présentée par

Éric BUGHIN

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Spécialité : Mathématiques Appliquées

Vers une vectorisation automatique, précise et validée en stéréoscopie satellitaire en milieu urbain.

Towards automated, precise and validated vectorisation of disparity maps in urban satellite stereoscopy

---

Soutenue le 26 octobre 2011 devant le jury composé de :

Andrés ALMANSA	Telecom ParisTech	Directeur
Gwendoline BLANCHET	CNES	Examinateur
Vicent CASELLES	Universitat Pompeu Fabra	Rapporteur
Pascal MONASSE	École des Ponts ParisTech	Examinateur
Jean-Michel MOREL	École Normale Supérieure de Cachan	Président
Damien PICHARD	CS-SI	Examinateur
Marc PIERROT-DESEILLIGNY	Institut Géographique National	Rapporteur

Résumé:

Cette thèse se situe dans le cadre du projet MISS qui est une collaboration entre le CNES et différentes institutions universitaires. Le principal but de ce projet est le calcul d'une reconstruction 3D d'une scène urbaine à partir d'images satellitaires. Cette thèse se place dans la continuité du travail de Neus Sabater sur le calcul de cartes de disparités à partir de deux prises de vues différant l'une de l'autre d'un léger angle.
Notre objectif ici est de fournir une interprétation simple des cartes de disparités à travers une classification basée à la fois sur des critères bidimensionnels et tridimensionnels. Plus précisément, cette classification est faite en regroupant les points d'une carte de disparité appartenant à un même plan dans l'espace 3D.
Pour cela, plusieurs étapes ont été proposées. Dans un premier temps, un critère *a contrario* a été défini afin de pouvoir décider de manière objective quand un groupe de points peut être considéré comme planaire. Ce critère permet de plus de fixer certains des paramètres intervenant dans la plupart des méthodes de classification plane-par-morceaux pour les cartes de profondeurs. Dans un second temps, un algorithme rapide de calcul des plans basé sur une approche gloutonne est utilisé pour obtenir la classification. Les résultats obtenus montrent que notre algorithme couplé avec le critère de validation définissent une classification sensée permettant en plus de l'interprétation des données, le débruitage et l'interpolation. Enfin, dans un dernier temps, une nouvelle approche pour affiner les contours séparant deux plans de notre classification est décrite, ceci y compris dans les régions où les disparités sont inconnues.

Abstract:

This thesis is part of the MISS project which is a collaboration between the CNES (French Space Agency) and several academic institutions. The main goal of this project is the computation of 3D urban scenes from satellite images. This thesis is the continuity of the work of Neus Sabater on disparity map computation from two views with a low angle.
Our objective is to give a simple interpretation of disparity maps through a classification based on both 2D and 3D criteria. More specifically, this classification is obtained by grouping points from a disparity map when they stand on a same plane.
To achieve that, several steps are proposed. First, an *a contrario* criterion is defined in order to decide whether or not a point group should be considered as planar. This criterion also allows to set some of the parameters that are common to most piecewise-planar segmentation methods for range images. Then, a fast algorithm based on a greedy approach is used to compute the classification. The results show that our algorithm associated to our validation criterion defines a classification that allows data interpretation, denoising and interpolation. At last, a new technique is used to refine the contour separations between two planes from our classification even for regions with missing disparity data.

Remerciements

Tout d'abord, je souhaiterais remercier Andrés Almansa pour m'avoir dirigé ces 4 dernières années. J'ai beaucoup appris à ses côté aussi bien scientifiquement que humainement. Merci d'avoir été présent et disponible quand il le fallait tout en me laissant la liberté d'avancer la thèse comme je l'entendais.
Je tiens ensuite à remercier mon jury de thèse. Merci à Vicent Caselles et à Marc Pierrot-Deseilligny d'avoir bien voulu être mes rapporteurs. Merci de plus à Jean-Michel Morel, Pascal Monasse, Gwendoline Blanchet et Damien Pichard pour leur participation dans le jury.
J'aimerais remercier tous les membres du groupe MISS pour leurs présentations très enrichissantes lors des réunions. Merci donc à Bernard Rougé, Lionel Moisan, Tony Buades, Tomeu Coll, Sylvain Durand et tous ceux que je n'ai pas cités.
Je voudrais remercier toutes les personnes que j'ai eu la chance de croiser au CMLA. Merci tout d'abord aux anciens thésards. Merci à Julie, Aude et Adina, je n'aurais pas pu souhaiter meilleure compagnie pour ma dernière année (année et demie). Merci pour toutes ces pauses thés toujours animées (et peut-être un peu trop nombreuses) où beaucoup de personnes ont du entendre leurs oreilles siffler. Merci de plus à Neus pour m'avoir aidé et écouté durant 3 ans, y compris lorsque l'on a arrêté d'être voisins. Merci à toutes les 4 pour toutes ces pâtisseries. J'ai bien dû prendre 3 kilos grâce à vous. Merci à Bruno dont la présence au labo se traduisait par un déjeuner au restaurant américain. Merci de plus pour tous ces apéros à la SMAI et surtout de m'avoir bien fait penser à prendre ma raquette de tennis. J'aimerais remercier Julien pour toutes ces pauses café du matin qui m'ont permises de bien commencer la journée.
Merci à Nicolas kun, grand maître d'IPOL et de Megawave, pour avoir fait doubler le nombre de sorties au resto lorsqu'il était à Paris. Merci de plus Zhongwei, champion du monde de loup garou pour avoir bien voulu quand même jouer avec nous. Enfin merci à Rafa le Thésard le plus ancien (même s'il n'en est plus un) pour ses conseils toujours avisés. En vrac: merci à Ayman, Frédérique, Gael, Jérémie et Frédéric (le roi du dancefloor de la SMAI) et Jean-François Aujol (qui même s'il n'est pas un thésard a toujours su prendre le temps pour participer à nos pots).
Je souhaiterais de plus remercier la relève des thésards du CMLA. Merci à Ives et Yohann pour leur bon goût en matière cinématographique et culture télévisuelle. Peu de gens au CMLA peuvent affirmer avoir, comme eux, regardé la Classe Américaine en entier ou encore connaître les blagues pourries de Kad et Olivier. Peut-être qu'il y aura des CHIPS à mon pot. Merci à Momo pour avoir créer la Tradition. Merci à Morgan pour ses cheveux, Nicolas C. pour m'avoir fait perdre au Tennis. Merci à Marc qu'il m'arrive encore de croiser à DxO. Merci enfin à Benjamin, à José et à Matthieu.
Enfin, on ne peut pas quitter le CMLA sans remercier tout le secrétariat pour leur disponibilité, leur travail formidable et surtout leur bonne humeur constante (du moins avec moi...) Un grand merci donc à Véronique, Micheline, Virginie et Carine. Merci de plus à Nicholas Chriss pour le surnom ridicule qu'il a su me trouver. Enfin merci à Christophe (de l'entretien) pour toutes ces discussions avant 7h alors que le reste du monde dort.
Je tiens ensuite à remercier les personnes extérieures que j'ai pu rencontrer pendant ma thèse. Merci à Baptiste, fondateur de la Coulange Corp©. Enfin, un grand merci à toutes les personnes rencontrées durant mes années de cours à l'Ecole des Ponts ParisTech: merci donc à Pascal, Renaud M., David, Jamil, Stephane, Hiep, Nicolas, Anne-Marie, Jérôme et Renaud K.

Enfin je tiens tout particulièrement à remercier les personnes extérieures qui m'ont suivi pendant ces 4 dernières années. Merci à ma famille pour m'avoir fourni un soutien constant. Merci à mes 2 équipes de volley (Boulogne et Clamart). Merci à Julien et Clément membres actifs du GLA (peut-être que je pourrais enfin partir au ski cette année). Merci à Greg depuis Toulouse. Enfin un merci spécial pour Margaux pour m'avoir supporté pendant ces quatre années (et même plus) y compris pendant les nuit blanches ou les réveils à 6h du matin. Je ne sais pas si cela aurait été possible sans toi.

Contents

Notations

- 2D point:
$$\mathbf{x} = (x, y) \in \mathbb{R}^2$$
- 2D point homogenous coordinate:
$$\tilde{\mathbf{x}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$
- 3D point:
$$\mathbf{X} = (x, y, z(\mathbf{x})) \in \mathbb{R}^3$$
- Reference image:
$$\begin{array}{rccc} u: & \Omega \subset \mathbb{Z}^2 & \mapsto & \mathbb{R} \\ & \mathbf{x} & \mapsto & u(\mathbf{x}) \end{array}$$
- Secondary image:
$$\begin{array}{rccc} \tilde{u}: & \Omega & \mapsto & \mathbb{R} \\ & \mathbf{x} & \mapsto & \tilde{u}(\mathbf{x}) \end{array}$$
- Disparity map:
$$\begin{array}{rccc} z: & \Omega & \mapsto & \mathbb{R} \\ & \mathbf{x} & \mapsto & z(\mathbf{x}) \end{array}$$
- Homologous points in a stereo pair: $\mathbf{x}$ and $\mathbf{x}'$.
- Plane $\pi = (a, b, c)$. Equation: $z(\mathbf{x}) = a \cdot x + b \cdot y + c$.
- Image affine transform induced by plane π: $\mathcal{T}_\pi : \Omega \mapsto \Omega$
- Squared patch of size s centered at point $\mathbf{x}$: $P_s(\mathbf{x})$.
- Probability: $\mathbb{P}$.
- Expectation: $\mathbb{E}$.

Introduction

Contexte de la thèse: le projet MISS

Le projet MISS (Mathématiques de l'Imagerie Stéréoscopique Spatiale) est une collaboration entre plusieurs institutions qui fut fondé en 2007. Il rassemble le Centre National des Études Spatiales (CNES), le Centre de Mathématiques et Leurs Applications (CMLA), l'Université René Descartes (Paris V), les écoles d'ingénieurs Télécom ParisTech et des Ponts ParisTech (Imagine), l'Universitat Pompeu Fabra (UPF) et l'Universitat de les Illes Balears (UIB).

Le but principal de ce projet est la reconstruction de Modèle Numériques d'Élévation (MNE) en milieu urbain, ce, à partir de deux prises de vues décalées l'une de l'autre d'un léger angle. Cela nécessite une maitrîse complète de chaque étape de la chaîne 3D: l'acquisition des images, la calibration, la rectification, le calcul des points 3D, la modélisation 3D. Le projet MISS s'intéresse de plus a d'autres problèmes indirectement liés à cette chaîne de part l'acquisition d'images: le débruitage, la compression de données ou encore l'échantillonnage irrégulier.

Dans le cadre de sa thèse [Sabater, 2009], N. Sabater, a développé une approche permettant de réaliser une étape cruciale de la reconstruction 3D: la mise en correspondance d'une paire d'images d'une même scène. Pour cela, elle s'est basée sur les recherches de B. Rougé portant sur les conditions nécessaires à une bonne reconstruction lorsque l'angle entre les deux prises de vues est faible. Cette configuration, appelée faible B/H (où B est la distance entre les deux prises de vues ou encore base, et H est la distance des prises de vues à la scène), présente de nombreux avantages sur les configurations classiques (B/H fort). Toutefois, elle requiert aussi une précision de calculs bien supérieure pour pouvoir espérer une précision identique sur les points 3D [Delon and Rougé, 2007].
Le travail de N. Sabater a non seulement permis de se rapprocher autant possible que la théorie le permet de cette valeur de précision mais en plus de rejeter tous les points pour lesquels le calcul n'était pas certain. Le résultat de ses algorithmes est une carte décrivant les décalages à appliquer en chaque point de la première image pour pouvoir superposer les deux images. Du fait du critère de rejet des points incertains, cette carte, appelée carte de disparités, ne sera pas complètement dense. Il peut donc être utile de trouver un modèle décrivant l'ensemble des points renseignés de manière à non seulement pouvoir interpoler la carte de disparités en ses points manquant, mais aussi éventuellement de débruiter le résultat final.

D'ici fin 2011, Pléiades, un satellite d'observation de la Terre à très Haute Résolution (THR), sera lancé en orbite et permettra entre autre l'acquisition de paires stéréoscopiques dans les conditions énoncées précédemment.

Contributions de la thèse:

La recherche d'un modèle simple permettant de décrire les données 3D quelque soit leur forme, a fait l'objet de nombreux travaux depuis les 30 dernières années, en particulier en environnement urbain. Cet engouement est motivé par les nombreuses applications d'un modèle 3D plutôt que des points. Citons par exemple, le débruitage des données, la visualisation avec un meilleur rendu, la compression des données, les simulations en milieu urbain (propagation d'ondes Wifi), l'établissement simplifie des dégâts causés par des catastrophes naturelles....

Ce problème peut se résumer par la recherche de la segmentation la plus simple possible ainsi que, pour chaque région, le modèle décrivant le mieux les données. Le nombre de régions à trouver, la classification de chaque point selon l'une des régions ainsi que les paramètres des modèles utilisés pour chacune des régions sont tous des inconnues du problème, ce qui le rend très difficile à résoudre. Ceci pourrait s'écrire sous la forme d'une minimisation d'énergie, par exemple:

$$\min_{\substack{N, l(\mathbf{x}) \in \{1, \cdots, N\} \\ \theta_1, \cdots, \theta_N}} \sum_{\mathbf{x} \in \Omega} \rho(|\mathcal{T}(\theta_{l(\mathbf{x})}, \mathbf{x}) - z(\mathbf{x})|) + \lambda N \tag{1}$$

où

- N est le nombre de modèles utilisés,
- $l(\mathbf{x})$ est le label de la région utilisée pour le point $\mathbf{x}$,
- θ_i est le jeu de paramètres du modèle décrivant la région i,
- $\mathcal{T}(\theta_{l(\mathbf{x})})$ est la valeur z obtenue par projection du point $\mathbf{x}$ par le modèle correspondant,
- ρ est une fonction croissante continue de $\mathbb{R} \mapsto \mathbb{R}$,
- λ est le paramètre servant à pondérer l'attache aux données par rapport à la complexité du modèle global.

Toutefois, cette énergie fortement non convexe est quasiment impossible à minimiser avec des approches classiques. Enfin, la nécessité d'introduire différents paramètres, souvent difficiles à estimer, pour pondérer la régularité de la solution ou la complexité du modèle rend les approches variationnelles peu attrayantes pour ce genre de problèmes.

En milieu urbain, la recherche de modèles est la plupart du temps restreinte à la recherche d'un jeu de facettes planes permettant de décrire la scène. Cela est généralement motivé par plusieurs points:

- La plupart des objets en milieu urbain sont souvent plans-par-morceaux (toits, murs, sols).
- Dans les cas non plans, une description suffisamment fine par un jeu de plans donne souvent une bonne description d'une scène observée.
- De part sa simplicité, cette description peut être obtenue en un temps plus raisonnable.

La plupart des méthodes utilisées sont le plus souvent limitées à des variantes d'algorithmes connus: division et fusion, croissance de régions, transformée de Hough ou encore RANSAC.

Le point commun de toutes ces approches est qu'elles utilisent souvent le caractère local et connexe des facettes recherchées. Ceci permet de restreindre la recherche et d'éviter de nombreuses erreurs comme par exemple le mélange de plusieurs facettes planes.

Cependant, une contrainte presque nécessaire à imposer est le rejet par seuillage des points trop éloignés des différents plans recherchés. Cette étape cruciale apporte de la robustesse aux différentes méthodes car elle permet de ne pas prendre en compte des points aberrants dans le modèle final. Elle nécessite cependant l'introduction d'un nouveau paramètre peu évident à fixer (le seuil de rejet) et dont la valeur peut complètement changer le résultat.

Le but de ce mémoire est de proposer des approches permettant de segmenter les cartes de disparités obtenues par stéréovision binoculaire en régions supposées planes. Le nombre de plans à détecter, les paramètres de ces plans et l'association de chaque point de la carte à un unique plan sont les différentes inconnues du problème. A travers le manuscrit, nous garderons en tête l'objectif de limiter au maximum l'utilisation de paramètres ou, du moins, nous chercherons des moyens permettant de les fixer intuitivement. Nous nous intéresserons donc aux trois étapes suivantes:

1. Le contrôle des fausses détections de plans et l'estimation des paramètres.

2. La détection des plans dans une carte de disparités.

3. L'extension et le raffinement de la segmentation.

Contrôle des fausses détections:

Cette thèse introduit un modèle stochastique des cartes de disparités. Ce modèle *a contrario* repose sur la théorie de la Gestalt et permet l'évaluation d'une configuration de points 3D en tant qu'une facette plane. En définissant la notion de plan significatif, ce modèle assure qu'en moyenne moins d'une fausse détection de plan sera validée en présence d'une carte de disparités aléatoire. Ceci permet donc de rejeter l'explication planaire pour des groupes de points aléatoires.

Le modèle est ensuite adapté au cas de plusieurs plans ceci permettant de choisir pour un groupe de points 3D quelle est la description la plus adaptée:

- Un modèle à un seul plan, plus simple mais s'attachant moins aux données.

- Un modèle à deux plans, plus complexe mais commettant moins d'erreurs dans sa description.

Une dernière application de ce modèle, va être de décider quel est le meilleur choix de seuil de rejet des points aberrants. Ce seuil est en effet implicitement intégré au modèle *a contrario* et sa valeur peut modifier la significativité d'un plan par rapport à du bruit. Pour chaque groupe de points 3D pour lequel on teste la planarité, on pourra alors choisir le seuil de rejet le plus adapté à sa description. Ceci permet donc une sélection automatique du paramètre le plus critique utilisé dans les méthodes de détection des plans.

Détection de plans:

La recherche de la meilleure description plane-par-morceaux d'une carte de disparités sans aucune information sur le nombre de primitives planes est un problème $NP-$complet. Il est donc nécessaire de limiter la recherche par des algorithmes rapides fournissant une solution approximative. Nous proposons deux stratégies différentes d'exploration des données:

- La première est basée sur une approche division fusion. Un groupe de points 3D est divisé si une interprétation sous forme de deux plans semble plus adaptée. Une fois l'ensemble des divisions effectuées, on tente de fusionner les groupes voisins géographiquement.
- La seconde méthode, nettement plus rapide que la première, est basée sur de la croissance de régions. Pour une région donnée, les points connexes sont testés comme possiblement appartenant au même modèle plan. Le critère de validation d'un point 3D est défini par un seuillage dur sur les points aberrants.

Les décisions dans les deux méthodes proposées sont basées sur le modèle *a contrario* précédemment défini. Ceci qui permet, contrairement aux approches classiques, une sélection automatique des paramètres à utiliser.

Extension et raffinement de la segmentation

Cette thèse traite ensuite les trois problèmes suivants qui sont liés à la segmentation plane-par-morceaux des cartes de disparités:

- En stéréovision, les algorithmes de calcul des cartes de disparités ne fournissent pas nécessairement des cartes denses. En effet, dans la méthode utilisée en entrée de nos algorithmes [Sabater, 2009], un critère de rejet est défini afin de ne garder que les points dont la disparité calculée est sûre. Au final, les parties occluses ou peu texturées d'une image à l'autre seront filtrées et n'auront pas de disparité assignée.

  Toutefois, en définissant un modèle possible pour décrire les données dans ces régions, une valeur de disparité peut être définie. La question que l'on se pose est donc de savoir comment étendre aux points inconnus une classification plane-par-morceaux obtenue sur les points connus.

- Un second problème provient des algorithmes locaux de calculs de cartes de disparités qui souffrent pour la plupart d'un artefact appelé "adhérence". Cet artefact se caractérise par une dilatation de certains objets dans la carte de disparité lorsque leurs contours sont trop fortement marqués par rapport à la texture locale. On se demande donc s'il est possible de corriger l'adhérence à partir de la segmentation plane-par-morceaux d'une carte de disparités.

- Le dernier problème provient des algorithmes de segmentations eux-mêmes où l'association d'un point à l'un des modèles plans donnés se fait par un seuillage sur la distance du point au modèle. Selon ce critère, un point peut potentiellement être validé pour plusieurs modèles. Comment peut-on alors décider le modèle le plus adapté pour ce point?

Ces trois problèmes sont traités comme un seul: à partir d'une segmentation plane-par-morceaux d'une carte de disparités, quel est pour chaque paire de plans voisins le meilleur contour les séparant?

L'approche considérée est la recherche d'un contour par minimisation d'énergie. Trois cas sont alors envisagés:

- la validation de l'intersection entre les deux plans comme séparation.
- la validation du contour maximisant la photo-consistance entre les deux images de la paire stéréo. Cela correspond à trouver le modèle pour lequel la reprojection de l'image de référence sur l'autre image est la meilleure possible.
- la validation du contour le plus contrasté dans l'image entre les deux plans considérés.

Le choix entre les différentes possibilités est fait par rejet ou validation au fur et à mesure de chacune des hypothèses. L'ordre considéré est le suivant: photo-consistance, intersection puis contour contrasté. La validation de la photo-consistance est faite à partir de la définition d'un nouveau modèle *a contrario*. L'intersection est validée par des critères purement géométriques. Le contour fortement contrasté est enfin considéré comme dernier recours lorsque les deux premières approches ont échoué.

Organisation du rapport:

Le Chapitre 1 introduit tout d'abord le problème en s'attardant sur les systèmes d'acquisition 3D, tout particulièrement la stéréo binoculaire. Puis une étude bibliographique des différentes approches de segmentation 3D est fournie, principalement concentrée sur les approches proposées pour les cartes de profondeurs en milieu urbain. Dans le Chapitre 2, le modèle *a contrario* de validation des plans est défini et étudié. Puis l'algorithme de diffusion fusion est testé à partir de ce modèle. Dans le Chapitre 3, l'algorithme de croissance de régions est présenté ainsi que des résultats sur sa validité théorique. Des expériences supplémentaires sur d'autres données ainsi qu'une amélioration de l'algorithme du Chapitre 3 sont ensuite fournies dans le Chapitre 4. Dans le Chapitre 5, la méthode de raffinement des contours d'une segmentation plane-par-morceaux est expliquée. Enfin, les différentes méthodes de calculs des plans ainsi que des résultats théoriques sur la précision espérée pour certains cas sont décrits en Annexe A.

Ce travail a fait l'objet de deux publications:

- Bughin, E. and Almansa, A. (2010). Planar patch detection for disparity maps. *3DPVT'10*.
- Bughin, E., Almansa, A., Grompone Von Gioi, R. and Tendero, Y. (2010). Fast plane detection in disparity maps. *ICIP'10*.

Introduction (in English)

Context: the MISS Project

The MISS project(Mathematics for Stereoscopic Space Imagery) is a collaboration launched in 2007 including several institutions such as the French Space Agency (CNES), the Center for Mathematics and their applications (CMLA), Rene Descartes University (Paris V), Telecom ParisTech engineering school, Ponts et Chaussées ParisTech engineering school, Pompeu Fabra University (UPF) and the Balearic Island University (UIB).

The main goal of this project is the reconstruction of Digital Elevation Models (DEM) in urban area from two separate views shifted from a low angle. This requires the complete understanding of each step of the 3D reconstruction chain: image acquisition, camera calibration and rectification, 3D computation and 3D modelling. The MISS project is moreover interested in any problem related to this chain or to image acquisition: denoising, data compression or irregular sampling.

In her Ph. D. thesis [Sabater, 2009], N. Sabater proposed a new approach to a crucial step of the 3D reconstruction: making correspond two images of a same scene. Her work is based on the results of B. Rougé on the necessary conditions to a good reconstruction when the angle between the views is low. Such configuration, the so-called low B/H conditions (where B stands for the baseline or the distance between the two cameras, and H is the distance to the scene or height) shows several advantages over the classical configurations (large B/H). However, to achieve the same precision magnitude on the resulting 3D points, the low B/H conditions require a lot more precision in the computations.

Sabater obtained as precise results as the theory allowed to get and moreover proposed a statistical approach to reject points for which the 3D value is uncertain. The results of her algorithm is a map describing the shift necessary at each point of the first image (reference image) to match the second image. Due to her rejection criterion, these so-called "disparity maps" are not completely dense. A model describing the known points can then both be useful to interpolate the missing data points but also to denoise the results.

From now till the end 2011, Pléiades, a very High-resolution Earth observation satellite (VHR) will be launched and will allow the acquisition of stereoscopic pairs under those low B/H conditions.

Contributions of this thesis:

The search for a simple model describing 3D points, especially under urban conditions, has been a large source of inspiration over the last 3 decades. This has been mainly motivated by the large amount of applications of a 3D model compared to 3D points. Let's mention for instance data denoising, visualisation with a nicer rendering, simulations in urban environment such as Wifi wave propagation, estimation of recoveries after natural catastrophes.
This problem can be summarized by the search of the simplest segmentation and the model best describing the data for each of the segmented regions. The number of regions, the classification of each point to a unique region as well as the model parameters for each region are the unknown of the problem, which makes it hard to solve. This could be written as an

energy minimization problem such as:

$$\min_{\substack{N, l(\mathbf{x}) \in \{1, \cdots, N\} \\ \theta_1, \cdots, \theta_N}} \sum_{\mathbf{x} \in \Omega} \rho(|\mathcal{T}(\theta_{l(\mathbf{x})}, \mathbf{x}) - z(\mathbf{x})|) + \lambda N \tag{2}$$

where

- N is the number of models and regions that are used,
- $l(\mathbf{x})$ is the region label associated to any point $\mathbf{x} \in \Omega$,
- θ_i is the parameter set of the model describing region i,
- $\mathcal{T}(\theta_{l(\mathbf{x})})$ is the z value obtained after the projection of $\mathbf{x}$ using its corresponding model,
- ρ is an increasing and continuous function of $\mathbb{R} \mapsto \mathbb{R}$,
- λ is the parameter weighting the data fitting compared to the global model complexity.

However, this strongly non-convex energy is almost impossible to minimized using classical methods. Moreover, the necessity to introduce hard-to-estimate parameters to weight the model complexity or regularity compared to the data fitting, makes variational approaches not the best choice to solve this kind of problems.

In urban areas, this search of a 3D model is most of the time limited to the search of planar facets. This is motivated by several things:

- Most man-made objects in urban situations are piecewise-planar (rooftops, walls, ground).
- Whenever this is not the case, a fine enough piecewise-planar description still gives a good approximation of the observed scene.
- Because of its simplicity, a piecewise-planar model can be obtained with reasonable computations.

Most method to find a coherent piecewise planar description are usually adaptations of well-known algorithms: split and merge, region growing, Hough transform or RANSAC.

They all have in common to impose a local constraint by most of the time validating only connected groups. This allows to restrain the search space and avoid classic errors such as validating mixtures of several planes as a single one.

Moreover, they add another constraint which is almost necessary to impose, is the rejection by hard thresholding of the points for which the distance to the planes is too large. This crucial step brings robustness to the methods because outlier points are not taken into account in the final model estimation. This however requires the introduction of a new parameter (the rejection threshold) which value is hard to set and can completely change the final result.

The goal of this thesis is to propose approaches to obtain a piecewise-planar segmentation of a disparity map obtained from binocular stereo-vision. The number of planes, the plane parameters as well as the association of any point of the disparity map to one of those planes are all unknown. Through the whole manuscript, we keep in mind to limit as much as possible the use of parameters or at least we try to find coherent ways to set their values. We will draw our interest on the three following steps:

1. Controlling the false detection of planes and estimating the parameters.
2. Detecting a set of planes from a disparity map.
3. Refining the model and extending it to unknown points.

False detection control

In this thesis we introduce a stochastic model of disparity maps. This *a contrario* model is based on the Gestalt theory and allows the evaluation of 3D point groups as planar facets. Through the definition of meaningful planes, this model ensures that a mean of at most 1 false detection of a plane should occur with a random disparity map. This allows to reject randomly distributed point groups during the plane detection.

This model is then adapted to the multi-plane case which allows to choose the most adapted configuration for a 3D point group:

- A single plane model which is simpler but fit less to the data points.
- A two-planed model, more complex but better suited to the data.

A last application of this model is to be able to choose which outlier rejection threshold is the most adapted. This threshold is implicit to the *a contrario* model and its value modifies the meaningfulness of a plane within random points. For each 3D point group for which the planar hypothesis is tested, the most adapted threshold value is then chosen. This allows an automatic selection of the most critical parameter in classical plane detection methods.

Plane detection

The search of the best piecewise-planar explanation of a disparity map without any *a priori* knowledge on the number of planar primitives is an $NP-$complete problem. Using a fast algorithm which gives an approximative solution is absolutely necessary to limit the search of the solution. We propose two different exploration strategies:

- The first one is based on a split and merge approach. A 3D point group is divided if a two-planed model gives a better explanation than a single plane one. Once all the divisions are done, a merging step is added on every neighboring groups.
- The second method, which is a lot faster than the first one, is based on a region growing procedure. For a given region, the connected points are tested as possibly belonging to the same corresponding plane. The validation criterion of a 3D point is defined with a hard thresholding of the outlier points.

All the decisions in the two methods are based on the *a contrario* previously defined. Contrarily to classic approach, this allows an automatic selection of the parameter to be used.

Segmentation refinement and extension

This thesis then treats three problems linked to the piecewise-planar segmentation of disparity maps:

- In stereo vision, the disparity maps are not necessarily dense. Indeed, the approach that we use as input to our algorithms [Sabater, 2009] is coupled with a criterion to reject points for which the computed disparity is not sure. In the end, occluded regions or low-textured regions are filtered and are not assigned any disparity.
  However, a parametric model describing the data in those regions allows the definition of a disparity. The question that remains is how the piecewise-planar classification can be extended to the unknown points.

- Another problem related to local disparity computation algorithms is an artifact called adhesion. This artifact causes the dilatation of some objects in disparity maps whenever the contours delimiting those objects are a lot more contrasted than the neighboring local textures. Is it then possible to correct this artifact from an initial segmentation which gives information on the objects present in a disparity map?

- A last problem comes from segmentation algorithms themselves for which the association of a point to one of the models is done by hard thresholding on the distance of the point to the model. According to such criterion, how can we decide which model is the most adapted to each point of the disparity map?

All those problems are treated as a single one: given an initial piecewise-planar segmentation of a disparity map, what is the best separation between a pair of planes?

The contour is searched here using an energy minimisation approach. Three possibilities are considered here:

- The validation of the intersection of the two planes as the separation.

- The validation of the contour maximizing the photo-consistency of the two images of the stereo pair. This is equivalent to finding the model that is the most appropriate to reproject the reference image onto the second image.

- The validation of the most contrasted contour between the two planes.

The choice between the three models is done by successively considering each hypothesis and choosing the first valid one. The order considered is the following: photo-consistency, intersection and at last contrasted contour. The validation of the photo-consistency model is done by introducing a new *a contrario* model. The intersection hypothesis is validated by a purely geometric criterion. At last, the most contrasted contour is validated as a default choice when none of the two previous model has been validated.

Plan of the thesis

Chapter 1 introduce the global problem and explains how the 3D data are acquired, especially binocular stereo vision. Then a bibliographic review of the 3D segmentation methods is given with a particular focus on the methods related to urban areas. In Chapter 2 the *a contrario* validation of planes is defined and thoroughly studied. Then the split and merge algorithm is defined from this model. In Chapter 3, the region growing algorithm is explained and a mathematical justification is given. More experiments on the algorithm of Chapter 3 as well as an adaptation to other data-type are given in Chapter 4. In Chapter 5, the method for contour refinement of an initial piecewise-planar segmentation is explained. At last, the

different existing methods for plane computation as well as some theoretical results on the expected precision are given in Appendix A.

Two publications were made in the context of this thesis:

- Bughin, E. and Almansa, A. (2010). Planar patch detection for disparity maps. *3DPVT'10*.
- Bughin, E., Almansa, A., Grompone Von Gioi, R. and Tendero, Y. (2010). Fast plane detection in disparity maps. *ICIP'10*.

Chapter 1

General problem and state of the art

Contents

Résumé: Dans ce chapitre, nous introduisons les différentes difficultés pouvant apparaître lors de la modélisation 3D en milieu urbain. Dans un premier temps, nous rappellerons les différents types de données pouvant être utilisées. Puis, nous passerons en revue les différentes méthodes utilisées dans la littérature pour faire de la modélisation 3D ou de l'ingénierie inverse à partir de points 3D. Enfin, nous étudierons les approches utilisées pour affiner un modelé 3D existant et définir des contours précis entre deux primitives.

Abstract: In this chapter, we refer to the main difficulties that occur in urban 3D modeling. We first introduce the different data types that are usually used. We then review the various methods used in the literature for 3D modeling or reverse engineering of 3D data. At last, we study the different approaches used to refine an existing 3D model and define precise contours between the model primitives.

1.1 Introduction

Finding a three-dimensional model of architectural scenes has been an important and challenging problem in the computer vision and graphic communities for the last two decades. Such modeling can be useful for several reasons and applications:

- First of all, as pointed out in [Digne, 2010], a pure 3D point cloud is unexploitable as itself. Several notions and relations between points must be introduced such as point neighborhoods or surface normal at each point for a proper visualization. A 3D model gives all that by introducing a global surface where each point stands and are related to each others.
- A good model can be used to reduce the noise values in the measurements of the input 3D data points.
- A model reduces the storage of the 3D shape and allows interpolation of data points. One can think for instance of the case of thousands of points standing on same plane (a wall for instance). Stocking the plane parameters (3 parameters for the plane + the parameters defining the plane delimitations) instead of the points is a lot more economic in memory. Moreover, points that were not present in the original data points can easily be interpolated from the plane equation.
- At last, 3D models can be used for various applications such as wave reachability simulations, urban planning or disaster recovery.

However, finding a simple model to explain data is not a simple task since the number of regions, the association of each point to one of them as well as the model parameters used to describe each region are all unknown of the problem. This makes the search for an optimal solution an $NP-$complete problem.

A large amount of methods varying with the type of input data have been developed with a sole and unique goal: giving a simple description of a 3D point cloud. The most common approach is to look for a piecewise planar explanation. This is mainly justified by two points:

- most man-made objects especially buildings are often piecewise-planar,
- whenever this is not the case, a sufficiently fine piecewise-planar model gives a good approximation of the data.

This chapter is organized as follows. For a better comprehension of the problems and challenges of 3D modeling, we first give a short description of 3D point acquisition systems. Then we review the different methods that have been developed for 3D modeling. At last, we review the different techniques used for fine contour detection of 3D models.

1.2 3D Point acquisition

3D points acquisition is done in several ways which can essentially be classified into two categories: active or passive approaches.

In active acquisitions, the 3D points are deduced from an analysis of the time of flight after reflexion of a pulsed wave (radio or light waves) onto the observed scene. The most well known active acquisition systems are the LIDAR (*light detection and ranging*) and the RADAR (*Radio Detection and ranging*) which have different properties and are therefore used for different applications.

In passive acquisitions, 3D points are computed from one or several images of the observe scene. In the *shape from shading* approach, a coherent 3D description is found from a single image [Prados, 2004] [Durou, 2007]. In *stereo-photometry* [Durou and Courteille, 2007], several identical pictures of a 3D scene are taken under different (and known) light conditions. The 3D geometry is then deduced from the variations in reflection of the various objects. In a *structured light 3D-scanner* device, a structured light enlights the scene. The 3D is then deduced from a measure of the variation of the light structure which depends on the observed objects. At last, in *stereoscopy*, several pictures of the same scene are taken from different points of view. The position of each object varies from an image to the other depending on the point of view and the distance between the object and the camera. The 3D is then deduced from the variations in positions in the different images.

In urban situations, the two most adapted techniques are LIDAR acquisitions and stereoscopy since their devices are very simple and feasible under such conditions. LIDAR approaches produce very precise results but still very expensive. On the other hand, stereoscopy is rather cheap since it requires only the use of a single camera. Another advantage of stereoscopy upon LIDAR is the images that can be used as additive information to the 3D map and give a better understanding of the observed 3D scene.

In this thesis, we therefore supposed that the data were obtained by stereo-vision even if most of our work is also adaptable to LIDAR data. We now propose to explain a bit more the principle of stereo-vision since this will be used as input to our algorithms.

1.2.1 Binocular stereo-vision

The 3D reconstruction from two separate views is done in several steps: camera calibration and rectification, 2D point matching in the two views and 3D reconstruction. The final result is an image, called disparity map, representing the shifts necessary at each point to match the points of one image with the one of the other image. In what follows, we roughly explain the different steps necessary to the reconstruction.

Pin-hole camera

In computer vision, it is usually common to assume that the image was acquired using the pin-hole camera model. In this model, each 2D point in the image is obtained by projection

of a 3D point of the scene onto the image plane:

$$\begin{array}{rccc} P: & \mathbb{R}^3 & \mapsto & \mathbb{R}^2 \\ & \mathbf{X} = (x, y, z) & \mapsto & (f\frac{x}{z}, f\frac{y}{z}) \end{array} \tag{1.1}$$

Where C is supposed to be the center of the camera and f its focal length. The x and y axis are then the ones imposed by the image plane, the $z-$axis is defined as the normal vector to the image plane.

Camera calibration and rectification

Calibration is the first step of the 3D reconstruction from a camera. The objective is to find the intern and extern geometric properties of the acquisition system. With a pin-hole camera configuration, these parameters are: the focal length f, the camera center C, the pixel dimensions, the obliquity angles of the pixels and the 3 axis of the scene. For more details on the calibration techniques, we refer to [Hartley and Zisserman, 2000], [Faugeras and Luong, 2001] and [Lavest et al., 1998] which gives fundamental results on this crucial step.

In binocular stereoscopy, one compares two views of a same scene obtained with two different configurations. To be able to do so, re-project the two images needs to be reprojected onto a common 3D referential. This is the rectification step. For a given 3D point $\mathbf{X}$ in the observed scene, an epipolar plane is defined by $\mathbf{X}$, and the two centers of the cameras C and C' (see Fig. 1.1). The intersection of the epipolar plane of $\mathbf{X}$ with each image defines two lines called the epipolar lines. Each epipolar line of one image intersect with the line formed by the two camera centers at a single point called epipole. This epipole corresponds to the projection of the center of the other camera onto the image plane.

Epipolarly rectifying images consists in putting the epipoles at infinity which is equivalent to re-projecting the two image planes onto a single one. The epipolar lines are then parallel to each other (this is usually done in such a way that they are parallel to horizontal axis). In [Zhang, 1998], a review the various approaches used for epipolar rectification is given. In [Loop and Zhang, 1999], the authors propose a way to find the epipolar rectification minimizing the distortion of the two images.

The rectification is computed by finding the so called fundamental matrix F, which is a 3×3 matrix of rank 2 such that any pair of homologous points in the two images (2D points in the two images referring to a same 3D point in the scene) satisfies:

$$\tilde{\mathbf{x}} F \tilde{\mathbf{x}'} \tag{1.2}$$

where $\tilde{\mathbf{x}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ and $\tilde{\mathbf{x}'} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$ are the two homologous points in homogeneous coordinates.

From disparity to 3D point

Once the two images have been epipolarly rectified, any 3D point $\mathbf{X}$ will be projected onto a single line in the two images. The camera arrangement is then simplified to the one illustrated in Fig. 1.2. From Fig. 1.2, one can see that the 3D depth of a 2D point can be deduced from its shift in position (disparity) within the two images. From Thales theorem, one has:

$$\frac{B}{D} = \frac{H-h}{h} \sim \frac{H}{h} \text{ and } \frac{d}{D} = \frac{f}{H} \tag{1.3}$$

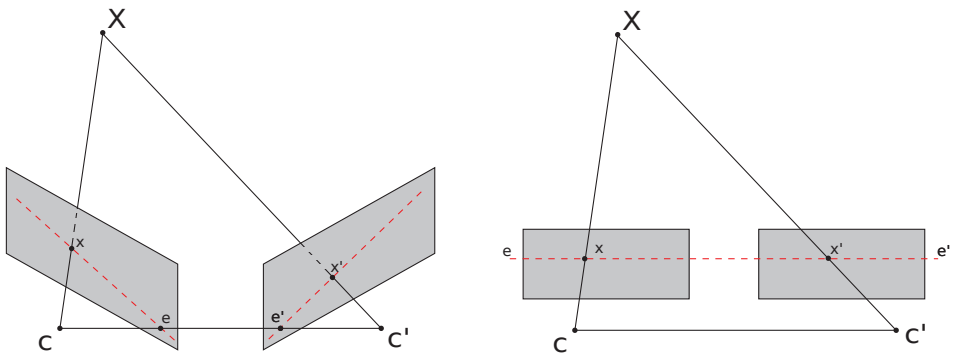


Figure 1.1: Epipolar rectification. Left: camera configuration before rectification. Right: rectified cameras. The 3D point $\mathbf{X}$ and the two camera centers C and C' define a plane. Its intersection with the two image planes defines two lines: the epipolar lines for $\mathbf{X}$ (dashed red lines). The intersection of the epipolar lines of $\mathbf{X}$ with the *baseline* (CC') (e and e') are called the epipoles. They represent the projection of the camera centers onto the other image plane.
The goal of the rectification is to find two transformations (one for each image) such that the two resulting image planes are a single one. The epipolar lines for any point $\mathbf{X}$ are then all parallel to the baseline (CC') in both images.

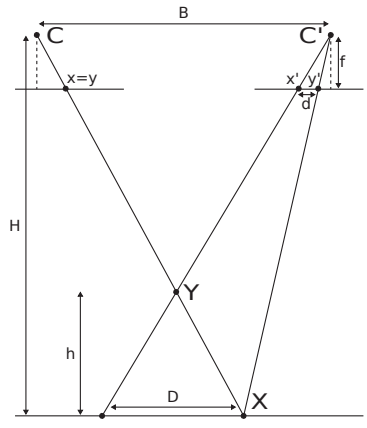


Figure 1.2: Computation of the depth of a 3D point in epipolar geometry. From Thales theorem, one can see that the height h of Y is proportional to the shift in position d.

The disparity in the two images and the height of the 3D point are then linked by:

$$d \sim \frac{Bf}{H^2} h \tag{1.4}$$

Disparity map computation

Matching points from the two images is one of the key elements to 3D reconstruction. As seen in Eq. 1.4, the translation of a given point from one image to the other is proportional to the distance of the corresponding 3D point to the image planes. We now explain how to compute the disparity map, which is the map of translations necessary to match the two images.

There are two main categories of methods to compute a disparity map: local ones and global ones. In local approaches, the disparity is computed separately at each point. In global methods all the disparities are searched at the same time by minimizing a global criterion. We will however concentrated on local methods.

The most classical approach is the block-matching. The disparity is computed by finding the translation minimizing a cost function:

$$\mu(\mathbf{x_0}) = \arg\min_{\mathbf{t}} C(\mathbf{x_0}, \mathbf{t}) \tag{1.5}$$

The cost function usually compares the gray level values for a given neighborhood in the two images. The most common cost function is certainly the Normalized Crossed Correlation (NCC) defined as:

$$C(\mathbf{x_0}, \mathbf{t}) = \frac{\sum_{\mathbf{x} \in \mathcal{N}(\mathbf{x_0})} (u_1(\mathbf{x}) - \bar{u}_1)(u_2(\mathbf{x}+\mathbf{t}) - \bar{u}_2)}{\sqrt{\sum_{\mathbf{x} \in \mathcal{N}(\mathbf{x_0})} (u_1(\mathbf{x}) - \bar{u}_1)^2 \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x_0}+\mathbf{t})} (u_2(\mathbf{x}') - \bar{u}_2)^2}} \tag{1.6}$$

where $\bar{u}_1$ (resp. $\bar{u}_2$) is the mean value of the first image in $\mathcal{N}(\mathbf{x_0})$ (resp. $\mathcal{N}(\mathbf{x_0}+\mathbf{t})$). Note that in the case of the NCC, the best candidate is the one maximizing the cost function, not minimizing.

High B/H V.S. low B/H

Reducing the distance between the two points of view (hence the B/H factor) has several advantages. First, it makes it easier to take the pictures almost simultaneously which reduce the changes (light conditions or objects) from one image to the other. Moreover, when the two views are close to each other, fewer objects are occluded from one image to the other. This allows to obtain denser disparity maps.

However, reducing the B/H also has an impact on the point precision. Differentiating Eq. 1.4, we obtain:

$$\delta h \sim \frac{H}{B} \cdot \frac{H}{f} \delta d \tag{1.7}$$

One can see that to achieve a similar precision on the 3D estimation, a more precise disparity computation is required with a lower B/H factor. To overcome this, sub-pixelic precision methods have been proposed such as the ones in [Delon and Rougé, 2007] and [Sabater, 2009]. This two methods will be the one that we use in our experiments (see Chapter 4).

The adhesion artifact (fattening)

Though probably the most popular approach, the computation of disparity maps by block matching can introduce serious errors such as the adhesion artifact (also named fattening or border errors). This artifact is due to the combination of a highly contrasted texture or edge in the image and a large local variation in the scene depth. The most common and troublesome situation appears near object delimitations which are usually characterized by a very contrasted edge in the image and a jump in the depth values of the 3D scene. In this case, the consequence is the dilatation the foreground object in computed disparity map.

The theoretical explanation to adhesion is that disparity values are supposed to be locally constant in the windows that are matched in the stereo pair while this is clearly not the case when there is a discontinuity for instance. The pixels that are likely to be affected by this dilatation are the pixels such that their distance to the discontinuity is inferior to half the size of the window used for the block-matching.

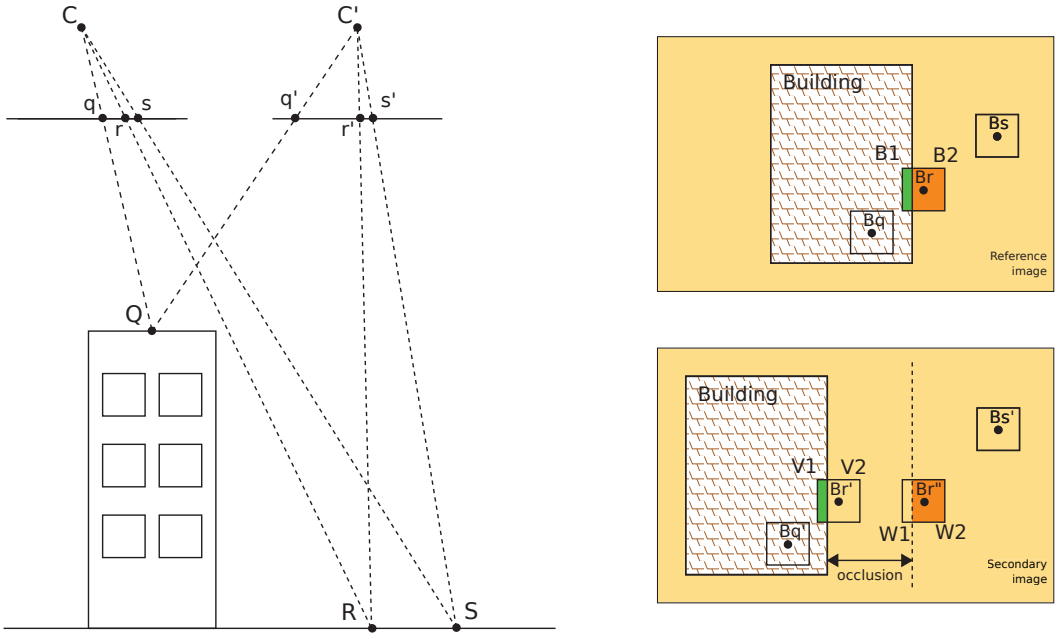


Figure 1.3: Adhesion artifact. The correct disparity to **r** is **r"** but B_r is mismatched to $B_{r'}$. This figure was inspired from a figure on adhesion in Neus Sabater's PhD's manuscript [Sabater, 2009].

Figure 1.3 represents a situation where adhesion happens. Three points are being observed here: a point on the roof of a building **Q**, and two points on the ground **R** and **S**. Their projection on the reference (resp. the secondary) image are noted q, s and r (resp. q', s' and r'). Using a block matching approach, q and s are well matched supposing that there is enough texture to match the blocks in both images. However, r is matched to r' (which gives a disparity similar to a point on the roof) instead of being matched to r''.

Indeed, in the reference image, the window around r is divided into two parts: the points on the roof (sub-window B_1) and the points on the ground (sub-window B_2). The two parts are separated by a very contrasted edge. Similarly, in the secondary image, the window around r' contains points on the roof (V_1) and points in an occluded region (V_2) and the

window around r'' contains points into the occluded region (W_1) and points on the ground (W_2). r can be either matched with r' or r''. Since the window around r' contains a very contrasted edge, the correlation between r and r' is higher than the correlation between r and r''. Therefore, r is matched with a point on the roof.

When dealing with a slanted roof (which is often the case in urban stereo-vision), the dilatation due to adhesion is observed on both sides of a roof edge (see figure 1.4).

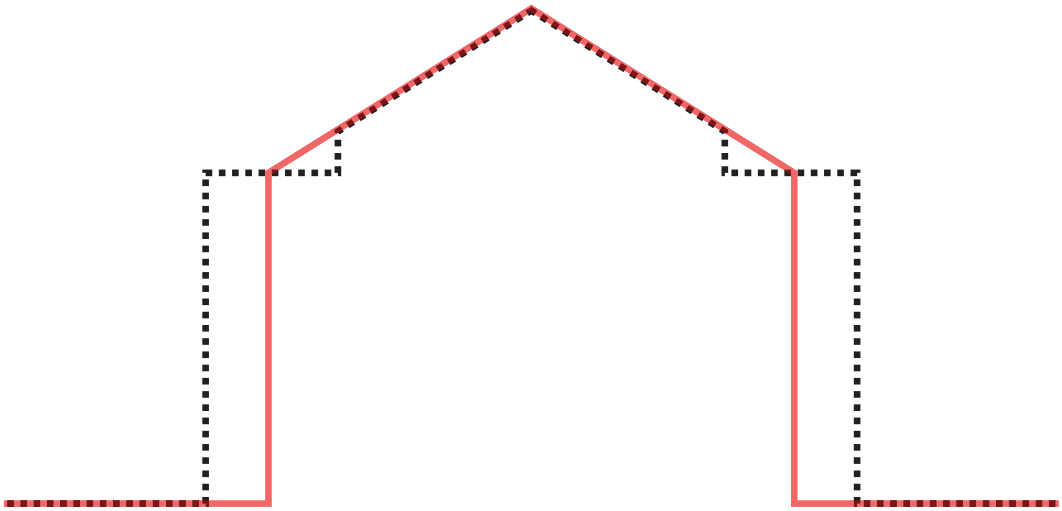


Figure 1.4: Continuous line: cut of a depth map of a house. Dotted line: estimated depth affected by adhesion. Adhesion causes the dilatation of the depth value of the roof edges.

1.2.2 Multi-view stereo-vision

With the appearance of picture databases where the same scene is sometimes observed from thousands of points of views, multi-view stereo-vision has become one of the most popular techniques to get dense 3D reconstructions.

To achieve such reconstructions, the cameras from the different views must be calibrated and rectified all at the same time. If we note N_c the number of camera configurations (points of views) and N_p the 3D points that are observed in more than one image, one can estimate both the camera parameters (internal and external) and the 3D coordinates of the points at the same time:

$$\min_{\{P_1,\ldots,P_{N_c}\},\{\mathbf{X_1},\ldots,\mathbf{X_{N_p}}\}} \sum_{i,j} v_{i,j} d(P_j(\mathbf{X_i}), \mathbf{x}_{i,j})^2 \tag{1.8}$$

where $v_{i,j}$ is a binary weight equal to 1 if the 3D point $\mathbf{X_i}$ can be seen from camera j and 0 otherwise and $\mathbf{x_{i,j}}$ is the observed corresponding 2D point in image j.

The approach that is commonly used to solve this kind of problems is the one proposed by [Lourakis and Argyros, 2009]. It uses the sparse properties of the equation system to achieve a fast resolution with a Levenberg-Marquardt algorithm.

The 3D points that are used are then the ones that have been found with Eq. 1.8. The final result is a 3D point cloud, possibly containing outliers, with no notion of neighborhood between the points.

1.3 3D modeling

Over the last two decades, a lot of methods have been developed to obtain a model describing a set of 3D points. The objective is to find a small set of primitives (usually planar facets) to describe the points. However, the problems and solutions strongly depend on the type of input data points.

1. A first set of methods was proposed to find a classification of objects from two non-rectified stereo images (unconstrained stereo-vision). Each object of the scene, is locally represented by a different projective transformation. The point here is then to determine these transformations from available feature points (usually obtained from SIFT [Lowe, 1985] or Harris [Harris and Stephens, 1988] detectors). The feature points are pairwise in the two images, usually sparse and may contain outliers. Moreover, the number of transformations that are looked for in those situations is usually limited to less than 10.

   The searched transformations are two dimensional and limited to isometries, affine transformations or homographies.

2. In presence of a range image in urban area, one usually looks for a set of planes to describe the roofs, the walls or the ground. The points here are distributed on a regular 2D grid $\Omega \subset \mathbb{R}^2$ which gives a natural notion of neighborhood. The depth at each point is assumed to be defined by a unique plane among a finite set of planes $(\pi_i)_{i=1..n}$, where $n << N_{points}$:
   $$\forall \mathbf{x} \in \Omega, \exists i, 1 \leq i \leq n / z(\mathbf{x}) = \mathcal{T}_{\pi_i}(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{1.9}$$
   where $\mathcal{T}_{\pi_i}$ is the affine transformation associated to the plane π_i and ε is an additive noise usually supposed to be Gaussian and I.I.D.. The objective is to find both the set of planes and the association to each 2D point.

   With LIDAR images, the range image is usually dense and the noise is rather low. The segmentation approaches are then only based on the 3D information of the points.

   With disparity maps obtained from binocular stereo, some points might be missing and the precision varies depending on the method that is used and the baseline between the two images. However, the two images used for the disparity computation give another source of information that may be used for the segmentation. Segmentation methods are then usually based on both 3D from the disparities and image information.

3. In the case of multi-view stereo or merged LASER scan, pure 3D points are considered (as opposed to 2.5D points for range image). The density and the precision of the points are usually high. The methods usually start by the definition of a neighborhood as well as the computation of the local normal of the surface at each point. When images are available, additional information can be used by computing 3D line segments or vanishing points.

Semi-automatic methods have been developed for building detection in range images [Nevatia and Price, 2002], [Flamanc and Maillet, 2005] and [Gruen and Wang, 1998]. However, user interaction seems poorly adapted to the case of dense building areas which tend to favor automatic approaches. The most popular techniques over the years have been split and merge approaches, region growing, clustering, model fitting from a dictionary and random procedures such as RANSAC. A good review of range image segmentation methods as well as an evaluation procedure is given in [Hoover et al., 1996].

1.3.1 Split and Merge

Originally introduced by [Horowidth and Pavlidis, 1974] for image segmentation, this algorithm is separated into two steps:

- the splitting step, where the goal is to obtain an over-segmentation of the map or point cloud,
- the merging step, where the connected regions are iteratively merged according to a criterion to refine the segmentation.

In [Boulanger and Godin, 1992], the initial partition is obtained by robust fitting of local planes and segmentation of the connected components sharing similar orientation and depth. Then a Bayesian approach is proposed to possibly merge neighboring regions. In [Parvin and Medioni, 1986] and [Xiang and Wang, 2004], the algorithm starts by assigning all the pixels to one group. Then as long as a certain figure of merit for fitting is higher than a threshold they keep on dividing. In [Taylor et al., 1989], the plane parameters are estimated in spherical coordinates using a local neighborhood. Regions are then split and merged if one of the local angle estimation deviates from the region estimation from more than a threshold or if in presence of a jump in z-values. In [Jiang and Bunke, 1994], the split and merge procedure is applied line by line and column by column to produce a very fast result. For each scan line, a first linear estimation is made. Then if the distance of the furthest point is more than a threshold, the line is split at there. The splitting continue on the two new estimations and so on until the rejection constraint is respected everywhere.

In [Taillandier et al., 2003], the authors start from an initial segmentation and use a watershed approach to merge connected regions at different scale. The result is the construction of a tree describing possible classification for each precision scale.

In [Igual et al., 2007], the authors propose an algorithm adapted to stereo-vision where images can be used as additional information to the 3D map. First, Mumford-Shah's algorithm [Mumford and Shah, 1985] is used on the reference image of the stereo pair to obtain an initial segmentation. Then an *a contrario* criterion based on a planar hypothesis is used to merge the neighboring regions. In [Facciolo and Caselles, 2009], a similar image-based approach is proposed for sparse disparity map. First, regions are obtained from the reference image by computing the grey level geodesic Voronoi cells of the known points. Then, the regions are merged with a simplified Mumford Shah's procedure with data fitting based on the range values and the estimated planes.

Remarks on the split and merge procedure

The problem of split and merge algorithms comes from the splitting step (or the first segmentation). If this initial segmentation is not precise enough, the merging step won't allow to recover the lost details.

When the initial segmentation comes from an image, since the 3D information may not be always related to the image information, some of the regions may contain outliers. This means that robust approaches must be used to estimate the plane parameters.

When the initial segmentation comes from a splitting procedure, a criterion to decide when to stop the splitting must be defined. This is usually done by introducing a threshold on the minimal precision required.

At last, when the groups are merged according to a certain order, the algorithm may become time consuming.

1.3.2 Region growing

Due to the quality of its results and its low computation time, region growing has been certainly the most popular approach for range image segmentation (see [Besl and Jain, 1988], [Poppinga et al., 2008] for instance).

The idea is very simple. First, as an initialization step, a local estimation of the plane is computed at each point of the map, usually using a least squares approach. Then seed points are selected. For each seed point, a region is created by adding the connected neighbors (on the 2D map) whenever their distance to the seed plane is less than a preset threshold and the angle between their local orientation and the plane orientation is less than another threshold. The validated points are marked and cannot be used for another group.

In [Fitzgibbon et al., 1997], a segmentation is computed by a quantification of the local curvature values. This approach is similar in some ways to the method proposed in [Digne et al., 2010] for mesh segmentation. However, in [Fitzgibbon et al., 1997], this step is used as a seed selection step for region growing. In [Pu and Vosselman, 2006], the authors used this approach for a first plane detection and added several criteria on the plane configuration to classify them as walls, ground, roofs, windows or doors. In [Ameri and Fritsch, 2000], the authors adapted this to aerial range imagery. Building parts are first detected by hard thresholding on the height values then the region growing is used to detect the roof facets. The main difference with the original approach of [Besl and Jain, 1988] is to select the seeds before region growing by testing the flatness of the neighborhood. In [Poullis and You, 2009], the authors use the region growing for large scale city modeling and set their threshold value from an estimation of the noise in the range values.

Recent works showed that the region growing procedure for plane detection could be adapted to other data-types. This is the case of [Fraundorfer et al., 2006] which proposes an approach for homography detection (thus 3D planes) in non-rectified stereo pairs. In their approach seed regions are found using the MSER algorithm to get a local homography estimation. Then the region grows from the seed region by adding the connected points whenever the gray level difference between the point and its projection on the second image is less than a threshold. The main advantage of this approach is that the segmentation of the stereo pairs is directly obtained without any computation of disparity. Another example is given in [Chauve et al., 2010] where a segmentation of pure 3D points is searched for. In this work, the growing step is done by using the k-nearest neighbors instead of using the 2D map grid such as it is done for range image segmentation approaches.

Remarks on the method

The good results of the region growing procedure are ensured by several things:

- The points within a detected group are connected.
- The hard thresholding on the distance to the planes ensures the absence of outliers within a validated group.
- This distance threshold defines the maximal error committed in the segmentation.

However, the main drawback of region growing is the constraint imposed by the choice of this distance threshold which is critical to obtain a good result and may sometimes be difficult. At last, when a point can potentially be associated to two planes (up to the threshold parameter), the greedy approach tends to favor groups that have been detected first.

1.3.3 Clustering and Hough transform

First introduced for the problem of line detection, the Hough transform [Hough, 1959] is probably one of the most natural approaches for multiple object detection. In the case of multiple plane detection, a version of the quantified space of plane parameters defines an histogram. This dual space to the 3D points is define by the set of vectors $(a, b, c) \in \mathbb{R}^3$ defining the following plane equation:

$$z = a \cdot x + b \cdot y + c \tag{1.10}$$

The algorithm works as follows. Each possible triplet of points defines a plane configuration which increments the corresponding bin in the histogram. The modes of this histogram then defines probable planes of the 3D scene (see [Flynn, 1990], [Han et al., 1987]).

As pointed out in [Toldo and Fusiello, 2008], [Rabin et al., 2009] and [Vosselman and Dijkman, 2001] several problems occur with this approach:

- Computing all the plane configurations can be very demanding.
- The choice of quantification is a compromise between precision and computation complexity.
- Some of the highest peaks in the plane parameter histogram are sometimes due to the mixture of several planes into a single one (which is not desirable).

The solution usually proposed in range image segmentation is to introduce neighborhood information of the 3D points while detecting the peaks in the histogram. In [Vosselman and Dijkman, 2001], the authors use cadastral ground planes to divide the range image into different local regions. Then the planar Hough transform is applied in each region to detect the probable planes which reduce the complexity and avoid false detection.

In [Filin and Pfeifer, 2006], a local estimation of the plane parameters is made instead of triplets of points. Then the mode seeking algorithm of [Haralick and Shapiro, 1993] is applied to detect clusters in the parameter space. The clusters made of connected points are kept and validated as planar facets.

A similar approach is the one proposed in [Peternell and Steiner, 2004]. Local planes are estimated at each point. A metric taking into account the points locality is then proposed to define the distance between two planar faces. At last clusters in the parameter space are found using this metric.

1.3.4 RANSAC

A fast alternative to the Hough transform is the RANSAC (*RANdom SAmpling Consensus*) algorithm [Fischler and Bolles, 1981] which was originally designed for fast robust regression. In the case of robust plane computation, the objective is to sort out random triplets of data points to define planes. Then for each plane, one counts the number of points whose distance to it is less than a threshold. The plane with the largest number of votes is then selected as the best one.

The number of iterations necessary to find at least one valid triplet of points is usually defined from prior knowledge on the data. If we suppose that a percentage p of the total number of points can be associated to the searched plane, then the probability of sorting out one good triplet within N_s is:

$$P = 1 - \left(1 - p^3\right)^{N_s} \tag{1.11}$$

Then supposing that one wants to ensure that the probability to get one good configuration after N_s sorts is more than p_0, we have:

$$N_s \geq \frac{\log(1-p_0)}{\log(1-p^3)} \tag{1.12}$$

Though originally designed for the robust regression of a single model, several approaches were proposed to adapt it to multiple model detection. The most intuitive and most used one is the sequential RANSAC [Vincent and Laganière, 2001] and [Kanazawa and Kawakami, 2004]. The idea is to detect a first plane with RANSAC, remove the validated points, run once again RANSAC for a new detection and so on until no points are left. However, as pointed out in [Stewart, 1995], this procedure has a good chance to detect "ghost" planes (mixture of several planes). Indeed, when one wants to detect one of the planar patches, the points standing on other planes should be considered as outliers. However, the structured distribution of these points goes against the uniform distribution hypothesis. A wrong plane then sometimes accumulates more inliers than the actual plane that should be detected.

Several solutions were proposed to adapt the RANSAC algorithm to multiple object detection in the case of unconstrained stereo-vision. One strategy is to try to find the multiple groups simultaneously [Zuliani et al., 2005]. However, this requires the knowledge of the number of groups which is an unknown parameter. An estimation of this parameter is proposed in [Zhang and Kosecká, 2006] by an analysis of the mode of the histogram of residual errors for each plane tested. Then once number of planes has been set, the groups are computed with this knowledge. However the experiments from [Toldo and Fusiello, 2008] tend to prove that all the previous methods are still not robust to real situations. The best approach seems to be the one proposed in [Rabin et al., 2009] where the authors couple the sequential RANSAC with a statistical decision criterion. This allows them to decide if a group is valid, set parameters and compare group configurations. To avoid false detection (mixture of several planes), they also try to split the final result to see if a better configuration is preferable. However, because of the considerably larger point density with range images or multi-view stereo data, their criterion and searching approach are no longer adapted since they do not take into account the local aspect of the planar facets in those situations. In those cases, the RANSAC approach proned there is indeed likely to detect mixtures of planes.

In the case of range images, a strategy to avoid "phantom" planes is to stay local when sorting out the triplets of points. In [Forlani et al., 2004], a first hard thresholding on the disparity values is done to separate buildings from the ground. Then the sequential RANSAC procedure is applied on each building which forces the locality.

In [Kada, 2006], the sequential RANSAC was used as a first step to simplify a mesh. The planes are detected by finding the mesh facets with identical orientation (up to an angle threshold) which lie on a same plane (up to a distance threshold). Such is first done to detect vertical walls. Each detected plane is used to segment the 3D space in two parts. A threshold then helps to decide whether or not a segmented part contains a building part. Then for each building part sequential RANSAC is used again to detect roof facets.

In [Labatut et al., 09] and [Schnabel et al., 2007b], sequential RANSAC is used to produce a simple mesh. As in [Kada, 2006] the difference with regular RANSAC, is that inliers are selected both from their angle and their distance to a tested plane. As an additional constraint, the validated plane is the one with the maximum number of connected inliers. The inlier connectivity is defined by the k-nearest neighbors.

Remarks on the method

The RANSAC algorithm shows the same advantages and drawbacks as the region growing approach. The threshold parameter is hard to set but is also what makes the approach robust.

In case of dense data, the connectivity constraint imposed on the final groups seems necessary. This step makes the algorithm look even more like region growing.

1.3.5 Other methods

Other approaches have been developed to find a good model description of a 3D scene or a good segmentation of range images.

In [Lin, 1998], the authors try to detect shadows of buildings to reconstruct the 3D from a single image. Using a segment detector, hypotheses are formulated on building delimitations and their shadows. The most likely configuration is selected and the 3D is deduced from the size of the shadow. The results are rather good considering that only one image is available but still not comparable to methods adapted for several images.

Other approaches are based on the detection of 3D features (points, lines) from several images. In [Bignone et al., 1996], 3D segments are computed by matching segments in several images and grouped into planes. Then 2D enclosure is searched to delimit the various planes. In [Baillard and Zisserman, 1999], 3D segments are computed from the various images. Then half planes are deduced by sweeping planes around the segment. In [Werner and Zisserman, 2002], planes are computed by translation along their supposed normal direction (obtained from vanishing points and 3D segments). At last, in [Sinha et al., 2008], 3D points, vanishing points and 3D lines are detected and filtered. Then planes are computed with a RANSAC procedure to maximize photo-consistency between images.

The main advantage of all these previous approaches is the absence of search of correspondences between images. However, they either suffer from a lack of precision on contour of planes or require a lot of images for a precise reconstruction.

At last, other approaches use a Bayesian framework to segment a range image into different object classes. In these methods, several possible model hypotheses are formulated. The Bayesian framework then quantifies the validity of a given model according to the observed data, as well as the cost of changing from one configuration to another. The final global model is then usually found using such approaches as Reversible Jump Monte Carlo Markov Chain (RJMCMC) [Green, 1995].

[Han et al., 2004] proposes to use such approach by selecting among 5 possible models (planes, conics, B-spline surface with four and nine control points, cluttered surfaces) based on their probabilities to happen knowing the data points. The probabilities are based on both range images and intensity images that are sometimes available with a LASER acquisition. In [Dick et al., 2004], the authors define parameters to qualify various part of different architectural style. First planes are searched by using 3D features (lines, 3D points) and a first classification of the planes is obtained. Then, from this MCMC algorithm is used to find the most probable model. In [Lafarge et al., 2008b] and [Lafarge et al., 2008a], the authors search for a set of oriented rectangles to describe buildings in a disparity map. Then from this segmentation, a collection of rooftop shape is defined and fitted to obtain a description of the observed scene. Both maximization steps are done using RJMCMC.

For each algorithm, the obtained results are visually good. However, the complex parameter optimization seems to be time consuming and not well adapted to complex configurations.

1.4 Segmentation refinement

The segmentation methods previously described usually associate the known points to a set of primitives. However, the final classifications may be unsatisfying for several reasons:

- Depending on the input data, some points might be missing. In unconstrained stereo-vision, only a few feature points are available. In binocular stereo-vision, some disparity computation methods reject a part of the points ([Sabater, 2009]). These points are usually either not precise enough (shadow regions) or missing in one of the two images (occluded object). In multi-view stereo-vision, some parts may be missing (no feature point or region occluded in every images).

  For all these missing points or regions, one may want to be able to extrapolate the classification. Since the models are usually parametric, the missing disparity values can then be interpolated.

- The classification methods are sometimes not given for all the points (some methods have a rejection threshold for outliers).

- The classification may be ambiguous for some of the points. As seen before, most of the classification methods are based on a rejection threshold that defines inliers and outliers. However, some points may be inliers for several groups which lead to an ambiguity. In this case, the approaches usually favor the first group detected which has no reason to be true.

- In stereo-vision some point disparities may be wrong because of computation artifacts such as adhesion. This can lead to misclassification.

- To define a 3D model, vertical planes need to be introduced which requires the introduction of clear contours in presence of a height jump.

All of this suggests the introduction of well delimited separations between the detected primitives.

In most approaches, the problem is usually not treated or is only treated by considering primitive intersections. This is usually the case when data points come from LASER range scanners and no additional information is available. We will now present the main approaches that have been used for model delimitation refinement.

1.4.1 Cadastre and rectangle fitting

A first class of approaches uses cadastral ground maps. The map delimitations are fitted to range images and used as additional information (see for instance [Vosselman and Dijkman, 2001], [Peternell and Steiner, 2004], [Durupt and Taillandier, 2006], [Flamanc and Maillet, 2005]). The idea is usually the same. Man made building facets share a lot of symmetries and parallelisms. This means that the orientations of the edges available in ground planes can be used as probable orientations for the contours of each planar facets. The other source of orientation comes from the possible intersections between planar facets.

Once a first piecewise segmentation has been obtained, the graph of adjacent planar facets is computed. In these approaches, the building has usually been delimited by the cadastre which means that only roof shapes has to be found and the quantity of facets is usually low. Then the objective is two-stepped. First, for each adjacent plane, detect if we consider either an intersection or a height jump. In the height jump case, fit a polygonal line with orientations similar to the principal directions in the cadastre.

Using cadastres will most certainly produce nice rectangular buildings with sharp edges, however, the result is rather imprecise. Moreover, cadastral maps are not always available and some more computations must be done to fit them to the range image.

Following a similar idea, in [Ortner et al., 2007] and [Lafarge et al., 2008b], propose an approach to describe buildings in dense disparity maps. Buildings are disjointed from the ground by height jumps, and are then described by a set of rectangles. The outer contour is then refined into several polygons for a better description. The inner building description, is then handled with their roof dictionary technique. However, as for the cadastral case, the produced 3D model is visually very nice but not very precise.

1.4.2 Image information

With stereo-vision data, one can use the information given by the images to better define the separations between planar facets. Indeed, under a Lambertian hypothesis, each oriented planar facet will reflect light with a different intensity. Thus, the delimitation of the planar facets in the image has a good chance to be associated to a jump in the grey level values (therefore a strong gradient value).

When adding image information to refine the range image segmentation, two strategies are generally used: use the image before and then the 3D information or the opposite.

In [Igual et al., 2007], a first segmentation is proposed using the Mumford-shah's algorithm [Mumford and Shah, 1985]. Then a merging strategy based on the 3D is used. In [Facciolo and Caselles, 2009], a sparse version of the disparity map is required. Geodesic Voronoi cells are then computed and merged according to a Mumford-Shah's strategy (using the 3D information as data fitting term). The problem of using the image segmentation before the 3D information is that if the initial segmentation is wrong, then the final one will be wrong too.

Another strategy, is to use image features as a post-processing to delimit planar facets. In [Ameri and Fritsch, 2000], image segments are mixed with intersection. The final decision is made by hard thresholding. In [Bignone et al., 1996], 3D segments are computed from the images and used to compute a set of planar facets. Then for each planar facet, the 2D enclosure is found by trying out all the possible combinations of closed contour from segments. The most likely combination is at last kept. [Vallet and Taillandier, 2005] finds the most likely 2D segments to be interpreted as a set of planar facets. Then the 3D model is adjusted to fit these segments using a Levenberg-Marquardt approach. The problem of all these methods is that a wrong model is produced if no segment is available in a region or if a wrong segment is chosen.

1.4.3 Plane adjustment

Another strategy is to refine a set of connected planar facets to find the best fitting model.

In [Taillandier and Deriche, 2004], a set of planar facets is found either by the algorithm described in [Taillandier et al., 2003], either by detection of vertical planes from the segments

and height jump. Then, 3D graph corresponding to all the possible configurations of planes is constructed and then filtered. The most likely graph configuration is at last kept using a Bayesian approach.

In [Brédif et al., 2008], start from an existing 3D polygonal description of a disparity map. Then they propose a kinetic framework to make the polygon evolve to a final solution that best fit the data points.

1.4.4 3D completion

At last, let's mention two approaches to enhance a 3D model and use it 3D completion in the case of pure 3D points. In [Chauve et al., 2010] and [Schnabel et al., 2007a], the authors both propose to use a graph-cut approach to find the best interpretation of data from an existing set of primitives. This interpretation is also used to interpolate 3D points in the most coherent way.

Chapter 2

A contrario plane detection

Contents

Résumé: Dans ce chapitre, nous proposons une nouvelle méthode sans paramètre pour détecter des patchs plans dans une carte de disparités quasi-dense. Dans un premier temps, nous introduirons un critère de décisions *a contrario* qui peut-être utilisé pour résoudre différents problèmes sur des configurations de points 3D: *(i)* cette configuration est-elle bien expliquée par un plan? *(ii)* quel est le nombre de plans optimal pour expliquer cette configuration? *(iii)* quelle valeur de seuil est la plus adaptée pour rejeter des points aberrants? Ces critères de décisions sont le coeur d'un algorithme qui recherche une explication plane-par-morceaux d'une carte de disparités lorsque cela est possible. Cette approche peut être utilisée pour la reconstruction 3D en milieu urbain, et tout particulièrement pour le cas de stéréo à faible écart entre les vues où les contraintes de précisions sont plus dures et où un choix pertinent de type et de quantité de régularité est indispensable pour l'obtention de résultats précis.

Abstract: In this chapter, we propose a new parameter-free method for detecting planar patches in quasi-dense disparity maps. We first introduce an *a contrario* decision criterion which may be used to solve various decision problems on configurations of 3D points: *(i)* is the configuration well explained by a plane?; *(ii)* what is the optimal number of planes that best explains the configuration? *(iii)* what threshold is best adapted to reject outliers? These decision criteria are the core of an algorithm that searches for an optimal explanation of a disparity map by planar patches whenever applicable. This method may be used for 3D reconstruction of urban environments, particularly in the context of low-baseline stereo where precision requirements are most strict, and a pertinent choice of the type and amount of regularization is key to achieving accurate results. It also suggests its use for automatic vectorization of urban DEMs, where a sensible geometric representation is key to achieving good visualizations.

2.1 Introduction

2.1.1 Urban 3D modeling

Modeling a 3D urban environment has been widely studied for three decades. Methods have been developed for various types of data such as unconstrained stereo-vision, two-view stereo-vision, LIDAR data, SAR interferometry, 3D data (points or meshes). Most methods are specific to the incoming data and are therefore rarely used for other data points even if some similarities can be observed.

In the case of urban modeling, it is often assumed that a scene can be well described by a piecewise-planar model. This approximation is made because first, man made objects especially buildings are often piecewise-planar, second, even when this is not the case, piecewise-planar models give a good approximation of smooth surfaces.

Small baseline stereo

Despite the potential applicability of our technique with other data sets, we concentrate here on the specific case of disparity maps obtained by photogrammetry from low-baseline stereo pairs [Delon and Rougé, 2007] (however, the method we propose here does still work in the

case of large-baseline stereo-vision). Such 3D measurement systems have a certain number of advantages:

- sure and independent punctual matches become feasible in a relatively dense area [Sabater, 2009];
- occlusions are reduced to a minimum and quasi-zenital views can be assumed.

They however introduce new challenges, since fattening artifacts become specially important, and highly subpixel-accurate disparities are required to obtain a usable accuracy in height. For this reason careful regularization techniques (like the robust affine regression we propose here) are crucial to obtaining the required accuracy level. At last, such a regularization should be applied only after verifying that the underlying data is well explained by the chosen regularization model (thus enabling the use of other regularization or interpolation techniques for non planar structures such as vegetation, domes, etc.) and level (thus avoiding over- of sub-regularization).

RANSAC and Hough transform.

Robust estimators based on RANSAC or Hough transform have been widely used in unconstrained stereo-vision.

Though the Hough transform seems pretty natural for the detection of multiple objects, its computational complexity makes it hard to use when the objects are defined by too many parameters (typically more than 2).

One of the challenges to overcome when using RANSAC is that it was originally designed to detect only one object among outliers. Several ways to cope with that were proposed. A first solution is to apply RANSAC sequentially by removing the inliers at each group validation [Vincent and Laganière, 2001; Kanazawa and Kawakami, 2004] which makes detecting a small object possible even in presence of a much bigger one. Other approaches propose to find all the objects simultaneously [Toldo and Fusiello, 2008; Zhang and Kosecká, 2006; Zuliani et al., 2005]. However, each method was proved to fail with the experiments on 2D segment and circle detection proposed in [Toldo and Fusiello, 2008]. This is mainly due to the detection of "phantom" objects made of the combination of several objects which are validated because of the high number of points attached to these objects by the RANSAC consensus.

Though [Toldo and Fusiello, 2008] looks robust in most cases, the main problem here seems to get rid of groups made only of outliers. This points out the need of a criterion to tell whether a group is valid or not.

Such a decision criterion was addressed in [Rabin et al., 2009] in the context of group matching of SIFT descriptors to find different objects in a scene. However, this technique does not scale well to quasi-dense correspondence maps, where both transformations to be detected and data points are far more numerous.

In [Schnabel et al., 2007b], [Labatut et al., 09], different adaptations of the sequential RANSAC method were proposed for the case of 3D segmentation. However, each case requires the fine tunning of several thresholds such as a point and an angle precision parameter which are critical for the result.

Geometric modeling from disparity maps.

Due to the larger number of both points and objects in disparity maps, other methods than RANSAC or Hough transform have been proposed.

In [Lafarge et al., 2008b] the authors used a dictionary of complex building models to fit the disparity map. However the applicability to the low-baseline case is less evident because the initial delimitations of buildings by rectangle-fitting to the disparity map is more error prone when the latter is noisy and affected by fattening (adhesion) artifacts. In addition, the slow convergence of the underlying non-convex optimization procedure, may scale up when the number of models in the dictionary is increased to more closely fit reality. Another version of this algorithm drops the rectangle-fitting part but has the main drawback of requiring considerable user interaction.

In [Baillard and Zisserman, 1999], the authors tried to match line segments of both images in order to find the height of a 3 dimensional edge. Then half planes are computed on each side of the segment. Despite their good results in urban areas, their method does not apply to low baseline stereo, because it relies on segment-to-segment matching, which proved to be not precise enough in this case [Sabater, 2009]. Furthermore, when segments are badly or not detected, no assumption can be made.

Various methods were proposed for the segmentation of range images [Jiang and Bunke, 1994; Taylor et al., 1989; Hoover et al., 1996] but they all lack at some point of either a generic criterion to decide when a group can be considered as planar, or a way to set thresholding parameters.

In [Igual et al., 2007], the authors propose an *a contrario* region merging procedure to obtain a piecewise affine disparity map. However, the procedure is highly dependent on an initial partition which can be error prone. This initialisation is obtained by assuming that quasi-uniform gray-levels imply a common affine model, which is often, but not always the case, even under Lambertian hypotheses.

Following the same hypothesis, [Facciolo and Caselles, 2009] uses the luminance-geodesic Voronoi cells of a sparse disparity map to provide an initial piecewise affine interpolation. From this partition, a merging procedure is proposed to find the final interpolation. However, due to the computational complexity of the geodesic distance, its use with our quasi-dense disparity maps becomes prohibitive.

2.1.2 Parameter settings and statistical decision

One of our main objectives is to limit as much as possible the use of parameters. In most approaches that have been presented before, some of the basic and sometimes critical decisions are made using parameters. These parameters are usually set to a completely arbitrary values which may vary depending on the input data. We shall focus especially on two of the parameters which are recurrent in piecewise-planar segmentation methods.

The first one is to decide wether or not a segmented group should be considered as planar. This step is usually made by using a coarse threshold on the number of points associated to the segmented group. Such validation is however not acceptable since it relies on the point density which can change from a data set to another.

In addition to that, most methods (RANSAC, region growing, split and merge) uses a distance threshold that we will note τ_z to decide when a point can be considered as part of a given plane. Points for which the distance to a plane is inferior to τ_z are associated to this plane, otherwise are rejected. When this parameter is wrongly set, the resulting segmentation may change a lot from the one that is expected (see Figure 2.1 for an illustration of the problems due to this threshold). Too small values tend to reject too many points which produces an over-segmentation. As opposed to that, a too high value associates too many

points to planes and produces a sub-segmentation. This parameter is both critical and hard to set.

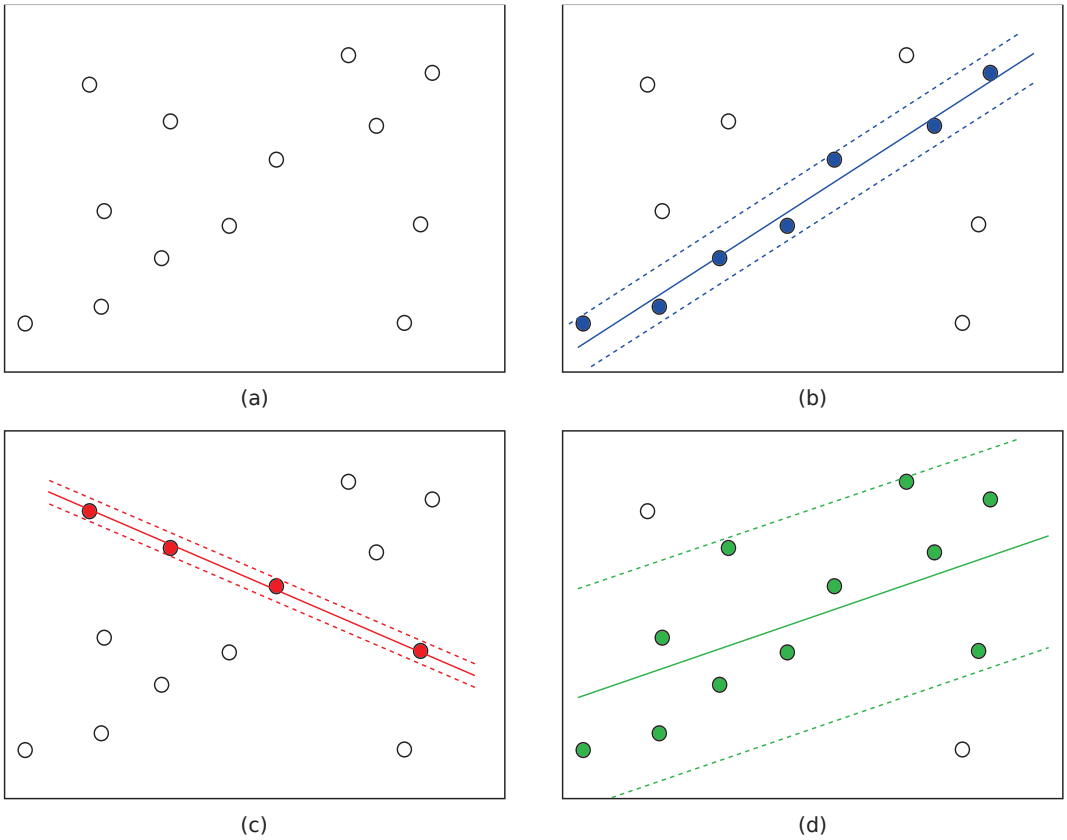


Figure 2.1: Influence of the distance threshold parameter on plane computation. (a) Initial point set. (b) Optimal detection threshold. (c) Too small threshold. (d) Too large threshold. A wrong threshold may change a lot the value of the computed plane.

Though the choice of these two parameters is of capital interest, an automatic setting has rarely been addressed in other segmentation methods. This problem will be addressed here by the *a contrario* methodology proposed by [Desolneux et al., 2008b] for image analysis. The core of this method is the so called Helmholtz principle, which supposes that a specific point organization has a low probability to occur in noise. The idea is then to compare a particular observed event to what could possibly happen with a background noise model. The *a contrario* framework will then allow to set the parameters with a statistical decision.

This type of hypothesis testing was first proposed in Computer Vision by [Lowe, 1985], [Grimson and Huttenlocher, 1991] or [Stewart, 1995]. In the next paragraph, we will discuss anterior *a contrario* methods. We will see that this framework was used among other things for various model detection, selection and validation problems.

Previous *a contrario* works

Burrus' image segmentation. [Burrus et al., 2009] recently proposed a new way to set detection thresholds for image segmentation. Their *a contrario* model is constructed in such a way that under their background hypothesis, the number of false alarms is ensured to stay below a given threshold. The detection thresholds are learnt by Monte-Carlo simulations in such way that a target NFA rate is ensured.

The point of this method is to simplify an initially over-segmented partition of an image. A set of thresholds is defined to decide when two neighboring regions can be considered as different. If this is the case, the two regions are kept separated otherwise they are merged.

Cao's N-dimensional clustering. In [Cao et al., 2007], the authors propose a method to detect meaningful clusters in an N-dimensional distribution of points. Their method was successfully applied to shape detection, matching and grouping problems in images. They first define an *a contrario* model to characterize clusters. A cluster is said to be meaningful if its density in a given N-dimensional region is far superior to what we are expecting to observe with data distributed according to their *a contrario* background model. The regions used for the density measurements are a finite number of predefined N-dimensional hyper-rectangles.

Their heuristic used for the cluster testing is based on the construction of a dendogram. Starting from the set of all singletons as the initial set of nodes, the two closest nodes are merged to form a new parent. This process is then iterated until the tree is fully built. In the end, the root node is the group containing all the points.

The dendogram is at last explored and the *a contrario* criterion is used to decide which groups are meaningful. An additional criterion is moreover defined to see if two groups in the dendogram should be merged or left separated.

Rabin's multi-model detection. In [Rabin et al., 2009], the authors proposed a new way to objects from pairs of images. They suppose that each object in a scene and common to both images, is described by an unknown number of feature points and a transformation to match those points from the first image to the second. The transformation is due to a change in the point of view of the object in the two images. It is supposed to be either a homography, an affine transformation, or a similitude.

The authors propose an extension of the *a contrario* criterion proposed in [Moisan and Stival, 2004] for epipolar geometry computation and validation. Here, a point transformation is considered as meaningful if the number of matched points for this transformation has a low probability to happen with random points. Points from the two images are considered as matched if their distance after applying the point transformation to the points of the first image is less than a threshold. The whole interest of their method is that the threshold parameter (common to all RANSAC approaches) is set automatically by the *a contrario* criterion.

This validation criterion is associated to a RANSAC approach (applied sequentially) to find all the principal transformations in the two images. To avoid possible ghost transformations (wrongly detected transformations that are the combination of several real ones), for each validated group a division into two subgroups with distinct transformations is tested. The choice between the two-grouped model and the single-grouped model is then chosen using an *a contrario* model selection.

Igual's plane validation. [Igual et al., 2007] propose a new approach for piecewise-planar

segmentation of disparity maps. From an initial segmentation into a given number of regions, the author define an *a contrario* measure to first define which region is planar and which one is not, and then to possibly merge two regions together. The initial segmentation is computed on the reference image using a simplified version of Mumford-Shah's algorithm[Koepfler et al., 1994].

For each region, the plane fitting the largest number of points is computed. If such an amount of points is not likely to happen under a random hypothesis then the region is validated as planar. The same criterion is then adapted to two regions for a merging purpose. If considering the two regions as a single one gives a lower number of false alarms then the regions are merged.

Though this method gives good results almost everywhere, it has two drawbacks:

- A wrong initial segmentation propagates error through the whole algorithm. Indeed, this image segmentation is sensitive to strong change of contrast which may only be due to texture on the scene. Moreover, the piecewise planar model is not always consistent with the images of the scene.

- The plane computation and validation depends on a distance threshold to reject points that are too far from the considered plane. As said before for the case of RANSAC-based algorithms, this threshold is critical on the final result.

2.1.3 Overview

The Chapter is organized as follows. In Section 2.2, we introduce a new *a contrario* criterion for validating a planar model and for selecting the best model among single or multiple planes. This criterion is similar to the one introduced in [Igual et al., 2007] but does not rely on the segmentation constraint or on a threshold parameter for plane computation. Unlike classical model selection criteria like AIC and BIC [Akaike, 1974; Schwarz, 1978] which are quite similar in nature, the proposed *a contrario* criterion serves also as a validation method. In Section 2.3, we propose a new algorithm using the *a contrario* criterion to search for the different planes in a disparity map by means of a split & merge strategy. The heuristic is only based on the knowledge of the 3D points which ensures a purely 3D segmentation. Using such an approach is moreover a sanity check for our criterion since it is based on its quality as a model selection criterion. At last, the experimental results in Section 2.4 support the effectiveness of the proposed algorithm and its potential use for interpolation, de-noising, and vectorization of urban DEMs.

2.2 *A contrario* plane validation

Most segmentation methods presented in the introduction lack of an automatic criterion to decide when a group found by any algorithm actually is a plane. When a decision is made, this is usually done by keeping the groups for which the size is superior to a predefined threshold, which is difficult to tune in a universal manner. We propose here to use an *a contrario* framework to make the decision automatic. This framework will moreover allow to set other critical parameters such as the outlier rejection distance threshold τ_z or to decide which model should be used to fit the data.

Following a similar methodology as the one done in [Igual et al., 2007], a group can be considered as planar if observing a similar planar configuration with purely random points

(*background process*) is very unlikely. In such a case, according to the Helmholtz principle, the background hypothesis has to be rejected in favor of the detection of a significant planar patch. The planarity of a group is measured by counting the points nearer than a certain distance threshold to a plane.

An intuitive justification of the *a contrario* methodology is given with Figure 2.2.

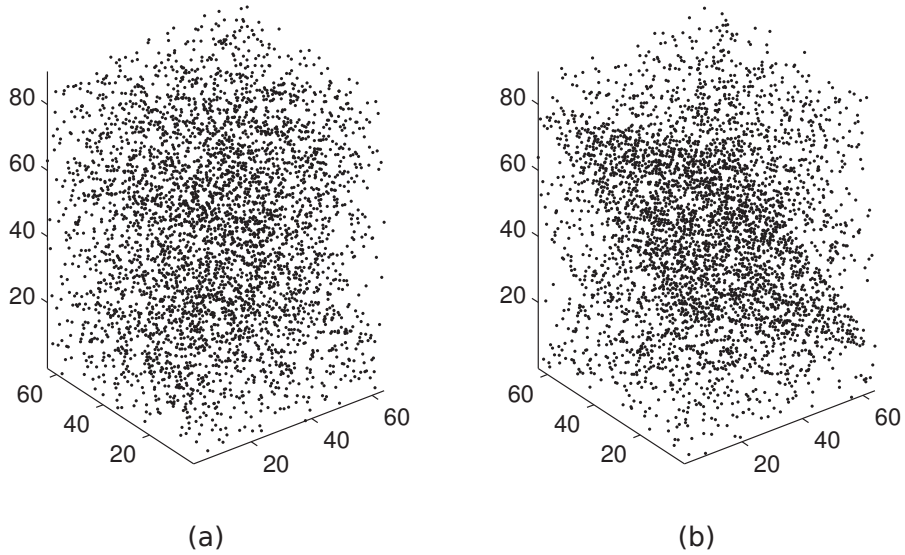


Figure 2.2: *A contrario* validation. (a) Points distributed randomly (*background process*). (b) Same distribution with 30% of points projected on a plane. As opposed to (a), (b) is very unlikely to be a pure realization of a random process. Visually, if a plane can be guessed among random points, then it is not likely that the points on the plane are part of the same random process as the random points. The Helmholtz principle then states that since it is very unlikely, under a randomness assumption, to observe as many points near a plane as we do in (b), (b) is not due to random. It is therefore validated as a plane.

2.2.1 Data and background process

The *a contrario* framework is based on the comparison between data and points randomly distributed according to a process called **background process**.

A (deterministic) disparity map z is a mapping where each 2D point $\mathbf{x}$ on a discrete grid $\Omega \subset \mathbb{Z}^2$ (image plane) is associated to a disparity (or height) $z \in \mathbb{R}$. z depends on the observed object on the disparity map and can take any value in $[z_{min}, z_{max}] \subset \mathbb{R}$. It can also be seen as the realization of the random process $\mathcal{Z}(\mathbf{x})$ defined as:

Definition 1 (Background process) *The background process is a finite process* $\mathcal{Z}(\mathbf{x}) \sim \mathcal{U}([z_{min}, z_{max}])$, $\mathbf{x} \in \Omega$, *made of mutually independent variables.*

Let N_Ω be the number of points in the discrete grid Ω. Let's consider a group G of N_G data points, a plane π and a distance rejection threshold τ_z. We can now compare $\mathcal{K}$, the (random) number of points out of N_Ω with a random disparity value (drawn from the

background process) nearer than τ_z to π, to k the actual number of points from G nearer than τ_z to π. If the probability $\mathbb{P}[\mathcal{K} \geqslant k]$ is small enough then the planar grouping π of the points in G cannot be simply explained by the background process. The group G is then considered as a meaningful planar patch.

Such a comparison is unfair since it penalizes small groups (even when they are actually planar) because the comparison is made on all the N_Ω points. Since our goal is to detect planar facets which are very localized in space, the comparison of the background process should take that locality into account. We therefore introduce a set of regions within the disparity maps of various size and shape to be able to compare point groups even when they are spatially very localized. Let $\mathcal{R}$ be such a sufficiently rich set of regions such that $\bigcup_{R \in \mathcal{R}} R = \Omega$.

2.2.2 Meaningful planes

Definition 2 (Number of False Alarms (NFA)) *Let $G \subseteq \Omega$ be a group of points of the disparity map. Let $R \supset G$, $R \in \mathcal{R}$ be a region containing G, π be a plane and τ_z a tolerance threshold defining when a point belongs to π (τ_z may be different at each point). The NFA of G according to (R, π, τ_z) is defined as:*

$$NFA(G, R, \pi, \tau_z) \equiv N_{tests} \mathbb{P}\left[\mathcal{K}(R, \pi, \tau_z) \geqslant k(G, \pi, \tau_z)\right] \tag{2.1}$$

where

- $k(G, \pi, \tau_z) = \sum_{\mathbf{x} \in G} \mathbb{1}_{\{|z(\mathbf{x}) - z_\pi(\mathbf{x})| < \tau_z(\mathbf{x})\}}$ counts the number of points from G that are sufficiently close to π. In *a contrario* methods, k is commonly referred to as the *degree of coincidences*.
- $\mathcal{K}(R, \pi, \tau_z) = \sum_{\mathbf{x} \in R} \mathbb{1}_{\{|\mathcal{Z}(\mathbf{x}) - z_\pi(\mathbf{x})| < \tau_z(\mathbf{x})\}}$ is a random variable counting the number of points in R that are sufficiently close to the plane π supposing that the disparity values are random variables (following the *background process*).
- N_{tests} is the number of configurations of planes and regions that can be tested.

For simplicity reasons, we will note k and $\mathcal{K}$ to refer $k(G, \pi, \tau_z)$ and $\mathcal{K}(R, \pi, \tau_z)$.

Analysis of equation (2.1)

The computation of k and $\mathcal{K}$ is done by computing a distance between a point and its projection along the z-axis on a plane π.

The probability $P\left[\mathcal{K} \geqslant k\right]$ is the probability that a random vector $\mathcal{Z}(R)$, of size defined by a region R, has at least as many coincidences for plane π as the observed data vector $z(G)$.

From figure 2.3, it can be easily seen that for any point $\mathbf{x} \in R$, $\mathcal{K}(\mathbf{x}, \pi, \tau_z)$ is a Bernouilli random variable of parameter $p \leq 2 \cdot \tau_z / (z_{max} - z_{min})$. Supposing a constant value for τ_z and p (each plane is equi-probable), $P\left[\mathcal{K} \geqslant k\right]$ is statistically the same for any region $R \in \mathcal{R}$ of constant size. In that case, $NFA(G, R, \pi, \tau_z)$ is the expected number of time that one can observe at least as many coincidences with random points in a region R as what is observed for G and π, this by testing all the possible planes and all the possible regions of the same size as R.

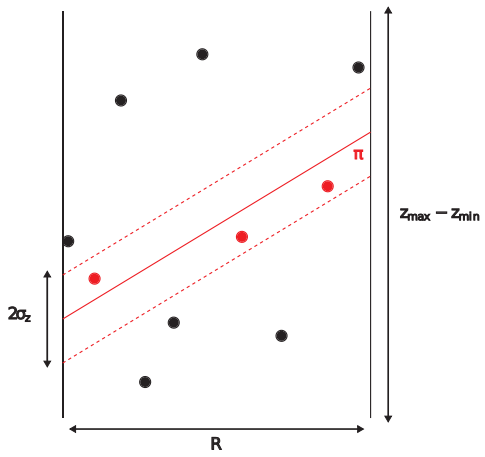


Figure 2.3: Probability of a random point to be associated to plane π

Remark (distance threshold): In most methods described in the introduction, the distance rejection threshold τ_z is supposed to be known. We recall (see Figure 2.1) that the value of this parameter is usually critical to the result. In [Sabater, 2009], Sabater gave an estimation of the finest disparity precision that can be expected at each point when the disparity was computed with a block matching approach. The parameter τ_z in the NFA computation can therefore be set to the value given by [Sabater, 2009].

However, in the more general case, the value of τ_z is unknown. We will see in section 2.3.4, how to set this parameter using the *a contrario* framework. A similar methodology was used in [Rabin et al., 2009] for plane detection in unconstrained stereo-vision.

In the rest of the chapter, we will therefore consider the value of τ_z as known.

Remark (regions and planes): From Equation (2.1) we see that the NFA of a group G depends on a region R to which it is compared to, and on a plane π that we want to fit to the points of G.

One could compare G to any point group of the same size as itself, but in that case, would loose the information of locality of G since a group of N_G points can be widespread in Ω as well as concentrated in a small area. The purpose of using a region $R \supset G$ is then to keep that information in mind during the comparison. The choice of the set of regions $\mathcal{R}$ is then very important. It will be discussed in detail in section 2.2.3.

For a given group G, the region R and the plane π can be chosen as the ones minimizing the NFA:

Definition 3 *The NFA of a group $G \subset \Omega$ is defined:*

$$NFA(G) = \min_{\substack{R \in \mathcal{R}, R \supset G \\ \pi \in \Pi_R}} NFA(G, R, \pi) \tag{2.2}$$

where Π_R is the set of all planes defined for region R.

From equation 2.1, it can be easily seen that the region $R \in \mathcal{R}$ minimizing the NFA of a group G is the smallest region containing G. Indeed, the larger the region, the larger $\mathcal{K}$ is expected to be, which increases the value of $\mathbb{P}[\mathcal{K} \geqslant k]$.

In addition to that, the plane minimizing the NFA is the plane that best fits the points of G. Since any plane is expected to give the same value for $\mathcal{K}$, we then concentrate on the value of k. The more the plane fits to the points of G, the bigger k is, which decreases the value of $\mathbb{P}[\mathcal{K} \geqslant k]$.

A given group G can now be associated to a plane π and a region R: the ones minimizing the NFA. From now on, when referring to the NFA of a group G, we will refer to Equation (2.2).

The next definition and proposition define when a group G can be considered as planar.

Definition 4 (ε-meaningful plane) *A group G is said to be an ε-meaningful planar patch whenever $NFA(G) < \varepsilon$.*

Proposition 1 *Let $\mathcal{S}$ be the set of all the possible pairs (R, π). If we consider a random data set following the background model, then the expected number of ε-meaningful planes in $\mathcal{S}$ is less than ε.*

Proof Let's note S the random variable defined as:

$$S = \sum_{(R,\pi)\in\mathcal{S}} \chi_{(R,\pi)}$$

where, $\chi_{(R,\pi)} = \mathbb{1}_{(R,\pi) \text{ is } \epsilon\text{-meaningful}}$. Using the linearity of the expectation operator one has:

$$E[S] = \sum_{(R,\pi)\in\mathcal{S}} E[\chi_{(R,\pi)}] \tag{2.3}$$

Then, using definition 2 we can write: $E[\chi_{(R,\pi)}] = P[\mathcal{K} \geqslant k(\varepsilon)] \leqslant \frac{\varepsilon}{N_{tests}}$. At last using the definition of the number of tests given in section 2.2.3, and substituting it in equation 2.3, we obtain the result:

$$E[S] \leqslant \sum_{(R,\pi)\in\mathcal{S}} \frac{\varepsilon}{N_{tests}} = N_{tests} \cdot \frac{\varepsilon}{N_{tests}} = \varepsilon \tag{2.4}$$

□

Proposition 2 is of capital importance in *a contrario* methods. It basically says that less than ε a ε-meaningful plane are expected to be detected with a random data set (following the *background process*). Setting the value of ε to 1 (which is a classical choice in *a contrario* approaches) then means that at most one false detection of 1-meaningful plane is expected to occur using random data. From now on, we therefore refer to 1-meaningful planes when we speak about meaningful groups.

2.2.3 Number of tests

The number of tests is given by counting all possible region-plane configurations:

$$N_{tests} = \sum_{R\in\mathcal{R}} \#\Pi_R \tag{2.5}$$

As pointed out before, the choice of the set of regions $\mathcal{R}$ is very important. On one hand, a too small set may penalize some groups since it may not represent well all the groups (for

instance $\mathcal{R} = \{\Omega\}$ is unfair to small groups). On the other hand, a too large set makes it difficult for a group G to be meaningful and does not necessary well describe the situations encountered.

The first thing that we should note is that we look for planar facets. This means that we don't want to validate groups of points scattered in Ω.

A simple choice for $\mathcal{R}$ is the set of all the rectangles in Ω (oriented along the 2 main directions of the map). This choice is similar to the one made in [Cao et al., 2007] for their *a contrario* clustering method for N-dimensional points. As in [Cao et al., 2007], scattered groups will be automatically rejected because they are compared to large rectangular regions. For each scale of planar patch, there will be a rectangle of the proper scale fitting the patch. A small patch will then have the possibility to be validated as meaningful.

This region set can be reduced without loss of precision by limiting it to rectangles with a power of 2 size ($2 \times 2, 2 \times 4, 4 \times 4$, etc.). The effect is then to limit the number of tests which allows the validation of smaller groups. Another possibility, this time to fit even more to the groups is to give other possible orientation to the rectangles than the two main axis.

To describe all the possible planes in a region R, we compute all the triplets of points in R. The number of tests then becomes:

$$N_{tests} = \sum_{R \in \mathcal{R}} \#R \cdot (\#R - 1) \cdot (\#R - 2) \tag{2.6}$$

More specifically, using the rectangular region set along the two main axis and with an $M \times N$ disparity map we have:

$$N_{tests} = \sum_{i=1}^{M} \sum_{j=1}^{N} ij \cdot (ij - 1) \cdot (ij - 2) \cdot (M + 1 - i) \cdot (N + 1 - j) \tag{2.7}$$

2.2.4 Probability of false alarms.

Computing the probability $\mathbb{P}[\mathcal{K} \geqslant k]$ depends on the probability of a random point from the background process to be near a given plane. Supposing a constant threshold τ_z for any point $\mathbf{x}$, and supposing the probability of a random point to be associated to any plane constant with value $p = 2 \cdot \tau_z / (z_{min} - z_{max})$, the probability of false alarms is then the tail of the binomial law:

$$\mathbb{P}[\mathcal{K} \geqslant k] = \mathcal{B}(\#R, k, p) = \sum_{j \geqslant k}^{\#R} \left(\begin{array}{c} \#R \\ j \end{array} \right) p^j (1 - p)^{\#R - j} \tag{2.8}$$

When τ_z is not constant (this can happen when the precision estimation at each point is used as values for τ_z), $\mathbb{P}[\mathcal{K} \geqslant k]$ can be accurately approximated using the Hoeffding theorem [Hoeffding, 1963]:

$$\mathbb{P}[\mathcal{K} \geqslant k] \leqslant e^{\#R\omega(\eta - \mu, \mu)} \tag{2.9}$$

with,

$$\omega(\eta, \mu) = (\mu + \eta) \log(\frac{\mu}{\mu + \eta}) + (1 - \mu - \eta) \log(\frac{1 - \mu}{1 - \mu - \eta}) \tag{2.10}$$

and

$$\eta = \frac{k}{\#R}, \quad \mu = \mathbb{E}[\frac{\mathcal{K}}{\#R}] = \frac{1}{\#R} \sum_{\mathbf{x} \in R} 2 \frac{\tau_z(\mathbf{x})}{z_{max} - z_{min}} \tag{2.11}$$

2.2.5 Validation of the *a contrario* model

Let's now numerically test the validity of Definition 2 and Proposition 2. To do so, we first created a 512×512 random disparity map where $Z(\mathbf{x}) \sim \mathcal{U}([0, 512]), \forall \mathbf{x} \in \Omega$.

Since computing all the NFA for all the rectangular regions and all the triplets of points in Ω, is an $\mathcal{O}(N_\Omega^5)$ algorithm (where N_Ω is the number of points), testing all the possibilities would require 30 billion years of computation with 1GHz processor.

We propose to reduce the number of tests to limit the number of computations. The first step to that is to limit the number of regions. As proposed in section 2.2.3, one can use rectangular regions for which the size is only a power of 2: 4×4, 4×8, 8×8, 4×16 etc. From these regions, we then take only a few of the rectangles so that the centers of rectangles of a given size are regularly spaced on the grid with a distance of half their size.

Compared to taking all the rectangular regions, for a disparity of size $M \times N$, the number of regions is reduced from $\sum_{i=1}^{M}\sum_{j=1}^{N} ij = \mathcal{O}\left((MN)^2\right)$ to $\sum_{i=1}^{\log_2 M}\sum_{j=1}^{\log_2 N} (2^i+1)\cdot(2^j+1) = \mathcal{O}(MN)$.

The second step to reduce computations is to reduce the number of planes to be tested. This can be done by taking only a limited number of planes sorted out randomly (we used $M \cdot N$ planes for this experiment).

We tried to compute the NFA for the limited region set and by computing 512×512 random planes (triplets of points). When computing Equation (2.1), N_{tests} was adapted to our reduced set of planes and regions. For each plane, we tried several precision threshold $\tau_z \in \{0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256\}$. For each tested value, no meaningful plane was found in our random disparity map as expected from Proposition 2. Though the test was done on a limited set of planes and regions, this experiment tend to confirm the validity of our model.

2.2.6 Correlated points and sparse data

In the definition of the background process (Definition 1) that is used to define the data, the disparity of each point is assumed to be independent of the other points. In practice, when a disparity map has been computed using a block-matching approach, this assumption is false since the correlation is computed on a neighborhood given by the correlation window. Each point is then related to all the points within this correlation window.

To only consider independent points, one can subsample the data by taking only one point per correlation window. In practice, this is equivalent to weighting the values of k and $\mathcal{K}$ in Equation (2.2). Instead of being the number of valid points, k and $\mathcal{K}$ are changed to the sum of the density weights $w_d\mathbf{x}$ of each valid point $\mathbf{x}$ where the density weights are defined as:

$$w_d(\mathbf{x}) = \frac{1}{\sum_{\mathbf{x}' \in B_\mathbf{x}} \mathbb{1}_{\{z(\mathbf{x}') \text{is defined}\}}} \tag{2.12}$$

where $B_\mathbf{x} \subset \Omega$ is the correlation at point $\mathbf{x} \in \Omega$.

The second advantage of this normalization is that it is adapted to deal with regions where not all the points are known. This allows to still validate meaningful planes in sparse configurations.

2.2.7 *A contrario* model selection

Due to its generic form, the NFA can be easily adapted to test the validity of a configuration with several groups, planes and regions. This allows to choose between two configurations: a

simple one with one plane and a more complicated one with several planes. We give here an example for the case of 2 groups by using a similar approach as the one done in [Igual et al., 2007].

Using similar notations as before, we define the joint NFA for two groups G_1 and G_2 as:

Definition 5 (joint NFA for two groups, NFA_j) *Let $G_1 \subset \Omega$ and $G_2 \subset \Omega$, $G_1 \bigcap G_2 = \emptyset$, be two distinct group of points. We define the NFA of the two groups to be planar as:*

$$NFA_j(G_1, G_2) \equiv N''_{tests} \cdot \mathbb{P}\left[\mathcal{K}_1 + \mathcal{K}_2 \geqslant k_1 + k_2\right] \tag{2.13}$$

where,

- *k_1 (resp. k_2) is the number of coincidences in G_1 (resp. G_2) of the plane best fitting to its points.*
- *$\mathcal{K}_1$ (resp. $\mathcal{K}_2$) is the random variable counting the number of coincidences in the region best fitting to G_1 (resp. G_2) of any plane.*
- *N''_{tests} is the number of configuration of pairs of planes and pairs of regions that can be tested. Following similar notations as before, the number of tests is defined as:*

$$N''_{tests} = \sum_{\substack{R_1 \in \mathcal{R} \\ R_2 \in \mathcal{R} \setminus R_1}} \#\Pi_{R_1} \cdot \#\Pi_{R_2} \tag{2.14}$$

  Note that this number of tests is larger as it was before for a single group. This is a way of penalizing more complex models that best fit to the data.

The point of the joint NFA is to be able to decide which model gives the best description of a given group: two planes or a single one. To do this comparison for a group $G = G_1 \bigcup G_2$, we compare $NFA(G)$ and $NFA_j(G_1, G_2)$. However, for a fair comparison, the same conditions needs to be used especially for the computation of the background number of coincidences ($\mathcal{K}$ for $NFA(G)$ and $\mathcal{K}_1 + \mathcal{K}_2$ for $NFA_j(G_1, G_2)$). This means that the regions used for the background computation needs to be the same. We therefore introduce a new NFA definition of a group $G = G_1 \bigcup G_2$ for a proper comparison with $NFA_j(G_1, G_2)$.

Definition 6 (comparison NFA, NFA_c) *Let $G_1 \subset \Omega$ and $G_2 \subset \Omega$, $G_1 \bigcap G_2 = \emptyset$, be two distinct group of points, and let $G \subset \Omega$ be the group defined as $G = G_1 \bigcup G_2$. We define the NFA of G to be a plane knowing the division into two subgroups G_1 and G_2 as:*

$$NFA_c(G = G_1 \bigcup G_2) \equiv N'_{tests} \cdot \mathbb{P}\left[\mathcal{K}_1 + \mathcal{K}_2 \geqslant k\right] \tag{2.15}$$

where,

- *k is the number of coincidences in G of the plane best fitting to its points.*
- *$\mathcal{K}_1$ (resp. $\mathcal{K}_2$) is the random variable counting the number of coincidences in the region best fitting to G_1 (resp. G_2) of any plane.*
- *N'_{tests} is the number of configuration of planes and pairs of regions that can be tested:*

$$N'_{tests} = \sum_{\substack{R_1 \in \mathcal{R} \\ R_2 \in \mathcal{R} \setminus R_1}} \#\Pi_{R_1 \bigcup R_2} \tag{2.16}$$

To decide which configuration is the best between a pair of planar groups (G_1, G_2) and a single one $G = G_1 \bigcup G_2$, one just needs to compute $NFA_c(G)$ and $NFA_j(G_1, G_2)$. The smallest value then determines the best configuration.

Discussion on rectangular regions

A question that raises when using pairs of rectangular regions (for the comparison with the background model) is what happens when the two regions overlap.

In such case, taking R_1 and R_2 separately is equivalent to considering the background model twice in the region $R_1 \bigcap R_2$. One could choose to count only once the intersection region but this would make the number of tests a lot harder to compute we therefore choose to keep considering R_1 and R_2 separately even when they overlap.

Now, what are the consequences of doing so? First of all, since the background model is considered twice, this will make it harder for a given group (or pair of groups) to be meaningful. Now, looking at Equations (2.13) and (2.15), we note that the background model is computed under the same conditions. Since the goal here is comparison, overestimating the background model is not really a problem as long as this is done equivalently for both configurations.

Analogy with AIC and BIC

In statistical model selection, standard criteria are AIC [Akaike, 1974] and BIC [Schwarz, 1978]. Both depend on the likelihood and on the complexity of the tested model. Though different in their way of computation (AIC is found using the Kullback-Leibler distance and BIC is obtained by approximating the integrated likelihood), their final expressions are pretty similar. For a group G and a tested model $\mathcal{M}_i$ they are defined by:

$$\begin{aligned} AIC &= -2\log(\mathcal{L}_{\mathcal{M}_i}(Z,\hat{\theta}_i)) + 2V_i \\ BIC &= -2\log(\mathcal{L}_{\mathcal{M}_i}(Z,\hat{\theta}_i)) + 2V_i\log(\#G) \end{aligned} \tag{2.17}$$

where $\mathcal{L}_{\mathcal{M}_i}$ is the likelihood of model $\mathcal{M}_i$ for the data Z and V_i is the number of parameters of model $\mathcal{M}_i$.

Let's now take a look at the log of the NFA. We will consider here the case of two groups G_1 and G_2 and compute $\log\left(NFA_j(G_1,G_2)\right)$ and $\log\left(NFA_c(G_1 \bigcup G_2)\right)$ as we would normally do for model selection:

$$\begin{aligned} \log\left(NFA_j(G_1,G_2)\right) &= \log\left(\mathbb{P}[\mathcal{K}_1+\mathcal{K}_2 \geqslant k_1+k_2]\right) + \log\left(N''_{tests}\right) \\ \log\left(NFA_c(G_1 \bigcup G_2)\right) &= \log\left(\mathbb{P}[\mathcal{K}_1\mathcal{K}_2 \geqslant k]\right) + \log\left(N'_{tests}\right) \end{aligned} \tag{2.18}$$

For simplicity reasons, let's now suppose that we only have regions of a single size $\#R$ (we note N_R the number of such regions). The two equations before can then be written as a single one. Using similar notations as for the AIC and the BIC we then have:

$$\log(NFA(G,\mathcal{M}_i)) = \log(\mathbb{P}[\mathcal{K} \geqslant k(\mathcal{M}_i)]) + V_i\log(\frac{V_1}{V_i}\#R) \tag{2.19}$$

where V_i is the number of parameters of the considered model and V_1 is the number of parameters of a single plane.

We now see that all the three expressions have two terms:

- The first one tells how well the model fit the data.

  For the AIC and BIC criterion, this term is the negative of the log likelihood. Because of the minus sign, minimizing it means that you maximize the log likelihood which means that your model well describes your data.

  For the log NFA, the first term is the probability that the model explains random data as well as it explains the observed data. When the model well describes the data, then it usually doesn't well explain random data and the log of the probability is minimum.

- The second term in all three criteria, is a regularity term. It describes how complex the model is (number of parameters of the model) and is minimum when the model is simple. Its value is different for the three criteria but is always a function of V_i, the number of parameters necessary for the model.

All three criteria then give a trade-off between fitting the data and having a simple model. However, the main advantage of the *a contrario* criterion over the other two is to give a way to automatically validate planar groups.

2.3 Plane search

In the previous section, we introduced a method to first, decide when a point group of a disparity map can be considered as planar, then, to choose between two possible models to describe the points. Since testing the planarity of all the possible groups is impossible for obvious computational reasons, we need to find a method to explore the data and find groups that may be defined as planar (up to our criterion). The main difficulties here are that we do not have any information on the final number of planar groups or on their size, shape or position.

In this section, we propose a heuristic based on the NFA to find potential planes in disparity maps. Note that other heuristics can be used in conjunction with the NFA as a validation criterion. The algorithm is based on the assumption that up to a certain scale, any smooth and continuous surface can be considered as planar. The algorithm we propose is two-step: first, a top-down dyadic division (splitting step) is done until a good solution is reached, then the resulting groups are merged (merging step) to refine the result. In both the division and the merging step, the NFA is used as a decision criterion. The main advantage of this method over [Igual et al., 2007; Facciolo and Caselles, 2009] is to rely only on the 3D information of the points which avoids errors due to image segmentation.

2.3.1 Splitting step

A 3-Dimensional bounded plane can be seen here as the realization of a 3-dimensional Normal random process for which one of the eigen-values of the covariance matrix is very weak compared to the other two (the corresponding eigen-vector is then orthogonal to the considered plane). Since, as said before, up to a certain scale, a disparity map can be considered as piecewise planar, a Gaussian mixture model seems to be a good description of the point distribution. We recall that under the Gaussian mixture hypothesis, an observed 3D point $\mathbf{x}$ is supposed to be due to the contribution of several Gaussian distributions, where the influence of each distribution Γ_i on $\mathbf{x}$ is given by:

$$p_{\Gamma_i}(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{3/2}|\Sigma_i|} e^{-1/2(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)} \tag{2.20}$$

The EM algorithm [Dempster et al., 1977], is an approach to find the best segmentation in the maximum likelihood sense, of a data point set into N point distributions. The algorithm is the alternation of two steps:

- *Expectation* step: given, the parameters of the N distributions, each point is associated to the distribution that best describes it.

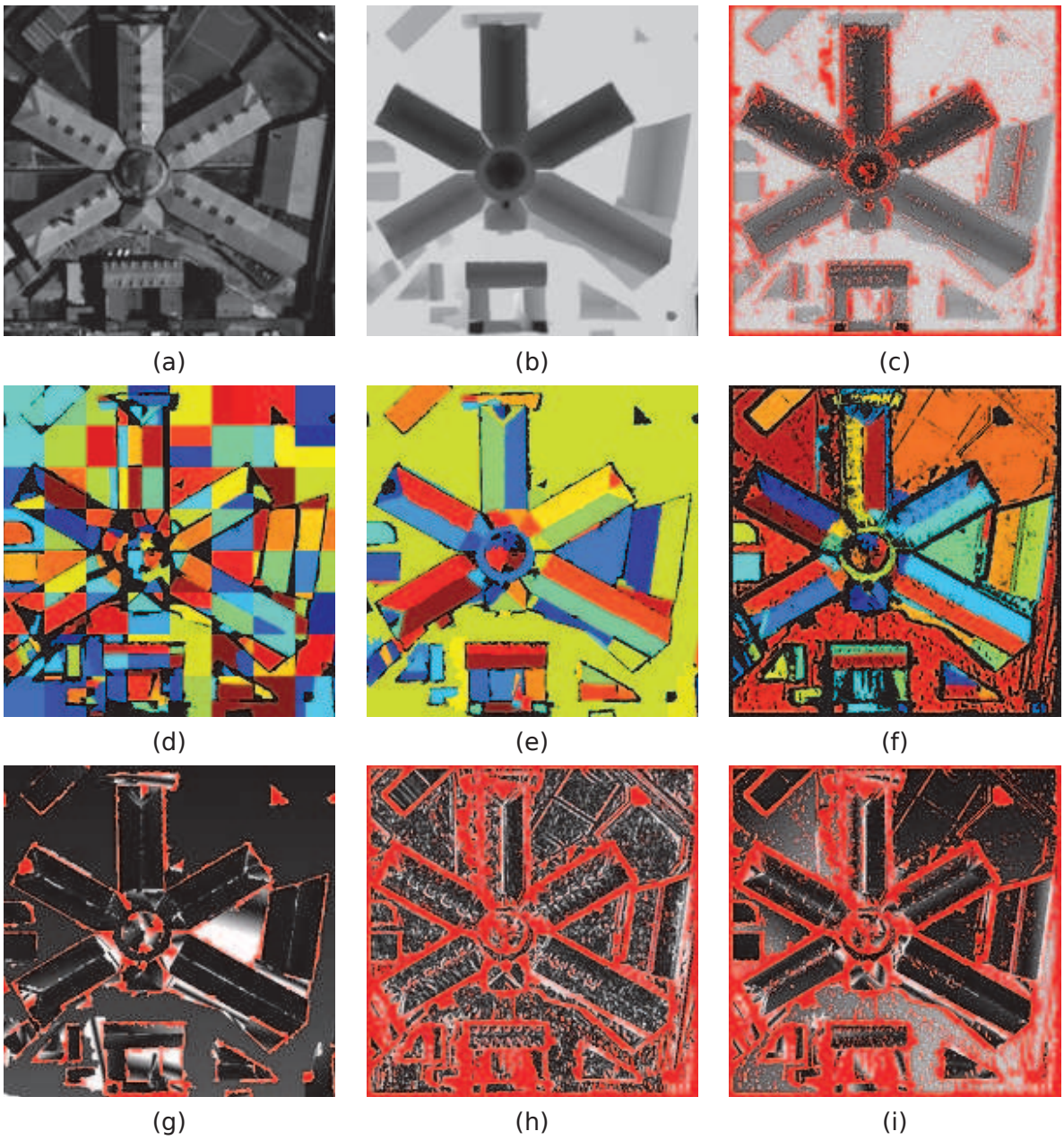


Figure 2.4: Split and merge algorithm on Toulouse's St-Michel Jail dataset. (a) Reference Image. (b) Ground truth disparity map. (c) Computed disparity map [Sabater, 2009] (red parts are unknown). (d) Segmentation found after splitting step (with extension proposed in Section 2.3.5 on the ground truth with additive noise (each color represents a meaningful-plane, black parts are either unknown points or non-meaningful plane). (e) Merging step on (d). (f) Segmentation obtained with real data (c). (g) Absolute value of the difference (L_1-error) between the ground truth and the planar projection with our segmentation (dark = low error, bright = large error). (h) L_1-error between the ground truth and the initial disparity map (c). (i) L_1-error between the ground truth and the projection with the planes obtained in (f). The non-structured errors have been removed from (h) to (i).

- *Maximization* step: given a segmentation of the points into N groups, the parameters of each distribution are computed according to the maximum likelihood

The algorithm converges to a maximum, however, there is no insurance that this is a global maximum. The result then depends on the initialization which can be done either on the segmentation, either on the parameter of the N distributions.

The main drawback of the EM algorithm is that it is based on the knowledge of the number of classes we look for (which we actually do not know). We instead propose to build a dyadic tree which only requires the division into two groups at each node of the tree, which can now be done using EM. At each possible node, a decision is made to choose if splitting the group into two parts was reasonable or not. The choice of the configuration is based on the computations of the NFA as suggested in Section 2.2.7. The construction of the tree starts from the group containing all the points and the division stops when a good configuration is reached.

The decision between one group or two groups is made by choosing the most meaningful configuration according to the computation of the NFA. Moreover, we chose to force the division in the case where the main group is not meaningful. Indeed, a group that is not considered as a meaningful plane may contain a meaningful subgroup. The divisions needs therefore to continue in order to find small meaningful planes. To sum up, the one-grouped configuration is kept when:

$$\begin{cases} NFA(G) < 1 \\ NFA_c(G = G_1 \bigcup G_2) < NFA_j(G_1, G_2) \end{cases} \tag{2.21}$$

where, G_1 and G_2 are the two resulting groups of the EM algorithm on G. A summary of the whole splitting step is given by Algorithm 1.

Stop criterion

The first condition of division, $NFA(G) > 1$ implies that if a group G does not contain any meaningful subgroups, it will be divided until it is reduced to monomes. The problem is that even if G is not a meaningful plane, keeping G as a single group may be a better description than dividing it into monomes. Another difficulty due to this over-segmentation is that the merging step of the algorithm (see Section 2.3.2) gets time consuming because the number of pairs to be merged increases a lot.

This problem can be overcome by limiting the division to groups bigger than the smallest size authorized by the NFA. Indeed, for each region size, one can compute the minimal size for a group to be meaningful: N_{min}. N_{min} depends on the number of tests and on the region the group is compared to. We note R_{min} the smallest region containing the minimal possible meaningful group. For instance, given a 512×512 disparity map, testing only the regions of size that is a power of two and supposing that the points are uncorrelated, R_{min} is a 4×4 region and $N_{min} = 9$.

2.3.2 Merging step

Since the division process is dyadic, the final partition of the points might not be optimal. Better configurations may then be found by merging some of the groups. Since we are interested in finding planar patches, we only try to merge groups that are neighbors to each other. We therefore build an adjacency graph of the groups. The groups that are connected are the candidates for merging.

Algorithm 1: Splitting step

Data:
G_0, a group of points of Ω
D_{max}, the maximal depth of the tree
Result:
$\mathcal{G}_{final} = \{G_1, \ldots, G_N\}$ such that $G_0 = \bigcup_{i=1}^{N} G_i$

```
1  begin
2      G = {G_0} is the set of all the groups
3      G_next = ∅ the next set of groups to be tested
4      G_final = ∅ is the set of validated groups
5      D = 0, the current depth of the tree
6      while D < D_max do
7          foreach G ∈ G do
8              (G_1, G_2) ⟵ EM algorithm(G)
9              nfa ⟵ NFA(G)
10             nfa1 ⟵ NFA_c(G_1 ⋃ G_2)
11             nfa2 ⟵ NFA_j(G_1, G_2)
12             if [(nfa < 1) and (nfa1 < nfa2)] or [G ⊂ R_min and #G < N_min] then
13                 add G to G_final
14             else
15                 Add (G_1, G_2) to G_next
16             end
17         end
18         D = D + 1
19         if D < D_max then
20             G = G_next, G_next = ∅
21         else
22             G_final = G_final ⋃ G_next
23         end
24     end
25 end
```

The decision of merging or not a pair of groups is made using the same criteria as for the splitting step. However, the final segmentation may be dependant on which groups are merged first. We therefore need a criterion to decide the merge order.

Following a similar reasoning as [Burrus et al., 2009], we introduce the following contrast factor:

$$F(G_1, G_2) = \frac{NFA_c(G_1 \bigcup G_2)}{NFA_j(G_1, G_2)} \tag{2.22}$$

The lower $F(G_1, G_2)$ is the lower the NFA of a single group is compared to the one of two groups. In other words, the lower the contrast factor is the more likely it is to have a configuration using a single group. A priority queue is built using growing contrast factor (the lowest contrast factor is the first).

Whenever two groups G_1 and G_2 are merged, G_1 and G_2 are removed from the adjacency graph and the new node $G_1 \bigcup G_2$ is added and connected to whatever node G_1 or G_2 were connected. The same goes for the priority queue, each pair containing either G_1 or G_2 is removed, and the new edges created in the adjacency graph are added. The complete merging process is shown in algorithm 2. Its effectiveness is illustrated with Figure 2.4 (d) and (e) where the groups from the initial segmentation are merged into a coherent segmentation.

Algorithm 2: Merging step

Data:
$\mathcal{V} = \{G_1, \ldots, G_N\}$, where G_i is a group of points of Ω
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the adjacency graph of the groups.
Result:
$\mathcal{G}'$ a simplification of the adjacency graph $\mathcal{G}$ obtained by merging nodes of $\mathcal{G}$

```
1  begin
2      G' = G.
3      Q = ∅ is the priority queue of merges.
4      foreach (G1, G2) ∈ E do
5          add (G1, G2) to Q
6      end
7      while #Q > 0 do
8          (G1, G2) = arg min_(G,G') F(G, G')
9          Remove (G1, G2) from Q
10         if (F(G1, G2) < 1)&&(NFA(G1 ∪ G2) < 1)
11         then
12             Merge G1 and G2 in graph G'
13             foreach k such that (G1, Gk) ∈ Q or (G2, Gk) ∈ Q do
14                 Remove (G1, Gk) and (G2, Gk) from Q
15                 Add (G1 ⋃ G2, Gk) to Q
16             end
17
18     end
19 end
```

2.3.3 Plane computation

When computing the NFA for both the splitting and the merging step, we need to compute the planes (or the two planes) maximizing the number of coincidence of a group G for a given precision τ_z. Choosing the best plane is of capital importance since the segmentation is based

on the comparison between configurations. For instance, in the division process, if the planes computed for the two subgroups are poorly estimated, this may favor the configuration with a single group and stop the division process here.

Due to the potential presence of outliers (the splitting step starts with the group containing all the points, which is not likely to be planar), computing the plane by a least squares approach is likely to fail. A robust computation method is then necessary.

From the discussion on the various robust plane regression methods done in Appendix A, we chose to use a RANSAC algorithm (sort N_{iter} triplets of points and take the one giving the best number of coincidence). The number of iteration N_{iter} used for the RANSAC algorithm is computed as proposed in appendix A. We supposed that for each considered group, at least half of the points are part of the best plane. We moreover expect that the probability that none of the N_{iter} tested triplets of points are on the best plane is less than 0.0001. Then $N_{iter} \geq \log(0.0001)/\log(1-0.5^3) \sim 70$.

2.3.4 Estimating the precision threshold τ_z

As discussed in the introduction, the precision threshold τ_z that is used to decide what point is considered as inlier to the plane is critical to the RANSAC result. When its value is unknown, it is preferable to find a way to set it.

In [Rabin et al., 2009], the authors sort the points by their distance to a given plane. They then try various threshold values defined as the distance of each point to the plane. Each threshold then defines the addition of one more point to the inlier list. From this threshold list, the one minimizing the NFA value is the one that is kept. Though the method works very well for their problem where at most a few thousands of points are considered, it is not applicable in our case where the number of points is a lot larger.

Instead of defining the possible thresholds by the distance of the points to the plane, we propose to use a set of predefined thresholds. In our experiments, we used the following set:

$$\tau_z \in \left\{ \frac{(z_{max}-z_{min})}{2}, \frac{(z_{max}-z_{min})}{4}, \cdots, \frac{(z_{max}-z_{min})}{2^K} \right\} \tag{2.23}$$

where, z_{min} (resp. z_{max}) is the minimum (resp. maximum) disparity value and for an $M \times N$ disparity map, $2^K \geq 2 \cdot \max(M,N)$ and $2^{K-1} < 2 \cdot \max(M,N)$. Then, as in [Rabin et al., 2009], the threshold is chosen as the one minimizing the NFA value for a given group.

Choosing τ_z using the NFA, requires some adaptation in the algorithm as well as in the NFA computations.

First, it changes the values of the number of tests in Equations (2.1), (2.13) and (2.15) by multiplying its former value by K, the number of possible thresholds.

Then, in the model selection, since we compare several possible configurations, it has to be done under the same conditions. This means that when two groups G_1 and G_2 are tested for merging, a unique value of τ_z has to be used for the computation of $NFA_j(G_1,G_2)$ and $NFA_c(G_1 \bigcup G_2)$. The only case that concerns us here is when one of the groups is meaningful (otherwise, no comparison is necessary since the division is chosen). Since the final goal is to validate planes, we first reject the non-meaningful groups ($NFA > 1$) among G_1, G_2, and $G_1 \bigcup G_2$. Then for each meaningful group, the value of τ_z is computed and the smallest one is kept for the computation of $NFA_j(G_1,G_2)$ and $NFA_c(G_1 \bigcup G_2)$. This ensures the selection of the threshold best fitting to planes.

2.3.5 Possible improvements

The principal risk of split/merge heuristic, is that if a division is not made during the splitting step, the final segmentation will fail. Such situations can happen in the large scales where groups may be validated because the intersection of a plane and a surface may contain a large amount of points. In this case, the division into two groups may not be more meaningful especially if none of the groups are really a plane.

To avoid those local minima of the NFA in the division process, a solution is to force the division of large groups by roughly dividing the disparity map (for instance one can divide the disparity maps into blocks). Then for each group, the splitting process can be applied. The merging step is then applied as before on the set of all point groups and the result does not suffer from the initial over-segmentation (see Figure 2.4).

The interest of doing so is twofold:

- This rough division limits the risk of detecting intersections of planes with large surfaces since these surfaces are likely to be already divided.
- Since the division is binary, the division tree may be unbalanced. In this case, the division process is likely to take a lot of time. Dividing the splitting process, simplify the problem into simpler ones which speeds up the splitting step.

Another possible improvement is to add image information to this algorithm. For instance, one could use the line segments of the reference image (by using for instance [Grompone von Gioi et al., 2008]) to force group divisions. Indeed, the line segments might suggest separations between different objects in the image. Therefore they might be used to guide the divisions or to try other divisions than the ones given by the EM algorithm.

2.4 Experimental Results

We tried our algorithm on some of the disparity maps from the Middlebury database [Scharstein and Szeliski, 2002]. In each experiment, the precision threshold was chosen automatically as proposed in section 2.3.4. We obtained similar results as when we used a constant value of 1 pixel (which is the quantification step of the disparity maps). The results are given in Figure 3.6 and Table 2.1. The disparity maps of these data sets are sometimes only made of planes (Venus, Sawtooth) and sometimes made of more complex structures. Extremely irregular structures are rejected. If, however, structures are smooth enough, they can be locally approximated by planes up to the given precision and this is the answer of our algorithm. Note that further extensions of our model selection criterion are possible, which should distinguish for instance quadrics from planes.

The piecewise planar approximation is not too simplistic as shown by the error maps given in Figure 3.6 (e) and the error measurements of Table 2.1. The remaining error after projection on the various planes seems to be mostly due to the quantification step of the disparity maps. This explains the oscillations of the errors. Each different period of oscillation corresponds to a different plane. Had the planes been badly estimated, or some of the points been associated to a wrong plane, more errors would be visible (see Figure 2.5). On the another hand, the obtained classifications do not seem to be over-segmented. The various periods of oscillations seem to correspond to one plane most of the time.

Another result is that when the precision parameter is set manually, the method stays robust for different values of precisions and gives a planar approximation at different scales

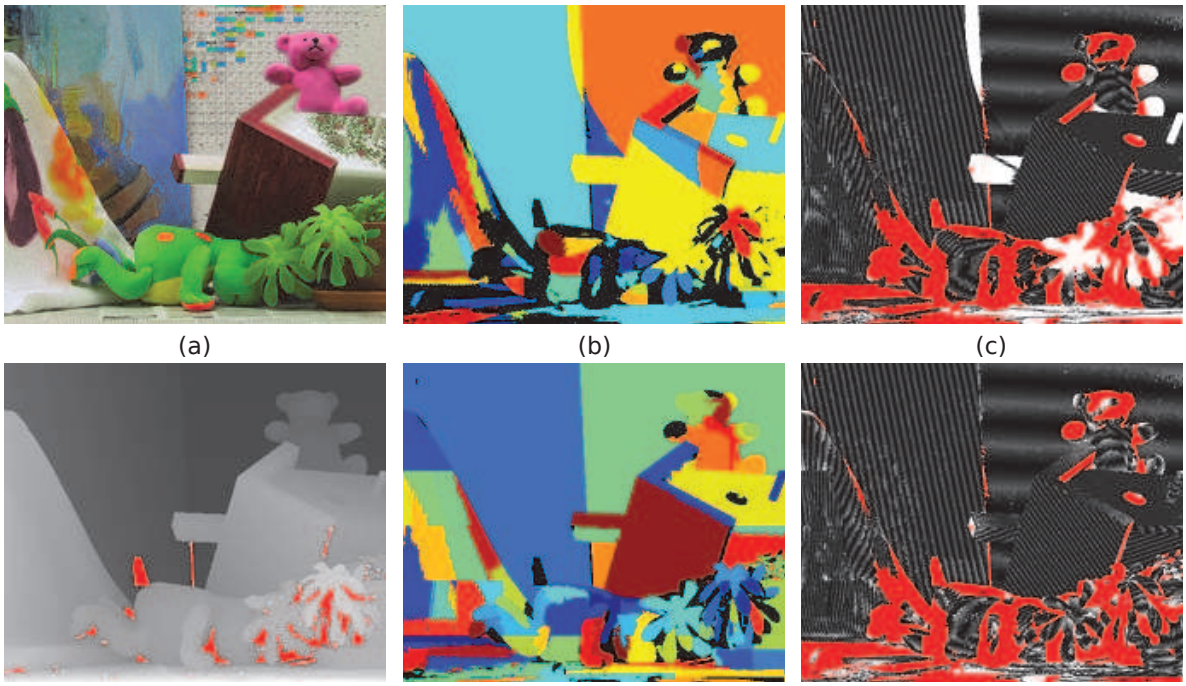


Figure 2.5: Failure case of the original algorithm (Teddy from the Middlebury dataset). (a) reference image. (d) ground truth. (b) and (e) segmentation computed with our algorithm (black parts are non meaningful groups). (b) original version of our algorithm: all the yellow parts on the right are considered as a single meaningful group which is of course false. (e) algorithm with improvements from Section 2.3.5: the groups are well segmented this time. (c) and (f) error maps of the projection on the planes of (b) and (e) (dark = low error, bright = large error). The red parts are non meaningful groups in either one of the two possible segmentations. In (c), the error is large because of the yellow group on the right. In (f), the only remaining error is the one due to the quantification of the data.

(see Figure 2.6). This is usually not the case of RANSAC based approaches which tend to fail when the precision is not optimal.

We also tried our algorithm on a simulated stereo pair of Toulouse's St-Michel Jail. For this experiment, a subpixel precision is expected. As before the threshold parameter was estimated automatically. Two tests were performed:

- We added an additional Gaussian noise of variance 0.02 pixels to the ground truth disparity. The results are given in Figure 2.4 and Table 2.1, column Toulouse. For this experiment, most of the planes seem to have been detected. The error was significantly reduced (Figure 2.4 (f)) and is mostly localized around the edges (RMSE = 0.007 with edges). However, the planes were estimated with a great precision (RMSE = 0.005 pixels without the edges).

- The algorithm in [Sabater, 2009] was used on a simulated low-baseline stereo pair to obtain a disparity map with precision 0.024 pixels. After grouping with our algorithm and re-projection of the measured disparity data on the detected planes we observe that the RMSE is reduced to 0.021 pixels (see Table 1, column Toulouse2) which proves the correctness of our planar approximation.

Figure 2.6: Classification obtained with different precisions. (a) precision=1 pixel, 85 planes detected; (b) precision=2 pixels, 44 planes detected; (c) precision=5 pixels, 23 planes detected.

	mixed		planar		non-planar		
	Toulouse	Toulouse2	Venus	Sawtooth	Cones	Teddy	Books
Initial variance(τ_z)	0.02	0.024	1	1	1	1	1
RMSE	0.005	0.021	0.29	0.29	0.95	1.15	1.01
Error$\geqslant \tau_z$ (%)	1.8	-	0	0	3.9	6	3.4
Error$\geqslant 2\tau_z$ (%)	0.7	-	0	0	0.8	1.9	0.9

Table 2.1: Error measurements. First line initial: precision of the ground truth. Second line: RMSE (ℓ_2 error). Third and fourth lines: percentage of outliers.

2.5 Conclusion

We presented an algorithm for optimally grouping 3D point clouds into planar patches. Inspired from computational gestalt theory [Desolneux et al., 2008b], it allows the use of simple grouping laws to robustly detect simple patterns (planar patches here), and to apply later these laws recursively (for instance symmetry or similarity of planar patches) in order to obtain more complex structures, like those proposed in Lafarge's dictionary [Lafarge et al., 2008b], without making an *a priori* explicit list of all possibilities. As opposed to the method proposed in [Igual et al., 2007; Facciolo and Caselles, 2009], our algorithm does not rely on an initial segmentation which can be error prone. The various parameters can be easily set which makes it almost automatic. The ε value can be set to 1 due to its statistical meaning and the distance threshold τ_z used to reject outliers can be estimated as the one minimizing the NFA.

Our experiments show that the proposed approach is capable of detecting a reasonable piecewise affine decomposition even in complex scenes (as opposed to RANSAC based approaches). Moreover, the corresponding regularization reduces the error of punctual disparity measures.

Several applications and improvements are thought of. The piecewise planar grouping can be used as a basis for interpolation and vectorization algorithms. However, these applications will require a stronger use of luminance (as a post processing refinement of the boundaries between several planes).

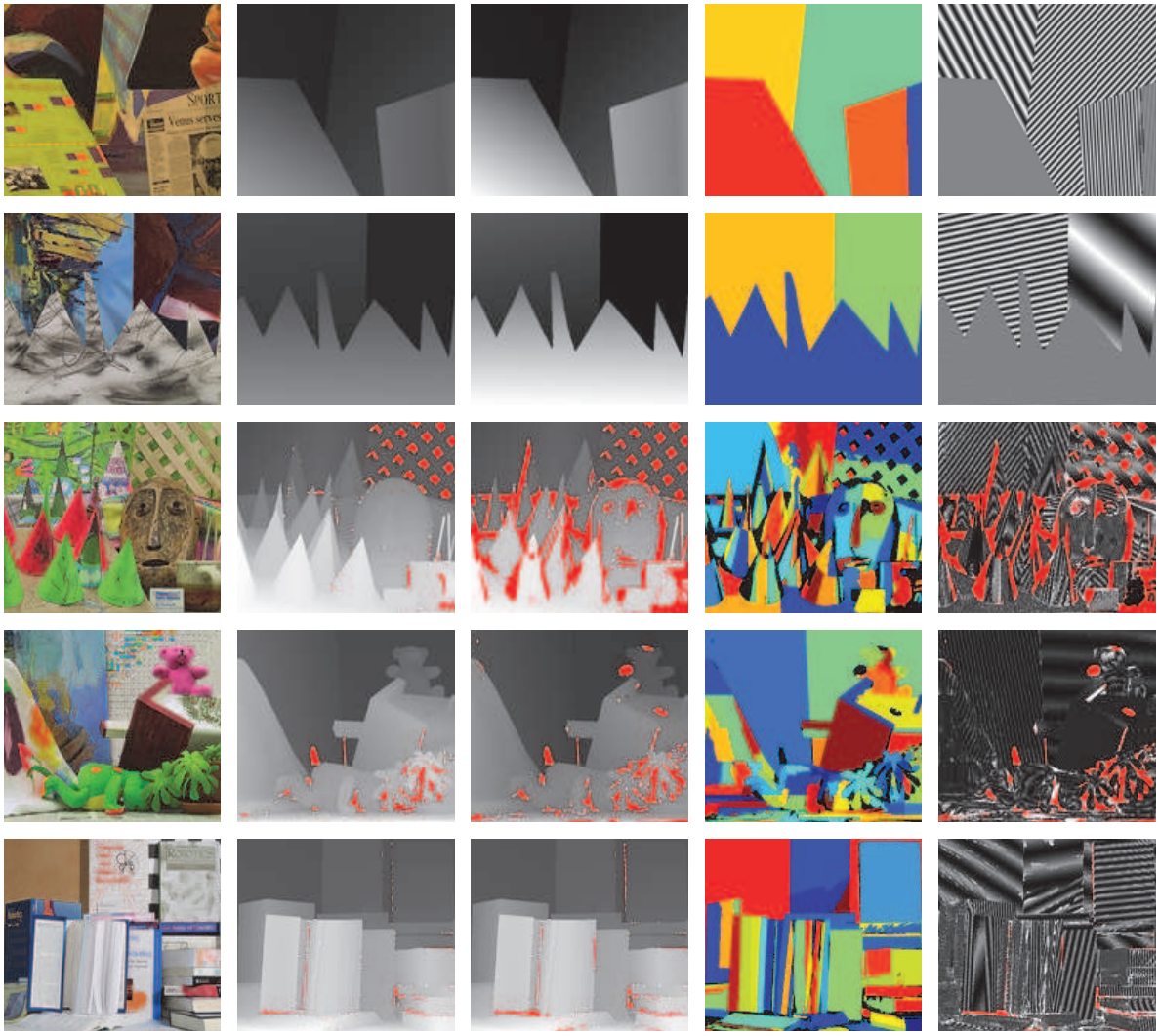


Figure 2.7: Results obtained with Middlebury's ground truth datasets with the automatic threshold parameter. From top to bottom, Venus, Sawtooth, Cones, Teddy and Books. (a) Reference Image, (b) ground truth (unknown parts are shown in red), (c) disparity map obtained after projection on the planes that were found with our algorithm (parts not referred as planar are shown in red), (d) planar classification of the points (each color refer to one plane), (e) residual error after re-projection on the planes. The errors are mostly due to the quantification of the disparity map (which explains the oscillations).

Chapter 3

Fast plane computation

Contents

Résumé: Dans ce chapitre, nous proposons une nouvelle méthode pour la détection de facettes planes dans les cartes de disparités. Cette approche est similaire à celle introduite dans [Grompone von Gioi et al., 2008] pour la détection rapide de segments dans les images. Dans un premier temps, nous utilisons une approche par croissance de régions à partir de graines aléatoires. Puis, le critère de décisions introduit dans le Chapitre 2 est utilisé pour garder les patchs plans. Le principal avantage de notre méthode en comparaison à la littérature est sa capacité à pouvoir estimer les paramètres critiques, ceci la rendant quasi automatique. Cet technique est particulièrement adaptée au cas de la reconstruction 3D en milieu urbain à partir de paires stéréo à faible écart entre les vues où un modèle plan-par-morceaux peut-être appliqué.

Abstract: In this chapter, we propose a new method for fast detection of planar patches in disparity maps. This approach is similar to the one introduced in [Grompone von Gioi et al., 2008] for fast line segment detection in images. We first use a region growing algorithm on random seeds. Then, the parameter-free criterion introduced in Chapter 2 is used to keep only the patches that are planar. The main advantage of our method is to be able to estimate a critical parameter. This method is specially well suited to 3D reconstruction of urban environments from low-baseline aerial or satellite stereo pairs where a piecewise-planar model can be applied.

3.1 Introduction

In the previous Chapter, an *a contrario* criterion was defined which allowed us to make some capital decisions in model detection algorithm:

- When can a group of points from a disparity map considered as a planar patch?
- What is the best point configuration in terms of planarity between two groups?
- What is the best choice for outlier rejection threshold?
- For a given point group, what model best explains the data between one plane or two planes?

A split and merge algorithm which decisions were based on this *a contrario* framework was then defined. However, this procedure requires a lot of computation resources especially during the merging phase where a priority queue has to be updated at each merge. In this Chapter, we propose a fast algorithm for plane detection in disparity maps. Once again, the *a contrario* framework will prove itself very useful especially to decide what parameter should be used.

Various methods have been proposed to achieve a fast piecewise planar segmentation of a 3D model.

In [Jiang and Bunke, 1994] a line-based and column based split and merge approach is proposed for a fast segmentation. The first segmentation for each line and column is based on the splitting into two groups as long as a precision threshold is not respected. Then a region growing approach is used for to merge the possible group. However, their approach do not seem very precise near the planes separation.

Applying RANSAC sequentially by removing the detected groups from the data has been a popular approach for fast plane detection. Initially used for multi-plane detection in unconstrained stereo-vision [Vincent and Laganière, 2001] and [Kanazawa and Kawakami, 2004], it was proved to be unadapted as itself in [Toldo and Fusiello, 2008] and [Stewart, 1995] where the detection of "phantom" planes (mixture of several models) was observed. Indeed, the RANSAC algorithm was originally designed to fit single models to points corrupted by outliers and fails when the outliers are structured because of other models.

To overcome this, an *a contrario* framework is proposed in [Rabin et al., 2009] and the possible division of RANSAC-validated groups is tested. However, this approach seems notwell adapted to disparity maps where the amount of points is considerably larger and do not contain outliers. In [Schnabel et al., 2007b] and [Labatut et al., 09], the authors adapted sequential RANSAC to 3D point cloud by introducing strong local constraints on the validation. In their procedure, only connected groups (according to a $k-$ nearest neighbor) can be validated and local plane orientation at each point has to be similar to the global orientation of the validated group. More parameters have to be finely tuned which goes against the automatic detection we want to achieve.

Another popular approach for piecewise planar segmentation is region growing. This approach, which is the base of the algorithm proposed here, has been successfully adapted to both range image segmentation (see [Besl and Jain, 1988] and [Poullis and You, 2009] for instance) and to 3D point clouds (see [Chauve et al., 2010]). However, in all these papers, no automatic criterion is proposed to set the various thresholds.

The aim of this chapter is to develop a fast method to obtain a piecewise planar description of a 3D point cloud. Based on region growing, our approach provides a faster search of planar regions for similar error values than the methods described in Chapter 2 and [Labatut et al., 09] both combined with the decision criterion of Chapter 2 as illustrated in Fig. 3.1. Our second objective is to propose ways to automatically set the common parameters to both region growing and RANSAC methods. This is the main advantage of our method since some parameters such as the outlier rejection threshold are critical to obtain good results.

This chapter is organized as follow. First a global description of the algorithm is given in section 3.2. Then more precisions on the steps of the algorithm and their mathematical justication are given in section 3.3, 3.4 and 3.5. At last, some experimental results are given in section 3.6.

3.2 Global description

In [Grompone von Gioi et al., 2008], the authors made a breakthrough in line segment detection by proposing a linear time algorithm with a false detection control. The algorithm they propose is divided into three steps: (i) find line support regions using a greedy approach, (ii) approximate regions by a rectangle, (iii) validate or not a line segment found by Desolneux *et al.*'s theory [Desolneux et al., 2008b].

The algorithm we propose here is based on an adaptation of step (i) and (iii) to the case of planar region detection. A description of our version of step (i) is given in this section and an explanation of how to set the parameters is given in the next section. Step (iii) can be easily solved by the parameter-free decision criterion of Chapter 2 and is therefore not explained in this Chapter. The optimization step (ii) is irrelevant in our case since planar regions may have any arbitrary shape, not necessarily rectangular.

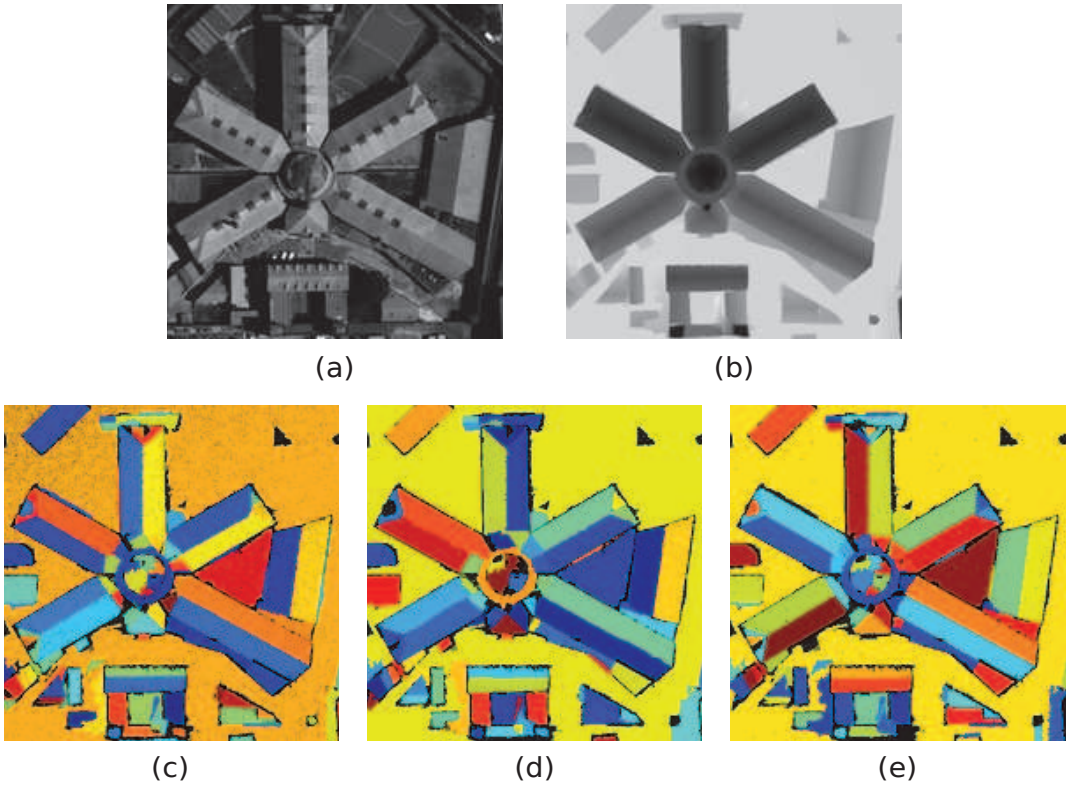


Figure 3.1: Comparison of [Labatut et al., 09], method of Chapter 2 and the proposed algorithm on St-Michel disparity map with a constant precision parameter (size: 512×512 pixels, initial RMSE = 0.02 pixels). *(a) Reference image. (b) Disparity map. Bottom row: Planar patch classification. Each color corresponds to a different plane. The black parts are defined as non planar according to the validation theory of Chapter 2. (c) [Labatut et al., 09] computation time 50 s, 103 planes found, 91.2% points, RMSE = 0.0121 pixels. (d) Method of Chapter 2, computation time 4 m 02 s, 89 planes found, 92.6% points, RMSE = 0.0062 pixels. (e) Our algorithm, computation time 9 s, 92 planes found, 92.2% points, RMSE = 0.0066 pixels.*

3.2.1 Region growing

Let us first introduce some notations. A disparity map is considered here as a mapping z from a discrete grid $\Omega \subset \mathbb{Z}^2$ to $\mathbb{R}$. Each point $\mathbf{x} = (x, y) \in \Omega$ has a unique value $z_{\mathbf{x}} \equiv z(\mathbf{x})$ and is therefore associated to a unique 3D point. We also define a patch of size s as a 2D square neighborhood of a point $\mathbf{x} = (x, y)$, $P_s(\mathbf{x}) = \{\mathbf{x}' = (x', y') \in \Omega, |x - x'| \leqslant s, |y - y'| \leqslant s\}$.

Our algorithm is based on region growing from a point. A region starts from a patch $P_s(\mathbf{x})$ centered at a given point $\mathbf{x}$. From the local neighborhood $P_s(\mathbf{x})$, a first (imprecise) estimation of the plane passing through $\mathbf{x}$ can be computed (more details on the plane computation and a fast scheme to do it are given in section 3.5. Then, pixels connected[1] to the region are tested: if their z-distance to the plane is less than a given distance threshold τ_z, they are added to the region. Each time the size of the region doubles, the plane is re-estimated for a more precise estimation. The growing stops whenever none of the pixels neighboring the region verify the distance precision constraint. An illustration of the previously described region growing step is given in Fig. 3.2.

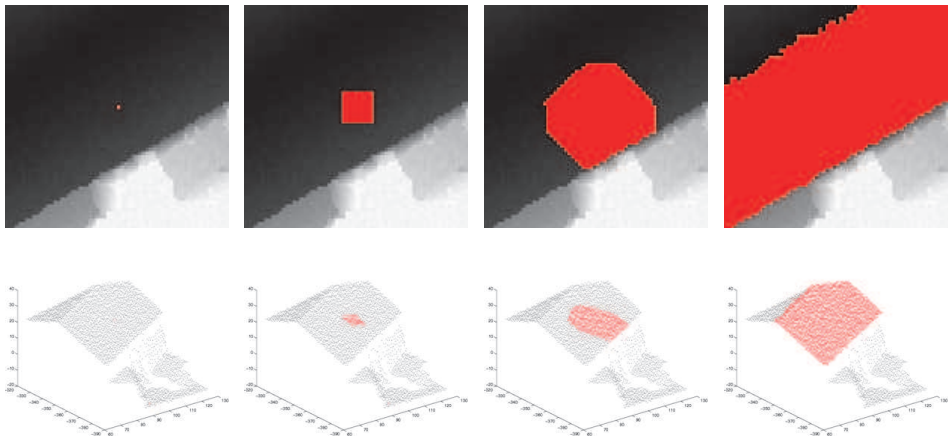


Figure 3.2: Region growing. From an initial point, the points are added progressively if the distance between the real disparity and the projection on the estimated plane is less than a threshold τ_z. *Top row: disparity map. Bottom row: corresponding 3D points.*

Pixels within the detected region are marked and cannot be used again. Then, the local planes of the remaining points are re-estimated omitting marked points which avoid mixing planes in the local estimation. Another region then starts growing from a new point and so on until the disparity map is entirely marked. The region growing algorithm is summarized with algorithm 3.

3.2.2 Starting points

The choice of the initial point is critical in the region growing approach. The local plane estimated from a point $\mathbf{x}_0$ standing on the edge of two planes will be a mixture of those two planes. The initial plane estimation will give a poor description of the local neighborhood of $\mathbf{x}_0$ and the region growing is likely to fail (either by growing on unwanted neighbor points or by stopping right away). The starting region must then be chosen carefully.

[1] 4-connected pixel neighborhood is used here

Algorithm 3: Region Growing

Data:
$\mathbf{x_0} \in \Omega$, starting point
$\{G_1, \cdots, G_n\}$, list of already validated planes
τ_z, distance threshold.
Result:
$G_{\mathbf{x}0}$, the group of all the points on the same plane as $\mathbf{x_0}$

begin
  $G_{\mathbf{x_0}} = P_s(\mathbf{x}_0) \setminus G_1 \cup \cdots \cup G_n$
  $\pi_0 \leftarrow$ compute plane of $G_{\mathbf{x}_0}$
  $N_0 = \#G_{\mathbf{x_0}}$
  $G_{candidates} = 4-\text{connected}(G_{\mathbf{x_0}}) \setminus (G_{\mathbf{x_0}} \cup G_1 \cup \cdots \cup G_N)$
  $G_{explored} = G_{\mathbf{x_0}}$
  foreach $\mathbf{x} \in G_{candidates}$ **do**
    if $\mathbf{x} \notin G_{explored}$ & $dist(\mathbf{x}, \pi_0) \leq \tau_z$ **then**
      add $\mathbf{x}$ to $G_{\mathbf{x_0}}$
      add $4-\text{connected}(\mathbf{x}) \setminus (G_{\mathbf{x_0}} \cup G_1 \cup \cdots \cup G_N)$ to $G_{candidates}$
    add $\mathbf{x}$ to $G_{explored}$
    if $\#G_{\mathbf{x_0}=2N_0}$ **then**
      $\pi_0 \leftarrow$ compute plane of $G_{\mathbf{x}_0}$
      $N_0 = \#G_{\mathbf{x_0}}$
  end
end

To avoid this kind of situation, we propose to sort the patches according to their "flatness". Such is done by computing the mean square error (MSE) between the original $z-$value of each point and the estimated one after projection on the local plane:

$$MSE(P_s(\mathbf{x})) = \sum_{\mathbf{x}' \in P_s(\mathbf{x})} \frac{\left(z(\mathbf{x}') - z_{\pi_\mathbf{x}}(\mathbf{x}')\right)^2}{\#P_s(\mathbf{x}) - 3} \tag{3.1}$$

where $z_{\pi_\mathbf{x}}(\mathbf{x}')$ is the z-value of plane $\pi_\mathbf{x}$ at point $\mathbf{x}'$, $\#\cdot$ is the cardinal operator and $\#P_s(\mathbf{x}) - 3 > 0$. If we suppose that the $z-$values are distributed according to a piecewise-planar model plus an *i.i.e.* Gaussian additive noise, then it can be proved that the expectation of $MSE(P_s(\mathbf{x}))$ is minimal if all the z-values of the points of $P_s(\mathbf{x})$ were distributed from a single plane (see section 3.4). Points with lowest MSE are then more likely to be distributed from a single planar model. The sorting step then favors patches for which the local plane description corresponds best to the real planar model. Other points will appear later in the starting point queue and may never be used to grow a region.

3.2.3 Remarks on the algorithm

- The other heuristic proposed in Chapter 2 for piecewise-planar segmentation had a limited number of parameters which for the most could be set automatically. In section 3.3 we propose a way to automatically set the most critical parameter of our method: the distance rejection threshold τ_z.
- The local estimation combined with the region growing procedure ensures that each region detected is made of connected points. This constraint on the result is in fact a

good thing since this is what happens in practice. Indeed, two disjoint regions in an image describes most of the time two distinct objects.

- The region growing threshold τ_z also represents the maximal error difference between the original (and possibly noisy) $z-$value of a point and its projection on the plane it is associated to after the piecewise-planar segmentation.

3.3 Point precision and rejection threshold estimation

The region growing rejection threshold parameter τ_z which is necessary for our region growing, is common to most of the methods presented before. In RANSAC-based algorithms, it gets critical because the plane selection process depends on the number of inliers according to that precision. A wrong rejection parameter may lead to the validation of planes which do not correspond at all to the data.

This parameter is usually supposed to be known, but this is actually not the case. To our knowledge, only [Sabater, 2009] gives an estimation of the expected precision when they compute disparities. However, the result is an inferior bound and is imprecise in case of adhesion (see Chapter 5) which excludes using it to reject outliers.

Using our approach, an estimation of the distance rejection parameter is however possible. This estimation is based on an iterative computation of the residual noise of the validated planar groups by the mean of the previously defined MSE. For all that follows we supposed that the disparity at each point can be described by a piecewise-planar model and an additive Gaussian noise (due to the computation method).

3.3.1 Estimation from validated groups

Let's first consider the following situation that happens while using the region growing algorithm. M planar groups have already been found with the previous steps of the algorithm (we use the validation criterion of Chapter 2 to decide whether a group can be considered as planar). We now want to find the $M+1^{th}$ group using the same procedure as before.

For each group validated as planar, one can estimate the mean square residual noise by re-projecting the points on the associated plane. Such is done with the MSE definition given by Equation (3.1). The larger the group is, the more accurate this noise estimation is.

The residual noise estimation can be computed iteratively. By noting $(MSE_i)_{i=1..M}$ the MSE of each validated group and $(N_i)_{i=1..M}$ their respective size, one can estimate a global MSE:

$$MSE_{glob} = \frac{\sum_{i=1}^{N} MSE_i \cdot (N_i - 3)}{-3 + \sum_{i=1}^{N} N_i} \tag{3.2}$$

Now, let's consider the point group G subject to the region growing procedure and let's make the following assumptions:

- All the points from G can be explained by a single plane π with an additive Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$ along the z coordination.

- The estimated plane parameters are the real one.

Then using classical results of Gaussian statistics we can state that for any point $\mathbf{x} \in \Omega$ following the same model (plane π + Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$):

$$\begin{aligned} P(|z(\mathbf{x}) - z_\pi(\mathbf{x})| < \tau_z) &= P(-\tau_z \leq \varepsilon(\mathbf{x}) \leq \tau_z) \\ &= \text{erf}(\tfrac{\tau_z}{\sqrt{2}\sigma}) \end{aligned}$$

where erf is the error function. Then to ensure that 95% of the points following the right model are validated, one can choose $\tau_z = 2 \cdot \sigma$. In practice, since σ is unknown, we use the unbiased estimation of σ given by $\hat{\sigma} = \sqrt{MSE_{glob}}$.

At last, the rejection threshold can be refined by estimating it progressively with both the global MSE and the MSE of the patch to overcome the fact that the precision may not be the same everywhere:

$$\tau_z = 2 \cdot \frac{\sqrt{MSE_{glob} \cdot (N_{glob} - 3) + MSE_{patch} \cdot (N_{patch} - 3)}}{\sqrt{N_{glob} + N_{patch} - 3}} \tag{3.3}$$

3.3.2 First estimation

Algorithm 4: Distance threshold initialization

Data:
Seeds $= \{\mathbf{x_0}, \cdots, \mathbf{x_n}\} \in \Omega^n$
$\Theta = \{\tau_1, \cdots, \tau_p\}$, possible threshold values
Result:
τ_z, the best distance threshold

```
begin
    NFA_global = ∞
    τ_z = ∞
    foreach x in Seeds do
        NFA_min = ∞
        foreach τ ∈ Θ do
            G(x, τ) ← region growing of x for threshold τ with Algorithm 3
            nfa ← compute NFA of (G, τ)
            if nfa < NFA_min then
                NFA_min = nfa
                if NFA_min < NFA_global then
                    NFA_global = NFA_min
                    τ_z = τ
        end
    end
end
```

In the previous paragraph, we proposed a way to set the rejection threshold parameter knowing at least one planar group of points. However, the question still remains on hold when the region growing algorithm starts and no group has been validated.

For this step, we chose to set the parameter empirically by trying out several possible initialisation. For a given starting point, we try several rejection threshold which all give a different final group after region growing. The *a contrario* criterion described in Chapter 2

then allows to decide which threshold was the best for this particular starting point. This part is then exactly the same as what was done in Chapter 2 to set the rejection threshold. It can also be seen once again as an adaptation of what is done in [Moisan and Stival, 2004] and [Rabin et al., 2009] to larger dataset.

Without any prior information, we can only assume that the precision on along the z axis is at most equal to the resolution along the x or y axis. We therefore choose the following set of rejection threshold:

$$\Theta = \left\{ (z_{max} - z_{min}), \frac{(z_{max} - z_{min})}{2}, \ldots, \frac{(z_{max} - z_{min})}{2^n} \right\}$$

with $M < 2^n \leq 2M$ for an $M \times M$ disparity map.

To make sure we find one of the largest planes in the disparity map, we propose to try several initial points. In practice, we try the 10 initial points for which $MSE(P_s(\mathbf{x}))$ is minimal. As demonstrated in the next section, this order is justified by the fact that $MSE(P_s(\mathbf{x}))$ is minimal (in expectation) if all the points of $P^s_{\mathbf{x}}$ are distributed according to a single planar model. At last, the final initial configuration is once again the one for which the NFA is minimal. The algorithm for the distance threshold initialization is given in Algorithm 4.

As shown in Figure 3.3, the residual noise between the ground truth and the input data is approximately the same as the residual noise between the input data and our piecewise planar approximation. This suggest that both the classification and the precision estimation are good since a wrong classification would change the characteristics of the residual noise. This result was obtained both with synthetic and real data.

3.4 Justification of the sorting step

In the global description of the algorithm, we proposed to sort the initial points so that the one used first for region growing are the one for which the local neighborhood is best describe by a single plane. We propose here to give a mathematical explanation of this step.

The following theorem gives a justification of the sorting step the data are described by a reduced number of planes:

Theorem 1 *Let's consider a dense disparity map described as a 2D finite grid $\Omega \subset \mathbb{Z}^2$ and z values on this grid distributed according to a finite set of planes $(\pi_i)_{i=1..n_p}$, $n_p \ll \#\Omega$ and corrupted by an i.i.d. additive Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$:*

$$\forall \mathbf{x} \in \Omega,\ \exists i,\ 1 \leq i \leq n_p, \nearrow z(\mathbf{x}) = z_{\pi_i}(\mathbf{x}) + \varepsilon(\mathbf{x}) \tag{3.4}$$

If we note $\hat{\pi}_{s,\mathbf{x}}$ the least squares estimate of the plane from a local neighborhood of $\mathbf{x}$, $P_s(\mathbf{x})$, then the expectation of the residual error is the sum of two terms:

$$\mathbb{E}\left[\sum_{\mathbf{x}' \in P_s(\mathbf{x})} (z_{\hat{\pi}_{s,\mathbf{x}}}(\mathbf{x}') - z(\mathbf{x}'))^2 \right] = \lambda(s)\sigma^2 + B(\mathbf{x}) \tag{3.5}$$

where $\lambda(s)$ only depends on the local neighborhood size $\#P_s(\mathbf{x})$ and $B(\mathbf{x})$ is a bias term depending on the model describing the z distribution within $P_s(\mathbf{x})$. Moreover, the bias term is null if the local model is made of a single plane.

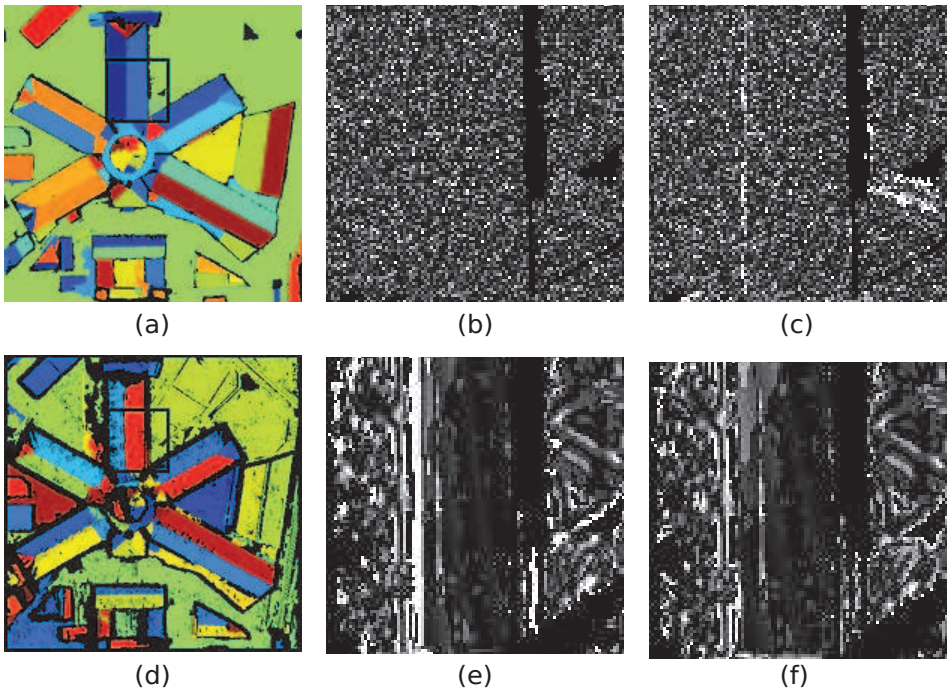


Figure 3.3: Automatic error estimation (black parts are either non-planar regions or points with an unknown disparity). From left to right: classification obtained after our algorithm, residual noise between the initial data and the ground truth in the black square zone, residual noise between the initial data and our piecewise-planar approximation in the black square zone. The residual noise looks the same. The noise estimation is therefore precise in the planar regions. *Top row: data = ground truth + Gaussian white noise. Ground truth residual noise:* 0.0199 *pixels. Piecewise-planar residual noise:* 0.0212 *pixels. Bottom row: real data obtained using [Sabater, 2009]. Ground truth residual noise:* 0.0306 *pixels. Piecewise-planar residual noise:* 0.0214 *pixels.*

Proof The demonstration follows the similar steps as the ones given to estimate the re-projection error of a plane that is given in Appendix A.

Let 's first introduce some notations. For any 2D point $\mathbf{x} = (x, y) \in \Omega$, we note $X = (x, y, 1)$ the corresponding point in homogeneous coordinate. We note μ the parameter defining a projection onto a plane of a 2D point. For a given set of points $P_s(\mathbf{x})$, its plane parameters are computed as the one minimizing the ℓ_2 error of the disparity values:

$$\begin{aligned} \hat{\mu} &= \arg\min_{\mu=(a,b,c)} \sum_{\mathbf{x}' \in P_s(\mathbf{x})} \big(z(\mathbf{x}') + (a \cdot x' + b \cdot y' + c)\big)^2 \\ &= \arg\min_{\mu} \sum_{\mathbf{x}' \in P_s(\mathbf{x})} \big(z(\mathbf{x}') + X'^T \mu\big)^2 \\ &= \big(\mathbf{A^T A}\big)^{-1} \mathbf{A^T b} \end{aligned} \tag{3.6}$$

with,

$$\mathbf{A} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} \text{ with } (\mathbf{x}_j)_{i=1..n} = P_s(\mathbf{x}),\ \mathbf{b} = \begin{pmatrix} z_(\mathbf{x_1}) \\ z_(\mathbf{x_2}) \\ \vdots \\ z_(\mathbf{x_n}) \end{pmatrix}, \mu = -\begin{pmatrix} a \\ b \\ c \end{pmatrix} \text{ and } \mathcal{E} = \begin{pmatrix} \varepsilon(\mathbf{x}_1) \\ \varepsilon(\mathbf{x}_2) \\ \vdots \\ \varepsilon(\mathbf{x}_n) \end{pmatrix}$$

Let's first demonstrate Theorem 1 for a point $\mathbf{x}$ for which the local neighborhood is modelled by two planes. $\mathbf{A}$, $\mathbf{b}$ and $\mathcal{E}$ can be divided into two blocks, each corresponding to one of the two plane models:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A_1} \\ \mathbf{A_2} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{pmatrix} \text{ and } \mathcal{E} = \begin{pmatrix} \mathcal{E}_\mathbf{1} \\ \mathcal{E}_\mathbf{2} \end{pmatrix}$$

such that

$$\begin{aligned} \mathbf{b_1} &= \mathbf{A_1}\mu_1 + \mathcal{E}_\mathbf{1} \\ \mathbf{b_2} &= \mathbf{A_2}\mu_2 + \mathcal{E}_\mathbf{2} \end{aligned}$$

We can now express $\hat{\mu}$ as a function of μ_1 and $\Delta\mu = \mu_1 - \mu_2$:

$$\begin{aligned} \hat{\mu} &= \big(\mathbf{A^T A}\big)^{-1} \mathbf{A^T b} \\ &= \big(\mathbf{A^T A}\big)^{-1} \mathbf{A^T} \left(\begin{pmatrix} \mathbf{A_1}\mu_1 \\ \mathbf{A_2}\mu_2 \end{pmatrix} + \mathcal{E} \right) \\ &= \big(\mathbf{A^T A}\big)^{-1} \mathbf{A^T} \left(\mathbf{A}\mu_1 + \begin{pmatrix} O \\ \mathbf{A_2}\Delta_\mu \end{pmatrix} + \mathcal{E} \right) \\ &= \mu_1 + \big(\mathbf{A^T A}\big)^{-1} \mathbf{A_2^T A_2} \Delta\mu + \big(\mathbf{A^T A}\big)^{-1} \mathbf{A^T} \mathcal{E} \end{aligned} \tag{3.7}$$

Then, for any point $\mathbf{x}_1 \in P_s(\mathbf{x})$ described by model μ_1, if we note $X_1 = (x_1, y_1, 1)$, the

expectation of its residual square error (SE) is:

$$\begin{aligned}
\mathbb{E}\big[SE(\mathbf{x_1})\big] &= \mathbb{E}\big[(z(\mathbf{x_1}) - X_1^T\hat{\mu})^2\big] \\
&= \mathbb{E}\big[(X_1^T(\mu_1 - \hat{\mu}) + \varepsilon(\mathbf{x_1}))^2\big] \\
&= \mathbb{E}\big[\big(X_1^T\big((\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}\Delta\mu + (\mathbf{A^T A})^{-1}\mathbf{A^T}\mathcal{E}\big) + \varepsilon(\mathbf{x_1})\big)^2\big] \\
&= X_1^T(\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}\Delta\mu\Delta\mu^T\mathbf{A_2^T A_2}(\mathbf{A^T A})^{-1}X_1 \\
&\quad + 2X_1^T(\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}\Delta\mu\underbrace{\mathbb{E}\big[\mathcal{E}^T\big]}_{\mathbf{0}}\mathbf{A}(\mathbf{A^T A})^{-1}X_1 \\
&\quad + 2X_1^T(\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}\Delta\mu\underbrace{\mathbb{E}\big[\varepsilon(\mathbf{x_1})\big]}_{0} \\
&\quad + 2X_1^T(\mathbf{A^T A})^{-1}\underbrace{\mathbf{A^T}\mathbb{E}\big[\mathcal{E}\varepsilon(\mathbf{x_1})\big]}_{\sigma^2 X_1} + \underbrace{\mathbb{E}\big[\varepsilon(\mathbf{x_1})^2\big]}_{\sigma^2} \\
&\quad + X_1^T(\mathbf{A^T A})^{-1}\mathbf{A}\underbrace{\mathbb{E}\big[\mathcal{E}\mathcal{E}^T\big]}_{\sigma^2 I_n}\mathbf{A}(\mathbf{A^T A})^{-1}X_1
\end{aligned} \tag{3.8}$$

Simplifying the different terms and using clearer notations at last gives:

$$\mathbb{E}\big[SE(\mathbf{x_1})\big] = \sigma^2\big(1 + 3X_1^T(\mathbf{A^T A})^{-1}X_1\big) + X_1^T\mathbf{B_1}X_1 \tag{3.9}$$

with $\mathbf{B_1} = (\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}\Delta\mu\Delta\mu^T\mathbf{A_2^T A_2}(\mathbf{A^T A})^{-1}$.

An analogous result is obtained for any point $\mathbf{x}_2 \in P_s(\mathbf{x})$ described by model μ_2:

$$\mathbb{E}\big[SE(\mathbf{x_2})\big] = \sigma^2\big(1 + 3X_2^T(\mathbf{A^T A})^{-1}X_2\big) + X_2^T\mathbf{B_2}X_2 \tag{3.10}$$

with $\mathbf{B_2} = (\mathbf{A^T A})^{-1}\mathbf{A_1^T A_1}\Delta\mu\Delta\mu^T\mathbf{A_1^T A_1}(\mathbf{A^T A})^{-1}$.

At last, using Eq. 3.9 and Eq. 3.10 to sum over all the points $\mathbf{x}' \in P_s(\mathbf{x})$ we obtain the sum of square errors $\mathbb{E}[SE(P_s(\mathbf{x})]$:

$$\begin{aligned}
\mathbb{E}\left[SE(P_s(\mathbf{x}))\right] &= \sigma^2(n + 3\text{trace}(\mathbf{A}(\mathbf{A^T A})^{-\mathbf{T}}\mathbf{A^T})) + \text{trace}(\mathbf{A_1 B_1 A_1^T}) + \text{trace}(\mathbf{A_2 B_2 A_2^T}) \\
&= \sigma^2(n + 6) + B(\mathbf{x})
\end{aligned} \tag{3.11}$$

The simplification of $\text{trace}(\mathbf{A}(\mathbf{A^T A})^{-\mathbf{T}}\mathbf{A^T})$ into $\text{rank}(\mathbf{A}) = 2$ is part of the proof of proposition 4 in Appendix A. At last, noting that $\mathbf{B_1} = \mathbf{B_2} = 0$ when $\Delta\mu = 0$ (only one model), and that $\big(\text{trace}(\mathbf{A_1 B_1 A_1^T}) + \text{trace}(\mathbf{A_2 B_2 A_2^T})\big) >= 0$ completes the proof.

Let's now prove Theorem 1 in the general case considering more than two planes. Let's consider a point $\mathbf{x}$ for which $P_s(\mathbf{x})$ is modelled by N planes. We use the following notation for $i, j = 1..N$, $\Delta\mu_{i,j} = \mu_i - \mu_j$. Eq. 3.7 then becomes:

$$\mu = \mu_i + \big(\mathbf{A^T A}\big)^{-1}\mathbf{A^T}\mathcal{E} + \sum_{\substack{j\,=\,1..N \\ j\,\neq\,i}} \big(\mathbf{A^T A}\big)^{-1}\mathbf{A_j^T A_j}\Delta\mu_{i,j} \tag{3.12}$$

This means that Eq. 3.9 can be re-written with a similar form for any $\mathbf{x_i}$ following the model μ_i with $i = 1..N$:

$$\mathbb{E}\big(SE(\mathbf{x_i})\big) = \sigma^2\big(1 + 3X_i^T(\mathbf{A^T A})^{-1}X_i\big) + X_i^T\mathbf{B'_i}X_i \tag{3.13}$$

with this time $\mathbf{B'_i} = \big(\sum_{\substack{j=1..N \\ j\neq i}} (\mathbf{A^T A})^{-1}(\mathbf{A_j^T A_j})\Delta\mu_{i,j}\big)\big(\sum_{\substack{j=1..N \\ j\neq i}} (\mathbf{A^T A})^{-1}(\mathbf{A_j^T A_j})\Delta\mu_{i,j}\big)^T$.

At last, summing the residual error over all the points as done in Eq. 3.11 finally gives:

$$\begin{aligned}\mathbb{E}\left[SE(P_s(\mathbf{x}))\right] &= \sigma^2(n + 3\text{trace}(\mathbf{A}(\mathbf{A^T A})^{-\mathbf{T}}\mathbf{A^T})) + \sum_{i=1}^{N}\text{trace}(\mathbf{A_i}\mathbf{B'_i}\mathbf{A_i^T}) \\ &= \sigma^2(n+6) + B(\mathbf{x})\end{aligned} \tag{3.14}$$

Finally remarking that the bias term is always positive, and still null when $P_s(\mathbf{x})$ is modelled by a single plane completes the proof.

□

Another interpretation of Theorem 1 is that if we consider a multi-plane description, the expectation of the reprojection error for a given patch $P_s(\mathbf{x})$ if all the points in $P_s(\mathbf{x})$ were issued from a same plane. This means that during the sorting step of our algorithm, the patches containing only points from a same plane model are expected to be considered first.

Let's now analyze more thoroughly Eq. 3.9. The first term of the equation only depends on the noise and the 2D spatial distribution. Since the point set we consider here are square neighborhoods of constant size, this term is always the same. It can be interpreted as the distance between the noisy data and the estimated model if the data were distributed according to only one planar model.

The second term of Eq. 3.9 can be seen as the bias on the estimation that is introduced by all the points following a different model. Let's now take a look at the term $(\mathbf{A^T A})^{-1}\mathbf{A_2^T A_2}$. The part $\mathbf{A^T A}$ is computed from N points whereas the part $\mathbf{A_2^T A_2}$ was computed from n points. This means that if we normalize the matrices by the number of points, a factor n/N appears. In other words, the more points the second planar model contains, the bigger the bias term will be for the points of the first planar model which makes sense.

We note that in the N plane cases, the bias term may be null for a given point since one plane can nullify the effect of another. However, this cannot be the case for all the points in a given neighborhood which plays in favor of choosing neighborhood where only one plane describes the whole dataset.

Let's now illustrate these results experimentally. We considered two possible cases:

- Two planes separated at their intersection and forming an angle α with their orientations.
- Two horizontal planes separated by a step along the z-axis.

From these two cases, we considered different patches of points $P_s(\mathbf{x})$ containing from 50% to 100% of points from the first plane and the rest from the second plane. We then computed the z-variance as proposed in Eq. 3.1. Figure 3.4 shows the results obtained for the two possible

situations with different angles and steps. The experiments were done on 9×9 neighborhood with an additive noise on the disparity of standard deviation $\sigma = 0.5$ (similar results were obtained with different size of neighborhood and different standard deviation). In order to get the expectation, we took the average result obtained from 100 different realizations of input noise. The results go in accordance with the mathematical analysis and the intuitive prediction. The more points of a single model, the lower the error is expected to be.

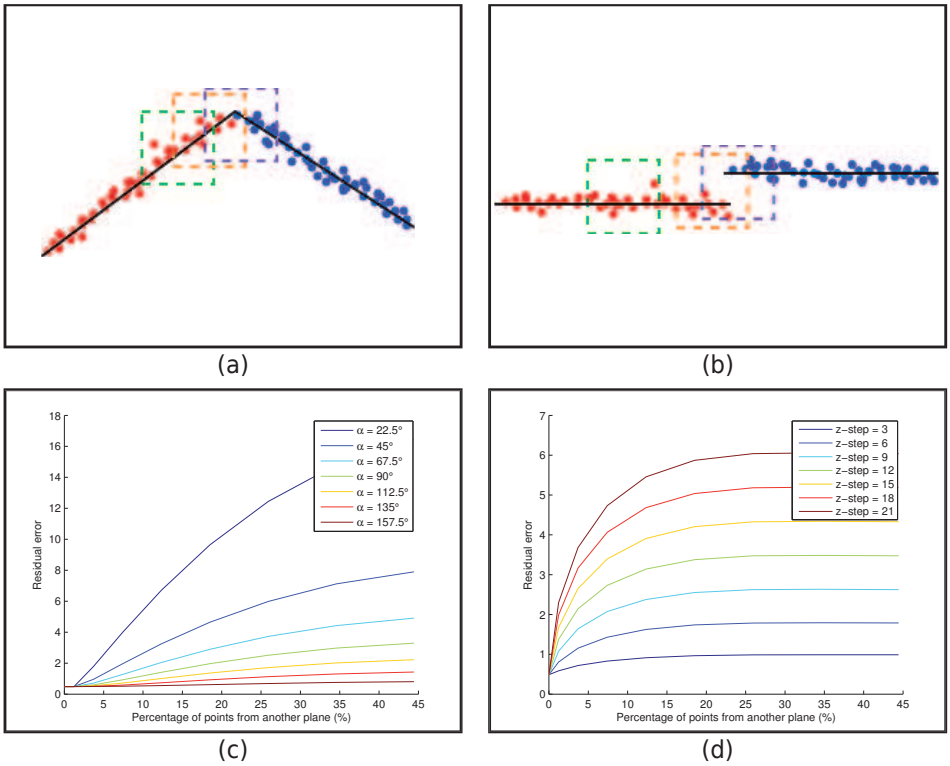


Figure 3.4: $z-$variance of the plane estimation depending on the percentage of points from another model. (a) First experiment: two intersecting planes forming an angle α with their orientations. (b) Second experiment: two horizontal planes separated by a step along the z-axis. (c) Results of first experiment for different angles. (d) Results of second experiment for different steps.

3.5 Plane estimation

The validation of points by hard thresholding in the region growing algorithm ensures that no outliers will be present in the final group. The estimation of the plane parameters can therefore be done by least squares minimization. A survey of the different approaches, on the best choice with our data as well as results on the expected precision from an estimation are given in Appendix A. We will concentrate here on the best estimator for our case given by Eq. 3.6 and try to give an explicit formulation for a fast and iterative expression adapted to region growing.

Another classical approach to solve Eq. 3.6 is proposed in [Taylor et al., 1989]. The result is obtained by deriving Eq. 3.6 with respect to each component of μ and equating each

equation to 0. The following equation system is then obtained:

$$\begin{cases} 2\sum_{i=1}^{N} x_i(z(\mathbf{x_i}) + \hat{a}x_i + \hat{b}y_i + \hat{c}) = 0 \\ 2\sum_{i=1}^{N} y_i(z(\mathbf{x_i}) + \hat{a}x_i + \hat{b}y_i + \hat{c}) = 0 \\ 2\sum_{i=1}^{N} z(\mathbf{x_i}) + \hat{a}x_i + \hat{b}y_i + \hat{c} = 0 \end{cases} \tag{3.15}$$

which gives the following solution:

$$\begin{cases} \hat{a} = \dfrac{\sigma_{x,y}\sigma_{y,z} - \sigma_{x,z}\sigma_{y,y}}{\sigma_{x,x}\sigma_{y,y} - \sigma_{x,y}^2} \\ \hat{b} = \dfrac{\sigma_{x,y}\sigma_{x,z} - \sigma_{y,z}\sigma_{x,x}}{\sigma_{x,x}\sigma_{y,y} - \sigma_{x,y}^2} \\ \hat{c} = -(\hat{a}\bar{x} + \hat{b}\bar{y} + \bar{z}) \end{cases} \tag{3.16}$$

where for $(\alpha, \beta) \in \{x, y, z\}^2$:

$$\begin{cases} \bar{\alpha} = \dfrac{1}{N}\sum_{i=1}^{N} \alpha_i \\ \sigma_{\alpha,\beta} = \dfrac{1}{N}\sum_{i=1}^{N}(\alpha_i - \bar{\alpha})(\beta_i - \bar{\beta}) = \dfrac{1}{N} \cdot \left(\sum_{i=1}^{N} \alpha_i\beta_i\right) - \bar{\alpha}\bar{\beta} = s_{\alpha,\beta} - \bar{\alpha}\bar{\beta} \end{cases}$$

From this explicit formulation of the solution, one can reestimate all the parameters using an iterative scheme each time a new point is added:

$$\begin{cases} \bar{\alpha}^{(n+1)} = \frac{n}{n+1}\bar{\alpha}^{(n)} + \frac{\alpha_{n+1}}{n+1} \\ s_{\alpha,\beta}{}^{(n+1)} = \frac{n}{n+1}s_{\alpha,\beta}{}^{(n)} + \frac{\alpha_{n+1}\beta_{n+1}}{n+1} \\ \sigma_{\alpha,\beta}{}^{(n+1)} = s_{\alpha,\beta}{}^{(n+1)} + \bar{\alpha}^{(n+1)}\bar{\beta}^{(n+1)} \end{cases} \tag{3.17}$$

An interesting result is that this estimation is independent from a scaling along the $z-$axis. From Eq. 3.16, one can easily deduce that the scaling $z' \mapsto \lambda z$ gives the same parameter scaling: $\hat{a}' = \lambda\hat{a}$, $\hat{b}' = \lambda\hat{b}$ and $\hat{c}' = \lambda\hat{c}$. Then, the plane detection is robust to any scaling of the values of a disparity map.

A second thing to note is that only few changes on the previous results are necessary in the weighted case. If we consider the weighted problem:

$$\min_{a,b,c} \sum_{i=1}^{N} w_i(z(\mathbf{x_i}) + ax_i + by_i + c) \tag{3.18}$$

Then the solution is Eq. 3.16 is the weighted least squares solution with the following modi-

fications for $(\alpha, \beta) \in \{x, y, z\}^2$:

$$\begin{cases} \bar{\alpha} = \dfrac{\sum_{i=1}^{N} w_i \cdot \alpha_i}{\sum_{i=1}^{N} w_i} \\ \sigma_{\alpha,\beta} = \dfrac{\sum_{i=1}^{N} w_i \cdot (\alpha_i - \bar{\alpha})(\beta_i - \bar{\beta})}{\sum_{i=1}^{N} w_i} = \dfrac{\sum_{i=1}^{N} w_i \cdot \alpha_i \beta_i}{\sum_{i=1}^{N} w_i} - \bar{\alpha}\bar{\beta} = s_{\alpha,\beta} - \bar{\alpha}\bar{\beta} \end{cases}$$

At last, a significant simplification in the expression of $\hat{\mu}$ can be made in presence of a rectangular neighborhood (see [Haralick, 1980]):

$$\begin{cases} \hat{a} = -\dfrac{\sigma_{x,z}}{\sigma_{x,x}} \\ \hat{b} = -\dfrac{\sigma_{y,z}}{\sigma_{y,y}} \\ \hat{c} = -\bar{z} \end{cases} \tag{3.19}$$

3.6 Complete algorithm and experimental results

The complete algorithm with all the steps is given in Algorithm 5.

Our algorithm was tested on various different disparity maps both piecewise planar and not. For all the experiments, 9×9 square patches were used independently of the size or the precision. At last, the plane validation criterion of Chapter 2 was used for each experiment to remove non planar regions.

3.6.1 Pure noise

As a sanity check experiment for both the algorithm and the *a contrario* validation, we first tried our method on randomly distributed disparity maps. The first disparity map was generated from a uniform distribution $\mathcal{U}([0, 100])$, the second from a normal distribution $\mathcal{N}([0, 100])$. The results obtained in the two situations are given in Fig. 3.5.

In the uniform distribution case, since the input data are exactly the same as the background model used the *a contrario* validation, no plane can possibly be detected. This result is the one expected from proposition 2 of Chapter 2.

In the Normal distribution case, the background model is estimated from the extrema of the distributed values. The more points, the more likely these extrema values are to be far from the mean of the normal distribution and the more narrow the point concentration will be compared to the extrema. Then this means that the Normal distribution is meaningful compared to the Uniform distribution which explains the final result. At last, the correct noise estimation was made in this case.

3.6.2 Disparity maps

Our next experiment was to try our algorithm on disparity maps. First, we used as input the St-Michel ground truth disparity map corrupted by an additive Gaussian noise of variance

Algorithm 5: Fast plane segmentation

Data:
Input disparity map: $(\Omega, z(\Omega))$
s: half the size of the local patch
Result:
$\mathcal{G} = \{G_1, \cdots, G_N\}$ a segmentation of Ω into disjoint groups
$\Pi = \{\pi_1, \cdots, \pi_N\}$ the list of corresponding planes

begin
  $\mathcal{G} = \emptyset$
  $Seeds = \emptyset$ **foreach** $\mathbf{x} \in \Omega$ **do**
    compute local plane and local residual error of $P_s(\mathbf{x})$
    add point $\mathbf{x}$ to Seeds
  end
  $Seeds \leftarrow$ sort $Seeds$ by local residual error
  $\tau_z \leftarrow$ compute initial threshold with Algorithm 4 on $Seeds(1..10)$
  foreach N_{min} *in* $\left\{(2s+1)^2, \frac{3}{4}(2s+1)^2, \frac{1}{2}(2s+1)^2, \frac{1}{4}(2s+1)^2, 0\right\}$ **do**
    $Seeds2 = \emptyset$
    foreach $\mathbf{x} \in Seeds$ **do**
      if $\#\big(P_s(\mathbf{x}) \setminus \cup_{G \in \mathcal{G}} G\big) \geq N_{min}$ **then**
        $G \leftarrow$ region growing on $\mathbf{x}$ with Algorithm 3 with τ_z
        add G to $\mathcal{G}$
        update τ_z value
      else
        add $\mathbf{x}$ to $Seeds2$
      end
    end
    $Seeds \leftarrow$ sort $Seeds2$ by local residual error
  end
end

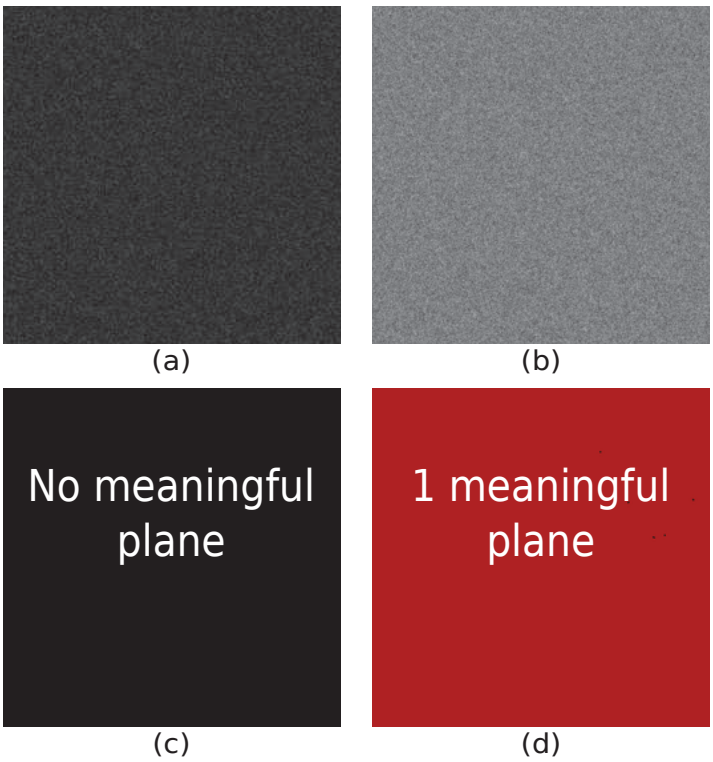


Figure 3.5: Noise as input disparity map. (a) $\mathcal{U}([0, 100])$. (b) $\mathcal{N}(0, 100)$. (c)-(d) Detected planes.

	init err	ℓ_2 err	N_{planes}	valid	time
Sawtooth	0.125 px	0.036 px	3	100%	2s
Venus	0.125 px	0.039 px	5	100%	2s
Cones	0.25 px	0.187 px	77	93.2%	3.9s
Teddy	0.25 px	0.189 px	72	93.1%	5.3s
books	0.25 px	0.149 px	63	98%	7.4s
St-Michel gt	0.02 px	0.0066 px	92	92.2%	8.9s
St-Michel [Sabater, 2009]	0.0304 px	0.0237 px	95	63%	12.6s

Table 3.1: Experimental results. First column: Initial uncertainty of input data in pixels. Second column: final ℓ_2 error between our piecewise planar approximation and the ground truth in pixels. Third column: Number of planes. Fourth column: percentage of valid points. Fifth column: computation time in seconds.

0.02 pixels. Our algorithm was compared to both methods in [Labatut et al., 09] and Chapter 2 using the same constant precision. The results given in Fig. 3.1 show that our algorithm is faster than the other two methods for the same order of precision.

The segmentation seems to be visually good which is all that can be said since no ground truth segmentation is available. In addition to that, the remaining ℓ_2 error (RMSE) is lower than the initial noise. This de-noising sanity check confirms the correctness of our planar grouping since a wrong segmentation provokes large errors after re-projection on planes.

At last, we used our automatic precision threshold estimation and obtained similar segmentation results. The resulting RMSE was exactly the same (0.0066) as the one obtained with an optimally hand-tuned threshold.

Similar conclusions were drawn when using our algorithm on real disparity maps computed with [Sabater, 2009]. The classification obtained in Figure 3.3 and the residual noise similar to the one of the ground truth proves that our piecewise-planar approximation is coherent. This is confirmed by Table 1 (last two rows), where the ℓ_2 error (RMSE between planar approximation and ground truth) is lower than the initial error (RMSE between input data and ground truth).

Our second experiment was done on Middlebury's disparity maps[2] [Scharstein and Szeliski, 2002] using the automatic precision of section 3. This time, the initial error q is only due to the quantization step, and the ℓ^2 error of our piecewise planar approximation is measured with respect to this quantized ground-truth. Thus $q/\sqrt{12}$ provides a lower bound to the ℓ^2 error (see Table 1). Since, this lower bound is met exactly for Sawtooth (and up to 10% accuracy for Venus), we conclude that we do not introduce any error by projecting on the planes that we found, and that the automatic noise estimation works correctly. For the other three maps, the result is a piecewise planar approximation of a scene containing non-planar surfaces, so the slightly larger errors may be attributed to model mismatch. The results of Middlebury experiments are shown in Fig 3.6 and the error measurements are all given in Table 1.

3.6.3 Expected re-projection error

In Appendix A, an important result on the expected error from linear regression was shown through proposition 4: the expectation of the re-projection error (square of the difference

[2] www.vision.middlebury.edu/stereo

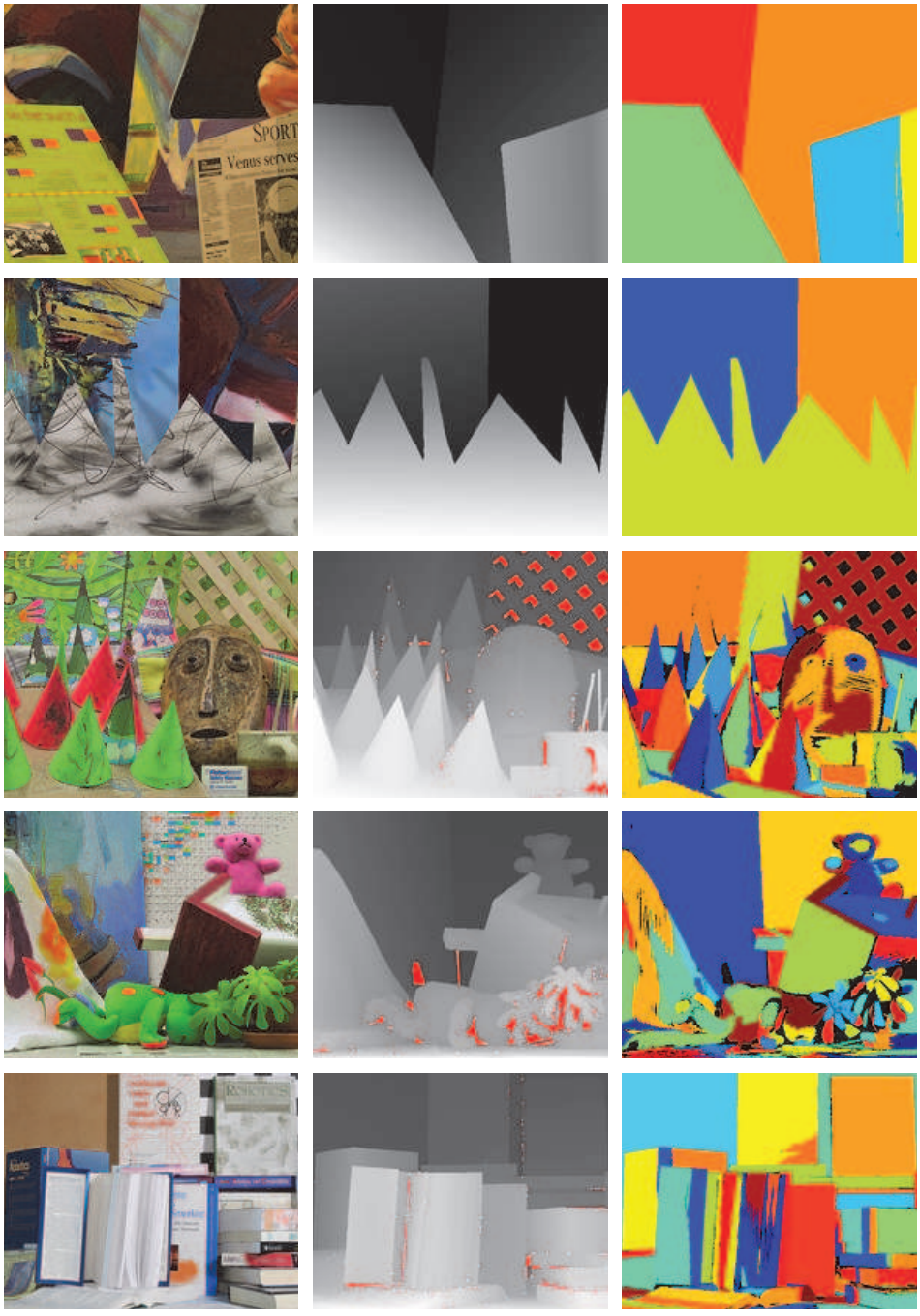


Figure 3.6: Middlebury results: from top to bottom, Venus, Sawtooth, Cones, Teddy, Books. *Left: reference image. Middle: original disparity map. Right: piecewise planar classification (each grey level corresponds to a different plane).*

between the perfect disparity and the one obtained after re-projecting on the plane) at each point is given by $\sigma^2 \text{rank}(\mathbf{A})/n$, where σ is the standard deviation of the additive Gaussian noise, $\mathbf{A}$ is the matrix used before for parameter estimation and n is the total number of points used for the plane estimation. This means that for any group of points, its expected cumulative re-projection error is independent from the number of points and is: $\sigma^2 \text{rank}(\mathbf{A})$.

We tried to observe this result experimentally with the obtained disparity map segmentation. For this experiment, we used two piecewise-planar ground truth disparity maps corrupted by an additive Gaussian noise (Toulouse St-Michel and village 1).

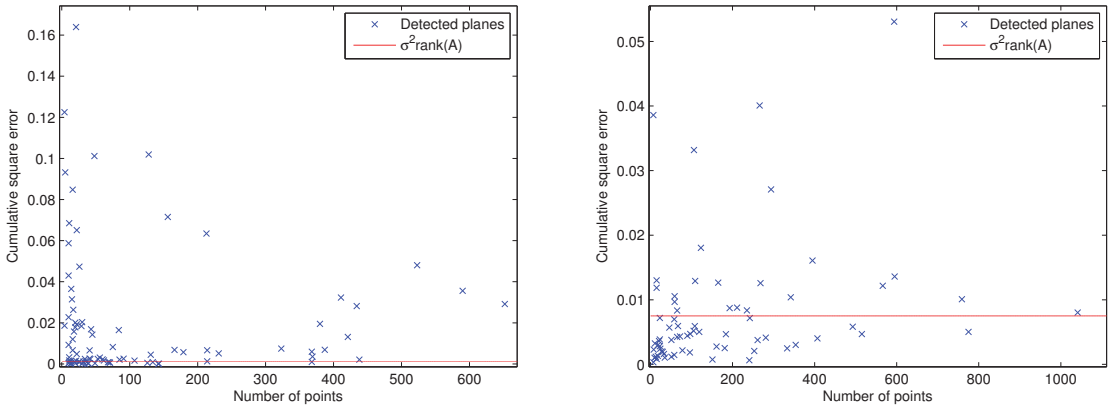


Figure 3.7: Cumulative square re-projection error for each plane detected: $\sum_{\mathbf{x}\in G}(z(\mathbf{x}) - z_\pi(\mathbf{x}))^2$. (a) Toulouse ground truth. (b) Village 1 ground truth.

The first thing to note is that the cumulative error is a square error at each point which means that any wrong point in a segmentation is likely to increase considerably the final measure. The main default of the approach presented here is its greedy component. A point standing near the edge of two distinct planes will be associated to the first one considered by the algorithm as long as the rejection conditions are respected (hard thresholding). This is done regardless the potential belonging to another planar group.

For the sake of this experiment, we propose to remove all these ambiguous points from the plane parameter estimations and the final measurements. To do so, we first propose to run the algorithm as proposed before. Then, the algorithm is run once again, for all the meaningful planes, this time in reverse order. The differences in segmentation are due to the ambiguous points up to the rejection threshold. The final measurements are done by considering only the meaningful planes without the ambiguous points.

Fig. 3.7, show the results obtained for each detected plane in the two disparity maps. In each case, the ground plane was removed from the results since its error was very large compared to the other planes which made the results impossible to view. One can see that in both cases, the results are coherent with what was expected for most planes which confirm that the segmentation is rather good up to the ambiguous points which were removed. The planes with larger errors must be taken into account carefully since a few points can make the measurements fail (because of the square errors). The bad results in that cases may then not be due to a bad segmentation.

3.7 Conclusion

We presented an algorithm for optimally grouping 3D points into planar patches. Inspired from the line-segment detection algorithm of [Grompone von Gioi et al., 2008], it is composed of a region growing step to find potentially planar regions in a disparity map, and of a validation step based on computational gestalt theory [Desolneux et al., 2008b]. The algorithm was proved to give faster results than the method of [Labatut et al., 09] and Chapter 2 for the same order of precision. Moreover, our algorithm has the main advantage of automatically estimating the precision of the disparity map, which is usually a critical parameter for other methods.

Our experiments show that the proposed approach is capable of detecting a reasonable piecewise affine decomposition in both complex urban scenes, and in non-piecewise-planar scenes. Moreover, the corresponding regularization actually reduces the surface approximation error contained in disparity measurements. In the piecewise planar case, the experiments suggests that theoretical errors (see proposition 4 of Appendix A) are obtained for most planes in the final classification.

Several applications and improvements are thought of. The piecewise planar grouping can be used as a basis for interpolation and vectorization algorithms. However, these applications will require a stronger use of luminance, such as the geodesic distance technique in [Facciolo and Caselles, 2009], that can be useful for fine-tuning of the border location between planar regions, as well as for model selection between continuous or discontinuous transitions between patches. Moreover, overall performance will be improved if completely automatic methods are developed for detecting the regions that are potentially affected by strong fattening effects. These points are tackled in the next Chapter of this thesis. At last, due to the good results of our algorithm obtained on disparity maps, an extension to 3D data is envisaged.

Chapter 4

Experiments and algorithm improvements

Contents

Résumé: Dans ce chapitre, nous donnons des résultats supplémentaires obtenus avec l'Algorithme 5 présenté dans le Chapitre 3 sur différents types de données. Dans un premier temps, nous avons testé l'algorithme sur la vérité terrain de Toulouse St-Michel corrompue par un bruit additif Gaussien et avec seulement 10% des points renseignés. Puis nous avons utilisé les images de profondeur ABW fournies par l'Université de Floride du Sud. Ces tests ont pointé la nécessité d'une correction de l'approche gloutone lorsque le bruit dans les données n'est pas homogène et que les plans intersectent le sol. Puis, dans un troisième temps, nous avons testé l'algorithme ainsi corrigé sur des données stéréo obtenues par différentes approches de calcul de cartes de disparités. Enfin, dans un dernier temps, nous avons testé la possibilité d'utiliser la connaissance de la précision de la carte de disparités pour améliorer le calcul des différents plans. Ces résultats tendent à montrer l'efficacité de notre algorithme pour plusieurs domaines: segmentation, débruitage et enfin interpolation.

Abstract: In this chapter, we give additional results obtained with Algorithm 5 introduced in Chapter 3 on various datatypes. First, we tried our algorithm on the Toulouse St-Michel ground truth corrupted with additive Gaussian noise and with only 10% of known disparities. Then, we tested the ABW range images from the University of South Florida. This pointed out the necessity of correcting our greedy algorithm when the noise is not homogeneous and when planes intersect the ground. We then tested our algorithm with the previous correction on real stereo dataset computed from various algorithms. At last, we tried to see if using the *a priori* knowledge of the point precision could improve the plane estimation. All these results tend to prove the efficiency of our method for 3 different things: segmentation, denoising and interpolation.

4.1 Sparse disparity maps

As a first test, we propose to try our algorithm on a sparse disparity map. It is necessary for our algorithm to work under those conditions since stereo data are usually not dense. We consider here the situation described in Fig. 4.1 (a): a noisy version (additive Gaussian noise) of the Toulouse St-Michel ground truth with only 10% of available points. To work in this case, the algorithm first needs some modifications.

In its original version, the region growing step consisted in testing the distance to the considered plane of the neighboring points and adding those points if the distance was less than a threshold. Since the points stood on a regular grid, the neighboring points were defined by the 4-connectivity.

In the sparse case, the points are still on a regular grid but some of them are missing. To define the connectivity in this case, we compute the Voronoi diagram of the remaining points. The connectivity is then defined by the connectivity of the Voronoi cells. When the sparse points stand on a regular grid, an approximation of the Voronoi diagram can be fast computed ($\mathcal{O}(\#\Omega)$ computations) using the 3-4 Chamfer distance [Borgefors, 1986] (see Fig. 4.1 (b) to see the Voronoi cells obtained in the tested case).

Figure 4.1 show the results that were obtained with this modification of the algorithm on the sparse disparity map of Toulouse St-Michel. The final classification is close to the classification that was obtained was a denser disparity map (see previous Chapter). Moreover,

combining the Voronoi cells to the classification, one can interpolate the missing disparities by projection on the associated plane. The result of the obtained interpolation is shown on Fig. 4.1 (d). The error committed after the reprojection on the planes from our classification are then compared to the one committed by interpolating the missing points with a median filter, which is the method proposed in [Sabater, 2009] to interpolate missing disparities (see Figure 4.1 (e) and (f) for these results). In these two images, the same dynamique was used: the error values are spanned between 0 pixels (white) and 0.10 pixels(black). One can see that the method we propose here seems far more adapted because it denoises the initial points. The remaining error are for the most located near the plane delimitations which is due to the fact that the contours of the Voronoi cells have no reason to correspond to the actual contours of the planes in the scene. This suggest the use of another approach to refine the contours of the plane classification (see next Chapter).

4.2 Range datasets

We now propose to use range data as an input to our algorithm. We used the ABW dataset from the University of South Florida[1]. This dataset is made of 40 range and intensity images obtained from a structured light device on piecewise planar scenes. Though the real final classifications are not available, the intensity images give a good hint on how the range images should be segmented. This dataset was part of the experiments made in [Hoover et al., 1996] to compare range image segmentation algorithms.

4.2.1 Misclassifications

After having run our algorithm on this dataset two shortcomings were pointed out:

1. A point is associated to a plane when the distance to that plane is less than the distance rejection threshold τ_z. The problem is that this condition may be true for more than one plane. Because our algorithm is greedy, the points are then associated to the first group selected for region growing for which the distance condition is respected. This means that the classification is probably wrong near plane intersections.

2. The estimation of the rejection threshold is made from the previously validated groups. However, if the point precision is not homogeneous through the different regions, this estimation may not be adapted everywhere. Since the first regions that are tested are the flattest ones (lowest error), the regions with a larger error will be over-segmented because the rejection threshold will be too low.

These two defects are illustrated with Figure 4.2 (c): the front green face of the object splits the ground in our classification because it appears before in the greedy classification. Moreover, the back plane is separated into several parts because the precision is not homogeneous.

Note that the defects did not appear with the experiments of the Chapter 3 for two reasons.

- In the previous experiments, the ground plane was always detected first because orthofrontal planes (in this case the ground) are favored by the seed sorting step and are therefore usually selected first for region growing. In the ABW dataset, the orthofrontal planes are the background and some object facets. The orthofrontal facets of objects

[1] http://marathon.csee.usf.edu/range/DataBase.html

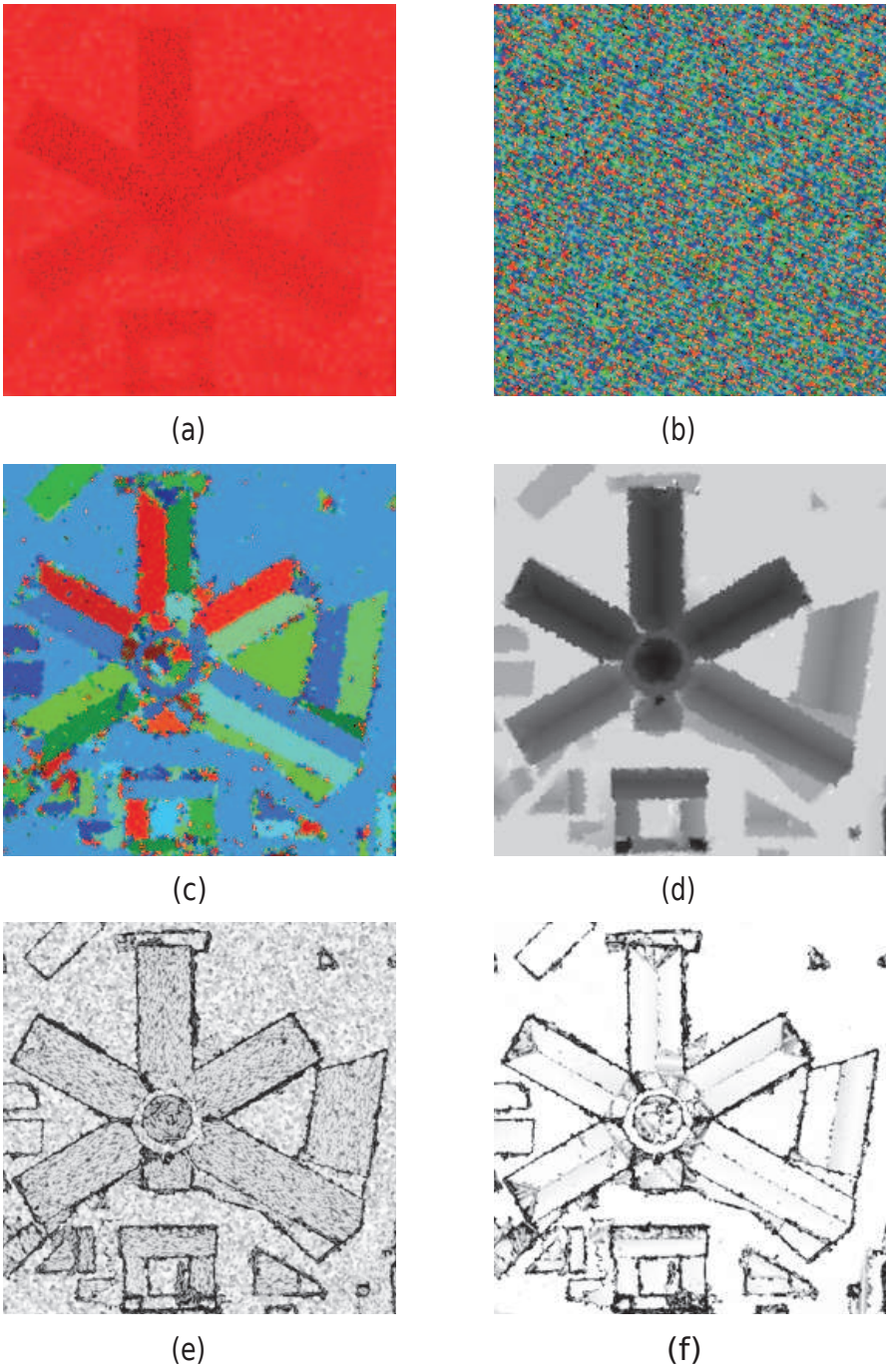


Figure 4.1: Toulouse St-Michel ground truth + additive noise: 10% of total number of points. (a) Input disparity map. Red parts represent unknown points. (b) Voronoi diagram of the known points. (c) Piecewise-planar classification obtained with our algorithm. (d) Interpolation of the missing disparities by reprojection from the classification. (e) Error after interpolation with median filter (0 pixel error = white, 0.1 pixel error = black). (f) Final reprojection error after plane classification (same image dynamique as (e)).

are detected before the ground plane and the points standing on the intersection line between the ground plane and the object facet are validated as part of the object facet.

- The second reason is that the noise was uniform in the tests that were made in Chapter 3.

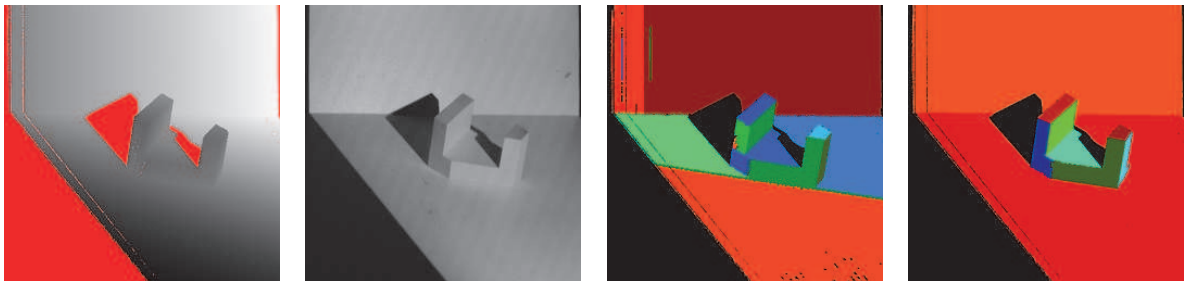


Figure 4.2: (a) Input range image (red parts are missing points). (b) Intensity image. (c) Classification with the original algorithm of Chapter 3. (d) Classification after correction of our algorithm.

4.2.2 Greedy correction algorithm

To overcome the two defects presented before, two corrections are needed:

- Merging the groups that have been separated by either one of the two defects.
- Associating the points with an ambiguous classification to the most likely plane.

The first thing to do before applying any of these two corrections, is to find the ambiguous points. To do so, we propose to run the greedy algorithm in reverse order. As the points are associated to the first valid group encountered, ambiguous points will then have a different classification. Moreover, this will point out the candidate group pairs for merging since these pairs should obviously share ambiguous points.

To both be able to merge groups and assign ambiguous points, we propose to use the local plane orientations. This is justified by the fact that points associated to a same plane should have locally a similar orientation than the global plane orientation. The choice of using local orientation has also been made in other methods such as [Chauve et al., 2010] or [Schnabel et al., 2007b] to gain robustness. In those methods, it was used as an additive criterion to reject points: the points for which the angle between the local and the global plane is larger than an angle threshold set by the user. However using another parameter is not in accordance with our will to limit parameters.

Following a similar reasoning as in Chapter 3, we propose to estimate the distribution of the differences between local plane orientations and global plane orientations. Once again, we suppose that these angle differences are distributed according to a Normal law $\mathcal{N}(0, \sigma_\theta)$. The

standard deviation σ_θ can then be estimated from the initial piecewise planar classification $(G_i)_{i=1..n}$:

$$\hat{\sigma}_\theta = \sqrt{\frac{\sum_{G_i} \sum_{\mathbf{x} \in G_i} \arccos < \mathbf{N}(\mathbf{x}), \mathbf{N}(G_i) >}{\sum_{G_i} \#G_i}} \tag{4.1}$$

where $\mathbf{N}(\mathbf{x})$ is the local orientation of point $\mathbf{x}$ computed from the local neighborhood $P_s(\mathbf{x})$, $\mathbf{N}(G_i)$ is the global orientation of plane G_i and $< \cdot, \cdot >$ is the scalar product between the two 3D vectors.

Since the local orientations are not precise when the points inside $P_s(\mathbf{x})$ are distributed according to several planes, the points near the group delimitations should be rejected from the estimation:

$$\hat{\sigma}_\theta = \sqrt{\frac{\sum_{G_i} \sum_{\mathbf{x} \in G_i \setminus P_s(\partial G_i)} \arccos < \mathbf{N}(\mathbf{x}), \mathbf{N}(G_i) >}{\sum_{G_i} \#(G_i \setminus P_s(\partial G_i))}} \tag{4.2}$$

From this estimation we can now define a simple merging criterion. Two planes sharing ambiguous points are merged if:

- The distance of the barycenter of each plane to the other plane is less than the distance rejection threshold τ_z given by Eq. 3.3 from Chapter 3.
- The angle between the two plane orientations is less than the $\hat{\sigma}_\theta$ given in Eq. 4.2.

The ambiguous point correction is then treated by associating the points to the plane with the orientation most similar to theirs. At last, for more robustness with the angle estimation, the algorithm can be run in loop until the final number of groups stays constant. A summary of the correction algorithm is given with Algorithm 6.

4.2.3 Results

The final result obtained after applying the correction algorithm is shown in Figure 4.2 (d). One can see that the planes were properly merged and the ambiguous points assigned to the right plane. The final segmentation is coherent with the intensity image.

The new algorithm was then tested on the 40 images of the dataset. Figure 4.3 shows some example of the obtained classifications as well as the error compared to the original range image after having reprojected the points on the planes that were found. Among the 40 range images we obtained a segmentation visually coherent with the corresponding intensity images for 36 of them. An over-segmentation for two of the planes was obtained for 2 of the range images. At last, 2 segmentation were wrong because of a wrong plane merging.

The RMSE (Root Mean Square Error) of the reprojection error [2] was then computed for each range image. Its mean value was 3.7025 for range images with values between 0 and 255

[2] $\sqrt{\frac{\sum_{\mathbf{x} \in \Omega} (z_\pi(\mathbf{x}) - z(\mathbf{x})^2)}{\#\Omega}}$

Algorithm 6: Non-homogeneous noise and greedy corrections

Data:
$\mathcal{G} = \{G_1, \cdots, G_N\}$ a list of meaningful planar groups of a disparity map $(\Omega, z(\Omega))$
Result:
$\mathcal{G}' = \{G'_1, \cdots, G'_M\}$ the corrected segmentation

begin
  $\mathcal{G}' \leftarrow$ run Algorithm 5 in reverse order
  $N_{groups} = 0$
  while $\mathcal{G}' \neq N_{groups}$ **do**
    $N_{groups} = \#\mathcal{G}$
    $\hat{\sigma}_\theta \leftarrow$ estimate the local angle distribution from $\mathcal{G}'$
    $\hat{\sigma}_z \leftarrow$ estimate the local error distribution from $\mathcal{G}'$
    foreach $(i, j), i \neq j$ *such that* $G_i \cap G'_j \neq \emptyset$ **do**
      $\alpha \leftarrow$ compute angle between normal of the planes of G_i and G'_j
      $dist_i \leftarrow$ compute distance between plane of G_i and barycenter of G'_j
      $dist_j \leftarrow$ compute distance between plane of G'_j and barycenter of G_i
      if $\alpha < \hat{\sigma}_\theta$ & $dist_i < \hat{\sigma}_z$ & $dist_j < \sigma_z$ **then**
        Merge G_i and G_j in $\mathcal{G}$
        Merge G'_i and G'_j in $\mathcal{G}'$
      else
        $G_{ambiguous} = G_i \cap G'_j$
        $G_i = G_i \setminus G_{ambiguous}$, $G'_i = G'_i \setminus G_{ambiguous}$
        $G_j = G_j \setminus G_{ambiguous}$, $G'_j = G'_j \setminus G_{ambiguous}$
        foreach $\mathbf{x} \in G_{ambiguous}$ **do**
          $\alpha_i \leftarrow$ compute angle between normal of the plane of G_i and local plane of $\mathbf{x}$
          $\alpha_j \leftarrow$ compute angle between normal of the plane of G'_j and local plane of $\mathbf{x}$
          if $\alpha_i < \alpha_j$ **then**
            add $\mathbf{x}$ to G_i and G'_i
          else
            add $\mathbf{x}$ to G_j and G'_j
          end
        end
      end
    end
  end
end

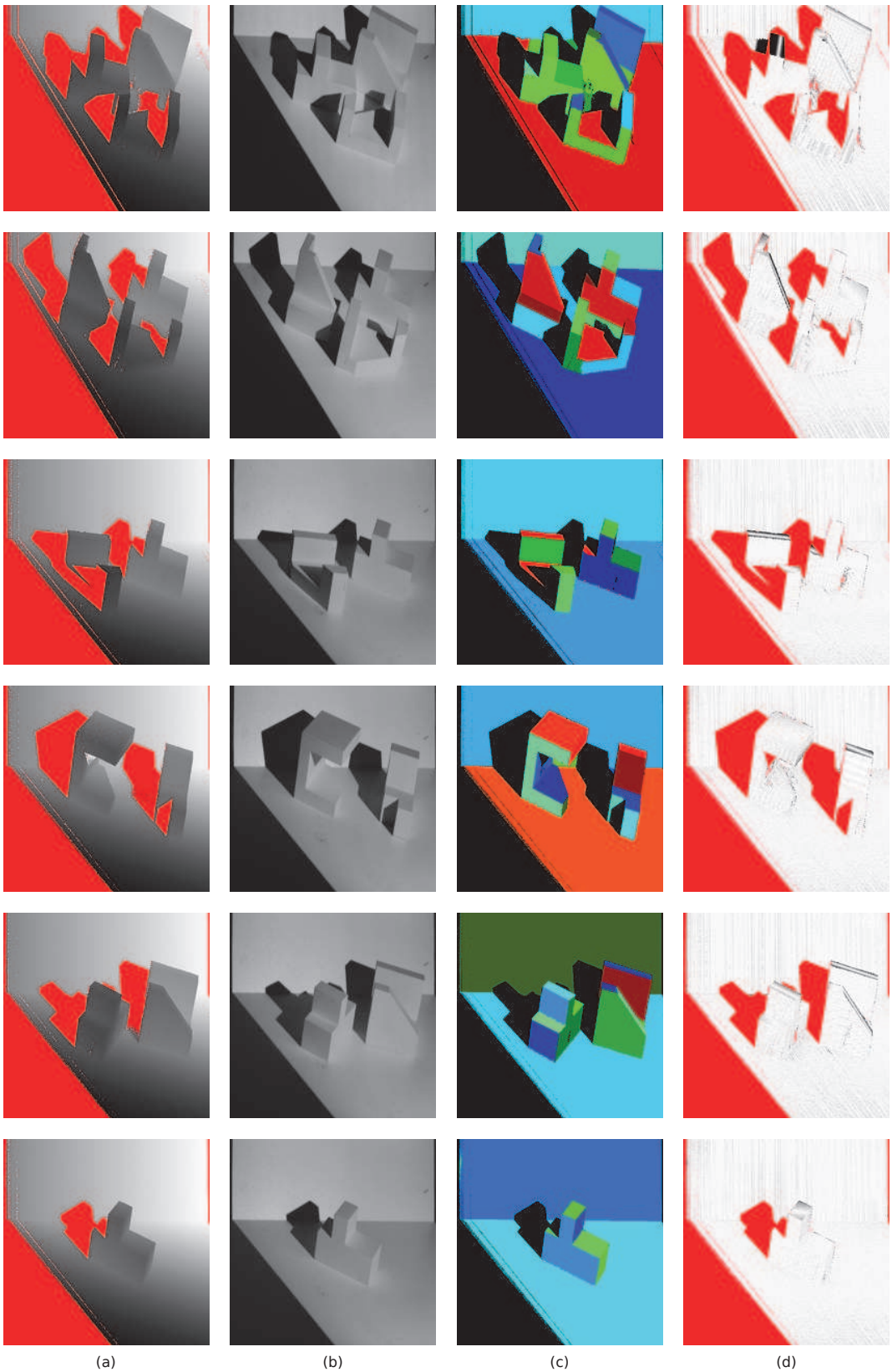


Figure 4.3: Experiments on range images. (a) Range values (red parts are unknown). (b) Intensity image. (c) Classification obtained with our algorithm. (d) Reprojection error (the larger the error, the darker the image)

which proves that both the classification and the planes computed are correct. The largest errors are mostly due to the planes which orientation is almost along the z-axis. In the case, the plane estimation is not very precise. This is confirmed by the aspect of the reprojection error images given in Figure 4.3 (d).

All these experiments confirm the validity of our algorithm with laser range images.

4.3 Real stereo dataset

To see how our algorithm behaves in real situations, we now try our algorithm on disparity maps which were computed with three different methods:

- MARC algorithm (Multi-resolution Algorithm for Refined Correlation) [Delon and Rougé, 2007] which was designed by the French Space Agency (CNES);
- MicMac algorithm (Multi-Image Correspondances, Méthodes Automatiques de Corrélation) [Pierrot-Deseilligny and Paparoditis, 2006] from the French Geographic Institute (IGN);
- and the algorithm proposed by Neus Sabater [Sabater, 2009] in her Ph.D. thesis.

Each of these methods has its own advantage compared to the other two which allows to face different situations and see what makes our algorithm succeed or not.

4.3.1 Short description of the methods

Let's first give a short description of each algorithm for a better understanding of the resulting disparity maps.

MARC [Delon and Rougé, 2007] and Sabater's algorithm [Sabater, 2009]

These two local methods were designed to work under low baseline conditions which requires a sub-pixelic precision of the final result (see Chapter 1 for more details on the low baseline configuration). Both uses the same two-stepped approach to achieve this precision:

1. Rough localization of the disparity at each point by maximizing the Normalized Crossed Correlation(see Chapter 1) and rejection of the uncertain points.
2. Sub-pixelic refinement by finding the local minimum of the weighted quadratic distance between the two images. This distance is given for a point $\mathbf{x_0}$ of the first image, a disparity μ and a smooth weighted window φ by the following formula:

$$d_\varphi(\mathbf{x_0}, \mu) = \int_{\mathbf{x}\in\Omega} \varphi(\mathbf{x} - \mathbf{x_0})\big(u(\mathbf{x}) - \tilde{u}(\mathbf{x} + (\mu, 0))\big)^2 d\mathbf{x} \tag{4.3}$$

The first step is what differs in each of the two methods.

In MARC, the chosen strategy is multi-scale (coarse to fine). At each scale, the disparity is computed by maximizing the Normalized Crossed Correlation(NCC) at the pixelic level for the considered scale. Then, the obtained estimation is used to limit the search at the next scale. An analysis of the correlation measure is moreover used to reject the disparities computed in regions with not enough texture to achieve a good precision. At last, the disparity computation is coupled with a correction to avoid the adhesion artifact (see Chapter 1 for

more details on this artifact). This correction assigns the computed disparity to the barycenter of the points within the correlation window. The result is then a sparse disparity map with points concentrated around highly contrasted edges and textures.

In Sabater's Algorithm, the first step is achieve by a rough search of the maximum of the NCC. This approach takes more computations than the multi-scale approach but avoids the propagation of errors through the scales. An *a contrario* criterion is then defined to reject the points where the measure may be false.

The second step of both algorithms is done at each point of the grid by first interpolating the quadratic distance of Eq. 4.3 for all the values of μ, then finding the local minimum within the interval $[\hat{\mu}(\mathbf{x_0}) - 0.5, \hat{\mu}(\mathbf{x_0}) + 0.5]$, where $\hat{\mu}(\mathbf{x_0})$ is the disparity estimated at point $\mathbf{x_0}$ at a pixelic precision level. The interpolation is performed using a 1D Shannon interpolation from the distance values at each μ with half a pixel resolution (see [Sabater, 2009] for a justification of that). This step then requires to zoom in the two images of a factor 2 to be able to compute Eq. 4.3 for half pixel values of μ. Such is done by zero-padding of the two images.

MicMac [Pierrot-Deseilligny and Paparoditis, 2006]

MicMac algorithm was designed to compute disparities under several conditions. Contrary to the other two methods, the approach the disparities are computed globally by minimizing the following energy:

$$E_\alpha(\mu) = \sum_{\mathbf{x}\in\Omega} A(\mathbf{x}, \mu(\mathbf{x})) + \alpha R(\mathbf{x}, \mu(\mathbf{x})) \tag{4.4}$$

where

- $A(\mathbf{x}, \mu(\mathbf{x}))$ is a data fitting term such as the reverse Normalized Crossed Correlation $1 - NNC(\mathbf{x}, \mu)$. This term is null for a perfect matching between images and equals to 1 when the images do not match at all.
- $R(\mathbf{x}, \mu(\mathbf{x})) = |\mu(x+1, y) - \mu(x, y)| + |\mu(x, y+1) - \mu(x, y)|$ is a term imposing regularity on the values of the disparity.
- α is a parameter weighting the relative importance of data fitting and regularization.

The energy is minimized by first quantifying the possible disparity values and applying a multi-scale variant of the min cut max flow algorithm [Roy and Cox, 1998]. The result is then a dense disparity map with quantified values.

4.3.2 Experiments

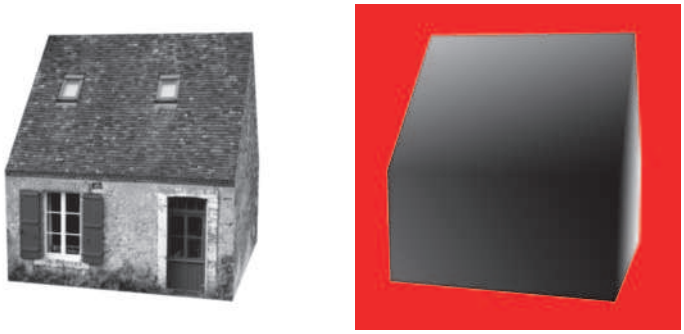
We tested the algorithm with various real stereo pairs. The result of the three algorithms is not available for all pairs. Moreover, a ground truth disparity map is sometimes available which allows comparison of the stereo computation algorithms. For a fair comparison, we chose not to use the RMSE since its value is not completely representative of what happens. Indeed, a single outlier can completely change the RMSE value even if the error in each point was reduced.

Instead of that, we computed error maps and used the same dynamique for each error image. The comparison are then made qualitatively by observing grey level values for each error image.

"Bergerie" dataset

This dataset is a stereo pair which was simulated by Lionel Moisan for CNES. For this map, a ground truth is available which allows to make error measurements. Moreover, the denoising effect of the planar reprojection can be observed.

The reference image of the stereo pair as well as the ground truth disparity are shown in Figure 4.4. The observed 3D scene is made of three distinct planes.



(a) (b)

Figure 4.4: "Bergerie" dataset: (a) Reference image of the stereo pair. (b) Ground truth disparity map.

The results that were obtained with the different algorithms as input are shown in Figure 4.5. The three different algorithms are tested here as well as the ground truth with an additive noise. The errors of the algorithms to the ground truth are shown in Figure 4.5 (d). For a better comparison, the same image dynamique was used on every error image (white = 0 pixel error, black = 0.10 pixel error). Then, the whiter the error image looks, the less error there is. As expected from the algorithm description, the method producing the lowest error is Sabater's algorithm. The results are better than MARC algorithm which uses the same sub-pixelic approach. This is mainly due to the multi-scale step of MARC which can propagate errors through the scales combined with the outlier rejection criterion which is less effective than the one in Sabater's algorithm. At last, MicMac algorithm is the less precise because it is not well adapted to compute sub-pixelic disparity. Moreover, the result is quantized which is a flaw compared to the other two algorithms. However the disparity map is dense which is an advantage over the other two algorithms.

Let's now look at the classification results obtained with the original algorithm from Chapter 3 (Figure 4.5 (b)). The disparity maps were well segmented with noisy ground truth, MARC and MicMac algorithms. However, the classification obtained with Sabater's algorithm is not satisfactory. The explanation is that the residual errors of the noisy ground truth, MARC and MicMac are almost homogeneous whereas it is not the case for Sabater's algorithm. As explained in section 4.2, as the outlier rejection threshold in our algorithm is estimated from the first planes that are detected (usually the planes with a lower error), the threshold is not adapted to the parts of the disparity maps with another error distribution.

Let's now give an explanation of the non-homogeneous error distribution of Sabater's algorithm. In this method, the disparity is computed by block matching and is based on the assumption that the disparity is constant in the blocks that are matched. This hypothesis is however not in accordance with the 3D geometry of the scene and is not adapted to very slanted planes. The disparity error will then be higher for slanted planes than for orthofrontal

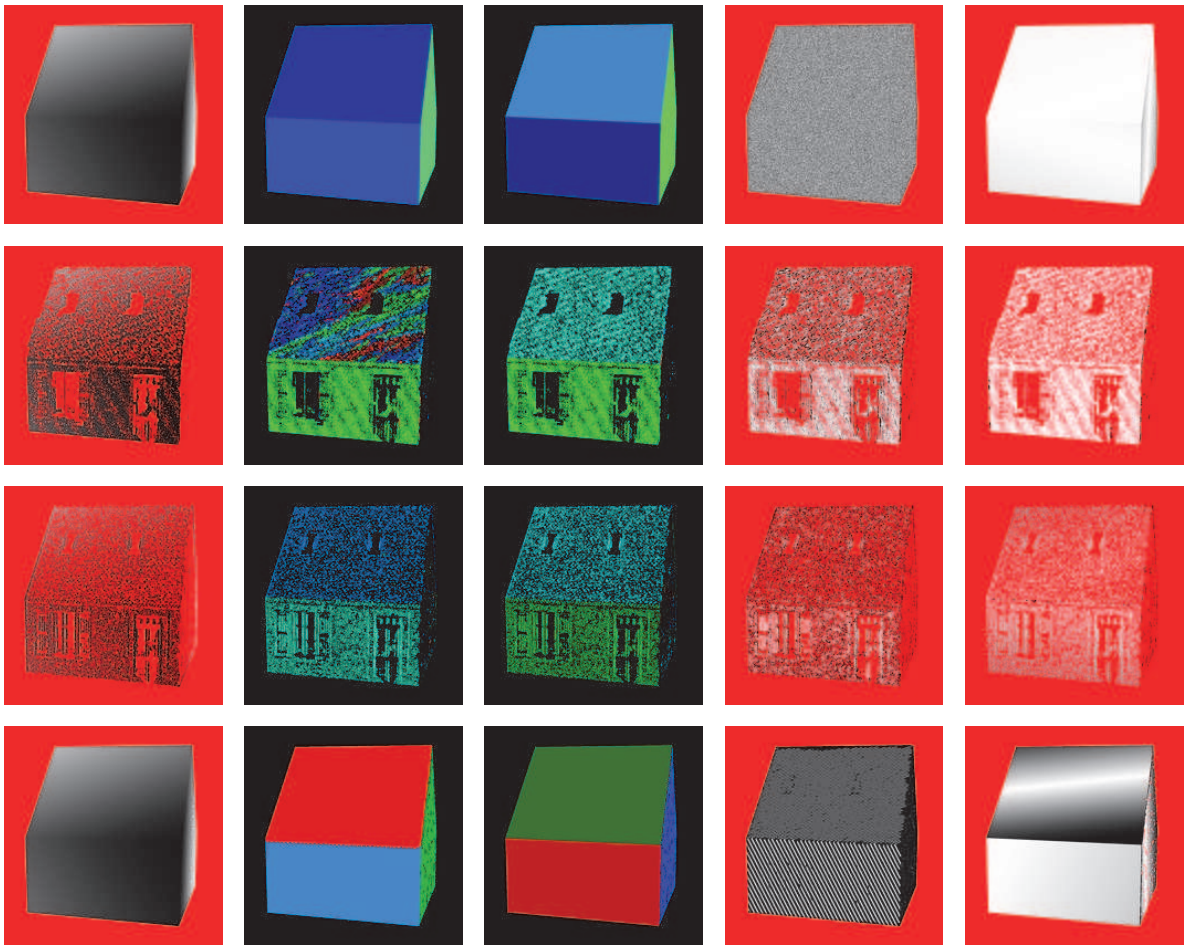


Figure 4.5: Results with different input algorithms. From top to bottom: ground truth + additive Gaussian noise, Sabater's algorithm, MARC algorithm, MicMac algorithm. (a) Disparity map (red parts are missing points). (b) Classification with the initial algorithm of Chapter 3. (c) Classification after the correction of section 4.2.2. (d) Initial error to ground truth (0 pixel = white, 0.1 pixels = black). (e) Final error after reprojection on the classification planes (0 pixel = white, 0.1 pixels = black).

planes (which is one of the aspects of the adhesion artifact, see Chapter 5). For MARC and MicMac algorithm, this error is partly corrected by the barycentric correction (MARC) and by the global minimization (MicMac) which explains the more homogeneous distribution of the global error.

To correct the classification errors due to the non-homogeneity of the error distribution, we used the correction algorithm that was described in Algorithm 6 (Figure 4.5 (c)). After applying the correction, good segmentations were obtained in every situations.

Let's now compare the residual error maps before and after reprojection on the classified planes (Figure 4.5 (d) and (e)). We see that with all the input algorithm, the images are whiter than before which means that the error to the ground truth has been reduced after the reprojection with our classification. This tends to justify the usefulness of this algorithm for disparity map denoising in piecewise-planar situations. Moreover, we see that the results obtained with Sabater's algorithm are a lot better than the ones found with the other approaches.

At last, in Figure 4.6 we show that our plane classification algorithm can be used for the interpolation of missing data. To do so, each unknown point is reprojected on the same plane as the nearest known point. The comparison with the median filter interpolation, as proposed in [Sabater, 2009], shows that our method is more adapted for data interpolation. This is due to the fact that the median filter is not adapted to noisy data which is always the case with real disparity maps. As opposed to that, since our method can also be used to denoise the data points, it is more adapted to interpolate missing points.

"Village" dataset

As the previous dataset, "Village" dataset is a stereo pair which has been simulated by Lionel Moisan. Once again a ground truth is available which allows to compare the input methods and the validity of interpolation. Figure 4.7 shows the reference image as well as the ground truth disparity map.

For this dataset, only the results of MicMac algorithm and Sabater's algorithm were available. Figure 4.8 shows the results obtained on the resulting disparity maps as well as the ground truth corrupted with additive Gaussian noise. As for the "Bergerie" dataset, a coherent classification was obtained after the application of our algorithm with the correction from section 4.2.2. Once again, the disparity maps were denoised in all the situations after reprojection of the points on the planes that were found.

At last, Figure 4.9, shows that the obtained classification can once again be used to interpolate the missing disparities in a coherent way. As before, by looking at the error images, one can see that the found interpolation is more adapted than the median filtering interpolation proposed in [Sabater, 2009].

Toulouse PELICAN dataset

PELICAN images are aerial images acquired by CNES. A stereo pair of the city of Toulouse was acquired and Sabater's algorithm was then ran on it. Since this stereo pair is a real one, no ground truth is available in this case. Therefore, the analysis of the results can only be based on the coherence with the reference image. The results are shown in Figure 4.10. Even in this case were the data are very noisy, we obtained a classification which seems coherent with the reference image. This coherence is confirmed by the fact that after the reprojection of the disparitiy map with our classification, the disparity map seems to be unchanged. However,

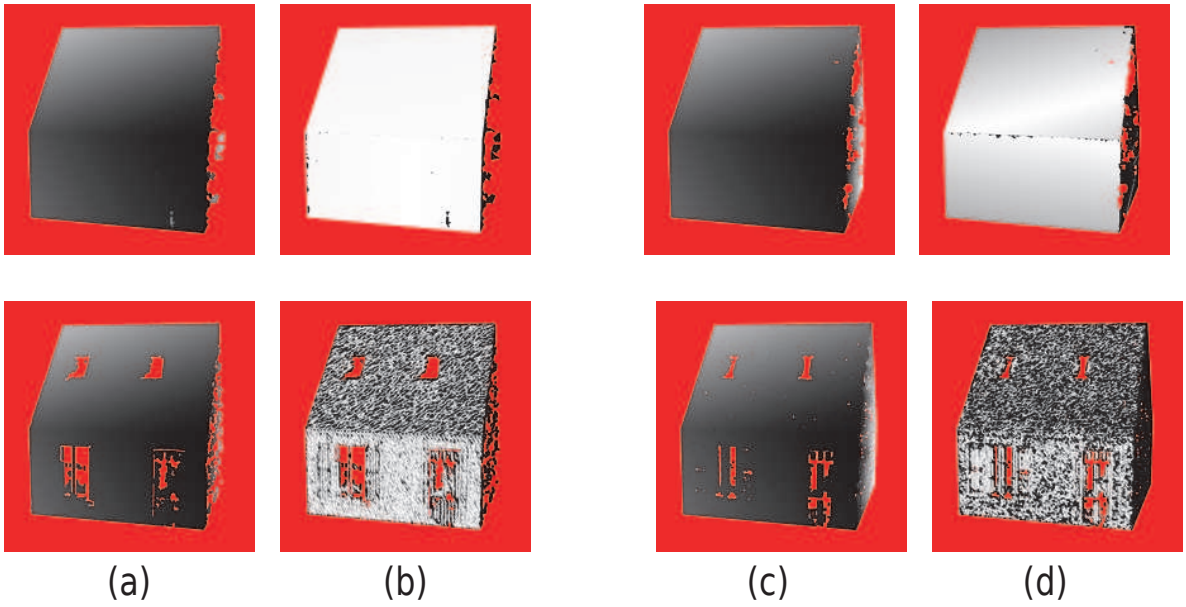


Figure 4.6: Interpolation of the missing points from the classification. Top Row: interpolation by reprojection on the nearest plane found with our algorithm. Down row: interpolation with median filtering. (a) Interpolated disparity for Sabater's algorithm disparity map. (b) Error between the reprojected disparity and the ground truth for Sabater's algorithm disparity map (0 pixel = white, 0.1 pixels = black). (c) Interpolated disparity for MARC algorithm disparity map. (d) Error between the reprojected disparity and the ground truth for MARC algorithm disparity map (0 pixel = white, 0.1 pixels = black). In both cases, the error commited with our interpolation is a lot lower than the one obtained with median filtering.

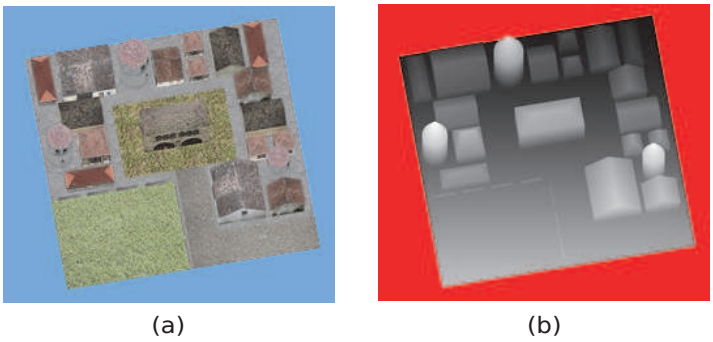


Figure 4.7: "Village" dataset. (a) Reference image. (b) Ground truth disparity map.

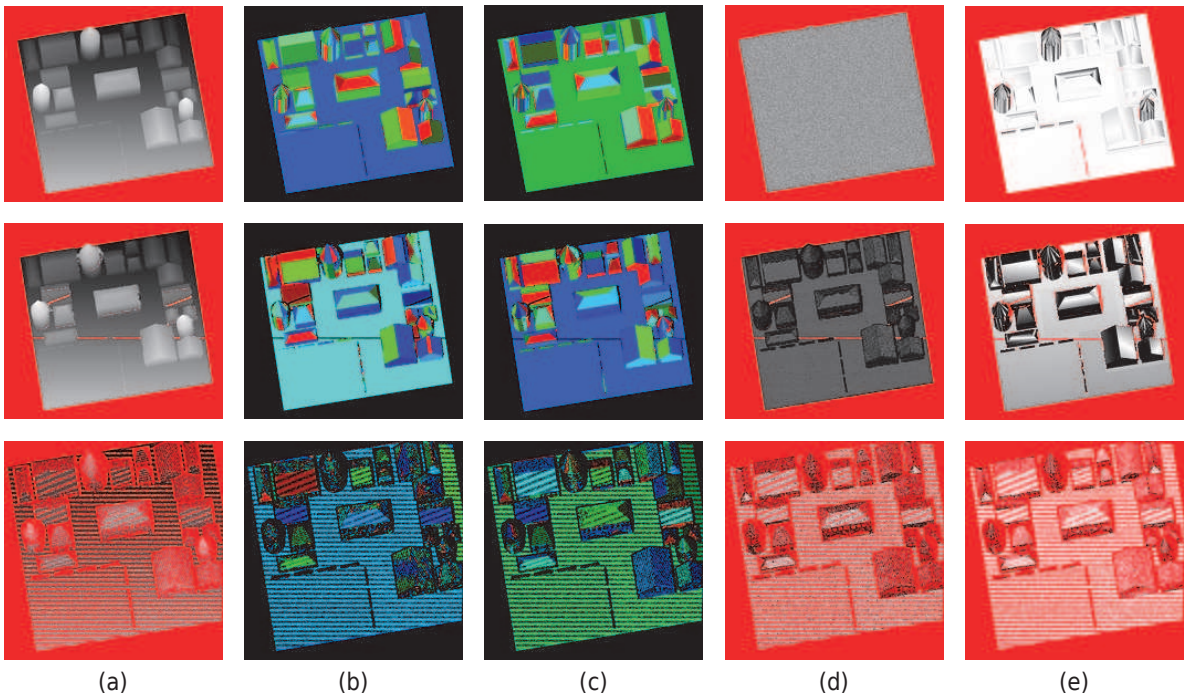


Figure 4.8: Result on the "Village" dataset. From top to bottom: ground truth + additive Gaussian noise, MicMac algorithm, Sabater's algorithm. (a) Disparity map (red parts are missing points). (b) Classification with the initial algorithm. (c) Classification after the correction. (d) Initial error to ground truth (dynamique: 0 pixel = white 0.1 pixels = black). (e) Final error after reprojection on the classification planes (dynamique: 0 pixel = white 0.1 pixels = black).

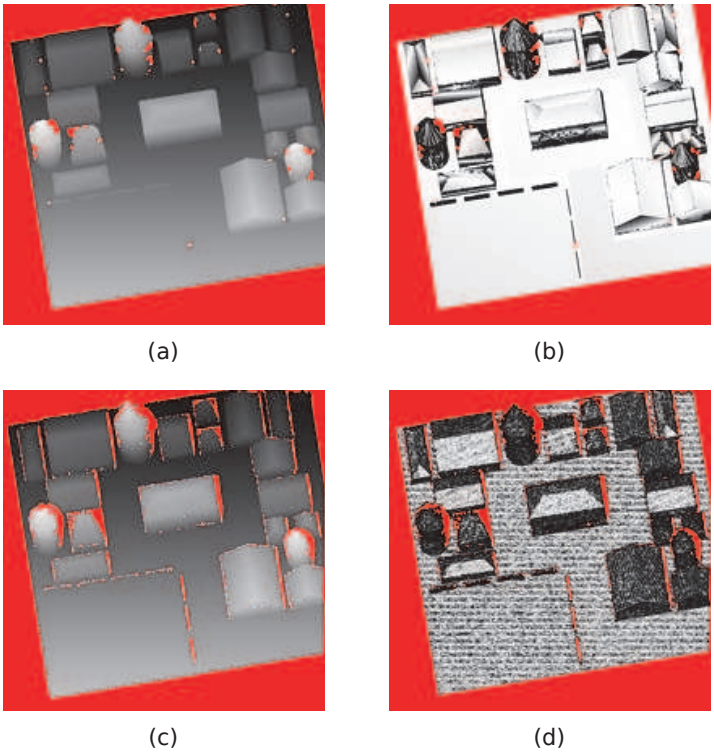


Figure 4.9: Interpolation of the missing points in disparity map obtained with Sabater's algorithm. (a) Interpolated disparity map by reprojection from our classification. (b) Error between the interpolation and the ground truth (the larger the error, the darker the image). (c) Interpolated disparity map with median filtering. (d) Error between the interpolation and the ground truth.

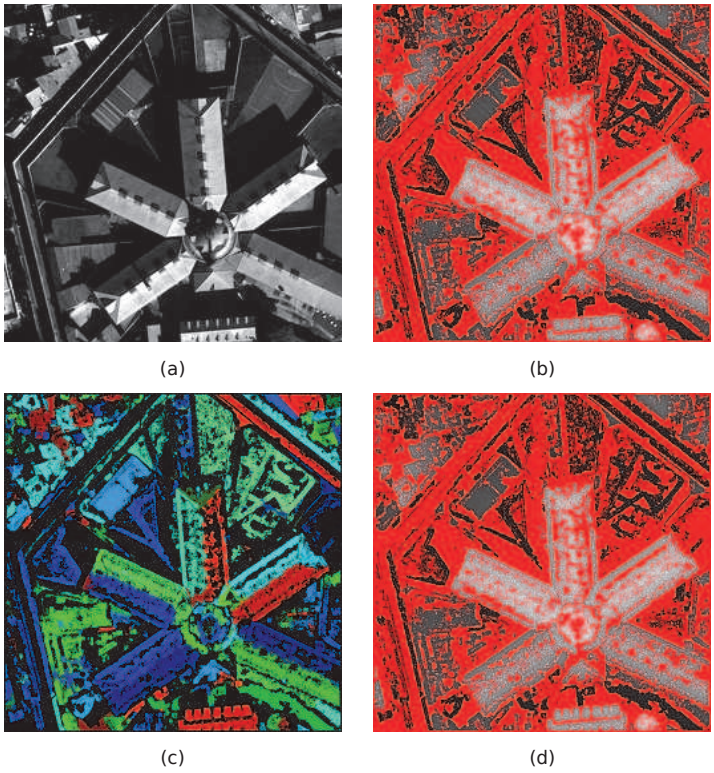


Figure 4.10: Toulouse PELICAN dataset. (a) Reference image. (b) Disparity map obtained with Sabater's algorithm. (c) Classification result with our algorithm. (d) Reprojection on the planes found with our classification

using the result of MicMac algorithm as input data gave poor results because the precision of MicMac's output was not precise enough.

Middlebury Venus and Sawtooth

As a last experiment, we ran our algorithm on the Middlebury[3] disparity maps obtained from Sabater's algorithm. We only tried here "Venus" and "Sawtooth" disparity maps because they are both piecewise planar. Our results are shown in Figure 4.11. The classification that

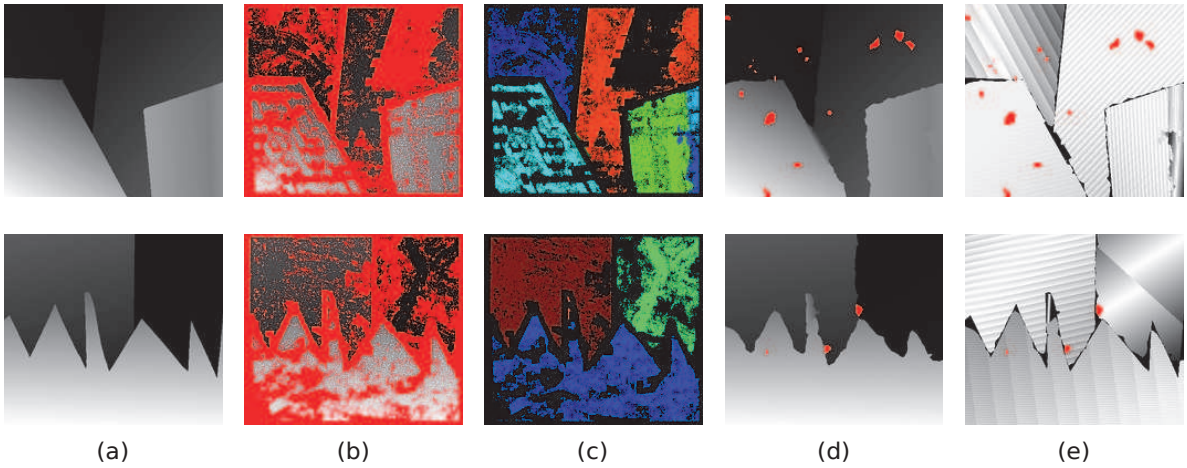


Figure 4.11: "Venus" and "Sawtooth" disparity maps. (a) Ground truth disparity map. (b) Disparity maps obtained with Sabater's algorithm. (c) Classification obtained with our algorithm. (d) Interpolated disparity map from our classification. (e) Error between interpolation and ground truth (the darker the image, the larger the error).

was obtained was more or less the one expected and the interpolation a lot similar to the ground truth. The error maps confirm all that since the remaining error is almost everywhere less than the quantification step of the ground truth. The black parts in the error images (large errors) are all localized near the planes delimitations. This result was expected since the unknown points were affected to the nearest group which do not necessarily corresponds with their actual plane. This suggests the use of a contour detection method to correct these misclassification. This will be the topic of the next chapter of this thesis.

4.4 Weighted planes

We now propose two modifications of our algorithm to see if the *a priori* knowledge on the point precision can be used for a better classification or a better plane estimation:

- Use the point precision as the distance rejection threshold in Algorithm 5 instead of using the automatic estimation.
- Weight the plane computations according to this precision.

[3] www.vision.middlebury.edu/stereo

In her Ph.D. thesis, Neus Sabater gave a formulation of the errors committed during the computation of disparity maps. In particular, she gave an expression of the variance of the main error term which is a function of the image noise variance σ^2:

$$\mathrm{Var}(\mathcal{E}_{\mathbf{x_0}}) = 2\sigma^2 \frac{\int [\varphi(\mathbf{x}-\mathbf{x_0}) u_x(\mathbf{x}+\varepsilon)]_N^2 d\mathbf{x}}{\left(\int \varphi(\mathbf{x}-\mathbf{x_0}) u_x(\mathbf{x}+\varepsilon)^2 d\mathbf{x}\right)^2} \tag{4.5}$$

where $u(\mathbf{x})_x$ is the derivative according to the x axis of the image at point $\mathbf{x}$, ε is the disparity, and φ is the smooth weighted window used for correlation. From Eq. 4.5, one can see that the error variance will be low in well contrasted regions (high gradient values) and high in poorly contrasted regions (low gradient values). However, in presence of adhesion, this error estimation does not reflect well what happens. Indeed, since adhesion happens in presence of a highly contrasted edge or texture, the error variance estimation from Eq. 4.5 is very low but the actual error can be very high because of the adhesion.

In order to still be able to use this error estimation in our experiments, we propose to use the RAFA algorithm (Rectification de l'Adhérence par Fenêtre Adaptative) [Blanchet et al., 2011] to compute disparity maps since it is made to avoid adhesion artifact. However, the cost is a lower quality of the disparity maps.

4.4.1 RAFA algorithm description

RAFA algorithm [Blanchet et al., 2011] is a disparity map computation algorithm which aims to limit the effects of the adhesion artifact. The algorithm is based on a fine analysis of the correlation measure and on the observation that adhesion is among other things due to highly contrasted textures or edges. This is characterized by a peak in the values of the image gradient.

To avoid this during the disparity computation, the authors proposed to locally weight the classical window used for the correlation ρ with the inverse of the gradient of the image along the x direction u_x:

$$\varphi_{\mathbf{x_0}}(\mathbf{x}) = \frac{\rho_{\mathbf{x_0}}(\mathbf{x})}{(u_x(\mathbf{x}+\varepsilon(\mathbf{x})))^2} \tag{4.6}$$

The effect of this weight is a noticeable reduction of the adhesion in regions where the disparity is continuous.

We ran RAFA algorithm on the following three stereo pairs: the simulated pair of Toulouse St-Michel, the "Bergerie" pair and the "Village" pair. The results of the algorithms, the expected error from Eq. 4.5 and the actual error to the ground truth are shown in Figure 4.12. Note that the predicted error do not completely match the actual one. For the Toulouse disparity map, the predicted error is a little pessimistic and the real error is a little noisier. However, the regions of low and high error seem to match.

For the other two maps, the prediction is not the same as the observed error. As explained before in the "Bergerie" experiment, the real error is stronger in some regions because the block matching approach does not take into account the 3D geometry of the observed scene. However, when this bias is removed, one can see that the error estimation and the real error look similar. This means that for each considered planar region, the error prediction gives an information on where the disparity computation should be less precise which is what we need here.

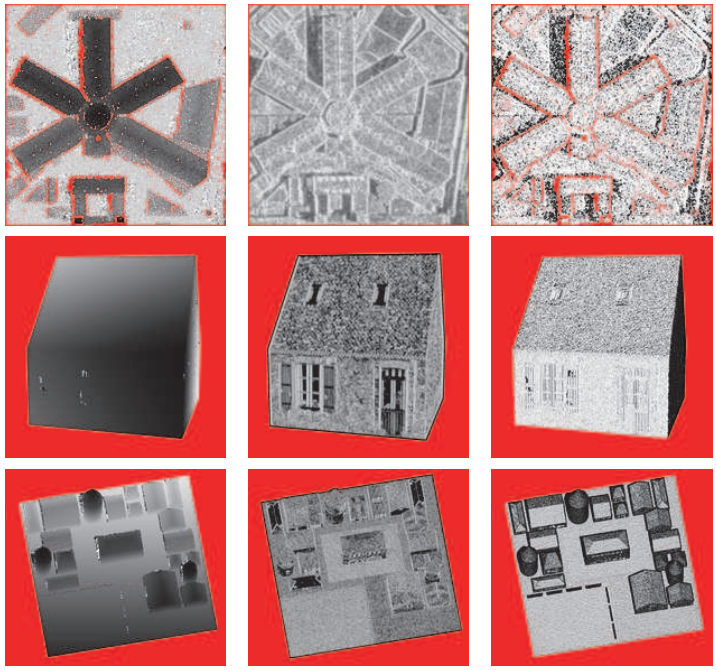


Figure 4.12: Disparity obtained with RAFA algorithm. From top to bottom: Toulouse, "Bergerie" and "Village". (a) Disparity map. (b) Precision estimated from Eq. 4.5. (c) Error to the ground truth.

4.4.2 Experiments

As a first experiment, we try to use the error prediction as the rejection threshold at each point in our algorithm. The classifications obtained are shown in Figure 4.13. One can see that the error prediction is not well adapted to the Toulouse disparity map because it is overestimated. The results obtained with the classical approach are not completely satisfactory but this is mostly due to the fact the disparity map is not very precise.

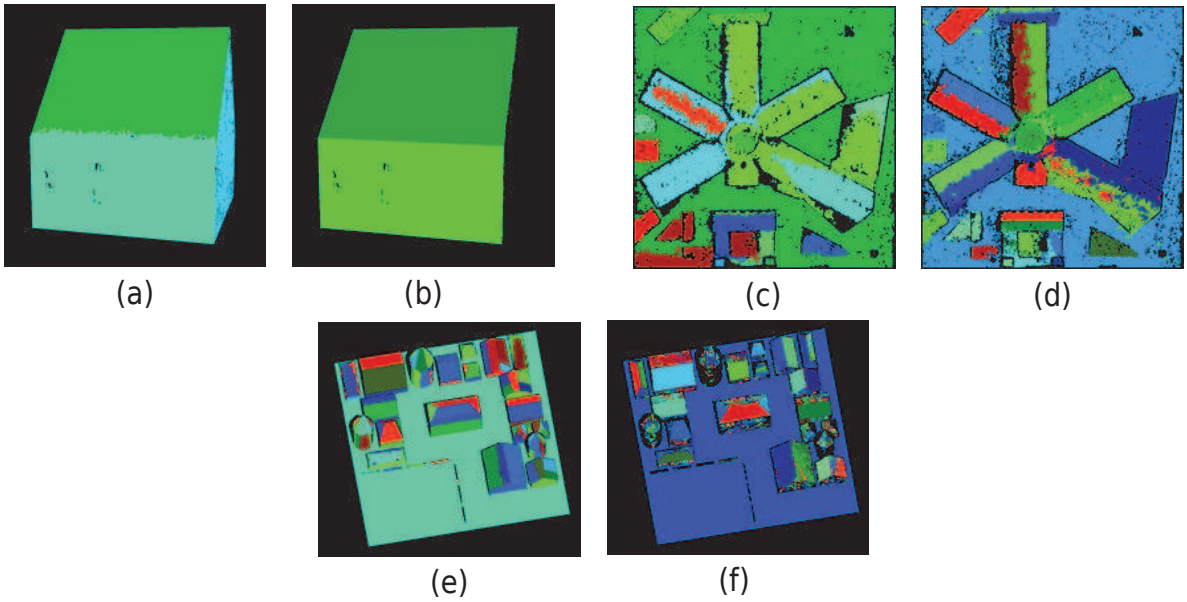


Figure 4.13: (a), (c) and (e): classification obtained using the error prediction. (b), (d) and (f): result using the classical approach.

We now propose to observe the effect of using weights in the computations of the classified planes. The goal here is to attenuate the importance of the less precise points to improve the plane estimation. We therefore decide to use the following weight at each point:

$$w(\mathbf{x}_0) = \frac{1}{0.01 + \sqrt{\mathrm{Var}(\mathcal{E}_{\mathbf{x}_0})}} \tag{4.7}$$

where $\mathrm{Var}(\mathcal{E}_{\mathbf{x_0}})$ is the predicted error variance given by Eq. 4.5.

We tried this computation on the three different disparity maps and used two different classifications each time: the one obtained from the ground truth disparity map as an input to our algorithm and the one obtained by applying our algorithm on the real disparity map. In the latter case, to avoid misclassification errors which are not related to the disparity map error, we also tried to measure the reprojection error after the rejection of the most error prone points (points for which the reprojection error is more than 0.5 pixels). The values of the RMSE (Root Mean Square Error) between the projected points and the ground truth in each situation are given in Table 4.1.The results show that the weighted estimation and the least square estimation give similar results which tends to favor the least square estimation since it requires less computations.

	GT classification			Disparity classification			Filtered classification		
	Init	Classic	Weighted	Init	Classic	Weighted	Init	Classic	Weighted
Toulouse	0.2610	0.0338	**0.0323**	0.1593	0.1353	**0.1353**	0.1416	**0.1283**	0.1283
"Bergerie"	0.6537	0.0396	**0.0360**	0.5248	0.4843	**0.4843**	0.0310	**0.0169**	0.0169
"Village"	3.6728	**1.6288**	1.6543	1.6435	**0.7411**	0.7487	0.0595	**0.2843**	0.2851

Table 4.1: Root Mean Square Error in pixels between the reprojected disparity and the ground truth.

4.5 Conclusion

In this Chapter, we completed the experiments of Chapter 3 with other data. Using data with a non-homogeneous error distribution pointed out the need of a correction. With this corrected algorithm, coherent segmentations were obtained on the different dataset. Moreover, by reprojecting the points on the segmented planes we were able to reduce the error of the disparity maps, which tend to confirm the quality of the segmentation. This segmentation was also used to interpolate points with an unknown disparity in what seems to be a coherent way. However, so far the missing points are just associated to the same group as their nearest known point which produce errors near the plane delimitation. In the next Chapter, a solution will be proposed to overcome these errors by finding a proper separation for each pair of plane in the classification. At last, the experiments on the error prediction showed that the plane estimation was not significantly improved by weighting the points with this error. Using an error estimation may then be seen as an overkill compared to our original method seems similar results were obtained with less computations.

Chapter 5

Contour Detection

Contents

Résumé: La méthode introduite dans les chapitres précédents permet de segmenter une carte de disparités (ou des cartes de profondeur laser) en régions où un modèle planaire peut être appliqué. Toutefois, la segmentation obtenue peut être imprécise au niveau des frontières entre plusieurs régions. Cette imprécision de classification est causée par différents facteurs variant selon les données utilisées. Pour le cas des cartes de disparités, cela est par exemple dû à:

- un non référencement des points c'est-à-dire que soit la disparité n'a pas été calculée en un point donné, soit que le point ne fait pas parti de la segmentation plane par morceaux;
- un mauvais calcul de la disparité dû par exemple à l'effet d'adhérence;
- l'incertitude sur la disparité faisant qu'un point peut potentiellement appartenir à plusieurs régions en même temps.

Dans ce chapitre, nous introduisons une nouvelle approche pour calculer les séparations entre deux régions planaires en prenant en considérations les différents points exposés précédemment. On se servira pour cela à la fois de l'information fournie par les images utilisées pour calculer la carte de disparités mais aussi de l'information fournie par le modèle plan de chaque région et l'interpolation qui en résulte selon l'endroit où l'on suppose que s'arrête chaque région.

Ce calcul a de nombreuses applications dont, entre autres, la correction de l'adhérence, l'interpolation des points manquants ou encore la vectorisation de la carte de disparités.

Abstract: The method previously described in the former chapters provides a segmentation of disparity maps (or range images) into regions where a planar model can be applied. However, the resulting segmentation may be imprecise near regions boundaries. These misclassifications are caused by various elements that depends on the data. In the case of disparity maps, this is for instance due to:

- the unknown value of the disparity at one point or its non-classification during the segmentation process;
- an error in the computation of the disparity that may be due to adhesion;
- the uncertainty on the disparity computation that makes it possible to be associated to several planar groups.

In this chapter we introduce a new approach to compute the boundary between two planar regions that takes into account the points previously exposed. We will use the information given by the images of the stereo pair as well as the information given by the planar model of each region which allows the interpolation of missing data depending on where we suppose that region boundaries are.

This computation has a lot of applications including adhesion correction, interpolation of missing points or disparity map vectorization.

5.1 Introduction

In the previous Chapters, two approaches were proposed to achieve the piecewise-planar segmentation of a disparity map. These methods were both simply based on the 3D information given by the map and thus could be applicable to other type of range images. However, since the belonging to a plane is made up to a distance threshold, there may be an ambiguity in the segmentation when a point is distant of less than this threshold for several planes. Moreover, if we consider that the input disparity maps were computed with a block matching approach (such as the one proposed by [Sabater, 2009]), the data points may suffer from the fattening artifact (see Chapter 1 for more details on this artifact). The consequence of this artifact is the dilatation of some objects in the disparity map which can then be transfered to our final segmentation. At last, in methods like [Sabater, 2009], a rejection criterion is defined to avoid potentially miscalculated points. As a result, the disparity map may not be completely dense and so is our segmentation from that. All of this suggests the use, at some point, of the information given by the two images of the stereo pair to refine our piecewise planar segmentation and define it at all points of the grid.

5.1.1 Previous work on adhesion correction

Several authors proposed approaches to correct or suppress adhesion during the disparity map computation. A first way to deal with this artifact is to adapt the size and shape of the windows used for correlation. Reducing the size of the correlation window reduces the probability of having a discontinuity inside it. However, the smaller the windows are, the more the sensitive to noise the results are. [Okutomi and Kanade, 1993] proposes to adapt (in both size and shape) the window used at each point depending on the local variation of both intensity and disparity. [Lotti and Giraudon, 1994] argues that this solution gives good results as long as the discontinuities are already well localized. They propose to constrain the windows by the contours but still to keeping the window size constant. This however may imply a strong deformation of the windows in one of the directions. [Boykov et al., 1998] chooses an arbitrary window that varies at each pixel. Their results tend to be better than other classical correlation methods. However, the authors point out a systematic error that occurs when propagating information from highly-textured area to low-textured area. [Veksler, 2002] and [Veksler, 2003] propose to choose a window among windows of various size and shape, however this window selection needs much parameter tuning. At last, [Hirschmüller et al., 2002] proposes a two-step approach. Disparities are fast-computed a first time using windows of multiple support. Then, the disparities in regions with discontinuities are computed a second time with a split window.

Other existing methods set the size and shape of correlation windows but assign weights to each pixel of the window to improve results near edges [Prazdny, 1987], [Darrell, 1998], [Xu et al., 2002] or more recently [Yoon and Kweon, 2006]. [Delon and Rougé, 2007] and [Delon, 2004] analytically study the correlation measure and propose to correct adhesion using a so called *barycentric correction*. For a given correlation window, instead of associating the disparity found to the central pixel, it is associated to the weighted barycenter of the contributing points. The effect of this correction is an irregularly sampled disparity map with points concentrated around well contrasted edges. This correction is optimal when the compared patch contains only one edge with only one discontinuity in depth which is unfortunately not always the case.

Inspired from that, [Sabater, 2009] proposes to remove points that may suffer from adhe-

sion. The author first computes a corrected disparity map. First, for each correlation window, the disparity that is found is associated to the 25% points whose gradient orientation match best. As opposed to that, classical approaches associate the block disparity only to the central pixel, and the barycentric correction only to the block barycenter. In the final result, some of the points will be associated to several disparities and some others to none. Then, the median disparity is taken for each point (when it exists). From the corrected disparity map, the points risking adhesion can be found. Due to the influence of a window on points, these points are completed by all the points nearer than half a correlation window from them. The main drawback of this approach is that it removes the points suspected of adhesion instead of correcting them. A lot of information is then lost in the process.

Some other approaches match features instead of blocks of pixels and are therefore not sensitive to adhesion. However, this is often at the cost of a substantial reduction of the match density. [Schmid and Zisserman, 2000] proposes various methods to match individual line segments and curves. [Robert and Faugeras, 1991] matches cubic B-splines interpolation of 2-dimensional edges. [Musé et al., 2006] and [Cao et al., 2007] propose to use an *a contrario* framework to automatically match pieces of level lines. [Matas et al., 2004] matches homogeneous and stable regions but the results is still sparse. Apart from the possibly poor density of features, all these approaches may still suffer from adhesion. Indeed, all the features presented depend at some point of a neighborhood. As an example, even if the Laplacian extrema of the SIFT are very local, the feature descriptor involves an 8×8 pixel window.

To sum up, all the local methods correcting adhesion are not a 100% effective and may leave residual adhesion. In another hand, global methods for disparity computation do not suffer from adhesion but may propagate wrong information in the disparity map. The approach proposed by [Sabater, 2009] remove most of the adhesion points as well as other points and useful information for post-segmentation algorithms are lost in the process. The disparity map segmentation method we propose here, takes into account the possible adhesion artifacts and uses the additional information given by the computed planes to simulate disparity and adhesion.

5.1.2 Previous work on contour refinement in range segmentation algorithms

The contour delimitation and refinement problem was treated in different ways by several authors.

In some methods adapted to urban situations, the segmentation is made with a strong *a priori* on the shape of the buildings. In [Ortner et al., 2007] and [Lafarge et al., 2008b], a rectangle fitting approach is proposed as a pre-segmentation. Each building is roughly described by a set of rectangles and then refined with another approach. These methods produce a visually nice 3D model of data but are rather imprecise. Moreover, they are adapted to dense disparity maps.

In a similar vein, some authors use cadastral ground plans to simplify the building segmentation (*e.g.* [Vosselman and Dijkman, 2001], [Durupt and Taillandier, 2006], [Flamanc and Maillet, 2005]). These plans are first fitted to the range images, then parallelism and symmetry assumptions on the walls and roof separations are made to guide the final segmentation. However, as for the rectangle fitting approach, the final segmentation is usually not very precise since cadastral plans are not rarely adapted to the views.

Some other methods use a combination of 2D and 3D feature segments to define a polygonal enclosure to each planar facet defining a rooftop. In [Ameri and Fritsch, 2000], the segments

are mixed with intersection segments and the best configuration is chosen by thresholding. In [Bignone et al., 1996], a set of possible contours for a planar facet is defined by all the combination of segments. The most likely combination is then kept. In [Vallet and Taillandier, 2005], the 3D plane configuration is modified to fit a set of 2D segments with a Levenberg-Marquardt minimization.

In some methods, an initial piecewise-planar configuration is adjusted to better fit a dense disparity map. In [Taillandier and Deriche, 2004], a graph corresponding which vertices correspond to all the possible planes (including vertical ones) is computed. The most likely configuration is then chosen with a Bayesian approach. In [Brédif et al., 2008], an initial 3D polygon evolves with a kinetic framework to better fit the data.

At last, when no other information is available, some methods ([Schnabel et al., 2007a], [Chauve et al., 2010]) both use a graph-cut approach to extrapolate a set of primitives to missing points and chose where these primitives stop when the points are available. However, using graph-cut introduces new parameters which are hard to tune.

In this chapter, we propose a method that refines an initial piecewise planar segmentation especially where there may be some uncertainty. Considering two planes, we first define a region where the association of a point to one plane or the other may be wrong. Then we search for the best separation between these two planes within this ambiguity region. The separation is computed either by considering the intersection between the two planes (when it exists), either by finding a contour that fits to the strong gradients of the reference image, or at last by finding the contour minimizing the re-projection error of the reference image onto the second image. The decision between these three possible solutions is made using an *a contrario* criterion. As in the previous chapters, we try to avoid as much as possible the use of parameters. To do so, we limit our search to continuous polygonal contours. This moreover ensures the simplicity of the result. The last advantage is the possible analytic expression of the final contour.

The chapter is organized as follow. First, we will give a global description of the contour detection algorithm. Then, each detail of the different steps will be given in the following sections.

5.2 Boundary refinement between two planar groups

We now introduce a new method to refine the piecewise-planar segmentation of disparity maps introduced in the former chapters. As opposed to other segmentation approaches, our method assumes that the disparity map is not perfect. In particular, since the input disparity maps were computed using a block matching approach, we assume that it may suffer from adhesion. This artifact is indeed the source of segmentation errors since foreground objects tend to be fattened.

Assuming the presence of adhesion allows us to correct it. The main advantage of our approach compared to other adhesion corrections methods (see previous section), is to know the two possible models that can be used near an object edge. Knowing this, disparities can be removed and interpolated. The only thing that remains to know is where to stop using one model and start using the other one.

To our knowledge, our segmentation algorithm is the first one to take into account the possible errors in the disparity maps. Since methods like [Igual et al., 2007] and [Facciolo and Caselles, 2009] are based on an image segmentation, they are likely to be robust to adhesion. However, this image segmentation is also a source of errors since the piecewise-

planar segmentation does not always match an image segmentation. This is indeed the case when an edge is not contrasted enough in the image or when for instance a spherical surface has to be approximated by several planes.

Overview of our method

Let's now give a global description of our algorithm. Given two planar regions of a disparity map segmented with Algorithm 5 and 6 (see Chapter 3 and 4), we aim to refine the separation between these two regions. The separation is refined in a zone where the disparity points are either unknown, risking adhesion or possibly associated to both model plane according to the distance threshold that is used.

Generally speaking, the search for an optimal separation between two planes π_1 and π_2 can be viewed as an energy minimization problem (or a maximization depending on the energy that is used):

$$\gamma_{1,2} = \arg\min_{\gamma \in \Gamma} E(\mathcal{R}_{1,2}, \gamma) \tag{5.1}$$

where $\gamma_{1,2}$ is the optimal separation that is found, Γ is a set of possible separations, and $\mathcal{R}_{1,2}$ is the research region and therefore the target set of the parametric curve $\gamma_{1,2}$.

For instance, if we consider the Snakes energy [Kass et al., 1988], then Γ is the set of parametrical curves of $[0,1] \mapsto \mathcal{R}_{1,2}$ twice differentiable, and $E(\mathcal{R}_{1,2}, \gamma)$ can be defined as:

$$E(\mathcal{R}_{1,2}, \gamma) = \int_0^{L(\gamma)} g(|\overrightarrow{Du}(\gamma(s))|)ds + C \int_0^{L(\gamma)} (a + |\overrightarrow{Curv}(\gamma(s)))|ds \tag{5.2}$$

where u is in our case the reference image of the stereo pair, a and C two parameters, $\overrightarrow{Curv}(\gamma(s)) = \overrightarrow{\gamma}_{ss}(s)$, and g is a positive and decreasing function. However, since the energy increase when the curve gets longer, minimizing it tends to shrink the snake curve.

A more recent and simpler snake approach proposed in [Kimmel and Bruckstein, 2003] avoids this length dependent energy by maximizing the average contrast:

$$E(\gamma) = \frac{1}{L(\gamma)} \int_0^{L(\gamma)} g(\overrightarrow{Du}(\gamma(s) \cdot \overrightarrow{n}(s)))ds \tag{5.3}$$

where g is this time an increasing function and positive function usually $|\cdot|^\alpha$. However, as pointed out in [Desolneux et al., 2008a] with their experiments, the choice of this remaining parameter α affects a lot the result since it partly defines the regularity of the result.

The construction of our method was driven by two goals:

- To avoid as much as possible the use of parameters or at least to be able to set them using a simple decision criterion.
- To find an energy that properly describes our situation and to handle as much as possible all the information that is available: 3D information and image information.

Constrained set of solutions Γ

As a choice for Γ, we used the set of polygonal curves $[0,1] \mapsto \mathcal{R}_{1,2}$.

The first interest of doing so is that the regularization of the solution γ is self imposed by the number of vertices that are used to define the curve. This number can be set naturally as it will be exposed later by dividing the search regions.

Another advantage of using the set of polygonal curves is that the solution γ can be expressed analytically at any point.

Energy choice

Different situations may arise when seeking to localize the curve that separates two planar patches of a surface or DEM. Each situation calls for different ways to look for the optimal separation by defining a specific energy for each case. In addition we need an algorithm to decide in which situation we are. Our approach, follows the Occam's razor principle, and consists of progressing from the simplest possible explanation to more complex ones. If a simple solution passes a coherence test, it is kept. Otherwise more complex solutions are tested until one of them is kept as the final solution.

In our case, three different situations have been identified:

1. **Plane Intersection.** In this case the surface is continuous and both neighboring patches are sufficiently large and camera-facing to be detected as independent groups. In this case the intersection between the two planes is the natural location of the intersection. We identify this situation as coherent with the measured data if the plane intersection lies within the uncertainty region AND the plane intersection model provides a sufficiently photo-consistent explanation of the captured images.

2. **Very different planes ; Discriminant photo-consistency.** In this case the surface presents a discontinuity between the two surface patches or the angle between the two continuous planes is very sharp. This may be due to a real discontinuity, or to the fact that the joining surface patch is either occluded or non-detected because it is too small. In this case we try to locate the border between both surface patches by maximizing photo-consistency between both images in the pair, as long as this criterion provides sufficiently discriminative evidence. The photo-consistency between the reference image and the second image using each plane models π_1 and π_2 is then likely to be very different. In [Vu et al., 2009], the authors made a similar reasoning and used the photo-consistency to refine an existing mesh obtained from multi-view stereo data.

3. **Discontinuous surface; Non-discriminant photo-consistency.** If the surface is discontinuous but photo-consistency is not discriminative enough we are left with a mere heuristic to locate the surface discontinuity at its most probable position. As we shall deduce later from the properties of Lambertian surfaces, a sensible heuristic can be formulated as a gradient energy minimization. Computing this energy is then similar to computing the geodesic distance as introduced in [Yatziv and Sapiro, 2006], [Bai and Sapiro, 2007]. A similar approach was also used in [Facciolo and Caselles, 2009] to compute geodesic Voronoi cells as a prior segmentation to their algorithm for piecewise-planar segmentation of sparse disparity maps.

In the first situation, the location of the border between planar patches is trivial and reduces to standard affine 3D geometry. The main difficulty lies in the latter two, however, require a more sophisticated energy to be minimized as stated in Equation (5.1). In the general case the problem leads to non-convex optimization in a quite large domain, with several local minima. Such can be observed easily if one consider the third energy functional. Two distinct highly contrasted level lines in the search region will produce two different minima of the energy, which implies that the problem is non-convex. A similar reasoning can be applied to the photo-consistency energy.

In order to make the problem more tractable, we looked (in each of these two cases) for sensible energies that can be minimized numerically by fast algorithms like dynamic programming, without tricky convergence issues. Such is done by dividing $\mathcal{R}_{1,2}$ into small subregions

where the searched separation is likely to be linear. Note that this particular case in not adapted to graph-cuts since they do not ensure a single continuous separation between the points of $\mathcal{R}_{1,2}$.

The next three sections are devoted to the derivation that has been followed to obtain suitable energies in both cases (2 and 3), the hypothesis testing methodologies that are used to decide between the two possible energies, the resolution with the dynamic-programming framework, and on how to solve the trade-off between curve complexity and data-fitting. Next section focus on the definition of a search region for the contour between two planes $\mathcal{R}_{1,2}$. The section after, gives details on the computation of the different energies and on how to decide which energy best describes the situation encountered. The adaptation of the problem to make the energy minimization by dynamic programming possible is then explained in a last section.

A global description of the method is given in Algorithm 7.

Algorithm 7: Plane separation refinement

Data:
$\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$, a partition of the disparity map into planes
Result:
$\mathcal{G}_\mathcal{R}(\mathcal{R}, \mathcal{Q})$ the adjacency graph of each region
$\forall(\mathcal{R}_i, \mathcal{R}_j) \in \mathcal{Q}$, $\gamma_{i,j}$ the polygonal curve that best separates the two planes

```
1  begin
2      Compute the adjacency graph of regions G_R(R, Q).
3      foreach (R_i, R_j) ∈ Q do
4          Compute the search region for the contour R_{i,j}.
5          Divide R_{i,j} into subregions (R_{i,j,k})_{k=1..M} to simplify the problem
6          foreach R_{i,j,k} do
7              Find all the possible segments
8          end
9          Find γ_{i,j}, the sequence of segments minimizing E(R_{i,j}, γ), using dynamic
           programming
10     end
11 end
```

5.3 Search regions

5.3.1 Adjacency graph of regions $\mathcal{G}_\mathcal{R}$

The first step of our method is to compute the adjacency graph of regions $\mathcal{G}_\mathcal{R}$ of our disparity map segmentation. To deal with potential holes in disparity maps, such is achieved by computing the Voronoi map of each segmented region. Neighboring regions are then defined as regions whose Voronoi cells share a mutual edge.

One of the main advantage of using disparity maps as data is that the points stand on a regular grid $\mathcal{D}$. The computation of the Voronoi cells can then be considerably reduced using the 3-4 Chamfer distance [Borgefors, 1986] (which only require two explorations of the disparity map).

Since the 3-4 Chamfer distance is an approximation of the L_2 one, this may produce small errors in the Voronoi segmentation. This however does not really matter since the goal here is not the precision but just to find out the neighboring regions to construct $\mathcal{G}_{\mathcal{R}}$. A similar approach was chosen in [Ameri and Fritsch, 2000] in their segmentation algorithm. Figure 5.1 illustrates what was said before with an example of a real disparity map segmentation.

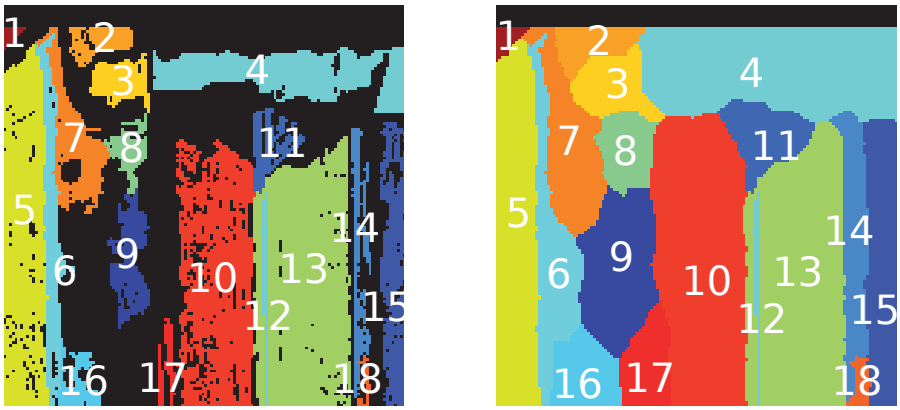


Figure 5.1: Neighboring regions. Left: initial segmentation, the black parts represents holes in the segmentation. Right: Voronoi cells of each region. From the Voronoi segmentation, finding the neighboring regions becomes obvious. In this case the edges of the adjacency graph of regions are: $\{(1-5);(1-7);(2-3);(2-4);(2-7);(3-4);(3-7);(3-8);(3-10);(4-10);(4-11);(4-13);(4-14);(4-15);(5-6);(5-7);(6-7);(6-9);(6-16);(7-8);(7-9);(8-9);(8-10);(9-10);(9-16);(9-17);(10-11);(10-13);(10-17);(11-13);(12-13);(13-14);(13-18);(14-15);(14-18);(15-18);(16-17)\}$

5.3.2 Search region for the separation between two planes

Now that the neighboring regions have been defined, we need to define where the separation between two regions can be found. As said previously, we suppose that separation between two regions can evolve wherever the segmentation of the two regions may be wrong.

Let's now draw our attention on two neighboring regions that we will call $\mathcal{R}_1$ and $\mathcal{R}_2$. Let's note $\mathcal{R}_{V_1}$ and $\mathcal{R}_{V_2}$ their associated Voronoi cells. We define Ω_0 as the common edge border of $\mathcal{R}_{V_1}$ and $\mathcal{R}_{V_2}$.

$$\Omega_0 = \left\{ \mathbf{x} \in \mathcal{R}_{V_1} \bigcup \mathcal{R}_{V_2} / d(\mathbf{x}, \mathcal{R}_1) = d(\mathbf{x}, \mathcal{R}_2) \right\} \tag{5.4}$$

where $d(\mathbf{x}, \mathcal{R}_i) = \min_{\mathbf{x}_i \in \mathcal{R}_i} ||\mathbf{x} - \mathbf{x}_i||_2$. Note that when $\mathcal{R}_1 = \mathcal{R}_{V_1}$and $\mathcal{R}_2 = \mathcal{R}_{V_2}$, then Ω_0 is the initial separation between the two regions $\mathcal{R}_1$ and $\mathcal{R}_2$.

From Ω_0 we can now define the points that were possibly misclassified by the segmentation algorithm. This research region, $\mathcal{R}_{1,2}$, is defined by:

- the unknown points between the two regions (no classification label). Since we are only interested in the unknown points that may be related to the contour, we define two sets:

The points of $\mathcal{R}_{V_1} \setminus \mathcal{R}_1$ closer to $\mathcal{R}_2$ than any other set $\mathcal{R}_i \in \mathcal{R}$, $i \neq 1, 2$ and reversely:

$$\begin{array}{l} \Omega_{1\to 2} = \{\mathbf{x} \in (\mathcal{R}_{V_1} \setminus \mathcal{R}_1) \ / \ d(\mathbf{x}, \mathcal{R}_2) < d(\mathbf{x}, \mathcal{R}_i),\ \forall \mathcal{R}_i \in \mathcal{R},\ \mathcal{R}_i \neq \mathcal{R}_1, \mathcal{R}_2\} \\ \Omega_{2\to 1} = \{\mathbf{x} \in (\mathcal{R}_{V_2} \setminus \mathcal{R}_2) \ / \ d(\mathbf{x}, \mathcal{R}_1) < d(\mathbf{x}, \mathcal{R}_i),\ \forall \mathcal{R}_i \in \mathcal{R},\ \mathcal{R}_i \neq \mathcal{R}_1, \mathcal{R}_2\} \end{array} \tag{5.5}$$

- the points that could have been associated to both regions $\mathcal{R}_1$ and $\mathcal{R}_2$ by the segmentation algorithm. This is due to the fact that a point is associated to a plane if its distance to the plane along the z-axis is less than the distance threshold τ_z. A point may then be valid for several planes according to this criterion. The set is then defines as:

$$\Omega_2 = \left\{\mathbf{x} \in \mathcal{R}_1 \bigcup \mathcal{R}_2 / \ |z(\mathbf{x}) - z_{\pi_1}(\mathbf{x})| < \tau_z \text{ and } |z(\mathbf{x}) - z_{\pi_2}(\mathbf{x})| < \tau_z\right\} \tag{5.6}$$

- the points for which the disparity computation may have been corrupted by the adhesion artifact.

Since $\mathcal{R}_{1,2}$ should not contain missing interior points of $\mathcal{R}_1$ and $\mathcal{R}_2$, $\Omega_{1\to 2}$ and $\Omega_{2\to 1}$ are reduced to their subsets that are connected Ω_0. Moreover, Ω_2 is reduced to its subsets that are either connected to $\Omega_{1\to 2}$, $\Omega_{2\to 1}$ or Ω_0. We note $\Omega'_{1\to 2}$, $\Omega'_{2\to 1}$ and Ω'_2 these reductions of $\Omega_{1\to 2}$, $\Omega_{2\to 1}$ and Ω_2. In the discrete case, this connectedness can be defined for example using the 4-connectivity. This can be computed using a greedy approach. For clarity reasons, we will note:

$$\Omega_{1,2} = \left(\Omega_0 \bigcup \Omega'_{1\to 2} \bigcup \Omega'_{2\to 1} \bigcup \Omega'_2\right) \tag{5.7}$$

From the description of the adhesion artifact (see Chapter 1), one can see that the points closer than half a correlation window ($W/2$) to an edge in the disparity map may have been corrupted by adhesion. Given two objects in a 3D scene, the segmentation of the front object may then have been dilated of a correlation window W. This means that all the points nearer than half a correlation to a set border may be wrong and should be removed from the classification if they belong to a plane in front of another one. To take that into account during the search of the plane separation, we define the search region as the dilatation of $\Omega_{1,2}$ by a correlation window whenever the points belong to the front region:

$$\mathcal{R}_{1,2} = (\Omega_{1,2} \oplus W_1) \bigcup (\Omega_{1,2} \oplus W_2) \tag{5.8}$$

where

$$\begin{array}{l} (\Omega_{1,2} \oplus W_1) = \Omega_{1,2} \bigcup \{\mathbf{x} \in \mathcal{R}_1 \ / \ z_{\pi_1}(\mathbf{x}) > z_{\pi_2}(\mathbf{x}) \text{ and } d(\mathbf{x}, \Omega_{1,2}) < W/2\} \\ (\Omega_{1,2} \oplus W_2) = \Omega_{1,2} \bigcup \{\mathbf{x} \in \mathcal{R}_2 \ / \ z_{\pi_1}(\mathbf{x}) < z_{\pi_2}(\mathbf{x}) \text{ and } d(\mathbf{x}, \Omega_{1,2}) < W/2\} \end{array} \tag{5.9}$$

An illustration of the construction of a search region is given in Figure 5.2 and an example of such construction on real data is shown in Figure 5.3.

5.4 Energy choice.

As discussed in Section 5.2, the second step of the algorithm is to choose an energy functional in Eq. (5.1) to quantify how good a separation between two planes is. Three different solutions, each corresponding to a different situation, are proposed:

1. computing the intersection between planes,

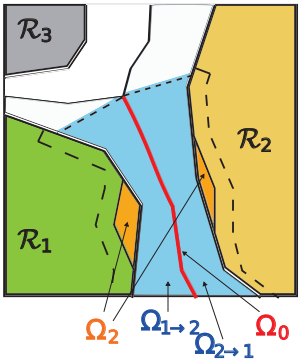


Figure 5.2: Construction of a search region. $\mathcal{R}_1$ and $\mathcal{R}_2$ are the two regions for which we want to refine the separation. The search region $\mathbb{R}_{1,2}$ is defined as the dilatation by a correlation window of Ω_0 (edge of the two Voronoi cells, in red here), Ω_1 (unknown points, in blue) and Ω_2 (points possibly belonging to both planes, in orange). The dilatation is delimited by the dash lines.

2. computing an energy quantifying the photo-consistency of the reference image and the second image by applying at each point the two possible disparities implied by the two possible planes,

3. computing an energy that is minimal when the selected contour correspond to a maxima of the gradient of the reference image

To decide which situation should be considered in priority, an *a contrario* criterion is defined.

This section is organized as follow. First, each of the two energies are described in details (gradient energy for case 3, and L_2 re-projection energy for case 2, see Sections 5.4.1 and 5.4.2). Then we address the problem of how to decide between cases 2 and 3 by means of an *a contrario* methodology (see Section 5.4.3).

5.4.1 Gradient energy

In the absence of any further evidence we can use the following commonly accepted heuristics: DEM discontinuities coincide to a large extent with gray-level discontinuities. There is not a perfect coincidence but a large correlation and an almost inclusion relationship of the DEM's topographic map within the luminance's topographic map.

The reason is the following:

- *Most common case: different objects.* A surface discontinuity is most often due to an occlusion (a close object that hides a more distant one). The object in the foreground and the one in the background have no reason to have similar color, or texture. And even if they do have the same texture, it is extremely unlikely that the texture patterns between foreground and background coincide perfectly. This leads to high luminance gradients at the border separating the foreground object to the background one.

  Another exception happens whenever these high luminance gradient are not the highest gradients in the considered region. In this special case, finding the maximal gradient will fail, but luckily, computing the photo-consistency between images should work.

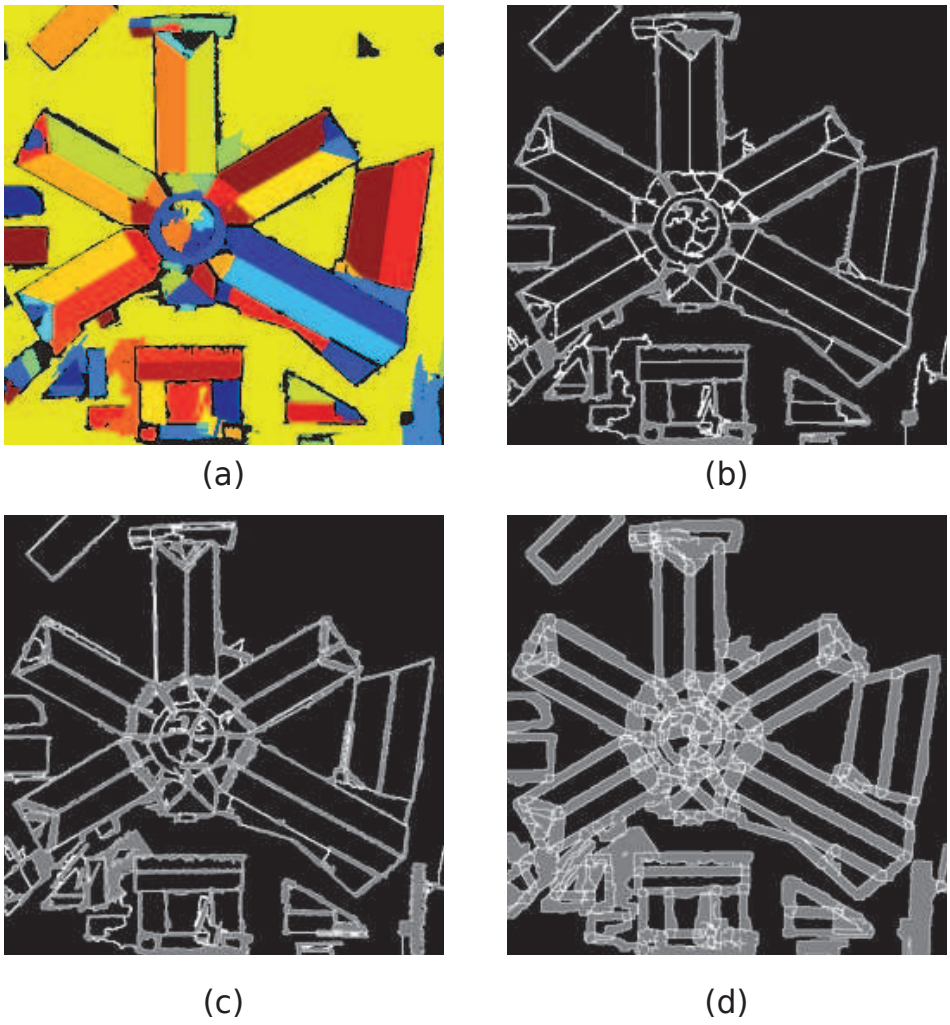


Figure 5.3: Example of construction of the search on Toulouse dataset. (a) Input classification. (b) Region formed using the points with unknown disparities. (c) Regions without intersection points. (d) Regions without adhesion points (dilatation of (b) and (c))

- *Similar objects, different luminance.* Now assume the less probable case where both objects have similar color (albedo) and texture. To simplify the reasoning consider the case where the surfaces are Lambertian. In that case the observed luminance is directly proportional to the cosines of the angles between the surface normal and the light source, and between the surface normal and the gazing direction. Hence in all probability, even if the albedo of both surface patches is the same, the observed luminance will be in all probability different between both patches.
- *Similar objects, similar luminance.* Still, if both surface patches are parallel (but at different depths) or in a few pairs of (non-parallel) surface orientation combinations, the observed intensity may be the same and no significant luminance gradient is observed in the image. This is a highly unlikely situation, that may occasionally happen. Luckily, when it does happen, most often the surface is continuous and the intersection criterion (see Section 5.2) holds valid or when it does not, the photo-consistency between images is likely to be valid.

Henceforth we shall assume that when all else fails, plane separation is trust-worthily provided by the most regular and contrasted luminance edge that goes through the uncertainty region $\mathcal{R}_{1,2}$.

The gradient energy functional that we adopt here takes advantage of this hypothesis by computing the gradient of an image orthogonally to a given contour:

$$E_{\nabla}(\mathcal{R}_{1,2},\gamma) = -\int_0^1 \left|\overrightarrow{\nabla} u(\gamma(s)) \cdot \overrightarrow{\dot{\gamma}}(s)\right| ds \tag{5.10}$$

$E_{\nabla}(\mathcal{R}_{1,2},\gamma)$ is highly negative if the contour γ follows a well contrasted edge in the image and close to 0 otherwise.

This energy is the core of several well known segmentation tools that are based on geodesic distance computation [Bai and Sapiro, 2007]. It was also used in [Facciolo and Caselles, 2009] as the base of their planar segmentation algorithm with geodesic Voronoi cells. In all those works the minimizing curve γ is used to defined a geodesic distance between points, where the weighted metric ensures that geodesics hardly go though contrasted edges.

Figure 5.4 shows an experiment done on the Toulouse Data set using only the gradient Energy as described before. One can clearly see that this energy is not adapted in a lot of cases and fails when the most contrasted curve does not correspond to the actual 3D separation. This suggests the use of other energies (such as the photo-consistency or L_2 re-projection described in the next section) or the use of purely geometric criteria such as plane intersections.

5.4.2 L_2 re-projection error energy (photo-consistency energy)

Using the gradient energy functional gives good results in most situations. However, it will fail whenever a strong gradient in $\mathcal{R}_{1,2}$ do not match the edge between planes. We therefore introduce another energy functional based on the computation of the re-projection error to deal with these situations.

Let's first define the following norm and scalar product:

$$\begin{cases} < u, v >_{\mathcal{T},A} = \displaystyle\int_{\mathbf{x}\in A} u(\mathcal{T}(\mathbf{x})) \cdot v(\mathbf{x}) \\ ||u||_A = \sqrt{\displaystyle\int_{\mathbf{x}\in A} u^2(\mathbf{x})} = \sqrt{< u, u >_{\mathcal{I},A}} \end{cases} \tag{5.11}$$

Figure 5.4: Resulting contours between planes using the gradient energy. (a) Input disparity map classification. (b) Zoom of the gradient of the reference image into the search regions. (c) Resulting contours. This energy seems very adapted in cases where the 3D separation corresponds to a unique highly contrasted curve, but, however fails in cases where the gradient is not very contrasted or where the most contrasted curve is not the right one.

Where $u, v : \mathbb{R}^2 \longrightarrow \mathbb{R}$ are two functions that can be viewed as the reference and the secondary images, $\mathcal{I}, \mathcal{T} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ are two point transformation functions ($\mathcal{I}$ is the identity transformation) and $A \subset \mathbb{R}^2$ is a point set within two images.

Let's consider as before the case of two regions $\mathcal{R}_1$ and $\mathcal{R}_2$ associated to the planes π_1 and π_2 for which we wish to find the separation. Each curve $\gamma \in \Gamma_s$ splits $\mathcal{R}_{1,2}$ into two parts: $\mathcal{R}_{\gamma,1}$ which is associated to plane π_1 and $\mathcal{R}_{\gamma,2}$ which is associated to π_2.

We define the L_2 re-projection error energy as the grey level difference between the two images after applying the disparity induced by each planar model:

$$E_\varepsilon(\mathcal{R}_{1,2}, \gamma) = ||u \circ \mathcal{T}_\gamma - \tilde{u}||_{\mathcal{R}_{1,2}} \tag{5.12}$$

where

$$\mathcal{T}_\gamma : \begin{array}{c} \mathbb{R}^2 \mapsto \mathbb{R}^2 \\ \mathbf{x} \mapsto \left\{ \begin{array}{l} \mathcal{T}_{\pi_1}, \text{ if } \mathbf{x} \in \mathcal{R}_{\gamma,1} \\ \mathcal{T}_{\pi_2}, \text{ if } \mathbf{x} \in \mathcal{R}_{\gamma,2} \end{array} \right. \end{array} \tag{5.13}$$

and where $\mathcal{T}_\pi$ is the affine point transformation induced by plane π. The point transformation $\mathcal{T}_\gamma$ depends on the partition of $\mathcal{R}_{1,2}$ that is induced by γ and on the two planes π_1 and π_2. Modifying γ changes the partition and therefore the point transformation $\mathcal{T}_\gamma$. We then search for the curve γ that will minimize the error between the re-projection of reference image using $\mathcal{T}_\gamma$ and the secondary image. Note that a similar energy was used in [Vu et al., 2009] to adapt the mesh of a 3D scene to better fit to the corresponding stereo images.

Similarities with the correlation measure

The computation of disparity maps using a block matching approach is usually made by maximizing the normalized crossed correlation between the images. Let's consider a window $W_{\mathbf{x}}$ around point $\mathbf{x}$ and the translation point transformation $\mathcal{T}_{\mathbf{x}} : \mathbf{x} \in \mathbb{R}^2 \mapsto \mathbf{x} + \mathbf{t}$ where

$\mathbf{t} \in \mathbb{R}^2$. Then the normalized crossed correlation between two images u and $\tilde{u}$ around point $\mathbf{x}$ and for a translation vector $\mathbf{t}$ is given by:

$$\rho_{\mathbf{t}}(\mathbf{x}) = \frac{< u, \tilde{u} >_{\mathcal{T}_{\mathbf{t}}, W_{\mathbf{x}}}}{||u||_{W_{\mathbf{x}+\mathbf{t}}} \cdot ||\tilde{u}||_{W_{\mathbf{x}}}} \tag{5.14}$$

The translation vector $\mathbf{t}$ maximizing (5.14) is then defined as the disparity at point $\mathbf{x}$.

The L_2 re-projection error energy given by Equation (5.12) is pretty similar to the correlation measure (5.14). Indeed developing (5.12) we obtain:

$$E_{\varepsilon}(\mathcal{R}_{1,2}, \gamma) = ||u \circ \mathcal{T}_{\gamma}||^2_{\mathcal{R}_{1,2}} + ||\tilde{u}||^2_{\mathcal{R}_{1,2}} - 2 < u, \tilde{u} >_{\mathcal{T}_{\gamma}, \mathcal{R}_{1,2}} \tag{5.15}$$

Considering the first two terms of (5.15) as constants (which is not true since the first term depends on $\mathcal{T}_{\gamma}$) then minimizing (5.15) is equivalent to maximizing the non-normalized crossed correlation for a non constant point transformation $\mathcal{T}_{\gamma}$, that is $< u, \tilde{u} >_{\mathcal{T}, \mathcal{R}_{1,2}}$.

There are however two main differences with the standard correlation used for disparity computation:

- The correlation measure is used on a block region to compute the disparity of a single point whereas the re-projection error energy is computed for a all the points at the same time.
- In the correlation, the disparity is assumed to be constant in the neighborhood $W_{\mathbf{x}}$ (which is false for instance when the 3D points on a tilted plane or in presence of a discontinuity). In the re-projection error energy, each point is associated to its own disparity with supposedly correct model. The discontinuities in disparity are handled by the two possible model.

The two previous points are the reason of the adhesion artifact in the correlation measure (see Chapter 1). Assuming that the disparity in $\mathcal{R}_{1,2}$ can be represented by two possible planar models, and adapting the "window" A to best separate both models is what makes us robust to it.

Occlusions

The energy derived in Eq. 5.12 was somewhat naive, since it ignores occlusion artifacts. In stereo-vision, some objects are occluded in some views but not in others (see Figure 5.5 for an illustration of that). This must be taken into account for a realistic model. When the images are rectified in epipolar geometry, this occlusion is observed only along horizontal lines.

When one computes an energy based on the similarity between two images such as the re-projection error energy, the occlusion introduces errors since the energy compares objects that are not present in both images. This can therefore make the energy minimization fail.

To avoid these occlusion errors, we propose to compute occluded regions and remove them from the energy computation. As shown in Fig. 5.5 (c), a region is occluded whenever the profile $x + z(x)$ is discontinuous and the jump is negative (a positive jump represents a desocclusion).

The occluded regions can be deduced from the 3D model. To see how they are computed, let's consider a given line $y = y_0$ and two planes π_1 and π_2 modelling what happens on the left and right of a point x_0. If $z_{\pi_1}(x_0) > z_{\pi_2}(x_0)$ then the points before x_0 are possibly occluded. The occluded points are computed iteratively: starting from point x_0, a point x is occluded

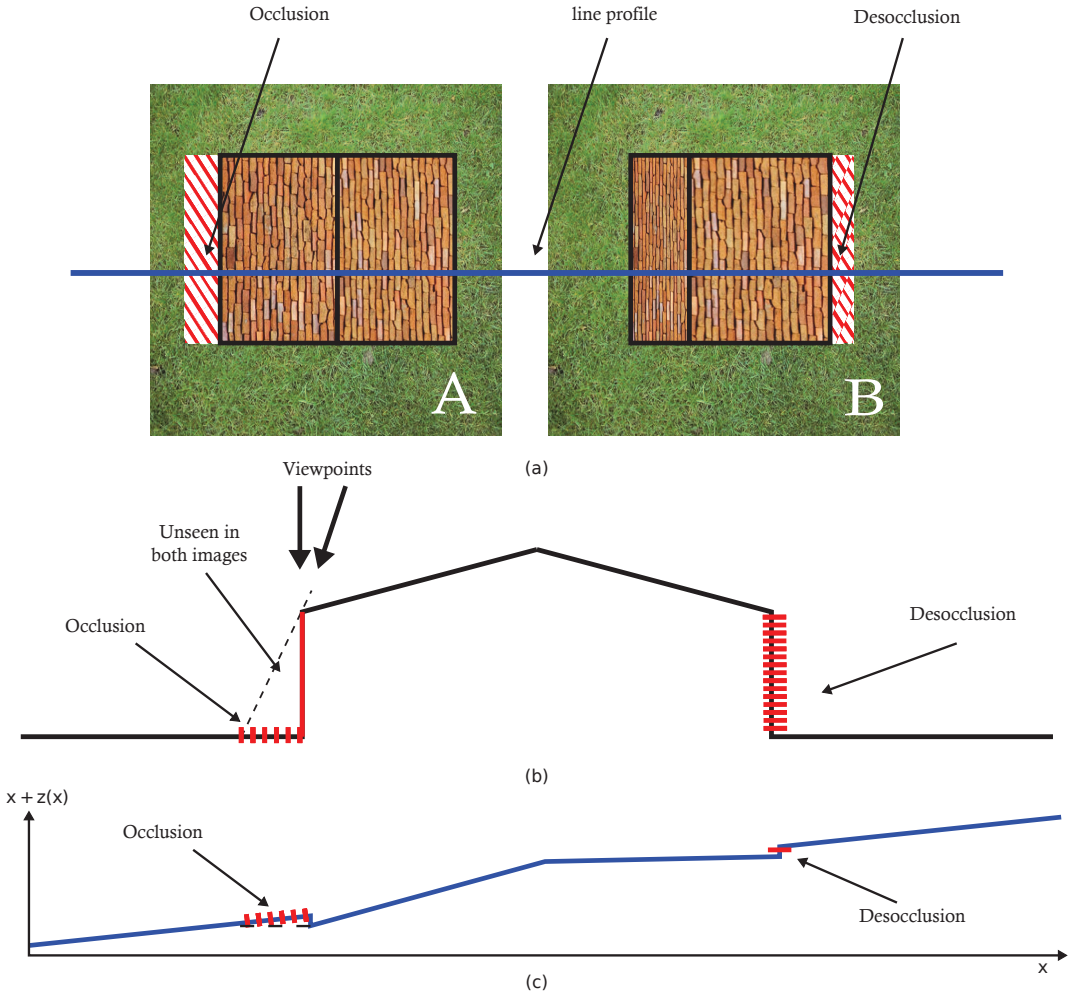


Figure 5.5: Occlusion in stereo-vision. (a) Two views of a same scene. (b) Disparity values along the blue line. (c) $x + z(x)$.

if $x + z_{\pi_1}(x) > x_0 + z_{\pi_2}(x_0)$.

Let's now integrate this in the re-projection error energy given by Eq. 5.12. Considering a situation with two planes π_1 and π_2, each possible separation $\gamma \in \Gamma_s$ induces a different 3D model and therefore a possibly different occlusion region. Let us call $\mathcal{R}_{1,2,\gamma,occ}$ the occluded region caused by the closest of objects 1 or 2, when the separation between $\mathcal{R}_1$ and $\mathcal{R}_2$ occurs along curve γ, and the disparity in regions $\mathcal{R}_1$ and $\mathcal{R}_2$ follow respectively the affine models π_1 and π_2. We can then define a more accurate re-projection energy considering occlusion as follows:

$$E_{\varepsilon,occ}(\mathcal{R}_{1,2}, \gamma) = E_{\varepsilon}(\mathcal{R}_{1,2} \setminus \mathcal{R}_{1,2,\gamma,occ}, \gamma) \tag{5.16}$$

The problem that occurs with Equation (5.16) is that since the energy is computed only on non-occluded points, the number of points considered Equation (5.16) may vary depending on the curve γ that is considered. We propose to overcome this situation by adding to Equation (5.16) an expectation of the error that should occur in the occluded region considering what is observed in the visible region. If the error at each point is seen as a random variable,

the expectation of the error of the occluded points can then be estimated from the observed points. Equation (5.16) then becomes:

$$\begin{aligned} E'_{\varepsilon,occ}(\mathcal{R}_{1,2},\gamma) &= E_\varepsilon(\mathcal{R}_{1,2}\setminus\mathcal{R}_{1,2,\gamma,occ},\gamma)+\mathbb{E}\left(E_\varepsilon(\mathcal{R}_{1,2,\gamma,occ},\gamma)\right) \\ &= \left(1+\frac{\#\mathcal{R}_{1,2,\gamma,occ}}{\#(\mathcal{R}_{1,2}\setminus\mathcal{R}_{1,2,\gamma,occ})}\right)\cdot E_\varepsilon(\mathcal{R}_{1,2}\setminus\mathcal{R}_{1,2,\gamma,occ},\gamma) \end{aligned} \tag{5.17}$$

where $\#\cdot$ is the cardinal operator of a finite set.

5.4.3 Contour validation for the L_2 error energy.

Whenever two planes are very close to each other or when the texture in the image is not well defined, the re-projection error may be unreliable. In the first situation, since the planes are close to each other, the disparities extrapolated from their equation have similar values. The computation of the re-projection is then almost the same for both planes which makes it hard to separate them. In the second situation, when the texture is not well contrasted, any disparity gives about the same error value after re-projection. We therefore introduce an *a contrario* to decide when it is preferable to reject the result obtained by the re-projection error.

We first propose to simplify the re-projection error energy by binarizing it. This way, it becomes a lot easier to define a sensible background model to describe the data for the *a contrario* criterion. Though this new energy is simpler than before, the results obtained are comparable to the one obtained with the classical re-projection error energy $E_{\varepsilon,occ}$. Moreover, the failure cases, are the same as before: similar planes or no texture. The last interest is that the energy gets contrast invariant and is no longer sensitive to big contrast changes.

The binarization is done by first assigning a binary label to each point. For any point that we consider, this label will be 1 if plane π_1 causes the lower re-projection error than plane π_2 and 2 otherwise. The binarized error is then measured by counting the number misclassified points for a given contour:

$$E_{\#\varepsilon}(\mathcal{R}_{1,2},\gamma)=\sum_{\mathbf{x}\in\mathcal{R}_{1,2}}\mathbb{1}_{\{|u\circ\mathcal{T}_\gamma(\mathbf{x})-\tilde{u}(\mathbf{x})|>|u\circ\bar{\mathcal{T}}_\gamma(\mathbf{x})-\tilde{u}(\mathbf{x})|\}} \tag{5.18}$$

where $\bar{\mathcal{T}}_\gamma$ is the complementary point transformation of $\mathcal{T}_\gamma$:

$$\bar{\mathcal{T}}_\gamma : \begin{array}{l} \mathbb{R}^2\mapsto\mathbb{R}^2 \\ \mathbf{x}\mapsto\left\{\begin{array}{l}\mathcal{T}_{\pi_1},\text{ if }\mathbf{x}\in\mathcal{R}_{\gamma,2}\\ \mathcal{T}_{\pi_2},\text{ if }\mathbf{x}\in\mathcal{R}_{\gamma,1}\end{array}\right.\end{array} \tag{5.19}$$

Including occlusion as before gives:

$$E_{\#\varepsilon,occ}(\mathcal{R}_{1,2},\gamma)=\left(1+\frac{\#\mathcal{R}_{1,2,\gamma,occ}}{\#\mathcal{R}_{1,2}\setminus\mathcal{R}_{1,2,\gamma,occ}}\right)\cdot\sum_{\substack{\mathbf{x}\in\\ \mathcal{R}_{1,2}\setminus\mathcal{R}_{1,2,\gamma,occ}}}\mathbb{1}_{\{|u\circ\mathcal{T}_\gamma(\mathbf{x})-\tilde{u}(\mathbf{x})|>|u\circ\bar{\mathcal{T}}_\gamma(\mathbf{x})-\tilde{u}(\mathbf{x})|\}} \tag{5.20}$$

This energy measures if the two classifications given in one hand by the separation γ and in another hand by the binary plane association are in accordance. When the edge between the two planes is well defined, the plane labels are clearly separated into two distinct groups and the curve minimizing the re-projection error fits to this separation (see Figure 5.6 (d) and

(e)). As opposed to that, when the two planes are very close to each other or when the texture is not well contrasted, the plane labelling is arbitrary and seems to be random (see Figure 5.6 (b) and (c)). The curve γ minimizing the energy is not necessarily the one separating best the two planes. The randomness of the plane label distribution therefore tells how reliable the contour detection will be. The more random the less reliable. This is a perfect environment to define an *a contrario* criterion and define when the curve obtained by minimizing Equation (5.20) is valid.

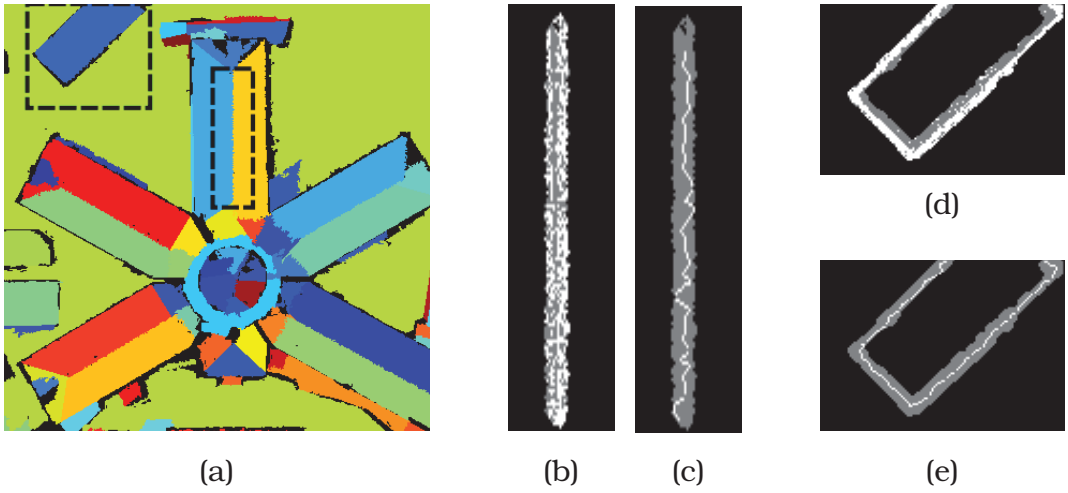


Figure 5.6: Illustration of the *a contrario* validation. (a) shows a piecewise-planar segmentation obtained with our method. We focus on two regions and show which plane gives the lowest re-projection error in $\mathcal{R}_{1,2}$, (b) and (d). We then show the contour obtained by the minimization of Equation (5.20), (c) and (e). In the first region, (b) and (c), the plane labels seem to be distributed randomly and the obtained contour is not validated by the *a contrario* criterion. In the second region, (d) and (e), the plane labels are organized and the contour is validated by the *a contrario* criterion.

A contrario contour validation

We now define an *a contrario* model to quantify for a given contour how likely the energy measure done in (5.20) is to happen randomly. This criterion then says if the contour found by minimizing (5.20) should be used or not.

To do so, we need to define two distribution model: one describing the plane labelling and one describing the error distribution.

Definition 7 (Label distribution H_L) *For any deterministic point $\mathbf{x} \in \mathcal{R}_{1,2}$, let's note $L(\mathbf{x})$ the i.i.e. random variable following the Bernouilli distribution with value 1 if point $\mathbf{x}$ is associated to the plane π_1 and value 2 if $\mathbf{x}$ is associated to π_2:*

$$\mathbb{P}\left(L(\mathbf{x}) = l\right) = \begin{cases} p_l, & \text{if } l = 1 \\ 1 - p_l, & \text{if } l = 2 \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

The distribution parameter p_l can be either supposed to be equi-distributed for each label ($p_l = 0.5$) or estimated over the data.

$$p_l = \frac{\#\{\mathbf{x} \in R_{1,2}, l(\mathbf{x}) = 1\}}{\#\mathcal{R}_{1,2}} \tag{5.22}$$

where $l(\mathbf{x})$ is the observed label value at point $\mathbf{x}$:

$$l(\mathbf{x}) = \begin{cases} 1, \text{ if } |u \circ \mathcal{T}_{\pi_1}(\mathbf{x}) - \tilde{u}(\mathbf{x})| < |u \circ \mathcal{T}_{\pi_2}(\mathbf{x}) - \tilde{u}(\mathbf{x})| \\ 2, \text{ otherwise} \end{cases}$$

Considering that any contour $\gamma \in \Gamma_s$ splits $\mathcal{R}_{1,2}$ into two parts, the validity of labelling of a point $\mathbf{x} \in R_{1,2}$, with label $L(\mathbf{x})$ following H_L, is then a Bernouilli random variable $V(\mathbf{x})$ of parameter $p(\gamma)$:

$$p(\gamma) = \frac{\#\mathcal{R}_{\gamma,1}}{\#\mathcal{R}_{1,2}} \cdot p_l + \frac{\#\mathcal{R}_{\gamma,2}}{\#\mathcal{R}_{1,2}} \cdot (1 - p_l) \tag{5.23}$$

This parameter varies from $\min(p_l, 1 - p_l)$ to $\max(p_l, 1 - p_l)$. We then define the following distribution model for a point to be well classified:

Definition 8 (Valid Point distribution(background process) H_0) *We call background process the set of i.i.e. Bernouilli random variable $V_\gamma(\mathbf{x})$, $\mathbf{x} \in \mathcal{R}_{1,2}$, where $V_\gamma(\mathbf{x})$ equals 1 if the label of $\mathbf{x}$ induced by γ is valid, and 0 otherwise. V_γ is then defined as:*

$$\mathbb{P}\left(V_\gamma(\mathbf{x}) = v_\gamma\right) = \begin{cases} p_0 = \max(p_l, 1 - p_l), & \text{if } v_\gamma = 1 \\ 1 - p_0 = \min(p_l, 1 - p_l), & \text{if } v_\gamma = 0 \end{cases} \tag{5.24}$$

From that, we now define the Number of False Alarms:

Definition 9 (Number of False Alarms (NFA)) *Given a parametric curve $\gamma \in \Gamma_s$ that separates $\mathcal{R}_{1,2}$ into two disjoint subsets $\mathcal{R}_{\gamma,1}$ and $\mathcal{R}_{\gamma,2}$, we define the number of false alarms of γ and $\mathcal{R}_{1,2}$ as:*

$$NFA(\mathcal{R}_{1,2}, \gamma) = N_{tests} \cdot \mathbb{P}\left[\mathcal{K}(\mathcal{R}_{1,2}) \geq k(\mathcal{R}_{1,2}, \gamma)\right] \tag{5.25}$$

where,

- *k is the number of points properly classified by γ;*

$$k(\mathcal{R}_{1,2}, \gamma) = \sum_{\mathbf{x} \in \mathcal{R}_{\gamma,1}} \mathbb{1}_{\{l(\mathbf{x})=1\}} + \sum_{\mathbf{x} \in \mathcal{R}_{\gamma,2}} \mathbb{1}_{\{l(\mathbf{x})=2\}}$$

- *$\mathcal{K}$ is a random variable that counts the number of well classified random points that would be obtained if the data were distributed according to model H_0:*

$$\mathcal{K}(\mathcal{R}_{1,2}) = \sum_{\mathbf{x} \in \mathcal{R}_{1,2}} V_\gamma(\mathbf{x})$$

- *N_{tests} is the number of possible separations that can be tested:*

$$N_{tests} = \#\Gamma_s$$

- $\mathbb{P}\left[\mathcal{K}(\mathcal{R}_{1,2}) \geq k(\mathcal{R}_{1,2}, \gamma)\right]$, *is the probability to obtain as many valid points with the background model* H_0 *as what we observe with the data. It can be computed using the tail of the binomial law:*

$$\begin{aligned}\mathbb{P}\left[\mathcal{K}(\mathcal{R}_{1,2}) \geq k(\mathcal{R}_{1,2}, \gamma)\right] &= \mathcal{B}\big(\#\mathcal{R}_{1,2}, k, p_0\big) \\ &= \sum_{j \geq k}^{\#\mathcal{R}_{1,2}} \left(\begin{array}{c} \#\mathcal{R}_{1,2} \\ j \end{array} \right) p_0^j \cdot (1-p_0)^{\#\mathcal{R}_{1,2}-j}\end{aligned}$$

The NFA is the expected number of times to obtain a random configuration as good as what observed. If the NFA is low, the observed configuration is not likely to happen randomly.

Definition 10 (ε-meaningful contour) *Given a region* $\mathcal{R}_{1,2} \in \mathcal{D}$*, a contour* $\gamma \in \Gamma_s$ *is said to be* ε*-meaningful for* $\mathcal{R}_{1,2}$ *whenever* $NFA(\mathcal{R}_{1,2}, \gamma) < \varepsilon$*.*

The following proposition provides a sanity check, ensuring that the definition of the NFA (and the detection rule based on it) have actually the intended meaning, namely limiting the expected number of false detections due to noise below ε:

Proposition 2 *Given a region* $\mathcal{R}_{1,2}$ *and supposing that the valid points are distributed according to the background model, the expected number of* ε*-meaningful separation* γ *that is obtained by testing all the possible separations in* Γ_s *is less than* ε*.*

Proof Let's note S the random variable defined as:

$$S = \sum_{\gamma \in \Gamma_{1,2}} \chi_{(\mathcal{R}_{1,2}, \gamma)}$$

where, $\chi_{(\mathcal{R}_{1,2},\gamma)} = \mathbb{1}_{\gamma \text{ is } \varepsilon\text{-meaningful}}$. Using the linearity of the expectation operator we have:

$$\mathbb{E}[S] = \sum_{\gamma \in \Gamma_s} \mathbb{E}[\chi_{(\mathcal{R}_{1,2},\gamma)}] = \sum_{\gamma \in \Gamma_s} \mathbb{P}\left[B(\#\mathcal{R}_{1,2}, \mathcal{K}(\mathcal{R}_{1,2}), p_0) < \frac{\varepsilon}{N_{tests}}\right]$$

Since the survival function of $\mathcal{K}$ is $k \to B(\#\mathcal{R}_{1,2}, k, p_0)$, we have

$$\mathbb{P}\left[B(\#\mathcal{R}_{1,2}, \mathcal{K}(\mathcal{R}_{1,2}), p_0) < \frac{\varepsilon}{N_{tests}}\right] < \frac{\varepsilon}{N_{tests}}$$

then

$$\mathbb{E}[S] < \sum_{\gamma \in \Gamma_s} \frac{\varepsilon}{N_{tests}} = \varepsilon$$

□

Proposition 2 used with $\varepsilon = 1$ implies that less than one false alarm is expected using the background model as data and testing all the possibilities. This threshold is therefore the one usually used in *a contrario* methods.

Note that for simplicity reasons, we supposed that the background model was the same for all possible contour $\gamma \in \Gamma_s$ instead of using equation (5.23). This supposition is a little optimistic for the background model which makes it harder for a contour to be meaningful.

The experiments of section 5.6 were all done under this assumption. Figure 5.7 shows the result of the *a contrario* model on the Toulouse dataset.

One may see that almost no contour was NFA validated at a plane intersection. This can be explained by the fact that when two planes intersect each other with a large angle, since the disparity is very close using one plane or the other in the contour search region, it is hard to say which plane fits best the data.

In another hand, almost all the contours were validated when the two planes were very different (large discontinuity, intersection with a sharp angle). In the Toulouse example, an exception is noticeable at the top branch of the star-shaped building. In that case, the contour was rejected because of the low texture in the shadow region.

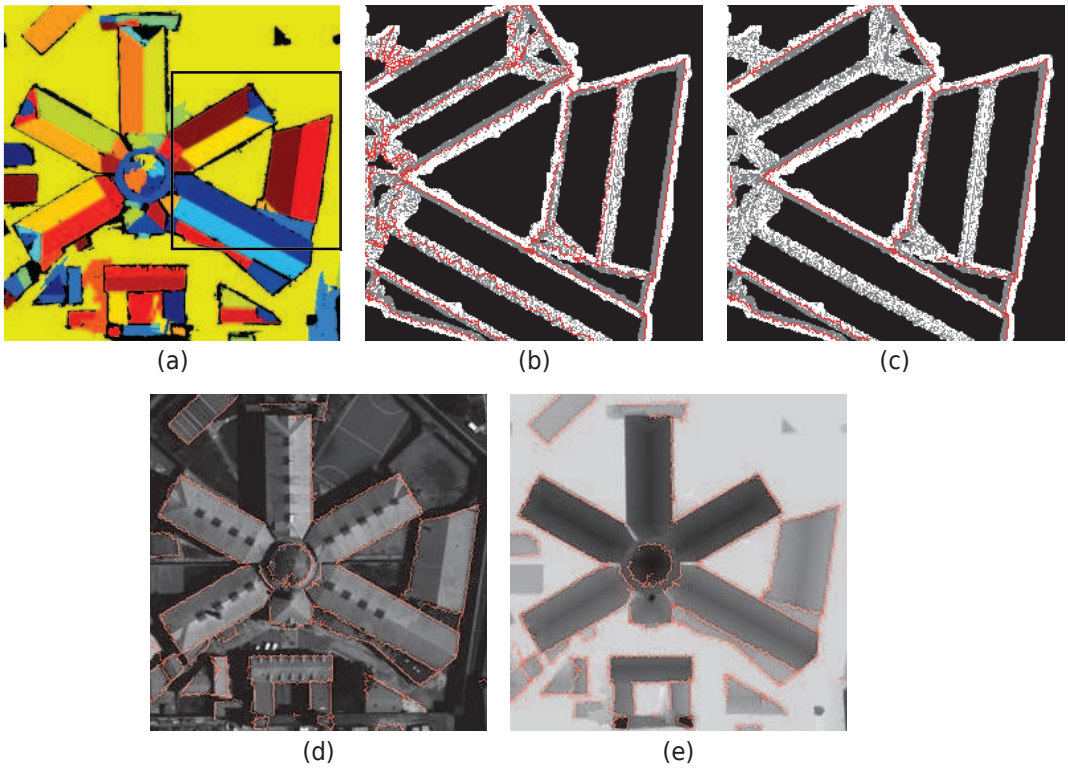


Figure 5.7: Results of the *a contrario* criterion on Toulouse dataset. (a) Input planar classification. (b) and (c) Binary re-projection error and found contours before (b) and after (c) NFA filtering. (d) Reference image and NFA validated contours. (e) Ground truth disparity map and validated contours. One can see that the validated contours tend to fit the data.

5.5 Resolution schemes

As discussed before in the general overview of the algorithm, we propose to solve all the energy minimization problems using dynamic programming. To do so, some adaptation must be made. We propose to divide the search region into subregions where the reduction of the final contour can be supposed linear. From this, a graph where each possible segment of a

subregion is an edge can be build. Dynamic programming then allows to find the sequence of connected segments (one segment per subregion) minimizing each proposed energy.

The reasons of doing so are the following:

- Minimizing the proposed energies is a non convex problem. To understand that, one can think of two highly contrasted curve in the search region which can be represents two local minima of the gradient energy. A similar situation can occur with the L_2 re-projection error energy.
- The result obtained from dynamic programming is a global minimum (since all the possibilities are tried).
- The division into subregions forbids the final curve to go backward or make loops.
- The regularization of the curve is directly induced by the division into subregions which allows simple ways to set this parameter.

This section is organized as follow. We start with a short review of dynamic programming (see Section 5.5.1). Then we describe how the search region $\mathcal{R}_{1,2}$ can be divided into a finite set of subregions $(\mathcal{R}_{1,2,i})_{i=1..N}$ adapted to the contour we are looking for.

5.5.1 Dynamic programming

Dynamic programming is a powerful algorithmic paradigm in which a complex problem is solved by breaking it into a collection of subproblems and by using the answer to the subproblems to compute the complex problem solution. When applicable, dynamic programming considerably reduces computations compared to naive resolution schemes.

Dynamic programming implicitly requires the construction of a Direct Acyclic Graph (DAG). The nodes of the DAG can be considered as the subproblems and the edges are the relations between the subproblems. For a better comprehension, let's now focus on the special case illustrated in Figure 5.8. Now let's illustrate with this example how a dynamic

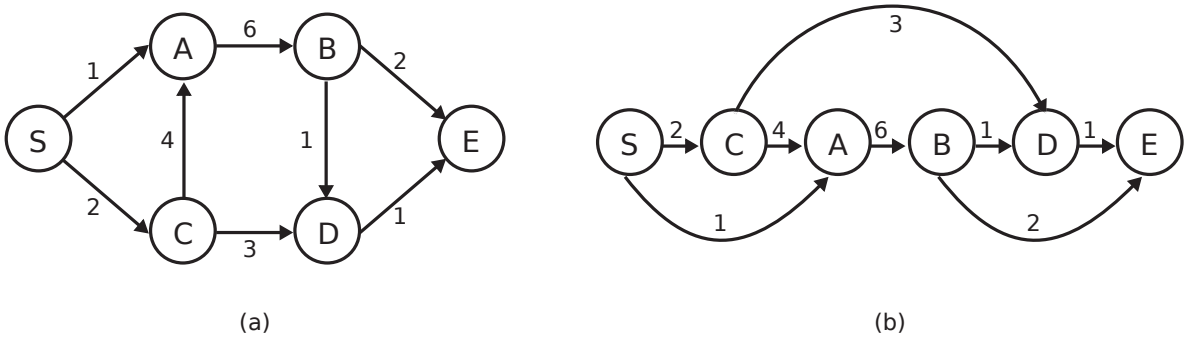


Figure 5.8: (a) A DAG $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and (b) its linearization.

programming algorithm proceeds to find the shortest path from S to E. We will denote by $dist(\mathbf{x})$ the minimal distance from S to any node $\mathbf{x}$ of the graph.

A dynamic programming algorithm starts from the end E and applies a Divide & Conquer strategy to recursively find $dist(E)$ in terms of $dist(\mathbf{x})$ for all nodes $\mathbf{x}$ directly connected to E by a single edge. In our case, since E can only be reached from nodes $\mathbf{x} = B$ (with cost 2)

and $\mathbf{x} = D$ (with cost 1), it is obvious that $dist(E) = \min(dist(B)+2, dist(D)+1)$. Then the computation of $dist(E)$ can be simplified by solving the two (smaller) subproblems $dist(B)$ and $dist(D)$. The same reasoning can then be done recursively until we reach the subproblem of computing $dist(S) = 0$.

Figure 5.8.b shows a linearized version of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 5.8.a. In this linearized version S is always the first node, E is always the last node, and the remaining nodes $\mathbf{x}$ are sorted in one of the linear orders that are compatible with the pre-order induced by the DAG. By this we mean that if $\mathbf{x} < Y$ in the linearized graph, then no directed edge $(Y, \mathbf{x})$ goes from Y to $\mathbf{x}$ in G. There may be several such linear orders, but we only care here about the fact that at least one exists, because G is a DAG.

Using the linearized version of the graph, this recursive reasoning can be implemented by the more efficient iterative Algorithm 8, which follows the linear order from S to E. In fact, observe that when the loop reaches node v, then $dist(u)$ already contains the right (finite) value for any $u < v$. Therefore $dist(u)$ is well defined for any $(u, v) \in \mathcal{E}$. The shortest path from S to E can then be computed in a single path using Algorithm 8.

Algorithm 8: Shortest path in a DAG with dynamic programming.

Data:
A DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with S the start node and E the end node
A weight function of each edge $w : \mathcal{E} \mapsto \mathbb{R}$
Result:
$dist(E)$

```
1 begin
2 |  ∀v ∈ V, dist(v) = ∞
3 |  dist(s) = 0
4 |  foreach v ∈ V,in linearized order do
5 |  |  dist(v) = min_{(u,v)∈E)} dist(u) + w(u, v)
6 |  end
7 end
```

5.5.2 Division into subregions

To be able to use dynamic programming to find the contour minimizing one of the energies presented before(see Sections 5.4.1 and 5.4.2), we slice the search region into smaller subregions. The division into subregions should be such that the restriction of the final separation contour $\gamma_{1,2}$ to any subregion is a segment.

Let's consider as before the case of two regions $\mathcal{R}_1$ and $\mathcal{R}_2$ for which we want to find the separation $\gamma_{1,2}$ and let's note $\partial\mathcal{R}_1$ (resp. $\partial\mathcal{R}_2$) the common border of $\mathcal{R}_1$ (resp. $\mathcal{R}_2$) with the search region $\mathcal{R}_{1,2}$ (see previous subsection). We are looking for a set of pairs of points $(P_i, Q_i) \in \partial\mathcal{R}_1 \times \partial\mathcal{R}_2$ such that each segment $[P_i, Q_i]$ is entirely contained in $\mathcal{R}_{1,2}$ and do not intersect with any other segment $[P_j, Q_j]$. According to this definition, each segment splits region $\mathcal{R}_{1,2}$ into two parts. The subregions are then defined by two consecutive segments. We note $\mathcal{R}_{1,2,i}$ the subregion defined by the two consecutive segments $[P_{i-1}, Q_{i-1}]$ and $[P_i, Q_i]$. Figure 5.9 shows an example of division into subregions and recall some of the notations.

The first interest of doing this division is that the number of vertices of the polygonal curve is set. Since the separation $\gamma_{1,2}$ is supposed to be linear in each subregions, the vertices

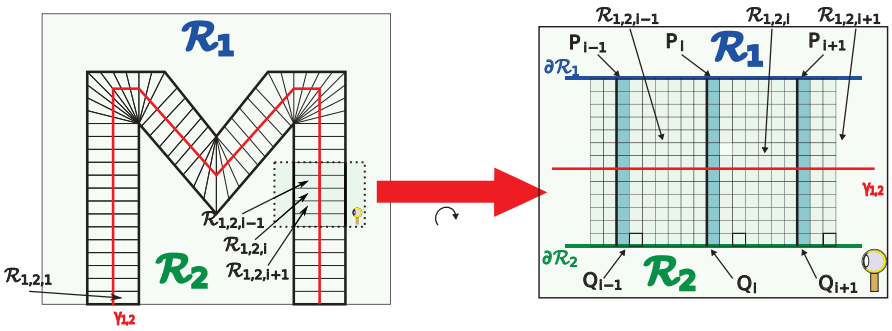


Figure 5.9: Notations used to define the curve search problem.

is set by the division into subregions. The problem is now reduced to finding a sequence of points G_i, $G_i \in [P_i, Q_i]$, defining the polygonal curve $\gamma_{1,2}$. We will note Γ_s the set of all possible polygonal curves defined by the previous sequence. Another interest of the subregion division is that it ensures that the polygonal curve found do not make any loops (which is something good since this curve is supposed to separate two regions). At last, this division allows a resolution of Equation 5.1 using dynamic programming (this sometimes requires an adjustment depending on the energy that is used).

The division of $R_{1,2}$ is achieved with the following steps (see Figure 5.10):

- First, we define the two border point sets $\partial\mathcal{R}_1$ and $\partial\mathcal{R}_2$. In the discrete case, this can be done using the 4-connectivity for instance.
- Then for each point $P \in \partial\mathcal{R}_1$ we find its closest point $Q \in \partial\mathcal{R}_2$. ($\forall Q' \in \partial\mathcal{R}_2, ||P-Q||_2 \leq ||P-Q'||_2$) such that the segment is entirely contained in $\mathcal{R}_{1,2}$ ($[P,Q] \subset \mathcal{R}_{1,2}$). The same is done for all the points of $\partial\mathcal{R}_2$.
- Since we want to achieve a partition of $\mathcal{R}_{1,2}$, we remove the identical segments and intersecting segments. This is likely to happen if the distance between $\partial\mathcal{R}_1$ and $\partial\mathcal{R}_2$ is not constant (as shown in Figure 5.10.
- At last, we interpolate the missing segments so that each point of $\partial\mathcal{R}_1$ and $\partial\mathcal{R}_2$ has at least one correspondence. This can be done by interpolating the correspondence linearly.

The results of this approach on data (Toulouse St-Michel and Village disparity maps) is shown on Figure 5.13.

5.5.3 Optimal contour search

We now explain how to combine the division into subregions and dynamic programming to find an optimal contour for each possible energy case.

Minimisation of Eq. 5.10 and Eq. 5.12

If consider the reduction of the searched contour γ to each subregion $\mathcal{R}_{1,2,i}$ of $\mathcal{R}_{1,2}$ as a segment γ_i, then one can easily see that the two energies from Eq. 5.10 and Eq. 5.12 can

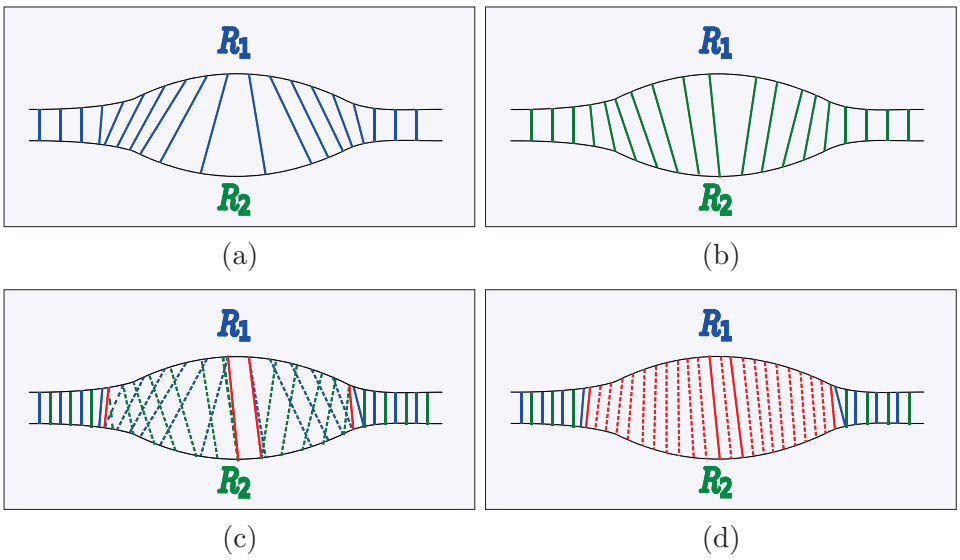


Figure 5.10: Division into subregions. (a) Find for each point in $\partial\mathcal{R}_1$ its closest point in $\partial\mathcal{R}_2$. (b) Find for each point in $\partial\mathcal{R}_2$ its closest point in $\partial\mathcal{R}_1$.(c) Remove the intersecting segments. In this case this happened because the search region was enlarged. (d) Interpolate the missing segments.

be computed independently within each subregion. The integrals can be decomposed into a finite sum of simpler integrals. The gradient energy case is pretty straightforward since the energy is computed along the contour:

$$E_\nabla(\mathcal{R}_{1,2},\gamma) = -\sum_{i=1}^{N_s}\int_0^1 \left|\vec{\nabla}u(\gamma_i(s))\cdot\vec{\dot{\gamma}_i}(s)\right|ds = -\sum_{i=1}^{N_s} E_\nabla(\mathcal{R}_{1,2,i},\gamma_i) \tag{5.26}$$

The same can be done with the L_2 re-projection error energy since it is a sum of square error at each point and that: $\cup_i\mathcal{R}_{1,2,i} = \mathcal{R}_{1,2}$ and $\mathcal{R}_{1,2,i}\cap\mathcal{R}_{1,2,j} = \emptyset$ if $i \neq j$. We then have:

$$E_\varepsilon(\mathcal{R}_{1,2},\gamma) = \sum_{i=1}^{N_s} E_\varepsilon(\mathcal{R}_{1,2,i},\gamma_i) \tag{5.27}$$

Using the same notation as in section 5.5.2, we note Γ_s the set of all the possible polygonal curves define by the division into subregions. Let's now build the following graph $\mathcal{G} = (\mathcal{V}_\nabla, \mathcal{E}_\nabla)$ where each node is the extremity of a possible segment γ_i and each edge is the segment γ_i. We at last consider the following weight function:

$$w_\nabla : \begin{array}{l}\mathcal{E}_\nabla \mapsto \mathbb{R}\\ \gamma_i \mapsto E_\nabla(\mathcal{R}_{1,2,i},\gamma_i)\end{array}$$

for the gradient energy case and

$$w_\varepsilon : \begin{array}{l}\mathcal{E}_\varepsilon \mapsto \mathbb{R}\\ \gamma_i \mapsto E_\varepsilon(\mathcal{R}_{1,2,i},\gamma_i)\end{array}$$

for the L_2 re-projection error energy case.

Each possible continuous polygonal contour is composed of a unique set of segments $(\gamma_i)_{i=1..N_s}$. By construction, each of these sets of segments is represented by a unique path in the graph $\mathcal{G}$. Conversely, each path through all the subregions, is equivalent to a unique set of segments $(\gamma_i)_{i=1..N_s}$ representing a continuous polygonal curve. Moreover, the weight w_∇ associated to each path in the graph, is exactly the Gradient energy of the associated contour as defined in Eq. (5.26). The same can be stated for the weight w_ε and Eq. 5.27.

Then finding the shortest path in graph $\mathcal{G}$ associated to the weight function w_∇ (resp. w_ε) is equivalent to finding the solution to Eq. 5.1 (resp. Eq. 5.27) using the gradient energy (resp. the L_2 re-projection error energy).

Since each path in the graph goes only once through each subregions, $\mathcal{G}$ is already a DAG. The shortest path can then be found by dynamic programming using Algorithm 8.

Occlusion

We saw in Section 5.4.2 that adding the occlusion information changed the re-projection error energy (see Eq. 5.17). Let's now consider the division into subregions in Equation(5.17):

$$E'_{\varepsilon,occ}(\mathcal{R}_{1,2},\gamma) = \sum_{i=i}^{N_s} \left(1 + \frac{\#(\mathcal{R}_{1,2,i} \bigcap \mathcal{R}_{1,2,\gamma,occ})}{\#(\mathcal{R}_{1,2,i} \setminus \mathcal{R}_{1,2,\gamma,occ})}\right) \cdot E_\varepsilon(\mathcal{R}_{1,2,i} \setminus \mathcal{R}_{1,2,\gamma,occ}, \gamma_i) \tag{5.28}$$

The reduction γ_i of $\gamma \in \Gamma_s$ to the subregion $\mathcal{R}_{1,2,i}$ may occlude points from other subregions $\mathcal{R}_{1,2,k}$. The energies in each subregion are then no longer independent. As a direct consequence, the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ described before is no longer suitable to describe the data since the weight associated to an edge $e \in \mathcal{E}$ may depend on the edges it is connected to. An example of this situation is shown in Figure 5.12. The occlusion of region $\mathcal{R}_0$ is not the same when segment e_0 is connected to $e_{1,1}$ and $e_{1,2}$. The correlation energy changes depending on what segment e_0 is connected to, which means that dynamic programming can no longer be used to solve Equation (5.1).

To avoid this, we propose to control the occlusion that each subregion causes on the others. To do so, we redefine the subregion partitioning so that the occlusion induced by the segments in each subregion affects at most its two neighboring subregions. With this new partition, the occlusion induced by a contour γ onto a subregion $\mathcal{R}_{1,2,i}$ is completely defined by the reduction of γ to 3 subregions: $\mathcal{R}_{1,2,i-1}$, $\mathcal{R}_{1,2,i}$ and $\mathcal{R}_{1,2,i+1}$. Then the occlusion energy of any segment γ_i from a subregion $\mathcal{R}_{1,2,i}$ can be computed knowing γ_{i-1} and γ_{i+1}. This filtering is both explained and illustrated in Figure 5.11.

We then construct a new graph $\mathcal{G}_{occ} = (\mathcal{V}_{occ}, \mathcal{E}_{occ})$ in a similar way as for graph $\mathcal{G}$. Each vertex is one of the possible segment extremity and each edge is one of the possible segments. However, this time a segment appears as many times in the graph as it has different occlusion energies. This depends on the segments it is connected to. In Figure 5.12, since e_0 has a different energy if it is connected to $e_{1,1}$ and $e_{1,2}$, the vertex B of $\mathcal{G}$ is split into B and B' in $\mathcal{G}_\varepsilon$.

The number of edges of the new graph $\mathcal{G}_{occ}$ is $\mathcal{O}((\#E)^3)$ which is three orders of magnitude as big as before. However, dynamic programming can now be used, and the fact that most configurations do not produce occlusion ensures relatively fast results.

5.5.4 Contour simplification

The result obtained with each energy functional is rather noisy due to the large number of vertices of the polygonal curve. We therefore need a way to simplify the obtained contour.

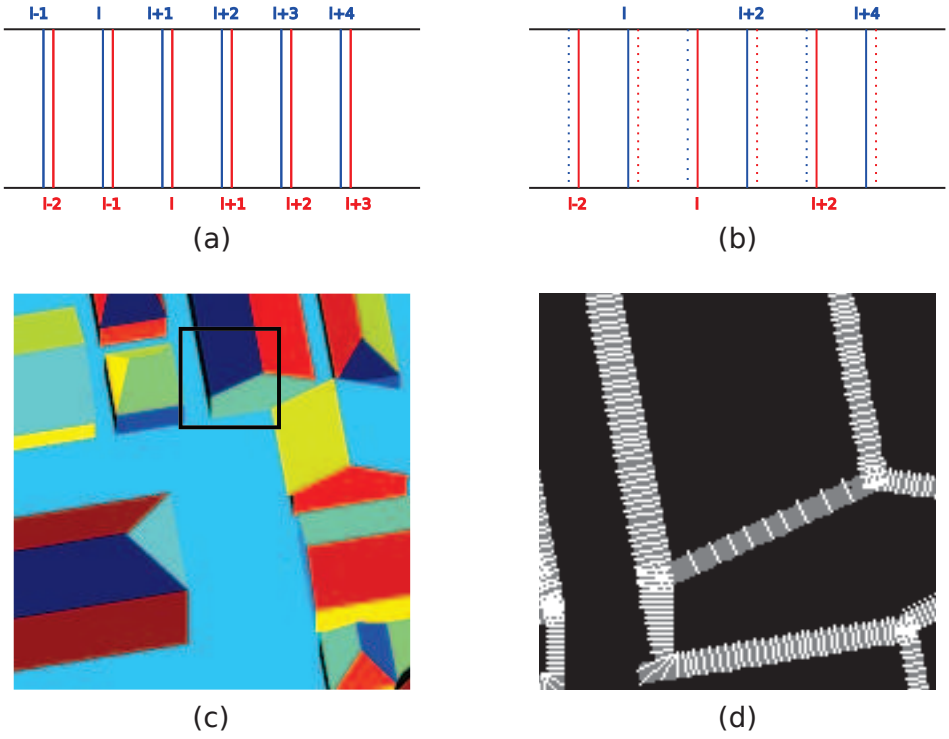


Figure 5.11: Filtering of the subregions of $\mathcal{R}_{1,2}$.(a) and (b) explicative drawing, (c) and (d) example on real data. (a) Initial segmentation of $R_{1,2}$: in red, the segments delimiting the subregions, in blue, the maximal occlusion caused by the delimiting segments. (b) Filtering of the delimiting segments. Starting from segment "i-2", segment "i-1"is filtered as its occlusion is beyond segment "i-2". The next one then becomes segment "i". (c) Input disparity map classification. (d) New partitioning of the search regions after the occlusion filtering. Note that depending on the pair of planes that is considered, the subregions are more or less big depending on the possible occlusion.

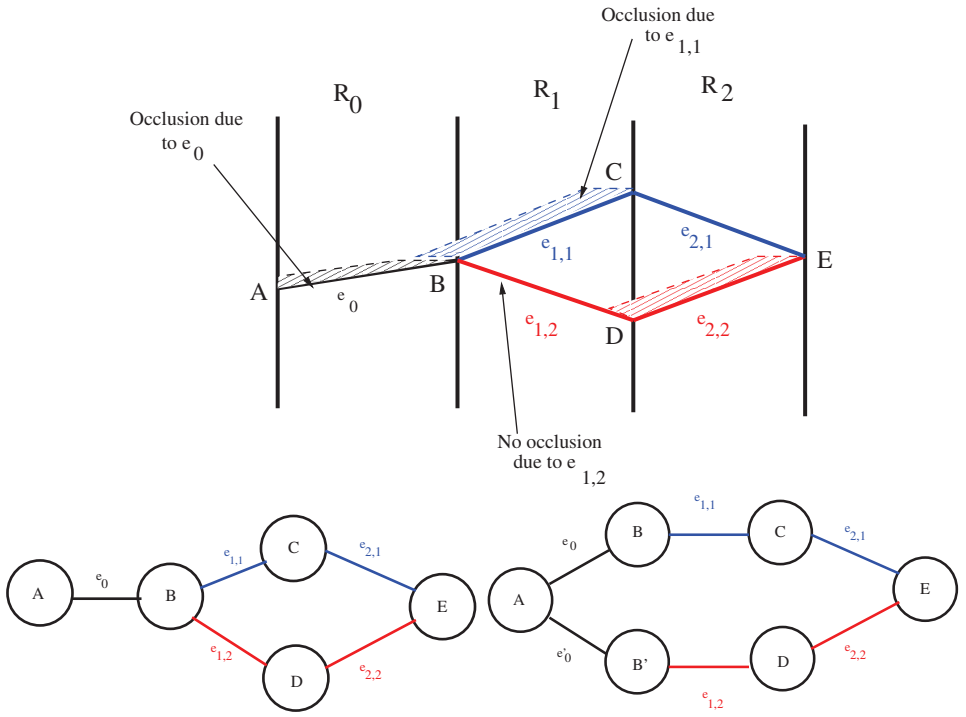


Figure 5.12: Occlusion and graph

Introducing a measure to compare model depending on their complexity is a classical approach in statistical model selection problems. This is usually done using AIC [Akaike, 1974] or BIC [Schwarz, 1978] criteria but can also be done by adapting the *a contrario* criterion from Section 5.4.3 to polygonal curves with less vertices. This way, polygonal curves of different complexity can be compared. Reducing the number of vertices of a curve reduces the number of tests and therefore the NFA in Equation (5.25). However, a simpler model also increases the error. The NFA comparison of two polygonal curves is then a trade-off between precision and model simplicity.

This procedure was used successfully in the case of piecewise planar segmentation as it was shown in Chapter 2. However, in this case, the operation is too difficult to solve. Indeed if one wants to merge two consecutive parts of γ, γ_i and γ_{i+1}, composed of three vertices G_{i-1}, G_i and G_{i+1} there are two possibilities to keep the continuity of γ:

1. The simplest solution is to remove the vertex common to γ_i and γ_{i+1}, G_i, and only consider the merged segment formed by G_{i-1} and G_{i+1}. The problem is that simplifying γ by just removing some of the vertices gives bad results if the vertices are not good from the beginning. A good example would be to consider a sawtooth polygonal curve. If we wish to simplify the curve into a single line, then the result is not the line minimizing the mean square error but the line connecting the two extremities of the polygonal curve.

2. The second solution is to test all the possibilities, which means that merging γ_i and γ_{i+1} requires recomputing the minimal paths for each new possible edge in subregion $\mathcal{R}_{1,2,i} \bigcup \mathcal{R}_{1,2,i+1}$. However, for each new path, one cannot use former results because the computation of the minimal path in a graph is done dynamically. This is because the number of possible paths is too high to store the distance value for each of them.

The solution would then be to compute the minimal path for each possible configuration of subregions of $\mathcal{R}_{1,2}$. If we consider that $\mathcal{R}_{1,2}$ can be divided into at most N_s subregions, this would require running the algorithm $(N_s - 1)$ times which is of course unacceptable.

Instead of running Algorithm 8 for each possible configuration of subregions, one can think of a way to simplify the partition of $\mathcal{R}_{1,2}$ and run it only once. We propose to merge the neighboring subregions whenever the polygonal curve can be assumed to be a single line segment. To do so we compute the principal directions of $\mathcal{R}_{1,2}$, γ_{dir}, and assume that γ behaves similarly as to γ_{dir} (*i.e.* it has as many linear segments as principal directions in $\mathcal{R}_{1,2}$). We therefore divide $\mathcal{R}_{1,2}$ depending on the principal directions that were found. The contour is then found as before using the simplified subregion set to build the solution graph.

Principal directions of $\mathcal{R}_{1,2}$

Finding principal directions of a 2D region is similar in some ways to finding the planar segmentation of a 3D scenes. Instead of fitting planes to 3D data, we want to find line segments describing 2D data. The principal directions are computed from an approximate skeleton of $\mathcal{R}_{1,2}$. The skeleton gives a good description of how $\mathcal{R}_{1,2}$(and therefore the final separation $\gamma_{1,2}$ behaves especially when their main direction changes. Then from this simplified point set, we propose to find the main directions using RANSAC sequentially as proposed in [Vincent and Laganière, 2001] and [Kanazawa and Kawakami, 2004] for plane detection.

To find the skeleton data points, we propose to use the result of the subregion division described in Section 5.5.2 to obtain a sampling of the skeleton of $\mathcal{R}_{1,2}$. We recall that each subregion $\mathcal{R}_{1,2,i}$ is completely defined by the boundary of $\mathcal{R}_{1,2}$ with $\mathcal{R}_1$, $\partial\mathcal{R}_1$, the boundary of $\mathcal{R}_{1,2}$ with $\mathcal{R}_2$, $\partial\mathcal{R}_2$, and two segments $[P_{i-1}, Q_{i-1}]$ and $[P_i, Q_i]$, $(P_{i-1}, P_i) \in \partial\mathcal{R}_1^2$ and $(Q_{i-1}, Q_i) \in \partial\mathcal{R}_2^2$.

At last, as pointed out in [Zuliani et al., 2005], one of the problems of using RANSAC recursively for structured data is the validation of phantom models which can in fact be the mix of several separate models. However, in our situation, the algorithm stays robust because each model found has to be included in $\mathcal{R}_{1,2}$ (which limits possible shortcuts) and the data set is limited to the skeleton (which limits the number of outliers).

An example of the search region partitioning and of its simplification on two dataset is shown in Figure 5.13.

5.6 Experimental results

We tried our algorithm on several dataset. The input classifications were all obtained using the algorithm described in Chapter 3. The disparity maps used here were either ground truth maps with an additive noise or obtained from Neus Sabater's algorithm [Sabater, 2009]. In the latter case, the disparity map is polluted by adhesion.

In all the experiments, the contours were first computed using the binary L_2 re-projection energy (see Section 5.4.3) and validated by the *a contrario* criterion (see Section 5.4.3). In case of rejection, the intersection between the two planes is tested. At last, when none of the first two possibilities were successful, the contours computed using the gradient energy (see section 5.4.1) were used. All the results were then regularized as described in Section 5.5.4.

In Figure 5.14, 5.15 and 5.16 we first show the input classification of the disparity map, and then the reference image and the ground truth with the contour that were found. The red contours are the one that were computed using the binary L_2 re-projection energy and *a*

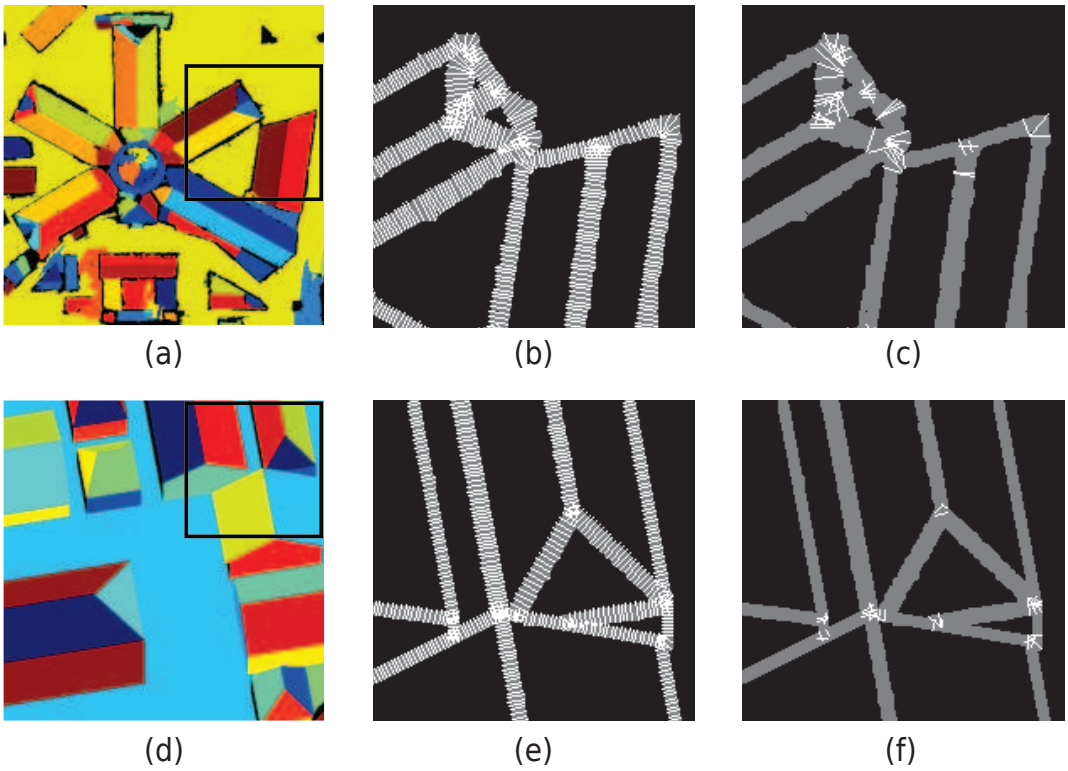


Figure 5.13: Example of search region partitioning. First raw Toulouse dataset. Second raw Village dataset. (a) and (d) input classification. (b) and (e) Result of the search region partitioning as explained in Section 5.5.2. (c) and (f) simplification of the partition.

contrario validated. The green ones are the validated intersection. At last, the blue contours are the ones found using the gradient Energy.

In the Toulouse experiment (See Figure 5.14), we obtained good results both with the ground truth and the disparity map computed with [Sabater, 2009]. As expected, the result with the computed disparity map is a bit noisier than the one of obtained for the ground truth. This is mostly due to the classification that contains a lot less points (See Figure 5.14 (b)) which are moreover a bit less precise. At last, the results need to be observed both with the reference image and the ground truth because they do not exactly correspond each other. The remaining errors of the L_2-re-projection energy are mostly due to the fact that the ground truth shows some adhesion near some of the separations.

The "village" and "campagne" results are rather good. However, some errors still remain (see Figure 5.16: top left zoom region, the red and blue contours; bottom left zoom region). They are mostly due to the fact that the simulated images (reference and secondary) lack of enlightenment conditions. Therefore, the separation is not always clearly delimited especially if the color information of the images is not used. Adding the color information to the computation would certainly enhance the results.

5.7 Conclusion

We presented a new algorithm to refine the separations between planes from a piecewise-planar disparity map segmentation. The refinement may be used as a correction of the adhesion artifact since it is supposed that the data are polluted by it. Another application could be the interpolation of missing points from the detected contour. The contours are computed by taking advantage of both image and 3D information using 3 possible computation criteria.

Though we obtained good results with both noisy ground truth disparity maps as well as computed disparity maps (using [Sabater, 2009]), there is still room for some improvements. For instance, one could think of adding color information to the computation to give more robustness to the result.

So far, the contours are computed only considering the planes two by two. Other improvements could then be to deal with junctions of more than two planes. This amounts to extending curves until they intersect with another curve. A simple way to do so would be the following:

- Construct the adjacency graph $G(\mathcal{R}, \Gamma)$ composed of the regions $R \in \mathcal{R}$ as vertices, and the detected contours $\gamma_{i,j} \in \Gamma$ as edges joining two adjacent regions R_i and R_j. This graph has *planar graph* structure by construction (meaning that no two edges cross in the plane).

- Each minimal loop of length l in the planar graph Grepresents a junction of l planar patches $R_1, \ldots, R_l$ where a decision between $1, \ldots, l-1$ between extended curves γ_{ij} has to be made

At last, an application of this contour refinement could be to recompute the disparity maps in the following way in order to limit adhesion:

- correlation windows restricted to belong to a single class in the piecewise planar segmentation of the image domain,

- affine rectification of each correlation window before computations.

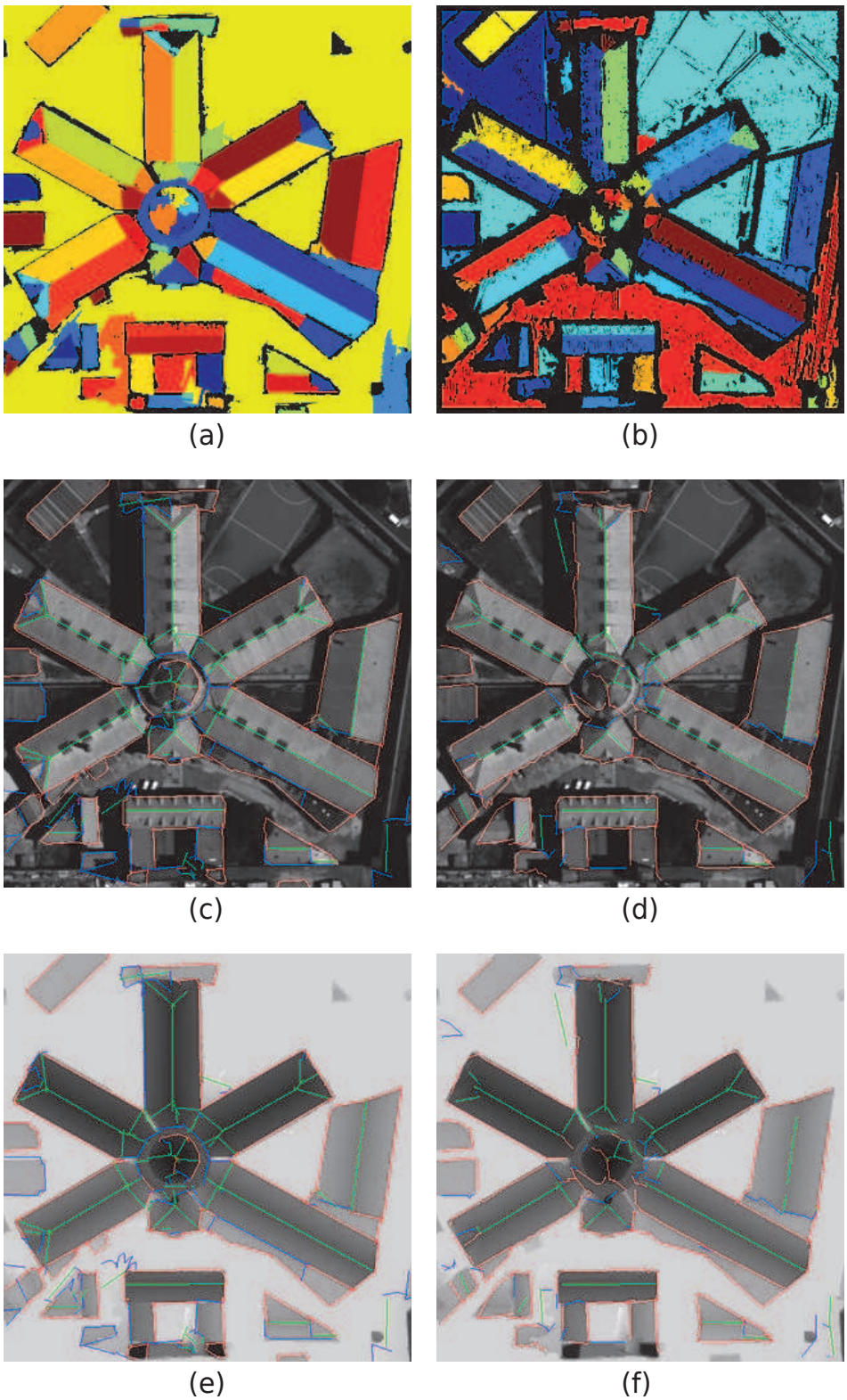
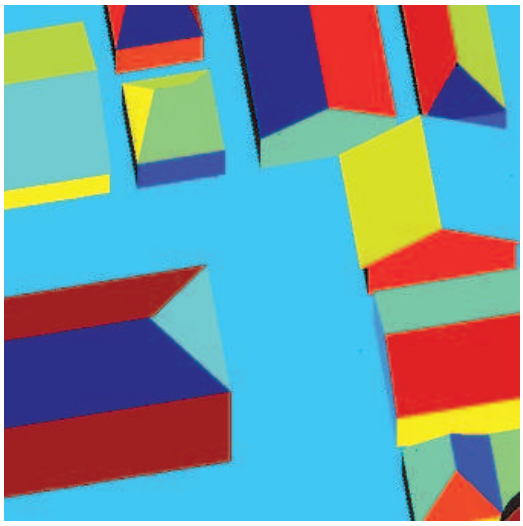
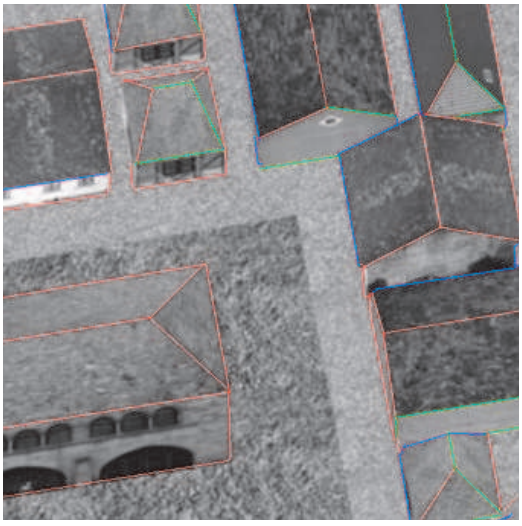


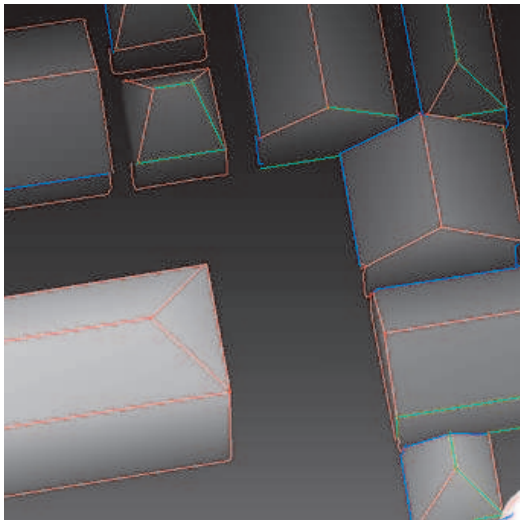
Figure 5.14: Toulouse St-Michel experiment. First column: noisy ground truth as input, Second column: disparity map computed using [Sabater, 2009]. (a) and (b) input classification, (c) and (d) reference image with computed contours, (e) and (f) ground truth disparity map with computed contours.



(a)



(b)



(c)

Figure 5.15: Village experiment. (a) Input classification. (b) Reference image with computed contours. (c) Ground truth disparity map with computed contours.

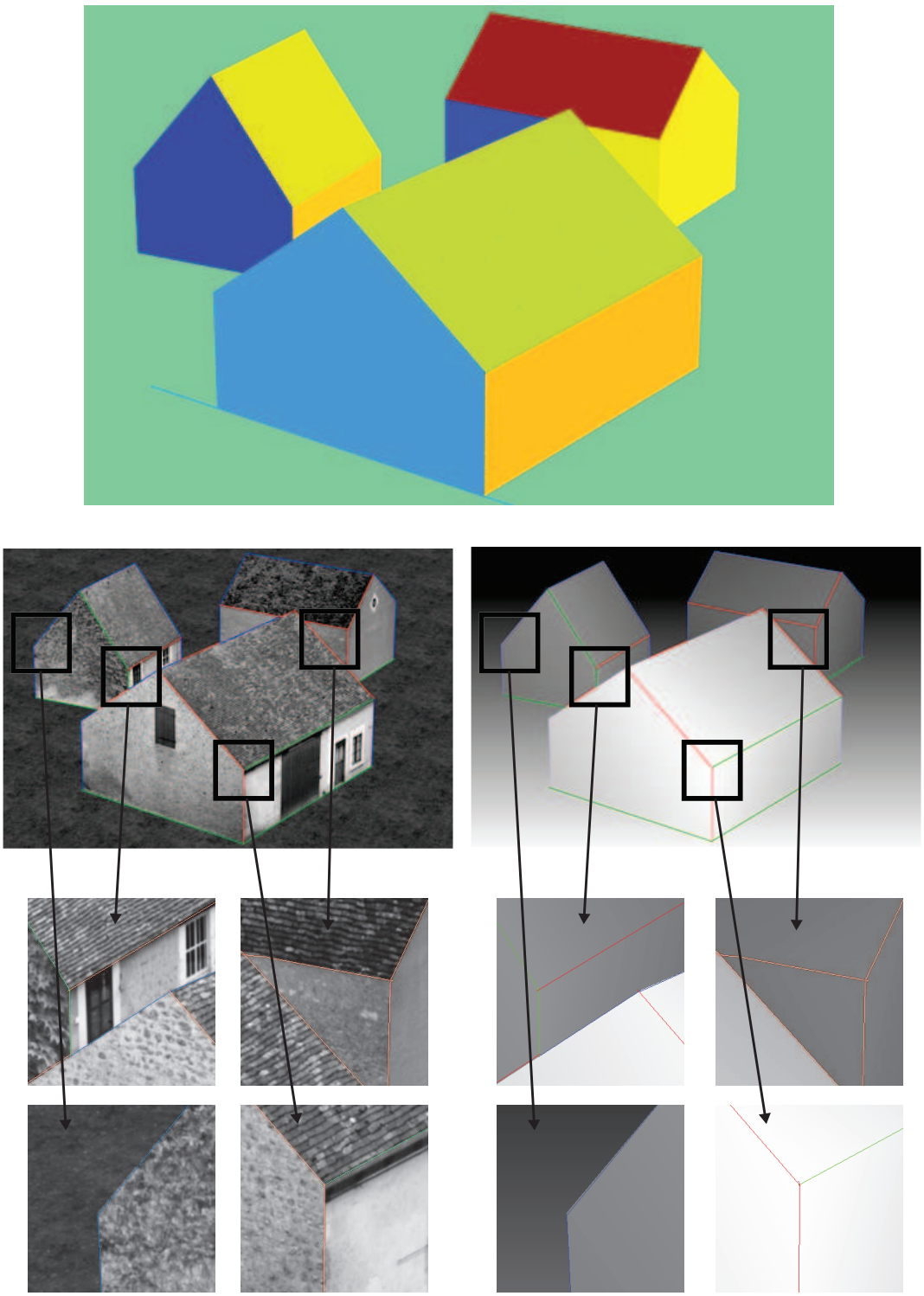


Figure 5.16: Country side experiment. First row: input classification. Second row: Reference image with computed contours (left) and ground truth disparity map with computed contours (right). Last row: zoom on some regions of the reference image and the ground truth.

Chapter 6

Conclusion et perspectives

Dans ce travail de thèse, nous avons étudié la segmentation affine-par-régions des cartes de disparités. Nous nous sommes en particulier intéressés au cas de cartes de disparités issues de stéréoscopie à faible B/H en milieux urbain. Pour toutes les différentes étapes que nous avons étudiées, nous avons mis l'accent sur le caractère automatique et proposé des moyens de sélectionner les valeurs des différents paramètres.

L'approche que nous avons proposée repose sur la définition d'un critère *a contrario* permettant non seulement de définir lorsqu'une configuration de points peut être considérée comme une facette plane mais aussi la comparaison entre différentes configurations. Nous avons ensuite utilisé une approche gloutonne permettant d'obtenir rapidement la segmentation d'une carte en différents groupes plans. Nous avons de plus défini une méthode pour fixer le seuil de rejet des points aberrants. Ce seuil critique aux résultats est commun à la plupart des méthodes robustes de segmentation de données 3D. Toutefois, peu d'auteurs proposent des solutions permettant de le choisir. Enfin, afin d'affiner le résultat de la segmentation et de la définir aux endroits où la disparité n'est pas renseignée, nous avons proposé une méthode de calcul des contours entre deux facettes planes.

Nos expériences ont prouvé que la reprojection sur les plans obtenus par notre segmentation permettait à la fois de débruiter la carte de disparités mais aussi d'interpoler efficacement les données manquantes.

Nous envisageons plusieurs pistes pour continuer et améliorer notre travail.

Gestion des contours et vectorisation

Pour l'instant, le calcul fin des contours a été effectué en ne considérant les plans que deux à deux. Le comportement de cette approche dans les régions où plus de deux plans interagissent n'est donc pas clairement défini. Une première approche pourrait consister à utiliser des distances géodésiques pour assigner les différents points manquant dans ces régions à l'un des plans disponibles.

Une étape finale serait enfin de pouvoir définir un contour polygonal fermé pour chacune des facettes planes. Il suffit pour cela d'appliquer les propositions faites en conclusion du Chapitre 5 c'est-à-dire de considérer les extensions de chaque contour dans les zones où plus de deux régions planes s'intersectent et de choisir la meilleure explication. Le résultat serait alors une carte de disparités vectorielle définie par un ensemble de plans 3D et leur contour polygonal associé. Ceci permettrait enfin d'ajouter toutes les facettes planes verticales manquantes en prenant en compte les plans voisins présentant une discontinuité. Une surface continue serait

alors accessible, ceci permettant une visualisation 3D de la scène.

Recalcul fin des disparités

L'un des défauts du calcul de disparités par mise en correspondance de blocs ("block matching") est de supposer une disparité constante à l'intérieur du bloc considéré. Ceci ne prend pas en compte la géométrie 3D de la scène et est à l'origine d'artefacts tels que l'adhérence. La conséquence a pu être observée dans les expériences du Chapitre 4: les plans 3D non-ortho-frontaux au plan image sont calculés avec un biais et les bords des objets en premiers plans sont dilatés. À partir d'une segmentation affine-par-morceaux telle que celle que nous proposons, deux corrections sont possibles pour affiner la mise en correspondance en évitant l'adhérence:

- Appliquer la transformation affine locale trouvée par la segmentation au moment de la recherche du meilleur bloc et supprimer le biais dans les calculs de disparités dans les plans non-orthofrontaux.
- Adapter la forme et la taille de la fenêtre de recherche à proximité d'un contour de la segmentation de manière à ne jamais prendre en compte plusieurs objets 3D lors du calcul de disparités.

Ces deux corrections couplées à l'algorithme RAFA permettraient alors d'obtenir une carte de disparités avec une adhérence minimale.

Appendix A

Plane parameter estimation

Contents

Résumé: La recherche des meilleurs paramètres pour décrire un plan dans une carte de disparités représente un grand intérêt. Dans cette annexe, nous testons différentes approches pour calculer ces paramètres à partir d'un groupe de points.

Abstract: Finding the best parameters describing a plane in a disparity map is of capital interest. In this appendix, we try different different way of computing the best parameters of a plane given a group of points.

A.1 Introduction

Given a 2-dimensional point set $\mathbf{x_i} = (x_i, y_i)_{i=1..n}$ and their corresponding depth values $(z(\mathbf{x_i}))_{i=1..n}$, we want to find the best planar model that fits these data. In this appendix, we review several methods to estimate the parameters of a plane given a data set. First, we will concentrate on data with no outliers, since it is the case for the region growing algorithm. In that case, the data are supposed to respect the following model:

$$z(\mathbf{x_i}) = z_{\pi_0}(\mathbf{x_i}) + \varepsilon(\mathbf{x_i}), \text{ for } i = 1 \dots n \tag{A.1}$$

with,

$$a_0 \cdot x_i + b_0 \cdot y_i + c_0 \cdot z_{\pi_0}(\mathbf{x_i}) + d_0 = 0, \text{ for } i = 1 \dots n, \tag{A.2}$$

$\mu_0 = (a_0, b_0, c_0, d_0)^T$ is a vector representing the ideal plane parameters and $\varepsilon(\mathbf{x}) \sim \mathbb{N}(0, \sigma)$ are random variables uniformly distributed following a Normal law. For this case, optimal results are obtained using least square approaches. We will therefore concentrate on different ways to compute a least squares estimation and see which approach is the best in our case.

For the case of the split and merge algorithm, another model has to be chosen since a data set may contain outliers. Thus, robust estimators are required in this situation. For a review of the various techniques (robust and least squares) on parameter estimation, see [Zhang, 1997] .

A.2 Least squares

In this section, we are looking for a vector $\mu = [a, b, c, d]^T$ minimizing an energy E:

$$\begin{aligned} \hat{\mu} &= \arg\min_{(a,b,c,d)} \sum_{i=1}^{N} (a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d)^2 \\ &= \arg\min_{(a,b,c,d)} E(\mu) \end{aligned} \tag{A.3}$$

Three remarks come from this equation:

- The parameters for a same plane are defined up to a multiplicative constant. This means that a minimum for E can be defined only under some constraints on μ. Indeed, if μ_1 were the minimum of E and $E(\mu_1) \neq 0$, then $\forall \mu \in \mathbb{R}^4, E(\mu) > E(\mu_1)$ and $\forall \lambda \in \mathbb{R}, E(\lambda\mu) = \lambda^2 E(\mu) > E(\mu_1)$ which is of course false.
- If we note $\mathbf{N} = (a, b, c)^T$ then for any point $\mathbf{X} = (x, y, z)^T \in \mathbb{R}^3$, $(a \cdot x + b \cdot y + c \cdot z + d)^2 = ||\mathbf{N}||_2^2 \cdot d(\mathbf{X}, \pi_\mu)^2$, where π_μ is the plane defined by μ and $d(\mathbf{X}, \pi_\mu)$ is the orthogonal distance of point $\mathbf{X}$ to plane π_μ.

This can be seen easily by computing the distance $d(\mathbf{X}, \pi_\mu)$. If we note $\mathbf{X}^\perp$ the orthogonal projection of X on π_μ and $\mathbf{N}^\perp = \mathbf{N}/||\mathbf{N}||_2$ then we have:

$$\begin{aligned} & \mathbf{X}^\perp = \mathbf{X} + \lambda \mathbf{N}^\perp \\ \Leftrightarrow \quad & a \cdot x^\perp + b \cdot y^\perp + c \cdot z^\perp + d = a \cdot x + b \cdot y + c \cdot z + d + \lambda(a^2 + b^2 + c^2)/||\mathbf{N}||_2 \\ \Leftrightarrow \quad & 0 = a \cdot x + b \cdot y + c \cdot z + d + \lambda ||\mathbf{N}||_2 \end{aligned}$$

where λ is the signed distance to the plane. Taking the square of the last equation then proves the result.

- If the additive noise is null, the ideal plane parameters are solution of equation (A.3) since $E(\mu_0) = 0$.

Depending on the constraint that one imposes, the minimization of (A.3) do not have the same meaning.

This section is organized as follow. We will first define the least squares problem minimizing the distance along the z-axis. We will see that in that case, an estimation of the expected error after re-projection on the plane can be given and that this estimator is the BLUE estimator for this particular problem. In a second part, we will define the least squares problem on the unitary sphere of parameters. At last, we will define the classical least squares problem that is usually proposed for plane estimation in 3D.

A.2.1 z-distance minimization: $c = c_0 = cste, c \neq 0$

The constraint $c = c_0 = cste, c \neq 0$ is equivalent to finding the plane minimizing the distance between each point and its projection on the plane along the z-axis. This can be easily seen by rewriting (A.3):

$$\hat{\mu} = \underset{(a,b,c=cste,d)}{\arg\min} \frac{1}{c^2} \sum_{i=1}^{N} \left(z(\mathbf{x_i}) - \frac{-(a \cdot x_i + b \cdot y_i + d)}{c} \right)^2 \tag{A.4}$$

Using matrix notations, $E(\mu)$ can be written as:

$$E(\mu) = ||\mathbf{A}\mu - \mathbf{b}||_2^2 \tag{A.5}$$

with

$$\mathbf{A} = \frac{1}{c} \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} z(\mathbf{x_1}) \\ z(\mathbf{x_2}) \\ \vdots \\ z(\mathbf{x_n}) \end{pmatrix} \text{ and } \mu = - \begin{pmatrix} a \\ b \\ d \end{pmatrix}$$

Note that this is a slight abuse of notations since the vector μ that we search here is 3-dimensional instead of being 4-dimensional as in the introduction.

Solution

Proposition 3 *The solution of Equation (A.4) is an unbiased estimator of the real plane parameter μ_0 and its covariance can be expressed as a function of the noise variance σ^2:*

$$cov(\hat{\mu}) = \mathbb{E}\left((\hat{\mu} - \mu_0)(\hat{\mu} - \mu_0)^T \right) = \sigma^2 (\mathbf{A}^T \mathbf{A})^{-1} \tag{A.6}$$

Proof Let's first find the solution of Equation (A.4). Deriving equation (A.5) and equating it to 0 to find the minimum, we obtain:

$$\mathbf{A}^T\mathbf{A}\hat{\mu} = \mathbf{A}^\mathbf{T}\mathbf{b} \tag{A.7}$$

Since $\mathbf{A}^T\mathbf{A}$ is rang complete, we can invert it and obtain the minimum:

$$\hat{\mu} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{A.8}$$

Developing $\mathbf{b}$ to express it as a function of the ideal parameters gives:

$$\begin{array}{rcl} \hat{\mu} & = & (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \\ \hat{\mu} & = & (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{b_0} + \varepsilon) \\ \hat{\mu} & = & (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{A}\mu_0 + \varepsilon) \\ \hat{\mu} & = & \mu_0 + (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\varepsilon \end{array} \tag{A.9}$$

At last using the linearity of the expectation operator gives the first result: $\mathbb{E}(\hat{\mu}) = \mu_0$. Using this result, the covariance of $\hat{\mu}$ can be easily computed and expressed as a function of the noise variance σ^2.

$$\begin{array}{rcl} \mathrm{cov}(\hat{\mu}) & = & \mathbb{E}\left((\hat{\mu} - \mu_0)(\hat{\mu} - \mu_0)^T\right) \\ & = & \mathbb{E}\left((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\varepsilon\varepsilon^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T}\right) \\ & = & (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbb{E}(\varepsilon\varepsilon^T)\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T} \\ & = & (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\sigma^2 I_n\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T} \\ & = & \sigma^2(\mathbf{A}^T\mathbf{A})^{-T} \\ & = & \sigma^2(\mathbf{A}^T\mathbf{A})^{-1} \end{array} \tag{A.10}$$

□

Note that using constant data with an additive noise ε, then $\mathrm{cov}(\hat{\mu}) = \mathrm{cov}(\varepsilon)$ which is what should be expected.

It can be shown (see [Beck and Arnold, 1977] for more precision on that property) that if the ε_i are independents and of constant variance σ^2, this estimator is optimal in terms of minimal covariance of μ.

Error using the estimated parameters

Let's now compute the expected error that we make by projecting the data on the estimated plane along the z-axis (re-projection error). First it can be noted that for any point from the data, the square re-projection error at this point is given by:

$$e_i^2 = \left(z_{\pi_0}(\mathbf{x_i}) - \frac{-(\hat{a}\cdot x_i + \hat{b}\cdot y_i + \hat{d})}{c}\right)^2 = \tilde{\mathbf{x_i}}^T(\hat{\mu} - \mu_0)(\hat{\mu} - \mu_0)^T\tilde{\mathbf{x_i}} \tag{A.11}$$

with $\tilde{\mathbf{x_i}} = \frac{1}{c}(x_i\ y_i\ 1)^T$. Using the linearity of the expectation operator and Equation (A.6) we then have:

$$\mathbb{E}(e_i^2) = \sigma^2\tilde{\mathbf{x_i}}^T(\mathbf{A}^T\mathbf{A})^{-1}\tilde{\mathbf{x_i}} \tag{A.12}$$

This last formula shows that the error depends on the 2-dimensional distribution of the points.

The following proposition gives another formulation of the re-projection error. It shows that the expected mean square error has a very simple expression that only depends on the noise variance and the dimension of the data. This result was observed experimentally on simulated data as shown in Figure A.1.

Proposition 4 (Reprojection error) *The expected mean square re-projection error is a function of the variance of the noise and the dimension of the data:*

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}\left(z_{\pi_0}(\mathbf{x_i}) - \frac{-(\hat{a}\cdot x_i + \hat{b}\cdot y_i + \hat{d})}{c}\right)^2\right) = \frac{1}{n}\mathbb{E}\left(||\mathbf{A}\hat{\mu} - \mathbf{b_0}||_2^2\right) = \frac{\sigma^2 \cdot rank(A)}{n} \quad \text{(A.13)}$$

Proof Let's first rewrite the total square re-projection error using equation (A.9):

$$\begin{aligned} ||\mathbf{A}\hat{\mu} - \mathbf{b_0}||_2^2 &= ||\mathbf{A}\hat{\mu} - \mathbf{A}\mu_0||_2^2 \\ &= ||\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\varepsilon||_2^2 \\ &= \varepsilon^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-T}\mathbf{A}^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\varepsilon \\ &= \varepsilon^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-T}\mathbf{A}^T\varepsilon \\ &= \varepsilon^T\mathbf{C}\varepsilon \end{aligned} \quad \text{(A.14)}$$

where $\mathbf{C} = \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-T}\mathbf{A}^T$. Taking the expectation of this error and developing the expression gives:

$$\begin{aligned} \mathbb{E}(||\mathbf{A}\hat{\mu} - \mathbf{b_0}||_2^2) &= \sum_{i,j}\mathbb{E}(\varepsilon_i c_{i,j}\varepsilon_j) \\ &= \sum_{i,j} c_{i,j}\mathbb{E}(\varepsilon_i\varepsilon_j) \\ &= \sum_{i,j} c_{i,j}\sigma^2\delta_{i,j} \\ &= \sigma^2 \cdot \text{trace}(\mathbf{C}) \end{aligned} \quad \text{(A.15)}$$

Let's now prove that $\text{trace}(\mathbf{C}) = \text{rank}(\mathbf{A})$. The first thing is to note that $\mathbf{CA} = \mathbf{A}$.

Taking the SVD decomposition of $\mathbf{A}$ we then have $\mathbf{A} = \mathbf{U\Sigma V}$, where $\mathbf{V}$ is $n\times n$ orthogonal matrix, $\mathbf{U}$ is a 3×3 orthogonal matrix and $\mathbf{\Sigma}$ is a $3\times n$ diagonal matrix. We note λ_i, $1 \leq i \leq 3$, the non-null singular values $\mathbf{A}$ and $\mathbf{u_i}$ (resp. $\mathbf{v_i}$) the left (resp. right) singular vector associated to them. We have:

$$\mathbf{A}\mathbf{v_i} = \lambda_i\mathbf{u_i} = \mathbf{C}\lambda_\mathbf{i}\mathbf{u_i} \quad \text{(A.16)}$$

which means that $\mathbf{u_i}$ is an eigen vector of $\mathbf{C}$ associated to the eigen value 1. Since $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{A})$ then the vectors $\mathbf{u_i}$ are the only non-null eigen vectors of $\mathbf{C}$ and $\text{trace}(\mathbf{C}) = \text{rank}(\mathbf{A})$.

□

This last proposition, can be easily extended to N-dimensional data on an hyperplane with noise along one direction (since its proof is done using matrices).

At last, let's remark that this computation is invariant to a compression of the z-axis. Indeed, taking $\alpha\mathbf{b}$ instead of $\mathbf{b}$ gives $\alpha\hat{\mu}$ instead of μ, which is exactly the same.

Weighted z-minimization

When one has extra information on the data points, it can be interesting to add weights when computing the optimal plane parameters. This can be the case for instance when the 2-dimensional points are not regularly distributed or when some extra information on the points precision are available.

Minimizing the weighted problem is pretty similar to what was done previously. Using the same notation as before, we are now looking for a vector $\hat{\mu}_W$ minimizing:

$$E_W(\mu) = ||\sqrt{\mathbf{W}}(\mathbf{A}\mu - \mathbf{b})||_2^2 \quad \text{(A.17)}$$

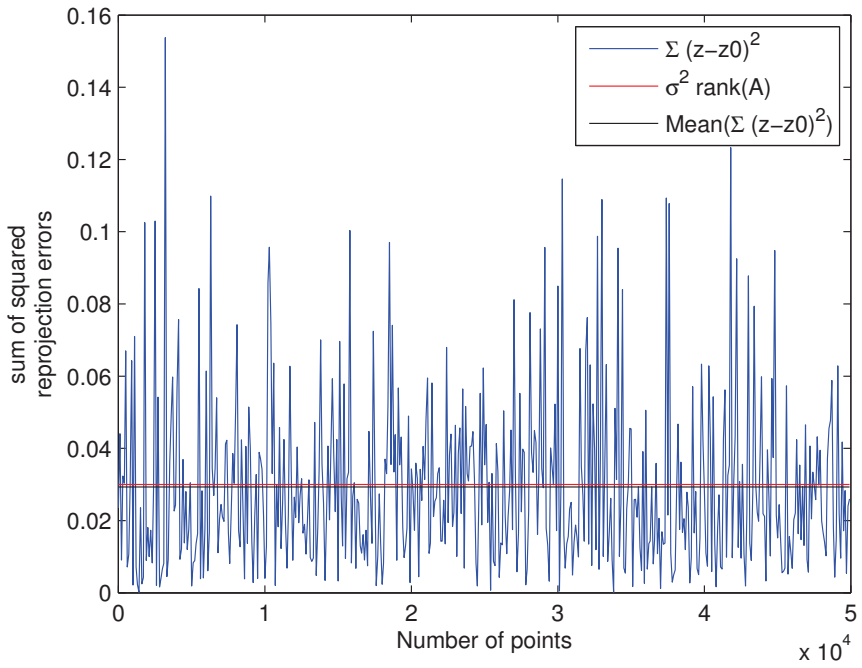


Figure A.1: Sum of squared re-projection errors for simulated data. The figure shows that this sum do not depend on the number of points. Moreover, the mean result is close to what is expected from Proposition 4.

where $\mathbf{W}$ is $n \times n$ diagonal matrix of weights. Using the same approach as before, the solution is given by:

$$\begin{aligned} \hat{\mu}_W &= (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b} \\ &= \mu_0 + (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\varepsilon \end{aligned} \tag{A.18}$$

Taking the expectation of that shows that this estimator is still unbiased. As before, the covariance of $\hat{\mu}$ can be computed, except that this time, no simplification is possible:

$$\begin{aligned} \mathrm{cov}(\hat{\mu}_W) &= \mathbb{E}\left((\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\varepsilon\varepsilon^T\mathbf{W}\mathbf{A}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-T}\right) \\ &= \sigma^2(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{A}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-T} \end{aligned} \tag{A.19}$$

The expected re-projection at each point can still be computed as before, however, the global re-projection error has no simple expression because there are no matricial simplification.

A.2.2 Minimization on the sphere of parameters: $||\mu||_2^2 = 1$

Another solution to find the best parameter set is to look for μ minimizing E such that $||\mu||_2^2 = 1$. Using matrix notations, we can write:

$$E(\mu) = \mu^T\mathbf{A}_P^T\mathbf{A}_P\mu = \mu^T\mathbf{B}_P\mu \tag{A.20}$$

with

$$\mathbf{A_P} = \frac{1}{c}\begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{pmatrix} \text{ and } \mu = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

Note that this time μ is a 4-dimensional parameter vector as in the introduction.

Solution

The matrix $\mathbf{B}_P$ is real symmetric therefore diagonalisable. Moreover, since the minimization is a sum of square, the eigen values of $\mathbf{B}_P$ are all positive.

Let's note $\mathbf{e}_0$, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ the eigen vectors of $\mathbf{B}_P$ and v_0, v_1, v_2, v_3 its eigen values, such that $v_0 \leq v_1 \leq v_2 \leq v_3$. Each vector μ can be written as a linear combination of $\mathbf{e}_0$, $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$:

$$\mu = \alpha_0\mathbf{e}_0 + \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_3\mathbf{e}_3 \tag{A.21}$$

such that $\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1$. Then:

$$E(\mu) = v_0\left(\alpha_0^2 + \frac{v_1}{v_0}\alpha_1^2 + \frac{v_2}{v_0}\alpha_2^2 + \frac{v_3}{v_0}\alpha_3^2\right) \geq v_0 \tag{A.22}$$

Therefore, E is minimum for $\mu = \mathbf{e}_0$.

A.2.3 Orthogonal minimization: $a^2 + b^2 + c^2 = 1$

Whenever $a^2 + b^2 + c^2 = 1$, then for any $\mathbf{X} = (x, y, z)^T \in \mathbb{R}^3$, $a \cdot x + b \cdot y + c \cdot z + d$ is the orthogonal distance of point $\mathbf{X}$ to the plane generated by $\mu = (a, b, c, d)^T$, π_μ.

Proposition 5 (Orthogonal minimizer) *The plane minimizing the orthogonal distance to the points is defined by the barycenter of the points and the smallest the eigen vector of the covariance matrix of the* 3*-dimensional points.*

Proof Let's rewrite function E in it's unconstrained form:

$$E'(\mu) = \sum_{i=1}^{N}(a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d)^2 + \lambda(a^2 + b^2 + c^2 - 1) \tag{A.23}$$

where λ is the Lagrange multiplier. Deriving E' with respect to a, b, c, d and λ and equating everything to 0 gives the following system:

$$\begin{cases} \sum_{i=1}^{N} x_i(a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d) + \lambda a = 0 \\ \sum_{i=1}^{N} y_i(a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d) + \lambda b = 0 \\ \sum_{i=1}^{N} z(\mathbf{x_i})(a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d) + \lambda c = 0 \\ \sum_{i=1}^{N}(a \cdot x_i + b \cdot y_i + c \cdot z(\mathbf{x_i}) + d) = 0 \\ a^2 + b^2 + c^2 = 1 \end{cases} \tag{A.24}$$

Replacing d by the value given by the fourth equation in all the other equations is then equivalent to finding the solution of the following constrained problem:

$$\min_{\mathbf{N}=(a,b,c)^T} \sum_{i=1}^{N} \left(a \cdot (x_i - \bar{x}) + b \cdot (y_i - \bar{y}) + c \cdot (z(\mathbf{x_i}) - \bar{z})\right)^2 \tag{A.25}$$

under the constraint $||\mathbf{N}||_2^2 = a^2 + b^2 + c^2 = 1$. This can be written using matrix notations as:

$$\min_{\mathbf{N}} (\mathbf{A}_\perp \mathbf{N})^T (\mathbf{A}_\perp \mathbf{N}) = \mathbf{N}^T \mathbf{B}_\perp \mathbf{N} \tag{A.26}$$

where $\mathbf{B}_\perp$ is the covariance matrix of the point set $(x_i, y_i, z(\mathbf{x_i}))_{i=1..N}$. Following the same approach as for the minimization on the sphere, it can be easily demonstrated that the solution vector $\hat{\mathbf{N}}$ is the eigen vector associated to the smallest eigen value of the covariance matrix.

□

A.2.4 Conclusion on the theoretical results

We proposed three different ways to pose the least squares problem for plane estimation. In our particular case, the only random part in the data points is their z coordinate. The z-minimization approach (section A.2.1) then seems more appropriate since the minimization is made on the random parts of the data. As opposed to that, the two other approaches, including the least squares solution usually used with pure 3D points, mix every coordinates of the points for their minimization. At last, it can be proven that in this particular case, the least squares estimator of section A.2.1 is the BLUE estimator (see [Sen and Srivastava, 1990] for instance). All of this then suggests that we choose the z least squares estimator to compute planes with our particular data.

A.3 Robust regression

The least squares regression is adapted to the case where no outliers are present among the points tested. This situation is the one encountered using the algorithm of Chapter 3 since a hard thresholding is done in the region growing which excludes most outliers (at least the ones that are far from the expected plane). However, in Chapter 2 a split and merge procedure is proposed which suggests the presence of outliers until a good solution is reached.

In this section, we do an exhaustive survey of existing techniques on robust regression.

A.3.1 Regression diagnostics

A first method for robust regression is the so-called *Regression diagnostics* [Belsey et al., 2005]. The idea here is to try to detect possible outliers from a first estimation of the regression parameters. The algorithm outline is the following:

1. Estimate the regression parameters.
2. Mark points whose residual error is above a preset threshold as outliers and do another parameter estimation with the remaining inliers.
3. Re-estimate the parameters with the new inlier list.

4. If the difference between the two estimations is not to large stop. Otherwise go back to step 2.

The main drawback of this technique is that a good final result is not guaranteed especially if the first estimation is bad. However, in presence of a not too large amount of outliers and more importantly with reasonable error.

Another version of this algorithm tries to point out the points causing the largest error in the estimation. This is done by computing for each point, a new model without it, and by rejecting the point that changes the most the error estimation. The algorithm stops when the measure of fit at an iteration is acceptable.

A.3.2 M-estimators

A popular approach for robust parameter estimation are the M-estimators. The point of this technique is to find the parameter vector μ minimizing the problem:

$$\begin{aligned}\hat{\mu} &= \underset{(a,b,c=cste,d)}{\arg\min} \sum_i \rho(r_i) \\ &= \underset{(a,b,c=cste,d)}{\arg\min} \sum_i \rho\left(\left| z(\mathbf{x_i}) - \frac{-(a \cdot x_i + b \cdot y_i + d)}{c} \right|\right)\end{aligned} \tag{A.27}$$

where ρ is a positive-definite function with a unique minimum at 0, chosen to be less increasing than square and r_i is the residual error at each point. Note that by taking $\rho : x \mapsto x^2$ we find the classical least squares problem given by Eq. A.4.

To solve this problem, the classical approach is to implement it as a re-weighted least squares problem. The first thing to note is that μ is solution of the following equation system.

$$\sum_i \rho'(r_i) \frac{\partial r_i}{\partial \mu_j} = 0, \text{ for } \mu_j \in \{a, b, d\} \tag{A.28}$$

Introducing the so-called *weight function* $w : x \mapsto \rho'(x)/x$, we obtain:

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial \mu_j} = 0, \text{ for } \mu_j \in \{a, b, d\} \tag{A.29}$$

We notice that the Equation system (A.29) is the same that we would obtain by solving the iterated re-weighted least squares problem:

$$\hat{\mu}^{(k)} = \underset{(a,b,c=cste,d)}{\arg\min} \sum_i w(r_i{}^{(k-1)}) r_i^2 \tag{A.30}$$

Several ρ functions are commonly used for the M-estimator problem. The choice of a function over another can be based on its convexity as well as its behaviour in presence of a standard normal distribution with no outlier. According to [Rey, 1983], one of the best ρ functions which yields to a nice converging scheme is the "*Fair*" function defined as:

$$\rho : x \mapsto c^2 \left(\frac{|x|}{c} - \log\left(1 + \frac{|x|}{c}\right)\right) \tag{A.31}$$

where the constant c is often set to 1.3998 to achieve a 95% asymptotic efficiency in presence of a normal distribution.

A.3.3 Hough transform

A standard procedure to robust regression is to use the Hough Transform [Hough, 1959]. The idea is to define a quantized parameter space to have another representation of the data. Then each possible triplet of points uniquely defines a plane which corresponds to a particular quantized value within this parameter space. Once all the triplets have been considered, the quantized parameter value that has the largest number of votes is the one corresponding to the main plane transformation.

This procedure is rather robust even in presence of a large amount of gross outliers. However, the number of tests required to compute a final transformation makes it hard to use in its raw form. To limit the computations, on can think of adding randomness to the procedure which make it more looks like the RANSAC algorithm.

A.3.4 RANSAC

A faster alternative to the Hough Transform is the RANSAC (*RANdom SAmpling Consensus*) algorithm [Fischler and Bolles, 1981] which consists in the following steps:

1. Sort out a N_{tri} random triplets of points and compute their associated planes.
2. For each triplet, find among all the remaining points which one are the inliers (residual error less than a threshold).
3. keep the triplet with the largest amount of inliers as the final estimator.

The number of iteration can be estimated from prior knowledge on the real percentage of inliers p_{in}. Indeed, the probability that among the N_{tri} triplets one of them is actually a triplet of inliers is:

$$P = 1 - \left(1 - p_{in}^3\right)^{N_{tri}} \tag{A.32}$$

which implies the following number of iterations:

$$N_{tri} = \frac{\log(1-P)}{\log\left(1 - p_{in}^3\right)} \tag{A.33}$$

For instance if one wishes to be sure up to $P = 99\%$ to get at least one triplet of inliers supposing 40% of inliers, at least 70 iterations are necessary.

The main drawback of this approach is the threshold parameter which is critical for a good solution. The following approach is pretty similar to the RANSAC procedure but propose a way to set this threshold when the amount of outliers is less than 50%.

A.3.5 Least median of squares

The least median of squares procedure consists in solving the following non-linear minimization problem:

$$\hat{\mu} = \underset{(a,b,c=cste,d)}{\arg\min} \ \text{median}\ r_i^2 \tag{A.34}$$

Since no direct estimation of the solution of Eq. A.34 is possible, a random procedure similar to the one done in the RANSAC algorithm is usually used. The only difference is in the second and third step: instead of estimating for each plane the number of points whose residual is

within a threshold, the median residual is computed. The best triplet is then the one with the lowest square residual.

Compared to the RANSAC algorithm, no prior on the data precision is required (which is a critical parameter of RANSAC algorithms). However, in presence of more than 50% of outliers, the least median of squares estimator has a good chance to fail.

At last, as pointed out in [Rousseeuw and Leroy, 1987], the Least median of squares estimation is poor in presence of Gaussian noise (the same can be stated for RANSAC algorithm). It is then common to refine the result given by each of the algorithms by computing the least squares estimator from the final inliers of each method. In the least median of squares case, the inlier threshold is deduced from the robust estimate of the standard deviation [Rousseeuw and Leroy, 1987]:

$$\hat{\sigma} = 1.4826\,(1 + 5/(N-p))\ \text{median}\ |r_i| \tag{A.35}$$

The points are then considered as inliers if their residual to the estimated plane is less than $2.5\hat{\sigma}$.

A.4 Experimental results

In this section, we compare the methods previously described. We first describes our experiments done on the least squares fitting which is more adapted with the method described in Chapter 3. Then we propose other experiments for robust estimation that are adapted to situations encountered in the split and merge algorithm of Chapter 2.

A.4.1 Least squares

We propose here several experiments to point out which least square estimation is preferable for the situations encountered with the algorithm of Chapter 3. In the first two experiments, a set of 3D points was created from a set of 2D points on a regular grid. Each 3D point is obtained by computing the projection on a given plane and by adding a uniformly distributed Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$. In the first experiment the noise is only added along the third direction whereas this is done independently along every direction for the second experiment. For each experiment, the error was averaged through different plane orientations.

Figure A.2 shows the evolution of the error along the z-axis with the initial number of points for the three methods. Figure A.3 shows the evolution of the angle between the real plane and estimated plane with the initial number of points for the three methods.

From these experiments it can be seen that the least squares plane estimation with $c = cste \neq 0$ tend to give better results than the orthogonal least squares plane estimation when the additive corruption noise is along the z-axis. This is however the opposite when the additive noise is along every direction. The least square plane estimation with $||\mu||_2^2 = 1$ seems to give similar results to the least squares with $c = cste$ but is never better. At last, it can be noted that each of the three methods tend to the real solution both when the number of points tend to infinity and when the 2D grid span is a lot larger than the additive noise.

In the third experiment, piecewise planar disparity maps (Toulouse, village1, village2, countryside) were corrupted with an additive Gaussian noise (along the z-direction). Then the three plane estimation methods were used to run the region growing algorithm. Such was done to compare the results both in quality and computation time obtained for the 3 approaches. For a fair comparison the error measurements were done using a same classification for the 3 methods.

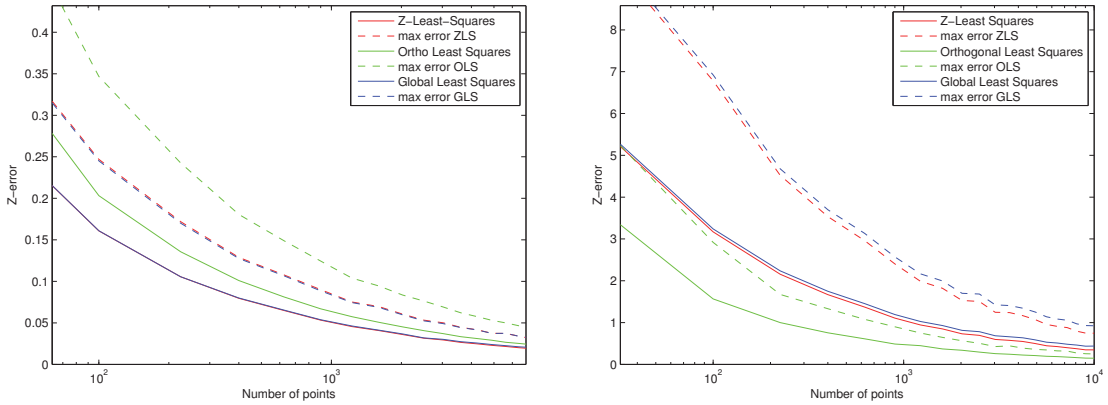


Figure A.2: Mean error (plain lines) and max error (dashed lines) along the z-axis. In red, least squares with $c = cste \neq 0$; in green, orthogonal least squares; in blue, least squares with $||\mu||_2^2 = 1$. Left image: Data set corrupted with an additive Gaussian noise on the z-axis only. Right image: Data set corrupted with an additive Gaussian noise on each axis.

	Toulouse			Village1			Village2			Countryside		
	time	error	N_π	time	error	N_π	time	error	N_π	time	error	N_π
ZLS	**9.9s**	0.009	95	49.2s	**0.0234**	193	**1m52s**	**0.0272**	88	**56s**	0.0024	11
OLS	11.5s	**0.009**	93	**47.3s**	0.0285	191	2m23S	0.0272	93	57s	**0.0024**	11
GLS	-	0.0475	-	-	0.0699	-	-	0.0504	-	-	0.026	-

Table A.1: Execution time (s), mean root square residual error (pixel) and number of detected planes on the 4 disparity maps corrupted with Gaussian noise. First line: least squares approach with $c = cste \neq 0$. Second line: orthogonal least squares approach. Third line: least squares approach with $||\mu||_2^2 = 1$.

Table A.1 and Figure A.4 show the results of this experiment.

In this experiment, the least squares minimization on the parameter sphere failed. This is due to the initialisation patches which were too small for a good first estimation and therefore a good region growing. As shown on Figure A.4, the plane classification obtained with the z-least squares approach and the one obtained with the orthogonal least squares are pretty similar. The residual errors are located at the same place but at are slightly different. This is confirmed by the numerical results of Table A.1. Both the computation times and errors are almost the same for the two approaches even if the $c = cste$ least squares are slightly better.

Even if the difference is mere, we chose to use the $c = cste$ least squares approach for our algorithm. This choice was also driven by the fact this method is invariant by compression along the z-axis which avoids setting the baseline parameter as an input.

Least squares V.S. M-estimators

We at last tried to estimate a plane using both least squares and M-estimators. In this experiment, several "slanted" planes were simulated and corrupted along their $z-$ axis by an

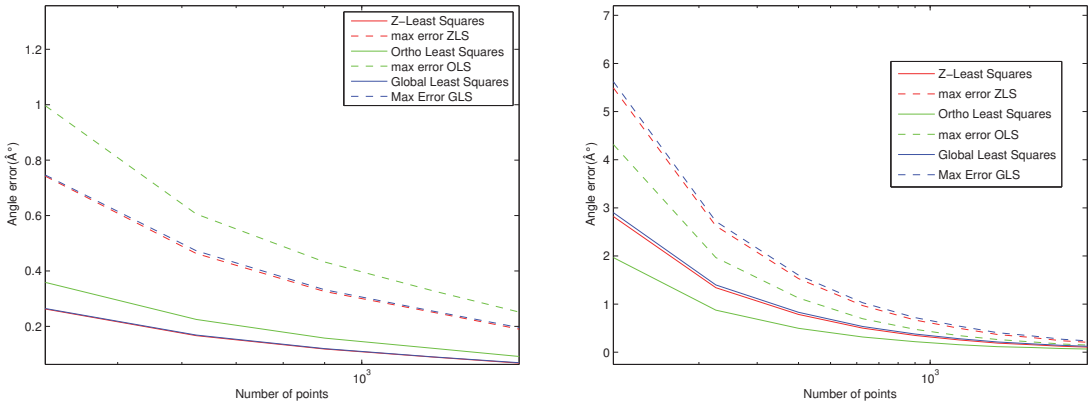


Figure A.3: Mean error angle (plain lines) and max error angle (dashed lines) between the real plane and the estimated plane. In red, least squares with $c = cste \neq 0$; in green, orthogonal least squares; in blue, least squares with $||\mu||_2^2 = 1$. Left image: Data set corrupted with an additive Gaussian noise on the z-axis only. Right image: Data set corrupted with an additive Gaussian noise on each axis.

additive Gaussian noise. Then for each configuration, the final plane was estimated by both least squares and M-estimators using the fair function.

The obtained estimation gave the same error order with a slight preference for Least squares when the noise variance gets large.

A.4.2 Robust estimators

In this section we propose two sets of experiments to find out which estimator is more adapted to the situations encountered in Chapter 2. We considered the two following situations:

- Two planes separated at their intersection forming an angle α.
- Two planes identically oriented separated by a step edge Δ_z.

For each experiment, the data were polluted with an additive i.i.e. Gaussian noise along the $z-$direction.

Since the goal here is to measure the capacity of each method to find the right model, we computed the Mean Square Errors of each algorithm considering only the points of the principal plane. For each situation, we ran the algorithm with the following proportions of points distributed according to the first plane: 50%, 75% and 100%. Table A.2 show the mean and median results obtained using different values angle α and step edge Δ_z for the three following algorithms: M-estimators (Mest), Least Median of Squares (LMS) without and with M-estimator refinement (M-LMS) and RANSAC (Ran) without and with M-estimator refinement (M-Ran). In this experiment, the threshold for the RANSAC algorithm was chosen as 2.5σ where σ is the standard deviation of the additive Gaussian noise. The threshold for the refinement of the Least Median of Squares was chosen as $2.5\hat{\sigma}$ where $\hat{\sigma}$ is the robust standard deviation computed according to Eq. A.35.

From the results, one can see that the RANSAC algorithm refined using the M-estimators almost always gives the best results, supposing that you know the right threshold for it.

Experiment	Intersection				Step edge				Single plane	
Inliers:	50%		75%		50%		75%		100%	
	mean	median	mean	median	mean	median	mean	median	mean	median
M-est	**3806**	1293	1.93	1.47	**1961**	**1878**	5.3	5.26	**2 E^{-4}**	**2 E^{-4}**
Ran	25127	0.27	0.072	0.056	19857	19434	0.081	0.076	0.048	0.039
LMS	20550	0.25	0.053	0.045	23341	22308	0.062	0.05	0.038	0.038
M-Ran	5832	**6.5 E^{-3}**	**8 E^{-4}**	**5 E^{-4}**	3116	4911	**6 E^{-4}**	**6E^{-4}**	8 E^{-4}	6 E^{-4}
M-LMS	33969	0.053	0.0046	0.0043	3596	4862	0.0032	0.0031	0.0062	0.0057
M-Ran:σ	15666	0.13	0.03	0.023	0.034	0.029	8613	0.16	0.026	0.04
M-Ran:5σ	23574	0.043	0.027	0.004	23496	22496	3 E^{-4}	3 E^{-4}	2 E^{-4}	2 E^{-4}
M-Ran:10σ	378	1.01	0.54	0.13	23564	22504	3 E^{-4}	3 E^{-4}	5 E^{-4}	5 E^{-4}

Table A.2: Mean Square Error of robust algorithms for the points of the first model.

The results are slightly better than the ones using Least Median of Squares. However, with the latter algorithm the threshold can be set automatically. At last, one can see that the M-estimators are not well adapted to robust estimation. In most cases, they are not able to separate the two models (MSE $>$ 1000). Moreover, when they do, the results are less precise than the ones obtained with RANSAC or Least Median of squares. The two exception situations are:

- A single plane describes the data. In This case the M-estimators are a better descriptor because they are more adapted to this type of data.
- A step edge where half the points are distributed according to one plane and the other half corresponds to the other plane. In this situation, neither RANSAC nor Least Median of Squares are able to separate the two models. Then once again, the M-estimators give a better mean description. Note that if a smaller threshold is used for RANSAC (σ instead of 2.5σ), the separation can be found.

The second part of Table A.2 shows the effect of using different threshold values for the RANSAC algorithm and the refinement with M-estimators. One can see that if a wrong threshold is used, the Least Median of Squares + robust standard deviation is preferable in almost all the configurations.

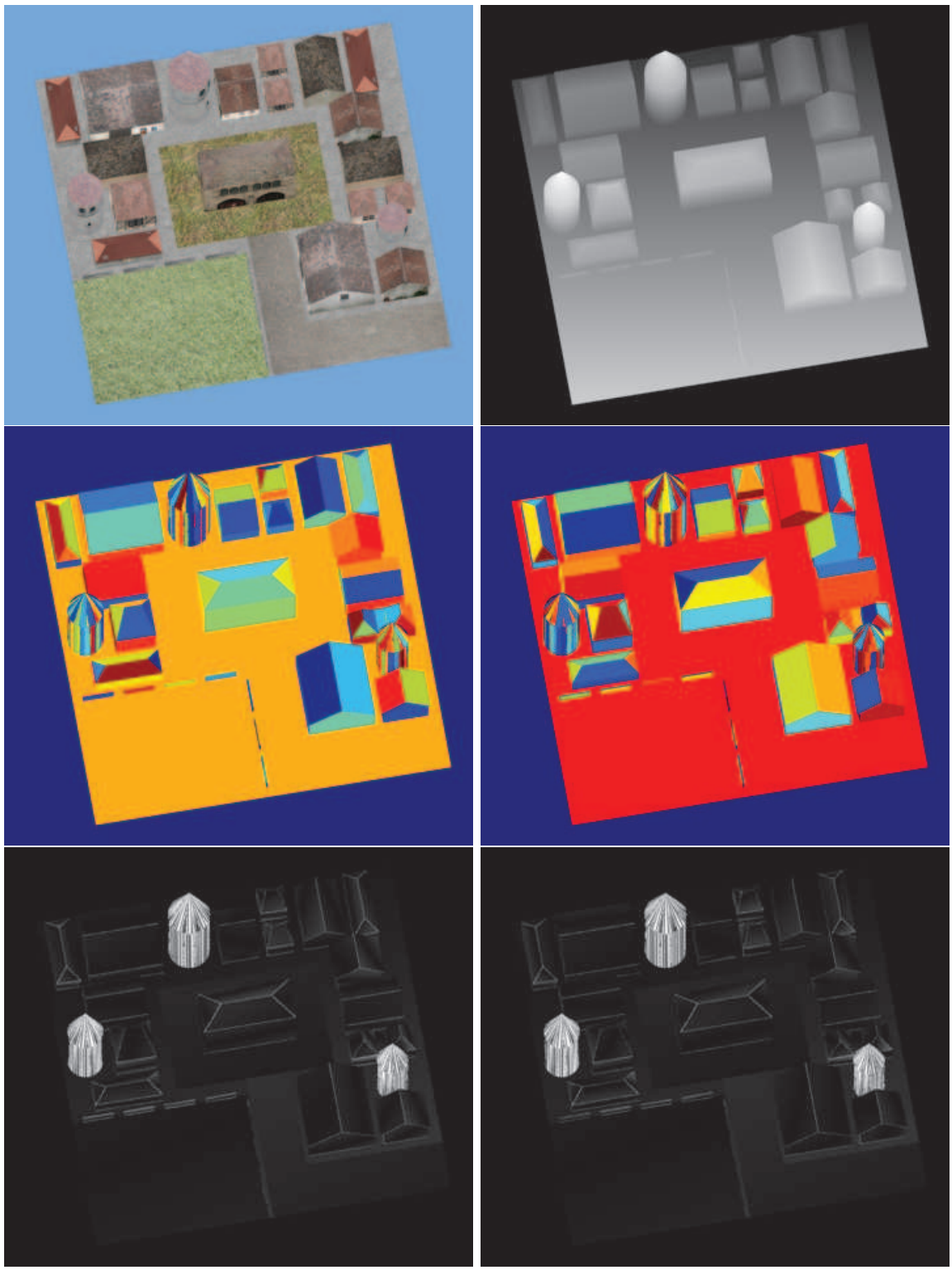


Figure A.4: Results on Village1 disparity map. First line: reference image(left) and disparity map(right). Second line: plane classification obtained with least squares minimization with $c = cste$(left) and orthogonal constraint(right). Third line: residual error obtained with least squares minimization with $c = cste$(left) and orthogonal constraint(right).

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19:716–723.

Ameri, B. and Fritsch, D. (2000). Automatic 3d building reconstruction using plane-roof structures. *ASPR 00*.

Bai, X. and Sapiro, G. (2007). A geodesic framework for fast interactive image and video segmentation and matting. *In IEEE International Conference on Computer Vision*.

Baillard, C. and Zisserman, A. (1999). Automatic reconstruction of piecewise planar models from multiple views. In *CVPR, 1999. IEEE Computer Society Conference on.*, volume 2, page 565 Vol. 2.

Beck, J. and Arnold, K. (1977). Parameter estimation in engineering and science. *Wiley series in probability and mathematical statistics*.

Belsey, D. A., Kuh, E., and Welsch, R. E. (2005). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons, Inc.

Besl, P. J. and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:167–192.

Bignone, F., Henricsson, O., Fua, P., and Stricker, M. (1996). Automatic extraction of generic house roofs from high resolution aerial imagery. *European Conference on Computer Vision*, 1064:83–96.

Blanchet, G., Buades, A., Coll, B., Morel, J., and Rougé, B. (2011). Fattening free block matching. *Journal of Mathematical Imaging and Vision*, pages 1–13.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34:344–371.

Boulanger, P. and Godin, G. (1992). Multiresolution segmentation of range images based on bayesian decision-theory. *In Proceedings of Intelligent Robots and Computer Vision*, pages 338–350.

Boykov, Y., Veksler, O., and Zabih, R. (1998). A variable window approach to early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1283–1294.

Brédif, M., Boldo, D., Deseilligny, M. P., and Maître, H. (2008). 3d building model fitting using a new kinetic framework. Technical report.

Burrus, N., Bernard, T. M., and Jolion, J.-M. (2009). Image segmentation by *a contrario* simulation. *Pattern Recognition*, 42(7):1520–1532.

Cao, F., Delon, J., Desolneux, A., Musé, P., and Sur, F. (2007). A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2):91–119.

Chauve, A.-L., Labatut, P., and Pons, J.-P. (2010). Robust piecewise-planar 3d reconstruction and completion from large scale unstructured point data. *Computer Vision and Pattern Recognition*.

Darrell, T. (1998). A radial cumulative similarity transform for robust image correspondence. *In Proceedings IEEE Conference on Computer Vision and Pattern Recognition*.

Delon, J. (2004). *Fine comparaison of images and other problems*. PhD thesis, Ecole Normale Supérieure de Cachan, France.

Delon, J. and Rougé, B. (2007). Small baseline stereovision. *Journal of Mathematical Imaging and Vision*, 28(3):209–223.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum-likelyhood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B*, 39.

Desolneux, A., Moisan, L., and Morel, J.-M. (2008a). *From Gestalt theory to image analysis*, volume 34 of *Interdisciplinary Applied Mathematics*. Springer Verlag, New-York. A probabilistic approach.

Desolneux, A., Moisan, L., and Morel, J.-M. (2008b). *From Gestalt Theory to Image Analysis: A Probabilistic Approach.* Springer Verlag.

Dick, A. R., Torr, P. H. S., and Cipolla, R. (2004). Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60:111–134.

Digne, J. (2010). *Géométrie Inverse: du nuage de points brut à la surface 3D. Théorie et Algorithmes*. PhD thesis, ENS Cachan (France).

Digne, J., Morel, J.-M., Audfray, N., and Mehdi-Souzani, C. (2010). The level set tree on meshes. In *Proceedings of the Fifth International Symposium on. 3D Data Processing,Visualization and Transmission*, Paris, France.

Durou, J.-D. (2007). Shape from shading, eclairages, réflexions et perspectives.

Durou, J.-D. and Courteille, F. (2007). Integration of a normal field without boundary condition. *Proceedings of the First International Workshop on Photometric Analysis For Vision (PACV)*.

Durupt, M. and Taillandier, F. (2006). Reconstruction automatique de bâtiments à partir d'un mne et de limites cadastrales: une approche opérationnelle. *Reconnaissance des Formes et Intelligence Artificielle*.

Facciolo, G. and Caselles, V. (2009). Geodesic neighborhoods for piecewise affine interpolation of sparse data. *In Proceedings of International Conference on Image Processing*.

Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images: The Laws That Govern The Formation of Imags of A Scene and Some of Their Applications.* The MIT Press.

Filin, S. and Pfeifer, N. (2006). Segmentation of airborne laser scanning data using a slope adaptative neighborhood. *International Journal of Photogrammetry and Remote Sensing*, 60:71–80.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395.

Fitzgibbon, A. W., Eggert, D. W., and Fisher, R. B. (1997). High-level cad model acquisition from range images. *Computer Aided Design*, 29:321–330.

Flamanc, D. and Maillet, G. (2005). Evaluation of 3d city model production from pleaides hr satellite images and 2d ground maps. *URBAN*, pages 46–51.

Flynn, P. (1990). *CAD-based computer vision: modelling and recognition strategies.* PhD thesis, Michigan State University.

Forlani, G., Nardinocchi, C., Scaioni, M., and Zingaretti, P. (2004). Building reconstruction and visualization from lidar data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34.

Fraundorfer, F., Schindler, K., and Bischof, H. (2006). Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24:395–406.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732.

Grimson, W. E. L. and Huttenlocher, D. P. (1991). On the verification of hypothetized matches in model-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13.

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., and Randall, G. (2008). Lsd: A fast line segment detector with a false detection control. *PAMI'08.*

Gruen, A. and Wang, X. (1998). Cc-modeler: a topology generator for 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:286–295.

Han, F., Tu, Z., and Zhu, S.-C. (2004). Range image segmentation by an effective jump-diffusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1138–1153.

Han, J., Volz, R., and Mudge, T. N. (1987). Range image segmentation and surface parameter extraction for 3d object recognition of industrial parts. *In Proceedings of International Conference on Computer Vision*, pages 380–386.

Haralick, R. (1980). Edge and region analysis for digital image data. *Computer Graphics Image Processing.*

Haralick, R. M. and Shapiro, L. G. (1993). *Computer and Robot Vision.* Addison-Wesley.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings of the fourth Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry*. Cambridge University Press.

Hirschmüller, H., Innocent, P. R., and Garibaldi, J. (2002). Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30.

Hoover, A., Jean-Baptiste, G., Jiang, X., and Flynn, P. (1996). A comparison of range image segmentation algorithms. *IEEE Transactions PAMI*, 18(7):673–689.

Horowidth, S. L. and Pavlidis, T. (1974). Picture segmentation by a direct split and merge procedure. *In Proceedings of the 2nd IJCPR*, pages 424–433.

Hough, P. (1959). Machine analysis of bubble chamber pictures. *International Conference on High Energy Accelerators and Instrumentation*, pages 554–556.

Igual, L., Preciozzi, J., Garrido, L., Almansa, A., Caselles, V., and Rougé, B. (2007). Automatic low baseline stereo in urban areas. *Inverse Problems and Imaging*, 1(2):319–348.

Jiang, X. Y. and Bunke, H. (1994). Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications*, 7:115–222.

Kada, M. (2006). 3d building generalization based on half-space modeling. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:58–64.

Kanazawa, Y. and Kawakami, H. (2004). Detection of planar regions with uncalibrated stereo using distribution feature points. In *British Machine Vision Conference*, pages 247–256.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contours models. *International Journal of Computer Vision*, pages 321–331.

Kimmel, R. and Bruckstein, A. M. (2003). Regularized laplacian zero crossings as optimal edge integrators. *International Journal of Computer Vision*, 53(3):225–243.

Koepfler, G., Lopez, C., and Morel, J.-M. (1994). A multiscale algorithm for image segmentation by variational method. *SIAM Journal on Numerical Analysis*, 31:282–299.

Labatut, P., Pons, J.-P., and Keriven, R. (09). Hierarchical shape-based surface reconstruction for dense multi-view stereo. *The 2009 IEEE International Workshop on 3-D Digital Imaging and Modeling*.

Lafarge, F., Descombes, X., Zerubia, J., and Deseilligny, P. M. (2008a). Building reconstruction from a single dem. In *Proc. IEEE CVPR*, Anchorage, Alaska, U.S.

Lafarge, F., Descombes, X., Zerubia, J., and Pierrot-Deseilligny, M. (2008b). Automatic building extraction from dems using an object approach and application to the 3d-city modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(3):365–381.

Lavest, J.-M., Viala, M., and Dhome, M. (1998). Do we really need an accurate calibration pattern to achieve a reliable camera calibration? *Proceedings of the 5th European Conference on Computer Vision*, 1:158–174.

Lin, C. Nevatia, R. (1998). Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72:101–121.

Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:1125.

Lotti, J.-L. and Giraudon, G. (1994). Correlation algorithm with adaptative window for aerial image in stereovision. *Image and Signal Processing for Remote Sensing*, 2315.

Lourakis, M. I. A. and Argyros, A. A. (2009). Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 96.

Lowe, D. G. (1985). *Perceptual organization and visual recognition*. Kluwer Academic Publisher.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767.

Moisan, L. and Stival, B. (2004). A probabilistic criterion to detect rigid point matches between images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201–218.

Mumford, D. and Shah, J. (1985). Boundary detection by minimizing functionals. *In Proceedings of International Conference on Acoustic, Speech and Signal Processing*, pages 22–26.

Musé, P., Sur, F., Cao, F., Gousseau, Y., and Morel, J.-M. (2006). An a contrario decision method for shape element recognition. *International Journal of Computer Vision*, 69(3):295–315.

Nevatia, R. and Price, K. (2002). Automatic and interactive modeling of buildings in urban environments from aerial images. *IEEE International Conference on Image Processing*, 3:525–528.

Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4).

Ortner, M., Descombes, X., and Zerubia, J. (2007). Building outline extraction from altimetric data in dense urban areas. *International Journal of Computer Vision*, 72:107–132.

Parvin, B. and Medioni, G. (1986). Segmentation of range images into planar surfaces by split and merge. *In Proceedings of Computer Vision and Pattern Recognition*, pages 415–417.

Peternell, M. and Steiner, T. (2004). Reconstruction of piecewise planar objects from point clouds. *Computer Aided Design*, 334:333–342.

Pierrot-Deseilligny, M. and Paparoditis, N. (2006). A multiresolution and optimization-based image matching approach: an application to surface reconstruction from spot-5hrs stereo imagery. *In International Society for Photogrammetry and Remote Sensing*, 36.

Poppinga, J., Vaskevicus, N., Birk, A., and Pathak, K. (2008). Fast plane detection and polygonalization in noisy 3d range images. *International Conference on Interlligent Robots and Systems*.

Poullis, C. and You, S. (2009). Automatic reconstruction of cities from remote sensor data. *Computer Vision and Pattern Recognition*.

Prados, E. (2004). *Application of the theory of the viscosity solutions to the Shape From Shading problem.* PhD thesis, Université de Nice-Sophia Antipolis (France).

Prazdny, K. (1987). Detection of binocular disparities. *Reading in computer vision: issues, problems, principles, paradigms*, pages 73–79.

Pu, S. and Vosselman, G. (2006). Automatic extraction of building features from terrestrial laser scanning. *International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, 36:320–325.

Rabin, J., Delon, J., Gousseau, Y., and Moisan, L. (2009). Mac-ransac: a robust algorithm for the recognition of multiple objects. *3DPVT'10*.

Rey, W. J. (1983). *Introduction to robust and quasi robust statistical methods*. Springer Verlag.

Robert, L. and Faugeras, O. (1991). Curve-based stereo: figural continuity and curvature. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley & Sons.

Roy, S. and Cox, I. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem. *In proceedings of International Conference on Computer Vision*, pages 492–499.

Sabater, N. (2009). *Reliability and accuracy in stereovision. Application to aerial and satellitel high-resolution images.* PhD thesis, École Nationale Supérieure de Cachan.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42.

Schmid, C. and Zisserman, A. (2000). The geometry and matching of line and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–234.

Schnabel, R., Degener, P., and Klein, R. (2007a). Completion and reconstruction with primitive shapes. *Computer Graphics Forum*, 26.

Schnabel, R., Wahl, R., and Klein, R. (2007b). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 2:461–464.

Sen, A. K. and Srivastava, M. (1990). *Regression analysis: theory, methods and applications*. Springer Verlag.

Sinha, S., Steedly, D., and Szeliski, R. (2008). Piecewise planar stereo for image based rendering. *In Proceedings of SIGGRAPH Asia*, 27.

Stewart, C. V. (1995). Minpran: A new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):925–938.

Taillandier, F. and Deriche, R. (2004). Automatic building reconstruction from aerial images: a bayesian framework. *International Archives of Photogrammetry, Remote Sensing and Spatial Information*, 35.

Taillandier, F., Guigues, L., and Deriche, R. (2003). A framework for constrained multi-scale range image segmentation. *In Proceedings of International Conference on Image Processing*.

Taylor, R., Savini, M., and Reeves, A. (1989). Fast segmentation of range imagery into planar regions. *Computer Vision, Graphics, and Imagery Processing*.

Toldo, R. and Fusiello, A. (2008). Robust multiple structures estimation with j-linkage. In *ECCV '08*, pages 537–547, Berlin, Heidelberg. Springer-Verlag.

Vallet, B. and Taillandier, F. (2005). Fitting constrained 3d models in multiple aerial images. *British Machine Vision Conference*.

Veksler, O. (2002). Stereo correspondence with compact windows via minimum ratio cycle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1654–1660.

Veksler, O. (2003). Fast variable window for stereo correspondence using integral images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:556–561.

Vincent, E. and Laganière, R. (2001). Detecting planar homographies in an image pair. In *2nd International Symposium on Image and Signal Processing and Analysys*, pages 182–187.

Vosselman, G. and Dijkman, S. (2001). 3d building model reconstruction from point clouds and ground planes. *International Archives of Photogrammetry and Remote Sensing*, XXXIV.

Vu, H.-H., Keriven, R., Labatut, P., and Pons, J.-P. (2009). Towards high-resolution large-scale multi-view stereo.

Werner, T. and Zisserman, A. (2002). New techniques for automated architectureal reconstruction from photographs. *In Proceedings of the 7th Europeean Conference on Computer Vision*, pages 541–555.

Xiang, R. and Wang, R. (2004). Range image segmentation based on split and merge clustering. *In Proceedings of the 17th International Conference on Pattern Recognition*, pages 614–617.

Xu, Y., Wang, D., Tao, F., and Shum, H.-Y. (2002). Stereo radial computation using radial adaptative windows. *International Conference on Pattern Recognition*, 3:595–598.

Yatziv, L. and Sapiro, G. (2006). Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, 15(5).

Yoon, K.-J. and Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656.

Zhang, W. and Kosecká, J. (2006). Nonparametric estimation of multiple structures with outliers. In *ECCV 06*, pages 60–74.

Zhang, Z. (1997). Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 2:161–195.

Zuliani, M., Kenney, C. S., and Manjunath, B. S. (2005). The multiransac algorithm and its application to detect planar homographies. In *ICIP*.