Vision à distance par champs de lumière

Grégoire Nieto

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel du 25 mai 2016

Préparée au sein **du Laboratoire Jean Kuntzmann**
à l'INRIA Rhône-Alpes
et de **l'Ecole Doctorale de Mathématiques, Sciences**
et Technologies de l'Information

Light Field Remote Vision

Présentée par

Grégoire NIETO

Thèse dirigée par **James CROWLEY**
et codirigée par **Frédéric DEVERNAY**

Thèse soutenue publiquement le **3 octobre 2017**,
devant le jury composé de :

M. Rémi RONFARD
Directeur de recherche à INRIA Grenoble - Rhône-Alpes, France, Président

M. Ivo IHRKE
Chercheur à INRIA Bordeaux - Sud-Ouest, France, Rapporteur

Mme. Christine GUILLEMOT
Directrice de recherche à INRIA Rennes - Bretagne Atlantique, France, Rapporteur

M. Patrick PEREZ
Distinguished Scientist chez Technicolor, Examinateur

M. James CROWLEY
Professeur à Grenoble INP, France, Examinateur

M. Frédéric DEVERNAY
Chercheur à INRIA Grenoble - Rhône-Alpes, France, Examinateur

Abstract

Light fields have gathered much interest during the past few years. Captured from a plenoptic camera or a camera array, they sample the plenoptic function that provides rich information about the radiance of any ray passing through the observed scene. They offer a pletora of computer vision and graphics applications: 3D reconstruction, segmentation, novel view synthesis, inpainting or matting for instance.

Reconstructing the light field consists in recovering the missing rays given the captured samples. In this work we cope with the problem of reconstructing the light field in order to synthesize an image, as if it was taken by a camera closer to the scene than the input plenoptic device or set of cameras. Our approach is to formulate the light field reconstruction challenge as an image-based rendering (IBR) problem. Most of IBR algorithms first estimate the geometry of the scene, known as a *geometric proxy*, to make correspondences between the input views and the target view. A new image is generated by the joint use of both the input images and the *geometric proxy*, often projecting the input images on the target point of view and blending them in intensity.

A naive color blending of the input images do not guaranty the coherence of the synthesized image. Therefore we propose a direct multi-scale approach based on Laplacian rendering to blend the source images at all the frequencies, thus preventing rendering artifacts.

However, the imperfection of the *geometric proxy* is also a main cause of rendering artifacts, that are displayed as a high-frequency noise in the synthesized image. We introduce a novel variational rendering method with gradient constraints on the target image for a better-conditioned linear system to solve, removing the high-frequency noise due to the *geometric proxy*.

Some scene reconstructions are very challenging because of the presence of non-Lambertian materials; moreover, even a perfect *geometric proxy* is not sufficient when reflections, transparencies and specularities question the rules of parallax. We propose an original method based on the local approximation of the sparse light field in the plenoptic space to generate a new viewpoint without the need for any explicit *geometric proxy* reconstruction. We evaluate our method both quantitatively and qualitatively on non-trivial scenes that contain non-Lambertian surfaces.

Lastly we discuss the question of the optimal placement of constrained cameras for IBR, and the use of our algorithms to recover objects that are hidden behind a camouflage.

The proposed algorithms are illustrated by results on both structured (camera arrays) and unstructured plenoptic datasets.

Keywords Image-Based Rendering, Computational Photography, 3D Reconstruction, Light Field, Plenoptic Imaging.

Résumé

Les champs de lumière ont attisé la curiosité durant ces dernières décennies. Capturés par une caméra plénoptique ou un ensemble de caméras, ils échantillonnent la fonction plénoptique qui informe sur la radiance de n'importe quel rayon lumineux traversant la scène observée. Les champs lumineux offrent de nombreuses applications en vision par ordinateur comme en infographie, de la reconstruction 3D à la segmentation, en passant par la synthèse de vue, l'*inpainting* ou encore le *matting* par exemple.

Dans ce travail nous nous attelons au problème de reconstruction du champ de lumière dans le but de synthétiser une image, comme si elle avait été prise par une caméra plus proche du sujet de la scène que l'appareil de capture plénoptique. Notre approche consiste à formuler la reconstruction du champ lumineux comme un problème de rendu basé image (IBR). La plupart des algorithmes de rendu basé image s'appuient dans un premier temps sur une reconstruction 3D approximative de la scène, appelée *proxy géométrique*, afin d'établir des correspondances entre les points image des vues sources et ceux de la vue cible. Une nouvelle vue est générée par l'utilisation conjointe des images sources et du *proxy géométrique*, bien souvent par la projection des images sources sur le point de vue cible et leur fusion en intensité.

Un simple mélange des couleurs des images sources ne garantit pas la cohérence de l'image synthétisée. Nous proposons donc une méthode de rendu direct multi-échelles basée sur les pyramides de laplaciens afin de fusionner les images sources à toutes les fréquences, prévenant ainsi l'apparition d'artefacts de rendu.

Mais l'imperfection du *proxy géométrique* est aussi la cause d'artefacts de rendu, qui se traduisent par du bruit en haute fréquence dans l'image synthétisée. Nous introduisons une nouvelle méthode de rendu variationnelle avec des contraintes sur les gradients de l'image cible dans le but de mieux conditionner le système d'équation linéaire à résoudre et supprimer les artefacts de rendu dus au *proxy*.

Certaines scènes posent de grandes difficultés de reconstruction du fait du caractère non-lambertien éventuel de certaines surfaces ; d'autre part même un bon *proxy* ne suffit pas, lorsque des réflexions, transparences et spécularités remettent en cause les règles de la parallaxe. Nous proposons méthode originale basée sur l'approximation locale de l'espace plénoptique à partir d'un échantillonnage épars afin de synthétiser n'importe quel point de vue sans avoir recours à la reconstruction explicite d'un *proxy géométrique*. Nous évaluons notre méthode à la fois qualitativement et quantitativement sur des scènes non-triviales contenant des matériaux non-lambertiens.

Enfin nous ouvrons une discussion sur le problème du placement optimal de caméras contraintes pour le rendu basé image, et sur l'utilisation de nos algorithmes pour la vision d'objets dissimulés derrière des camouflages.

Les différents algorithmes proposés sont illustrés par des résultats sur des jeux de données plénoptiques structurés (de type grilles de caméras) ou non-structurés.
Mots-Clés Rendu basé image, photographie computationnelle, reconstruction 3D, champ de lumière, imagerie plénoptique.

Publications liées à cette thèse

1. **G. Nieto**, F. Devernay et J. Crowley, "Placement optimal de caméras contraintes pour la synthèse de nouvelles vues", dans *Journees francophones des jeunes chercheurs en vision par ordinateur*, 2015. (Nieto *et al.*, 2015)
2. **G. Nieto**, F. Devernay et J. Crowley, "Rendu basé image avec contraintes sur les gradients", dans *Reconnaissance des Formes et Intelligence Artificielle*, 2016. (Nieto *et al.*, 2016b)
3. **G. Nieto**, F. Devernay et J. Crowley, "Variational Image-Based Rendering with Gradient Constraints", dans *International Conference on 3D Imaging*, 2016. (Nieto *et al.*, 2016a)
4. **G. Nieto**, F. Devernay et J. Crowley, "Rendu basé image avec contraintes sur les gradients", dans *Traitement du signal* (numéro spécial, à paraître), 2017. (Nieto *et al.*, 2017b)
5. **G. Nieto**, F. Devernay et J. Crowley, "Linearizing the Plenoptic Space", dans *LF4CV*, 2017. (Nieto *et al.*, 2017a)

Le code source de cette thèse est disponible à l'adresse `<https://github.com/Nighteye/light-field-remote-vision>`.

Remerciements

Un grand merci à Frédéric Devernay sans lequel cette thèse n'aurait été possible. Merci à James Crowley pour ses conseils pertinents et ses réunions enrichissantes.

Merci à Christine Guillemot et Ivo Ihrke d'avoir rapporté cette thèse, et merci à Patrick Perez d'avoir été examinateur. Vos critiques m'ont beaucoup apporté, et j'ai particulièrement apprécié notre échange durant la soutenance. Merci à Rémi d'avoir accepté sans hésitation de présider la soutenance de thèse.

Je remercie la DGA et l'INRIA d'avoir financé ces travaux de thèse.

Merci Sergi de m'avoir guidé et inspiré alors que je débutais en première année. Tes conseils ont porté leurs fruits.

Je remercie de tout coeur l'équipe IMAGINE, que j'ai eu la chance d'intégrer lors en cours de thèse. Je remercie particulièrement les collègues avec qui j'ai eu le plaisir de partager le bureau : Alexandre, Romain, Guillaume, Geoffrey, Even et Maxime. Ce fut un vrai plaisir de partager un repas, un café ou une cigarette avec vous, ou tout simplement profiter de votre bonne humeur quotidienne.

À Sandra et José pour le temps passé ensemble, leur amitié précieuse et leur aide dans les moments difficiles.

À mes compagnons d'escalade et confrères du CEA, pour les murs que l'on a monté et les bières que l'on a descendu !

À Louis-Clément, pour avoir été un colocataire exemplaire et un ami de confiance. Je te souhaite de réussir. Merci à Raphaël, Pedro et Giuseppe pour les bonnes soirées que l'on a passé ensemble.

Aux Deliriums, pour les vacances et les soirées parisiennes qui m'ont beaucoup aidé à décompresser.

À ma famille, mes amis, pour le soutient qu'ils m'ont apporté durant ces trois longues années.

Notation

Dans cette thèse nous utilisons la plupart des conventions usuelles en vision par ordinateur. En particulier nous notons les points, matrices et vecteurs en gras. Les matrices $\mathbf{M}$ sont toujours représentées avec une lettre majuscule, et les vecteurs $\mathbf{v}$ avec une lettre minuscule ; les scalaires s sont notés en italique. Par souci de lisibilité nous noterons souvent les vecteurs colonnes sous la forme d'un tuple $\mathbf{v} = (v_1, v_2)$ plutôt que $\mathbf{v} = [v_1, v_2]^\intercal$. De manière générale le produit en croix est noté $\mathbf{v} \times \mathbf{u}$ et le produit scalaire $\mathbf{v}^\intercal\mathbf{u}$, où $^\intercal$ désigne la transposée. † désigne le pseudo-inverse.

Voici une liste des principales notations que nous utilisons :

$\mathbb{R}$	ensemble des nombres réels
$\mathbf{X}$	point 3D
$\mathbf{r}$	rayon optique (vecteur 3D)
$\mathbf{n}$	normale à la surface
$\mathbf{I}$	intensité ou couleur (RGB)
z	profondeur
$\mathbf{x} = (x, y)$	point image 2D
$\bar{\mathbf{x}} = (x, y, 1)$	coordonnées étendues 3D
$\tilde{\mathbf{x}} = \tilde{w}(x, y, 1)$	coordonnées homogènes 3D
$\mathbf{P}$	matrice 3×4 de projection
$\mathbf{K}$	matrice 3×3 de paramètres intrinsèques
$\mathbf{R}$	matrice de rotation 3×3
$\mathbf{t}$	vecteur de translation 3D
$\mathbf{C}$	centre optique de la caméra (point 3D)
I	une image quelconque
M	un masque

(I_n)	pyramide gaussienne, ou pyramide d'images de I
∇I	gradient de I
ΔI	laplacien de I
(ΔI_n)	pyramide laplacienne de I
$f(\mathbf{x})$	fonction de $\mathbf{x}$
$\mathbf{J}_f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}}$	jacobienne (matrice) de f en $\mathbf{x}$
$\lvert\mathbf{J}_f(\mathbf{x})\rvert$	jacobien (déterminant de la jacobienne)
(u, v, s, t)	point 4D selon la paramétrisation *light slab*
$L(u, v, s, t)$	radiance au point 4D $(u, v, s, t) = (\mathbf{u}, \mathbf{s})$
f	distance focale en pixels
d	distance au plan focal (refocalisation)
b	fonction d'étalement du point (PSF)
Ω_k	domaine des images sources
Γ	domaine de l'image cible
v_k	image source
u	image cible
$\tau_k : \Omega_k \to \Gamma$	transformation avant, de l'image source à l'image cible
$\beta_k : \Gamma \to \Omega_k$	transformation inverse, de l'image cible à l'image source
$\mathbf{\Sigma_{xy}}$	matrice de covariance de la distribution $\mathbf{x}$ avec $\mathbf{y}$
$\mathbf{\Sigma_{x,y}}$	matrice de covariance de la distribution jointe $\mathbf{x}, \mathbf{y}$
σ_x^2	variance de x
$P(x\|y)$	probabilité de x sachant y
K	nombre de vues sources
W	largeur de l'image en pixels
H	hauteur de l'image en pixels

Lexique

Voici quelques termes anglais et leur équivalent en français tels qu'ils sont explicités dans cette thèse.

- *backward warping* : projection rétrograde
- *baseline* : entraxe
- *data term* : terme d'attache aux données
- *difference of Gaussian* : différence de gaussiennes
- *evidence* : évidence
- *forward warping* : projection directe
- *ghosting* : effet fantôme
- *image-based rendering* : rendu basé image
- *Laplacian blending* : mélange laplacien
- *light field* : champ de lumière/champ lumineux
- *likelihood* : vraisemblance
- *local coordinates* : coordonnées caméra locales
- *multi-view stereo* : stéréo multi-vues
- *overfitting* : sur-ajustement
- *pinhole* camera : sténopé
- *point spread function* : fonction d'étalement du point
- *posterior* : probabilité *a posteriori*
- *prior* : probabilité *a priori*
- *quad* : quadrilatère
- *refocussing* : refocalisation
- *scale space* : espace d'échelle
- *smoothness/regularization term* : term de lissage/régularisation
- *splat/footprint* : éclaboussure/empreinte
- *splatting* : rendu par éclaboussures
- *sweeping plane* : plan de balayage
- *underfitting* : sous-ajustement
- *warps* : fonction de déformation

– *world coordinates* : coordonnées monde

Table des matières

1 Introduction

1.1 Motivations

Durant ces dernières décennies les champs de lumière (*light fields*) ont attisé la curiosité des académiciens mais aussi du grand public, grâce à la commercialisation de caméras plénoptiques comme *Lytro* ou *Raytrix*. Le concept fut introduit à l'occasion de l'étude de la fonction plénoptique 5D, qui retourne la radiance le long de n'importe quel rayon lumineux passant par n'importe quel point 3D de la scène. Le champ de lumière décrit la fonction plénoptique réduite à quatre dimensions sous l'hypothèse que le rayon optique mesuré ne rencontre pas d'obstacle, et par conséquent que sa radiance est la même pour tous les points le constituant. Reconstruire le champ de lumière consiste à retrouver les parties manquantes étant donnés les échantillons mesurés. Dans cette thèse nous nous attelons au problème de la reconstruction du champ de lumière dans le but de synthétiser un nouvelle image, vue par une caméra virtuelle située plus proche que les caméras ou le dispositif capturant une scène lointaine. L'intérêt de la synthèse de point de vue est double : se rapprocher de la scène sans faire appel au zoom, qui génère des effets non désirables perturbant la vision 3D stéréoscopique, et de voir des objets cachés derrière des camouflages en reconstruisant une image ne comportant que les rayons issus de la scène occultée.

1.2 Approche

Capturer, modéliser et reconstruire le *light field* Tout comme une caméra traditionnelle échantillonne l'espace 2D des rayons provenant d'un seul point 3D, l'appareil qui mesure le champ de lumière (généralement une caméra plénoptique, ou un ensemble de caméras standards) échantillonne l'espace 4D des rayons optiques de l'espace. De tels dispositifs sont nombreux, de la simple caméra portative à des ensembles de dizaines caméras aux configurations diverses (*Lytro immerge*, *Stanford camera array*), en passant par les caméras professionnelles de cinéma (*Lytro Cinema*) ou les dispositifs panoramiques (*Facebook 360 Surround*). Aucun de ces dispositifs

n'est réellement adapté à la vision à longue distance. Par conséquent nous cherchons dans cette thèse à élaborer des algorithmes de synthèse d'image à partir d'un ensemble non structuré de caméras. Pour cela nous nous basons sur une approche de rendu basé image, où l'espace plénoptique (l'ensemble des rayons lumineux de la scène) est échantillonné par la capture de plusieurs photographies, modélisé, puis reconstruit pour recréer une nouvelle vue. Délivré des contraintes matérielles que nous imposent les dispositifs cités plus haut, on s'interroge sur de nouvelles configurations de caméras contraintes qui optimisent le résultat des algorithmes de rendu basé image (Nieto *et al.*, 2015).

Notre approche : le rendu basé image Les méthodes de rendu basé image sont légion ; elles ont été étudiées par Shum *et al.* (2008), qui proposent de les classer selon le type de modèle géométrique qu'elles utilisent. On appelle *continuum IBR* cette classification, décrivant des méthodes allant de celles qui utilisent une géométrie implicite (*light field* par exemple) aux méthodes qui utilisent une géométrie explicite (basées texture par exemple). On peut remarquer que quelle que soit la représentation adoptée, la quasi totalité de ces méthodes se basent sur un schéma commun : la construction d'un modèle de l'espace plénoptique (la géométrie des rayons et leur radiance) et le rendu à proprement parler. La première étape consiste souvent à modéliser la scène, de façon plus ou moins précise (*proxys géométriques*, cartes de profondeurs denses, nuages de points, modèles texturés), à partir de données (images), et d'hypothèses (scène rigide, statique et lambertienne). Le but d'un tel modèle est d'en extraire des fonctions de déformation qui mettent en correspondance les pixels des vues source avec ceux de la vue à synthétiser. La deuxième étape consiste à effectuer le rendu de la vue cible à partir des fonctions de déformation et des images sources. Le rendu peut être direct : les images sources sont projetées sur la vue cible, et mélangées de sorte que la couleur résultante est une combinaison linéaire des couleurs des images sources (les poids de mélange varient selon la contribution de chaque pixel de chaque vue). D'autres méthodes, dites variationnelles, proposent d'ajuster un modèle d'image aux données, ayant des connaissances *a priori* sur l'image solution, afin d'obtenir une image par minimisation d'une fonction de coût.

Modéliser l'espace plénoptique La plupart des méthodes de reconstruction de la géométrie de la scène (appelées *multi-view stereo*) dont nous nous sommes efforcés de dresser un bref aperçu, présentent les mêmes défauts. Premièrement, elles sont souvent limitées par la configuration des caméras et de la scène ; deuxièmement, leur but est souvent de représenter la géométrie de la scène de manière explicite, ce qui n'est pas être nécessaire lorsqu'on souhaite seulement synthétiser une nouvelle vue ; troisièmement, elles reposent sur l'hypothèse que les surfaces sont lambertiennes, ou bien essayent de séparer la composante diffuse de la composante spéculaire, ou encore traitent des surfaces réfractives, mais ces problèmes sont toujours traités séparément.

Nous avons conçu une méthode (Nieto *et al.*, 2017a) ne cherchant pas à modéliser un comportement particulier de la lumière, mais plutôt à approximer localement

l'espace plénoptique. On cherche alors à reconstruire des *points visuels*, qui ne sont pas forcément des points 3D, représentés par des paramètres géométriques et photométriques. Les modèles que nous concevons contiennent l'information sur la façon dont le point bouge et comment sa couleur change lorsque la caméra bouge. On modélise de fait des effets optiques complexes comme des réflexions, des réfractions, ou des variations d'indice de réfraction, de même que des surfaces non lambertiennes.

Méthodes de rendu directes L'approche la plus classique de rendu basé image est le mélange en intensité. On procède par projection directe (*forward warping*), c'est-à-dire la projection des points des images sources ou du *point visuel* vers l'image destination, et accumulation des contributions. Cela implique de régler des problèmes de ré-échantillonnage, et la création de *splats*, objets géométriques qui approximent localement la surface. Le processus de rendu de *splats* est appelé rendu par éclaboussures ou *splatting* ; il est bien connu dans la littérature de rendu basé point. Afin de produire une image sans discontinuité ou effet fantôme (*ghosting*) nous avons adapté l'idée de la décomposition en pyramides laplaciennes au rendu basé image. Nous proposons de décomposer chaque image source en pyramide, projeter chaque bande de fréquence dans la vue cible avec les conséquences en terme de changement d'échelle cela implique, puis mélanger les différentes contributions afin d'obtenir une pyramide laplacienne cible que nous pouvons effondrer pour créer la nouvelle image. Cela soulève évidemment des problèmes théoriques que nous détaillons.

Méthodes de rendu variationnelles L'idée clé pour un rendu en haute résolution est que la qualité de l'image solution dépend souvent de la contrainte de l'espace de recherche. Par conséquent, trouver la bonne régularisation ou *a priori* sur la solution est une question cruciale pour l'obtention d'images d'excellente qualité, comme peut l'attester la littérature sur la super-résolution par exemple. Les méthodes de rendu variationnelles offrent la possibilité d'intégrer un terme de régularisation pour contrôler la qualité de l'image résultante. Malheureusement cela n'est pas suffisant pour empêcher l'apparition d'artefacts visuels dans la solution, dus aux discontinuités dans les poids des images sources, à la géométrie de la scène ou au placement des caméras.

La méthode de rendu que nous proposons (Nieto *et al.*, 2016b,a, 2017b) ne repose pas sur de forts *a priori* sur la nouvelle image à synthétiser, mais tente plutôt de mieux exploiter les données procurées par les images sources pour ajouter de nouvelles contraintes sur la solution. Nous montrons qu'une façon d'éviter ces artefacts est d'imposer des contraintes supplémentaires sur le gradient de l'image synthétisée. Nous proposons une approche variationnelle suivant laquelle l'image cherchée est solution d'un système linéaire résolu de façon itérative. Nous testons la méthode sur plusieurs jeux de données multi-vues structurés et non-structurés, et nous montrons que non seulement elle est plus performante que les méthodes de l'état de l'art, mais elle élimine aussi les artefacts créés par les discontinuités de visibilité.

Application : vision d'objets occultés par des camouflages Il est possible de voir des objets occultés par un camouflage à l'aide d'une caméra classique en faisant la mise au point au-delà de l'obstacle (arbre, grillage, buisson). En effet si l'ouverture de l'objectif photographique est assez large, elle collecte une partie des rayons provenant de l'objet occulté ; la mise au point sur ce dernier aura pour effet de flouter l'avant-plan occultant et de restituer l'arrière-plan avec plus de netteté. L'idée clé est de concevoir une ouverture synthétique à l'aide d'un ensemble de caméras suffisamment large pour capter un grand nombre de rayons émanant de l'arrière-plan. Par une de nos méthodes de rendu basé image qui généralise la notion de refocalisation, on arrive à synthétiser une image ne contenant que l'arrière-plan. D'autre part, si nous éliminons les rayons optiques correspondant à l'obstacle, il est possible de reconstruire une image de l'arrière-plan à partir des pixels épars des images sources.

1.3 Contenu de la thèse

Chapitre 2 Dans le chapitre 2 nous introduisons certaines notions importantes de l'imagerie plénoptique comme le *champ lumineux*, la *fonction plénoptique* ou encore la paramétrisation *light slab*. En particulier nous définissons l'*espace plénoptique*, l'espace des rayons (orientations et radiances), qui est un concept phare de la modélisation des *points visuels* présentée au chapitre 6. Nous présentons le problème de reconstruction du champ de lumière comme celui qui consiste à retrouver les rayons manquants dans un espace plénoptique échantillonné de façon éparse. Nous montrons que ce problème peut-être traité sous une approche de rendu basé image : la synthèse d'une nouvelle image à partir de photographies prises à divers endroits de la scène. Puis nous détaillons brièvement le modèle de caméra utilisé qui est celui du sténopé, la chaîne logicielle, composée d'une phase d'estimation des fonctions de déformation et d'une phase de rendu, et enfin les critères et jeux de données utilisés pour évaluer nos résultats.

Chapitre 3 Ce chapitre apporte une réflexion sur les différentes façons d'obtenir les fonctions de déformation, première étape dans le rendu d'une nouvelle vue. L'approche la plus commune consiste à estimer un *proxy géométrique*. Nous présentons les principaux algorithmes de stéréo multi-vues (MVS) utilisé dans le rendu basé image, qui fournissent un nuage de point, un maillage, ou un ensemble de cartes de profondeur. Nous montrons comment calculer les fonctions de déformation à partir de ces *proxys*, mais aussi comment propager l'incertitude sur la géométrie estimée, très utile pour calculer les poids de rendu. Enfin nous discutons du choix de *proxy*, en comparant leurs avantages et les artefacts de rendu qu'ils génèrent.

Chapitre 4 Deux méthodes directes de rendu basé image sont présentées dans le chapitre 4. Dans un premier temps nous détaillons l'approche de rendu par éclaboussures (*splatting*) bien connue de la littérature en infographie. La distorsion des images sources (compressions et dilatations) causées par les fonctions de

déformation nous oblige à ré-échantillonner les textures sources projetées. Pour cela nous effectuons un rendu d'éléments de surface 3D, appelés *splats*, qui prennent en compte cette déformation. Le rendu des vues sources, pondéré par les contributions de chaque vue, est accumulé puis normalisé par la somme des contributions. L'image générée possède souvent des trous, que nous remplissons par un algorithme d'*inpainting*. Afin de répondre au problème des artefacts de rendu causés par un mélange des vues sources en intensité et des imperfections du *proxy géométrique*, nous présentons une nouvelle méthode directe de rendu multi-échelle. Elle s'appuie sur une décomposition des images sources en pyramides laplaciennes, qui sont ensuite projetées et combinées par niveau de résolution. Nous montrons quelles sont les limites mathématiques de la déformation de l'espace d'échelle, et comment outrepasser ces difficultés.

Chapitre 5 Dans ce chapitre nous cherchons à résoudre les problèmes de bruit haute fréquence qui apparaît dans l'image générée par une méthode variationnelle. Dans un premier temps nous présentons le modèle de formation de l'image, puis l'approche bayésienne du rendu, consistant à trouver l'image solution qui minimise une énergie basée sur la probabilité *a posteriori* d'avoir la solution étant données les images sources (nos données). Le terme d'attache aux données de cette énergie n'imposant que des contraintes sur l'intensité de la solution, le rendu est souvent bruité et manque de continuité. Afin de mieux contraindre la solution, nous avons modifié le terme d'attache aux données en forçant la solution à être proche des images sources dans le domaine du gradient. Une série d'expériences sur des jeux de données structurés et non-structurés nous a permis de montrer la performance de notre énergie en terme de correction d'artefacts.

Chapitre 6 Certaines scènes sont difficiles à reconstruire à cause de la présence de réfractions, transparences ou spécularités. Le *proxy géométrique* obtenu est très incertain, ce qui affecte grandement le rendu d'une nouvelle vue. Même avec un *proxy* précis, les rayons dévient de l'hypothèse lambertienne, soit parce que les points reconstruits appartiennent à une surface non diffuse ou isotrope, soit parce les rayons optiques qui visualisent un point ne suivent pas les règles de la géométrie épipolaire. Dans le chapitre 6 nous présentons une nouvelle technique de d'échantillonnage et de paramétrisation de l'espace plénoptique, ainsi qu'un procédé d'optimisation pour ajuster des modèles à l'échantillonnage de l'espace plénoptique. Nous ajustons aux données de l'espace plénoptique des modèles plus complexes de *points visuels* qui permettent une synthèse de vue plus précise de scènes difficiles à reconstruire qui contiennent des spécularités, réfractions et transparences.

Chapitre 7 Enfin le chapitre 7 ouvre sur des applications possibles de nos algorithmes de rendu basé image et de reconstruction du champ de lumière à la vision à longue distance. Dans un premier temps nous abordons la question du placement de caméras contraintes qui optimise le rendu d'une nouvelle vue, en particulier une vue cible plus proche du sujet que les vues sources. Nous établissons une fonction de coût qui prend en compte à la fois l'incertitude sur le *proxy géométrique* estimé et

l'incertitude sur la destination des pixels projetés sur la vue cible. Nous montrons sur un exemple simple que cette fonction possède un minimum global, et que sa minimisation conduit à un placement qui fait un compromis entre un *proxy géométrique* précis et un rendu précis. Dans un deuxième temps nous montrons l'intérêt de la synthèse de point de vue plus proche du sujet, plutôt que la synthèse d'une longue focale, et nous montrons les limites d'une telle approche. Enfin nous abordons la question nouvelle du rendu d'objets camouflés, étant donné un *proxy géométrique* que nous ne cherchons pas à estimer. Nous montrons par des expériences que la suppression du camouflage dans les images sources par un algorithme de *matting* combinée à l'usage de notre algorithme de rendu multi-échelle améliore grandement la qualité de l'image rendue, même dans des cas d'occultation extrême (plus de la moitié des images sources).

2

Le rendu basé image comme solution au problème

2.1 Le champ lumineux

Définitions Le terme de *champ lumineux* fut introduit pour la première fois par Gershun (1939) pour étudier les propriétés radiométriques de la lumière. La théorie du champ lumineux défend l'hypothèse selon laquelle chaque point de l'espace est traversé par une infinité de rayons lumineux allant dans toutes les directions possibles (figure 2.1). Il est intrinsèquement lié à la *fonction plénoptique* de Adelson et Bergen (1991) qui pour chaque point 3D d'une scène, renvoie la valeur de la radiance le long d'un rayon lumineux donné. Cette fonction a initialement 7 paramètres : 3 pour la position du point dans l'espace, 2 pour la direction du rayon lumineux le traversant, ainsi que la longueur d'onde et le temps. Il est courant dans la littérature de ne trouver que les 5 premiers paramètres précédemment cités s'il l'on suppose que la fonction plénoptique est indépendante de la longueur d'onde et que la scène est statique. Sous l'hypothèse que la rariance est constante le long d'un rayon optique ne rencontrant pas d'obstacle, alors on peut réduire la fonction plénoptique à 4 paramètres (Levoy et Hanrahan, 1996). On appelle communément *champ lumineux* (*light field* en anglais) cette fonction plénoptique réduite 4D. La radiance est la même en tout point du rayon optique, qui est alors entièrement décrit par ces paramètres. L'espace des paramètres qui caractérisent les rayons optique est appelé l'*espace plénoptique*. Les 3 (ou 2 pour la fonction réduite 4D) paramètres *spatiaux* caractérisent le *centre de projection* ou *centre optique*, pupille de l'observateur au travers de laquelle une infinité de rayons se concentre. L'ensemble de ces rayons, ou plutôt leur direction, est définie par les deux autres paramètres, que l'on qualifie d'*angulaires*.

Dispositifs de capture Un sténopé (*pinhole camera* en anglais) capture l'infinité de rayons qui passent par son centre de projection. Selon la théorie du champ lumineux, chaque point est un sténopé qui collecte une fraction de la lumière émanant

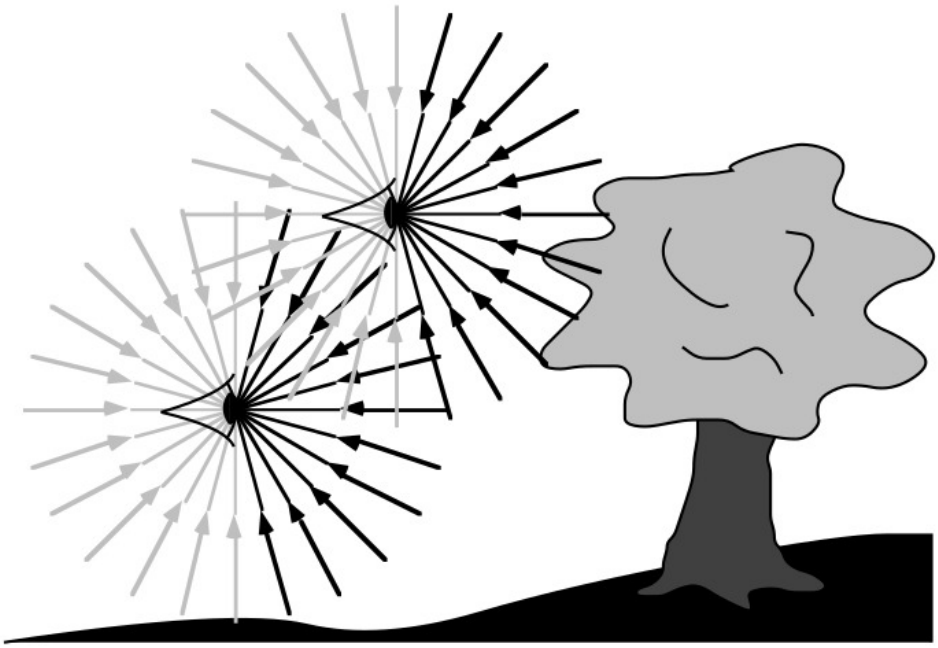


Fig. 2.1 – *Figure de Adelson et Bergen (1991). Ici deux observateurs ponctuels ont été représentés. En chacune de ces deux pupilles, assimilées à des points 3D, une infinité de rayons passent dans toutes les directions.*

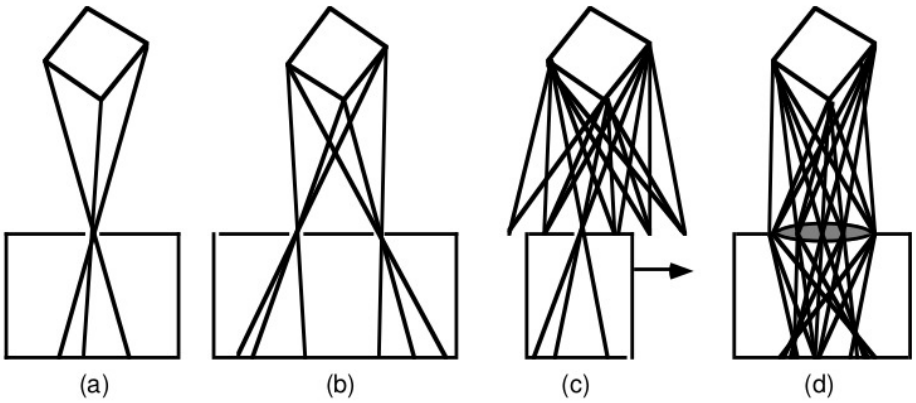
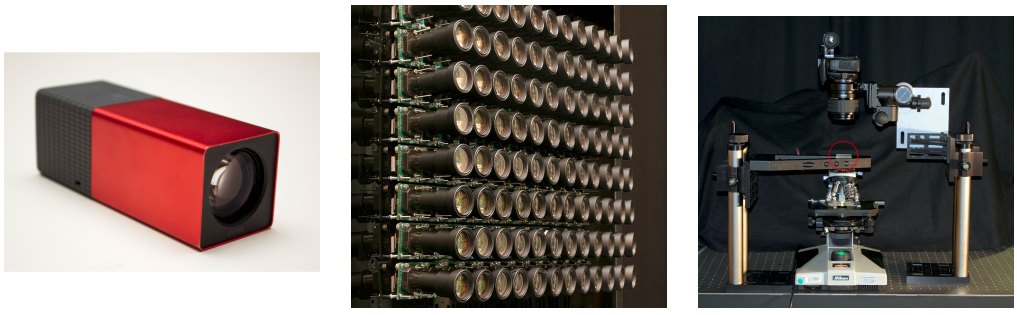


Fig. 2.2 – *Figure de Adelson et Wang (1992). (a) Un sténopé collectant l'ensemble des rayons passant par son centre de projection. (b) Une caméra stéréoscopique, constituée de deux centres de projection, collecte un plus grand nombre de rayons. (c) Un dispositif multi-stéréoscopique, ici utilisant la parallaxe de mouvement pour capturer une multitude de rayons passant par différents centres de projection. (d) Une caméra conventionnelle pouvant être apparentée à une infinité de sténopés dont les centres de projection infinitésimaux composent la lentille principale.*

d'une scène. Il restitue une image 2D, c'est à dire un échantillonnage de cette collection de rayons (figure 2.2a). Il n'y a pas de dimension angulaire : un même point 3D n'est échantillonné qu'une seule fois, via le rayon qui le lie au centre de projection du sténopé. Une caméra stéréoscopique ou multi-stéréoscopique (figure 2.2b-c) collecte bien plus de rayons de l'espace plénoptique, car plusieurs centres de projections sont utilisés. On a ajouté une dimension angulaire : deux rayons issus d'un même point sont capturés par le dispositif. Même avec un dispositif multi-stéréoscopique, on échantillonne plus de rayons issus d'un même point 3D, mais le domaine angulaire capturé (ensemble des rayons issus d'un même point 3D) est toujours discret. On peut densifier l'échantillonnage angulaire par un dispositif mobile exploitant la parallaxe de mouvement (figure 2.2c). Une caméra conventionnelle (figure 2.2d) possède une lentille qui se décompose en une infinité de centre de projection, collecte un ensemble 4D de rayons. Cependant le processus d'intégration de la lumière sur le plan capteur réduit la dimension de l'espace échantillonné. En d'autres termes le capteur de la caméra intègre spatialement un infinité 5D de rayons optiques, pour tous les centres optiques infinitésimaux constituant la lentille, ce qui enlève la dimension angulaire. Chaque point 3D n'est observé d'une seule fois, échantillonné par l'intégration des rayons lumineux émanants captés par la lentille : il est vu flou à un niveau qui dépend de la taille du diaphragme (domaine d'intégration des rayons).

Un dispositif de capture du champ lumineux au contraire permet de restituer un échantillonnage 5D ou 4D de la fonction plénoptique. S'il en existe de toutes les sortes, ils ont en commun qu'ils tentent de contourner la perte d'information due à l'intégration du capteur optique. Les ensembles de caméra comme celui de Wilburn *et al.* (2005) (figure 2.3b) reposent sur le même principe qu'une caméra multi-stéréoscopique : capturer la scène depuis des points de vue qui diffèrent légèrement. Chaque caméra peut-être modélisée par un sténopé qui échantillonne l'ensemble des rayons passant par son centre optique. On obtient non pas une image 2D mais un ensemble d'images 2D, soit un échantillonnage 4D du champ lumineux. Le *Lego Mindstorms gantry* de l'université de Stanford repose sur le principe de la parallaxe de mouvement : si la scène est statique, on peut recréer un ensemble de caméras en prenant une photographie avec une caméra classique se déplaçant à



(a) (b) (c)

Fig. 2.3 – *Dispositifs de capture plénoptiques. (a)* Lytro *(b) Ensemble de caméras de Wilburn et al. (2005) (c) Microscope plénoptique de Levoy (2006).*

intervalle régulier. Les *caméras plénoptiques* échantillonnent le champ lumineux par un réseau de micro-lentilles posé derrière l'objectif principal (Adelson et Wang, 1992; Venkataraman *et al.*, 2013). Très populaires, certaines de ces caméras comme *Lytro* (figure 2.3a) ou *Raytrix* ont connu un succès commercial. Le microscope plénoptique (Levoy, 2006) (figure 2.3c) est également basé sur le concept du réseau de micro-lentilles. D'autres dispositifs utlisent des ensembles de caméras ou de micro-lentilles couplés à des boîtes à lumière (Wetzstein *et al.*, 2011) ou des arrières-plan texturés pour coder les différentes dimensions de l'espace plénoptique. Parmi tous ces dispositifs de capture optique, aucun n'est réellement adapté à la vision à longue distance du fait de leur faible parallaxe qui nuit à l'échantillonnage du domaine angulaire des rayons : il n'y a pas de « télescope plénoptique ».

Paramétrisation *light slab* La paramétrisation de l'espace plénoptique la plus populaire est sans doute le *light slab* (Levoy et Hanrahan, 1996), illustrée sur la figure 2.4. La radiance $L(u, v, s, t) = L(\mathbf{u}, \mathbf{s})$ d'un rayon est paramétrée par l'intersection de ce rayon avec deux plans (parallèles dans selon notre paramétrisation). On obtient alors deux couples de coordonnées, soit 4 paramètres. Notez que l'hypothèse selon laquelle la radiance est constante le long du rayon doit être vérifiée. Dans la littérature *light field*, on assimile souvent le plan (u, v) au domaine spatial (plan image d'une caméra), et le plan (s, t) au domaine angulaire (plan des micro-lentilles pour des caméras plénoptiques, ou plan des centres optiques des caméras pour des grilles de caméras). Ici nous ne faisons pas de distinction entre ces deux plans, car dans notre paramétrisation des rayons optiques, ils ne correspondent pas forcément au plans image ou au plan contenant les centres optiques de nos ensembles de caméras. Notons qu'il existe d'autres façons de paramétrer les rayons, comme avec des angles, ce qui est plus adapté aux dispositifs de capture sphériques.

La représentation *light slab* permet de décrire facilement l'observation des rayons lumineux issus d'un même point 3D $\mathbf{X}$ vus par un ensemble de caméras, comme l'illustre la figure 2.5. Le dispositif illustré est un ensemble de caméras pas nécessairement structuré ; c'est ce modèle de dispositif de capture qui sera utilisé dans la suite. Un point 3D $\mathbf{X}$ capturé par ce dispositif est visible dans chacune de ces

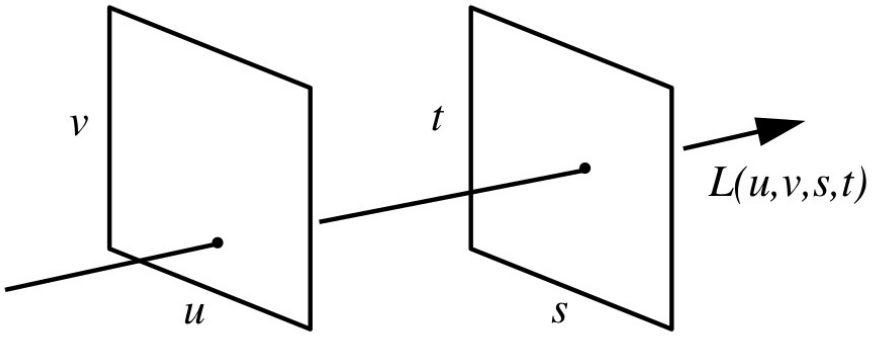


Fig. 2.4 – *Paramétrisation* light slab *de Levoy et Hanrahan (1996).*

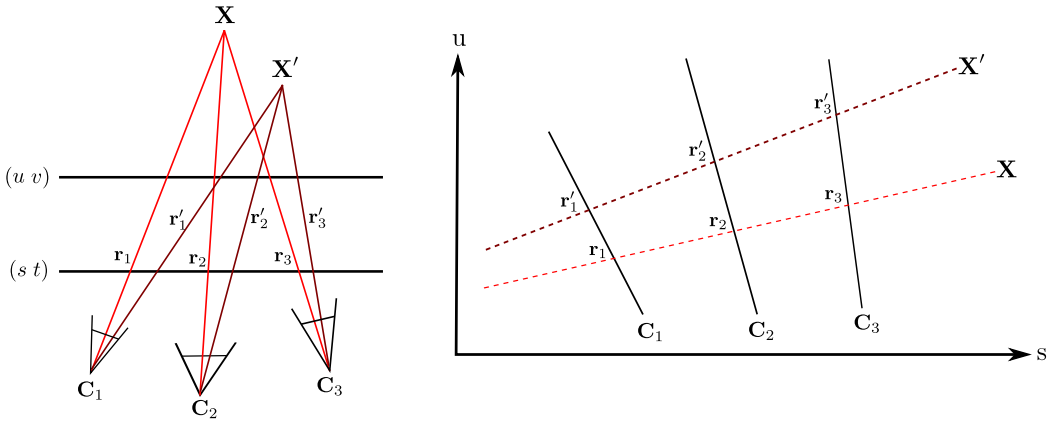


Fig. 2.5 – *La paramétrisation* light slab *de rayons issus de deux points 3D* $\mathbf{X}$ *et* $\mathbf{X}'$. *Les rayons* $\mathbf{r}_1$, $\mathbf{r}_2$ *et* $\mathbf{r}_3$ *sont issus de* $\mathbf{X}$ *et passent par les centres optiques respectifs* $\mathbf{C}_1$, $\mathbf{C}_2$ *et* $\mathbf{C}_3$. *Idem pour* $\mathbf{r}'_1$, $\mathbf{r}'_2$ *et* $\mathbf{r}'_3$ *issus du point* $\mathbf{X}'$. *Chaque rayon est paramétré par deux couples de coordonnées* $\mathbf{u} = (u, v)$, *intersection avec le premier plan, et* $\mathbf{s} = (s, t)$, *intersection avec le deuxième plan. Ces points 4D sont représentés en 2D sur la figure de droite. Une droite noire représente l'ensemble des rayons passant par le centre optique d'une caméra. Une droite en pointillés représente l'ensemble des rayons issus d'un même point 3D. Les rayons échantillonnées sont les intersections entre ces droites.*

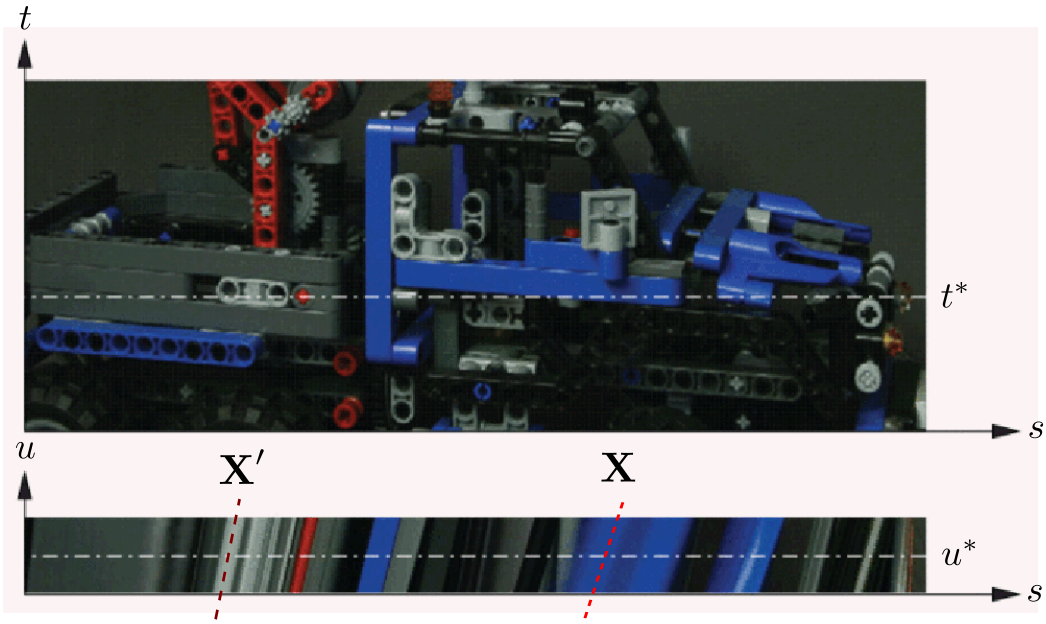


Fig. 2.6 – *Si le champ lumineux est suffisamment dense, c'est-à-dire si le nombre de points de vue (et donc de rayons) capturés est suffisamment important, et si les points de vue sont rectifiés (les plans images sont identiques au plan* (s, t)*), alors la représentation* light slab *nous donne une EPI. Sous réserve de la validité de l'hypothèse lambertienne, les droites qui modélisent les points 3D sont clairement visibles, et leur pente renseigne sur la distance du point 3D. On a par exemple représenté en pointillés les droites (l'ensemble des rayons issus) des points* $\mathbf{X}$ *et* $\mathbf{X}'$.

caméras (nous laissons en suspens la question des auto-occultations pour le moment) représentées par leur centre optique $\mathbf{C}$. Chaque rayon $\mathbf{r}_k$ du faisceau issu de $\mathbf{X}$ relie $\mathbf{X}$ à $\mathbf{C}_k$. Sa paramétrisation *light slab* est un point 4D (u, v, s, t). Dans cet espace plénoptique, l'ensemble des rayons issus d'un même point est un plan, représenté par une droite en pointillés sur la figure 2.5 de droite. Chaque observation par une caméra $\mathbf{C}_k$, c'est-à-dire chaque rayon issu du faisceau qui passe par le centre optique $\mathbf{C}_k$ est un échantillon de cette droite. Si l'échantillonnage (c'est-à-dire le nombre de caméras $\mathbf{C}_k$ qui observent le point $\mathbf{X}$) est suffisamment dense, alors on peut représenter ces plans par des images appelées EPI (*Epipolar Plane Image*) qui sont des coupes 2D de l'espace plénoptique 4D (figure 2.6). Cela suppose aussi que les points observés sont lambertiens et que les images sont rectifiées, c'est-à-dire que les plans image sont identiques.

Les EPI permettent de visualiser facilement l'espace plénoptique 4D : un rayon correspond à un échantillon 2D sur l'EPI (un pixel). L'EPI permet également d'estimer la profondeur de la scène. Ainsi sous l'hypothèse que les points observés sont lambertiens et qu'il respectent les règles de parallaxe, les EPIs représentent un ensemble de droites monochromes de pentes variées. Ces pentes sont proportionnelles à la disparité, ou inversement proportionnelles à la profondeur, et dénotent de la parallaxe, horizontale (EPI (u, s)) ou verticale (EPI (v, t)). Les algorithmes de reconstruction 3D basés sur les champs lumineux estiment la pente des droites par le calcul d'un tenseur de structure (Wanner et Goldluecke, 2012a). L'échantillonnage du champ lumineux doit être suffisamment dense et peu bruité pour pouvoir estimer la pente des droites avec précision.

Hypothèses Un point est lambertien si sa radiance ne dépend pas de la direction d'observation. Autrement dit quel que soit le point de vue, le point 3D observé a la même couleur. Selon cette assertion tous les échantillons de rayons issus du même point 3D observé ont la même radiance, et donc la droite qui le représente dans l'EPI est monochrome (figure 2.6). Si cette hypothèse n'est pas valide, la recherche de la pente de la droite est d'autant plus difficile que la radiance varie le long de celle-ci. La deuxième hypothèse nécessaire à ce type de reconstruction 3D consiste à supposer que les rayons suivent les règles de la parallaxe : le point image, projection du point 3D dans une vue se déplace vers la gauche lorsque cette vue se déplace vers la droite, et vice versa. Il s'agit de la parallaxe horizontale ; de la même manière on définit la parallaxe verticale. Cela se manifeste par le fait que les échantillons d'un même faisceau de rayons suivent un plan en 4D, et donc une droite dans l'EPI. Si cette hypothèse n'est pas vérifiée alors ces échantillons ne suivent pas une droite et la recherche d'une profondeur associée à sa pente n'a pas de sens. Ces hypothèses seront valides dans la majeure partie de ce manuscrit ; nous verrons comment traiter les cas où elles ne le sont pas dans le chapitre 6.

Domaine spatial versus domaine angulaire Dans la majeure partie de la littérature sur les champs lumineux, il est question d'un *domaine spatial* qui est l'espace des images, s'opposant au *domaine angulaire* qui est l'espace des directions des rayons (comme l'illustre la figure 2.7). Cela provient des dispositifs de captures,

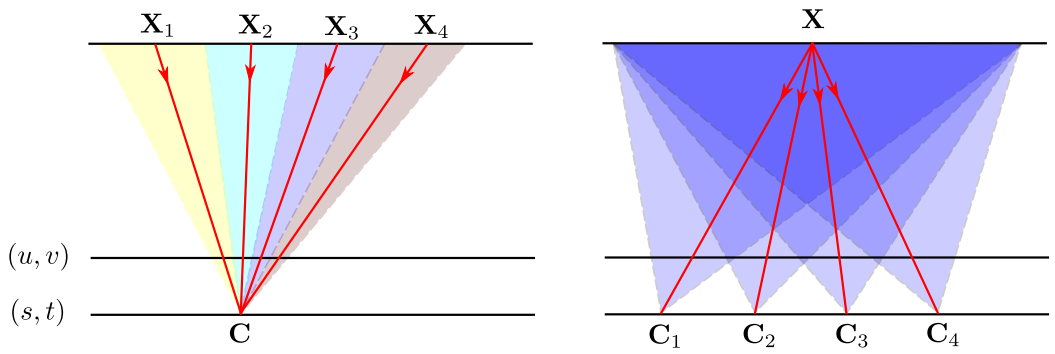


Fig. 2.7 – *Domaine spatial vs. domaine angulaire. Le plan image (u, v) échantillonne le domaine spatial tandis que le plan des centres optiques (s, t) échantillonne le domaine angulaire. Une surface plane est observée par deux dispositifs de capture. À gauche un sténopé de centre de projection $\mathbf{C}$ modélise une caméra classique dont la résolution des capteurs permet d'échantillonner la surface observée en quatre rayons (en rouge) respectivement issus de $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$ et $\mathbf{X}_4$. Plus précisément chacun des quatre capteurs intègre les rayons de la scène dans un cône (en couleur) centré sur un des points 3D de la surface. Mais la caméra ne capture qu'un seul des rayons issus de chaque point 3D (parmi une infinité) : la résolution spatiale est plus élevée que la résolution angulaire. Au contraire le dispositif de droite ne capture qu'un seul point $\mathbf{X}$, mais plusieurs directions de rayons issus de ce point 3D via les multiples centres optiques $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ et $\mathbf{C}_4$. Chaque sténopé possède un capteur unique qui intègre l'ensemble des rayons émis par la surface (cône bleu) : la résolution spatiale est plus faible que la résolution angulaire.*

qui contrairement à une caméra conventionnelle pouvant se modéliser grossièrement par un sténopé, capturent non pas une image 2D mais une matrices d'images 2D, soit deux dimensions supplémentaires qui constituent le domaine angulaire. Dans les configurations particulières de type caméra plénoptique ou ensemble structuré de caméras, il est commun de se représenter un ensemble de sténopés ayant leur centre de projection dans le même plan (s, t) et le plan image identique au plan (u, v). Ainsi tous les sténopés modélisant le dispositif de capture ont le même plan image : il sont rectifiés ; les coordonnées (u, v) deviennent alors les coordonnées images ou pixel, ou encore coordonnées *spatiales*. Les coordonnées (s, t) permettent d'indexer chaque point de vue dans la matrice d'images 2D. Néanmoins dans la suite de ces travaux nous ne faisons pas la différence entre coordonnées *angulaires* et *spatiales*, qui font partie d'un même espace 4D sans distinction interne. En effet nous avons pour objectif de traiter des ensembles non structurés de caméras ; dans ce but les plans (u, v) et (s, t) sont totalement génériques et leur choix, arbitraire, n'entrave pas à la généralité de l'approche. (u, v) ne représente pas plus un point image que (s, t) ne représente une direction. Pour définir les modèles photométriques et géométriques, nous avons fait le choix de faire varier respectivement la couleur $\mathbf{I}$ et les coordonnées (u, v) en fonction de (s, t) mais ce choix n'a pas d'incidence.

Reconstruire le champ lumineux Le champ lumineux capturé n'est qu'un échantillonnage de la fonction plénoptique à 4 dimensions. La reconstruction du champ lumineux se définit comme le fait de retrouver les rayons manquants. Faisant la distinction entre le domaine angulaire et spatial, la littérature qui traite la reconstruction du champ lumineux s'attèle soit à la densification du domaine spatial dans

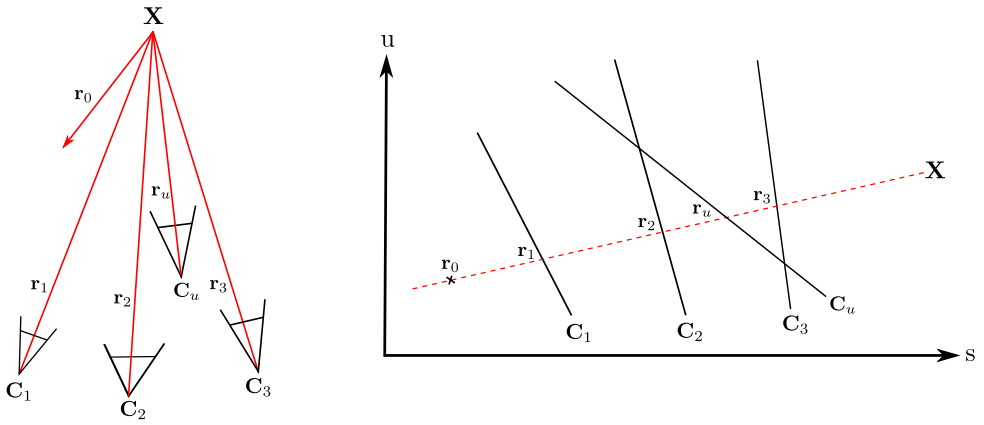


Fig. 2.8 – *La synthèse d'un point de vue* $\mathbf{C}_u$ *consiste à trouver le rayon* $\mathbf{r}_u$ *issu de chaque point* $\mathbf{X}$ *de la scène, à partir des rayons échantillonnés* $\mathbf{r}_1$, $\mathbf{r}_2$ *et* $\mathbf{r}_3$, *c'est-à-dire capturés par les vues sources* $\mathbf{C}_1$, $\mathbf{C}_2$ *et* $\mathbf{C}_3$ *respectivement. Dans l'*espace géométrique *4D, un modèle géométrique (droite rouge en pointillés) décrit l'ensemble des rayons issus de* $\mathbf{X}$, *dont par exemple le rayon* $\mathbf{r}_0$ *qui n'est capturé par aucune vue source. Dans cette représentation le rayon* $\mathbf{r}_u$ *est l'intersection de l'ensemble des rayons qui passent par* $\mathbf{C}_u$ *avec le modèle.*

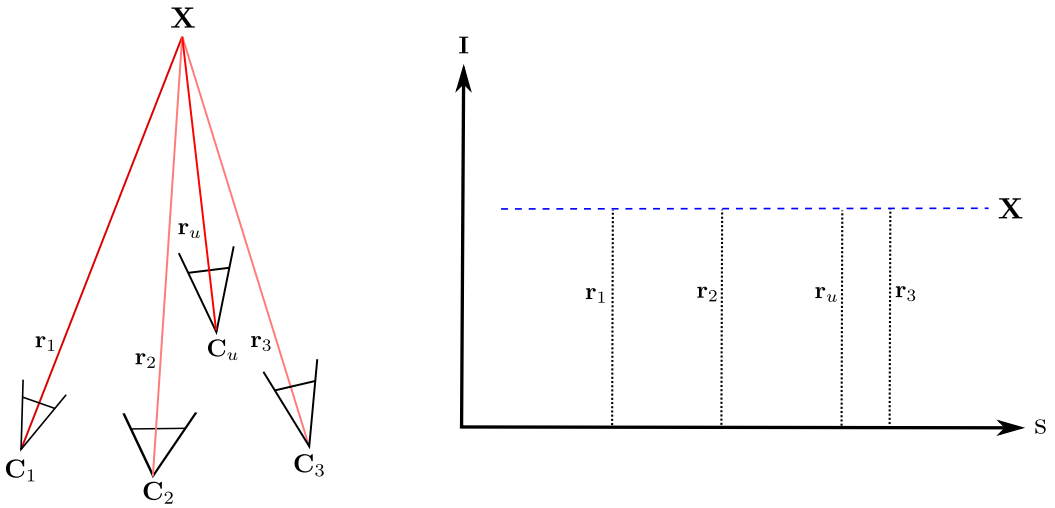


Fig. 2.9 – *Illustration du modèle photométrique. À gauche nous avons représenté des rayons issus du même point* $\mathbf{X}$ *qui n'ont pas la même couleur mesurée. Reconstruire l'*espace photométrique, *c'est trouver la « bonne couleur » du rayon cherché* $\mathbf{r}_u$ *à partir des couleurs mesurées. À droite : le modèle photométrique le plus simple, représenté par la droite bleue en pointillés, est le modèle constant ou lambertien. Tous les rayons capturés, y compris le rayon cherché* $\mathbf{r}_u$ *ont la même couleur* $\mathbf{I}$. *Des modèles plus complexes seront évoqués au chapitre 6.*

le cadre de la super-résolution, soit de la densification du domaine angulaire dans le cadre de la synthèse de point de vue, soit traite des deux problèmes joints (Wanner et Goldluecke, 2012b). Ne faisant pas de distinction entre les deux domaines, nous définissons la reconstruction du champ lumineux comme la densification de l'espace géométrique des rayons à quatre dimensions, à partir des données c'est-à-dire de l'ensemble des rayons échantillonnés. Il s'agit en d'autres termes d'interpoler ou d'extrapoler de nouveaux rayons à partir des rayons capturés. Comme l'illustre la figure 2.8, le problème de reconstruction du champ lumineux est intrinsèquement lié à celui de la synthèse d'un nouveau point de vue $\mathbf{C}_u$. Il faut retrouver les rayons lumineux issus de tous les points $\mathbf{X}$ de la scène à partir des rayons issus de ces mêmes points capturés par le dispositif plénoptique, ici modélisé par un ensemble (pas nécessairement structuré) de sténopés $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, etc.

Outre la densification de l'espace géométrique 4D des rayons optiques, un autre aspect important de la reconstruction du champ lumineux est la densification de l'espace photométrique. L'*espace photométrique* est l'espace des radiances des rayons ; la fonction plénoptique retourne une valeur de radiance, mais pour nous il s'agit d'un paramètre au même titre que la géométrie du rayon. Nous le représentons en 3 dimensions, une pour chaque canal de couleur RGB du rayon. L'*espace géométrique* (usuellement champ lumineux) 4D et l'*espace photométrique* 3D constituent ce que nous appelons l'*espace plénoptique*, qui a 7 dimensions. L'*espace photométrique* est souvent omis car sous l'hypothèse que le point observé $\mathbf{X}$ est lambertien, tous les rayons qui en sont issus ont la même radiance, d'où un modèle photométrique pris constant pour chaque point de la scène. Comme nous l'expliquons dans le chapitre 6, cette hypothèse est souvent remise en cause et la reconstruction du champ lumineux doit aussi prendre en compte la recherche de la « bonne couleur » des nouveaux rayons interpolés. Dans un cas plus pratique (voir figure 2.9 de droite), on peut mesurer des couleurs différentes pour chaque rayon issu d'un même point $\mathbf{X}$ de la scène ; retrouver la couleur du rayon cherché $\mathbf{r}_u$ est alors non trivial. Jusqu'au chapitre 6, nous utiliserons le modèle lambertien constant de la figure 2.9 (gauche) pour décrire nos scènes.

2.2 Introduction au rendu basé image

Présentation L'approche proposée pour synthétiser une nouvelle vue virtuellement plus proche de la scène est le rendu basé image (IBR, pour *Image-Based Rendering* en anglais). Le rendu basé image a ceci de commun avec le rendu classique en infographie qu'il consiste à calculer une image 2D. Il s'en distingue par le fait que la scène dont on souhaite avoir une image est réelle, et que sa représentation virtuelle (géométrie, illumination) se base sur des données images réelles. Notre dispositif de capture plénoptique est un ensemble générique de caméras modélisées par des sténopés (voir section 2.4). Chaque sténopé capture une partie du champ lumineux et en restitue une image 2D. Cette image source est un échantillonnage 2D de l'ensemble des rayons passant par le centre optique de la caméra source (voir figure 2.10 de gauche). Le rendu basé image consiste à générer une nouvelle image 2D (échantillonnage de l'ensemble des rayons passant par le centre optique cible) à

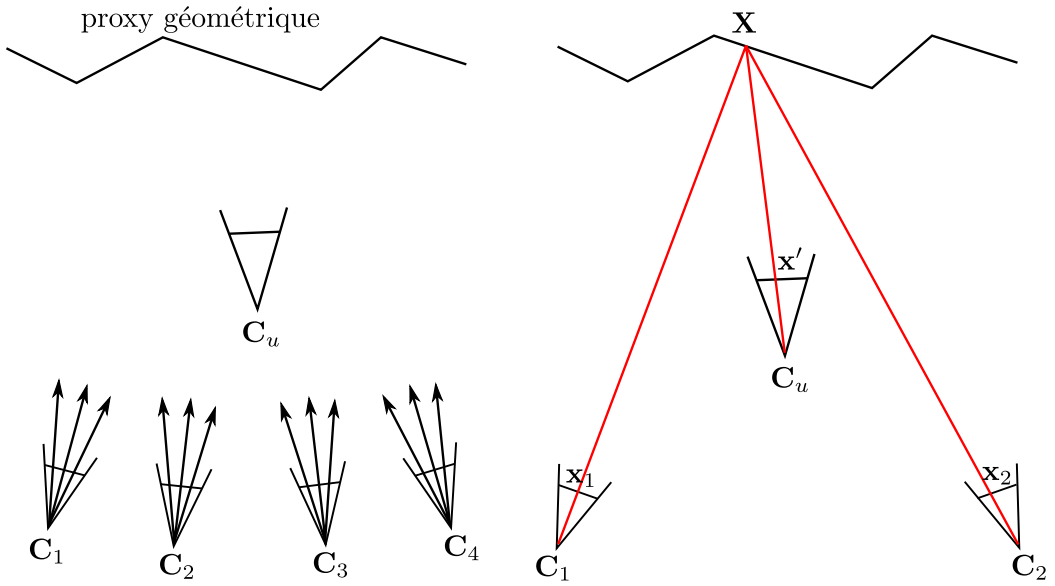


Fig. 2.10 – *La synthèse de point de vue : un problème de rendu basé image. À gauche : nous représentons le dispositif de capture comme un ensemble de vues sources* $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ *et* $\mathbf{C}_4$ *qui échantillonnent le champ lumineux en capturant les rayons représentés par des flèches. L'échantillonnage résultant est un ensemble d'images 2D qui nous permettent d'estimer un modèle de la scène : le* proxy géométrique. *À droite : connaissant les paramètres des caméras* $\mathbf{C}_1$ *et,* $\mathbf{C}_2$, *on trouve les rayons correspondant aux observations 2D* $\mathbf{x}_1$ *et* $\mathbf{x}_2$ *d'un point* $\mathbf{X}$ *de la scène. Le point 3D* $\mathbf{X}$ *est trouvé par intersection avec le* proxy, *ce qui nous permet d'obtenir le point d'observation 2D* $\mathbf{x}'$ *dans la vue cible* $\mathbf{C}_u$. *Les deux vues contribuent à la synthèse du même point, mais leur contribution peut varier selon le poids qu'on leur attribue.*

partir des images sources.

Le rendu basé image est plus spécifique encore que la reconstruction du champ lumineux car il pré-suppose une estimation d'un modèle géométrique de la scène, souvent approximatif, appelé *proxy géométrique*. Ce *proxy géométrique* peut être un nuage de points, un maillage ou un ensemble de cartes de profondeur par exemple, estimé à partir des données sources (les images prises de la scène) à l'aide d'un logiciel de stéréo multi-vues (MVS, pour *Multi-View Stereo* en anglais). Le rendu basé image suppose aussi la calibration des caméras, c'est-à-dire l'estimation d'un modèle de caméra, le plus souvent par le biais d'un logiciel de SfM (*Structure from Motion* en anglais). Ainsi, connaissant les paramètres des caméras sources (extrinsèques et intrinsèques), on est capable de faire correspondre chaque point image 2D des vues sources à la vue cible par un tracé de rayon ou un rendu de maillage texturé classique (figure 2.10 de droite).

L'utilisation de plusieurs vues sources est primordiale dans le rendu basé image, pour plusieurs raisons. Premièrement l'estimation du *proxy géométrique* nécessite déjà plusieurs images de la même scène. Plus le nombre de points de vue est grand, plus l'estimation de la géométrie sera précise ; cela a un impact important sur la qualité du rendu. Deuxièmement la scène comporte souvent des auto-occultations : des parties de la scène ne sont pas visibles car des objets au premier plan perturbent la visibilité. Il est donc nécessaire d'avoir plusieurs points de vue qui couvrent l'ensem-

ble de la scène, ou du moins la partie observée par le point de vue cible. Enfin, comme le montre la figure 2.10 de droite, deux vues peuvent apporter de l'information sur le même rayon cherché. Il s'agit alors de pondérer les vues par leur contribution respective, par des poids dont le calcul se base sur des heuristiques ou provient d'une formalisation mathématique plus poussée du rendu.

Le continuum IBR Les techniques de rendu basé image ont été en grande partie décrites et classifiées par Shum *et al.* (2008). La plupart des méthodes de l'état de l'art (Kopf *et al.*, 2013; Sinha *et al.*, 2012; Lipski *et al.*, 2014, 2010; Chaurasia *et al.*, 2013) utilisent cette reconstruction de la géométrie intermédiaire, le *proxy géométrique*. Shum *et al.* (2008) proposent une classification, le *continuum IBR*, qui repose sur la précision du *proxy géométrique* utilisé (figure 2.11). On remarque que moins une méthode utilise de géométrie (géométrie grossière ou implicite), plus elle requiert d'images sources. Les méthodes de rendu basées sur le champ lumineux se situent à gauche du spectre, c'est-à-dire qu'elle demandent beaucoup d'images, capturées par exemple par des ensembles de 17×17 caméras ou des caméras plénoptiques. Au contraire, les méthodes basées sur du rendu de texture *view-dependent* utilisent un *proxy* très précis, avec relativement peu d'images sources. Les méthodes intermédiaires basées sur le rendu de maillage (Buehler *et al.*, 2001; Davis *et al.*, 2012) se prêtent particulièrement bien aux applications de *free viewpoint*, où la caméra virtuelle est libre de se déplacer dans la scène. Notre approche est *view-dependent* ; en ce sens les points de vue qui contribuent réellement à la synthèse sont rares et le *proxy* aussi précis que possible pour limiter les artefacts de rendu. Le chapitre 3 détaille les motivations de notre choix de *proxy*.

Notations élémentaires Nous définissons l'image cible u et les K images sources $v_k,\ k \in [1 \dots K]$, respectivement comme les applications

$$\begin{array}{rccc} u: & \Gamma & \to & \mathbb{R} \\ & \mathbf{x}' & \mapsto & u(\mathbf{x}'), \end{array} \tag{2.1}$$

$$\begin{array}{rccc} v_k: & \Omega_k & \to & \mathbb{R} \\ & \mathbf{x} & \mapsto & v_k(\mathbf{x}), \end{array} \tag{2.2}$$

Définis tels quels, il s'agit de signaux génériques. Dans le cas d'images, Ω_k et $\Gamma \subset \mathbb{R}^2$. L'ensemble d'arrivée peut être $\mathbb{R}$ dans le cas d'une image en niveaux de gris, ou $\mathbb{R}^3$ dans le cas d'une image couleur codée avec 3 canaux. L'estimation d'un *proxy géométrique* permet de définir des fonctions de déformation (ou *warps*), qui projettent un point d'une vue source sur le plan image cible :

$$\begin{array}{rccc} \tau_k: & \Omega_k & \to & \Gamma \\ & \mathbf{x} & \mapsto & \mathbf{x}' = \tau_k(\mathbf{x}). \end{array} \tag{2.3}$$

Sur la figure 2.12, le *proxy* illustré est un maillage (texturé pour l'illustration). Les

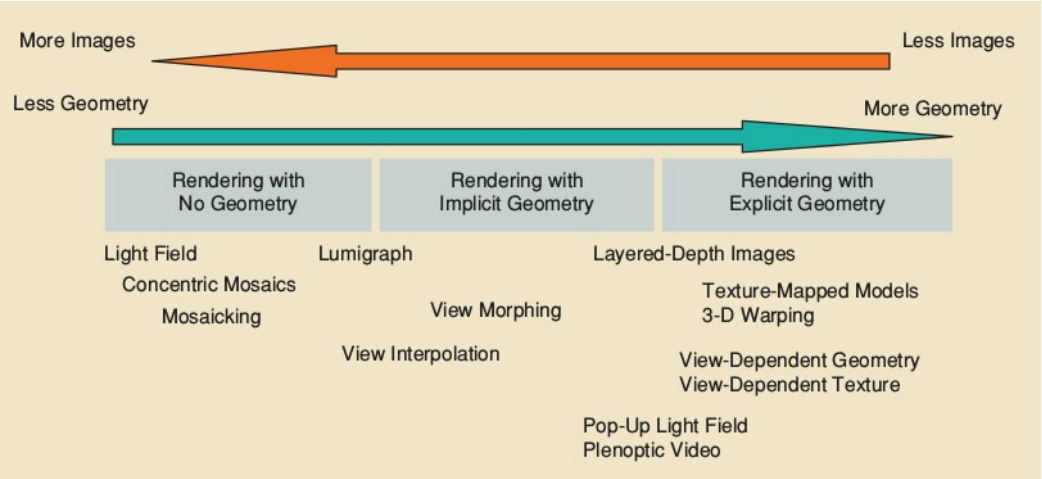


Fig. 2.11 – *Le* continuum IBR *de Shum* et al. *(2008).*

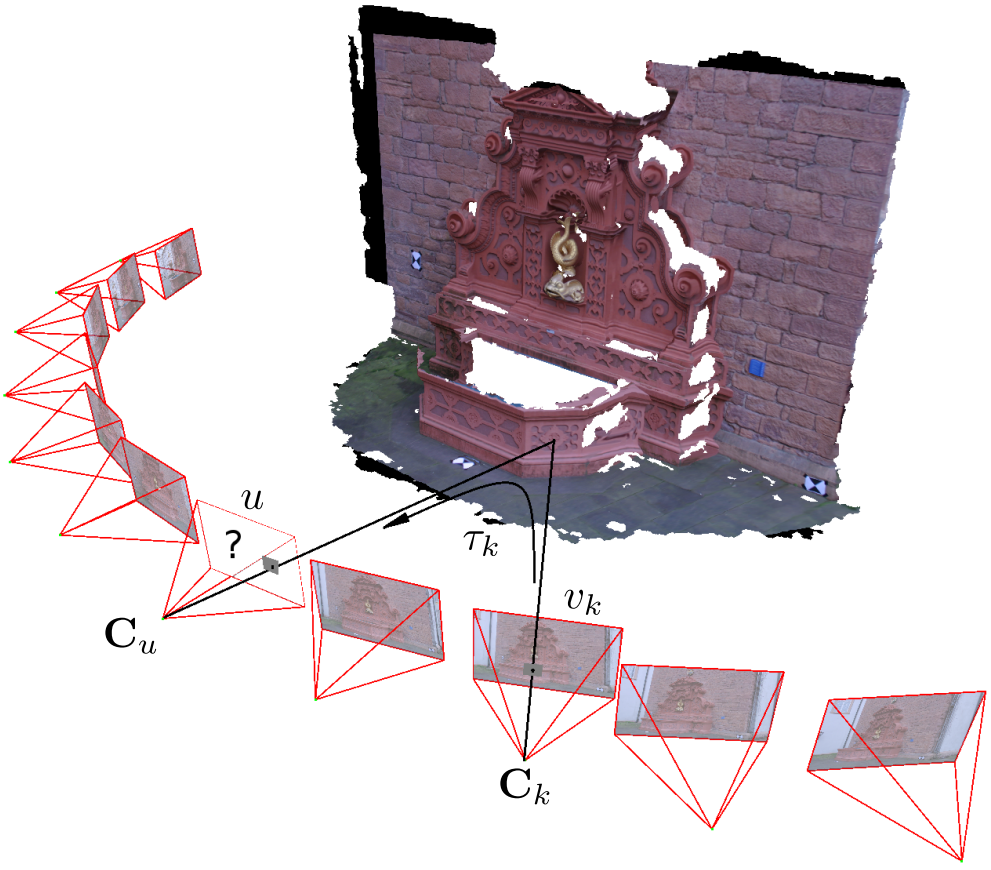


Fig. 2.12 – *Illustration du rendu basé image. Les warps ou fonctions de déformation τ_k établissent des correspondances entre les points des vues sources $\mathbf{C}_k$ et les points de la vue cible $\mathbf{C}_u$. Les images sources sont les applications v_k et l'image cherchée est l'application u.*

données sont extraites du jeu *fountain* (Strecha *et al.*, 2008). Il y a une ambiguïté dans la définition d'une image, qui peut-être comme présentée précédemment une application d'un sous-ensemble de $\mathbb{R}^2$ dans $\mathbb{R}$ ou $\mathbb{R}^3$ (selon le nombre de canaux pour décrire une couleur), mais qui sera parfois employée (à tort) pour décrire une vue ou une caméra.

Méthodes directes et méthodes variationnelles Nous classifions les algorithmes de rendu basé image en deux catégories distinctes : les méthodes *directes* et les méthodes *variationnelles*. Nos travaux contribuent aux deux approches, respectivement dans les chapitres 4 et 5.

L'approche *directe* est l'approche la plus naïve, et consiste à générer l'image cherchée par projection des images sources sur la vue cible par le biais des fonctions de déformation. La couleur de l'image cible est alors une combinaison des couleurs des images sources, pondérées selon leur contribution par pixel qui dépend de la proximité à la vue source, à la scène, et de l'incertitude sur le *proxy* reconstruit. En réalité l'approche est plus complexe qu'une simple moyenne pondérée, car il faut tenir compte des dilatations et des contractions d'image dues au fonctions de transformation, des occultations, des phénomènes d'accumulation de plusieurs points image projetés sur le même pixel d'arrivée dans la vue cible. Les méthodes directes se scindent elles-mêmes en plusieurs catégories suivant le type de projection utilisé (de la vue cible à la vue source, de la vue source à la vue cible, ou une méthode hybride), qui seront détaillées dans le chapitre 4.

L'approche *inverse*, ou *variationnelle*, consiste à générer une image par minimisation d'une *fonction de coût*, ou *énergie*. L'approche décrite dans le chapitre 5 est basée sur l'estimation MAP de u vue comme le paramètre d'un modèle (on estime en fait l'espérance du modèle). L'énergie à minimiser est dérivée d'une formulation bayésienne de la probabilité *a posteriori* de u sachant les données, c'est-à-dire les images sources. L'approche bayésienne est un type de méthode variationnelle qui permet d'inclure des *a priori* sur les paramètres du modèle à estimer, en particulier des contraintes sur l'image u indépendamment des données observées. La génération de l'image est plus lente qu'avec une approche directe car repose sur des algorithmes d'optimisation itératifs. Mais l'ajout d'un *a priori* permet la régularisation de la solution, qui est plus stable et donc moins sujette au bruit des données et au sur-ajustement (*overfitting* en anglais).

2.3 Concepts clés

Nous allons maintenant définir quelques concepts clés qu'il est utile de connaître pour la compréhension de ce manuscrit. Dans un premier temps nous abordons les propriétés essentielles des signaux que nous utilisons, que ce soient les images 2D, la géométrie 3D, ou la fonction plénoptique. Ces signaux ont un *support* qui est l'espace sur lequel ils sont définis. Ainsi les images sont définies sur des sous-ensembles de $\mathbb{R}^2$, la géométrie dans l'espace 3D et l'espace plénoptique se définit comme un espace à 7 dimensions (4 pour l'espace géométrique des rayons, et 3 pour l'espace des couleurs).

Tous ces supports sont continus, mais échantillonnés en signaux discrets lors de la mesure (capture d'une image par exemple) ou parce que l'algorithme de traitement repose sur la manipulation de signaux discrets. Il est par exemple nécessaire de discrétiser le signal image lors du rendu, même si les équations de rendu théoriques font intervenir les signaux continus.

Stabilité La *stabilité* est la propension d'un signal reconstruit à rester le même lorsque le nombre de données échantillonnées augmente. Ainsi une méthode de rendu variationnelle est dite *stable* si l'image 2D qu'elle estime varie peu lorsque l'on ajoute du bruit dans les images sources. On peut accroître la stabilité du rendu par l'ajout d'un terme de régularisation (voir chapitre 5). La reconstruction de Poisson d'une surface à partir d'un nuage de points (Kazhdan *et al.*, 2006), par exemple, est une méthode stable car peu sensible à l'incertitude de localisation du nuage de points estimé sur lequel elle s'appuie. De façon plus générale, les modèles stables sont des modèles avec peu de paramètres, peu sensibles au bruit des données sur lesquelles on les ajuste. Cependant il y a un risque de sous-ajustement (*underfitting*), qui peut-être compensé par l'ajout d'un *prior* jouant le rôle de régularisation.

Précision Au contraire on parlera de modèles *précis* lorsque les modèles sont fortement ajustés aux données, donc plus fidèles à l'observation. De ce fait ils sont généralement moins stables et plus sensibles au bruit des données : il s'agit donc de trouver un compromis entre *précision* et *stabilité*. Concernant les signaux échantillonnés, la *précision* désigne la faible incertitude de la mesure du signal, ou la confiance en l'estimation. Un nuage de points 3D estimé au laser est plus précis qu'un nuage estimé par n'importe quel algorithme MVS.

Sparsité Un signal *épars* est un signal échantillonné qui possède beaucoup de valeurs non informatives. On parlera souvent de champ lumineux *épars* pour signifier que la capture est peu dense, ce qui veut dire que peu d'images de la scène ont été prises, ou de façon équivalente que peu de rayons optiques ont été mesurés, comme dans le cas applicatif du camouflage. La *sparsité* peut aussi qualifier des nuages de points peu denses, des cartes de profondeur incomplètes. Il est l'antonyme de *densité*, mais se distingue de la résolution.

Résolution La *résolution* d'un signal peut signifier plusieurs choses. Il s'agit principalement de l'*échelle* du signal, c'est-à-dire du niveau de détail du signal. Une image de faible résolution est une image « floue », tandis qu'une image de haute résolution est « nette ». La décomposition d'un signal en plusieurs niveaux de résolution se fait par l'utilisation de filtres gaussiens, aussi appelé *pyramide gaussienne* lorsqu'elle est accompagnée d'un sous-échantillonnage des niveau supérieurs. La *résolution* est aussi employée (à tort) pour désigner la densité d'échantillonnage : on parle par exemple de *résolution d'image*, en pixel. Il s'agit de la taille de la grille employée pour discrétiser un signal. La *résolution d'image* est souvent liée au premier sens de *résolution*, car un sur-échantillonnage n'est pas nécessaire lorsque la

résolution, c'est-à-dire le niveau de détail, est faible. En effet d'après Nyquist, le taux d'échantillonnage suffisant pour reconstruire le signal continu dépend de la *résolution* du signal. Mais ces deux notions sont distinctes, car une images de faible *résolution* peut très bien être échantillonnée de façon dense.

Continuité Au sens large la continuité d'un signal est sa propension à changer d'une valeur infinitésimale lorsque l'un de ses paramètres change d'une valeur infinitésimale. Ainsi un signal est continu s'il n'a pas de changement brusque. Généralement un modèle ayant de nombreux paramètres s'ajustant fortement aux données observées, tend à être discontinu si les données sont bruitées. C'est le cas des cartes de profondeur haute résolution par exemple, dont on préfèrera une reconstruction de Poisson lisse en terme de continuité de surface. On parle aussi souvent de *ccohérence* plutôt que de continuité, surtout lorsque le signal est temporel (*cohérence temporelle*). Le champ lumineux reconstruit comme un ensemble de rayons estimé à partir des rayons mesurés dans les vues sources est sujet à cette propriété de *continuité*. Un changement infinitésimal dans l'espace géométrie 4D (le rayon estimé change légèrement de direction ou de position) doit résulter en un changement infinitésimal de couleur. Comme cette couleur est le plus souvent estimée à partir des couleurs des rayons mesurés dans les vues sources, la pondération des ces rayons doit aussi être une fonction continue de la position du rayon à estimer dans l'espace géométrique. Il s'agit de la propriété de *continuité* d'un algorithme de rendu basé image énoncée par Buehler *et al.* (2001).

2.4 Modèle du sténopé

Il y a k caméras sources, $k \in [1 \dots K]$. Nous avons choisi le modèle du sténopé pour représenter les caméras : une matrice 3×3 de paramètres intrinsèques $\mathbf{K}_k$, ainsi que la matrice 3×3 de rotation $\mathbf{R}_k$ et le vecteur de translation 3D $\mathbf{t}_k$ permettent le changement en coordonnées monde/caméra. Le centre optique de chaque caméra peut être calculé à partir des paramètres extrinsèques : $\mathbf{C}_k = -\mathbf{R}_k^\intercal \mathbf{t}_k$. Si l'on appelle $\mathbf{X}$ le point 3D associé au point image homogène $\tilde{\mathbf{x}}$ situé à une distance orthogonale z de la caméra k, on a alors $\tilde{\mathbf{x}} = \mathbf{K}_k(\mathbf{R}_k\mathbf{X} + \mathbf{t}_k)$ ou encore en coordonnées monde $\mathbf{X} = \mathbf{R}_k^\intercal \mathbf{K}_k^{-1}\tilde{\mathbf{x}} + \mathbf{C}_k$. Nous supposons connus les paramètres de la vue à synthétiser $\mathbf{K}_u$, $\mathbf{R}_u$, $\mathbf{t}_u$ et $\mathbf{C}_u$.

Paramètres intrinsèques La forme générale de la matrice $\mathbf{K}$ des paramètres intrinsèques, selon le modèle du sténopé, est

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

$f_x = f.m_x$ et $f_y = f.m_y$ sont les distances focales en pixel, et m_x et m_y sont les facteurs définissant le ratio des dimensions d'un pixel. En pratique on prend ce ratio

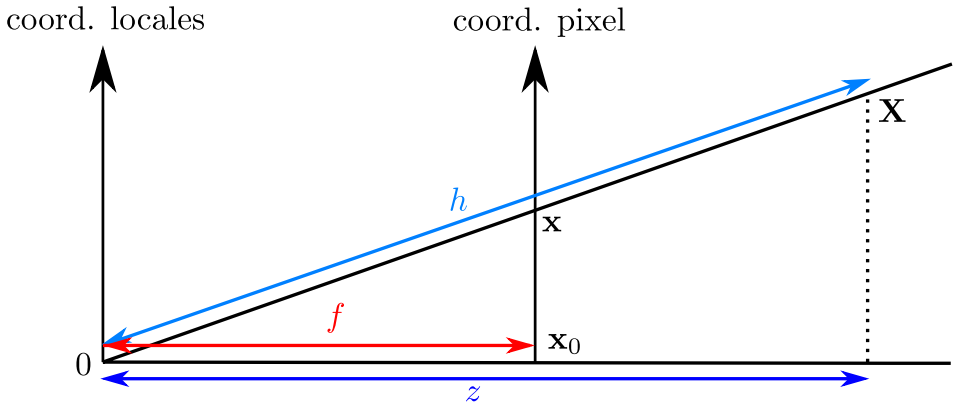


Fig. 2.13 – *La conversion de la profondeur radiale h en profondeur orthogonale z nécessite le passage des coordonnées pixel $\mathbf{X}$ aux coordonnées locales à la caméra k du point 3D $\mathbf{x}$. La distance focale f et le décalage de l'origine $\mathbf{x}_0$ sont les paramètres intrinsèques de la caméra k contenus dans la matrice $\mathbf{K}$.*

égal à 1 et on exprime la distance focale f en pixel, de telle sorte que $f_x = f_y = 1$. s est le *skew*, qui est nul en pratique. Les décalages x_0 et y_0 seront la plupart du temps égaux à la moitié de la largeur et respectivement de la hauteur de l'image en coordonnées pixel. On utilise donc le plus souvent une matrice de la forme

$$\mathbf{K} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{et son inverse} \quad \mathbf{K}^{-1} \begin{pmatrix} \frac{1}{f} & 0 & \frac{-x_0}{f} \\ 0 & \frac{1}{f} & \frac{-y_0}{f} \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.5}$$

La transformation inverse $\mathbf{K}^{-1}$ permet, étant donnés un point image 2D et une profondeur z, de retrouver le point 3D en coordonnées locales à la caméra. On note $\tilde{\mathbf{x}} = z.\bar{\mathbf{x}} = z.(x, y, 1)$ le point homogène associé à $\mathbf{x} = (x, y)$. En coordonnées locales à la caméra, le point 3D s'écrit $\mathbf{X} = \mathbf{K}^{-1}\tilde{\mathbf{x}} = z.\mathbf{K}^{-1}\bar{\mathbf{x}}$.

$$\mathbf{X} = \mathbf{K}^{-1}\tilde{\mathbf{x}} = z.\mathbf{K}^{-1}\bar{\mathbf{x}} = z. \begin{pmatrix} \frac{\mathbf{X}-\mathbf{x}_0}{f} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{z}{f}.(\mathbf{x}-\mathbf{x}_0) \\ z \end{pmatrix}. \tag{2.6}$$

d'après la figure 2.13.

Coordonnées homogènes et coordonnées euclidiennes On notera $\bar{\mathbf{x}} = (x, y, 1)$ le point image 2D $\mathbf{x} = (x, y) \in \Omega_k$ auquel on a ajouté une troisième coordonnée, en unités pixel. De la même façon on note $\tilde{\mathbf{x}} = z.\bar{\mathbf{x}} = z.(x, y, 1)$ le point homogène associé. Le passage des coordonnées homogènes en coordonnées euclidiennes se fait par le biais de la matrice de normalisation, qui divise un point en coordonnées homogènes par sa dernière composante, ici la profondeur orthogonale z du point 3D correspondant:

$$N_e(\tilde{\mathbf{x}}) = \mathbf{x} = \begin{pmatrix} \frac{1}{z} & 0 & 0 \\ 0 & \frac{1}{z} & 0 \end{pmatrix} \tilde{\mathbf{x}}. \tag{2.7}$$

Le passage de coordonnées euclidiennes en coordonnées étendues 3D se fait via la transformation N_h telle que

$$N_h(\mathbf{x}) = \bar{\mathbf{x}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{2.8}$$

La jacobienne J_h de N_h est constante :

$$J_h(\mathbf{x}) = \frac{\partial N_h}{\partial \mathbf{x}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{2.9}$$

En revanche celle de N_e dépend des coordonnées du point homogène :

$$J_e(\tilde{\mathbf{x}}) = \frac{\partial N_e}{\partial \tilde{\mathbf{x}}} = \frac{1}{z}. \begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \end{pmatrix}. \tag{2.10}$$

Ces jacobiennes de normalisation en coordonnées homogènes sont données par Heuel (2004) à la page 110. Nous invitons le lecteur à consulter cet ouvrage pour de plus amples informations sur la géométrie projective dans le contexte de la propagation d'incertitude.

Logiciels de calibration Quel que soit l'algorithme utilisé pour reconstruire le champ lumineux, ou dans une moindre mesure un *proxy géométrique*, il convient dans un premier temps de calibrer les caméras. Cette calibration doit s'effectuer par un logiciel de SfM (*Structure from Motion*), s'inscrivant dans une *pipeline* complète de reconstruction automatique, des images capturées aux estimations des fonctions de déformation. Le logiciel *openMVG* (Moulon *et al.*, 2013) est utilisé pour estimer les paramètres des caméras. À l'issu de cette étape les matrices de projection des caméras sont connues et on peut donc utiliser la géométrie épipolaire pour estimer des *proxys géométriques* par MVS (*Multi-vues Stereo*). Un autre logiciel de SfM communément utilisé pour la calibration des caméras dans des *pipelines* de reconstruction est *Bundler* (Snavely *et al.*, 2006a, 2008). Les algorithmes de SfM procurent aussi une reconstruction grossière de la scène sous la forme d'un nuage de points. Le nuage reconstruit n'est pas assez dense pour des applications de rendu basé image, mais peut servir à l'initialisation d'un algorithme de MVS, dont le but est de fournir un *proxy géométrique* plus dense, complexe et précis.

2.5 La chaîne logicielle

La chaîne logicielle classique du rendu basé image se décompose en deux parties : la reconstruction d'un *proxy*, puis le rendu. C'est au rendu que nous contribuons

principalement, car les algorithmes employés pour la reconstruction d'un *proxy* sont extraits de logiciels préexistants. La première étape consiste donc à reconstruire la géométrie 3D pour estimer des fonctions de déformation. L'obtention des fonctions de déformation et la reconstruction du *proxy* font l'objet du chapitre 3. Cette étape se décompose en plusieurs sous-étapes :

- Traitement préliminaire des images sources : conversion de format et correction de la distorsion radiale. En effet la distorsion radiale ne faisant pas partie de notre modèle de caméra, il est nécessaire de la corriger au préalable.
- Calibration des caméras par un logiciel de SfM : *Bundler* (Snavely *et al.*, 2006a, 2008) ou *openMVG* (Moulon *et al.*, 2013) par exemple. À l'issue de cette étape on a estimé les matrices des caméras sources.
- Reconstruction du *proxy géométrique* (nuage de points, cartes de profondeur ou maillage) : à l'aide d'un logiciel de MVS, MVE (*Multi-View Environment*) (Fuhrmann *et al.*, 2014) par exemple. Parfois des étapes intermédiaires d'affinage de la géométrie sont nécessaires, comme le traitement du nuage de points avant la reconstruction de la surface pour éliminer les *outliers*.
- Post-traitement du *proxy* : affinage de la surface reconstruite ou lissage des cartes de profondeur par exemple.
- Calcul des fonctions de déformations et de leurs dérivées (par rapport à $\mathbf{x}$ et z à partir du *proxy géométrique*. Accessoirement ces dérivées permettent de propager l'incertitude et de calculer les poids des contributions de chaque pixel de chaque vue source (voir chapitre 4).

L'étape de rendu dépend intégralement de la méthode de rendu employée : directe ou inverse. Nous proposons des contributions dans chacune d'entre elles. Elles sont présentées dans les chapitres 4 et 5.

De façon optionnelle, le rendu peut s'accompagner d'une étape supplémentaire de traitement d'image pour faire disparaître les zones de l'image laissées en noir par manque d'information (occultations, *proxy* incomplet). Ce processus est appelé *inpainting*, ou encore *hole filling* (remplissage de trous). Pour y parvenir on peut s'aider des multiples vues sources, ou alors effectuer le remplissage de façon plus naïve en ne considérant que l'image synthétisée, par diffusion par exemple.

2.6 Évaluation

Jeux de données Nous évaluons nos résultats sur les jeux de données réels et synthétiques produits initialement pour les algorithmes de MVS.

- La base de données de Strecha *et al.* (2008) [1] est idéale pour tester les méthodes de rendu basé sur une reconstruction MVS classique. Elle contient plusieurs lots d'une dizaine à une trentaine d'images en haute résolution (6 Mpx) de bâtiments, prises en extérieur. Les points de vue sont non structurés, peu nombreux (comparés aux jeux de données de type *light field*) et d'entraxe élevé, ce qui en fait un échantillonnage angulaire épars. Elle présente cependant peu de difficultés et les images résultats sont de très bonne qualité, ce qui rend la

1. http://cvlabwww.epfl.ch/data/multiview/denseMVS.html

distinction entre plusieurs algorithmes de rendu difficile. En effet, les surfaces sont faciles à reconstruire car lambertiennes et très texturées ; on utilisera donc les images sous-échantillonnées pour reconstruire la géométrie, dans le but de mettre en évidence des artefacts de rendu.

- *Stanford Light Field Archive* est une compilation de jeux de données de type *light field*. Elle propose un échantillonnage angulaire dense (17 × 17 images), une résolution spatiale d'image élevée (8 Mpx), et de faibles entraxes. Certaines scènes sont difficiles à reconstruire car elles comportent de nombreuses réfractions et spécularités (*amethyst*, *tarot*, *chest* par exemple), ce qui en fait une archive de choix pour évaluer la performance de notre algorithme de reconstruction de champ lumineux (chapitre 6). La capture a été faite en laboratoire, à l'aide d'un bras mécanique ou d'un ensemble de caméras (Wilburn *et al.*, 2005). Nous utilisons les images rectifiées (1 Mpx) ou les images brutes (8 Mpx), selon l'algorithme que nous souhaitons évaluer. L'archive comporte en outre deux jeux de données de camouflage pris en extérieur.
- *HCI Light Field Benchmark Datasets* (Wanner, Sven *et al.*, 2013) est un jeu de données synthétiques et réelles créé par un script de rendu *Blender* imitant un bras mécanique, ou un ensemble de caméras. Il vient enrichir le jeu de données de la *Stanford Light Field Archive*.
- Nous utilisons aussi nos propres jeux de données : le lot d'images synthétiques *skull*, et les lots d'images prises en extérieur avec un appareil portable *lion*, *charce* et *hercule*. Ces derniers présentent des scènes diffuses et statiques, capturées en haute résolution, en variant le point de vue (6 degrés de liberté) mais pas la focale.

Voici d'autres jeux de données que nous n'utilisons pas car ils présentent certains inconvénients que nous détaillons brièvement.

- *MPI Sintel* (Butler *et al.*, 2012) est conçu pour évaluer le flux optique. Le flux optique que nous utilisons dans nos expériences est pris comme une *boîte noire*, il n'est donc pas soumis à l'évaluation.
- *DTU* (Jensen *et al.*, 2014) est peu varié en termes de points de vue.
- *Middlebury dataset* (Scharstein et Szeliski, 2002; Scharstein *et al.*, 2014) ne proposent que des paires de vues stéréoscopiques, ou multi-stéréoscopiques avec un arrangement linéaire de caméras.
- *Middlebury MVS evaluation* (Seitz *et al.*, 2006) ne propose que quelques lots d'images, produites en intérieur dans des conditions bien contrôlées, ne comportant pas de grande difficultés en termes de réfraction, spécularités et autres phénomènes optiques complexes pouvant altérer la reconstruction du champ lumineux.
- *KITTI* (Geiger, 2012) offre de nombreux lots d'images pour évaluer le flux de scène (Menze et Geiger, 2015). Mais la résolution spatiale est faible (0.5 Mpx) et les scènes ne comportent que des routes et des rues car il est conçu pour la conduite automobile.
- *ETH3D* (Schöps *et al.*, 2017) est un jeu de données très récent. Il offre une grande variété de scènes, capturées par des paires stéréoscopiques et des ensembles de caméras, des vidéos, avec des points de vue variés, non structurés,

6 degrés de liberté et différentes focales. Malheureusement il est sorti bien trop tard pour que nous ayons pu faire fonctionner nos algorithmes dessus.

Critères d'évaluation Outre les comparaisons purement visuelles pour valider nos résultats, nous utilisons aussi des critères d'évaluation numériques. Ces critères permettent de comparer deux images : une image de référence et l'image générée par le rendu basé image, ou alors plusieurs images générées par des modèles ou des jeux de paramètres différents. Généralement la vue de référence est une vue du jeu de base qui est exclue de l'ensemble des vues sources, et que l'on souhaite générer. Deux critères sont utilisés dans ces travaux : le PSNR et le DSSIM.

- PSNR : *Peak Signal to Noise Ratio*. Il est relatif à l'erreur quadratique moyenne, qui s'appuie sur les différences par pixel entre les deux images comparées, l'image de référence u_0 et l'image résultat u, de taille égale $W \times H$ pixels :

$$PSNR = 10 \log_{10} \left(\frac{d^2}{EQM} \right). \tag{2.11}$$

  d est généralement 255 car nous comparons des images 8 bits, et l'énergie quadratique moyenne s'écrit

$$EQM = \frac{1}{WH} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} (u_0(i,j) - u(i,j))^2. \tag{2.12}$$

  Plus le PSNR est grand, plus les images sont proches (pixel à pixel).
- DSSIM : c'est l'opposé de la *Structural SIMilarity* (Wang *et al.*, 2004). DSSIM permet mesurer la dissimilarité de structure entre deux images, plutôt que des mesures d'erreur pixel à pixel comme le PSNR. DSSIM se prétend plus proche de la perception de l'œil humain, et des artefacts que l'on peut visuellement déceler dans les images. Elle est en outre invariante à certaines transformations comme le changement d'échelle ou la translation. La valeur de DSSIM est calculée sur une fenêtre centrée sur le pixel, en faisant intervenir des mesures de variance, covariance et moyenne. Plus le DSSIM est petit, plus les images sont proches.

L'évaluation objective de la qualité visuelle d'une image par comparaison avec une référence est cependant un problème qui reste ouvert.

3 Estimation des fonctions de déformation

L'estimation d'un *proxy géométrique* fait partie intégrante de nombreuses méthodes de rendu. En effet ce *proxy géométrique*, quelle que soit sa nature (maillage, nuage de points ou cartes de profondeur), permet d'établir des correspondances entre les vues sources et la vue cible, afin d'effectuer la synthèse de la vue cible. Intuitivement, un rendu en haute qualité nécessite une reconstruction 3D fidèle de la scène, le corollaire étant qu'avant de perfectionner notre méthode de synthèse de vue nous devons d'abord garantir l'obtention d'un *proxy géométrique* sans artefacts. Ces artefacts sont de deux types : le bruit haute fréquence (figure 3.1a) et l'incertitude de localisation (figure 3.1b), et se répercutent sur l'image synthétisée. Le bruit haute fréquence se caractérise par une forte discontinuité spatiale du *proxy géométrique* due à la surévaluation de la résolution de reconstruction par rapport à l'incertitude sur les données. Les fonctions de déformation calculées à partir d'un *proxy géométrique* sont alors elles-mêmes bruitées et de fortes discontinuités sont observables dans la vue synthétisée. Une forte incertitude de localisation n'est pas corrélée à l'apparition de discontinuités, mais il en résulte un mauvais alignement des vues sources, ce qui conduit souvent à du flou ou un effet fantôme (dédoublement de l'image) dans l'image synthétisée. Dans ce chapitre nous donnons un aperçu des méthodes MVS (*Multi-View Stereo*) courament utilisées en IBR pour estimer un *proxy géométrique*. Nous montrerons dans quelle mesure l'algorithme de reconstruction crée les artefacts cités précédemment, et quelle méthode privilégier pour les éviter.

3.1 Triangulation classique d'un point lambertien

Le problème de triangulation d'un point est très simple, mais il est à la base de notre méthode de linéarisation de l'espace plénoptique présentée en section 6. Il est voisin du problème d'ajustement de faisceau, qui sert généralement à affiner la géométrie obtenue par SfM (*Structure from Motion*), à la différence que les matrices

(a)

(b)

Fig. 3.1 – *Principaux artefacts de rendu causés par une mauvaise reconstruction 3D de la scène. (a) Discontinuités causées par du bruit haute-fréquence. Le rendu (à gauche) d'une vue d'une scène de Strecha et al. (2008) présente des artefacts haute-fréquence (en haute à droite). En bas à droite, une carte de profondeur présentant des discontinuités dues à la visibilité à cet endroit de l'image. (b) Un effet fantôme extrême provoqué par un mauvais alignement des vues. La géométrie reconstruite est plane, d'où l'incertitude de localisation très élevée.*

de caméra sont déjà estimées et que seule la position du point 3D est inconnue. Il se distingue des problèmes de MVS qui cherchent à reconstruire des nuages de points car le problème de mise en correspondance est occulté.

On cherche à estimer la position d'un point 3D $\mathbf{X}$ visible dans un ensemble de K caméras observant ce point, dont on connait les matrices de projection $\mathbf{P}_k$, avec $k \in [1..K]$. On suppose qu'on a mis en correspondance des points image se rapportant aux même faisceaux de rayons, par une méthode quelconque. On a donc un ensemble de points 2D $\mathbf{x}_k^*$, qui représentent les données du problème.

Estimation par DLT (*Direct Linear Transform* La méthode de triangulation par DLT est peu précise mais il s'agit de la plus simple ; elle peut en outre servir à initialiser la solution pour accélérer la convergence lors d'une estimation ML par exemple. En exploitant le fait que le point 3D $\mathbf{X}$ est lié aux points image $\mathbf{x}_k$ dans les vues sources via les projections $\mathbf{P}_k$, on obtient un système d'équations linéaires sur-contraint que l'on résout. En effet pour chaque point image $\mathbf{x}_k = (x, y)$ dans une vue source k, le produit en croix $\mathbf{x}_k \times \mathbf{P}_k\mathbf{X}$ est nul, ce qui nous donne trois équation par vue source :

$$x(\mathbf{P}^{3\intercal}\mathbf{X}) - (\mathbf{P}^{1\intercal}\mathbf{X}) = 0 \tag{3.1}$$
$$y(\mathbf{P}^{3\intercal}\mathbf{X}) - (\mathbf{P}^{2\intercal}\mathbf{X}) = 0 \tag{3.2}$$
$$x(\mathbf{P}^{2\intercal}\mathbf{X}) - y(\mathbf{P}^{1\intercal}\mathbf{X}) = 0 \tag{3.3}$$

où les $\mathbf{P}^{i\intercal}$ désignent les lignes de $\mathbf{P}$. Mais les données images sont bien souvent perturbées par le bruit du capteur ou l'erreur de mise en correspondance. Par conséquence le système n'a pas de solution exacte (les rayons optiques ne s'intersectent pas). Le système est résolu par une méthode aux moindres carrés linéaire. Plus de détails peuvent être trouvés dans le livre de Hartley et Zisserman (2003) à la page 312.

Estimation ML (*Maximum Likelihood*) La méthode de triangulation par maximisation de vraisemblance (ML) cherche à trouver un estimateur qui minimise l'erreur de reprojection dans les vues sources. Comme nous l'avons vu précédemment, le bruit du capteur et l'erreur de mise en correspondance perturbent les données, et la projection du point 3D sur une caméra $\mathbf{x}_k$ ne correspond pas exactement au point image $\mathbf{x}_k^*$. On note cette erreur $\varepsilon = \mathbf{x}_k^* - \mathbf{x}_k = \mathbf{x}_k^* - \mathbf{P}_k\mathbf{X}$. On suppose que l'erreur de reprojection sur la vue k suit une distribution normale, d'espérance nulle et de matrice 2×2 de covariance $\Sigma_{\mathbf{x}_k\mathbf{x}_k}$ que l'on prendra égale à l'identité dans un premier temps. On cherche à trouver un estimateur $\hat{\mathbf{X}}$ de la variable aléatoire $\mathbf{X}$,

par maximisation de la vraisemblance (estimateur ML, pour *Maximum Likelihood*) :

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} P((\mathbf{x}_k^*)_{k\in[1..K]}|\mathbf{X}). \tag{3.4}$$

$$= \arg\min_{\mathbf{X}} -\sum_{k=1}^{K} \ln P(\mathbf{x}_k^*|\mathbf{X}). \tag{3.5}$$

$$= \arg\min_{\mathbf{X}} -\sum_{k=1}^{K} (\mathbf{x}_k^* - \mathbf{P}_k\mathbf{X})^\intercal \Sigma^{-1}_{\mathbf{x}_k\mathbf{x}_k} (\mathbf{x}_k^* - \mathbf{P}_k\mathbf{X}). \tag{3.6}$$

$\mathbf{P}_k$ est la matrice de projection de la caméra k. Il s'agit de résoudre un problème des moindres carrés non linéaire. La fonction de coût à minimiser est

$$f(\mathbf{X}) = \sum_{k=1}^{K} f_k(\mathbf{X}) \quad où \quad f_k(\mathbf{X}) = (\mathbf{x}_k^* - \mathbf{P}_k\mathbf{X})^\intercal \Sigma^{-1}_{\mathbf{x}_k\mathbf{x}_k} (\mathbf{x}_k^* - \mathbf{P}_k\mathbf{X}).$$

Incertitude sur le point 3D Il est possible d'exprimer l'incertitude sur le point 3D reconstruit à partir de l'incertitude associée à l'erreur de reprojection dans le plan capteur de chaque caméra source. En effet l'estimateur qui maximise la vraisemblance est l'espérance d'une distribution, que l'on déduit comme étant gaussienne par propagation d'incertitude. On peut la représenter comme un ellipsoide 3D centré sur le point 3D reconstruit, dont l'enveloppe convexe est bornée par les cônes d'incertitude projetés de chaque caméra. Il s'agit en réalité d'une propagation d'incertitude *rétrograde* (*backward propagation*), puisque la propagation classique serait du point 3D sur le plan capteur de la caméra via la matrice de projection. A l'inverse on cherche à exprimer la matrice de covariance $\Sigma_{\mathbf{XX}}$ du point 3D en fonction des incertitudes des points image $\Sigma_{\mathbf{x}_k\mathbf{x}_k}$. Une formule pour calculer l'incertitude $\Sigma_{\mathbf{XX}}$ par propagation d'incertitude *rétrograde* est donnée par Hartley et Zisserman (2003). Le calcul de cette formule repose sur celui d'une jacobienne $2K \times 3$ $\mathbf{J}_{\mathbf{P}}(\mathbf{X})$ des projections $\mathbf{P}_k$ du point 3D sur chaque vue source qui voit ce point :

$$\mathbf{J}_{\mathbf{P}}(\mathbf{X}) = (\mathbf{J}_{\mathbf{P}_1}(\mathbf{X})^\intercal \ \ldots \ \mathbf{J}_{\mathbf{P}_k}(\mathbf{X})^\intercal \ \ldots \ \mathbf{J}_{\mathbf{P}_K}(\mathbf{X})^\intercal)^\intercal. \tag{3.7}$$

On peut démontrer que chaque $\mathbf{J}_{\mathbf{P}_k}(\mathbf{X})$ est obtenu en dérivant la transformation de projection $\mathbf{P}_k$ par rapport à $\mathbf{X}$:

$$\mathbf{J}_{\mathbf{P}_k}(\mathbf{X}) = \mathbf{J}_{N_e}(\mathbf{x}_k^h)\mathbf{K}_k\mathbf{R}_k. \tag{3.8}$$

On trouve $\Sigma_{\mathbf{XX}}$ par propagation rétrograde de l'erreur de reprojection, qui peut être modélisée par $\Sigma_{\mathbf{x}_k\mathbf{x}_k}$, une matrice de covariance $2K \times 2K$. Pour des simplicités d'implémentation nous prenons cette matrice égale à l'identité :

$$\Sigma_{\mathbf{XX}} = (\mathbf{J}_{\mathbf{P}}(\mathbf{X})^\intercal {\Sigma_{\mathbf{xx}}}^{-1} \mathbf{J}_{\mathbf{P}}(\mathbf{X}))^\dagger = (\mathbf{J}_{\mathbf{P}}(\mathbf{X})^\intercal \mathbf{J}_{\mathbf{P}}(\mathbf{X}))^\dagger, \tag{3.9}$$

où $\mathbf{A}^\dagger$ est le pseudo-inverse de $\mathbf{A}$, et $\Sigma_{\mathbf{xx}}$ est la matrice bloc-diagonale $2K \times 2K$ des $\Sigma_{\mathbf{x}_k\mathbf{x}_k}$. Les formules de propagation d'incertitude nous permettent facilement de

développer l'incertitude sur le point 3D reconstruit. Cela est très utile pour le rendu car le calcul des contributions des vues sources repose en partie sur l'incertitude de la géométrie, comme nous le verrons dans les chapitres 4 et 5.

Résumé La triangulation classique du point est un problème mathématique très simple, dont l'extension (voir chapitre 6) peut permettre d'ajuster des modèles plus complexes avec plus de 3 paramètres. En outre on peut aisément calculer l'incertitude sur le point 3D reconstruit (variance de la distribution) à partir des matrices de projection et de l'incertitude des points 2D (due à l'erreur de mise en correspondance ou de mesure). Cependant cela nécessite une méthode préalable de mise en correspondance, qui n'est pas détaillée ici, comme une méthode de MVS ou un flux optique par exemple (voir chapitre 6). Il est à noter qu'aucune information sur la normale de la surface locale n'est estimée. Par conséquent on ne peut reconstruire la surface, et le rendu basé point est la seule option pour synthétiser une nouvelle vue. Nous explorons par la suite des méthodes de MVS qui proposent de reconstruire à la fois des points et leur normale.

3.2 Reconstruction d'un nuage de points

Un grande partie des méthodes de MVS proposent de reconstruire la géométrie de la scène sous la forme d'un nuage de points. L'avantage d'un nuage est qu'il est facile à manipuler et ne produit pas de redondance de données comme les cartes de profondeur. Sa reconstruction se base le même principe de triangulation vu précédemment, avec le procédé de mise en correspondance en plus. Initialement un ensemble d'images d'une scène est requis, ainsi que les matrices de projection des caméras (pour pouvoir utiliser la géométrie épipolaire), obtenues par calibration. Il en résulte un nuage de points orientés, ou *patches* ; un point orienté est un triplé de coordonnées $\mathbf{X}$ associé à une normale $\mathbf{n}$. La sortie de l'algorithme est donc plus riche qu'une simple triangulation, et la normale estimée qui donne l'orientation du *patch* est utile au rendu ou à la reconstruction de surface. En effet, à l'issue de la reconstruction du nuage de points, on peut soit appliquer la méthode de rendu basé point appelée *splatting* ou rendu par éclaboussures, décrite dans le chapitre 4, soit continuer le processus d'estimation de *proxy* affin de reconstruire une surface sous la forme d'un maillage (voir section 3.4). La reconstruction d'un nuage de points n'est donc pas forcément une fin en soi. Ensuite, nous détaillerons comment passer d'un nuage de points épars à des cartes de profondeur denses, dans le but d'estimer des fonctions de déformation pour chaque vue en prenant en compte les occultations. Cela permet par exemple d'obtenir facilement l'incertitude sur les déformations estimées par propagation d'erreur à partir des incertitudes des points 3D triangulés.

Patch-based Multi-View Stereo Nous avons choisi d'expérimenter sur une méthode de reconstruction par *patch* (Furukawa et Ponce, 2010), implémentée dans les logiciels PMVS et PMVS2 (*Patch-based Multi-View Stereo*). Cette méthode fut étenue à la reconstruction de vastes scènes (Furukawa *et al.*, 2010), par regroupement d'images en collections se recoupant puis par fusion des nuages obtenus en

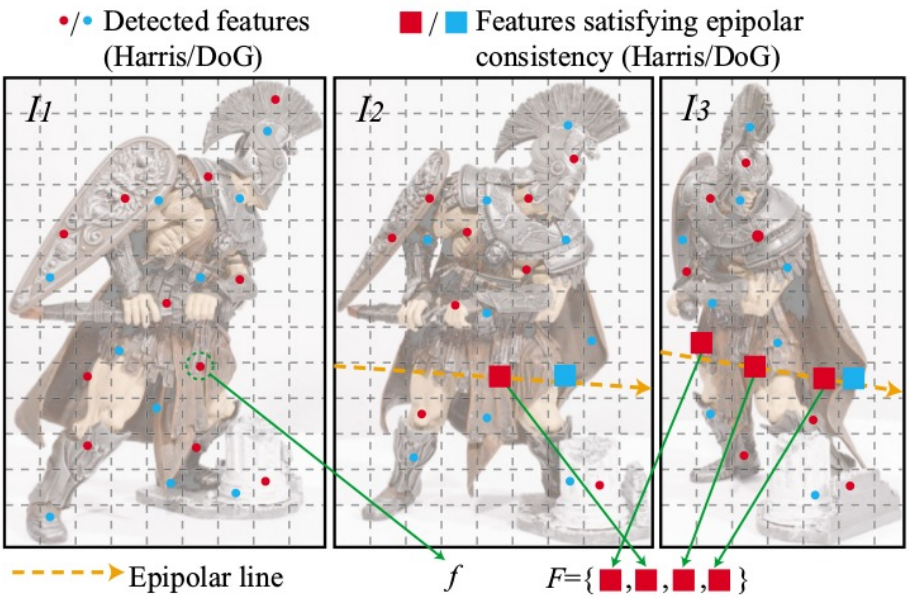


Fig. 3.2 – *Figure de Furukawa et Ponce (2010). On cherche à mettre en correspondance un point f d'une image de référence I_1 avec des points des images I_2 et I_3. Pour cela on sélectionne les points clé $f' \in F$ le long de la droite épipolaire dans les images I_2 et I_3, pris comme correspondants 2D potentiels, et on les range par ordre de distance au centre de l'image de référence croissante.*

exécutant l'algorithme sur chaque collection en parallèle. Cette extension (Furukawa *et al.*, 2010) est implémentée dans le logiciel CMVS [1] (*Clustering Views for Multi-view Stereo*).

La méthode par *patch* se décompose en trois étapes distinctes : la *mise en correspondance*, l'*expansion* et le *filtrage*.

- L'étape de *mise en correspondance* initialise un nuage de points épars par triangulation. Un détecteur de coins et de contours comme celui d'Harris (Harris et Stephens, 1988) ou une différence de gaussiennes (DoG, *Difference of Gaussian* en anglais) permet d'extraire des points clés des images. Pour une image donnée, on observe les correspondants 2D potentiels le long de la droite épipolaire (figure 3.2). On triangule un point 3D si le nombre de correspondants 2D photocohérents est suffisant. Le point 3D et sa normale sont ensuite affinés par un processus d'optimisation. La mesure de la photo-cohérence se fait par NCC (*Normalized Cross Correlation*). Elle est la plus utilisée en MVS, mais d'autres algorithmes utilisent aussi SSD (*Sum of Squared Differences*), SAD (*Sum of Absolute Differences*), *Census*, *Rank* ou encore MI (*Mutual Information*).
- L'étape d'*expansion* étend la reconstruction aux voisins des *patches* reconstruits, en assurant les contraintes en termes de photo-cohérence, de visibilité, et en garantissant une certaine homogénéité de la géométrie. L'objectif est d'obtenir un nuage de *patch* suffisamment dense pour recouvrir l'ensemble des surfaces visibles de la scène.
- L'étape de *filtrage* permet de supprimer les points qui n'appartiennent pas à la

1. *https://www.di.ens.fr/cmvs/*

surface, reconstruits soit devant, soit dernière la surface.

Du nuage aux cartes de profondeur Le nuage de points est idéal pour des applications de *free viewpoint* car la géométrie est indépendante des vues et la couleur de chaque point est calculée une fois, indépendamment du point de vue à synthétiser, comme combinaison des contributions des vues sources. Mais l'approche que nous avons du rendu basé image est celle de *view-dependent texture synthesis*, car elle produit des résultats de meilleure qualité. Cependant un nuage de points obtenu par triangulation ne peut, tel quel, produire un rendu de qualité, car il est souvent beaucoup trop épars (figure 3.3) L'idée clé est de densifier les données, soit par un rendu basé point qui exagère la taille des *patches*, ou *splat* (voir chapitre 4), soit par une méthode de rendu de cartes de profondeur qui sur-échantillonne les cartes et interpole intelligemment les profondeurs manquantes. C'est cette dernière option que nous allons décrire ici, car elle permet en outre de convertir le format « nuage de points » en un autre format qui nous permet plus facilement d'extraire nos fonctions de déformation. Nous présentons ici une méthode de sur-échantillonnage similaire au filtre bilatéral de Kopf *et al.* (2007). Si ce dernier propose de densifier des cartes de profondeur en augmentant la *densité* de l'échantillonnage puis en « remplissant les trous », nous proposons plutôt une approche de rendu de points. Ceux-ci sont vus comme des éclaboussures, ou *splats*, dont la taille est volontairement exagérée dans le but de combler les zones manquant d'information. Notre méthode a l'avantage de traiter la géométrie 3D directement, avant même d'effectuer le rendu, ce qui permet de prendre en compte les positions spatiales des points, les normales (orientation des éclaboussures), ainsi que l'incertitude des points triangulés ; alors que Kopf *et al.* (2007) ne prennent en compte que la profondeur des points 3D.

Notre algorithme s'effectue en deux temps : on projette d'abord le nuage de points sur chaque caméra source pour créer des cartes de profondeur éparses. On remplit ensuite les vides entre les projections des points par une méthode non-itérative. Le calcul des cartes de profondeur doit tenir compte du fait que la scène contient des auto-occultations : les surfaces qui occultent l'arrière-plan de la scène. Nous n'avons pas de connaissances *a priori* sur ces éventuelles occultations, et aucun Z-test n'est effectué durant la projection du nuage. Un autre point à prendre en compte est l'irrégularité du nuage de points estimé, en particulier pour des surfaces présentant des textures uniformes, dont les points image sont difficilement mis en correspondance.

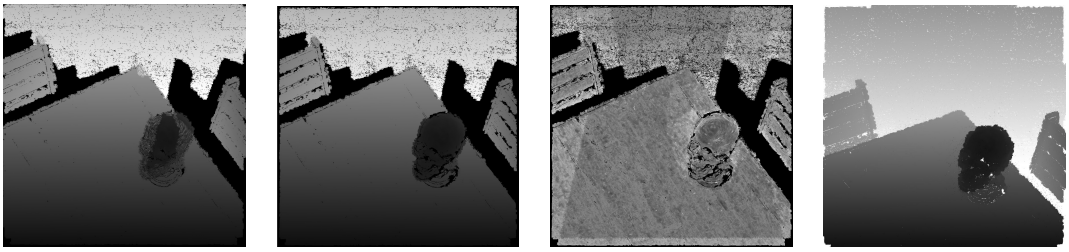
Le filtre proposé est similaire au filtre bilatéral (Smith et Brady, 1997) qui calcule la profondeur en un pixel donné en fonction des profondeurs des points projetés dans un certain voisinage spatial, et qui ont une couleur voisine. Ce voisinage spatial se définit via une distance dans le plan image mais aussi sur l'axe des profondeurs z. En somme le filtre proposé est la combinaison d'un filtre spatial (sur des coordonnées 3D) et d'un filtre en intensité (sur les couleurs des images sources, en 1D pouvant se généraliser à plusieurs canaux). Les filtres bilatéraux sont connus pour leur préservation des contours des images (qui correspondent souvent à des discontinuités de profondeur), et leur robustesse face aux données aberrantes. En outre ils sont facilement implémentables, et leur simplicité en font des arguments de choix

Fig. 3.3 – *Résoudre le problème de sparsité du nuage de points. À droite la scène* skull *reconstruite grâce à PMVS (Furukawa et Ponce, 2010). La densité du nuage n'est pas, telle quelle, suffisante pour produire un rendu de qualité. (b)*

pour la propagation d'incertitude. Ils supportent une grande variété d'applications telles que le *Tone Mapping*, la super-résolution, la colorisation, ou des opérations de traitement d'images basées sur la profondeur présentées par Kopf *et al.* (2007).

On part d'un ensemble de points 3D $\mathbf{X}_i$ donné par une méthode de reconstruction telle que celle de Furukawa et Ponce (2010), et on cherche à faire le rendu de cartes de profondeur denses. Pour chaque point $\mathbf{x}$ d'une image donnée, nous estimons sa profondeur par une moyenne pondérée des échantillons de profondeur disponibles $z_i = z(\mathbf{x}_i)$ dans une certaine fenêtre W centrée sur $\mathbf{x}$:

$$z(\mathbf{x}) = \sum_{i \in W, z_i \neq \infty} a_i \, z_i. \tag{3.10}$$



(a) (b) (c) (d)

Fig. 3.4 – *Une carte de profondeur de la vue k est obtenue par application d'un filtre bilatéral sur le point projeté, avec un terme de visibilité qui favorise les points plus proches pour tenir compte des occultations. (a) Sans terme de visibilité. (b) Avec le terme de visibilité $h(z_i)$. (c) L'incertitude sur la profondeur $\sigma^2_{z,k}$, de certain (sombre) à incertain (clair). (d) La carte de profondeur de la vue cible estimée par projection et Z-test des cartes de profondeur des vues sources.*

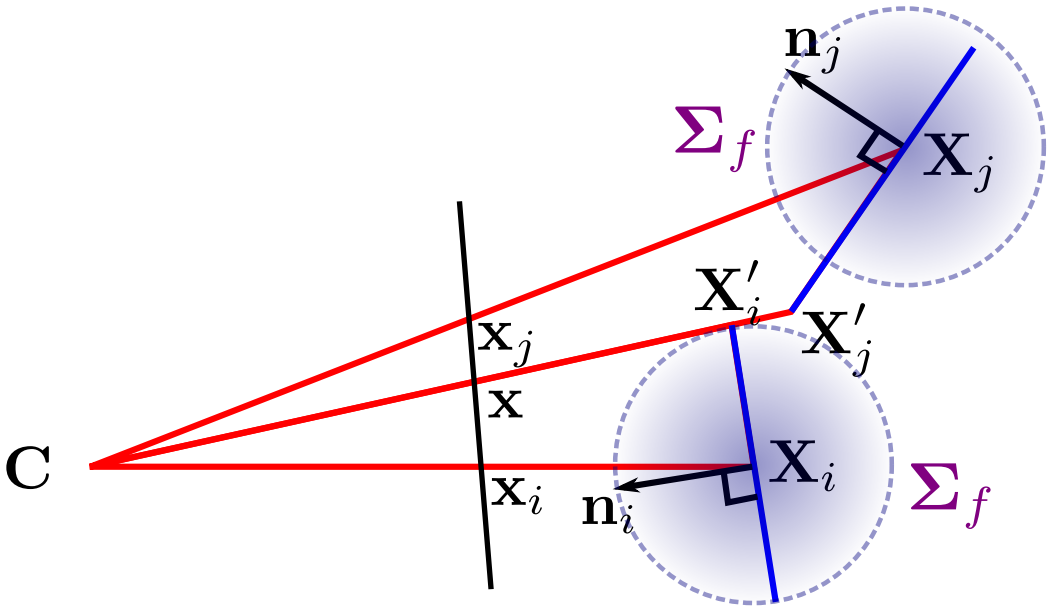


Fig. 3.5 – *Illustration du calcul du terme spatial f_s pour une fenêtre W centrée sur $\mathbf{x}$, contenant les points $\mathbf{x}_i$ et $\mathbf{x}_j$. Les profondeurs des points 3D $\mathbf{X}_i$ et $\mathbf{X}_j$ associés respectivement à $\mathbf{x}_i$ et $\mathbf{x}_j$ sont connues. Connaissant les normales $\mathbf{n}_i$ et $\mathbf{n}_j$, nous calculons les intersections $\mathbf{X}'_i$ et $\mathbf{X}'_j$ du rayon associé à $\mathbf{x}$ avec les surfaces élémentaires de $\mathbf{X}_i$ et $\mathbf{X}_j$. La distance inconnue de $\mathbf{x}$ reçoit les contributions de $\mathbf{X}'_i$ et $\mathbf{X}'_j$, dont le poids est fonction de $e^{-\frac{1}{2}(\mathbf{X}_i-\mathbf{X}'_i)^\intercal \boldsymbol{\Sigma}_f^{-1}(\mathbf{X}_i-\mathbf{X}'_i)}$. Nous avons dessiné les ellipsoïdes $\boldsymbol{\Sigma}_f$ en chaque point pour mettre en évidence le fait que dans ce cas, $\mathbf{X}_i$ contribue plus que $\mathbf{X}_j$.*

avec

$$a_i = \frac{1}{N(\mathbf{x})} f_s(||\mathbf{X}_i - \mathbf{X}'_i||)\, f_r(|v_k(\mathbf{x}_i) - v_k(\mathbf{x})|)\, f_z(z_i), \tag{3.11}$$

où $N(\mathbf{x})$ est un facteur de normalisation tel que $\sum_W a_i = 1$. Chaque coefficient a_i est le produit de trois termes :

- Le terme spatial f_s est une gaussienne d'un écart-type approximativement équivalent à la distance entre deux points 3D voisins dans la scène (figure 3.5). Pour chaque échantillon de profondeur z_i dans W, soit $\mathbf{X}_i$ son vecteur en coordonnées monde (WC, *World Coordinates* en anglais). $\mathbf{X}'_i$ est l'intersection entre le plan tangent en $\mathbf{X}_i$ avec le rayon optique issu du point $\mathbf{x}$. $||\mathbf{X}_i - \mathbf{X}'_i||$ est alors la distance euclidienne sur ce plan tangent. L'avantage d'utiliser ceci plutôt qu'une distance dans le plan image est de prendre en compte l'orientation locale de la surface. f_s a alors pour expression

$$f_s(||\mathbf{X}_i - \mathbf{X}'_i||) = e^{-\frac{1}{2}(\mathbf{X}_i-\mathbf{X}'_i)^\intercal \boldsymbol{\Sigma}_f^{-1}(\mathbf{X}_i-\mathbf{X}'_i)}, \tag{3.12}$$

  où la covariance 3D $\boldsymbol{\Sigma}_f$ contrôle la quantité de lissage.
- Le terme d'intensité f_r est une gaussienne qui mesure la probabilité que deux pixels de couleurs différentes appartiennent à la même surface. Une bonne estimation de l'écart-type σ_s est $\frac{1}{10}$ de l'intervalle d'intensités. f_r a alors pour

expression

$$f_r(|v_k(\mathbf{x}_i) - v_k(\mathbf{x})|) = e^{-(v_k(\mathbf{x}_i)-v_k(\mathbf{x}))^2/2\sigma_s^2}, \tag{3.13}$$

– Le terme de visibilité h est inspiré de Langer (2008) qui décrit la probabilité pour un point 3D d'être occulté par la géométrie de premier plan, suivant une loi de Poisson :

$$f_z(z) = \lambda e^{-\lambda(z-z_0)}. \tag{3.14}$$

z_0 est la profondeur du point le plus proche de la vue source, puisque la probabilité qu'un point soit visible à cette distance est de 1. Le paramètre λ dépend de la densité du nuage de points et de la direction des normales. Lorsque la densité du nuage de points augmente, la probabilité qu'un point soit occulté par d'autres points du nuage augmente également. De la même manière, les points de l'arrière-plan sont plus susceptibles d'être masqués si les normales des points du premier plan sont fronto-parallèles que si elles sont obliques. Plus de détails sur l'expression de λ sont donnés par Langer (2008). Les figures 3.4a et 3.4b illustrent l'effet de l'ajout du terme f_z.

Le résultat de ce filtre bilatéral modifié est une carte de profondeur dense de la vue k, et la variance associée. La fonction de déformation τ_k de la vue k vers la vue cible se calcule aisément à partir des cartes de profondeur obtenues (voir la section 3.3).

Propagation d'incertitude Le filtre proposé à l'avantage d'être simple, ce qui permet de calculer aisément l'incertitude sur la profondeur reconstruite. Considérant que les profondeurs voisines z_i sont des variables aléatoires normalement distribuées, nous sommons leur variance pour obtenir la précision de $z(m)$:

$$\text{L'équation (3.10) implique que} \quad \sigma_z^2 = \sum_{i \in W} a_i^2 \, \sigma_{z_i}^2 \tag{3.15}$$

$\sigma_{z_i}^2$ est la variance de la profondeur z_i issue d'un point 3D $\mathbf{X}_i$. On peut en trouver un estimateur de $\sigma_{z_i}^2$ et évaluer la variance σ_z^2 de la profondeur z au point $\mathbf{x}$ à partir de (3.10) :

$$\sigma_z^2 = \sum_{i \in W, z_i \neq \infty} a_i \, (z_i - z)^2 \, . \tag{3.16}$$

Nous n'utilisons pas cet estimateur car il est possible de calculer analytiquement la variance $\sigma_{z_i}^2$ en fonction des paramètres de caméras et du nuage de points orientés. Pour calculer la variance $\sigma_{z_i}^2$, nous procédons par propagation d'incertitude (figure 3.6). On suppose connue la matrice de covariance $\mathbf{\Sigma}_{\mathbf{X}_i\mathbf{X}_i}$ du point $\mathbf{X}_i$, qui peut être obtenue par la méthode de triangulation 3.1 par exemple. Il s'agit ici d'exprimer la profondeur z_i en fonction de $\mathbf{X}_i$, d'exprimer la jacobienne de cette fonction et d'en déduire l'expression de $\sigma_{z_i}^2$ en fonction de $\mathbf{\Sigma}_{\mathbf{X}_i\mathbf{X}_i}$.

Soit $\mathbf{r}_k$ le rayon passant par le centre optique $\mathbf{C}_k$ de la caméra k et le point image $\mathbf{x}$, on a $\mathbf{r}_k = \mathbf{K}_k^{-1}\bar{\mathbf{x}}$. $\mathbf{X}_i'$ étant la projection de $\mathbf{X}_i$ parallèlement à son plan tangent local sur le rayon $\mathbf{r}_k$, cela signifie deux choses : d'une part le vecteur $\mathbf{X}_i - \mathbf{X}_i'$ est orthogonal à la normale $\mathbf{n}_i$; d'autre part le point $\mathbf{X}_i'$ se situe sur le rayon $\mathbf{r}_k$. À partir de ces assertions on peut exprimer la profondeur z_i du point projeté par rapport

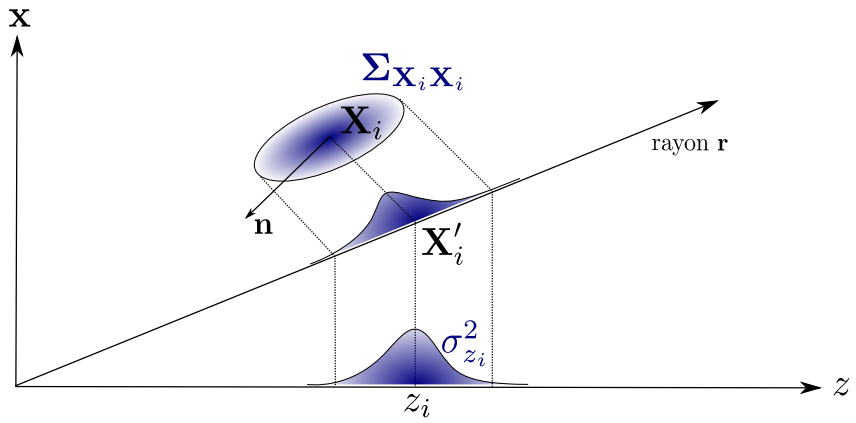


Fig. 3.6 – *Pour calculer la variance σ_{z_i} on projette les distributions des points 3D avoisinants, caractérisées par leur espérance $\mathbf{X}_i$ et leur matrice de covariance $\mathbf{\Sigma}_{\mathbf{X}_i\mathbf{X}_i}$, sur le rayon $\mathbf{r}$ parallèle à la normale $\mathbf{n}$. On obtient alors la distribution de covariance $\mathbf{\Sigma}_{\mathbf{X}'_i\mathbf{X}'_i}$, que l'on projette orthogonalement sur l'axe des profondeurs de la camera.*

aux paramètres de la caméra k :

$$z_i = \frac{\mathbf{n}_i^\intercal(\mathbf{X}_i - \mathbf{C}_k)}{\mathbf{n}_i^\intercal \mathbf{r}_k}. \tag{3.17}$$

La jacobienne $\mathbf{J}_{z_i}$ de la profondeur z_i est une matrice 1×3, puisque la distribution résultante est contrainte à un sous-espace de dimension 1 :

$$\mathbf{J}_{z_i}(\mathbf{X}_i) = \frac{\mathbf{n}_i^\intercal}{\mathbf{n}_i^\intercal \mathbf{r}_k}. \tag{3.18}$$

La formule de propagation d'erreur nous donne

$$\sigma^2_{z_i} = \frac{1}{(\mathbf{n}_i^\intercal \mathbf{r}_k)^2} \mathbf{n}_i^\intercal \mathbf{\Sigma}_{\mathbf{X}_i} \mathbf{n}_i \tag{3.19}$$

La figure 3.4c montre une carte d'incertitude. Les zones incertaines sont les zones occultations ou plus généralement les zones visibles par peu de caméras, derrière le crâne par exemple. Au contraire certaines parties telles que le sommet du crâne et le milieu de la table sont bien reconstruites, et ont donc une faible variance. La figure 3.4d illustre la carte de profondeur de la vue cible, servant à la gestion des auto-occultations (voir section 3.3).

3.3 Reconstruction de cartes de profondeur

Une autre grande famille d'algorithmes de MVS s'attelle à reconstruire la géométrie sous la forme de cartes de profondeur. Ces dernières peuvent être obtenues par rendu d'un nuage de points comme nous l'avons vu précédemment, mais aussi directement

à partir des images et des paramètres des caméras (obtenues par SfM). Parmi toutes ces méthodes on peut citer la stratégie « tout au vainqueur » (*Winner-takes-all* en anglais), les champs de Markov aléatoires (MRF, *Marvov Random Field* en anglais) ou encore l'algorithme du plan de balayage (*Sweeping plane algorithm* en anglais). Nous détaillons brièvement l'algorithme de Goesele *et al.* (2007), implémenté dans la chaine logicielle MVE (Fuhrmann *et al.*, 2014), qui utilise la stratégie « tout au vainqueur », car c'est l'algorithme que nous utilisons le plus souvent dans nos expériences de rendu basé image. Notez que, comme pour les nuages de points, la reconstruction de cartes de profondeur n'est pas systématiquement une fin en soi : on peut très bien s'en contenter pour faire du rendu basé image (ce qui est le cas dans la plupart de nos expériences), mais on peut aussi les fusionner pour obtenir un nuage de points ou un maillage (section 3.4).

Nous privilégions l'usage de cartes de profondeur car il offre de nombreux avantages en terme de souplesse de manipulation : il est aisé de traiter les cartes obtenues par filtrage, afin de les affiner ou de les densifier. Nous ajouterons que dans une approche *view-dependent* du rendu basé image qui est la notre, les cartes de profondeur sont idéales car elles procurent des informations de profondeur pour chaque pixel de chaque vue, ce qui rend le calcul des fonctions de déformation quasi-immédiat. Au contraire, un modèle unique commun à toutes les vues, tel un nuage de points ou un maillage, bien que ne présentant pas de redondance de données, est plus adapté à la synthèse de point de vue libre (*free viewpoint*). En outre, les algorithmes de reconstruction de cartes de profondeur sont simples et offrent un plus grand contrôle sur les propriétés de la reconstruction, en termes de *complétude*, *précision* ou *finesse* de la géométrie estimée. Ainsi ces algorithmes sont idéaux pour tester facilement de nouvelles méthodes de rendu sur des géométries à propriétés variables : on peut par exemple observer l'effet de la *complexité* d'un *proxy* sur les artefacts de rendu, ou évaluer la densité nécessaire à la prévention d'apparition de « trous » dans l'image synthétisée.

Stratégie « tout au vainqueur » robuste En chaque pixel $\mathbf{m}$ d'une vue de référence on souhaite estimer une profondeur radiale h et sa dérivée spatiale $h_{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}}$, où $\mathbf{x}$ désigne ses coordonnées (x, y) dans le plan image. Dans un premier temps, on cherchera une profondeur radiale, c'est-à-dire la distance euclidienne du centre optique de la caméra au point 3D $\mathbf{X}$, par opposition à la profondeur orthogonale z qui se mesure le long de l'axe optique. Nous passons de l'une à l'autre selon la conversion vue au chapitre 2. Le triplet de paramètres solutions (h, h_x, h_y) est celui qui maximise une mesure de photo-cohérence avec les autres images. Pour cela on crée d'abord une fenêtre centrée sur le pixel $\mathbf{m}$, de taille 7×7 par défaut. Cette fenêtre, projetée en 3D, forme un *patch* à la distance h et orienté par $h_{\mathbf{x}}$. On suppose que la taille de la fenêtre est petite devant la profondeur h, afin d'approximer la projection perspective en une projection orthogonale, ce qui donne lieu à un *patch* plan. Ce *patch* est ensuite projeté dans chaque vue source voisine afin de calculer une mesure de photo-cohérence. La fonction de photo-cohérence est en général NCC ou SSD. Dans le cas où SSD est prise par exemple (Goesele *et al.*, 2007), on obtient alors un problème des moindres carrés ; on résout alors un système d'équations que

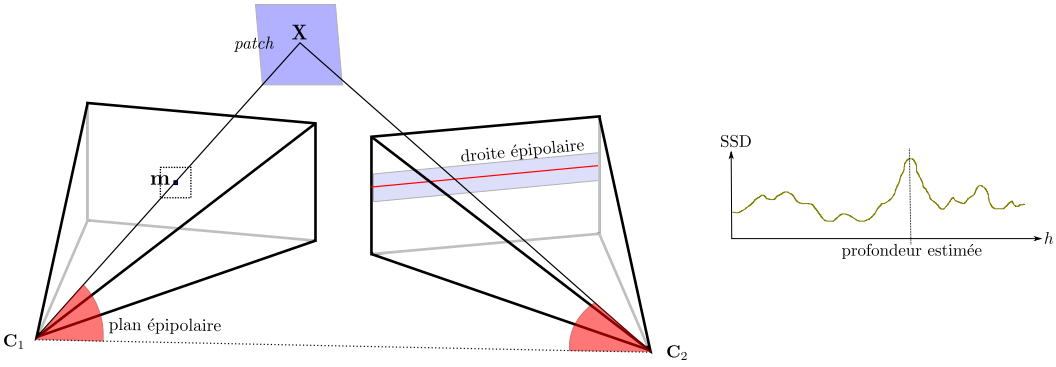


Fig. 3.7 – *Stratégie « tout au vainqueur ». A gauche on montre que pour chaque pixel* **m** *d'une vue de référence* $\mathbf{C}_1$ *dont on veut estimer profondeur radiale (distance au point* **X***) on crée une fenêtre (en pointillés) centrée sur* **m** *et le* patch *correspondant que l'on projette sur la ou les vues voisines (ici la vue* $\mathbf{C}_2$*). Quelle que soit la profondeur du* patch*, il est projeté sur la droite épipolaire, appartenant au plan épipolaire (*$\mathbf{C}_1\mathbf{C}_2\mathbf{X}$*) ; la recherche est donc contrainte à une seule dimension. Nous avons surligné en bleu clair la zone où le* patch *est projeté pour toutes les profondeurs possibles, où l'on calcule la mesure de photo-cohérence. A droite on représente une fonction de photo-cohérence (SSD par exemple) le long de la droite épipolaire. La profondeur qui l'emporte est celle qui maximise globalement la mesure de photo-cohérence.*

l'on peut linéariser.

Cette stratégie est dite « tout au vainqueur », car le triplet (h, h_x, h_y) associé à la plus forte mesure de photo-cohérence l'emporte sur les autres. La recherche de la mesure de photo-cohérence la plus élevée s'effectue le long des droites épipolaires, ce qui réduit le problème à une dimension (figure 3.7). On en déduit que la calibration des caméras est cruciale, car une erreur de quelques pixels dans l'estimation des droites épipolaires a de grandes conséquences sur la recherche de la profondeur. En général, les algorithmes de SfM dont on se sert pour calibrer les caméras ont une incertitude inférieure à un pixel. Une autre limite majeure de ces méthodes est que plusieurs mesures de photo-cohérence (plusieurs maxima locaux) peuvent convenir : on a donc plusieurs profondeurs candidates. Par conséquent on peut avoir une convergence plus lente, ou une convergence vers une mauvaise profondeur. La géométrie résultante est bruitée, avec de nombreux *outliers*. Goesele *et al.* (2007) proposent une technique plus robuste qui consiste à retirer pendant le processus d'optimisation les vues dont la mesure de photo-cohérence (NCC) est en dessous d'un certain seuil ρ. La méthode converge lorsque plus aucune vue n'est retirée et que la mesure de photo-cohérence n'évolue plus d'une itération à l'autre.

Cet algorithme de reconstruction offre beaucoup de contrôle sur les propriétés de la géométrie reconstruite. Si par exemple nous souhaitons éliminer les *outliers* il suffit d'augmenter le seuil ρ. Les cartes de profondeur estimées seront moins bruitées mais aussi moins denses, ce qui n'est pas un problème lorsque suffisamment de vues sont disponibles pour produire un rendu sans « trous ». Il est cependant inefficace d'augmenter la taille des fenêtres (et donc la taille des *patches*), car même si cela permet de réduire le nombre de maxima locaux de la fonction de photo-cohérence, le maximum résultant est moins marqué. La convergence est alors approximative, donnant lieu à plus d'imprécisions. Il est aussi possible de reconstruire avec des images

de plus faible résolution pour produire des cartes de profondeur plus grossières mais ayant moins de discontinuités. Il n'est pas inutile de passer du temps pour trouver un compromis entre ces différents paramètres influant sur la complétude, précision ou douceur du modèle dans le but d'optimiser la qualité de la reconstruction.

Profondeurs radiales et profondeurs orthogonales Pour chaque vue k, une carte de profondeur est estimée en utilisant l'algorithme de stéréo multi-vues (Goesele *et al.*, 2007). Les profondeurs obtenues h sont radiales – distances euclidiennes entre un point 3D de la scène et le centre de la caméra. Nous les convertissons en profondeurs orthogonales z :

$$z = \frac{h}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}\|}. \tag{3.20}$$

En effet d'après le modèle sténopé explicité à la section 2.4 du chapitre précédent, $\mathbf{X} = \mathbf{K}_k^{-1}\tilde{\mathbf{x}} =$. Ainsi en supposant que les distances sont en valeur absolue, on obtient $h = \|\mathbf{X}\| = z.\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}\|$ d'où l'égalité (3.20). De la même façon, la dérivée spatiale $h_{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}}$ qui donne l'orientation de la surface peut être convertie en

$$z_{\mathbf{x}} = \frac{\partial z}{\partial \mathbf{x}} = \frac{1}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}\|}(h_{\mathbf{x}} - h.\frac{(\mathbf{K}_k^{-1}\bar{\mathbf{x}})^\intercal(\mathbf{K}_k^{-1}[0]\,\mathbf{K}_k^{-1}[1])}{\|\mathbf{K}_k^{-1}\bar{\mathbf{x}}\|^2}) \tag{3.21}$$

où $\mathbf{K}_k^{-1}[0]$ et $\mathbf{K}_k^{-1}[1]$ représentent respectivement la première et la deuxième colonne de $\mathbf{K}_k^{-1}$.

Filtrage des cartes de profondeur Bien que facultative, l'étape de filtrage des cartes de profondeur estimées permet d'ajuster certaines propriétés de la reconstruction, de trouver un compromis entre *précision* et *complétude*, entre *complexité* et *douceur* du modèle. Comme il est écrit dans la documentation de MVE, « *It is rarely useful to reconstruct at full resolution as it will produce less complete depth maps with more noise at a highly increased processing time.* ». Une reconstruction avec des images de résolution originale est plus précise, mais le modèle est plus complexe, moins lisse et moins complet. Pour densifier les cartes, il est plus judicieux de reconstruire à résolution inférieure (images sources sous-échantillonnées), puis de sur-échantillonner les cartes de profondeur obtenues selon le filtre de Kopf *et al.*

Fig. 3.8 – *Rendu d'un* proxy *grossier par construction d'une pyramide d'images de hauteur 6. À gauche, réduction de la résolution. À droite, expansion de la résolution.*

(2007). MVE recommande justement de reconstruire à des échelles supérieures à 0 ; 2 est la valeur par défaut, ce qui correspond à un facteur $\frac{1}{4}$ sur les dimensions des images. Bien entendu on perd en précision (en terme de proximité avec les données), mais il est parfois plus judicieux d'utiliser un modèle plus simple, plus grossier, que de risquer à faire du sur-ajustement qui conduit à la perte de prédiction du modèle géométrique, une conséquence grave lorsque le but n'est pas l'obtention du modèle lui même mais son application au rendu basé image. Ce dilemme soulève des problèmes de sélection de modèle que nous aborderons dans le chapitre 6.

Pour obtenir des cartes de profondeur plus grossières (modèle géométrique plus simple), deux approches sont envisageables. La première consiste à reconstruire à partir d'images sous-échantillonnées (ou de manière équivalente à élargir la fenêtre de mesure de photo-cohérence). Mais à partir d'une certaine échelle (6 dans nos expériences) l'algorithme est incapable de produire des cartes de profondeur ; cette méthode n'est donc pas adaptée à la reconstruction à faible résolution. L'autre approche (figure 3.8) consiste à reconstruire à la résolution originale (scale 0) puis à filtrer successivement par une gaussienne ou un filtre l'approchant et sous-échantillonner les cartes par un facteur 2. On peut se passer de l'étape de sous-échantillonnage pour conserver des cartes de profondeur à la résolution originale, créant ainsi un espace d'échelle (*scale space* en anglais) plutôt qu'une pyramide.

Le filtre utilisé est le filtre `oddHDC` (Burt, 1981) avec des coefficients binomiaux. L'application répétée de ce filtre s'apparente à l'application d'un filtre gaussien. La spécificité du filtrage de cartes de profondeur est de n'utiliser que les pixels associés à des profondeurs reconstruites, c'est-à-dire d'ignorer les pixels marqués invalides. Un seuil de validité (profondeur très grande) est défini afin d'écarter les pixels non reconstruits. Une normalisation est nécessaire car l'ajout de ce test supplémentaire fait que le total des contributions n'atteint pas systématiquement la valeur 1. Une autre subtilité est la résolution du problème d'érosion ou de dilatation qui apparait aux frontière des zones reconstruites. En effet si par exemple un voisinage contient un seul pixel reconstruit, on va alors donner cette valeur de profondeur au pixel courant (alors qu'il est très probable que ce pixel appartienne à l'arrière-plan et doit par conséquent être laissé invalide car non reconstruit) : on obtient alors une dilatation des surfaces. Au contraire si on marque comme « invalides » tous les pixels dont le voisinage contient au moins un pixel invalide, on érode des surfaces. Un bon compromis est de fixer une valeur seuil de 0.5 pour les poids des contributions.

Cet algorithme de filtrage permet aisément d'ajuster le niveau de *complexité* du modèle géométrique. Il serait intéressant d'utiliser les techniques connues de sélection de modèle afin de trouver le niveau de complexité par partie de la scène qui optimise son rendu. Nous avons réalisé des expériences dans le chapitre 5 qui mettent en évidence de manière totalement empirique l'effet du niveau de complexité de la géométrie sur la qualité du rendu.

Calcul des fonctions de déformation Nous calculons les déformations d'images τ_k en chaque point $\mathbf{x_m}$ en projetant sur la vue cible chaque point 3D $\mathbf{X}$ estimé par la carte de profondeur de la vue source k. Le schéma suivant permet de se rendre compte des différentes opérations qui opèrent lors de la transformation d'un point

image d'une caméra à l'autre:

$$\tau_k : \mathbf{x_m} \underset{(a)}{\longrightarrow} \bar{\mathbf{x}}_\mathbf{m} \underset{(b)}{\longrightarrow} \tilde{\mathbf{x}}_\mathbf{m} \underset{(c)}{\longrightarrow} \mathbf{X} \underset{(d)}{\longrightarrow} \tilde{\mathbf{x}}_\mathbf{p} \underset{(e)}{\longrightarrow} \mathbf{x_p}. \tag{3.22}$$

(a) Le point image 2D $\mathbf{x_m}$ (en unités pixel) est étendu en $\bar{\mathbf{x}}_\mathbf{m}$ par ajout d'une troisième coordonnée valant 1 via la transformation en coordonnées homogènes N_h. (b) La multiplication par la distance orthogonale z vue de la caméra k nous donne le point homogène $\tilde{\mathbf{x}}_\mathbf{m}$. (c) Un changement de repère par le biais des matrices de la caméra k permet d'obtenir les coordonnées monde du point 3D $\mathbf{X} = \mathbf{R}_k^\intercal \mathbf{K}_k^{-1} \tilde{\mathbf{x}}_\mathbf{m} + \mathbf{C}_k$. (d) Puis le point homogène $\tilde{\mathbf{x}}_\mathbf{p}$ est obtenu par changement inverse dans le repère de la caméra ciblée $\tilde{\mathbf{x}}_\mathbf{p} = \mathbf{K}_u(\mathbf{R}_u\mathbf{X} + \mathbf{t}_u)$. (e) Enfin le point image $\mathbf{x_p}$ résultant (en unités pixels) est calculé par normalisation euclidienne $N_e(\tilde{\mathbf{x}}_\mathbf{p})$. Nous avons donc

$$\tau_k(\mathbf{x_m}) = N_e(\mathbf{K}_u(\mathbf{R}_u(z.\mathbf{R}_k^\intercal \mathbf{K}_k^{-1} N_h(\mathbf{x_m}) + \mathbf{C}_k) + \mathbf{t}_u)). \tag{3.23}$$

Gestion des occultations La gestion des occultations est une étape facultative de traitement des fonctions de déformation τ_k qui consiste à restreindre les domaines de départ Ω_k afin de les rendre injectives. En d'autres termes on définit seulement τ_k sur les points de Ω_k dont la projection se situe dans le champ de vue de la vue cible $\mathbf{C}_u$ et n'est pas occultée par une autre projection (auto-occultation). Si l'on restreint Γ à l'ensemble des points d'arrivée atteints, les fonctions de déformation sont alors bijectives. Si le rendu se fait par éclaboussures (chapitre 4), alors le calcul des occultations est déjà compris dans le processus de rendu. Sinon il constitue une étape préliminaire.

La gestion de la visibilité se fait en deux temps (figure 3.10). D'abord, les pixels des vues sources sont marqués invalides si leur projeté par τ_k se situe hors des bords de l'image de la vue cible. Ensuite, la carte de profondeur de la vue cible est estimée pour traiter les occultations inverses (figure 3.9). À partir de chaque pixel $\mathbf{m}$ de chaque vue source $\mathbf{C}_k$, un *quad* (quadrilatère 3D) dont la projection recouvre le pixel est créé à la distance estimée z et orienté par $z_\mathbf{x}$. Il est ensuite projeté sur la vue cible u en accumulant un *z-buffer* pour ne retenir que les profondeurs les plus proches. Le test de visibilité s'effectue en comparant la distance du point 3D reconstruit à la vue cible avec la profondeur estimée précédemment : si la différence se situe au-delà d'un certain seuil – fixé arbitrairement – alors le pixel $\mathbf{m}$ est marqué comme non visible depuis $\mathbf{C}_u$.

Dérivées et propagation d'incertitude Comme il est montré par Pujades *et al.* (2014), et plus tard dans ce manuscrit, on peut choisir des poids des contributions au rendu de telle sorte qu'ils s'obtiennent par propagation d'erreur. Sans entrer dans le détail de l'obtention de ces poids maintenant, nous expliquons comment obtenir les jacobiennes qui nous permettrons de les exprimer. Les poids des contributions de chaque vue source dans le rendu, tels que ceux proposés par Pujades *et al.* (2014), sont les *poids de résolution* et les *poids de déformation*.

Les *poids de déformation*, donnés par le jacobien $|\frac{\partial \tau_k}{\partial \mathbf{x}_m}|$ de la transformation τ_k, pénalisent les vues qui observent la surface de biais ou qui se situent loin de la vue

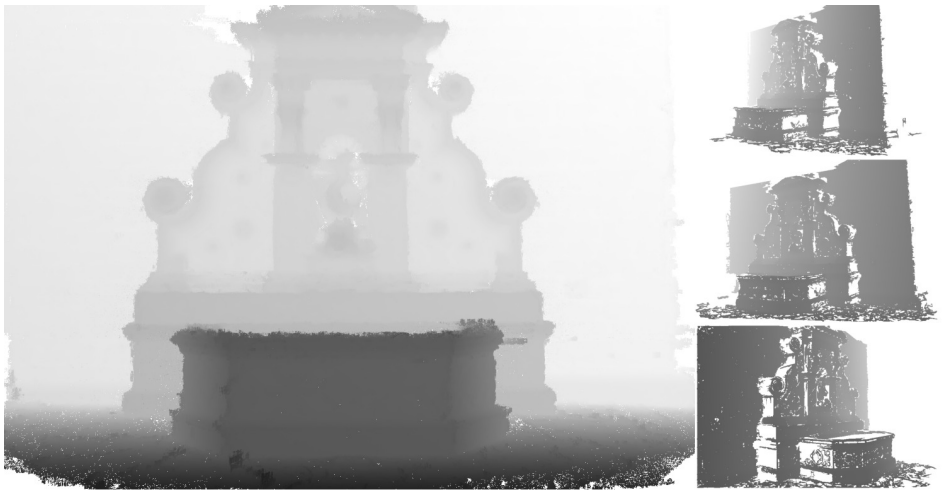


Fig. 3.9 – *À gauche : la carte de profondeur de la vue cible, obtenue par projection de* quads *depuis les vues sources. À droite : cartes de profondeur du jeu de données* fountain.

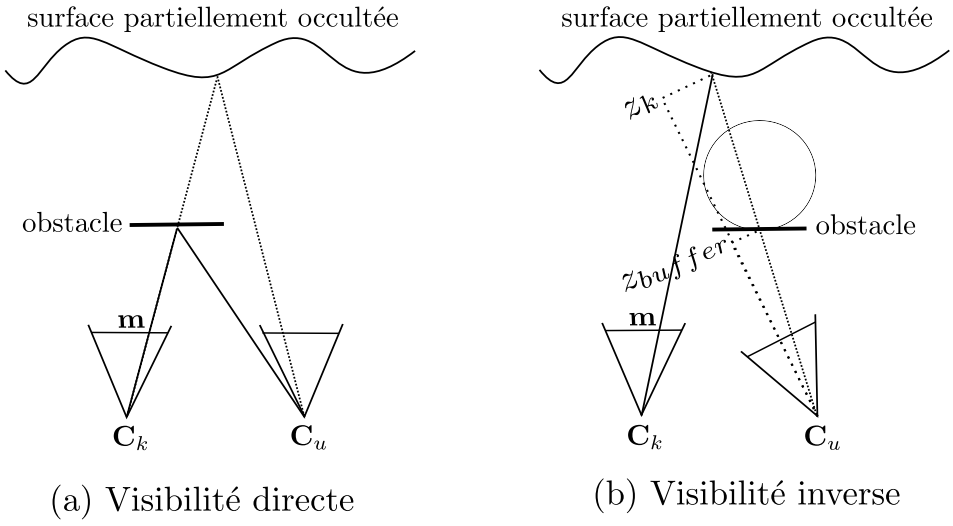


Fig. 3.10 – *Deux types d'occultations à traiter. (a) L'obstacle présent devant la surface en arrière-plan se situe devant la vue source* $\mathbf{C}_k$ *et le pixel* $\mathbf{m}$ *est projeté hors du champ de vue de* $\mathbf{C}_u$*. (b) Dans cette configuration l'obstacle occulte l'arrière-plan du point de vue cible* $\mathbf{C}_u$*. Nous estimons la carte de profondeur de* $\mathbf{C}_u$ *en accumulant les projections des profondeurs* z_k *dans un Z-buffer. Puis pour chaque vue* $\mathbf{C}_k$ *nous comparons la profondeur* z_k *estimée par MVS avec la profondeur stockée* z_{buffer}*. Si la différence est supérieure à un certain seuil il y a auto-occultation.*

cible. La jacobienne se calcule à partir des matrices des caméras, des profondeurs et des normales estimées, en dérivant en chaîne la fonction composée (3.23) par rapport au point image 2D $\mathbf{x}_m$. Dérivons alors chacune des transformations intermédiaires énoncées dans la chaîne (3.23). (e) $\frac{\partial \mathbf{x}_p}{\partial \mathbf{x}_m} = \mathbf{J}_e(\tilde{\mathbf{x}}_p)\frac{\partial \tilde{\mathbf{x}}_p}{\partial \mathbf{x}_m}$. (d) $\frac{\partial \tilde{\mathbf{x}}_p}{\partial \mathbf{x}_m} = \mathbf{K}_u\mathbf{R}_u\frac{\partial \mathbf{X}}{\partial \mathbf{x}_m}$. (c) $\frac{\partial \mathbf{X}}{\partial \mathbf{x}_m} = \mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\frac{\partial \tilde{\mathbf{x}}_m}{\partial \mathbf{x}_m}$. (b) $\frac{\partial \tilde{\mathbf{x}}_m}{\partial \mathbf{x}_m} = z.\frac{\partial \bar{\mathbf{x}}_m}{\partial \mathbf{x}_m} + \bar{\mathbf{x}}_m z_{\mathbf{x}}$. (a) $\frac{\partial \bar{\mathbf{x}}_m}{\partial \mathbf{x}_m} = \mathbf{J}_h$. Ces équations mises bout-à-bout, on obtient

$$\frac{\partial \tau_k}{\partial \mathbf{x}_m} = \mathbf{J}_e(\tilde{\mathbf{x}}_p)\mathbf{K}_u\mathbf{R}_u\mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\begin{pmatrix} z_x x_m + z & z_y x_m \\ z_x y_m & z_y y_m + z \\ z_x & z_y \end{pmatrix}. \tag{3.24}$$

Les *poids de géométrie*, garantissant la *déviation angulaire minimale*, c'est-à-dire pénalisant les caméras formant un angle trop grand avec la vue cible, découlent de l'incertitude de la géométrie estimée $\sigma_{g,k}^2$ et de la variance du bruit du capteur $\sigma_{s,k}^2$. Chaque vue source k est ainsi pondérée par $\omega_k(u) = (\sigma_{s,k}^2 + \sigma_{g,k}^2(u))^{-1}$ avec $\sigma_{g,k}^2(u) = \sigma_{z,k}^2(b * (\frac{\partial \tau_k}{\partial z}^\intercal \nabla u \circ \tau_k))^2$, et ∇u le gradient de la solution courante u. De la même façon que pour les poids de résolution, la dérivé $\frac{\partial \tau_k}{\partial z}$ s'obtient par composition des dérivées de la chaîne (3.23). (e) $\frac{\partial \mathbf{x}_p}{\partial z} = \mathbf{J}_e(\tilde{\mathbf{x}}_p)\frac{\partial \tilde{\mathbf{x}}_p}{\partial z}$. (d) $\frac{\partial \tilde{\mathbf{x}}_p}{\partial z} = \mathbf{K}_u\mathbf{R}_u\frac{\partial \mathbf{X}}{\partial \mathbf{x}_m}$. (c) $\frac{\partial \mathbf{X}}{\partial \mathbf{x}_m} = \mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\frac{\partial \tilde{\mathbf{x}}_m}{\partial z}$. (b) $\frac{\partial \tilde{\mathbf{x}}_m}{\partial z} = z.\frac{\partial \bar{\mathbf{x}}_m}{\partial z} + \bar{\mathbf{x}}_m$. (a) $\frac{\partial \bar{\mathbf{x}}_m}{\partial z} = \mathbf{0}_2$. Enfin, mis bout-à-bout :

$$\frac{\partial \tau_k}{\partial z} = \mathbf{J}_e(\tilde{\mathbf{x}}_p)\mathbf{K}_u\mathbf{R}_u\mathbf{R}_k^\intercal\mathbf{K}_k^{-1}\bar{\mathbf{x}}_m. \tag{3.25}$$

3.4 Reconstruction de la surface

À l'issu d'une reconstruction MVS, le format du *proxy géométrique* obtenu est généralement un nuage de points ou un ensemble de cartes de profondeur. Il n'est pas nécessaire d'aller plus loin que cette étape pour la synthèse d'image, puisque le rendu basé point est adapté aux applications de *free viewpoint* et les cartes de profondeur à la synthèse de texture *view-dependent*. Cependant on préfère souvent le format maillage au nuage de points, car il permet de densifier un nuage de points trop épars, de combler les éventuels trous, de simplifier le modèle retirant les points superflus appartenant à la même surface, ou encore de l'affiner. En outre, un maillage apporte moins de redondance qu'un ensemble de cartes de profondeur et se manipule plus facilement. Ici nous abordons quelques méthodes classiques de fusion de données volumétriques particulièrement bien adaptées aux méthodes MVS décrites précédemment. Les techniques d'affinage des maillages se situent hors de la portée de ce manuscrit.

Reconstruction dite « de Poisson » à partir d'un nuage de points Le travail de Kazhdan (2005); Kazhdan *et al.* (2006); Kazhdan et Hoppe (2013) propose de rompre avec la triangulation de Delaunay, chronophage, sensible au bruit et dépendante d'un échantillonnage de points uniforme. L'idée clé de son approche est de reconstruire une fonction implicite, aussi appelée fonction indicatrice ou fonction

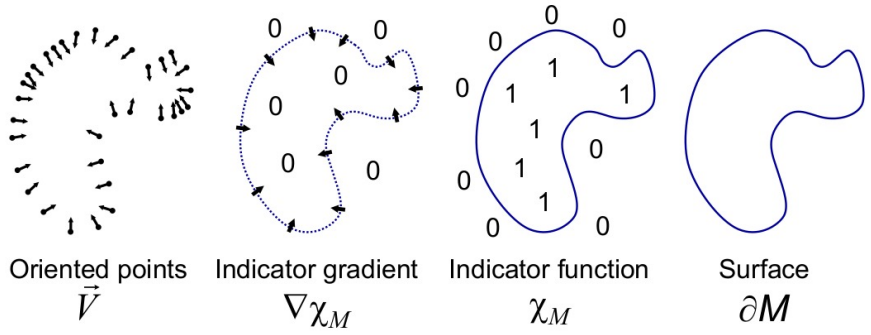


Fig. 3.11 – *Figure de Kazhdan et al. (2006). Un champ de gradient $\vec{V}$ est défini par le nuage de points orientés. On cherche à estimer la fonction indicatrice $\tilde{\chi}$ dont le gradient approche $\vec{V}$, au sens des moindres carrés, ce qui revient à résoudre une équation de Poisson. La surface reconstruite ∂M est obtenue par* Marching Cubes.

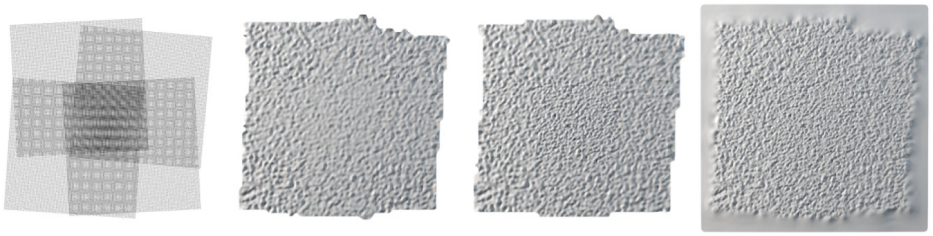


Fig. 3.12 – *Figure de Fuhrmann et Goesele (2014). Quatre échantillonnages bruités se recoupant au milieu. La partie centrale, du fait de la redondance de données non alignées, a une densité plus forte mais sa résolution (la* scale*) est la même qu'ailleurs. Se baser sur la densité de l'échantillonnage pour reconstruire la partie centrale peut provoquer l'apparition d'artefacts à haute fréquence (milieu droit). Au contraire un bon usage de la résolution réduit le bruit dans les parties redondantes. À droite, le résultat d'une reconstruction de Poisson, contenant du bruit à haute fréquence.*

caractéristique, et d'en extraire un isocontour. Un fonction indicatrice renvoie 1 à l'intérieur du modèle, et 0 à l'extérieur, de telle sorte que son gradient est approché par le champ $\vec{V}$, défini par les échantillons de points orientés à la surface du modèle à reconstruire. Le maillage est obtenu en effectuant un *Marching Cubes* (Lorensen et Cline, 1987) sur l'isocontour de la fonction indicatrice. La méthode propre à Kazhdan (2005) pour reconstruire la fonction indicatrice à partir d'un nuage de points (orientés) est de calculer ses coefficients de Fourier. Comme cela suppose de connaître la fonction indicatrice afin d'en calculer une intégrale volumique, ils exploitent le théorème de Gauss, qui transforme cette intégrale volumique en une intégrale surfacique, qui peut être estimée grâce à l'échantillon de points et leurs normales. Il a été prouvé par Kazhdan *et al.* (2006) que trouver une estimation $\tilde{\chi}$ de la fonction indicatrice par le théorème de Gauss est équivalent à résoudre l'équation de Poisson $\Delta\tilde{\chi} = \nabla \cdot \vec{V}$. L'estimateur cherché doit minimiser une fonction de coût. L'ajout d'un terme d'attache aux données (Kazhdan et Hoppe, 2013) à cette fonction de coût permet de contrebalancer l'effet de lissage de la surface, « forçant » la surface à être proche du nuage de points fourni en entrée.

Reconstruction à partir d'un ensemble de cartes de profondeur PSF (*Poisson Surface Reconstruction*) (Kazhdan *et al.*, 2006) peut servir à traiter des cartes de profondeur, une fois que celles-ci ont été triangulées. Cependant une limitation de PSF réside dans sa faible capacité à traiter des données de résolutions différentes. Si les cartes de profondeur de faible résolution sont de densité inférieure, alors elles ont peu de poids dans la reconstruction de la surface, comparées aux cartes de résolution supérieure. Mais si l'échantillonnage des cartes de faible résolution, naturellement plus bruitées du fait de la plus grande incertitude de localisation qui les caractérise, est plus élévé à cause d'une redondance de données mal alignées par exemple, alors l'impact sur la reconstruction de la surface est très négatif (Mücke *et al.*, 2011). Plus récemment, Fuhrmann et Goesele (2011) ont proposé un algorithme pour fusionner des cartes de profondeur de résolution différente, s'intégrant dans la *pipeline* de reconstruction initiée par Goesele *et al.* (2007). Il s'appuie sur la construction d'une hiérarchie de SDF (*Signed Distance Field* en anglais) qui représente des surfaces à plusieurs niveaux de détails. L'information de profondeur est ensuite propagée, des SDF aux niveaux les plus grossiers (faible résolution) aux SDF des niveaux les plus fins (haute résolution), en filtrant les valeurs qui n'atteignent pas un seuil de confiance afin d'éviter un lissage excessif par les faibles résolutions. La surface est enfin obtenue par l'extraction d'un isocontour basé sur une tétraédrisation de Delaunay. Toujours dans le but de fusionner des données à différentes résolutions, Mücke *et al.* (2011) partitionnent l'espace contenant les échantillons de points en volumes appelés *crusts*, suivant la résolution de l'échantillonnage. À l'intérieur d'un *crust*, chaque échantillon ajoute une valeur de confiance à une région du volume, résultant en une carte de confiance, utilisée pour recréer une surface en basse résolution par une méthode de *graph-cut*. Les partie en haute résolution sont successivement reconstruites en construisant des *crusts* plus fins dans les endroits du volumes les plus densément échantillonnés. Plus récemment, Fuhrmann et Goesele (2014) ont proposé une base de fonctions pour reconstruire la fonction implicite qui permet de prendre en compte les changements continus de résolution, améliorant nettement les résultats de Fuhrmann et Goesele (2011) et Mücke *et al.* (2011). Sur la figure 3.13, nous comparons deux *pipelines* classiques, l'une qui utilise successivement PMVS (Furukawa et Ponce, 2010) et PSR (Kazhdan *et al.*, 2006), et l'autre qui fusionne des cartes de profondeur obtenues par Goesele *et al.* (2007) avant de reconstruire un maillage avec FSSR (Fuhrmann et Goesele, 2014).

Limites et solutions La reconstruction de Poisson (Kazhdan *et al.*, 2006) présente l'inconvénient de lisser la surface, qui perd ainsi les détails au profit d'une diminution du bruit en haute fréquence. L'ajout d'un terme d'attache aux données introduit par Kazhdan et Hoppe (2013) permet de résoudre ce problème, mais d'autres inconvénients de leur approche persistent. D'une part la reconstruction de Poisson, et plus largement les algorithmes de reconstruction de surface, sont performants dans des applications de remplissage de trous, mais ce même avantage devient un inconvénient lorsque l'on souhaite reconstruire des scènes extérieures. En effet l'algorithme hallucine souvent des surfaces qui n'existent pas, ce qui impacte gravement le rendu basé image (figure 3.14d). D'autre part, la finesse de reconstruction est pro-

portionnelle à l'échantillonnage, ce qui altère la reconstruction lorsque des données faiblement résolues se superposent sans s'aligner. Le bruit inhérent à ces reconstructions de faible résolution peut être perçu comme du détail à reconstruire, créant ainsi des artefacts haute fréquence. De la même façon, si un objet très détaillé est capturé par de nombreuses vues redondantes faiblement résolues, et par quelques vues hautement résolues, les vues faiblement résolues l'emporteront à cause de la densité élevé de l'échantillonnage redondant, ce qui lissera le détail de la surface. Ces problèmes sont résolus par les algorithmes de fusion multi-résolution, en particulier celui de Fuhrmann et Goesele (2014). Mais une bonne méthode de reconstruction de surface ne suffit pas à l'obtention d'un *proxy géométrique* parfait, car la qualité de la reconstruction repose beaucoup sur celle du nuage de points. Afin de résoudre ce problème, Wolff *et al.* (2016) proposent d'intégrer un filtrage des points en amont afin de le pas pénaliser la reconstruction de la surface.

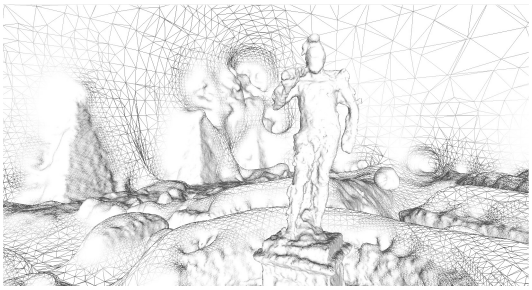
3.5 Choix de la méthode : une affaire de compromis

La diversité des méthodes MVS existantes rend difficile le choix de l'une d'entre elles pour des applications d'IBR. Chacune propose son lot d'avantages et d'inconvénients en termes de *stabilité*, *densité* ou *précision*, qui ne lèvent pas simultanément tous les problèmes d'artefacts qui peuvent impacter le rendu. À ces méthodes s'ajoutent des algorithmes d'affinage, tels que le filtre bilatéral de Kopf *et al.* (2007) pour les *proxys géométriques* sous forme de cartes de profondeur, ou des lissages de surface de type laplacien pour les maillages, qui permettent de céder de la précision au profit de la stabilité de la reconstruction 3D, ou d'effacer les discontinuités mais au prix d'une augmentation de l'incertitude de localisation. Le choix de la méthode est une affaire de compromis entre ces différents concepts.

Stabilité vs. précision Prenons le cas où le lissage est poussé à l'extrême de telle manière que la géométrie est réduite à un plan de profondeur constante (ou fronto-parallèle), suite à un lissage répété par une méthode itérative, ou à un paramètre σ de Kopf *et al.* (2007) très grand par exemple. Ce plan est le plan moyen de la géométrie, appelé plan focal dans le cas applicatif de la refocalisation. Alors l'image synthétisée à partir de ce *proxy* comporte de nombreux effets fantôme ou de flou selon le nombre de points de vue utilisés et leur proximité avec le point de vue cible, car l'erreur d'alignement entre les images est très grand. Seuls les points de la scène se situant exactement sur le plan focal estimé apparaîtront nets. A l'instar de ce cas extrême, tout *proxy* lisse, obtenu par une reconstruction 3D de faible résolution conduit à ce genre d'artefact visuel lors du rendu. Il s'agit de reconstructions 3D *stables* car l'ajout de points supplémentaires changerait peu le *proxy* reconstruit, mais peu *précises* car la faible résolution augmente l'incertitude de reconstruction.

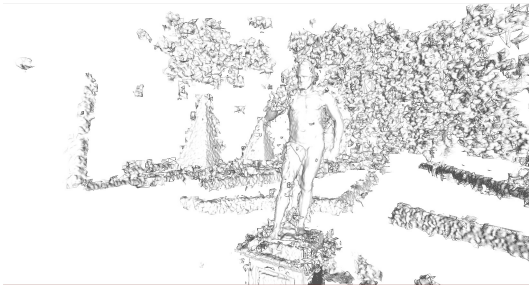
Au contraire si l'estimation de la géométrie est très *précise*, par reconstruction à haute résolution de cartes de profondeur de Goesele *et al.* (2007) par exemple, les discontinuités précédemment évoquées apparaissent, se répercutant aussi dans le rendu. Par *résolution* nous n'entendons pas *densité* de l'échantillonnage, mais

(a) PMVS (Furukawa et Ponce, 2010)



(b) PSR (Kazhdan *et al.*, 2006)

(c) (Goesele *et al.*, 2007)



(d) FSSR (Fuhrmann et Goesele, 2014)

(e) Nettoyage du maillage

Fig. 3.13 – *(a) Reconstruction d'un nuage de points par PMVS (Furukawa et Ponce, 2010). Pour des applications de rendu basé image, il est nécessaire de densifier l'information de profondeur par notre technique de rendu basé point présentée à la section 3.2. (b)* Poisson Surface Reconstruction *(Kazhdan* et al.*, 2006) produit un maillage. Notez qu'il est rarement utilisé pour des scènes en extérieur car il hallucine de la géométrie (blobs, remplissage de trous, dôme autour de la scène). (c) Nous fusionnons les cartes de profondeur obtenues par Goesele* et al. *(2007) en nuage de points dense et précis. (d) Éventuellement on peut produire un maillage de meilleure qualité avec FSSR (Fuhrmann et Goesele, 2014) plutôt qu'avec PSR. (e) On filtre le maillage en enlevant les triangles suspects.*

plutôt l'échelle à laquelle la géométrie est reconstruite, c'est à la taille de *patch* de la méthode de Goesele *et al.* (2007), ou l'échelle d'images sources utilisées pour la reconstruction 3D. Dans cette perspective, Kazhdan et Hoppe (2013) introduisent un nouveau terme permettant à la solution de mieux s'ajuster aux données. Si ce terme est trop pondéré, la géométrie a un aspect « bruité » : de nombreux artefacts haute fréquence apparaissent. Elle est plus fidèle au modèle original mais les fonctions de déformation résultantes présentent de grosses discontinuités. En outre ces discontinuités vont même compromettre le calculs des poids des contributions de chaque point de vue dans le rendu de la vue cible (Nieto *et al.*, 2016a).

La recherche d'un compromis entre les deux approches, l'une proche des données, l'autre plus lisse dans un souci de stabilité et de continuité du *proxy*, est une question épineuse dans le domaine de la reconstruction de surface et de la reconstruction multi-vues stéréo. Nous avons opté pour une estimation de cartes de profondeur par la méthode de Goesele *et al.* (2007), en haute résolution, c'est-à-dire avec une taille de *patch* réduite (7×7) et des images sources de même taille que les images originales. Ainsi nous privilégions une reconstruction *précise*, c'est-à-dire une faible incertitude sur la profondeur estimée, mais d'éventuelles discontinuités qui viennent polluer le rendu. La figure 3.14 illustre les raisons de notre choix d'algorithme MVS. Nous verrons dans les chapitres 4 et 5 comment résoudre le problème des discontinuités engendrées par l'imperfection du *proxy géométrique*.

3.6 Conclusion

Dans ce chapitre nous avons brièvement présenté les principales techniques de MVS qui permettent d'estimer un *proxy géométrique* servant au rendu basé image. Les différents formats de *proxy* ont chacun leur avantages et inconvénients, en termes de *simplicité*, *précision*, *densité*, etc. La recherche du « meilleur *proxy* » est vaine car son choix dépend de son application (*free viewpoint* ou *view-dependent synthesis*), et aucun ne permet de corriger conjointement les artefacts dus à l'incertitude de localisation et au bruit haute fréquence. Il s'agit donc d'une affaire de compromis entre plusieurs méthodes agrégées dont le choix dépend du type de rendu que nous souhaitons. Dans les expériences des chapitres 4 et 5 nous choisissons une *pipeline* classique d'estimation des cartes de profondeur en haute résolution, filtrage, et calcul des fonctions de déformation et de leurs dérivées partielles. Si cette méthode crée un *proxy* bruité, elle permet en revanche une estimation simple de l'incertitude de profondeur par pixel et un calcul des fonctions de déformation quasi-immédiat. En outre la précision des cartes de profondeurs estimées en fait un *proxy* adapté à notre application de synthèse de vue.

Ces méthodes de reconstruction de la géométrie posent néanmoins des problèmes susceptibles d'affecter le rendu d'autres manières dans certaines scènes bien particulières. En effet elles reposent toutes sur deux hypothèses qui sont souvent remises en cause : la scène est supposée lambertienne et les contraintes épipolaires sont respectées. Les surfaces spéculaires réfléchissent la lumière différemment selon la direction d'observation ; ainsi la radiance de l'objet observé varie d'une vue à l'autre, ce qui perturbe grandement la mise en correspondance. D'autre part les rayons issus

(a) Image source (b) Image cible

(c) (Goesele *et al.*, 2007) (d) PSR (Kazhdan *et al.*, 2006)

Fig. 3.14 – *(a) Image (source) 0 du jeu de données* hercule. *(b) Image (cible) 2. L'image 0 est projetée sur la vue 2, au moyen de fonctions de déformation calculée à partir de cartes de profondeur (Goesele* et al.*, 2007) (c), ou à partir d'un maillage obtenu par PSR (Kazhdan* et al.*, 2006). La reconstruction d'un maillage ajoute de l'information de profondeur qui fausse le rendu (autour de la silhouette d'Hercule). Nous préférons l'utilisation de cartes de profondeur, même éparses, car elles sont plus précises. La sparsité est compensée par un nombre important de vues.*

d'une même surface ponctuelle qui traversent des milieux transparents ou d'indice de réfraction variables sont déviés, ce qui fausse la recherche 1D sur les droites épipolaires. Se pose aussi la question de la performance : est-il nécessaire de reconstruire la géométrie de la scène de manière explicite, alors que notre objectif se borne à la synthèse d'un point de vue inédit ? Cette question est d'autant plus légitime que la reconstruction 3D est faussé par de mauvaises hypothèses, entrainant l'apparition d'artefacts spécifiques. Nous proposons une approche originale du rendu basé image dans le chapitre 6 qui tente de résoudre ce problème.

4 Rendu direct

4.1 Introduction

Les méthodes directes de rendu basé image consistent à générer une image en combinant les images sources. Chaque pixel de l'image obtenue est une moyenne pondérée des pixels des images sources, dont les poids dépendent de la position du point de vue à synthétiser vis-à-vis des vues sources, mais aussi d'autres heuristiques introduites par Buehler *et al.* (2001). Trois approches sont possibles pour synthétiser une image (Shum *et al.*, 2008) : la projection directe (*forward warping*), la projection rétrograde (*backward warping*) et l'approche hybride, qui ne sera pas présentée ici.

Projection directe et rétrograde La projection rétrograde consiste à peindre un point image de la vue cible en allant chercher le point image correspondant dans la vue source. Cette projection suppose l'utilisation de fonction de déformation inverse (*backward warp*). Ainsi, tous les pixels de la vue cible sont parcourus, et il suffit d'interpoler entre les valeurs des pixels des images sources, par exemple par interpolation bilinéaire ou bicubique. L'approche naïve consiste à interpoler en prenant la couleur du pixel de texture le plus proche dans l'image source, ce qui peut donner lieu à du crénelage (*aliasing*) dans les zones de dilatation de la déformation inverse. Nous préconisons donc un ré-échantillonnage (Heckbert, 1989) pour s'adapter au changement de fréquence résultant de la déformation d'une image à l'autre. La projection directe (*forward warping*) consiste à projeter les pixels des images sources sur l'image cible en accumulant les contributions puis en normalisant. Dans la section 4.2 nous décriront une méthode de rendu par projection directe, inspirée de Westover (1990), qui utilise le concept de *footprint*, ou *splat*, ou encore éclaboussure, qui prévient du crénelage en ré-échantillonnant. Nous estimons une surface locale en 3D pour chaque pixel des vues sources, que l'on déforme par la fonction de déformation directe τ_k et projette sur la vue cible. L'image est rendue par accumulation des contributions des *splats* et normalisation.

Panorama des méthodes de rendu basé image directes Shum *et al.* (2008) font la synthèse de la plupart des méthodes de rendu basé image ; parmi elles des méthodes qui utilisent un *proxy géométrique* grossier (Buehler *et al.*, 2001) ou pas de *proxy* du tout (Gortler *et al.*, 1996), où il est proposé d'échantillonner le champ lumineux dans une grille 4D (appelé *lumigraph* dans l'article) pour directement synthétiser une nouvelle vue par interpolation de rayons dans cette grille. Les méthodes de rendu basé image directes sont rapides et donc idéales pour les applications interactives comme la vidéo en point de vue libre (Zitnick *et al.*, 2004; Lipski *et al.*, 2010; Gurdan *et al.*, 2014; Lipski *et al.*, 2014) qui interpolent à la fois l'espace et le temps. Les *proxys* sont en général des ensembles de cartes de profondeur, très précises pour éviter les artefacts de rendu gênants dans les vidéos, ou un assemblage hybride entre cartes et correspondances denses entre vues sources (Lipski *et al.*, 2014). Le photo-tourisme est aussi un exemple d'application très en vogue. Dans cette optique Kushal *et al.* (2012), en utilisant un *proxy* estimé par Goesele *et al.* (2007), génèrent automatiquement un film suivant un chemin de caméras interpolé entre les points de vue capturés. Snavely *et al.* (2006b) misent sur l'intéractivité, avec l'objectif de parcourir une galerie d'images, les images de transition étant générées par projection via un maillage, rendues séparément et mélangées. Les poids de mélanges idéaux pour l'interpolation d'image sont développés par Takahashi (2010) qui se basent sur l'incertitude sur le *proxy* reconstruit. Certaines techniques d'interpolation d'images (Mahajan *et al.*, 2009; Linz *et al.*, 2010) se passent de *proxy* pour ne travailler que dans le domaine image. Comme Mahajan *et al.* (2009), Kopf *et al.* (2013) effectuent un rendu direct de gradient avant d'intégrer la solution pour retrouver la couleur, mais dans un contexte de génération de point de vue libre (et pas seulement une interpolation de point de vue). Il s'inspire de Sinha *et al.* (2012) dont l'objectif est de gérer les phénomènes de réflexion dans les scènes contenant des reflets et des transparences. Enfin on citera les récents travaux de Chaurasia *et al.* (2013) et Ortiz-Cayon *et al.* (2015) basés sur la sur-segmentation des images sources. Les premiers proposent de générer des profondeurs plausibles mais non nécessairement photocohérentes dans les zones non reconstruites, pour ensuite projeter chaque super-pixel indépendamment par une déformation préservant sa forme. Les derniers estiment la qualité de plusieurs algorithmes de rendu basé image afin de choisir le meilleur pour chaque super-pixel.

Dans ce chapitre nous commençons par détailler notre méthode de rendu basé image directe par fusion des images sources en intensité, puis nous présenterons une méthode de rendu multi-résolution. Nous aborderons le problème de la déformation d'espace d'échelle et de laplacien, les difficultés que cela engendre et les solutions proposées.

4.2 Fusion d'images en intensité

L'approche la plus classique de rendu basé image est le mélange en intensité. Autrement dit une fois les correspondances établies entre les images sources et la destination, on synthétise l'image cible en combinant les couleurs des images sources. Les informations de profondeur étant contenues dans les vues sources, il est peu ef-

ficace de synthétiser l'image cible par projection inverse (*backward warping*) étant donné que nous avons choisi un *proxy* constitué d'un ensemble de cartes de profondeur qui nous donne quasi-immédiatement l'expression des fonctions de déformation (directes) τ_k. La projection inverse reviendrait en effet à faire du lancer de rayons de la vue cible aux vues sources, alors qu'on ne connait pas la profondeur des pixels de la vue cible. Par conséquent, on procède par projection directe (*forward warping*), c'est-à-dire par projection des points des images sources vers l'image cible. Nous abordons dans un premier temps les problèmes de ré-échantillonnage que cela implique (section 4.2.1), puis la création de *splat*, objet géométrique approximant localement la surface (section 4.2.2), avant de détailler le processus de rendu par éclaboussures (*splatting*) (section 4.2.3). En fin nous traiterons des poids des contributions dans le mélange des images sources (section 4.2.4), puis de notre technique d'*inpainting* pour remplir les trous (section 4.2.5).

4.2.1 Le modèle de formation de l'image

Position du problème Un signal discret est l'échantillonnage de la fonction continue de notre image. On retrouve le signal continu par sa convolution avec la fonction d'étalement du point (PSF, *Point Spread Function* en anglais), qui peut être gaussienne ou rectangulaire par exemple (figure 4.1). Le cas continu ne pose pas de problème lors de la transformation par la fonction de déformation (*warp*) (figure 4.1). Mais lorsque les signaux sont échantillonnés, la projection d'un échantillon de l'image source peut atterrir entre deux points d'échantillonnages du signal d'arrivée (entre deux pixels). Les valeurs discrètes du signal d'arrivée doivent alors être ré-échantillonnées à partir de ces projections pour reconstruire une image constituée de pixels. La jacobienne de la déformation $\mathbf{J}$ joue un rôle crucial dans le ré-échantillonnage, car elle nous informe sur les parties du signal qui ont été dilatées ou comprimées. Les dilatations sont les zones où le jacobien $|\mathbf{J}|$, déterminant de la jacobienne $\mathbf{J}$, est > 1, les compressions celles où $|\mathbf{J}| < 1$. Lors d'une dilatation, l'information contenue dans le signal source est plus dense que dans le signal d'arrivée, la projection uniforme des pixels sur la vue cible (même échantillonnage à l'arrivée qu'au départ) donnera lieu à du crénelage. Il est nécessaire d'adapter la grille d'échantillonnage de départ à celle d'arrivée vie le jacobien $|\mathbf{J}|$ de la déformation. De même pour les zones de compression où une projection uniforme a pour effet de générer une image floue.

(Fuhrmann et Goesele, 2014) observent que « *Any sampled point acquired from a real-world geometric object or scene represents a finite surface area and not just a single surface point. Samples therefore have an inherent scale, very valuable information that has been crucial for high quality reconstructions* ». Ce qui est vrai pour les surfaces dans l'espace l'est aussi pour les signaux images. La reconstruction d'une image de haute qualité passe nécessairement par un ré-échantillonnage pertinent. Cette conclusion est basée sur l'observation du fait qu'un détail de l'image ne représente pas un point 3D infinitésimal de la scène, mais un élément de surface locale d'aire finie appelé *splat*, *footprint* ou éclaboussure. On doit tenir compte du fait que chaque vue ne capture pas avec la même résolution cette surface lo-

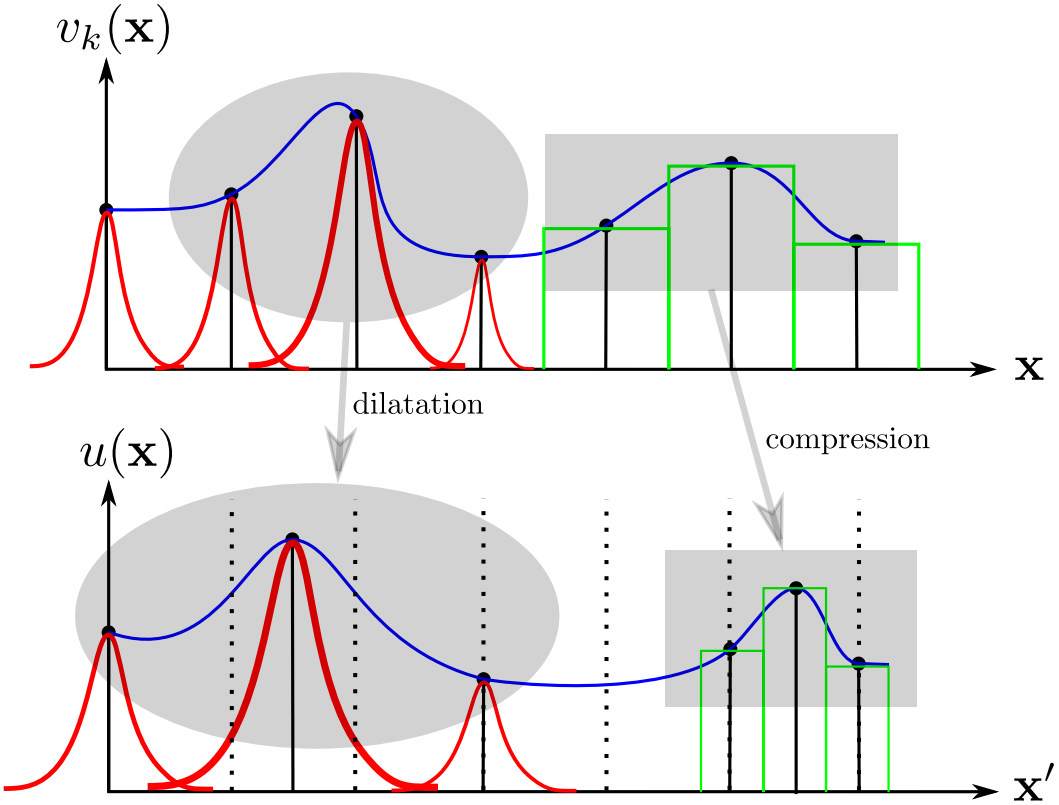


Fig. 4.1 – *Ré-échantillonnage des images. On représente par exemple des PSF gaussiennes (en rouge) et rectangulaire (en vert), indépendantes des zones de compression ou de dilatation. En haut : le signal source v_k échantillonné uniformément. En bas : le signal cible, obtenu par projection directe via la déformation τ_k, de jacobien $|\mathbf{J}|$. La déformation induit une dilatation ($|\mathbf{J}| > 1$) ou une compression du signal ($|\mathbf{J}| < 1$). Un ré-échantillonnage est nécessaire (en pointillé), soit à cause des dilatations et compressions (pour éviter le crénelage et le flou), soit parce que l'échantillon projeté ne tombe pas sur un nœud de la grille du domaine d'arrivée. L'interpolation des nouvelles valeurs est faite en utilisant la déformation locale de la PSF par τ_k, projection de ce que l'on appelle* splat, *ou éclaboussure.*

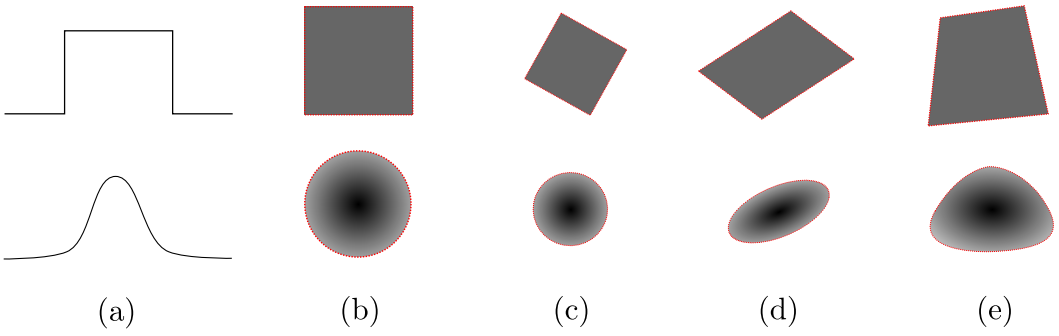


Fig. 4.2 – *Transformation de PSF. En haut : PSF carrée uniforme. En bas : PSF gaussienne. (a) Plan de coupe de la PSF originale. (b) PSF originale (dans la vue source). (c-e) PSF transformées, dans la vue cible, par une (c) similitude, (d) transformation affine, (e) transformation projective (homographie).*

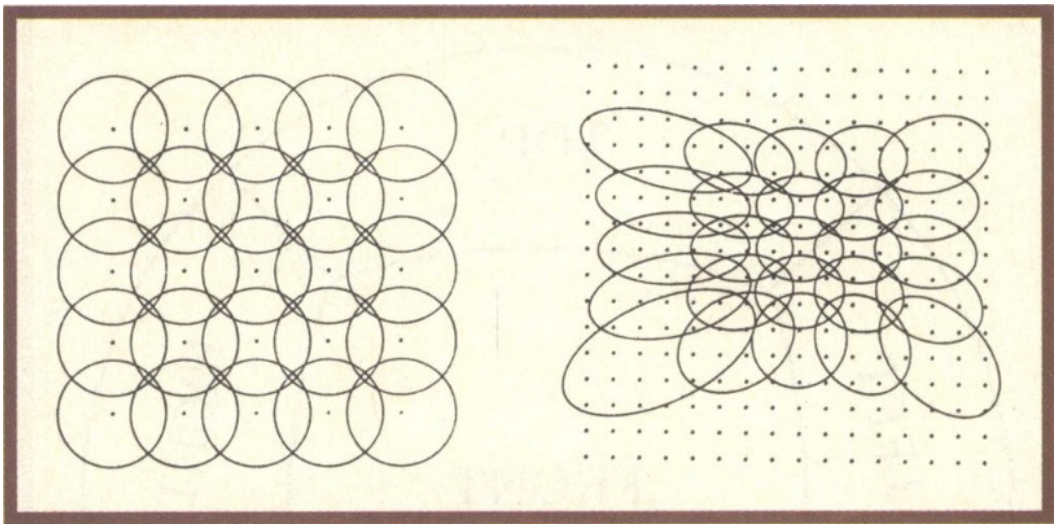


Fig. 4.3 – *Ré-échantillonnage avec des PSF circulaires (extrait des travaux de Greene et Heckbert (1986)). À gauche : image source. À droite : image cible. Chaque PSF a été déformée par une transformation locale affine pour ré-échantillonner le signal et prendre en compte les effets de compression et dilatation.*

cale, suivant l'angle de vue, la proximité de la scène, ou encore la distance focale. Les fonctions de déformation τ_k mettent en évidence ce changement d'échelle. Pour obtenir un échantillonnage cohérent avec la déformation employée, il faut choisir un filtre initial caractérisant l'échantillonnage de l'image, la PSF, et calculer sa transformée via τ_k pour trouver le nouvel échantillonnage. Greene et Heckbert (1986), en nous introduisant au filtre EWA [1], proposent un aperçu des différents filtres de ré-échantillonnage de texture. Ici comme nous faisons une projection directe, nous projetons l'image de l'*espace texture* (vue source) vers l'*espace écran* (vue cible). De la même manière, on compare sur la figure 4.2 les transformées des PSF de deux de ces filtres en fonction du type de transformation du plan. La figure 4.3 prend l'exemple d'un ré-échantillonnage de PSF déformées localement par des transformations affines.

Déformation de la PSF On peut montrer qu'on peut *localement* retrouver le signal cible u à partir du signal source v_k et de la transformation τ_k, en partant de l'équation $u \circ \tau_k = v_k$ (vrai partout où il n'y a pas d'occultation) liant les deux signaux dans le domaine continu (voir chapitre 2). On sait d'autre part que les signaux continus s'obtiennent par convolution du signal discret avec la PSF. Par un changement de variable dans la convolution et en exploitant les propriétés de la PSF on peut montrer que le signal cible continu s'exprime *localement* comme la convolution du signal source discret avec la transformée de la PSF par la fonction de déformation τ_k. Ce n'est pas réellement une convolution sur tout le support du signal car dans le cas général τ_k est spatialement variant ; τ_k n'étant pas uniforme, la transformée de la PSF est différente en chaque point. Nous devons donc projeter chaque échan-

1. Elliptical Weighted Average

tillon de l'image source (pixel) indépendamment et transformer la PSF par cette déformation locale. La transformée diffère selon le type de PSF et la transformation géométrique ; dans le cas générique et si les distorsions non-linéaires dues à l'optique sont négligeables, la transformation est localement projective. Les différentes transformations de PSF sont illustrées sur la figure 4.2. Si on suppose par exemple que la fonction d'étalement du point est gaussienne, la transformée par une fonction affine est encore une gaussienne, mais ne l'est plus par une transformation projective. De même un carré sera transformé en quadrilatère générique. La déformation de la PSF est déduite de la dérivée de la déformation. Les approximations qui seront faites par la suite sont des approximations au premier ordre.

Notre approche L'approche proposée est celle du rendu par éclaboussures (ou *splatting*), qui consiste à générer une image par accumulation d'une multitude d'éléments de surface 3D, les *splats* ou *footprints*, issus des PSF transformées. Le concept de *splat* est introduit pour la première fois par Pfister *et al.* (2000) sous le nom de *surfel* (pour élément de surface), un objet contenant une position, une couleur et une normale (orientation). L'idée de Zwicker *et al.* (2001) est alors de combiner le rendu de *splat* avec le filtrage elliptique de texture : c'est l'invention du *splat* elliptique. Par la suite des implémentations efficaces sur GPU du rendu de *splats* elliptiques ont été proposées par Ren *et al.* (2002); Botsch *et al.* (2005); Weyrich *et al.* (2007). Sachant que nous choisissons une PSF carrée uniformément distribuée, la PSF transformée est un quadrilatère comme décrit dans la figure 4.2e.

4.2.2 Création de *splats*

Partons d'une PSF simple telle qu'un carré uniforme (représentation très courante d'un pixel). L'objectif est de trouver la PSF transformée (dans le domaine image cible) par la déformation τ_k. On procède en deux temps : d'une part les données de profondeur recueillies par notre algorithme de reconstruction nous permettent de créer un objet 3D appelé *splat*, une sorte d'approximation locale de la surface. D'autre part on projette cet objet dans la caméra cible en accumulant sa contribution (rendu par éclaboussure), ce qui revient à interpoler l'image cible.

La création de *splat* à partir des images sources plutôt qu'à partir d'un nuage de points a de nombreux avantages :

- la taille du *splat* s'adapte à la profondeur de la surface observée : plus la surface est loin, plus taille du *splat* est grande.
- la taille du *splat* s'adapte aux paramètres de caméra tels que la distance focale : plus la focale est courte, plus taille du *splat* est grande.
- la taille du *splat* s'adapte aux variations de la déformation : une dilatation ($|\mathbf{J}| > 1$) augmente la taille du *splat*, compensant ainsi la formation de trous, et vice-versa pour les compressions ($|\mathbf{J}| < 1$).

Les points image $\mathbf{x} = (x, y)$ se notent aussi en coordonnées homogènes $\bar{\mathbf{x}} = (x, y, 1)$. On peut représenter ce vecteur comme directeur du rayon émergent de la caméra en coordonnées caméra locales (LC, *Local Coordinates* en anglais). Étant donnée la profondeur orthogonale z (vu de la caméra) du point 3D $\mathbf{X} = (X_1, X_2, X_3)$

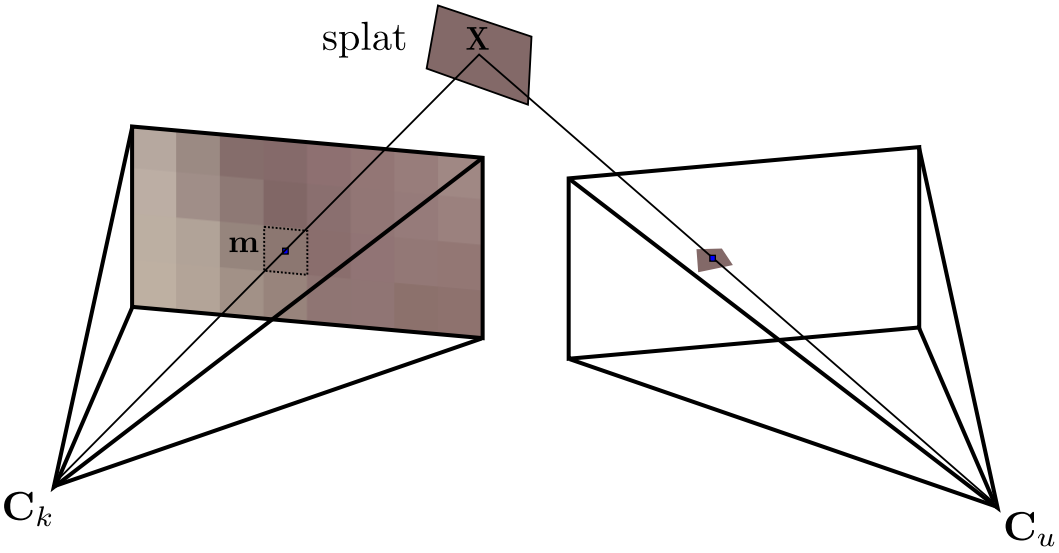


Fig. 4.4 – *Rendu par éclaboussure de quadrilatères (quad splatting).*

en coordonnées monde (WC), on a la relation

$$\mathbf{X} = \mathbf{C}_k + z.\mathbf{R}_k^\intercal \mathbf{K}_k^{-1} \bar{\mathbf{x}}. \tag{4.1}$$

À partir de cette équation et des données de profondeurs nous générons un *splat* quadrilatère comme l'illustre la figure 4.4.

Création de *quad* : cas de la refocalisation La refocalisation est une application du rendu-basé image à géométrie plane, c'est à dire un scène dont la représentation est un plan de profondeur constante. Le plan en question, appelé *plan focal* est positionné dans la scène à la profondeur à laquelle on souhaite mettre le focus. Supposons dans un premier temps que le plan focal est le plan d'équation $X_3 = d$, en coordonnées monde (WC). Pour chaque coin d'un pixel source nous cherchons l'intersection du rayon émergent avec le plan focal. La projection de l'équation (4.1) sur le troisième axe du repère monde nous donne la profondeur de chaque sommet du *quad* :

$$z = \frac{d - \mathbf{C}_k[3]}{(\mathbf{R}_k^\intercal \mathbf{K}_k^{-1}(\bar{\mathbf{x}} + (\Delta_x, \Delta_y, 0)))[3]}. \tag{4.2}$$

Pour trouver les coordonnées des sommets, il suffit alors de remplacer z par (4.2) dans (4.1) et de résoudre pour $\mathbf{X}$, en prenant pour $\bar{\mathbf{x}}$ les coordonnées des coins du pixel.

En pratique le plan focal n'a pas pour équation $X_3 = d$ en WC mais dans le repère local à la caméra cible (LC), de sorte qu'il soit parallèle à son plan image. On a alors

$$z = \frac{d - (\mathbf{R}_u \mathbf{C}_k + \mathbf{t}_u)[3]}{(\mathbf{R}_u \mathbf{R}_k^\intercal \mathbf{K}_k^{-1}(\bar{\mathbf{x}} + (\Delta_x, \Delta_y, 0)))[3]}. \tag{4.3}$$

Comme précédemment les sommets s'obtiennent en remplaçant z par (4.3) dans

(4.1).

Création de *quad* : cas générique Dans le cas général les informations recueillies par le logiciel d'estimation de cartes de profondeurs (Fuhrmann *et al.*, 2014) sont sous la forme $(z, z_{\mathbf{x}})$, avec $z_{\mathbf{x}} = \frac{\partial z}{\partial \mathbf{x}}$ représentant l'orientation de la surface vue de la caméra. Elles représentent localement un plan vu du point image $\mathbf{x}$, et nous permettent de créer un *quad* dans l'espace 3D pour chacun de ces points image. Pour cela nous créons un *shader* qui prend, pour chaque pixel de l'image, les coordonnées $\mathbf{x} + (\Delta_x, \Delta_y)$ $(\Delta_x, \Delta_y \in \{-0.5, 0.5\})$ de ses coins, et crée quatre sommets $\mathbf{X}$ obtenus par intersection des rayons émergents avec le plan focal. On approxime la profondeur de chaque coin par un développement limité au premier ordre, connaissant la profondeur au centre du pixel z ainsi que la dérivée au premier ordre $z_{\mathbf{x}}$. En somme, l'équation (4.1) nous donne une approximation des coordonnées des sommets du *quad* :

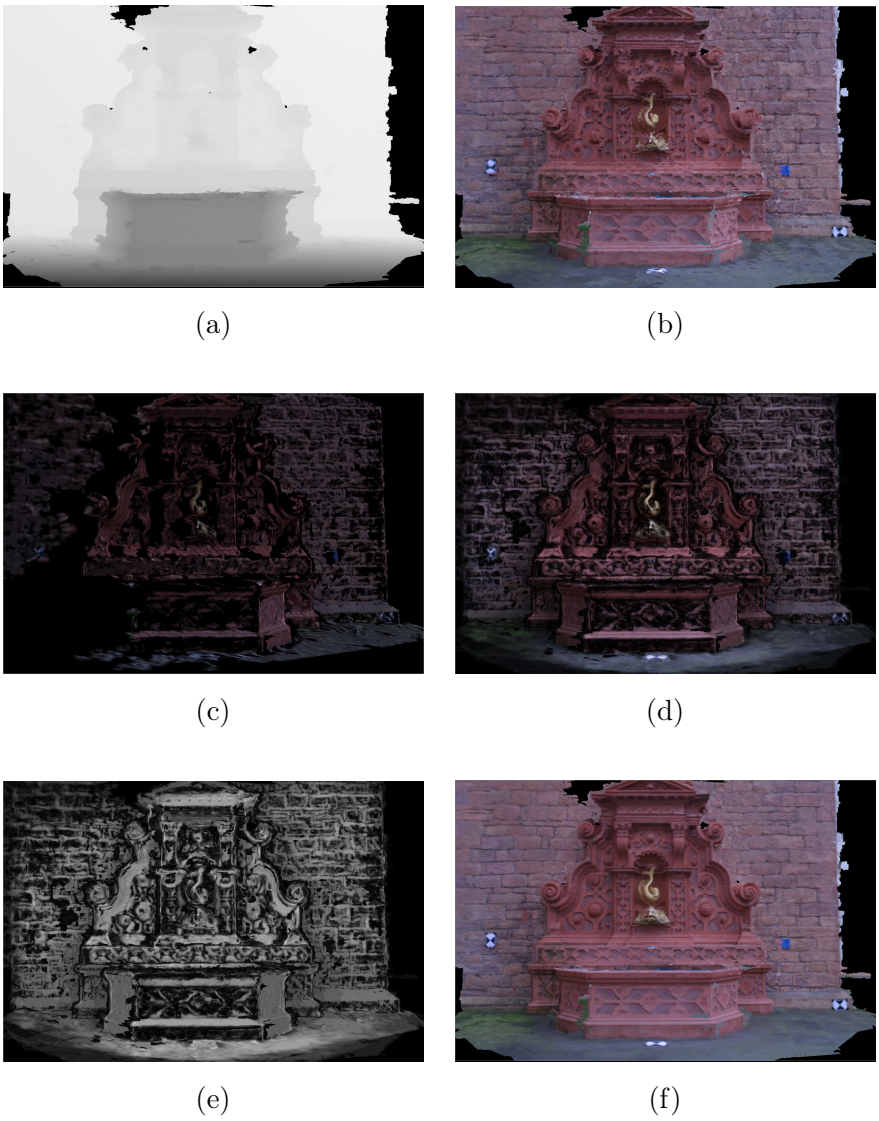
$$\mathbf{X} = \mathbf{C}_k + (z + \Delta_x z_x + \Delta_y z_y).\mathbf{R}_k^\intercal \mathbf{K}_k^{-1} \bar{\mathbf{x}}. \tag{4.4}$$

Lorsque la surface est vue de biais par le point de vue cible, une taille de pixel de $\{-0.5, 0.5\}$ est souvent insuffisante à la création d'un nuage de *splats* assez dense pour ne pas voir l'apparition de nombreux trous. Cela provient de l'erreur de localisation des éléments de surface locaux que nous procure l'algorithme de MVS. Nous augmentons alors artificiellement la taille d'un pixel dans la génération de *splats* pour compenser cet effet (figure 4.5).

4.2.3 Rendu par éclaboussures

L'algorithme de rendu dit « par éclaboussure » (ou *splatting*) permet de résoudre le problème de ré-échantillonnage abordé dans 4.2.1 : interpoler les pixels de l'image cible après *warping* (déformation et projection) des vues sources. Il s'effectue en trois phases décrites par Gross et Pfister (2011) : le calcul de la visibilité, le mélange des images et enfin la normalisation. Nous l'avons implémenté grâce à l'API `OpenGL`. Les deux premières phases partagent les mêmes trois *shaders* : géométrie, vertex et fragment. Le *shader* de géométrie prend en paramètres la géométrie estimée (pro-

Fig. 4.5 – *Rendu par éclaboussures avec différentes tailles de PSF (en pixels). À gauche : 0.5. Au milieu : 1.0. À droite : 2.0.*



(a) (b)

(c) (d)

(e) (f)

Fig. 4.6 – *(a) Z-buffer à la fin du calcul de visibilité. (b) Le rendu classique avec ce z-buffer, sans mélange des splats, présentent de nombreux artefacts. Étape de mélange : première (c) et dernière (d) passe. (e) Poids de normalisation. (f) Résultat du rendu par éclaboussures.*

fondeur du plan dans le cas de la refocalisation, et carte de profondeur dans le cas général), et crée pour chaque pixel source une primitive de *quad*. Les deux autres *shaders* font le rendu du *quad* dans la vue cible par rastérisation classique.

Calcul de la visibilité La phase de visibilité consiste à calculer un *z-buffer* dans la vue cible pour gérer les problèmes d'occultations (figure 4.6a). Nous effectuons le rendu de chaque vue à l'aide des *shaders* précédemment décrits, en conservant le même *z-buffer* qui se construit successivement. Sans *z-buffer*, l'image rendue se présente comme sur la figure 4.6b.

Mélange Durant la deuxième phase, nous réitérons le rendu pour toutes les vues sources, en accumulant les *quads* qui ont une profondeur voisine de celle du *z-buffer*, ce qui a pour effet de mélanger les parties non occultées des images sources (figure 4.6c-d). Le canal *alpha* de l'image est utilisé afin de comptabiliser le nombre de contributions dans la couleur du pixel final. Enfin, la normalisation vient diviser chaque pixel par le canal alpha.

Normalisation L'utilisation des poids de mélange s'effectue via le canal *alpha* (figure 4.6e). Lorsqu'un *quad* provenant d'une certaine vue source est rendu, la couleur du pixel source est multipliée par ce facteur contributif avant d'être accumulée dans les canaux RGB de l'image cible. Ce même poids est ajouté au canal *alpha* qui somme toutes les contributions. Le résultat est présenté sur la figure 4.6f.

4.2.4 Poids des contributions

Toutes les vues n'ont pas la même contribution dans le rendu de l'image cible, de même que tous les points image d'une même vue. Buehler *et al.* (2001) formulent des heuristiques pour décrire ces poids, suivant les propriétés qu'un algorithme d'IBR idéal devrait posséder : la *déviation angulaire minimale* et la *sensibilité à la résolution*.

Sensibilité à la résolution Selon la propriété de *sensibilité à la résolution*, les vues qui observent la scène avec plus de détail devraient contribuer plus que les autres. Ainsi sur la figure 4.7 de gauche, le point $\mathbf{x}_2$ de la vue $\mathbf{C}_2$ a moins de poids que le point $\mathbf{x}_1$ de la vue $\mathbf{C}_1$ car il « voit » la surface de biais et a une focale plus courte. La distance à laquelle est observée la scène joue aussi un rôle. Nous appelons ces poids les *poids de résolution* ou les *poids de déformation*. Comme il a été proposé par Wanner et Goldluecke (2012b) nous prenons comme poids $|\mathbf{J}_{\mathbf{x}}(\tau_k)|^{-1}$, l'inverse du jacobien de la déformation τ_k au point $\mathbf{x}$. Ces poids dépendent donc de la dérivée spatiale de la fonction de déformation.

Déviation angulaire minimale La *déviation angulaire minimale* consiste à privilégier les points des images sources dont le rayon incident (partant de la vue source) forme un petit angle avec le rayon réfléchi (arrivant sur la vue cible). En particulier,

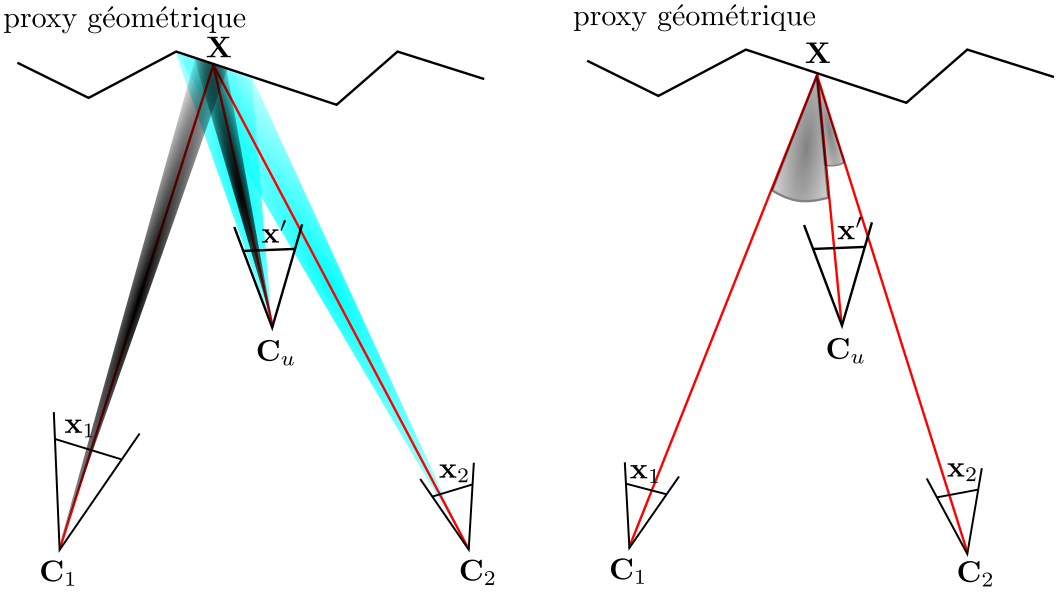


Fig. 4.7 – *Poids des contributions des pixels sources dans le rendu. À gauche : poids de déformation. À droite : poids de géométrie.*

si l'angle est nul, les deux rayons sont confondus, et le point image cible devrait être de la même couleur que le point image source : c'est la *cohérence épipolaire*. Sur la figure 4.7 à droite on voit que le point $\mathbf{x}_2$ a plus de poids, car le rayon correspondant est plus « proche » du rayon associé au point cible $\mathbf{x}'$. L'angle est une mesure de proximité des rayons optiques, mais d'autres existent. Pujades *et al.* (2014) proposent pour chaque vue k et chaque pixel $\mathbf{x}$ un poids $\omega_k(u)$ basé sur l'incertitude de localisation du *proxy géométrique*. Nous appelons ces poids les *poids géométriques* :

$$\omega_k(u) = (\sigma_{s,k}^2 + \sigma_{g,k}^2(u))^{-1}, \tag{4.5}$$

$$\text{avec } \sigma_{g,k}^2(u) = \sigma_{z,k}^2 (b * (\frac{\partial \tau_k}{\partial z}^t \nabla u \circ \tau_k))^2, \tag{4.6}$$

$$\text{et } \nabla u \text{ le gradient de la solution } u. \tag{4.7}$$

Les poids représentent l'incertitude sur la couleur de la vue k en fonction de celle sur la profondeur, connaissant la vue u. Ces poids dépendent de u, ou du moins sont calculés à partir d'une estimation de u dans le processus d'optimisation d'une méthode inverse. Dans une approche directe du rendu basé image, nous ne connaissons pas u et *a fortiori* son gradient. Par conséquent nous les exprimons en fonction des dérivées par rapport à la profondeur z seulement. b dénote la PSF, et l'opérateur $*$ représente la convolution. La variance $\sigma_{z,k}$ représente l'incertitude de localisation du *proxy* sur l'axe des profondeurs (locales, depuis la vue k), et $\sigma_{s,k}$ le bruit du capteur, pris constant.

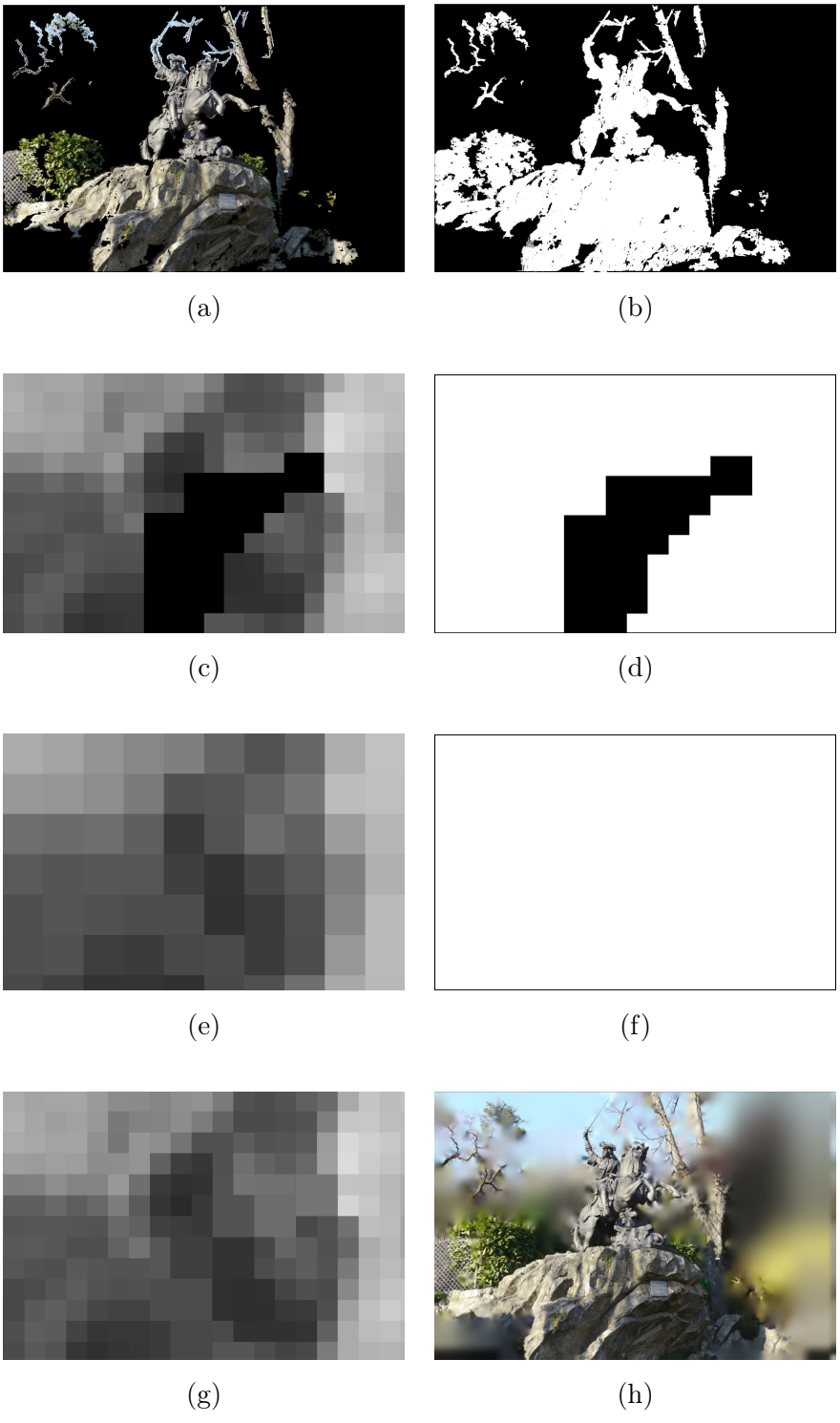


Fig. 4.8 – *Algorithme d'inpainting* `PUSH/PULL`. *L'image initial (a) présente des trous à combler, selon un masque (b) (*`false` *pour un trou,* `true` *sinon). L'étape* `PUSH` *consiste à construire une pyramide d'images. Nous passons d'un niveau de résolution, respectivement une image (c) et un masque (d) au niveau supérieur, respectivement une image (e) et un masque (f), par filtrage gaussien en ne prenant en compte que les pixels dont la valeur du masque est* `true`. *L'étape* `PULL` *consiste à effondrer cette pyramide. Nous passons d'un niveau de résolution à un autre en propageant les couleurs du niveau supérieur (e) dans les trous (masque vaut* `false`*) des niveaux inférieurs (g). (h) Résultat final.*

4.2.5 Remplissage de trous

Rappelons que ces images prises depuis les différents points de vue échantillonnent le champ de lumière 4D. Suivant l'échantillonnage choisi (c'est-à-dire le nombre et la position des points de vue), la scène est couverte avec plus ou moins d'éxhaustivité. Si le point de vue cible (une autre portion du champ de lumière) ne correspond pas entièrement à l'échantillonnage choisi, sa synthèse est alors incomplète. Cela se traduit par une zone floue lorsque l'échantillonnage spatial est faible ou par une zone inconnue (zones noires de l'image sur la figure 4.8a) lorsque l'échantillonnage angulaire est insuffisant. Dans ce dernier cas, on pourrait dire plus simplement que les zones de l'image cible ne sont vues par aucune vue source. On recourt alors à une méthode classique d'*inpainting*. Notre objectif ici n'est pas de donner un aperçu global des méthodes d'*inpainting*, ni d'apporter une solution nouvelle qui sortirait du cadre de cette thèse. Nous proposons cependant de détailler l'algorithme `PUSH/PULL` de Gortler *et al.* (1996) car il présente des similitudes avec notre algorithme d'IBR multi-résolution présenté à la section 4.3.3. Nous utilisons cet algorithme dans nos expériences afin de donner un rendu plus esthétique à nos images.

Puisque Gortler *et al.* (1996) ne donnent que peu de détails d'implémentation dans son article, nous allons essayer de compléter ses propos. L'idée est de décomposer l'image cible (à compléter) en pyramide gaussienne (voir appendice B) d'une certaine hauteur qui sera discutée plus tard, et de la recomposer en propageant les informations manquantes dans les zones à trous. Bien que nous ayons implémenté l'algorithme pour tourner sur CPU, il est aisément parallélisable car il repose sur des opérations sur des pyramides d'images. Sa complexité est linéaire en le nombre de pixels de l'image.

L'algorithme prend en entrée l'image à compléter I_0, ainsi qu'un masque booléen M_0 qui indique si un pixel possède une couleur (`true`) ou non (`false`).

Étape `PUSH` La première étape consiste à calculer une pyramide gaussienne I et une pyramide de masques M, de telle manière qu'à chaque niveau de résolution n, le masque réduit M_n corresponde aux pixels connus de l'image réduite I_n. Pour ce faire nous avons implémenté le filtre `Odd HDC` de Burt (1981) : un noyau binomial 5×5 K qui approxime une gaussienne. La différence avec une construction classique de la pyramide d'image est que nous prenons en compte le masque booléen. À chaque étape de réduction (opérateur `reduce`), les pixels contribuent seulement s'ils sont connus (valeur `true` dans le masque correspondant au même niveau de résolution) :

$$I_{n+1} = \texttt{reduce}(M_n I_n). \tag{4.8}$$

On sous-échantillonne successivement les images par 2. Au niveau maximal de la pyramide, tous les trous sont remplis. L'opérateur `OR` est utilisé pour créer la pyramide de masques : la valeur de l'image réduite vaut `true` si au moins un des pixels contribuant est connu. A la fin de cette étape `PUSH`, le masque du niveau maximal vaut `true` partout.

Étape `PULL` La seconde étape consiste à effondrer la pyramide, des niveaux de résolution inférieure aux niveaux de résolution supérieure, en appliquant un opérateur `expand` modifié. Cet opérateur ne calcule que les pixels dont la valeur est inconnue (le masque vaut `true`). Ainsi on a

$$I_{n-1} = M_{n-1}I_{n-1} + (1 - M_{n-1})\texttt{expand}(I_n). \tag{4.9}$$

Le résultat de l'`inpainting` est l'image de plus haute résolution de la pyramide résultante. L'algorithme est illustré par la figure 4.8.

La hauteur de la pyramide dépend de la taille de la plus grande zone à compléter de l'image. Dans le pire des cas elle remplit plus du quart de l'image et il est nécessaire d'aller jusqu'au niveau de résolution minimale (image de taille 1×1). Dans la plupart des cas cependant il est inutile d'appliquer autant d'opérations `reduce`, et il suffit de s'arrêter quand le masque M_n vaut `true` partout.

4.3 Fusion d'images multi-résolution

4.3.1 Motivations

Dans la section 4.2 nous avons présenté une méthode de rendu basé image qui vise à combiner les intensités des images sources afin de générer une nouvelle image. Un mélange en intensité peut créer des artefacts de rendu (figure 4.9) qui ont l'aspect d'un bruit en haute fréquence (HF) lorsque les fonctions de déformation et les poids des contributions des pixels sources sont eux-même bruités. Fusionner des signaux en intensité ne garantit pas la cohérence des couleurs des pixels voisins dans l'image finale.

Une première solution est de combiner non pas les *couleurs* des images sources, mais leur *gradients*. Une fois le gradient de l'image cible reconstitué par un rendu de gradient (Kopf *et al.*, 2013, 2014), l'image couleur est reconstituée par résolution d'une équation de Poisson (Pérez *et al.*, 2003). Cette résolution consiste à trouver l'image dont le laplacien est égal à la divergence du champ de gradient généré par le rendu de gradient. Cela revient à fusionner les images dans le domaine fréquentiel, mais seulement pour les hautes fréquences, éliminant les variations rapides de couleurs qui résultent des artefacts de *proxy* (figure 4.9). Pourquoi seulement les hautes fréquences ? Car le laplacien d'une image calculé à sa résolution originale de Pérez *et al.* (2003); Kopf *et al.* (2013, 2014) retient les détails les mieux résolus de l'image. En effet le laplacien peut s'estimer (approximativement) par une différence de gaussiennes (DoG) de résolutions discrètes voisines (deux niveaux discret successifs dans l'espace d'échelle). Il s'agit donc d'un filtre passe-bande centré sur une fréquence inversement proportionnelle à la résolution à laquelle il est calculé, en l'occurrence un passe-haut car calculé à la résolution originale des images.

L'approche qui est proposée est une approche de rendu multi-résolution, où l'on tente de mélanger les images sources à tous les niveaux de résolution, via des pyramides laplaciennes. Dans la littérature de *stitching* (Burt et Adelson, 1983a; Zomet

Fig. 4.9 – *Rendu de la vue centrale du jeu de données fountain. Une fusion des images en intensité crée des artefacts de rendu. Le bruit des cartes de profondeur fausse la projection des pixels des vues sources ; des pixels de couleur différente se retrouvent alors voisins. Cela motive l'élaboration d'une méthode de rendu multi-échelles pour fusionner les images dans le domaine fréquentiel, à la manière d'un mélange laplacien.*

et al., 2006), cela porte le nom de *mélange laplacien*[2]. En considérant que le laplacien d'une image se comporte comme un filtre passe-bande, fusionner les laplaciens des images sources est analogue à mélanger les images dans le domaine fréquentiel pour une bande de fréquences qui dépend de la résolution à laquelle le laplacien est calculé. L'idée clé du rendu multi-résolution est de faire du rendu de laplacien à chaque niveau de résolution, permettant un mélange laplacien complet à toutes les fréquences, pour prévenir l'apparition d'artefacts quelle que soit leur fréquence. Dans un premier temps nous verrons comment l'espace d'échelle d'une image source se transforme par le biais d'une fonction de déformation, et ce que cela implique pour le laplacien de l'image cible. Dans un second temps nous exposerons notre algorithme de rendu basé image multi-résolution qui prend en compte les difficultés soulevées par la déformation de l'espace d'échelle.

4.3.2 Déformation d'espace d'échelle

Notre objectif est de décomposer les images sources dans le domaine fréquentiel via l'*espace d'échelle*, une représentation de ces images en plusieurs niveaux de résolution, que l'on appellera de manière équivalente niveaux d'échelle, caractérisés par un *facteur d'échelle* σ, synonyme ici de *résolution*. Nous rappelons que le signal source peut s'exprimer comme la composition du signal cible par une fonction de déformation :

$$v_k = u \circ \tau_k. \tag{4.10}$$

Dans cette section nous étudions dans quelle mesure la fonction de déformation τ_k affecte l'espace d'échelle du signal, selon les hypothèses qui sont faites sur elle. En particulier nous nous penchons sur le cas des dérivées secondes, car nous représentons l'espace d'échelle via les laplaciens de l'image calculés à de multiples résolutions.

2. *Laplacian blending*

On définit U et V_k comme étant les signaux u et v_k convolués par une gaussienne de variance σ^2 ($*$ est l'opérateur de convolution) :

$$U(\mathbf{x}', \sigma') = u * g_{\sigma'}(\mathbf{x}') \tag{4.11}$$

$$V_k(\mathbf{x}, \sigma) = v_k * g_{\sigma}(\mathbf{x}) \tag{4.12}$$

Ainsi $V_k(\mathbf{x}, \sigma)$ représente le signal source à la résolution $\sigma > 0$, idem pour U. On désignera de manière équivalente la résolution et l'échelle du signal par l'écart type σ. On définit ainsi l'espace d'échelle l'espace des images résolues à l'échelle σ comme un espace à trois dimensions $(\mathbf{x}, \sigma)$. Nous cherchons ici à analyser les transformations d'espaces d'échelle, c'est-à-dire la transformation de $V_k(\mathbf{x}, \sigma)$ en $U(\mathbf{x}', \sigma')$ par le biais de τ_k, quel que soit $(\mathbf{x}, \sigma)$. $\sigma' > 0$ est la résolution ou le facteur d'échelle d'arrivée, qui est *a priori* différent de σ.

Dilatation en dimension $n = 1$ On suppose que la transformation τ_k est une dilatation de facteur s : $\tau_k(\mathbf{x}) = \mathbf{x}' = s.\mathbf{x}$. Ainsi $v_k(\mathbf{x}) = u \circ \tau_k(\mathbf{x}) = su(\mathbf{x})$. Par définition de la convolution

$$U(\mathbf{x}', \sigma') = \int_{\mathbf{t}' \in \Gamma} g_{\sigma'}(\mathbf{x}' - \mathbf{t}')\, u(\mathbf{t}')\, \mathrm{d}\mathbf{t}'. \tag{4.13}$$

En effectuant le changement de variable $\mathbf{t} \to \mathbf{t}' = s.\mathbf{t}$, on obtient

$$U(\mathbf{x}', \sigma') = \int_{\mathbf{t} \in \Omega_k} g_{\sigma'/s}(\mathbf{x} - \mathbf{t})\, v_k(\mathbf{t})\, \mathrm{d}\mathbf{t}. \tag{4.14}$$

En posant $\sigma' = s\sigma$, on a

$$U(s.\mathbf{x}, s\sigma) = V(\mathbf{x}, \sigma). \tag{4.15}$$

Une dilatation déplace le signal dans tout l'espace d'échelle, affectant non seulement le domaine spatial mais aussi la résolution σ.

Transformation affine en dimension $n > 1$ Les domaines Ω_k et Γ sont respectivement de dimensions n et $n' > 1$. On suppose que la transformation τ_k est affine, de telle manière que $\tau_k(\mathbf{x}) = \mathbf{a} + \mathbf{J}\mathbf{x}$ avec $\mathbf{J}$ la jacobienne de τ_k, de taille $n' \times n$. Le filtre gaussien est multidimensionnel de matrice de covariance $\mathbf{\Sigma}$ représentant l'échelle. En effectuant le changement de variable $\mathbf{t} \to \mathbf{t}' = \tau_k(\mathbf{t})$,

$$U(\mathbf{x}', \mathbf{\Sigma}') = \int_{\mathbf{t} \in \Omega_k} g_{\mathbf{J}^{-1}\mathbf{\Sigma}'\mathbf{J}^{-\intercal}}(\mathbf{x} - \mathbf{t})\, v_k(\mathbf{t})\, \mathrm{d}\mathbf{t}. \tag{4.16}$$

En posant $\mathbf{\Sigma}' = \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\intercal}$, on a

$$U(\tau_k(\mathbf{x}), \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\intercal}) = V(\mathbf{x}, \mathbf{\Sigma}). \tag{4.17}$$

Comme précédemment on peut conclure qu'**une transformation affine modifie toutes les dimensions de l'espace d'échelle, en particulier la résolution.**

Transformation rigide en dimension $n = 2$ Dans notre cas d'application (images 2D), les domaines Ω_k et Γ sont de dimension 2. De plus l'échelle est scalaire $\sigma > 0$ (l'espace d'échelle n'ajoute qu'une seule dimension au domaine spatial). En dimension 2, on peut écrire notre l'échelle $\mathbf{\Sigma} = \sigma^2.\mathbf{I}_2$ où $\mathbf{I}_2$ est la matrice 2×2 identité. On se demande alors quelle condition sur la déformation τ_k est nécessaire pour préserver une résolution scalaire, c'est-à-dire pour que $\mathbf{\Sigma}' = \sigma'^2.\mathbf{I}_2$ où σ' est un scalaire > 0. Nous avons vu que dans le cas général où τ_k est affine, la résolution σ est transformée en $\mathbf{\Sigma}' = \mathbf{J}\mathbf{\Sigma}\mathbf{J}^\intercal = \sigma^2.\mathbf{J}\mathbf{J}^\intercal$. Autrement dit $\mathbf{J}$ doit être une matrice orthogonale à un facteur s près. On a alors $\sigma' = |s|\sigma$. La transformation qui correspond à cette jacobienne est une similitude du plan : c'est une transformation orthogonale composée avec une dilatation et une translation. Le jacobien ne dépend que du facteur de la dilatation $|\mathbf{J}| = s^2$.

Nous voyons que si l'on souhaite conserver le facteur d'échelle (ou niveau de résolution), la déformation doit être une similitude, une hypothèse qui est très forte. Dans le cas général les déformations ne sont même pas affines. Inspirée par le modèle de la similitude, on fera l'approximation

$$U(\tau_k(\mathbf{x}), \sqrt{|\mathbf{J}|}\sigma) = V(\mathbf{x}, \sigma). \tag{4.18}$$

Pour conclure nous dirons que la déformation de l'espace d'échelle pose déjà deux problèmes majeurs :

- **Le facteur d'échelle change** : la résolution de départ n'est pas celle d'arrivée. Ainsi il n'est pas possible d'utiliser la décomposition des images en pyramides laplacienne de façon naïve pour reconstruire l'image cible exacte, puisque que pour un certain niveau de la pyramide de l'image source, sa déformation ne correspond pas au même facteur d'échelle qu'au départ. Une conséquence est que le signal issu d'un certain niveau de la pyramide peut être étalé sur plusieurs niveaux selon la déformation. Certaines parties de l'image sont dilatées et baissent en fréquence (et donc leur facteur d'échelle augmente), tandis que d'autres sont compressées par la déformation (le facteur d'échelle diminue). Nous proposons néanmoins d'approximer (4.18) le facteur d'échelle résultant, et donc de prévoir le niveau de la pyramide où se situera le signal déformé.
- **Le facteur d'échelle résultant est anisotrope**. La représentation de l'espace d'échelle de l'image cible en dimension 3 (2 pour le domaine spatial, et 1 pour la résolution) est faussée. Cette déformation de l'espace d'échelle dépend de la jacobienne de la transformation. Il est possible de quantifier cette déformation par le jacobien, ce qui donne une bonne indication des déformations susceptibles d'altérer le signal dans le domaine fréquentiel et dans quelle mesure elles le font.

Transformation des dérivées La déformation de l'espace d'échelle pose un autre problème lorsqu'il s'agit de transformer les dérivées du signal dans le cas général. Dans notre cas le plus simple où le signal est scalaire et où la déformation est une dilatation de facteur s, nous avons d'après les propriétés de dérivation d'une gaussienne

$$\frac{\partial}{\partial \mathbf{x}'} U(\mathbf{x}', \sigma') = \frac{1}{s} \frac{\partial}{\partial \mathbf{x}} V(\mathbf{x}, \sigma). \tag{4.19}$$

On démontre alors par récurrence, que $\forall n \in \mathbb{N}^*$,

$$\frac{\partial^n}{\partial \mathbf{x}'^n} U(\mathbf{x}', \sigma') = \frac{1}{s^n} \frac{\partial^n}{\partial \mathbf{x}^n} V(\mathbf{x}, \sigma). \tag{4.20}$$

En particulier dans notre cas en dimension $n = 2$, la dérivée seconde du signal cible est égale à la dérivée seconde du signal d'arrivée à un facteur $s^2 = |\mathbf{J}|$ près. Nous concluons de la même manière dans le cas où la transformation est une similitude du plan. Nous obtenons la formule exacte liant les laplaciens des deux signaux

$$\Delta U(\tau_k(\mathbf{x}), \sqrt{|\mathbf{J}|}\sigma) = \frac{1}{|\mathbf{J}|} \Delta V(\mathbf{x}, \sigma). \tag{4.21}$$

Dans le cas général où aucune hypothèse n'est faite sur la déformation, la formule est très complexe. On peut cependant négliger les ordres > 1 dans le développement limité de la déformation, ce qui revient à dire que la jacobienne $\mathbf{J}$ est constante par rapport à $\mathbf{x}$. Nous obtenons alors les expressions du gradient ∇ (4.22) et de la hessienne $\mathbf{H}$ (4.23) :

$$\nabla v_k = \mathbf{J}^\intercal \nabla u \circ \tau_k. \tag{4.22}$$

$$\mathbf{H} v_k = \mathbf{J}^\intercal (\mathbf{H} u \circ \tau_k) \mathbf{J}. \tag{4.23}$$

Nous pouvons alors exprimer le laplacien du signal source en prenant la trace de la hessienne. Mais cela ne permet pas d'obtenir une expression le liant directement au laplacien du signal cible :

$$\Delta v_k = \mathrm{Tr}(\mathbf{J}^\intercal (\mathbf{H} u \circ \tau_k) \mathbf{J}). \tag{4.24}$$

Nous interprétons l'impossibilité d'exprimer le laplacien source en fonction du laplacien cible par le fait que l'isotropie du laplacien est compromise par la déformation. Celui-ci ne conserve son caractère isotrope que si la déformation est une similitude. En pratique on se servira de l'équation (4.21) comme d'une approximation permettant de lier les laplaciens des signaux source et cible.

On conclut que **la valeur du laplacien est affectée par la déformation**, contrairement au signal image qui ne change pas de valeur. En effet une compression de l'image a pour effet d'accentuer les contours en comprimant les textures, d'où l'augmentation de la valeur du laplacien ; inversement pour une dilatation de l'image.

Lemme. *Le laplacien d'une image est invariant par rotation et translation, et par ces transformations d'image seulement.*

Démonstration. L'équation (4.24) montre que trouver les transformations d'image qui préservent le laplacien est équivalent à trouver les matrices $\mathbf{J}$ de taille 2×2, telles que $\forall \mathbf{H}$ une matrice de même taille,

$$\mathrm{Tr}(\mathbf{J}^\intercal \mathbf{H} \mathbf{J}) = \mathrm{Tr}(\mathbf{H}). \tag{4.25}$$

Notre intuition est que les $\mathbf{J}$ qui satisfont (4.25) sont les matrices orthogonales (et

seulement elles). Montrons le premier sens de l'équivalence, c'est-à-dire que si une matrice $\mathbf{J}$ est orthogonale, alors elle satisfait (4.25). Soit $\mathbf{J}$ une matrice orthogonale de taille 2×2, c'est-à-dire qui a la propriété $\mathbf{J}^{-1} = \mathbf{J}^\intercal$ ou de manière équivalente $\mathbf{J}^\intercal\mathbf{J} = \mathbf{J}\mathbf{J}^\intercal = \mathbf{I}_2$, où $\mathbf{I}_2$ est la matrice identité. Alors par propriété de la trace on a

$$\mathrm{Tr}(\mathbf{J}^\intercal\mathbf{H}\mathbf{J}) = \mathrm{Tr}(\mathbf{J}\mathbf{J}^\intercal\mathbf{H}) = \mathrm{Tr}(\mathbf{H}). \tag{4.26}$$

Pour montrer l'autre sens de l'équivalence, nous supposons maintenant que $\mathbf{J}$ satisfait (4.25) $\forall\mathbf{H}$. Alors en particulier si l'on prend pour matrice $\mathbf{H}$ les quatre matrices $\mathbf{H}_{ij}$ (i et j entre 0 et 1) dont tous les éléments sont nuls sauf l'élément (i, j), on montre que les vecteurs colonnes de $\mathbf{J}$ forment une base orthonormée. En d'autres termes, $\mathbf{J}$ est orthogonale. Puisque la jacobienne $\mathbf{J}$ est forcément une matrice orthogonale, alors la transformation d'image τ_k est une rotation, une translation ou une composition des deux. □

4.3.3 Application au rendu multi-échelles

Rendu multi-échelles naïf Nous proposons une première méthode naïve pour faire du rendu multi-échelles. Nous expliquerons par la suite en quoi les problèmes soulevés précédemment lors de la déformation de l'espace d'échelle compromettent cette approche naïve.

L'idée clé est de décomposer chaque image source en une pyramide laplacienne (voir appendice B) pour en avoir une décomposition fréquentielle (figure 4.10). Chaque niveau σ de la pyramide laplacienne est un laplacien de l'image source calculé à la résolution σ. Il représente une certaine bande de fréquence de l'image car le laplacien se comporte comme un filtre passe bande centré sur une fréquence inversement proportionnelle à sa résolution σ. La pyramide est calculée par des DoG (*Difference of Gausian*) (Burt et Adelson, 1983a) d'une pyramide d'image, selon le même filtre `Odd HDC` introduit par Burt (1981). Chaque niveau est sous-échantillonné par un facteur 2, de telle sorte que l'image couleur au sommet de la pyramide n'a que quelques pixels. Nous effectuons un rendu direct comme présenté dans la section précédente en prenant comme images sources les laplaciens à chaque niveau. Nous faisons ainsi un rendu par niveau de la pyramide (8 niveaux en pratique). Les poids de rendu sont obtenus en décomposant chaque carte de poids originale en une pyramide gaussienne. On obtient enfin une nouvelle pyramide laplacienne, que l'on effondre pour générer l'image cible.

Cette méthode naïve échoue sur plusieurs points évoqués précédemment. Premièrement, lors de la déformation du laplacien source à une certaine résolution σ, la résolution d'arrivée (le facteur d'échelle) σ' est différent. D'autre part l'échantillonnage de l'espace d'échelle est discret (8 niveaux), et la résolution transformée par la déformation τ_k peut se retrouver entre deux niveaux discrets. Cet échantillonnage de l'espace d'échelle ne tient pas compte de l'effet de la déformation ; combiné au sous-échantillonnage spatial successif par construction de la pyramide, nous obtenons des artefacts oscillatoires comme le montre la figure 4.11. Deuxièmement, la déformation

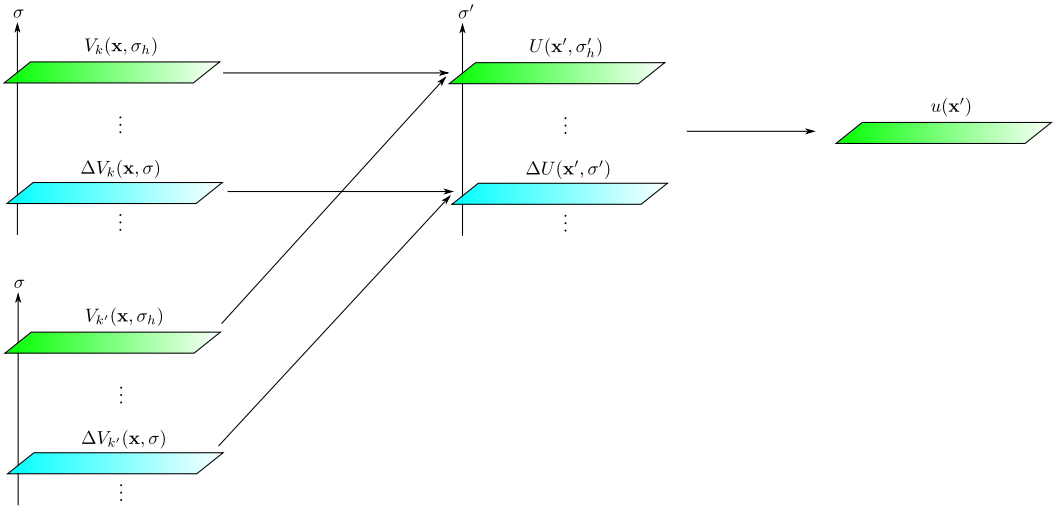


Fig. 4.10 – *Rendu multi-échelles naïf. Chaque image source v_k est décomposée en une pyramide laplacienne composée de laplaciens $\Delta V_k(\mathbf{x}, \sigma)$ et d'une image couleur $V_k(\mathbf{x}, \sigma_h)$ au sommet (résolution minimale σ_h). On effectue un rendu par niveau σ de tous les laplaciens $\Delta V_k(\mathbf{x}, \sigma)$, pour obtenir le laplacien cible $\Delta U(\mathbf{x}', \sigma')$ au niveau σ'. Idem pour le sommet de chaque pyramide utilisé pour le rendu de $U(\mathbf{x}', \sigma'_h)$. L'image cible u est alors retrouvée en effondrant la pyramide.*

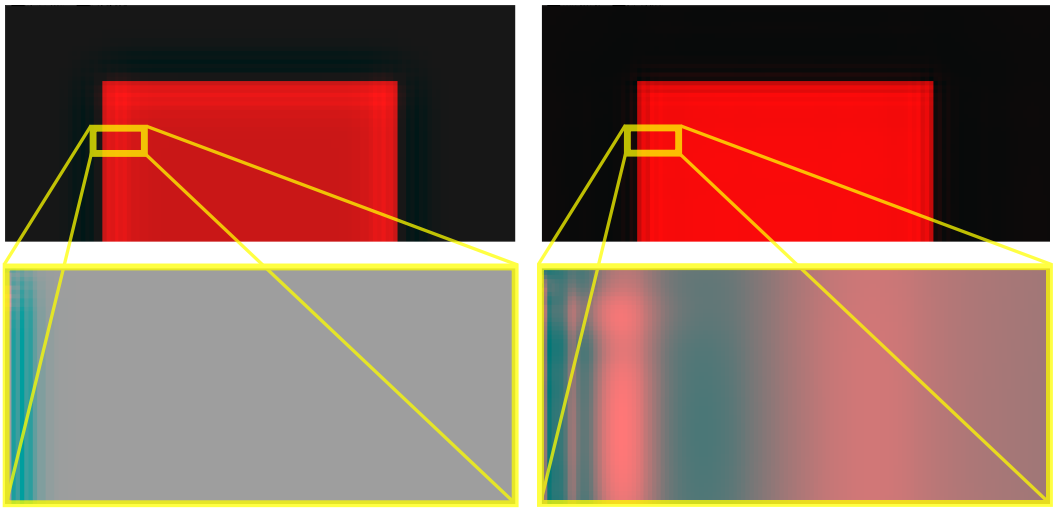


Fig. 4.11 – *Effet du sous-échantillonnage sur le rendu multi-échelles. On a représenté le rendu d'un cube ; la fonction de déformation entre la vue source et la vue cible est un grossissement. Pour effectuer le rendu nous avons naïvement décomposé l'image source selon une pyramide de laplaciens, puis nous avons grossi chaque laplacien pour finalement effondrer la pyramide obtenue. À gauche : sans sous-échantillonnage successif dans la décomposition en espace d'échelle. À droite : avec sous-échantillonnage successif dans la décomposition en pyramide classique. Le sous-échantillonnage doit prendre en compte l'effet de la déformation, sans quoi on observe l'apparition d'artefacts (ici oscillations).*

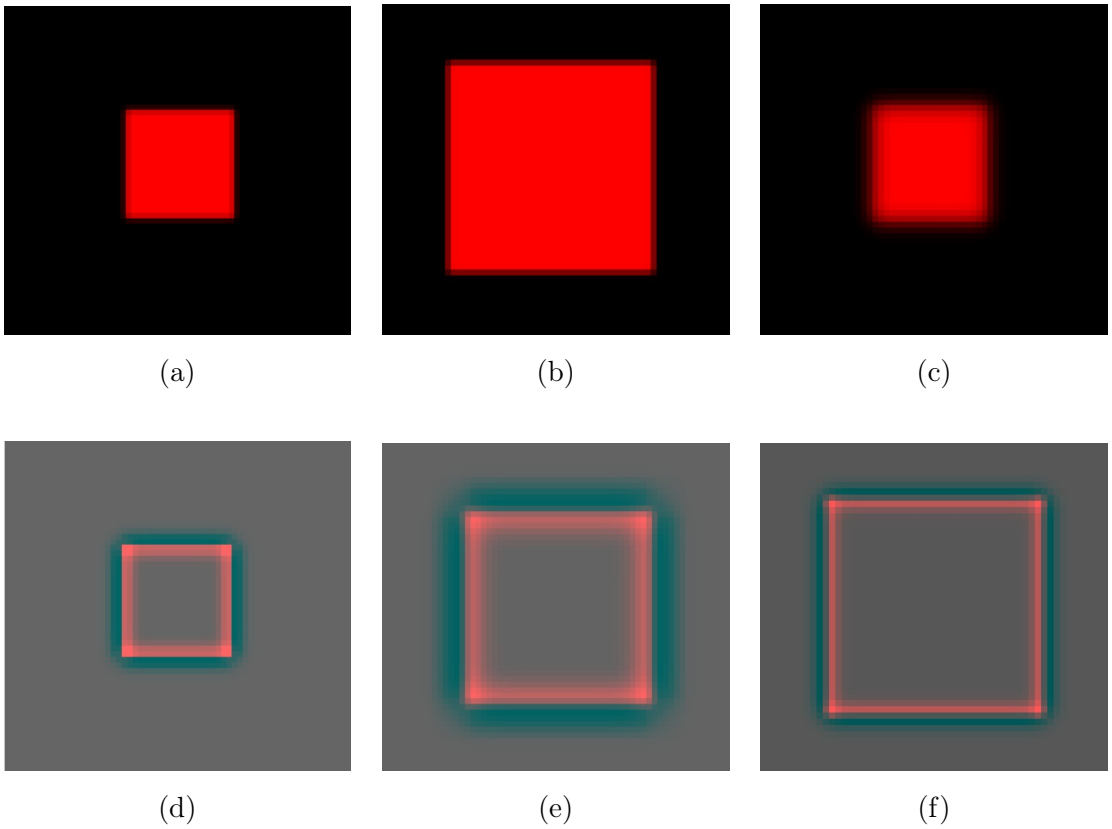


Fig. 4.12 – *Application d'un rendu multi-résolution naïf basé sur des transformation de l'espace d'échelle sur le grossissement d'un cube. (a) Image source à une certaine échelle σ. (b) Vérité terrain de l'image cible, la transformation τ appliquée est un grossissement. (c) Réduit et étendu, l'image source à l'échelle suivante. (d) Laplacien obtenu par DoG à l'échelle σ. (e) Laplacien à l'échelle σ auquel nous avons appliqué le grossissement. (f) Vérité-terrain du laplacien à l'échelle σ, obtenu à partir de (b).*

d'un laplacien induit un changement dans la valeur de ce laplacien. Par conséquent même si le facteur d'échelle d'arrivée est correct, la valeur du laplacien a été changée par la déformation et doit être corrigée. La figure 4.12 illustre ce phénomène en comparant le laplacien de l'image cible vérité-terrain et la dilatation du laplacien d'une image source. Il y a à la fois changement de **résolution** et d'**amplitude** du laplacien. Une conséquence du changement d'échelle est l'introduction des hautes fréquences par la déformation τ_k à des niveaux de résolution censés ne représenter que des basses fréquences. Ces détails non-voulus doivent être filtrés lorsque l'on fait le rendu des niveaux de faible résolution.

Rendu multi-échelles avec filtrage passe-bande La méthode proposée cherche une solution aux problèmes de déformation de l'espace d'échelle précédemment évoqués : une déformation change le facteur d'échelle, celui-ci ne correspond pas forcément à l'un des niveaux de l'espace d'échelle échantillonnés, et enfin la déformation

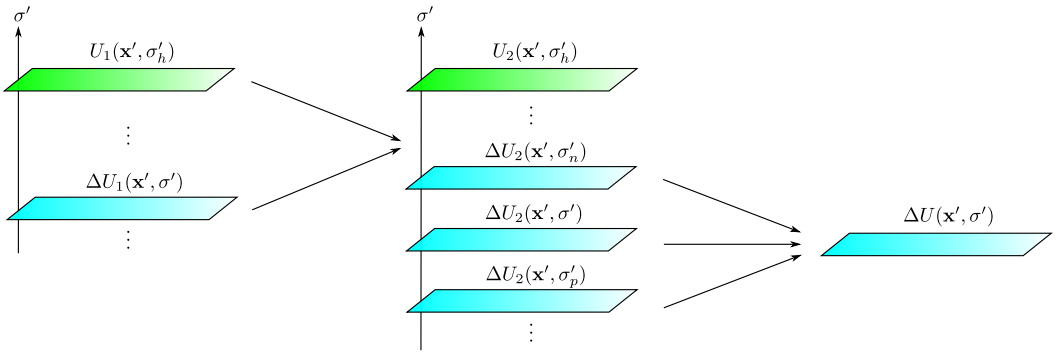


Fig. 4.13 – *Rendu multi-échelles avec filtrage passe-bande. Le rendu d'un laplacien cible à la résolution σ' n'est plus le laplacien final $\Delta U(\mathbf{x}', \sigma')$ mais une image intermédiaire que l'on appelle $\Delta U_1(\mathbf{x}', \sigma')$. Ce laplacien contient des fréquences qui ne correspondent pas à la résolution prévue, voisine de la résolution originale σ. Nous le décomposons en une autre pyramide, pour ne sélectionner que les niveaux voisins de σ' : $\Delta U_2(\mathbf{x}', \sigma'_n)$, $\Delta U_2(\mathbf{x}', \sigma')$ et $\Delta U_2(\mathbf{x}', \sigma'_p)$. Ces laplaciens sont ensuite additionnés pour former le laplacien final $\Delta U(\mathbf{x}', \sigma')$ à la résolution σ'.*

change la valeur des dérivées du signal source en particulier de son laplacien. La valeur du laplacien généré par rendu peut être corrigée en appliquant l'approximation (4.21) : nous divisons la valeur obtenue par le jacobien de la déformation τ_k. Les artefacts oscillatoires illustrés sur la figure 4.11 sont corrigés par l'arrêt des sous-échantillonnages successifs lors de la construction des pyramides : toutes les images de l'espace d'échelle ont alors la même résolution d'image en pixel. Il reste enfin à corriger les changements d'échelle introduits par la déformation. Si le jacobien de la déformation est proche de 1, alors le facteur d'échelle d'arrivée σ' sera voisin de celui de départ σ. Si par contre la déformation τ_k est bruitée, elle introduit des hautes fréquences dans le rendu de laplaciens en basse résolution. L'ajout de ces hautes fréquences à des niveaux de résolution incorrects doit être filtré ; de la même façon pour chaque rendu de laplacien à un niveau σ doivent être filtrées les fréquences trop éloignées de $\frac{1}{\sigma}$ car il est fort probable qu'elles ont été introduites par des artefacts de géométrie présents dans la déformation τ_k.

Reprenons maintenant le rendu d'un niveau σ de la pyramide de laplaciens dans la méthode naïve ; il permet selon elle de générer le laplacien cible à la résolution σ' voisine de σ, car la méthode naïve ne prend pas en compte le changement de résolution. En réalité une déformation, même idéale (sans artefacts), a provoqué un changement d'échelle : ce laplacien cible contient des détails qui ne correspondent pas à la résolution σ. En outre la déformation introduit des fréquences nouvelles car elle est souvent imparfaite (bruit haute fréquence par exemple). Nous décomposons à nouveau ce laplacien en une pyramide pour ne sélectionner que les niveaux voisins de σ (figure 4.13). Il s'agit de considérer que la déformation idéale transforme le facteur d'échelle dans les niveaux voisins du facteur d'échelle initial σ, et d'éliminer toutes les fréquences dégénérées introduites par erreur par l'imperfection de la déformation. Cette sélection de niveaux de résolution voisins fait office de filtre passe-bande centré sur la fréquence du niveau initiale proportionnelle à $\frac{1}{\sigma}$. Les laplaciens aux niveaux sélectionnés sont ensuite additionnés pour former le laplacien cible cherché. Plus le nombre de niveaux sélectionnés est grand, moins le filtrage passe-bande est sélectif,

et plus il est susceptible d'introduire des fréquences dégénérées. Mais si nous ne sélectionnons qu'un seul niveau (le niveau de résolution σ) alors on risque de perdre les parties de l'image qui ont subit un important changement d'échelle (jacobien très différent de 1, dans les zones de compression ou de dilatation). En pratique, nous ne retenons que le niveau σ, le niveau au-dessus σ_p et le niveau au-dessous σ_n (4.13).

4.3.4 Expériences

Les résultats de rendu multi-résolution avec filtrage passe-bande sont exposés sur les figures 4.14b et 4.14d, où nous les comparons avec le résultat d'un rendu classique avec fusion des images en intensité (figures 4.14a et 4.14c), comme présenté dans la section 4.2. L'expérience est réalisée sur le jeu de données *fountain* de Strecha *et al.* (2008). Toutes les vues sont prises en compte pour le rendu excepté la vue centrale qui sert de vue cible à générer. Le *proxy géométrique* employé est un ensemble de cartes de profondeurs estimées par MVE Fuhrmann *et al.* (2014). Le rendu de chaque laplaciens s'effectue selon notre méthode de *splatting* présentée à la section 4.2, où la contribution de chaque *splat* est donnée par le produits des poids de *déformation* et de *géométrie* décrits précédemment.

On observe que la fusion multi-résolution a permis un mélange plus doux des couleurs, avec disparition des artefacts HF dus au bruit des fonctions de déformation. Nous avons cependant constaté sur la figure 4.14e que le filtrage passe-bande avait occasionné une perte de gain des laplaciens, qui résulte en un perte de contraste et un rendu fade de l'image. Nous avons manuellement corrigé cette perte de contraste en augmentant manuellement les valeurs des laplaciens générés, comme le montre la figure 4.14f. Il est toutefois possible d'estimer cette perte de gain pour pouvoir la corriger automatiquement à l'aide de la formule (4.21), qui approxime la résolution d'arrivée du signal en fonction de la jacobienne de la déformation, et permettrait de prévoir la proportion du signal qui a été perdue par le filtrage. Nous laissons cette question en suspend pour de futurs travaux.

Un autre avantage du rendu multi-échelles est qu'il remplit les trous sans que l'on doive appliquer un algorithme d'*inpainting a posteriori*. Ajoutons que ce remplissage est équivalent à une diffusion isotrope, la même que celle produite par le `PUSH/PULL`. En effet, lors du rendu de laplacien, les zones non atteintes par les fonctions de déformation sont laissées nulles. Lorsque la pyramide finale est effondrée, les laplaciens sont additionnés et la couleur se propage d'un niveau à l'autre dans les trous de la même manière que lors de l'étape `PULL` de l'algorithme d'*inpainting*.

4.4 Conclusion

Ce chapitre a permis d'aborder les méthodes directes de rendu basé image. Nous avons vu que pour éviter les artefacts de crénelage et de flou dans les zones où les images sources sont dilatées ou compressées par les fonctions de déformation, il est nécessaire de procéder à un ré-échantillonnage de textures. Nous avons proposé une approche basée sur la projection directe des images sources sur le point de vue cible,

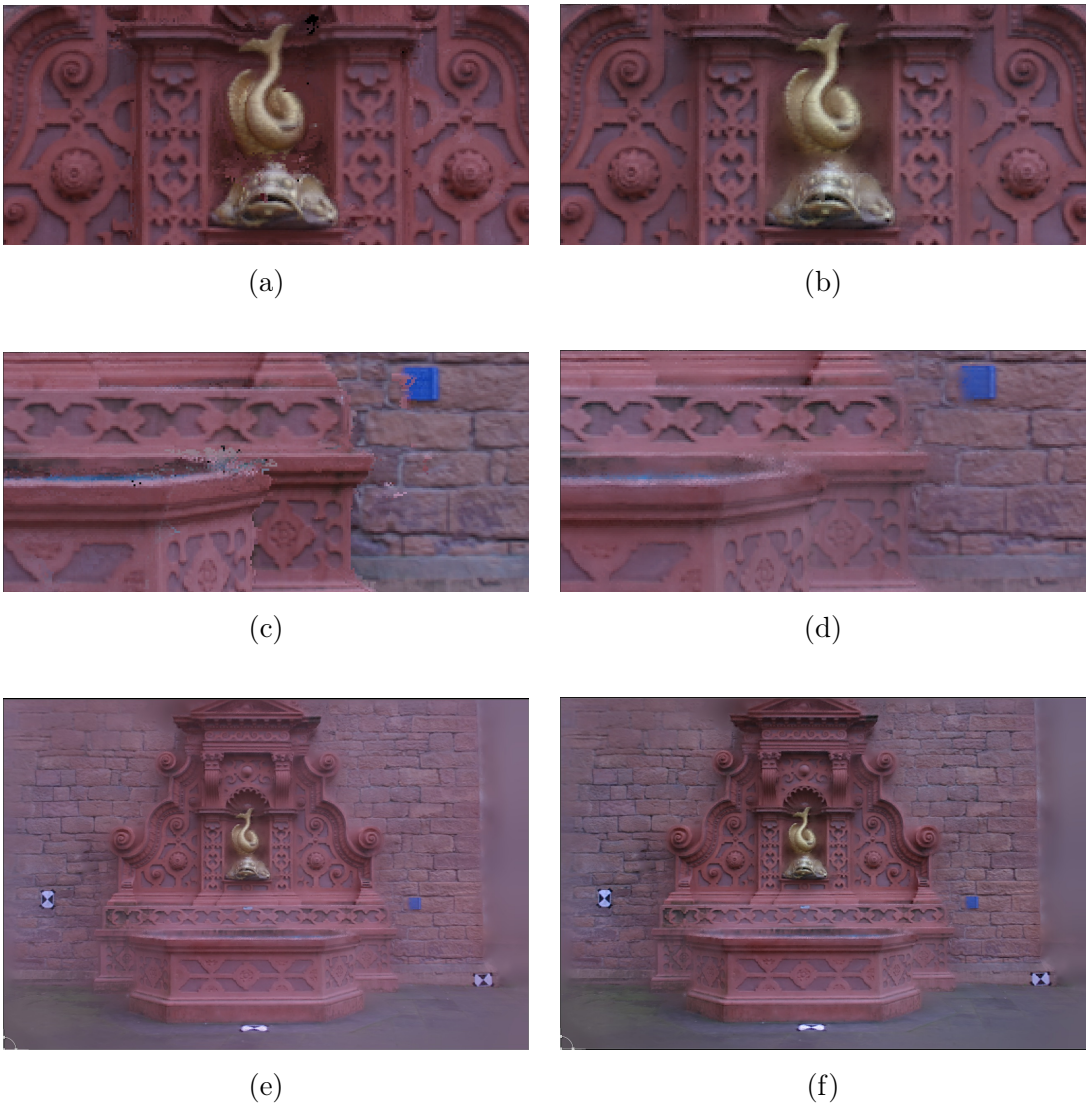


Fig. 4.14 – *Résultats de notre algorithme de rendu multi-résolution. (a, c) Fusion en intensité (avec artefacts HF) (b, d) Rendu multi-résolution (avec filtrage passe-bande). (e) Résultat sur la vue centrale de fountain. (f) Compensation de la perte de gain due au filtrage.*

par le biais d'éléments de surface finis appelés *splats*. Dans cette optique, un ré-échantillonnage consiste à adapter la forme des *splats* en fonction de la déformation afin de mieux interpoler l'image cible. Cette méthode de *splatting* est utilisée pour effectuer le rendu de laplacien à un niveau de résolution donné. En décomposant chaque image source en une pyramide laplacienne pour rendre séparément chaque niveau de résolution, nous combinons les images sources à toutes les fréquences, dans le but de prévenir l'apparition d'artefacts de rendu liés aux imperfections du *proxy géométrique*. Mais nous avons déduit d'une analyse de la déformation de l'espace d'échelle qu'un simple rendu multi-échelles naïf ne peut fonctionner. Nous avons donc proposé un nouvel algorithme de rendu multi-échelles qui prend en compte les effets des fonctions de déformation sur le laplaciens et l'espace d'échelle. Cette approche nous procure des résultats très satisfaisants d'un point de vue visuel, mais possède encore des défauts dus à la modification du contenu fréquentiel de chaque niveau par la fonction de déformation. Le chapitre suivant propose une autre formulation du problème de rendu basé image qui permet de corriger les artefacts de rendu produits par l'imperfection du *proxy géométrique*.

5 Rendu variationnel

5.1 Motivations

La méthode directe présentée au chapitre précédent n'offre aucun contrôle sur la solution obtenue en terme de proximité avec les données. Au contraire, une méthode *variationnelle* ou *inverse* repose sur la minimisation d'une énergie dans le but de trouver la solution la plus proche des images sources (les données). Il s'agit d'estimer les *paramètres* (la valeur des pixels de l'image cible) d'un *modèle* (notre image cible) à l'aide des données observées (les images sources). Dans le but d'éviter le *sur-ajustement* du modèle aux données, on a souvent recourt à un terme de régularisation permettant plus de contrôle sur la solution. Le problème de rendu basé image est mal posé car il est sous-contraint et par conséquent il existe une infinité de solutions possibles. L'ajout d'un terme de régularisation outre de contraindre la solution, pour un meilleur conditionnement du système à résoudre. Néanmoins si le terme de régularisation a trop d'influence, on risque l'éloignement des données et le *sous-ajustement* du modèle. Plutôt que de chercher le terme de régularisation idéal, nous proposons de résoudre le problème en exploitant mieux les données observées, pour renforcer les contraintes sur la solution, afin d'obtenir un système mieux conditionné sans risque de sur-ajustement.

Le but de notre approche variationnelle est d'estimer une image u – une fonction qui à tout pixel, ou point image 2D, associe une couleur – à partir des données, les k images sources appelées v_k^*. L'estimateur $\hat{u}$ de la solution u doit maximiser la probabilité *a posteriori* d'observer l'image cherchée u sachant nos données v_k^* : on l'appelle estimateur MAP (*Maximum a posteriori*). Ne sachant pas calculer cette probabilité *a posteriori* nous l'exprimons à l'aide du théorème de Bayes en fonction de la vraisemblance et des probabilités *a priori*. En prenant nos probabilités indépendantes et normalement distribuées, on arrive à la conclusion que la solution $\hat{u}$ doit minimiser la somme de deux termes E_{data} et E_{prior}. La méthode est aussi qualifiée d'*inverse*, car ne sachant pas exprimer la solution en fonction des données, on cherche plutôt à exprimer les données en fonction de la solution, supposée connue. La solution cherchée s'obtient par résolution d'un système linéaire, issu d'une

formulation bayésienne du problème. Notre travail s'inspire de Pujades *et al.* (2014), qui repose lui-même sur la formulation bayésienne de Goldluecke et Cremers (2009); Wanner et Goldluecke (2012b).

Chaque pixel de la solution est le résultat de la contribution de plusieurs pixels des vues sources. Ces contributions sont pondérées par des termes qui apparaissent dans l'expression de l'énergie. Pujades *et al.* (2014) ont montré que les poids des pixels des images sources dans l'énergie pouvaient se déduire formellement des propriétés de la caméra, du contenu de l'image, et de la précision du *proxy géométrique*, amenant une nouvelle formalisation des heuristiques de poids proposées initialement par Buehler *et al.* (2001). La plupart des « propriétés désirables qu'un algorithme idéal de rendu basé image devrait avoir »(Buehler *et al.*, 2001) eurent alors une explication formelle, sauf la propriété de *continuité*. En effets les contributions des vues sources varient brutalement d'un pixel de la vue cible à l'autre, soit parce que la limite du champ de vue d'une caméra est atteinte ou qu'elle est occultée (sa contribution tombe à 0), soit parce que les contributions ne sont pas lisses au sein même de la même image source, puisque l'estimation d'un *proxy géométrique* est bruitée et que le calcul des poids des contributions repose la reconstruction du *proxy*.

Dans ce chapitre, nous montrons qu'une façon d'éviter ces artefacts est d'imposer des contraintes supplémentaires sur le gradient de l'image synthétisée. Ces contraintes viennent d'une simple observation : les contours d'image dans la vue cible doivent aussi être des contours dans les images sources où ces parties sont visibles. Une fonctionnelle d'énergie similaire à celle de Pujades *et al.* (2014) est développée, composée de l'habituel terme sur les données (*data term*) et un terme de régularisation (*smoothness term*) ; mais le terme sur les données contient un terme additionnel qui prend en compte les contraintes sur les gradients. Nous montrons que tenir compte à la fois de l'intensité et du gradient dans les méthodes de rendu basé image apporte une solution élégante au renforcement de la propriété de *continuité* initialement énoncée par Buehler *et al.* (2001).

5.2 État de l'art

« *The ill-posed nature of superresolution, combined with the fast convergence of the conjugate gradient algorithm, results in oscillatory artifacts or "null objects" which must be dealt with by regularization.* » (Connolly et Lane, 1997)

Super-résolution L'usage de méthodes variationnelles dans le rendu basé image s'inspire de la littérature en super-résolution et en débruitage (*denoising, deblurring*) (Cristóbal *et al.*, 2008), en particulier le problème de déconvolution multi-vues aveugle (Sroubek *et al.*, 2007; Harmeling *et al.*, 2010; Faramarzi *et al.*, 2013). Celui-ci consiste à estimer une image à partir d'un certain nombre de vues rectifiées d'un même objet, moins bien résolues, sans connaître le noyau de flou qui permet de passer de l'image super-résolue à une image source. Le problème de super-résolution est un sous-problème de la reconstruction du champ de lumière, où seul le domaine spatial est à reconstruire et où les vues sont généralement translatées les unes par

rapport aux autres de moins d'un pixel. Dans le contexte de l'imagerie plénoptique, on peut citer les travaux de Bishop *et al.* (2009); Bishop et Favaro (2012); Mitra et Veeraraghavan (2012). La super-résolution ne se limite pas à la génération de textures planes, comme l'attestent Goldluecke et Cremers (2009) dont l'objectif est de texturer le modèle 3D d'un objet à partir d'une collection d'images sous-échantillonnées.

Estimation MAP Le problème de super-résolution étant mal-conditionné, Baker et Kanade (2002) suggèrent l'ajout d'un terme de régularisation, un *a priori* sur les images que l'on souhaite obtenir, en l'occurrence des visages dans leur application de reconnaissance faciale. En modélisant l'image par une distribution de Gibbs inspirée de la physique (Geman et Geman, 1984), on cherche à minimiser une énergie, somme d'un terme d'attache aux données hérité de la vraisemblance et d'un terme de régularisation hérité de notre *a priori* sur la solution. Cet *a priori* est crucial car il apporte plus de contraintes au problème très mal-conditionné qu'est la super-résolution. En exploitant cette information suplémentaire *a priori*, Hardie *et al.* (1997, 1998); Shen *et al.* (2007) proposent de résoudre simultanément le problème de super-résolution et d'alignement des images sources les unes par rapport aux autres, ou de l'estimation du mouvement de la caméra dans le but d'aligner les prises de vue. Dans notre cas les caméras sont déjà calibrées et la géométrie de la scène est déjà estimée ; mais le problème n'est pas plus facile, car contrairement au contexte de la super-résolution nous proposons de générer une vue totalement inédite.

Contraindre la solution L'idée-clé pour un rendu en haute résolution est que la qualité de l'image solution dépend souvent de la contrainte de l'espace de recherche. Par conséquent, trouver la bonne régularisation ou *a priori* sur la solution est une question cruciale pour l'obtention d'images d'excellente qualité. La contribution principale de Fitzgibbon *et al.* (2003) est l'utilisation d'*a priori* calculés à partir de grandes bases de données de textures afin de contraindre la solution, indépendamment des données relatives à la scène. Cette idée fut récemment étendue par Flynn *et al.* (2015) qui effectuent une synthèse de nouvelle vue à partir d'un réseau de neurones, entraîné par une gigantesque base d'images prises tout autour du monde. Au contraire, notre méthode ne repose pas sur de forts *a priori* sur la nouvelle image à synthétiser, mais tente plutôt de mieux exploiter les données procurées par les images sources pour ajouter de nouvelles contraintes sur la solution. En somme, notre algorithme ne requiert pas ces énormes bases de données pour produire des images de haute qualité. Notons que les deux approches ne sont pas incompatibles. Kalantari *et al.* (2016) proposent une approche similaire à celle de Flynn *et al.* (2015) appliquée au rendu basé sur les champs de lumière épars.

Rendu bayésien Nos travaux s'inspirent de Wanner et Goldluecke (2012b), dont l'idée clé consiste à se débarrasser des heuristiques et du réglage manuel des paramètres. La contribution de chaque vue dans l'estimation de la solution est automatiquement déduite d'équations mathématiques. Pujades *et al.* (2014) intègrent l'incerti-

tude géométrique dans le formalisme bayésien de Wanner et Goldluecke (2012b). Ils obtiennent alors de nouveaux poids qui favorisent les caméras satisfaisant à la fois la cohérence épipolaire (*epipole consistency*) et la déviation angulaire minimale (*minimal angular deviation*), deux principes établis par Buehler *et al.* (2001) pour décrire l'algorithme d'IBR idéal. Cependant leur méthode n'offre pas de cadre formel pour satisfaire le principe de continuité (*continuity principle*), en particulier près des limites du champ de vue des caméras. Nous montrons qu'introduire un terme additionnel dans la fonctionnelle énergie, qui contraint non seulement les intensités mais aussi les gradients de la solution, apporte une solution élégante au principe de continuité.

5.3 Modèle de formation de l'image

Nous détaillons ici les étapes de l'élaboration d'un modèle de formation de l'image à la résolution du système en passant par l'expression de la fonctionnelle énergie telle qu'elle a été présentée par Wanner et Goldluecke (2012b).

Comme on suppose en général dans la littérature sur la super-résolution (Baker et Kanade, 2002; Hardie *et al.*, 1997), la valeur d'intensité $v_k(\mathbf{x}_m)$ d'un point $\mathbf{x}_m$ dans l'image source k peut s'écrire comme la convolution de l'image cible avec la fonction d'étalement du point (PSF, pour *Point Spread Function* en anglais), notée b. Étant donnée une image idéale u au point de vue cible, définie sur le domaine Γ, et une déformation τ_k des points de Ω_k dans Γ, si nous mettons de côté les occultations pour le moment, l'intensité de l'image observée peut s'écrire comme la relation de convolution

$$v_k(\mathbf{x}_m) = \int_{\Omega_k} u \circ \tau_k(\mathbf{x})\, b(\mathbf{x} - \mathbf{x}_m)\, \mathrm{d}\mathbf{x}, \tag{5.1}$$

ou plus simplement $v_k = b * (u \circ \tau_k)$, où $*$ est l'opérateur de convolution.

La PSF $b : \Omega_k \to [0, 1]$ est la densité de probabilité qui peut s'écrire $b_k : \Gamma \to [0, 1]$ par changement de variable $\mathbf{x}' = \tau_k(\mathbf{x})$ de telle façon que

$$v_k(\mathbf{x}_m) = \int_{\Gamma} u(\mathbf{x}') b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\, \mathrm{d}\mathbf{x}'. \tag{5.2}$$

Il y a plusieurs manières de calculer la PSF transformée b_k, selon le modèle utilisé pour la PSF des images sources b. L'hypothèse la plus commune est de considérer la PSF comme étant une gaussienne 2D d'espérance $\mathbf{x}_m$ et de covariance $\mathbf{\Sigma}$. Un modèle plus simple de la PSF est de se représenter un pixel carré et uniformément sensible à la lumière, la PSF étant alors une densité uniforme. En notant A l'aire du pixel centré sur $(0, 0)$ dans une vue source k, on obtient

$$b(x, y) = \begin{cases} \frac{1}{A^2} & \text{if } -\frac{1}{A} \leq x, y \leq \frac{1}{A} \\ 0 & \text{sinon.} \end{cases} \tag{5.3}$$

Sous l'hypothèse que le *warp* τ_k est localement linéaire, la PSF transformée est un parallélogramme uniformément distribué (figure 5.1). Dans ce cas, on peut faire une

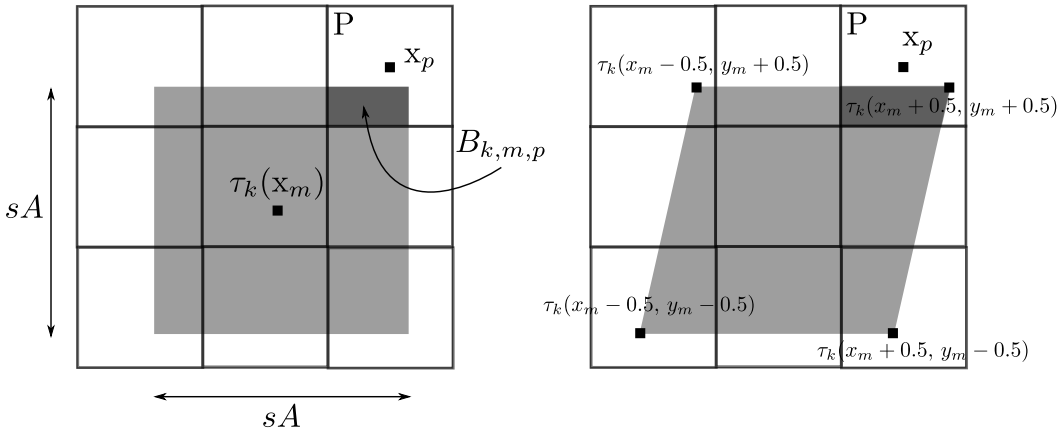


Fig. 5.1 – *L'aire de la projection du pixel est colorée en gris. À gauche : Nous supposons que la transformée de la PSF est toujours carrée ; l'intensité résultante est la moyenne des valeurs de l'image cible pondérée par les coefficients $B_{k,m,p}$ – l'intersection (en gris foncé) entre l'aire en gris et le pixel en haute résolution. À droite : Modèle plus précis où nous supposons seulement que le* warp *est localement linéaire et que la PSF transformée est obtenue en transformant chaque coin du pixel source.*

hypothèse encore plus forte et supposer que le *warp* préserve les pixels (leur aire et leur forme), ce qui est faux en réalité mais simplifie grandement l'implémentation. Désormais nous prendrons des pixels d'aire unitaire. Puisque l'intensité est constante et égale à $u(\mathbf{p})$ sur toute la surface du pixel $\mathbf{p}$ dans la vue cible, la relation de convolution (5.1) ci-dessus peut être écrite comme l'ont fait Hardie *et al.* (1997) :

$$v_k(\mathbf{x}_m) = \sum_{\mathbf{p}\in\Gamma} u(\mathbf{p}) \int_{\mathbf{p}} b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\,\mathrm{d}\mathbf{x}', \tag{5.4}$$

et l'intensité du pixel dans l'image source est

$$v_k(\mathbf{m}) = \sum_{\mathbf{p}\in\Gamma} B_{k,m,p} u(\mathbf{p}), \tag{5.5}$$

où $B_{k,m,p} = \int_{\mathbf{p}} b_k(\mathbf{x}' - \tau_k(\mathbf{x}_m))\,\mathrm{d}\mathbf{x}'$ est l'aire de l'intersection entre la projection du pixel dans la vue cible et le pixel $\mathbf{p}$ de cette même vue. Si l'échantillonnage des vues de départ et d'arrivée sont les mêmes, alors les aires d'intersection sont les coefficients bilinéaires : c'est équivalent à interpoler bilinéairement les intensités de u.

Notons que $\mathbf{p}$ désigne un pixel, donc un échantillon discret, tandis que Γ est un domaine continu. Il est donc inexact de sommer sur tous les $\mathbf{p} \in \Gamma$, mais pour des questions de lisibilité nous appellerons également Γ le domaine discret de la vue cible. De la même manière on notera $\mathbf{m} \in \Omega_k$ pour les domaines discrets des vues sources.

Discrétisation En pratique les images sont numériques, d'où la nécessité de discrétiser les équations ci-dessus. Soit $\mathbf{V}$ le vecteur de tous les pixels de toutes les

vues sources mis dans une seule grande colonne $(v_0(0), v_0(1), ..., v_{K-1}(M-1))$, $\mathbf{U}$ le vecteur colonne solution $(u(0), ..., u(N-1))$, et $\mathbf{B}$ la matrice de taille $KM \times N$ qui contient les coefficients $B_{k,m,p}$. Pour chaque pixel $\mathbf{m}$ de chaque vue source k nous obtenons une équation similaire à (5.5). Nous déduisons donc de (5.5) que

$$\mathbf{V} = \mathbf{BU}. \tag{5.6}$$

Ce système linéaire n'a généralement pas de solution exacte en $\mathbf{U}$, car il possède plus d'équations linéairement indépendantes que d'inconnues. Dans la suite nous montrons comment obtenir un estimateur de la solution. On verra que cet estimateur est solution d'un problème des moindres carrés associé à (5.6) (terme de régularisation mis à part).

5.4 Estimateur MAP (maximum *a posteriori*)

Formalisme bayésien Le but de l'approche variationnelle est d'estimer une image u à partir des données $(v_k^*)_{k\in[1..K]}$, où K est le nombre de vues sources. L'estimateur $\hat{u}$ de la solution u doit maximiser la probabilité *a posteriori* d'observer u sachant nos données en entrée. On l'appelle estimateur MAP (Maximum *a posteriori*) :

$$\hat{u} = \arg\max_u P(u|(v_k^*)_{k\in[1..K]}). \tag{5.7}$$

Ne sachant pas calculer cette probabilité *a posteriori* nous l'exprimons autrement à l'aide du théorème de Bayes en fonction de la vraisemblance et des probabilités *a prori*. On fait l'hypothèse que les v_k sont conditionnellement indépendants. Le terme $P((v_k^*)_{k\in[1..K]})$, appelé *évidence*, ne dépend pas de u et peut donc être retiré de l'équation :

$$\hat{u} = \arg\max_u \frac{P((v_k^*)_{k\in[1..K]}|u)\,P(u)}{P((v_k^*)_{k\in[1..K]})} = \arg\max_u \prod_{k\in[1..K]} P((v_k^*)|u)\,P(u). \tag{5.8}$$

C'est en cela que la méthode est qualifiée d'inverse : ne sachant pas exprimer la solution en fonction des données, on cherche plutôt à exprimer les données en fonction de la solution supposée connue.

Vraisemblance Le terme de vraisemblance [1] $P((v_k^*)_{k\in[1..K]}|u)$ est la probabilité d'obtenir les données (images sources) supposant connue la solution u. Elle s'exprime comme le produit des probabilités $P(v_k^*|u)$ dont la loi est prise normale ; chacune de ces probabilités est égale à une exponentielle à un facteur strictement positif près :

$$P(v_k^*|u) \propto e^{-E_{\text{color},k}(u)}. \tag{5.9}$$

$E_{\text{color},k(u)}$ est appelé fonctionnelle énergie par analogie avec la physique. Il s'agit d'un terme des moindres carrés représentant la somme des écarts aux images sources en

1. *Likelihood* en anglais

terme d'intensité (couleur du pixel). Il est aussi appelé terme d'attache aux données, E_{data} dans la littérature.

$$E_{\text{color},k}(u) = \frac{1}{2}\int_{\Omega_k} \omega_k(u)\,(b * (u \circ \tau_k) - v_k^*)^2 \,\mathrm{d}\mathbf{x}. \tag{5.10}$$

Le termes $\omega_k(u)$ sont les contributions par pixel de chaque image source. Ils dépendent du gradient de la solution courante u et de l'incertitude sur la géométrie reconstruite. Une formule explicite de ces contributions (3.25) est donnée dans le chapitre 4. En supposant que le bruit du capteur est gaussien et identique pour toutes les images, nous notons $\sigma_{s,k}^2 = \lambda$ sa variance, une constante strictement positive. La vraisemblance étant le produit des $P(v_k^*|u)$, elle s'exprime comme

$$P((v_k^*)_{k\in[1..K]}|u) \propto e^{-\frac{1}{\lambda}E_{\text{color}}(u)}, \text{ avec } E_{\text{color}}(u) = \sum_{k=1}^{K} E_{\text{color},k}(u). \tag{5.11}$$

En reprenant les notations matricielles précédentes (5.6), on exprime l'énergie (5.11) comme un système linéaire :

$$E_{\text{color}}(\mathbf{U}) = (\mathbf{BU} - \mathbf{V}^*)^\intercal \mathbf{W}(\mathbf{BU} - \mathbf{V}^*), \tag{5.12}$$

où $\mathbf{W}$ est une matrice $KM \times KM$ diagonale qui contient les poids des contributions. Pour minimiser cette énergie nous dérivons le système linéaire, et obtenons les équations normales qui nous apportent un estimateur de la solution $\hat{\mathbf{U}}$:

$$\mathbf{B}^\intercal\mathbf{W}\mathbf{B}\hat{\mathbf{U}} = \mathbf{B}^\intercal\mathbf{W}\mathbf{V}^*. \tag{5.13}$$

La matrice $\mathbf{B}^\intercal\mathbf{W}\mathbf{B}$ n'est en général pas inversible. Le système linaire peut être résolu par n'importe quelle méthode des moindres carrés linéaires.

Probabilité *a priori* La probabilité *a priori* $P(u)$ représente notre connaissance *a priori* sur l'image à synthétiser. Nous savons que cette dernière est naturelle, donc comporte peu de variations de gradient : le signal est *régulier*. Dans notre travail, nous utilisons un *a priori* de variation totale (Goldluecke et Cremers, 2010), norme L^1. Très utilisé en débruitage (figure 5.2), il possède plusieurs avantages : convexe, plus simple que d'autres *a priori* d'image comme ceux de Cho *et al.* (2012); Sun *et al.* (2008), il préserve en outre les bords abruptes et contours de l'image. Une preuve de convergence est fournie par Chambolle (2004).

$$P(u) \propto e^{-E_{\text{prior}}(u)}, \text{ avec } E_{\text{prior}}(u) = \int_\Gamma |\nabla u|. \tag{5.14}$$

E_{prior}, souvent appelé terme de régularisation, vient de la probabilité *a priori* de l'image. D'autres termes de régularisation sont bien entendu possibles, comme le récent TGV (*Total Generalized Variation*) (Bredies *et al.*, 2010) qui supprime les artefacts connus sous le nom d'effet d'escalier (*Staircasing effect* en anglais.

Fig. 5.2 – *Variation totale L^1 dans la littérature de débruitage.*

Énergie à minimiser On rappelle que l'estimateur MAP $\hat{u}$ doit maximiser la probabilité *a posteriori* :

$$\hat{u} = \arg\max_u P(u|(v_k^*)_{k\in[1..K]}) \tag{5.15}$$

$$\hat{u} = \arg\min_u -\ln P(u|(v_k^*)_{k\in[1..K]}) \tag{5.16}$$

$$\hat{u} = \arg\min_u -\ln P((v_k^*)_{k\in[1..K]}|u) - \ln P(u) \tag{5.17}$$

$$\hat{u} = \arg\min_u \frac{1}{\lambda} E_{\text{color}}(u) + E_{\text{prior}}(u). \tag{5.18}$$

Le paramètre λ strictement positif permet de contrôler la prépondérance du terme de lissage dans l'énergie. Notre solution $\hat{u}$ doit minimiser

$$E(u) = E_{\text{color}}(u) + \lambda E_{\text{prior}}(u). \tag{5.19}$$

λ pondère l'influence du terme de régularisation. Une augmentation de ce facteur résulte en un lissage de la solution finale en prévenant l'apparition de hautes fréquences. En effet le terme de régularisation permet d'éviter les problèmes de sur-ajustement : lorsque le modèle est trop proche des données, par sa complexité (nombre de paramètres à estimer par rapport à la quantité de données disponibles), son estimation perd en pouvoir de prédiction. Il est alors plus instable, plus sensible au bruit des données mais aussi des artefacts du *proxy*. C'est un problème récurrent dans la littérature des problèmes inverses, lorsque ceux-ci sont mal posés. Cependant une influence trop forte du terme de régularisation, due à un facteur λ élevé, peut nuire à l'estimation, qui sera alors plus éloignée des données, ce qui se traduit par une perte des détails contenus dans l'image. Vanhoey *et al.* (2013) propose une approche pour trouver un compromis entre sous-ajustement et sur-ajustement, entre stabilité et préservation des détails des images sources.

5.5 Ajout de contraintes sur les gradients

Dans cette section, nous montrons qu'une façon d'éviter ces artefacts est d'imposer des contraintes supplémentaires sur le gradient de l'image synthétisée.

Fusion d'images dans le domaine du gradient La fusion d'images dans le domaine du gradient a reçu beaucoup d'intérêt ces dernières années, en commençant par l'article phare de Pérez *et al.* (2003), pour des applications dans l'édition d'images (McCann et Pollard, 2008), l'*inpainting* (Levin *et al.*, 2003) ou les panoramas (Agarwala *et al.*, 2004; Zomet *et al.*, 2006). Le travail le plus proche du notre en rendu basé image est probablement celui de Kopf *et al.* (2013), qui effectue un rendu de gradient, suivi d'une intégration pour produire la couleur de l'image. Plus récemment Kopf *et al.* (2014) ont proposé un algorithme de stabilisation de vidéo prise à la première personne en trois étapes : une reconstruction du *proxy géométrique* par SfM et de cartes de profondeur denses, un calcul du chemin de caméra qui interpole les vues sources, et enfin la génération de points de vue qui suivent ce chemin. Le rendu consiste à projeter les images sources directement sur la vue cible, en sélectionner quelques-unes et les mélanger en choisissant le pixel source qui minimise une énergie. Le mélange des couleurs est ensuite égalisé par fusion dans le domaine des gradients en résolvant des équations de Poisson spatio-temporelles.

Les cartes de profondeur estimées sont bruitées et incomplètes. Le bruit fait référence à la variance élevée et aux nombreux *outliers* des carte de profondeur, dus à l'incertitude de localisation. Il peut être corrigé par un lissage ou un seuillage, au prix d'une perte de précision et d'une plus grande sparsité du modèle. Les discontinuités sont dues aux limites des champs de vue des caméras et aux auto-occultations. Par conséquent les $\omega_k(u)$ dans (5.10) et les τ_k peuvent être aussi bruités et discontinus, ce qui résulte en des artefacts dans la solution finale qui apparaissent comme de faux bords ou textures (figure 5.3). La méthode de rendu basé image devrait empêcher ces contours d'apparaître : en fait, un contour synthétisé dans l'image solution devrait également être présent dans les images sources, là où ces parties de la scène sont visibles.

Pour renforcer cette propriété, nous ajoutons un terme supplémentaire $E_{\text{grad}}(u)$, qui force la solution courante à se rapprocher des données dans le domaine du gradient. Ce terme permet en outre d'ajouter de nouvelles contraintes au système qui est alors mieux conditionné. Nous l'obtenons à partir de $P((\nabla v_k^*)_{k\in[1..K]}|\nabla u)$, la probabilité d'obtenir les gradients des images sources, sachant le gradient de l'image cible. Nous supposons que les variables aléatoires ∇v_k sont indépendantes,

Fig. 5.3 – *Les discontinuités des* warps τ_k *et des poids* ω_k *provoquent des artefacts. À gauche: la vue globale d'une scène de Strecha* et al. *(2008) estimée avec l'énergie donnée par l'équation 5.19. En haut à droite: un zoom révèle des artefacts haute-fréquence. En bas à droite : une carte de profondeur présentant des discontinuités dues à la visibilité à cet endroit de l'image.*

identiquement distribuées et obéissent à une loi normale, d'où

$$E_{\text{grad}}(u) \propto -\ln P((\nabla v_k^*)_{k\in[1..K]}|\nabla u) \tag{5.20}$$

$$E_{\text{grad}}(u) \propto -\sum_{k=1}^{K} \ln P(\nabla v_k^*|\nabla u) \tag{5.21}$$

$$E_{\text{grad}}(u) = \sum_{k=1}^{K} \frac{1}{2} \int_{\Omega_k} (\nabla v_k - \nabla v_k^*)^2 \, \mathrm{d}\mathbf{x} \tag{5.22}$$

$$E_{\text{grad}}(u) = \sum_{k=1}^{K} \frac{1}{2} \int_{\Omega_k} (\nabla (b * (u \circ \tau_k)) - \nabla v_k^*)^2 \, \mathrm{d}\mathbf{x}. \tag{5.23}$$

Trouver u qui minimise ce terme d'énergie particulier est alors équivalent à résoudre un système de K équations de Laplace :

$$\Delta(b * (u \circ \tau_k) - v_k^*) = 0, \tag{5.24}$$

où $\Delta = \nabla.\nabla$ représente le laplacien de l'image. On en déduit alors la différentielle de la fonctionnelle :

$$\mathrm{d}E_{\text{grad}}(u) = (|\frac{\partial \tau_k}{\partial z}|^{-1} \, \bar{b} * (\Delta(b * (u \circ \tau_k)) - \Delta v_k^*)) \circ \beta_k. \tag{5.25}$$

Les β_k sont les déformations inverses qui apparaissent à cause du changement de variable dans l'intégrale. $\bar{b}$ est l'adjoint de la PSF b. Les déformations τ_k sont celles qui ont été estimées auparavant, et manquent donc de précision. Cette incertitude a un effet néfaste sur le calcul de $\Delta(b * (u \circ \tau_k))$. Par conséquent, nous choisissons de calculer le laplacien de u d'abord, puis de le transformer dans le domaine Ω_k. Sous

l'hypothèse que les déformations τ_k sont localement linéaires, on peut négliger leurs dérivées au deuxième ordre et obtenir

$$\Delta(b * (u \circ \tau_k)) = b * \left(\frac{\partial \tau_k}{\partial x}^\intercal \mathbf{H}u \frac{\partial \tau_k}{\partial x} + \frac{\partial \tau_k}{\partial y}^\intercal \mathbf{H}u \frac{\partial \tau_k}{\partial y} \right), \tag{5.26}$$

où $\mathbf{H}u = \frac{\partial \nabla u}{\partial \mathbf{x}}$ est la hessienne de u.

Les cartes de profondeur mal estimées causant de fortes discontinuités dans les correspondances entre les vues, la hessienne peut être très instable. Pour l'implémentation et dans ce cas seulement, nous supposons que $\tau_k(\mathbf{x}) \approx \mathbf{R}\mathbf{x} + \mathbf{t}$, où $\mathbf{R}$ est une rotation dans le plan et $\mathbf{t}$ est une translation 2D, de telle façon que

$$\Delta(b * (u \circ \tau_k)) = b * (\mathrm{Tr}(\mathbf{H}u) \circ \tau_k) = b * (\Delta u \circ \tau_k). \tag{5.27}$$

Nous prouvons que l'équation (5.27) est valable pour les rotations et les translations (et ces déformations seulement) par le lemme de la section 4.3.2. La forme finale de l'énergie à minimiser est donc

$$E(u) = \alpha E_{\text{color}}(u) + \gamma E_{\text{grad}}(u) + \lambda E_{\text{prior}}(u). \tag{5.28}$$

De même que le terme sur les couleurs de l'image, le terme portant sur les gradients est

$$E_{\text{grad}}(\mathbf{U}) = (\mathbf{B}\nabla\mathbf{U} - \nabla\mathbf{V}^*)^\intercal(\mathbf{B}\nabla\mathbf{U} - \nabla\mathbf{V}^*) \tag{5.29}$$

et se dérive identiquement. Nous minimisons la fonctionnelle (5.28) via FISTA (*Fast Iterative Shrinkage Thresholding Algorithm*) (Beck et Teboulle, 2009), un algorithme implémenté dans la bibliothèque d'optimisation continue pour traitement de champs de lumière `http://cocolib.net` (Goldluecke *et al.*, 2012; Wanner et Goldluecke, 2014).

5.6 Expériences et validations

Afin de mettre en évidence l'influence du terme utilisant les gradients des images sur la qualité de la vue synthétisée, plusieurs expériences ont été conduites sur des scènes synthétiques et réelles, pour des placements de caméras structurés ou non.

Il est important de préciser que les parties de la vue cible qui ne sont visibles par aucune des vues sources sont remplies par l'algorithme `PUSH/PULL` de Gortler *et al.* (1996) décrit dans le chapitre 4. L'effet provoqué est celui d'un remplissage par diffusion de ces zones (figures 5.4 et 5.5).

Concernant le *proxy géométrique*, dans les expériences qui suivent nous avons opté pour une représentation en cartes de profondeur (Fuhrmann *et al.*, 2014) car elles sont un bon compromis entre précision et exhaustivité de reconstruction. En effet, la reconstruction d'un nuage de points à l'aide d'un algorithme de l'état de l'art (Furukawa et Ponce, 2010) est économe en données et très précise mais les données sont éparses. D'autre part, si une reconstruction de surface (Kazhdan *et al.*, 2006; Fuhrmann et Goesele, 2014) est faite à partir du nuage de points dans le but de

densifier les correspondances entre les vues, la précision de la géométrie diminue. Les cartes de profondeur offrent en outre l'avantage d'établir immédiatement les correspondances τ_k entre tout point $\mathbf{x}_m$ d'une vue source v_k son projeté $\mathbf{x}_p$ sur la vue cible u. Nous renvoyons au chapitre 3 pour plus de détails sur le choix du *proxy*.

5.6.1 Base de données structurées

Les premières expériences ont été réalisées à partir d'une base de données d'images *light-field* (Wanner, Sven *et al.*, 2013), prises du *HCI Light Field Benchmark Datasets* et de la *Stanford Light Field Archive*. Pour chaque base d'images nous utilisons une matrice de vues adjacentes (3×3 ou 1×9). Nous appliquons la méthode de Wanner et Goldluecke (2012a) pour estimer les disparités entre les vues et l'incertitude de la géométrie à partir des huit vues voisines. La vue centrale est rendue par chaque algorithme testé puis comparée avec l'image originale qui sert de référence pour évaluer les performances de l'algorithme. Toutes les images sources sont prises en compte dans la synthèse de point de vue.

α et λ sont fixés à leur valeur d'origine dans les expériences précédentes (Wanner et Goldluecke, 2012b; Pujades *et al.*, 2014), respectivement 1.0 et 0.1. Nous faisons varier γ de 0 à 3 pour observer l'influence du terme sur les gradients. Un γ nul est bien entendu équivalent à minimiser la même fonctionnelle que Pujades *et al.* (2014), mais notre implémentation diffère légèrement de la leur, ce qui explique pourquoi nous avons représenté les deux dans le tableau des résultats 5.1. Nous comparons également notre méthode à celle de Wanner et Goldluecke (2012b). Toutes les expériences sont réalisées sur carte graphique (nVidia GTX Titan). La résolution du système prend entre 2 et 3 secondes pour des images de résolution 768×768.

Le PSNR (*Peak Signal to Noise Ratio*, plus il est haut mieux c'est) et le DSSIM $= 10^4(1 - \text{SSIM})$ (Wang *et al.*, 2004) (*Structural dissimilarity*, plus il est faible mieux c'est) sont calculés par rapport à la vue de référence pour évaluer nos résultats. Le deuxième jeu d'expériences a été réalisé avec les mêmes images, mais avec une géométrie plane – la disparité estimée est constante, ce qui correspond à un plan, le *proxy géométrique* le plus grossier possible. L'incertitude de la géométrie est fixée à l'intervalle de profondeur de la scène.

Notre terme sur les gradients améliore systématiquement les résultats avec une disparité estimée, et très souvent pour une disparité plane. La qualité des images générées est une fonction croissante de γ. Nous interprétons ceci par le fait que le terme sur les gradients ajoute de nouvelles contraintes au système, permettant à l'algorithme d'optimisation une meilleure convergence vers le minimum global de l'énergie.

5.6.2 Base de données non structurées

Les expériences suivantes (figure 5.6) ont été réalisées sur des vues réelles prises de la base de données de Strecha *et al.* (2008), *fountain* et *herzjesu*, ainsi que sur nos propres jeux de données *charce* et *lion*. Les résultats numériques 5.2 viennent

	HCI light fields, raytraced				HCI light fields, gantry				Standford light fields, gantry					
	buddha		stillLife		maria		couple		truck		gum nuts		tarot	
Estimated disparity														
SAVSRWanner et Goldluecke (2012b)	42.84	17	30.13	58	40.06	53	26.55	226	33.75	408	31.82	1439	28.71	60
BVSPujades *et al.* (2014)	42.37	18	30.45	55	40.10	53	28.50	178	33.78	**407**	31.93	1437	28.88	58
$\gamma = 0.0$	43.07	15	30.75	50	39.91	53	32.93	92	33.73	434	31.98	1430	25.37	51
$\gamma = 1.0$	43.28	14	30.83	49	40.14	51	33.05	90	33.82	430	32.08	1428	25.58	48
$\gamma = 2.0$	43.43	13	30.86	49	40.36	48	33.15	88	33.91	427	32.17	1426	25.74	46
$\gamma = 3.0$	**43.59**	**12**	**30.90**	**48**	**40.55**	**46**	**33.22**	**87**	**33.98**	423	**32.25**	**1424**	**25.89**	**44**
Planar disparity														
SAVSRWanner et Goldluecke (2012b)	34.28	74	21.28	430	31.65	144	20.07	725	32.48	419	30.55	1403	22.64	278
BVSPujades *et al.* (2014)	37.51	44	22.24	380	34.38	99	**22.88**	**457**	33.79	386	31.30	1378	23.78	218
$\gamma = 0.0$	37.69	42	**22.27**	**377**	34.28	100	22.74	468	34.50	367	31.38	1359	24.47	189
$\gamma = 1.0$	37.74	41	**22.27**	378	34.34	97	22.74	468	34.57	365	31.39	1359	24.51	187
$\gamma = 2.0$	37.80	**40**	22.26	378	34.40	95	22.74	468	34.66	362	**31.43**	1358	24.50	187
$\gamma = 3.0$	**37.84**	**40**	22.25	378	**34.45**	**93**	22.74	468	**34.68**	**360**	31.42	**1357**	**24.58**	**185**

Table 5.1 – *Résultats numériques sur les bases d'images synthétiques et réelles (Wanner, Sven* et al., *2013). Notre méthode est comparée à celle de Wanner et Goldluecke (2012b) et de Pujades* et al. *(2014). Le* proxy géométrique *est soit estimé par Wanner et Goldluecke (2012b) soit mis à profondeur constante avec une grande incertitude. Pour chaque* light-field*, la première valeur est le PSNR (plus il est aussi mieux c'est), la seconde est* 10^{-4} *fois le DSSIM.* $\text{DSSIM} = 10^4(1 - \text{SSIM})$ *(Wang* et al., *2004) (plus il est faible mieux c'est). La meilleure performance est en gras. Voir la section 6.2 pour de plus amples détails sur l'expérience.*

Fig. 5.4 – *Résultats sur les jeux de données* fountain *et* herzjesu. *La colonne de droite montre un échantillon des images sources.*

confirmer figures d'illustration en terme d'efficacité de l'ajout du terme portant sur les gradients des images.

Comme le système est faiblement contraint, des hautes fréquences apparaissent dans les zones visibles depuis peu de caméras. Ces artefacts sont accentués par une estimation très bruitée de la profondeur près des régions d'occultation (autour du poisson de la fontaine, ou du bas-relief de Jésus). Le paramètre λ contrôlant le terme de régularisation *Total Variation* a été augmenté à 0.003 pour réduire l'apparition de ces hautes fréquences. Mais le résultat est peu convainquant car les images perdent alors du détail comparées aux originales. Nous avons alors baissé λ pour conserver tous les traits du bas-relief et ajouté le terme sur les gradients ($\gamma = 1.0$). Nous pouvons voir sur les images, quelle que soit la géométrie utilisée pour le rendu, que les artefacts disparaissent tout en préservant les détails de l'image. Le terme sur les couleurs est conservé pour que la couleur originale des images ne soit pas affectée

Fig. 5.5 – *Résultats sur les jeux de données* charce *et* lion. *La colonne de droite montre un échantillon des images sources.*

mais mis à une valeur faible ($\alpha = 0.1$). L'ajout du terme sur les gradients a en outre permis d'empêcher l'apparition de faux contours près des frontières de visibilité, garantissant ainsi la propriété de *continuité*.

Les résultats numériques mettent en évidence que l'amélioration de la qualité des images synthétisées sauf dans le cas du jeu de données *charce*. En effet les meilleurs résultats sont obtenus avec un fort terme de lissage, alors que l'ajout du terme sur les gradients n'influe quasiment pas. Précisons que la reconstruction 3D de cette scène est très mauvaise du fait de la diversité des points de vue utilisés et de la présence du ciel, arrière-plan non texturé. Par conséquent de nombreuses zones ne sont pas reconstruites, et l'ensemble des pixels sur lesquels nous pouvons évaluer la qualité des résultats (l'ensemble des pixels reconstruits) est trop restreint pour que l'évaluation soit pertinente. En outre on peut penser que dans le cas où la reconstruction est extrêmement bruitée, une forte stabilisation par le terme de

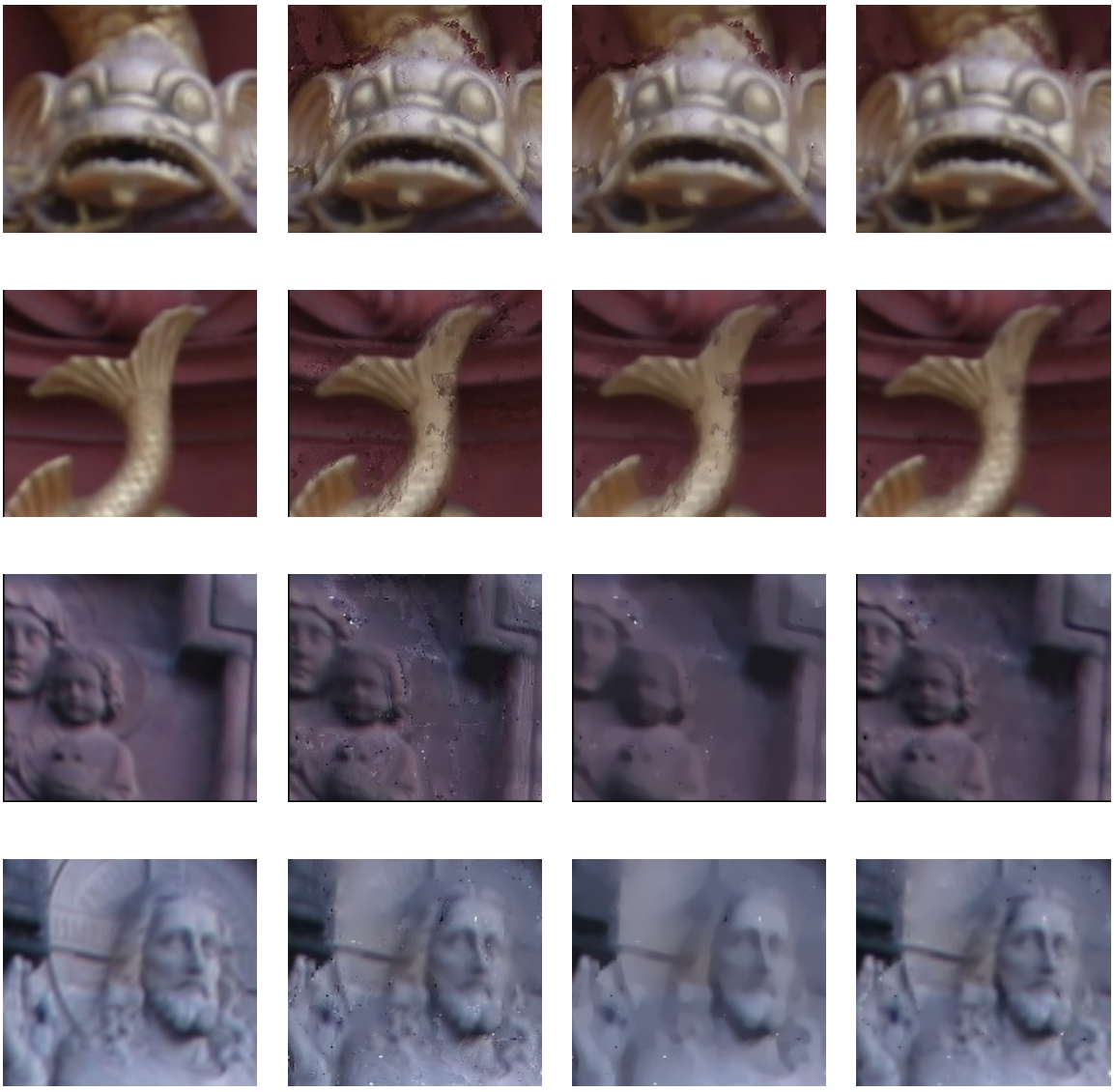


image de référence	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.002$	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.003$	$\alpha = 0.1, \gamma = 1.0,$ $\lambda = 0.002$

Fig. 5.6 – *Rendu avec différents jeux de paramètres* $(\alpha, \gamma, \lambda)$ *qui contrôlent la proportion des différents termes dans la formule de l'énergie (5.28). Les deux premières lignes montrent des résultats sur la collection d'images* fountain*, et les deux dernières sur la collection* herzjesu*. L'approche proposée est* $\gamma \neq 0$*.*

	$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.002$ État de l'art		$\alpha = 1.0, \gamma = 0.0,$ $\lambda = 0.003$ État de l'art		$\alpha = 0.1, \gamma = 1.0,$ $\lambda = 0.002$ (notre méthode)	
fountain – view 2	21.03	132	21.09	120	**21.16**	**107**
fountain – view 5	26.00	74	26.14	64	**26.36**	**51**
fountain – view 8	22.00	140	22.08	125	**22.16**	**111**
herzjesu – view 2	21.73	186	**21.96**	153	21.93	**143**
herzjesu – view 4	23.13	194	23.81	130	**23.90**	**115**
herzjesu – view 6	18.08	349	18.26	287	**18.31**	**273**
charce – view 4	13.74	905	**13.85**	**885**	13.73	901
charce – view 6	9.953	1360	**10.04**	**1317**	10.00	1352
charce – view 11	14.20	859	**14.38**	**793**	14.20	843
lion – view 1	24.40	190	24.36	198	**24.46**	**175**
lion – view 3	29.16	103	29.15	107	**29.57**	**87**
lion – view 5	22.18	298	22.25	294	**22.27**	**277**

Table 5.2 – *Résultats numériques sur des bases de données non structurées. Notre méthode est comparée aux méthodes de l'état de l'art (Wanner et Goldluecke, 2012b; Pujades et al., 2014), pour lesquelles il n'y a pas de contraintes sur les gradients de l'image ($\gamma = 0.0$). Pour chaque résultat, la première valeur est le PSNR (plus il est élevé mieux c'est), la seconde étant le DSSIM (plus il est faible mieux c'est).* $\text{DSSIM} = 10^4(1 - \text{SSIM})$. *La meilleure valeur est notée en gras.*

lissage est plus bénéfique que l'ajout du terme d'attache aux gradients.

5.7 Conclusion

Nous avons présenté une méthode de rendu basé image qui permet de générer une nouvelle vue à partir d'un ensemble générique et non structuré d'images. Cette méthode est inspirée par les travaux de Pujades *et al.* (2014), qui ont œuvré pour formaliser la plupart des « propriétés désirables » listées dans l'article phare de Buehler *et al.* (2001). Leur approche fut d'introduire une formulation bayésienne du problème de rendu et d'obtenir la vue cherchée par un processus d'optimisation. La seule propriété qu'ils n'ont pu formaliser fut la propriété de *continuité*, qui énonce que les contributions de chaque vue source doivent être des fonctions continues des coordonnées des pixels.

Nous avons montré qu'un moyen de garantir cette *continuité* est de déclarer que les contours, textures et détails ne devraient pas être créés dans l'image cible s'ils ne sont pas présents dans les images sources aux endroits visibles. Cela implique l'ajout d'un terme additionnel portant sur les données sources, basé sur les gradients des images. L'énergie ainsi modifiée peut être minimisée en résolvant de façon itérative un système linéaire dérivé de la fonctionnelle. Ce système est alors plus contraint et mieux conditionné que le précédent, ce qui empêche l'apparition d'artefacts près des frontières de visibilité.

Ce résultat montre une nette amélioration par rapport aux précédentes méthodes

de rendu basées sur les intensités, à la fois en termes de mesures qualitatives et en terme de qualité subjective.

Cette méthode pourrait être retravaillée pour optimiser directement les gradients de la vue cible, plutôt que les intensités ; puis l'intensité de l'image pourrait être reconstruite en résolvant l'équation de Poisson, comme il est fait par Kopf *et al.* (2013). Cela devrait complètement enlever toutes les variations dans l'image synthétisée qui viennent de discontinuités des fonctions de visibilité, qui sont toujours visibles dans nos résultats, bien qu'atténuées (figure 5.7).

Fig. 5.7 – *En haut : image résultant de l'aglorithme de Pujades et al. (2014) sur une partie de la fontaine. En bas : notre terme sur les gradients enlève la plupart des artefacts, mais des pixels colorés subsistent, un halo se formant autour d'eux.*

6 Linéarisation de l'espace plénoptique

6.1 Introduction

La reconstruction du champ lumineux et le rendu basé image en général utilisent le plus souvent la géométrie épipolaire pour estimer des cartes de disparité denses (Wanner et Goldluecke, 2014). Cette information de profondeur est alors combinée avec les images sources pour créer une nouvelle vue (Pujades *et al.*, 2014; Nieto *et al.*, 2016a). Cependant, la contrainte épipolaire et le fait qu'un rayon lumineux peut correspondre à une profondeur donnée sont basés sur l'hypothèse forte faite sur la scène elle-même, selon laquelle elle serait formée de surfaces opaques avec une réflectance presque lambertienne. Les scènes réelles peuvent contenir des réflexions spéculaires, des milieux semi-transparents, de la réfraction, ou même des milieux dont les indices de réfraction ne sont pas constants (comme dans les mirages). Si l'échantillonnage original du champ lumineux est assez dense, ce qui est le cas des ensembles de caméras ou des caméras plénoptiques, l'hypothèse lambertienne peut être suffisante pour décrire localement le champ lumineux, et pour l'approximer dans un petit voisinage autour des vues originales. Cependant, lorsque le champ lumineux devient épars, ou si la nouvelle vue contient des rayons qui se trouvent à l'extérieur de la région 4D échantillonnée, toute déviation de l'hypothèse lambertienne est susceptible d'être amplifiée et de causer des artefacts visuels ou une perte de réalisme dans la nouvelle vue.

On observe que les déviations de l'hypothèse lambertienne sont principalement de deux sortes. Certains points appartiennent à une surface, qui peut être texturée, mais n'ont pas une réflectance diffuse ou isotrope. Ils ont une profondeur, et les rayons optiques générés par ces points suivent les règles de la projection perspective et la géométrie épipolaire ; seulement leur radiance dévie de l'hypothèse lambertienne. Les autres points non-lambertiens dans les images correspondent à des chemins optiques complexes : les rayons optiques sont affectés par des séries de réflexions (ou de réflexions spéculaires), de réfractions (lorsque le rayon passe d'un milieu à un autre ayant un indice de réfraction différent, comme de l'air à l'eau ou au verre par exemple), ou des variations continues de l'indice de réfraction. Bien que dans

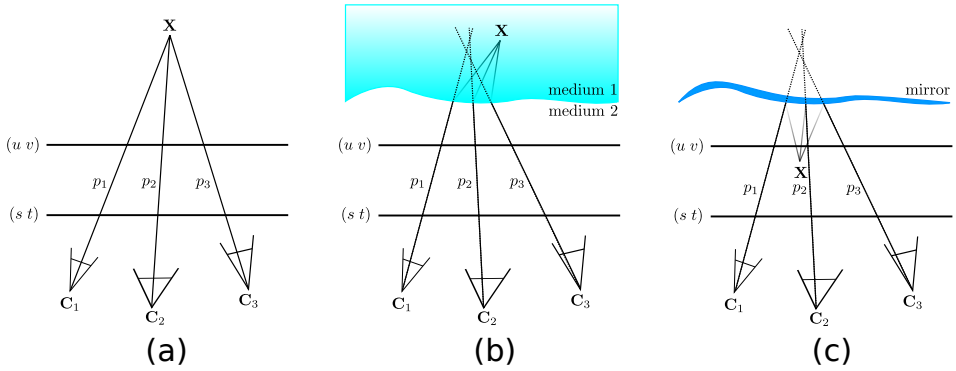


Fig. 6.1 – *Distorsion géométrique du champ de lumière. Un* point visuel **X***, correspondant à un point dans la scène, vu par trois caméras comme les rayons p_1, p_2 et p_3. (a) Pas de distorsion : tous les rayons appartiennent à un seul faisceau et s'intersectent en un même point 3D. (b) Réfraction par changement de milieu (ayant des indices de réfraction différents) qui dévie les rayons lumineux. Les rayons ne s'intersectent pas nécessairement et la triangulation échoue à localiser un unique point dans l'espace. (c) Les surfaces miroitantes dévient aussi les rayons lumineux, qui ne s'intersectent pas forcément en un unique point 3D.*

certains cas spécifiques des surfaces réfractives ou réflectives ou même des points 3D peuvent être reconstruits (Adato *et al.*, 2010), le problème général a trop d'inconnues, puisque chaque rayon optique peut rencontrer de nombreuses transitions de matériaux. La caractéristique commune de cette deuxième sorte de point est qu'ils ne suivent pas les règles communes de projection perspective ou parallaxe : quand l'œil de la caméra bouge à gauche, un point image peut bouger à droite (ce qui est le comportement attendu), mais aussi à gauche, en haut en en bas, violant ainsi la contrainte épipolaire. En se basant sur ces observations, nous proposons de nous concentrer sur la reconstruction du champ lumineux 4D lui-même, plutôt que de tenter d'expliquer le champ lumineux en reconstruisant les surface 3D et les matériaux qui composent la scène.

Quand un point 3D se trouvant sur une surface lambertienne est observé dans les images, en supposant qu'il n'y a pas d'occultations, les rayons optiques qui passent par ce point 3D ont la même radiance. L'image de ce point dans une caméra est donnée soit en projetant ce point 3D dans la caméra, soit en prenant dans le faisceau le rayon qui passe par le centre optique de la caméra. Supposons maintenant que la scène est statique, et que nous observons un point dans une caméra qui a un comportement plus complexe dû à des réflexions ou des réfractions : son mouvement apparent dans l'image quand le point de vue change légèrement peut ne pas être cohérent avec la parallaxe ou la géométrie épipolaire. Un tel point est appelé ici *point visuel*, et consiste en un ensemble bidimensionnel de rayons. Cet ensemble, nommé *congruence de droites* (Ponce *et al.*, 2016), généralise le concept de faisceau de rayons : pour un centre optique donné, les images de ce *point visuel* correspondent aux éléments de cette congruence qui passent par le centre optique. Il s'agit généralement d'une seule image, mais il peut y en avoir plusieurs s'il y a plusieurs chemins optiques possibles entre la source et le centre optique.

Dans ce travail, nous proposons d'extraire plusieurs rayons associés au même *point*

visuel en mettant plusieurs images en correspondance (via un flux optique comme dans les travaux de Park *et al.* (1998)), puis en ajustant des congruences de niveaux de complexité variés à ces *points visuels*. Dans notre modèle, la radiance est fonction des paramètres du rayon, modelant ainsi des variations à la fois de position et photométriques. Ce modèle est finalement utilisé pour retrouver les rayons manquants dans le champ lumineux, pour par exemple calculer tous les rayons passant par le centre optique d'une caméra donnée et ainsi faire le rendu d'une nouvelle vue. Pour valider notre approche, nous effectuons des expériences sur des jeux de données de champ de lumière épars dans lesquelles nous faisons le rendu de nouvelles vues et comparons avec des images de références. En outre nous offrons la possibilité de déterminer quel modèle plénoptique de congruence est le plus adapté grâce à une technique de sélection de modèle.

Nos contributions principales sont une technique inédite de d'échantillonnage et de paramétrisation de l'espace plénoptique, un procédé d'optimisation pour ajuster des modèles à l'échantillonnage de l'espace plénoptique, permettant une reconstruction plus précise des spécularités, transparences et réfractions, et une nouvelle méthode de rendu continue qui satisfait la plupart des propriétés qu'un algorithme d'IBR devrait avoir (Buehler *et al.*, 2001).

6.2 Travaux antérieurs

Reconstruire le champ lumineux Le champ lumineux fut introduit en tant que représentation intermédiaire du signal 4D de radiance. Il peut être traité pour produire des effets tels que la synthèse de nouvelles vues, la refocalisation, le *matting*, etc. (Levoy et Hanrahan, 1996; Gortler *et al.*, 1996). Lorsque l'on échantillonne le champ lumineux 4D, il y a un compromis entre la résolution spatiale et angulaire (Chai *et al.*, 2000) qui peut être compensée par interpolation (Georgeiv *et al.*, 2006). Une façon d'interpoler les échantillons épars est d'utiliser un *proxy géométrique*, c'est-à-dire une reconstruction 3D plus ou moins précise de la scène. Mais calculer une reconstruction précise de la scène peut se révéler coûteux (Heber et Pock, 2014, 2016; Wanner et Goldluecke, 2014). En outre cela repose généralement sur l'hypothèse que la scène est lambertienne : la radiance d'un point d'une surface 3D ne dépend pas du point de vue. Dorénavant nous ne cherchons plus à reconstruire la géométrie de la scène de façon explicite. Nous utilisons seulement des rayons mis en correspondance deux à deux par un flux optique, qui en réalité ne correspondent pas forcément à la projection d'un point 3D. Notre but est de traiter des scènes génériques en utilisant un modèle plus général de la fonction plénoptique (Adelson et Bergen, 1991).

Reconstruire des scènes spéculaires ou réflectives Interpoler le *flowed light field* comme un moyen de faire le rendu de n'importe quel point de vue de la scène capturée a été expérimenté par Einarsson *et al.* (2006). Cependant leur utilisation du flux optique est limitée à une interpolation bilinéaire pour synthétiser une nouvelle vue. Dans une approche plus infographique, Zhou *et al.* (2013, 2014) modélisent

les réflexions non lambertiennes par le modèle de BRDF de Phong. En jouant sur les exposants, ils modélisent différents types de surfaces (lambertienne, spéculaire ou plus terne), réduisant ainsi le taux d'échantillonnage du champ lumineux requis pour la synthèse de nouvelles vues. Sulc *et al.* (2016) séparent la composante diffuse de la composante spéculaire estimée à partir du flux spéculaire. Cela requiert néanmoins le calcul préliminaire d'une carte de disparité basé sur le tenseur de structure du premier ordre (Wanner et Goldluecke, 2014). Comme les autres méthodes précédemment citées, cette dernière ne traite pas le cas des surfaces réfractives.

Reconstruire des scènes réfractives ou transparentes Dans le contexte du champ lumineux, plusieurs articles proposent déjà des solutions pour traiter les surfaces transparents ou réfractives, et parfois pour les reconstruire de façon explicite. Wetzstein *et al.* (2011) suggèrent l'utilisation d'une caméra unique possédant un ensemble de micro-lentilles ainsi qu'une boîte à lumière utilisée en stéréo photométrique : diverses sources de lumière colorée codent le domaine spatial et le domaine angulaire. Iffa *et al.* (2012) proposent de séparer le flux optique selon s'il est dû à la parallaxe ou à la réfraction (qui crée une déviation lumineuse) en une seule prise plénoptique. Ils résolvent un problème classique de flux optique pour toutes les paires d'images simultanément, auquel s'ajoute un terme de régularisation divergence-rotationnel bien connu dans la littérature de mécanique des fluides. L'inconvénient est que cela requiert un arrière-plan très texturé (ce qui n'est pas le cas de notre méthode). Maeno *et al.* (2013) introduisent des caractéristiques de distorsion du champ lumineux pour décrire et reconnaître un object composé d'une surface réfractive et d'un arrière-plan texturé, en utilisant une caméra plénoptique commerciale. Alterman *et al.* (2013) utilisent un flux optique pour larges entraxes entre deux vues pour traiter les réfractions seulement. Comme nous, ils proposent une approche de triangulation multi-vues, bien qu'ils ne modélisent que des points lambertiens vus au travers de milieux réfractifs.

Résumé Les méthodes existantes de reconstruction de champ lumineux ont trois principaux inconvénients. Premièrement la plupart de ces méthodes sont limitées par la configuration des caméras et de la scène (caméras plénoptiques, boîtes de lumière, arrière-plan très texturé). Outre les caméras plénoptiques, les autres dispositifs de capture incluent les ensembles de caméras (Wilburn *et al.*, 2005), qui ont permis la constitution de la *Stanford Light Field Archive*. Bien que nous utilisons ces jeux de données dans nos expériences, notre méthode peut tout aussi bien être appliquée à des données capturées par n'importe quelle caméra plénoptique ou n'importe quel ensemble de caméras arrangées selon des configurations non structurées. Deuxièmement le but des méthodes présentées est souvent de représenter la géométrie de la scène de manière explicite, ce qui certes permet une meilleure interpolation du champ lumineux, mais ne devrait pas être nécessaire. En fait une erreur dans la reconstruction de la scène peut avoir un impact dramatique sur la qualité du champ lumineux reconstruit. Notre méthode travaille directement sur l'interpolation et l'extrapolation du champ lumineux, sans reconstruction explicite préliminaire de la scène. Troisièmement elles ne s'attaquent qu'à un seul problème à

la fois : soit elles essayent de séparer la composante diffuse de la composante spéculaire, soit elles traitent des surfaces réfractives. Les deux problèmes sont toujours traités séparément. Au contraire notre méthode ne cherche pas à modéliser un comportement particulier de la lumière, mais plutôt à approximer localement l'espace plénoptique.

6.3 Aperçu de la méthode

Notre méthode se décompose en plusieurs étapes clé. Nous calculons d'abord le flux optique entre des paires de vues adjacentes pour créer un ensemble d'échantillons de couleurs et d'orientations attachés à un seul *point visuel*, que nous appelons le *flot de lumière*. Chaque échantillon est un rayon définit par 4 paramètres qui nous renseignent sur son orientation dans l'espace (section 6.4) et 3 paramètres qui décrivent sa radiance. Lorsque le dispositif de capture est un ensemble de caméras, chaque rayon passe par le centre optique de la caméra qui voit le *point visuel*. La radiance est la valeur de couleur du pixel.

Ensuite nous ajustons un modèle de congruence de droites à chaque ensemble d'échantillons (section 6.5), qui a pour but d'expliquer le mouvement du *point visuel* dans l'espace plénoptique (paramétrisation 4D du champ lumineux et radiance). Le modèle à 3 paramètres est synonyme de faisceau de droites, lorsque tous les rayons passent par le même point 3D ; on conçoit aussi des modèles linéaires à 4 ou 6 paramètres, dans le but de prédire des phénomène comme la réfraction, la réflexion, ou les variations continues d'indice de réfraction. Un critère de sélection de modèle permet en outre de départager les modèles en trouvant un compromis entre la complexité (en terme de nombre de paramètres) et l'erreur résiduelle, dans le but d'éviter le sur-ajustement.

Enfin nous synthétisons une nouvelle vue en interpolant la projection du *point visuel* dans le plan capteur de la nouvelle caméra (section 6.6). Pour chaque *point visuel*, on trouve sa position dans le point de vue cible en intersectant le modèle de congruence de droites avec le centre optique de la caméra cible. Nous procédons au rendu de la scène selon la technique de rendu par éclaboussures (*splatting*) vue au chapitre 4.

6.4 Échantillonnage et paramétrisation de l'espace plénoptique

L'espace plénoptique est l'espace des rayons lumineux qui passent au travers d'une scène, par n'importe quel point et dans n'importe quelle direction. Nous supposons que dans la région de l'espace où nous voulons reconstruire le champ lumineux, la radiance est constante le long d'un rayon. Comme dans les travaux de Levoy et Hanrahan (1996), nous utilisons la paramétrisation 4D du champ lumineux appelée *light slab*, où les coordonnées (u, v, s, t) sont obtenues par intersection du rayon 3D avec deux plans de référence parallèles : (u, v) et (s, t) sont les coordonnées de

ces intersections avec chaque plan. Notons **I** la radiance de ce rayon, représentée par 3 coordonnées supplémentaires, ses composantes RGB (rouge, vert, bleu). En définitive chaque rayon est décrit par 7 coordonnées.

Considérons un point 3D dans l'espace. L'ensemble des rayons qui émanent de ce point est un espace à deux dimensions, appelé une congruence de droites. Puisque toutes les droites passent par le même point, cette congruence est réduite à un faisceau de rayons bidimensionnel.

Dans la plupart des cas, le chemin optique entre ce point 3D et la région où nous souhaitons reconstruire le champ lumineux traverse un milieu transparent et homogène ; son indice de réfraction est constant. Alors le faisceau de droites n'est pas modifié par le milieu (figure 6.1a). Cependant, s'il y a des réfractions, des réflexions ou des variations d'indice de réfraction, le faisceau de droites est déformé en une congruence de droites plus générique (figures 6.1b, 6.1c). De telles congruences de droites, notées P, décrivent l'espace bidimensionnel de rayons dans l'espace plénoptique que nous appelons un *point visuel*. Ces coordonnées 7D de rayons qui appartiennent à une même congruence de droites sont fortement corrélées. Par exemple les rayons associés à un point 3D lambertien qui passent au travers d'un milieu homogène (figure 6.1a) sont de radiance constante, et leur coordonnées (u, v, s, t) décrivent un plan en 4D paramétrisé par les coordonnées cartésiennes de ce point 3D. La congruence de droites est représentée en intégralité par un espace 3D et sa radiance (c'est-à-dire 3 + 3 paramètres).

Des modèles de congruences de droites plus complexes avec plus de paramètres peuvent décrire des systèmes optiques plus compliqués (Pottmann et Wallner, 2010). Nous nous restreignons ici aux modèles linéaires présentés dans la section 6.5, qui sont utilisés pour décrire avec fidélité la géométrie locale de ces congruences de droites.

Échantillonnage Une image caméra contient des rayons échantillonnés qui passent par le plan image et le centre optique de la caméra. Généralement, dans la congruence de droites qui correspond au *point visuel*, un seul rayon optique rencontre le centre optique de la caméra, de telle manière que plusieurs caméras sont requises pour obtenir plusieurs échantillons du *point visuel*. Considérons maintenant un rayon dans une image de référence. Les rayons associés dans les autres images peuvent s'obtenir à partir d'un flux optique. Remarquons que n'importe quelle autre méthode de mise en correspondance qui n'est pas contrainte par la géométrie épipolaire peut être utilisée à la place du flux optique. Le flux optique est seulement calculé entre les caméras voisines (voir figure 6.7), de sorte que l'on puisse supposer que les images ne diffèrent pas trop d'une vue à l'autre. Le flux optique est particulièrement adapté à l'approximation locale de l'espace plénoptique car sa mise en correspondance des images est d'autant plus précise que les vues sont proches les unes des autres (faible entraxe entre les caméras). En outre le flux optique doit satisfaire la contrainte d'intensité constante (*brightness constancy constraint*), ce qui est systématiquement validé lorsque plusieurs photographies sont prises en même temps, comme avec une grille de caméras ou une caméra plénoptique, ce qui en fait une méthode de mise en correspondance très adaptée à l'imagerie plénoptique. Un

point visuel est alors représenté par la liste des ses positions et radiances dans les images où il est visible. En considérant que le flux optique est dense, on a autant de vecteurs d'échantillons que de pixels dans chacune des images sources. Chaque point image est converti selon la paramétrisation 4D *light slab* détaillée dans la suite.

Paramétrisation des rayons Étant donnée une caméra source décrite par son centre optique $\mathbf{C}$, sa matrice de rotation $\mathbf{R}$ et sa matrice de paramètres intrinsèques $\mathbf{K}$, la représentation 4D *light slab* d'un point image $\mathbf{x}$ est obtenue en intersectant le rayon 3D passant par $\mathbf{x}$ et $\mathbf{C} = (C_x, C_y, C_z)$ avec les deux plans parallèles. Sans perte de généralité nous pouvons supposer que les deux plans ont pour équations $z = 0$ et $z = 1$. Soit $\mathbf{s} = (s, t)$ l'intersection avec le plan d'équation $z = 0$ et $\mathbf{u} = (u, v)$ l'intersection avec le plan d'équation $z = 1$. Le vecteur directeur du rayon $\mathbf{r} = (r_x, r_y, r_z)$ s'obtient selon la formule

$$\mathbf{r} = \mathbf{R}^\intercal \mathbf{K}^{-1} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}, \tag{6.1}$$

et les coordonnées *light slab* sont

$$s = C_x - C_z \frac{r_x}{r_z}, \qquad t = C_y - C_z \frac{r_y}{r_z}, \tag{6.2}$$

$$u = C_x + (1 - C_z).\frac{r_x}{r_z}, \qquad v = C_y + (1 - C_z).\frac{r_y}{r_z}. \tag{6.3}$$

Incertitude de mesure Dans un problème de triangulation classique, le modèle de point 3D est ajusté aux données (points image) en minimisant l'erreur de reprojection. Les erreurs sont habituellement considérées de variances identiques, gaussiennes et isotropes dans chaque plan image. Dans notre cas nous cherchons à ajuster un modèle congruence de droites à un ensemble de rayons paramétrés par (s, t, u, v), étant donnée l'erreur de mise en correspondance, qui est aussi mesurée dans les images. Par conséquent nous avons besoin d'exprimer les covariances des intersections avec les deux plans (s, t) et (u, v). Cette covariance est dérivée en propageant l'incertitude des points image sur les plans. La covariance de (u, v, s, t) est noté $\mathbf{\Sigma_{u,s}}$. Les matrices de covariance des erreurs marginales sur $\mathbf{s}$ et $\mathbf{u}$ sont notées $\mathbf{\Sigma_{ss}}$ et $\mathbf{\Sigma_{uu}}$ respectivement. Les jacobiennes de la paramétrisation sont

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = (1 - C_z).\mathbf{J_r}, \qquad \frac{\partial \mathbf{s}}{\partial \mathbf{x}} = -C_z \mathbf{J_r}, \tag{6.4}$$

avec

$$\mathbf{J_r} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{1}{\mathbf{r}_z}(\mathbf{I}_2 | -\mathbf{r})\mathbf{R}^\intercal \mathbf{K}^{-1}(\mathbf{I}_2 | \mathbf{0}_2)^\intercal. \tag{6.5}$$

On note l'incertitude sur le vecteur directeur du rayon

$$\mathbf{S} = \mathbf{J_r} \mathbf{\Sigma_{xx}} \mathbf{J_r^\intercal}. \tag{6.6}$$

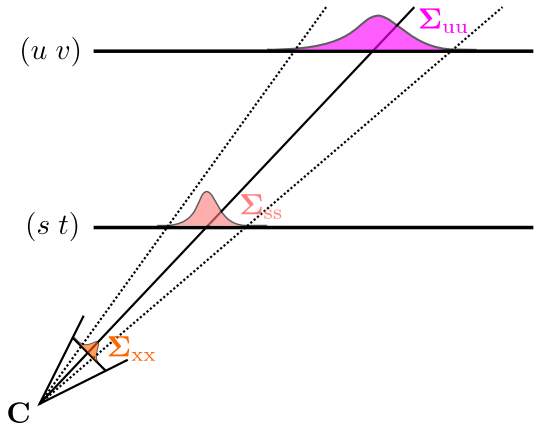


Fig. 6.2 – *Propagation de l'incertitude géométrique du plan image de la caméra sur les deux plans de paramétrisation.* $\mathbf{\Sigma_{xx}}$ *représente l'incertitude de mise en correspondance originale dans le plan image.* $\mathbf{\Sigma_{ss}}$ *et* $\mathbf{\Sigma_{uu}}$ *sont les variances des distributions marginales de* $\mathbf{s}$ *et* $\mathbf{u}$ *respectivement.*

La matrice de covariance $\mathbf{\Sigma_{xx}}$ représente l'incertitude sur la mise en correspondance du flux optique. Bien que nous fixions l'incertitude de mise en correspondance dans nos expériences, la plupart des méthodes de flux optique produisent aussi une image de mesure de qualité du flux qui pourrait être utilisée pour moduler cette incertitude (qui est par exemple plus grande dans les zones non-texturées). Observons que l'incertitude de mise en correspondance augmente lorsque deux rayons sont liés par un flux chaîné. Nous appelons « flux chaîné » une mise en correspondance indirecte de deux points image par une succession de plusieurs flux optiques : le premier flux optique est appliqué au premier point image, se qui nous donne une première destination dans la deuxième vue, à laquelle on applique le deuxième flux optique et ainsi de suite. En supposant que l'erreur de mise en correspondance est normalement distribuée, les covariances s'additionnent, et la covariance du flux chaîné est proportionnelle au nombre de flux : $\mathbf{\Sigma_{xx}}$ pour les associations directes de rayons voisins, $2\mathbf{\Sigma_{xx}}$ pour les chaînes de deux flux optiques, etc. Dans notre configuration expérimentale de caméras, puisque nous calculons le flux optique à partir des vues adjacentes seulement, les vues distantes de la vue centrale (vue de référence) ont une plus grande incertitude de mise en correspondance, et contribuent moins au modèle que les vues plus proches (Figure 6.7).

Finalement, nous obtenons les matrices de covariance des erreurs marginales

$$\mathbf{\Sigma_{uu}} = (C_z - 1)^2\mathbf{S}, \tag{6.7}$$

$$\mathbf{\Sigma_{us}} = \mathbf{\Sigma_{su}} = (C_z - 1)C_z\mathbf{S}, \tag{6.8}$$

$$\mathbf{\Sigma_{ss}} = C_z^2\mathbf{S} \tag{6.9}$$

et la matrice de covariance de la distribution jointe

$$\mathbf{\Sigma_{u,s}} = \begin{pmatrix} \mathbf{\Sigma_{uu}} & \mathbf{\Sigma_{us}} \\ \mathbf{\Sigma_{su}} & \mathbf{\Sigma_{ss}} \end{pmatrix}. \tag{6.10}$$

De la même façon l'incertitude des mesures de radiance se déduit de l'incertitude de mise en correspondance du flux optique. La matrice de covariance de l'erreur de radiance a pour expression

$$\mathbf{\Sigma_{I,s}} = \begin{pmatrix} \mathbf{\Sigma_{II}} & \mathbf{\Sigma_{Is}} \\ \mathbf{\Sigma_{sI}} & \mathbf{\Sigma_{ss}} \end{pmatrix}, \tag{6.11}$$

où les matrices de covariance des erreurs marginales sont

$$\mathbf{\Sigma_{II}} = \nabla\mathbf{I}\mathbf{\Sigma_{xx}}\nabla\mathbf{I}^\intercal \tag{6.12}$$
$$\mathbf{\Sigma_{Is}} = -C_z\nabla\mathbf{I}\mathbf{\Sigma_{xx}}\mathbf{J_r}^\intercal \tag{6.13}$$
$$\mathbf{\Sigma_{sI}} = -C_z\mathbf{J_r}\mathbf{\Sigma_{xx}}\nabla\mathbf{I}^\intercal \tag{6.14}$$
$$\mathbf{\Sigma_{ss}} = C_z^2\mathbf{J_r}\mathbf{\Sigma_{xx}}\mathbf{J_r}^\intercal. \tag{6.15}$$

Décomposition en valeurs propres Les matrices de covariance sont utilisées dans la suite afin de pondérer les différents échantillons dans le processus d'optimisation. Le modèle à ajuster aux données aura tendance à être plus proche des points mesurés qui ont une faible incertitude. La notion de proximité est définie ici par une distance de Mahalanobis qu'il est impossible de définir pour n'importe quel vecteur car la matrice de covariance est singulière. Nous détaillons ici la décomposition en valeurs propres des matrices de covariance dans le but de justifier l'utilisation de la distance de Mahalanobis et de donner une expression des valeurs propres qui servent au calcul d'une telle distance.

$\mathbf{S}$ est une matrice symétrique réelle, elle est donc diagonalisable d'après le théorème spectral en dimension 2. Notons λ_1 et λ_2 ses deux valeurs propres. De la même façon, $\mathbf{\Sigma}_{\mathrm{u,s}}$ a 4 valeurs propres, que nous notons μ_1, μ_2, μ_3 et μ_4. Toute valeur propre μ de $\mathbf{\Sigma}_{\mathrm{u,s}}$ est racine du polynôme caractéristique :

$$|\mathbf{\Sigma}_{\mathrm{u,s}} - \mu\mathbf{I_4}| = |\mu|^2|\mu\mathbf{I_2} - (C_z^2 + (C_z - 1)^2)\mathbf{S}| = 0$$

Dans un premier temps on remarque que 0 est solution double du polynôme caractéristique. Nous noterons sans perte de généralité que $\mu_3 = \mu_4 = 0$. Nous en déduisons que le noyau de $\mathbf{\Sigma}_{\mathrm{u,s}}$ est de dimension 2, et que $\mathbf{\Sigma}_{\mathrm{u,s}}$ est de rang 2. En effet le sous-espace vectoriel bidimensionnel engendré par ses vecteurs colonnes, de dimension 2 décrit l'ensemble des rayons qui passent par le centre optique de la caméra source. Cela n'explique notamment par le fait que l'ensemble des rayons capturés par une vue donnée est un faisceau ayant deux degrés de liberté. Remarquons par ailleurs que par construction du *light slab*, les coordonnées ne sont pas indépendantes, mais liées par deux relations paramétrées par le centre optique de la caméra :

$$u = \frac{C_z - 1}{C_z}s + \frac{C_x}{C_z}, \qquad v = \frac{C_z - 1}{C_z}s + \frac{C_y}{C_z}. \tag{6.16}$$

Il est important de noter par la suite que l'erreur de mesure est contrainte à ap-

partenir à ce faisceau de rayons passant par le centre optique $\mathbf{C}$ de la caméra (sous-espace vectoriel dimension 2).

Dans un deuxième temps on peut noter que si μ est différent de 0, alors μ est valeur propre de $(C_z^2 + (C_z - 1)^2)\mathbf{S}$ et $(C_z^2 + (C_z - 1)^2)\mathbf{S}$. Par conséquent les deux autres solutions du polynôme caractéristique sont $\mu_1 = (C_z^2 + (C_z - 1)^2)\lambda_1$ et $(C_z^2 + (C_z - 1)^2)\lambda_2$. On montre aussi par le calcul que si $\mathbf{e_1}$ est un vecteur propre associé à la valeur propre μ_1, alors un vecteur propre associé à la valeur propre μ_2 est $\mathbf{e_2} = (-e_{1,v}\ e_{1,u}\ -e_{1,s}\ e_{1,t})^\intercal$.

Pour résumer, la décomposition spectrale des deux matrices symétriques est la suivante :

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^\intercal \qquad \mathbf{\Sigma}_{\text{u,s}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\intercal \tag{6.17}$$

$$\mathbf{D} = diag(\lambda_1, \lambda_2) \qquad \mathbf{\Lambda} = diag(\mu_1, \mu_2, \mu_3, \mu_4). \tag{6.18}$$

Sachant que $\mathbf{\Sigma}_{\text{u,s}}$ s'écrit en fonction de $\mathbf{S}$:

$$\mathbf{\Sigma}_{\text{u,s}} = \begin{pmatrix} \mathbf{S} & \mathbf{0_{22}} \\ \mathbf{0_{22}} & \mathbf{S} \end{pmatrix} \begin{pmatrix} (C_z - 1)^2\mathbf{I_{22}} & C_z(C_z - 1)\mathbf{I_{22}} \\ C_z(C_z - 1)\mathbf{I_{22}} & C_z^2\mathbf{I_{22}} \end{pmatrix}$$

On a l'expression des vecteurs propres et valeurs propres de $\mathbf{\Sigma}_{\text{u,s}}$ en fonctions de ceux de $\mathbf{S}$:

$$\mathbf{\Lambda} = \begin{pmatrix} ((C_z - 1)^2 + C_z^2)\mathbf{D} & \mathbf{0_{22}} \\ \mathbf{0_{22}} & \mathbf{0_{22}} \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{U} & \mathbf{U} \\ \frac{C_z}{C_z - 1}\mathbf{U} & \frac{1 - C_z}{C_z}\mathbf{U} \end{pmatrix}.$$

6.5 Modélisation de l'espace plénoptique

Ayant obtenu les données 4D donnant les orientations des rayons et les données 3D de radiance (couleurs RGB) grâce à l'échantillonnage et à la paramétrisation précédemment détaillés, nous sommes maintenant en mesure d'ajuster un *modèle géométrique* et un *modèle photométrique* pour chaque ensemble d'échantillons (chaque *point visuel*). Dans un premier temps nous allons expliquer le modèle simple de la congruence de droites associée au point 3D $\mathbf{X}$, qui est un faisceau de rayons. Puis nous étendrons la méthode à des modèles géométriques plus complexes à plus de 3 paramètres. De même pour les modèles photométriques, nous passerons du modèle lambertien à 3 paramètres au modèle plus complexe à 9 paramètres.

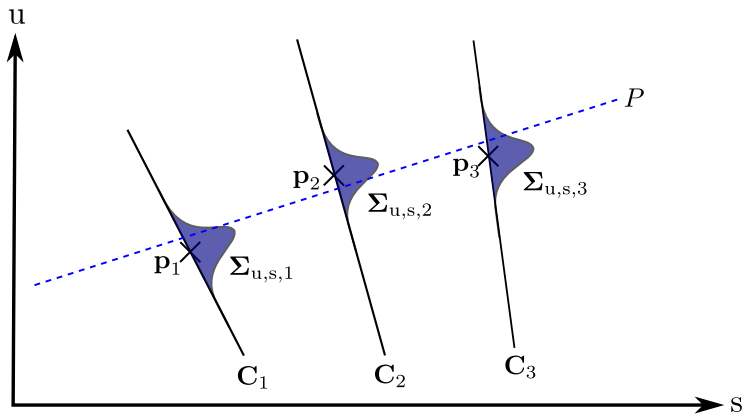


Fig. 6.3 – *Un* modèle géométrique *linéaire du faisceau de rayons P est ajusté aux données. Les échantillons de rayons des trois vues sont notés $\mathbf{p}_1$, $\mathbf{p}_2$ et $\mathbf{p}_3$. Les matrices de covariance des distributions géométriques jointes associées à ces échantillons sont notées $\mathbf{\Sigma}_{\mathbf{u,s},1}$, $\mathbf{\Sigma}_{\mathrm{u,s},2}$ et $\mathbf{\Sigma}_{\mathbf{u,s},3}$. Elles pondèrent les contributions de chaque vue dans le processus d'optimisation.*

modèle géométrique Soit P le faisceau de droites qui passent par le point $\mathbf{X} = (x, y, z)$. Pour chaque droite $\mathbf{q} = (u, v, s, t)$ passant par $\mathbf{X}$ on a

$$\mathbf{q} \in P \iff \begin{cases} u = \alpha s + \beta_u \\ v = \alpha t + \beta_v \end{cases}, \tag{6.19}$$

avec

$$\alpha = \frac{z-1}{z},\ \beta_u = \frac{x}{z} \ \text{ et } \ \beta_v = \frac{y}{z}. \tag{6.20}$$

Nous trouvons un estimateur de $\mathbf{X}$ en résolvant le système d'équations linéaires ci-dessus pour α, β_u and β_v, ce qui est équivalent à une triangulation du point classique. Le nombre de paramètres définit la dimension du *point visuel*. Nous nommons 3g un tel *modèle géométrique* définit par ses 3 paramètres. On peut aisément étendre ce modèle aux congruences de droites qui suivent des équations linéaires, comme l'équation (6.19), mais dont les rayons ne passent pas par un point 3D dans l'espace. De façon plus générique on peut écrire

$$\mathbf{q} \in P \iff \mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{b}. \tag{6.21}$$

Dans le cas précédent où tous les rayons intersectent un point 3D dans l'espace, $\mathbf{A} = \alpha \mathbf{I}_{22}$ and $\mathbf{b} = (\beta_u, \beta_v)$. On introduit alors deux nouveaux modèles, 4g :

$$\mathbf{A} = \begin{pmatrix} \alpha_u & 0 \\ 0 & \alpha_v \end{pmatrix},\ \mathbf{b} = \begin{pmatrix} \beta_u \\ \beta_v \end{pmatrix}, \tag{6.22}$$

et 6g :

$$\mathbf{A} = \begin{pmatrix} \alpha_{us} & \alpha_{ut} \\ \alpha_{vs} & \alpha_{vt} \end{pmatrix},\ \mathbf{b} = \begin{pmatrix} \beta_u \\ \beta_v \end{pmatrix}. \tag{6.23}$$

Les trois modèles peuvent s'interpréter de manière intuitive. Le plus simple, 3g, est la visualisation d'un point 3D à une certain profondeur : lorsque l'observateur effectue un déplacement dans une direction donnée, l'image du point se déplace dans la même direction, d'une distance proportionnelle à ce déplacement, dont le coefficient de proportionnalité est constant et dépend de la profondeur du point. Le modèle 4g décrit un comportement plus général où une direction quelconque de déplacement de l'image du point n'est pas conservée, parce que le ratio de déplacement entre l'image et l'observateur n'est pas le même selon l'axe vertical ou horizontal. Mais un déplacement de l'observateur sur l'axe horizontal provoque un déplacement de l'image sur ce même axe. Cela n'est pas vrai pour le 6g qui décrit le comportement linéaire le plus général.

Calculer les 3, 4 ou 6 paramètres géométriques de la congruence de droites peut se poser comme un problème des moindres carrés. On ajuste un modèle, à 3, 4 ou 6 paramètres, aux K échantillons de rayons 4D $\{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$. Pour cela on minimise la somme des carrés des distances de Mahalanobis aux données échantillonnées, connaissant la matrice de covariance de rang 2 de l'erreur de mesure $\mathbf{\Sigma}_{\mathrm{u,s}}$ de chaque échantillon. Il est en effet possible de définir une distance de Mahalanobis grâce à cette matrice car le vecteur résiduel se situe dans le sous-espace engendré par les vecteurs propres $\mathbf{e_1}$ et $\mathbf{e_2}$ associés aux valeurs propres non nulles μ_1 et μ_2 de $\mathbf{\Sigma}_{\mathrm{u,s}}$. Cette hypothèse procure une contrainte supplémentaire pour la construction du résidu.

Expression du résidu Soit $\mathbf{p} = (p_u, p_v, p_s, p_t)$ un rayon échantillonné (correspondant à un point image dans une caméra), et soit $\mathbf{r} = (r_u, r_v, r_s, r_t)$ son vecteur d'erreur résiduelle : $\mathbf{q} = \mathbf{p} + \mathbf{r} \in P$. Nous rappelons ici que P est la congruence de droites associée au *point visuel*. $\mathbf{r}$ est contraint d'appartenir au sous-espace vectoriel engendré par les vecteurs propres $\mathbf{e_1}$ et $\mathbf{e_2}$ associés aux valeurs propres μ_1 et μ_2 respectivement. Notre but est de trouver les paramètres du modèle qui minimisent le carré de la norme de Mahalanobis de $\mathbf{r}$. D'une part nous avons

$$\mathbf{p} + \mathbf{r} \in P \iff \begin{pmatrix} p_u + r_u \\ p_v + r_v \end{pmatrix} = \mathbf{A} \begin{pmatrix} p_s + r_s \\ p_t + r_t \end{pmatrix} + \mathbf{b} \tag{6.24}$$

et d'autre part

$$\mathbf{r} = r_1 \mathbf{e}_1 + r_2 \mathbf{e}_2. \tag{6.25}$$

Par substitution nous obtenons l'expression du résidu dans une base de vecteurs propres :

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = (\mathbf{E_u} - \mathbf{A}\mathbf{E_s})^{-1} \left(\mathbf{A} \begin{pmatrix} p_s \\ p_t \end{pmatrix} + \mathbf{b} \right), \tag{6.26}$$

avec

$$\mathbf{E_u} = \begin{pmatrix} \mathbf{e}_{1,u} & \mathbf{e}_{2,u} \\ \mathbf{e}_{1,v} & \mathbf{e}_{2,v} \end{pmatrix} \text{ and } \mathbf{E_s} = \begin{pmatrix} \mathbf{e}_{1,s} & \mathbf{e}_{2,s} \\ \mathbf{e}_{1,t} & \mathbf{e}_{2,t} \end{pmatrix}. \tag{6.27}$$

La fonction de coût Pour chaque échantillon $\mathbf{p}_k, k \in [1, K]$, soit $\mathbf{r}_k$ son résidu, $\mu_{k,1}$ et $\mu_{k,2}$ les valeurs propres associées à l'incertitude de mesure de l'échantillon. La fonction de coût à minimiser est

$$\|f(\mathbf{A}, \mathbf{b})\|^2 = \sum_{k=1}^{K} \|f_k(\mathbf{A}, \mathbf{b})\|^2_{\mathbf{D_k}}, \tag{6.28}$$

où

$$\|f_k(\mathbf{A}, \mathbf{b})\|^2_{\mathbf{D_k}} = \mathbf{r}_k^\intercal \mathbf{D}_k^{-1} \mathbf{r}_k, \text{ avec } \mathbf{D}_k = \begin{pmatrix} \mu_{k,1} & 0 \\ 0 & \mu_{k,2} \end{pmatrix}. \tag{6.29}$$

Dans le cas du modèle 3g, on peut montrer que cette optimisation est équivalente à la triangulation classique d'un point 3D (comme dans la technique d'ajustement de faisceaux). Cependant, au lieu de minimiser la somme des carrés des erreurs de reprojection, on minimise plutôt la somme des carrés des erreurs sur les rayons 4D. Chaque contribution est pondérée par l'inverse de l'incertitude propagée du plan image sur la paramétrisation 4D *light slab*. Cette formalisation apporte un énorme avantage car elle permet de modéliser des *points visuels* plus complexes de dimension supérieure à 3 (plus de 3 paramètres).

modèle photométrique En supposant que le *point visuel* P est lambertien, tous les rayons associés on la même couleur quelle que soit leur direction. Cela signifie que sa radiance $\mathbf{I} = (R, G, B)$ est constante par rapport à $\mathbf{s}$ et que le *modèle photométrique* a 3 paramètres. Nous nommons ce modèle 3p. Dans le cas général, l'hypothèse lambertienne n'est pas nécessairement vérifiée : la radiance du *point visuel* dépend du point de vue. Comme pour la géométrie de l'espace plénoptique, nous linéarisons la couleur $\mathbf{I}$ en fonction des deux premières premières coordonnées $\mathbf{s} = (s, t)$ du *light slab* : $\mathbf{I}(\mathbf{s}) = \mathbf{A}\mathbf{s} + \mathbf{I}_0$. Le nombre de paramètres à trouver est de 9 (6 pour la matrice $\mathbf{A}$ et 3 pour le vecteur $\mathbf{I}_0$). Il y a $3K$ mesures scalaires (3 canaux fois le nombre de cameras qui voient le point). On nomme 9p un tel modèle. Chaque couleur est pondérée par l'inverse de la matrice variance $\mathbf{\Sigma}_{\mathbf{I},\mathbf{s}}$ de la distribution jointe, et les paramètres sont estimés en résolvant un problème des moindres carrés.

Sélection de modèle Ajuster un modèle ayant beaucoup de paramètres permet généralement de diminuer l'erreur résiduelle ; mais lorsque le nombre de paramètres à estimer se rapproche du nombre d'échantillons, il y a un risque de sur-ajustement des données échantillonnées ; en découle une mauvaise prédiction du mouvement et de la couleur du *point visuel*. On utilise communément des techniques de sélection de modèle pour trancher entre différents modèles qui varient en terme de d'efficacité à s'ajuster aux données et en terme de complexité (nombre de paramètres). Après avoir estimé les paramètres de nos trois modèles géométriques, nous appliquons un critère d'information bayésien (BIC) (Schwarz, 1978) pour déterminer le meilleur modèle pour chaque *point visuel*. En notant $\hat{\mathbf{A}}$ et $\hat{\mathbf{b}}$ la matrice et le vecteur qui

contiennent les estimateurs des paramètres, la formule du BIC devient :

$$\mathrm{BIC} = \|f(\hat{\mathbf{A}}, \hat{\mathbf{b}})\|^2 + n.\ln K \tag{6.30}$$

où K est le nombre d'échantillons, n le nombre de paramètres et $\|f(\hat{\mathbf{A}}, \hat{\mathbf{b}})\|_2^2$ est -2 fois la log-vraisemblance des estimateurs. Pour chaque lot d'échantillons on sélectionne le modèle qui minimise le BIC. La figure 6.4 montre le résultat d'une sélection de modèle sur le jeu de données *tarot* tiré de la *Stanford Light Field Archive*. On peut remarquer que le modèle 3g + 9p suffit à décrire les zones opaques et diffuses telles que les cartes de tarot, mais il est supplanté par des modèles géométriques plus complexes comme 6g + 9p dans les zones réfractives telles que la boule transparente.

6.6 Rendu

Nous démontrons un exemple d'application de nos modèles de *point visuel* en synthétisant une nouvelle vue. Connaissant les paramètres d'une vue cible $\mathbf{C}$, $\mathbf{R}$ et $\mathbf{K}$, on cherche à trouver, pour chaque *point visuel* de la scène, l'orientation et la radiance de son rayon associé qui passe par le centre optique $\mathbf{C}$. Le but de la synthèse de nouvelle vue est de trouver la couleur de tous les pixels dans la vue cible. Pour y parvenir nous devons trouver pour chaque pixel de cette vue quels *points visuels* ont un rayon qui passe par ce pixel, et mélanger leur radiance : c'est le procédé appelé *backward warping*. En pratique, ce problème est computationnellement trop complexe ; il lui est donc préféré un *forward warping* : pour chaque *point visuel* on calcule le rayon du *point visuel* passant par le centre optique de la caméra cible, sa radiance, et on peint le pixel en conséquence. Pour chaque *point visuel*, on trouve le rayon lumineux capturé par la vue cible par intersection entre le modèle de congruence de droites (qui est un plan dans le *light slab* 4D) et le faisceau de rayons associé à la caméra (voir figure 6.5). Par exemple l'intersection entre le modèle 6g représenté par les paramètres $(\alpha_{u,}, \alpha_{ut}, \alpha_{vs}, \alpha_{vt}, \beta_{cu}, \beta_{cv})$ et la caméra de centre optique $\mathbf{C} = (C_x, C_y, C_z)$ a pour coordonnées en $\mathbf{s}$

$$\mathbf{s} = \begin{pmatrix} \alpha_{us} - \alpha_c & \alpha_{ut} \\ \alpha_{vs} & \alpha_{vt} - \alpha_c \end{pmatrix}^{-1} \begin{pmatrix} \beta_{cu} - \beta_u \\ \beta_{cv} - \beta_v \end{pmatrix} \tag{6.31}$$

avec

$$\alpha_c = \frac{C_z - 1}{C_z},\ \beta_{cu} = \frac{C_x}{C_z},\ \beta_{cv} = \frac{C_y}{C_z}. \tag{6.32}$$

Les coordonnées en $\mathbf{u}$ peuvent être calculées soit grâce au modèle du *point visuel*, soit grâce au centre optique de la caméra. Il est maintenant facile de trouver le point image associé au rayon reconstruit, en projetant le point (s, t) sur le plan capteur de la caméra :

$$\mathbf{x} = \mathbf{KR} \left(\begin{pmatrix} s \\ t \\ 0 \end{pmatrix} - \mathbf{C} \right). \tag{6.33}$$

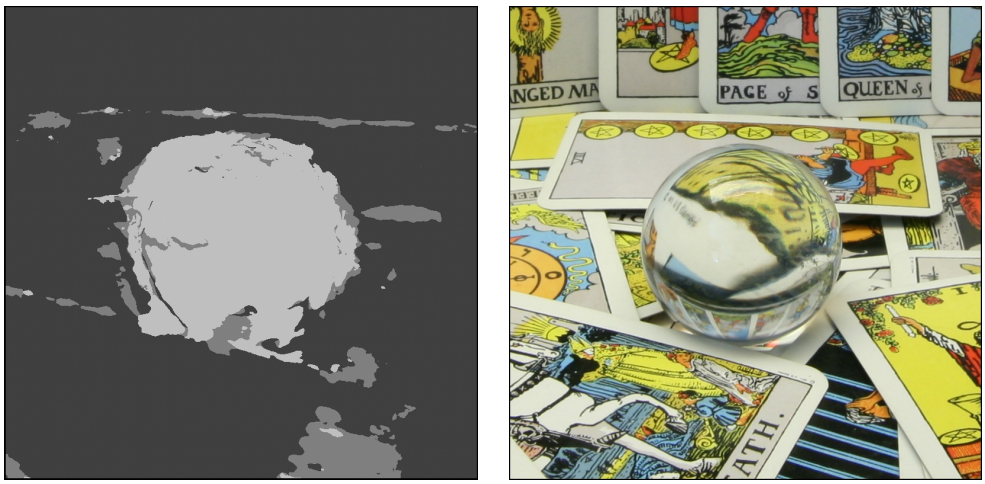


Fig. 6.4 – *Sélection de* modèle géométrique *par BIC sur le jeu de données tarot coarse. Gris clair :* $6\mathrm{g}+9\mathrm{p}$. *Gris :* $4\mathrm{g}+9\mathrm{p}$. *Gris foncé :* $3\mathrm{g}+9\mathrm{p}$. *Le BIC segmente ici les zones lambertiennes (cartes de tarot) des zones réfractives ou spéculaires (boule de verre transparente).*

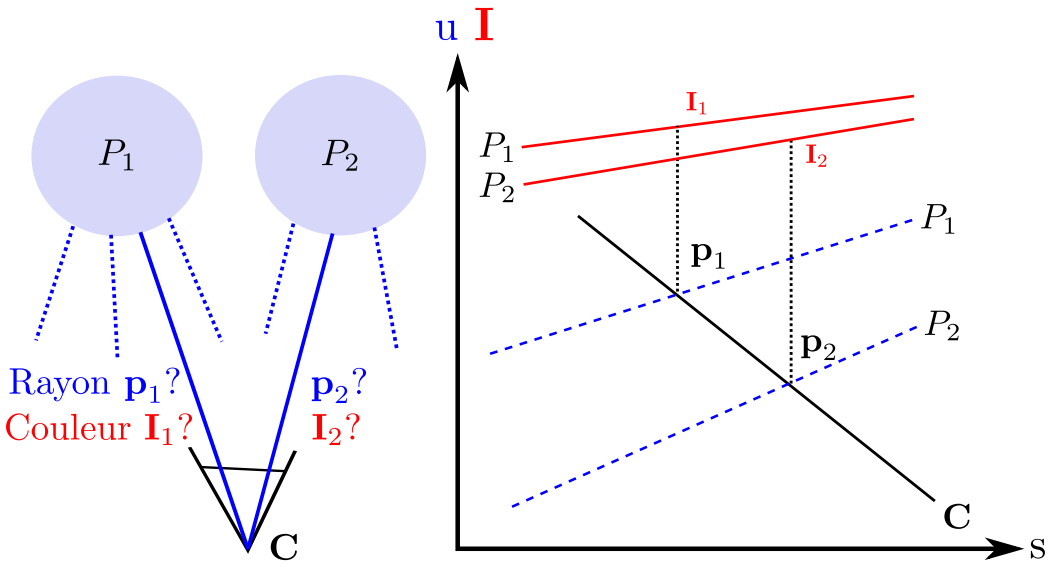


Fig. 6.5 – *Rendu du* point visuel. P_1 *et* P_2 *sont deux modèles de* points visuels. *Étant donnée une caméra cible de centre* $\mathbf{C}$, *il s'agit de trouver le rayon* $\mathbf{p}$ *passant par* $\mathbf{C}$ *et issu de chaque modèle, ainsi que sa couleur* $\mathbf{I}$. *La droite noire représente l'ensemble des rayons qui passent par le centre optique de la caméra cible* $\mathbf{C}$, *les droites bleues en pointillés représentent les relations géométriques linéaires entre* $\mathbf{u}$ *et* $\mathbf{s}$, *et les droites rouges continues représentent les relations photométriques entre* $\mathbf{I}$ *et* $\mathbf{s}$. *Les rayons interpolés sont les intersections* $\mathbf{p}_1$ *et* $\mathbf{p}_2$, *et leurs radiances sont respectivement* $\mathbf{I}_1$ *et* $\mathbf{I}_2$.

On peut noter que certains de ces points peuvent tomber en dehors du champ de vue de la caméra cible, puisque tous les *points visuels* ne sont pas forcément visibles dans une vue donnée. Nous trouvons la couleur du rayon en substituant les coordonnées en $\mathbf{s}$ dans le *modèle photométrique* estimé (voir figure 6.5).

Une fois que le point image destination est calculé, on *éclabousse* la couleur du *point visuel* sur un voisinage du point image. Nous rappelons que le *splatting* consiste à accumuler le rendu de primitives 3D (généralement des disques ou des ellipses) centrées sur le point 3D et orientées par une normale reconstruite. Parce que nous n'avons aucune information sur les normales, nous éclaboussons des carrés uniformes de la taille d'un pixel, ce qui revient à accumuler des contributions bilinéaires dans un voisinage 2×2. Le rendu par éclaboussures est fait couramment en trois passes : le calcul de la *visibilité*, le *mélange* et la *normalisation*. Nous passons la première, car calculer un *z-buffer* n'a de sens que lorsqu'une profondeur est définissable. Seul le *modèle géométrique* 3g contient de l'information de profondeur ($z = 1/(1-\alpha)$), mais même cette profondeur perçue peut être une illusion (comme dans l'illusion de la pièce de monnaie en lévitation, qui utilise deux miroirs paraboliques concaves [1]). Nous procédons directement par projeter tous les points visuels par éclaboussure, en accumulant les poids des contributions dans un canal `alpha` ; c'est l'étape de *mélange*. Enfin on *normalise* l'image en divisant par le canal `alpha`.

Cohérence épipolaire Comme il est mentionné par Buehler *et al.* (2001), un rayon qui passe par le centre de projection d'une caméra source « devrait être trivialement reconstruit à partir des rayons échantillonnés ». Notre algorithme de rendu ne satisfait pas pleinement cette propriété car les rayons qui forment le *point visuel* n'appartiennent pas nécessairement au vecteur original d'échantillons. En d'autres termes le modèle ne s'ajuste pas parfaitement aux données. Ce serait en effet le cas si le modèle ajusté passait exactement par tous les points 4D échantillonnés. Néanmoins le processus d'optimisation s'évertue à faire en sorte que le modèle estimé soit aussi proche que possible de l'ensemble des rayons échantillonnés en minimisant la distance de Mahalanobis.

Déviation angulaire minimale Un algorithme d'IBR idéal devrait garantir que les vues sources qui sont plus proches de la vue cible contribuent plus à la couleur finale du rendu. La déviation angulaire est une mesure usuelle de proximité. Une telle propriété est satisfaite par notre algorithme grâce au fait que nous modélisons la radiance comme une fonction linéaire des coordonnées en $\mathbf{s}$. Puisqu'un rayon qui est proche d'un autre en angle l'est aussi par rapport aux coordonnées en $\mathbf{s}$, sa radiance devrait être similaire.

Sensibilité à la résolution La couleur d'un *point visuel* peut être une moyenne pondérée des échantillons de couleurs présents dans les images sources, comme dans la plupart des techniques d'IBR. Chaque poids est fonction de la quantité de détails qu'une vue capture. Par conséquent une vue source contribue plus que les autres si

1. http://berkeleyphysicsdemos.net/node/724

elle est proche du *point visuel* observé, ou si sa distance focale est plus grande (elle « zoome » sur le *point visuel*). Ce rôle est ici joué par la matrice de covariance $\mathbf{\Sigma_{I,s}}$, qui pondère les vues sources lors de l'ajustement du *modèle photométrique*. Plus la caméra est loin, ou plus sa résolution est faible, plus l'incertitude représentée par la covariance est grande. On peut se le visualiser comme un cône d'incertitude projeté sur le plan $z = 0$. Donc un rayon qui a une grande variance, plus « erroné », contribue moins à l'ajustement du modèle. Au contraire une caméra qui est proche de la scène ou qui a une grande distance focale mesure précisément le *point visuel*, ce qui conduit à une variance plus faible.

Continuité Cette nouvelle méthode de rendu assure la continuité par rapport au changement de point de vue, une autre propriété désirable pour un algorithme d'IBR (Buehler *et al.*, 2001). Quand nous bougeons le point de vue cible de manière continue, la congruence de droites de la caméra cible, qui est un plan en 4D, bouge aussi de manière continue. Comme l'orientation et la radiance du rayon reconstruit sont calculées par intersection de deux congruences de droites, deux plans en 4D, elles bougent également de façon continue.

Remplissage des trous La technique de *forward warping* ne permet pas systématiquement de couvrir l'entièreté de l'image à synthétiser, ce qui conduit à des « trous » (voir figure 6.6). N'importe quel algorithme d'*inpainting* peut résoudre ce problème de remplissage tant qu'il est continu par rapport aux images sources (pas de changement brusque d'intensité d'un pixel original à son voisin reconstruit par l'*inpainting*). Nous utilisons ici l'algorithme `PUSH/PULL` introduit dans Gortler *et al.* (1996). C'est équivalent à effectuer une diffusion isotropique dans les zones à remplir.

6.7 Expériences

Notre méthode est conçue pour être la plus générique possible, et n'est donc pas limitée à un type de caméra spécifique ou à un agencement de caméras particulier. Nous effectuons nos expériences sur des images originales, non rectifiées, capturées par un ensemble de caméras (Wilburn *et al.*, 2005). Pour calibrer les vues nous utilisons la bibliothèque openMVG (Moulon *et al.*, 2013), qui calcule les paramètres extrinsèques et intrinsèques des caméras et qui corrige la distorsion (qui n'est pas prise en compte dans nos modèles). Le placement de caméras utilisé dans les expériences, décrit par la figure 6.7, se compose de 24 vues arrangées en un carré interne 3×3 et un carré externe 5×5 (on enlève la vue centrale, car on cherche par la suite à la synthétiser). Il s'agit à l'origine d'un carré 9×9 auquel nous avons enlevé une colonne et une ligne sur deux dans le but d'exacerber la sparsité du champ lumineux capturé. Les vues enlevées sont mises de côté et servent de référence pour évaluer numériquement nos résultats. Un algorithme de flux optique [2] traite chaque paire de

2. https://github.com/facebook/Surround360/blob/master/surround360_render/source/optical_flow/PixFlow.h

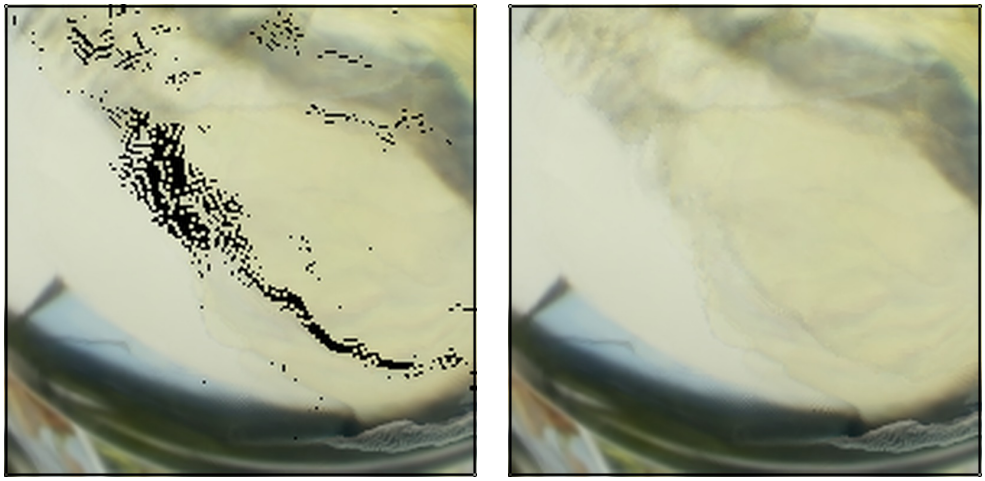


Fig. 6.6 – *Remplissage des trous grâce au* `PUSH/PULL`. *Les trous sont causés par des occultations ou une extrapolation lointaine. Une vue très distante a été synthétisée dans cet exemple pour exacerber ces trous et l'effet de l'algorithme.*

vues adjacentes, de sorte que l'on obtient autant d'échantillons que de flux optiques calculés (voir figure 6.7). La stratégie pour remplir ces vecteurs d'échantillons est la suivante :

- nous calculons le flux de la vue centrale vers les vues du carré intérieur (8 flux optiques),
- nous ajoutons les positions dans les vues internes aux vecteurs d'échantillons respectifs,
- des flux optiques sont calculés des vues internes vers les vues extérieures voisines (16 flux optiques),
- en partant des positions précédentes, les nouveaux échantillons de position sont trouvés via les flux optiques calculés.

Une fois que les vecteurs d'échantillons sont remplis avec les couleurs et les positions 2D, converties en 4D via la paramétrisation *light slab* (voir section 6.4), nous calculons les modèles de *point visuel* en utilisant Ceres Solver (Agarwal *et al.*, 2017), qui minimise la norme L^2 des résidus blocs $f_k(\mathbf{A}, \mathbf{b}) = \left(\frac{r_{k,1}}{\sqrt{\mu_{k,1}}}, \frac{r_{k2}}{\sqrt{\mu_{k2}}}\right)$ grâce à l'algorithme `DENSE_QR`. Quatre modèles sont testés : $3\mathrm{g}+3\mathrm{p}$, $3\mathrm{g}+9\mathrm{p}$, $4\mathrm{g}+9\mathrm{p}$ et $6\mathrm{g}+9\mathrm{p}$. Le premier chiffre indique le nombre de paramètres du *modèle géométrique* ($\mathbf{u}$ en fonction de $\mathbf{s}$), tandis que le second indique le nombre de paramètres du *modèle photométrique* ($\mathbf{I}$ en fonction de $\mathbf{s}$). Le modèle $3\mathrm{g}+3\mathrm{p}$ correspond à un point 3D avec une couleur RGB indépendante du point de vue, c'est-à-dire un point lambertien. En utilisant notre algorithme, on effectue le rendu de la vue centrale, une vue dans le coin supérieur droit entre le carré interne et le carré externe, et nous extrapolons une vue à droite, hors de la région du champ lumineux échantillonnée (figure 6.7). Les vues résultantes sont rognées à la dimension 800×800.

La figure 6.8 montre la vue centrale synthétisée avec notre méthode, la vue originale qui a été enlevée, la valeur absolue de la différence par pixel entre les deux images et le résidu final à la fin de l'optimisation du *modèle géométrique* à 3 paramètres.

Mis à part le jeu *tarot coarse*, tous les résultats sont très proches des images originales. La plupart des artefacts se produisent dans la boule de verre, où les rayons lumineux sont déviés par réfraction. Le modèle à 3 paramètres échoue à modéliser le comportement de la lumière déviée car il est improbable que les rayons émanent du même point dans l'arrière-plan (les cartes derrière la boule). Les modèles avec 4 ou 6 paramètres produisent de bien meilleurs résultats, comme le montre la figure 6.9. La même interprétation peut être faite pour les spécularités sur le trésor ou sur le bracelet. De plus le jeu de données *bracelet* montre que dans les régions non texturées le modèle échoue à s'ajuster aux données altérées par un flux optique erroné. Néanmoins cela n'affecte pas le rendu parce que la position estimée du rayon reconstruit importe peu dans les régions non texturées. La figure 6.9 montre des vues rapprochées des images synthétisées, pour démontrer l'effet du rendu avec différents modèles. Plus nous utilisons de paramètres, plus les résultats sont fidèles aux images originales. Cela est appuyé par des résultats numériques présents dans le tableau 6.1. On remarque assez peu de différences en changeant de modèle pour le rendu de la vue centrale, contrairement aux deux autres vues où l'on voit clairement l'intérêt d'augmenter la dimension du *point visuel*.

6.8 Conclusion

Nous avons proposé une nouvelle approche pour reconstruire le champ lumineux, basée sur l'approximation d'une congruence de droites qui forme le champ lumineux 4D. Alors que la plupart des méthodes de reconstruction de champ lumineux estiment dans un premier temps la géométrie 3D de la scène, notre méthode travaille directement sur la façon dont la scène est *perçue* au travers des images, sans la reconstruire explicitement. Dans cette représentation, chaque *point visuel* est représenté par des paramètres géométriques et photométriques. La représentation géométrique d'un *point visuel* est un ensemble de droites à deux degrés de libertés, aussi appelé congruence de droites, qui contient l'information sur la façon dont le point bouge lorsque la caméra bouge. Les paramètres photométriques contiennent l'information sur la façon dont varie la variance en fonction du point de vue. Par exemple un point sur une surface lambertienne est représenté par un faisceau de droites passant par le point 3D, et une couleur unique, ce qui fait 3 paramètres géométriques et 3 paramètres photométriques. Nous avons proposé des modèles avec 3, 4 ou 6

	Vue $(8, 8)$		Vue $(11, 11)$		Vue $(14, 8)$	
	PSNR	DSSIM	PSNR	DSSIM	PSNR	DSSIM
3g + 3p	26.37	64	23.61	109	24.00	102
3g + 9p	26.34	64	23.65	109	24.85	99
4g + 9p	26.32	64	25.08	89	24.71	96
6g + 9p	**26.44**	**63**	**25.57**	**74**	**27.20**	**69**

Table 6.1 – *Résultats numériques sur le jeu de données tarot coarse.*

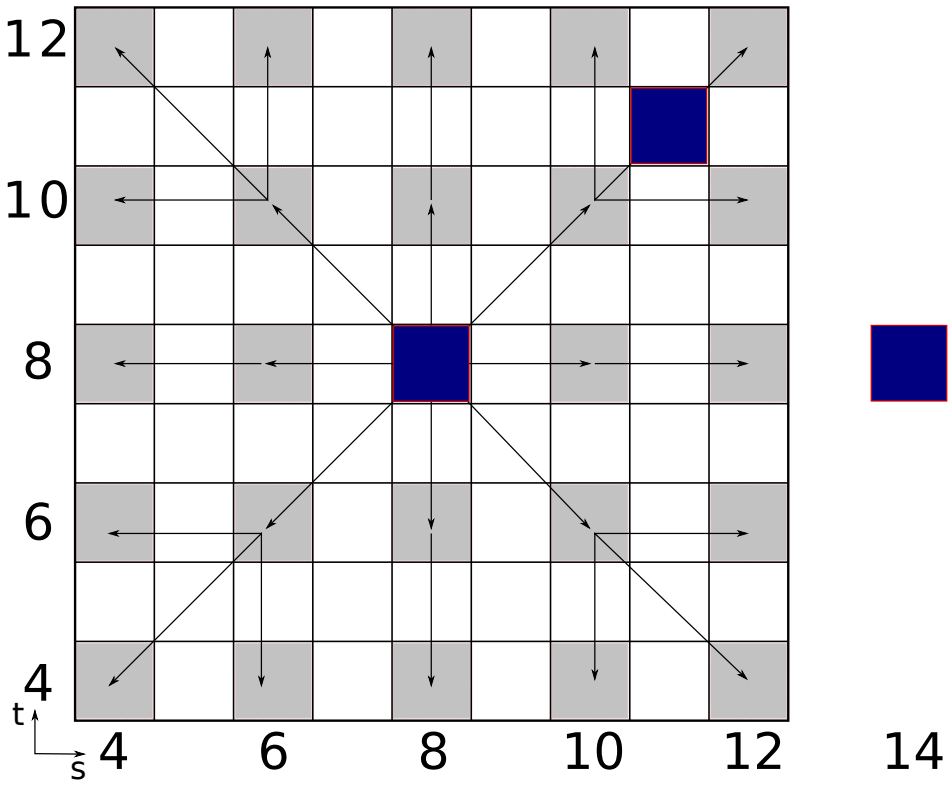
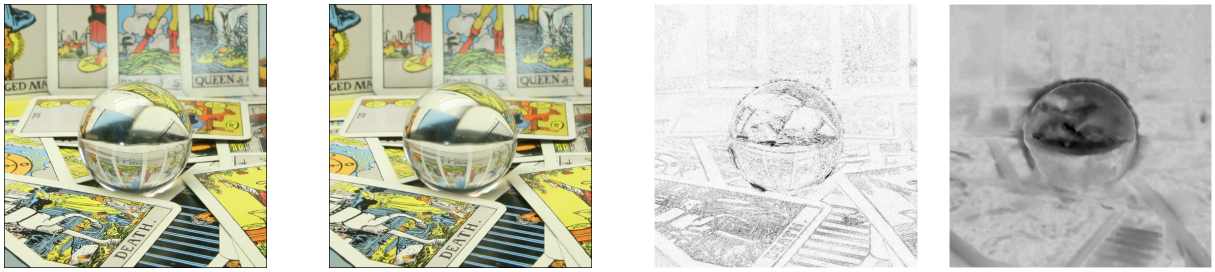


Fig. 6.7 – *Configuration de l'ensemble de caméras utilisé pour nos expériences. Les caméras des jeux de données de Stanford sont arrangées dans le même plan* (s,t)*. Chaque case du tableau représente une vue du jeu de donnée original. Les cases grisée sont les vues utilisées pour échantillonnées l'espace plénoptique. Les flèches indiquent comment on applique le flux optique. Les cases bleues sont les vues que nous synthétisons pour évaluer la méthode. Nous interpolons les vues* $(8,8)$ *et* $(11,11)$*, et nous extrapolons la vue* $(14,8)$*.*

paramètres géométriques, et 3 ou 9 paramètres photométriques, qui peuvent modéliser des effets optiques comme des réflexions, des réfractions, ou des variations d'indice de réfraction, de même que des surfaces non lambertiennes. Les expériences montrent que les différents modèles de *point visuel* sont capables de traiter les phénomènes optiques complexes qui ne peuvent pas être modélisés pas une reconstruction 3D explicite. Une méthode de sélection de modèle permet de séparer les points de la scène qui sont lambertiens ou presque lambertiens des *points visuels* qui ne peuvent être modélisés par un simple faisceau de rayons.

Le modèle du *point visuel* pourrait être enrichi, par exemple en modélisant les congruences de droites non linéaires, comme celles qui peuvent être causées par des surfaces sphériques ou cylindriques. L'algorithme de rendu pourrait aussi prendre en compte la visibilité de chaque *point visuel*, et faire le rendu d'un *point visuel* donné seulement si les rayons qui étaient utilisés pour calculer le modèle sont assez proches du point de vue synthétisé. Une autre extension serait d'incorporer une dimension temporelle dans nos modèles de *points visuels*, et de calculer des congruences de droites qui varient avec le temps. Cela pourrait être utilisé afin de synthétiser des vues à partir de vidéos asynchrones ou d'ensembles de photographies prises à des



Tarot fine PSNR: 35.03 DSSIM: 16

Tarot coarse PSNR: 26.34 DSSIM: 64

Bracelet PSNR: 39.93 DSSIM: 5

Chest PSNR: 35.01 DSSIM: 272

Image originale Résultats Valeur absolue Résidu final

Fig. 6.8 – *Résultats sur plusieurs jeux de données de la Stanford Light Field Archive. On compare les images synthétisées avec les images originales. On montre aussi la valeur absolue de la différence entre les deux, et la valeur finale de la fonction de coût. Le modèle* $3g + 9p$ *est utilisé dans cette expérience. On remarque que les valeurs d'erreur les plus hautes sont localisées dans les zones réfractives et spéculaires. Dans les résultats du jeu bracelet, la zone noire dans la valeur finale de la fonction de coût révèle une mauvaise reconstruction, qui pourtant n'affecte pas la qualité du rendu (faible erreur), à cause du manque de texture dans cette zone.*

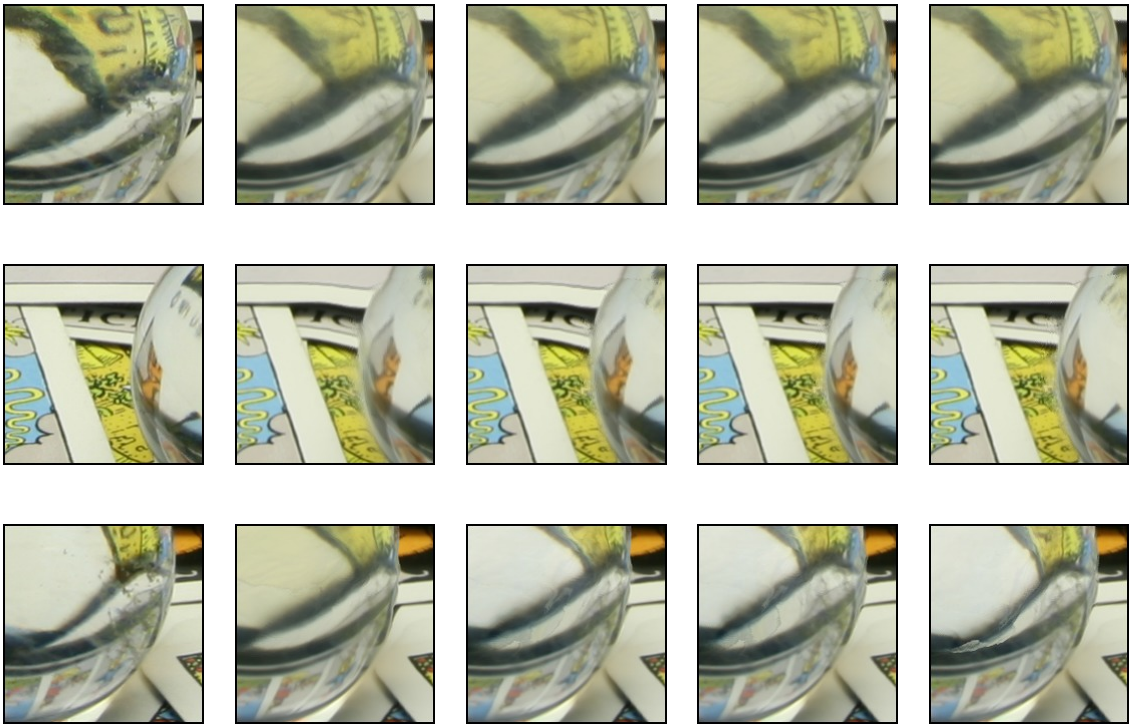


Fig. 6.9 – *Résultats sur des parties difficiles du jeu de données tarot coarse. Première ligne : vue centrale* $(8, 8)$*. Deuxième ligne : vue en haut à droite* $(11, 11)$*. Troisième ligne : vue extrapolée à droite* $(14, 8)$*. On voit que les artefacts principaux sont partiellement réparés par un modèle avec plus de paramètres.*

endroits et à des moments différents.

7

Application à la vision longue distance et travaux futurs

7.1 Introduction

Dans ce chapitre nous développons trois axes de recherche indépendants liés au rendu basé image et à la vision plénoptique. Les contributions scientifiques présentées ici sont incomplètes mais nous il nous semblait important d'aborder ces thèmes même si du travail reste à accomplir. Dans un premier temps nous introduisons le problème du placement idéal de caméras contraintes qui optimise le rendu d'une nouvelle vue. Cette question n'est pas nouvelle en reconstruction 3D, mais elle n'a jamais été posée dans le domaine du rendu basé image. Nous avons développé une fonction de coût à minimiser pour trouver le placement optimal, basée sur l'incertitude de la fonction de déformation. Nous étudions l'évolution de cette fonction de coût en fonction des paramètres des caméras pour une configuration simple de caméras contraintes à se déplacer sur un axe. Dans un second temps nous abordons le sujet de la vision à longue distance, application directe de nos algorithmes de rendu basé image, qui consiste à combiner des images sources pour générer un nouveau point de vue, plus proche du sujet observé. Nous comparons cette approche avec celle qui consiste simplement à générer le même point de vue source avec une distance focale plus grande, ce qui simule un zoom. Enfin nous appliquons nos algorithmes de rendu basé image à la vision d'objets camouflés. Nous montrons que si le problème de la reconstruction d'objets est déjà traité dans de nombreux articles, celui du rendu de la nouvelle vue (sans le camouflage) soulève encore des questions nouvelles et intéressantes.

7.2 Le dispositif optique idéal

Au cinéma ou à la télévision, nous sommes souvent contraints de placer les caméras à bonne distance de la scène, et par conséquent d'utiliser des longues focales pour

se « rapprocher » de l'objet filmé. Celles-ci présentent cependant des inconvénients majeurs comme la déformation de l'arrière-plan et la perte de relief, qui rendent difficiles les prises de vues stéréoscopiques avec un relief suffisant. Une solution à ce problème consiste à créer un nouveau point de vue à partir d'un ensemble de prises de vue : c'est le problème du rendu basé image qui a été exploré dans les chapitres précédents. L'objectif de cette section est de donner les clés pour trouver la meilleure configuration de caméras de prise de vue, sous contraintes, pour la synthèse d'une vue virtuellement « plus proche » de la scène. Si le problème a été maintes fois traité pour la reconstruction 3D, il est nouveau dans le domaine du rendu basé image.

Nous étudions le problème du placement optimal de plusieurs caméras contraintes (par exemple sur un demi cercle centré sur la scène, ou dans un plan) pour la synthèse de nouvelles vues. Cette synthèse est souvent précédée d'une phase de reconstruction 3D approximative, qu'il est nécessaire de prendre en compte dans notre problème de placement optimal. La notion d'optimum est définie comme la minimisation d'une fonctions de coût ; pour cette dernière nous avons choisi l'incertitude de projection des vues sources sur la vue à générer par l'algorithme de rendu. Nous l'exprimons en propageant l'incertitude sur la géométrie et l'incertitude de mesure des vues sources sur la nouvelle vue. Nous observons l'influence de l'*interoculaire* (ou *entraxe*) et de la distance focale des caméras sur l'erreur projetée, pour des distributions de points aléatoires à diverses profondeurs.

7.2.1 Travaux antérieurs

Dans le domaine de l'IBR Le rendu basé image a été l'objet de nombreux travaux (Shum *et al.*, 2008). Nous ne reviendrons pas sur les méthodes de l'état de l'art, qui ont été discutées dans les chapitres précédents. Notons qu'il est souhaitable que la méthode proposée pour trouver le placement optimal de caméras contraintes soit indépendante de l'algorithme de rendu et de l'algorithme d'estimation du *proxy géométrique*. Si le problème que nous posons est nouveau, notre approche pour trouver une solution, elle, s'inspire du logiciel de capture de Davis *et al.* (2012) et de leur « critère de couverture ». Proposant un échantillonnage du champ de lumière à partir de points de vue non structurés, il leur fallait un critère pour contrôler l'échantillonnage des vues qui ne sont pas arrangées de manière régulière comme sur une grille de caméras par exemple. Leur objectif était de couvrir la plus large portion de la scène observable avec le moins de points de vue possibles, tout en minimisant l'erreur de reprojection. Mais il ne s'agit pas d'une optimisation à proprement parler : leur placement ne découle pas de la minimisation d'une fonction de coût, mais les points de vue sont retenus de façon itérative dès que l'erreur de reprojection passe au-dessus d'un seuil, ce qui garantit que l'entraxe entre les caméras est assez grand (pour maximiser la zone couverte par les points de vue) pour une erreur de reprojection donnée. Plus récemment Waechter *et al.* (2016) ont proposé une méthode pour estimer la qualité de n'importe quelle méthode de rendu par une technique de validation croisée qui exploite l'erreur de reprojection dans chaque vue source.

Dans le domaine de la reconstruction 3D Dans le domaine de la reconstruction 3D, le problème de NBV (*Next Best View*) consiste à estimer les paramètres optimaux de caméra qui minimisent l'incertitude sur la géométrie par propagation d'erreur (Haner et Heyden, 2012). La matrice de covariance associée est obtenue par projection de l'erreur de mesure sur la géométrie 3D reconstruite (Beder et Steffen, 2006). Recker *et al.* (2012) présentent une nouvelle méthode pour estimer cette incertitude basée sur une mesure de l'erreur angulaire, mais elle n'apporte aucune information sur sa direction. Au contraire, la matrice de covariance du point 3D reconstruit nous informe à la fois sur l'amplitude de l'incertitude et sur sa direction. Supposons que l'ont cherche à estimer l'incertitude sur la position d'un point $\mathbf{X}$ de la scène à reconstruire, dans le cas général où K caméras sont utilisées. Supposons que $\mathbf{X}$ est visible par ces K caméras ($\mathbf{C}_1$, $\mathbf{C}_2$, …, $\mathbf{C}_K$) et ses images respectives dans ces caméras sont $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_k$. Les mesures de $\mathbf{X}$ sur le plan capteur des caméras suivent une loi normale de covariances $\mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_1}$, $\mathbf{\Sigma}_{\mathbf{x}_2\mathbf{x}_2}$, …, $\mathbf{\Sigma}_{\mathbf{x}_K\mathbf{x}_K}$. En linéarisant localement, on peut supposer que $\mathbf{X}$ suit une loi normale de matrice de covariance $\mathbf{\Sigma}_{\mathbf{XX}}$. Hess-Flores *et al.* (2014) nous donnent une expression de cette matrice de covariance :

$$\mathbf{\Sigma}_{\mathbf{XX}} = (\mathbf{N}^{-1})_{1:4,1:4} \tag{7.1}$$

avec $\mathbf{N}$ une matrice 5×5 :

$$\mathbf{N} = \begin{pmatrix} \mathbf{A}^\intercal \left(\mathbf{B} \begin{pmatrix} \mathbf{\Sigma}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{\Sigma}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{\Sigma}_K \end{pmatrix} \mathbf{B}^T \right)^{-1} & \mathbf{A} & \mathbf{X} \\ \mathbf{X}^T & & 0 \end{pmatrix}. \tag{7.2}$$

En géométrie Euclidienne, la matrice de covariance cherchée $\mathbf{\Sigma}$ s'obtient avec $\mathbf{\Sigma}_{\mathbf{XX}}$ et le Jacobien de la division de la partie euclidienne de $\mathbf{X}$ par sa partie homogène. Même en considérant les distributions des mesures $\mathbf{x}_k$ identiques et isotropes, le calcul de $\mathbf{\Sigma}_{\mathbf{XX}}$ nécessite l'inversion d'une matrice $2K \times 3K$ ($\mathbf{B}\mathbf{B}^\intercal$ dans ce cas). L'expression de cette matrice étant beaucoup trop compliquée, nous suggérons l'utilisation de la matrice de covariance d'un point 3D triangulé telle qu'elle a été développée dans le chapitre 3. Une alternative à l'expression purement analytique est d'estimer la matrice de covariance par une méthode de Monte Carlo (Rumpler *et al.*, 2011). Toujours dans le but d'estimer le placement de caméras qui optimise la reconstruction 3D (en l'occurence un nuage de points), Olague et Mohr (2002) maximisent l'élément diagonal le plus grand de la matrice de covariance du point $\mathbf{X}$. Ils triangulent le point 3D à l'aide d'une DLT (*Direct Linear Transform*) et se servent du système d'équations linéaire pour propager l'incertitude du plan capteur au point $\mathbf{X}$. L'optimisation est effectuée sur tout le nuage de points à l'aide d'un algorithme génétique. Cependant, il nous importe peu de chercher à minimiser l'incertitude sur les points 3D reconstruits, car une reconstruction 3D précise ne signifie pas un rendu de bonne qualité. La section suivante illustre cette assertion.

7.2.2 Aperçu de l'approche

Notre approche est similaire à celle de Olague et Mohr (2002), à la différence qu'elle ne cherche pas à optimiser la reconstruction du *proxy géométrique*, mais le rendu d'une nouvelle vue de paramètres connus $\mathbf{K}_u$, $\mathbf{R}_u$, $\mathbf{t}_u$ et $\mathbf{C}_u$ à partir d'un ensemble de K vues de paramètres à optimiser $\mathbf{K}_k$, $\mathbf{R}_k$, $\mathbf{t}_k$ et $\mathbf{C}_k$ pour $k \in [1 \dots K]$. Pour chaque point image $\mathbf{x}$ de chaque vue source k, nous cherchons à minimiser une mesure d'incertitude de sa reprojection $\mathbf{x}'$ sur la vue cible. La mesure du point image $\mathbf{x}$ n'est pas exacte : nous supposons qu'elle est distribuée selon une loi normale centrée sur $\mathbf{x}$ et de covariance $\mathbf{\Sigma_{xx}}$. Nous définissons notre mesure d'incertitude comme le plus grand élément diagonal de la matrice de covariance de la distribution $\mathbf{\Sigma_{x'x'}}$ de la projection de $\mathbf{x}$ en $\mathbf{x}'$ sur la vue cible, que l'on obtient par propagation d'erreur. En linéarisant localement la fonction de déformation autour du point $\mathbf{x}$, on peut considérer que la distribution du point projeté sur la vue cible suit aussi une loi normale. Nous considérons par la suite que $\mathbf{x}$ et $\mathbf{x}'$ sont des signaux aléatoires. La suite de cette section détaille le calcul de $\mathbf{\Sigma_{x'x'}}$ en fonction de $\mathbf{\Sigma_{xx}}$. Dans le but d'être la plus générique possible, notre approche ne prend pas en compte le calcul de $\mathbf{\Sigma_{xx}}$ comme le fait Olague et Mohr (2002) car cela dépend du dispositif de capture ; nous nous le donnons pour acquis.

Nous définissons la fonction de coût à optimiser comme la somme des mesures d'incertitude issues des vues sources. Des travaux futurs consisteraient à chercher les minima de cette fonction de coût pour des caméras contraintes à un axe situé à une distance donnée de la scène, puis pour des configurations plus originales (dans un plan ou en arc de cercle par exemple). Les vues sources, que l'on prend rectifiées, devront avoir au moins la totalité de la scène dans leur champ. Nous supposons connu le *proxy géométrique* de la scène qui sert au calcul des mesures d'incertitude, via les correspondances qu'il procure entre les points des vues sources $\mathbf{x}$ et les points de la vue cible $\mathbf{x}'$. Ce *proxy* consiste par exemple en un nuage de points triangulés, dont chaque point 3D $\mathbf{X}$ doit être vu par au moins deux caméras pour être reconstruit. La recherche du minimum de la fonction de coût pourrait s'effectuer grâce à un algorithme d'optimisation classique (Gauss-Newton, Levenberg-Marquardt) ou un algorithme génétique (Olague et Mohr, 2002).

7.2.3 Propagation de l'incertitude de mesure du point image pour une géométrie donnée

Considérons le signal aléatoire $\mathbf{x}$, un point image observant un élément de surface de la scène visible par au moins une autre caméra (sinon il est impossible d'estimer la profondeur de cet élément de surface par triangulation). Cet élément de surface est caractérisé par un point 3D $\mathbf{X}$ et sa normale $\mathbf{n}$ (figure 7.1). Dans un premier temps nous ne prenons pas en compte l'incertitude de reconstruction du *proxy géométrique* : $\mathbf{X}$ est connu, constant et n'est pour le moment pas considéré comme un signal aléatoire comme ce sera le cas dans la suite. Au point image $\mathbf{x}$ nous associons le rayon 3D $\mathbf{r} = \mathbf{R}_k^\intercal \mathbf{K}_k^{-1} \bar{\mathbf{x}}$ qui passe par ce point et le centre optique de la caméra. En reprenant le formalisme du chapitre 3, nous notons $\mathbf{X}'$ le point d'intersection du rayon optique avec l'élément de surface, qui n'est pas $\mathbf{X}$ car, à cause de l'erreur de

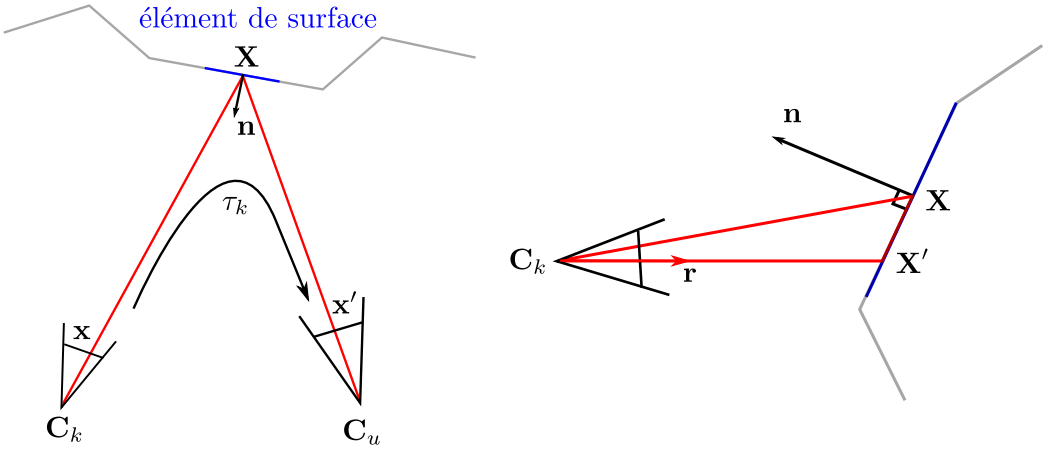


Fig. 7.1 – *Notre fonction de coût à minimiser pour trouver le placement idéal est la somme des mesures d'incertitude de projection des points image* $\mathbf{x}$ *des vues sources* $\mathbf{C}_k$ *sur la vue cible* $\mathbf{C}_u$ *en les points* $\mathbf{x}'$*. À gauche : supposant reconstruite une géométrie approximative de la scène, nous calculons cette mesure d'incertitude par propagation d'erreur du point* $\mathbf{x}$ *au point* $\mathbf{x}'$*, via un élément de surface constitué d'un point 3D* $\mathbf{X}$ *et de sa normale* $\mathbf{n}$*. À droite : le point image* $\mathbf{x}$ *qui est associé au rayon* $\mathbf{r}$ *passant par le centre de projection de la vue source* $\mathbf{C}_k$ *n'est pas la projection exacte du point 3D* $\mathbf{X}$*. Le rayon* $\mathbf{r}$ *intersecte l'élément de surface au point 3D* $\mathbf{X}'$ *situé sur la surface élémentaire.*

mesure, les rayons optiques ne s'intersectent pas forcément en un même point 3D (figure 7.1). Le point image $\mathbf{x}'$ est obtenu par projection de $\mathbf{X}'$ sur la vue cible et normalisation. La matrice de covariance de $\mathbf{x}'$ s'exprime en fonction de celle de $\mathbf{x}$ par l'intermédiaire de la jacobienne de la transformation $\frac{\partial \mathbf{x}'}{\partial \mathbf{x}}$, selon la formule de propagation d'incertitude :

$$\mathbf{\Sigma}_{\mathbf{x}'\mathbf{x}'} = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}} \mathbf{\Sigma}_{\mathbf{x}\mathbf{x}} \frac{\partial \mathbf{x}'^{\intercal}}{\partial \mathbf{x}} \,. \tag{7.3}$$

Le jacobien s'obtient par dérivation en chaîne selon le schéma de propagation suivant :

$$\mathbf{x} \underset{(a)}{\longrightarrow} \mathbf{r} \underset{(b)}{\longrightarrow} \mathbf{X}' \underset{(c)}{\longrightarrow} \mathbf{x}'. \tag{7.4}$$

Avant de donner l'expression des dérivées intermédiaires, il convient de développer le calcul du point 3D $\mathbf{X}'$. Nous exploitons le fait qu'il se situe sur le rayon optique $\mathbf{r}$ à une profondeur orthogonale z :

$$\mathbf{X}' = \mathbf{C}_k + z.\mathbf{R}_k^{\intercal}\mathbf{K}_k^{-1}\bar{\mathbf{x}} = \mathbf{C}_k + z.\mathbf{r}. \tag{7.5}$$

D'autre part sachant qu'il se situe également sur l'élément de surface centré sur $\mathbf{X}$ de normale $\mathbf{n}$, nous avons l'équation

$$\mathbf{n}^{\intercal}(\mathbf{X}' - \mathbf{X}) = 0. \tag{7.6}$$

Cela nous permet de trouver l'expression de la profondeur du point $\mathbf{X}'$:

$$z = \frac{\mathbf{n}^\intercal(\mathbf{X} - \mathbf{C}_k)}{\mathbf{n}^\intercal \mathbf{r}}. \tag{7.7}$$

On a donc l'expression de $\mathbf{X}'$ en fonction du centre de la caméra $\mathbf{C}$, de la normale $\mathbf{n}$ au point $\mathbf{X}$ et du rayon $\mathbf{r}$:

$$\mathbf{X}' = \mathbf{C}_k + \frac{\mathbf{n}^\intercal(\mathbf{X} - \mathbf{C}_k)}{\mathbf{n}^\intercal \mathbf{r}}.\mathbf{r}. \tag{7.8}$$

Maintenant nous avons toutes les clés pour calculer les dérivées chaînées :

$$\text{(c)} \ \frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = \mathbf{J}_e(\tilde{\mathbf{x}}')\mathbf{K}_u\mathbf{R}_u\frac{\partial \mathbf{X}'}{\partial \mathbf{x}}, \tag{7.9}$$

$$\text{(b)} \ \frac{\partial \mathbf{X}'}{\partial \mathbf{x}} = \frac{\mathbf{n}^\intercal(\mathbf{X} - \mathbf{C}_k)}{\mathbf{n}^\intercal \mathbf{r}}\left(\mathbf{I}_3 - \frac{\mathbf{r}\mathbf{n}^\intercal}{\mathbf{n}^\intercal \mathbf{r}}\right)\frac{\partial \mathbf{r}}{\partial \mathbf{x}}, \tag{7.10}$$

$$\text{(a)} \ \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \mathbf{R}_k^\intercal \mathbf{K}_k^{-1}\mathbf{J}_h. \tag{7.11}$$

Nous rappelons que $\mathbf{J}_e(\mathbf{x}')$ désigne la jacobienne de la normalisation en coordonnées euclidiennes et $\mathbf{J}_h$ la jacobienne de l'ajout d'une troisième coordonnée homogène. Finalement, le jacobien de la fonction de déformation s'écrit

$$\frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = \frac{\mathbf{n}^\intercal(\mathbf{X} - \mathbf{C}_k)}{\mathbf{n}^\intercal \mathbf{r}}.\mathbf{J}_e(\tilde{\mathbf{x}}')\mathbf{K}_u\mathbf{R}_u\left(\mathbf{I}_3 - \frac{\mathbf{r}\mathbf{n}^\intercal}{\mathbf{n}^\intercal \mathbf{r}}\right)\mathbf{R}_k^\intercal \mathbf{K}_k^{-1}\mathbf{J}_h. \tag{7.12}$$

7.2.4 Propagation de l'incertitude sur la géométrie pour une mesure donnée

Nous avons vu comment propager l'incertitude de mesure sur la vue cible pour une géométrie donnée ; c'est ignorer que la précision de la reconstruction dépend elle aussi des paramètres des vues sources. En effet comme nous l'avons vu au chapitre 3, il est possible d'exprimer l'incertitude sur le point 3D $\mathbf{X}$ reconstruit par triangulation. Le point 3D peut également être considéré comme une variable aléatoire qui suit une loi normale d'espérance $\bar{\mathbf{X}}$ (position mesurée dans l'espace, supposée connue) et de matrice de covariance 3D $\Sigma_{\mathbf{XX}}$. Nous invitons le lecteur à se reporter à ce chapitre pour l'expression de $\Sigma_{\mathbf{XX}}$ en fonction du point 3D et des paramètres des vues sources. Une extension de notre démarche consisterait à considérer la normale $\mathbf{n}$ comme une variable aléatoire, ce que nous ne faisons pas ici.

Nous avons donc deux signaux aléatoires intrinsèquement liés : l'un provient du point image $\mathbf{x}$ et l'autre du point 3D $\mathbf{X}$. Si l'on suppose que le signal issu de $\mathbf{x}$ ne dépend pas de $\mathbf{X}$, alors le signal du point $\mathbf{x}'$ dans la vue cible s'exprime comme la convolution des deux signaux. Cela revient à remplacer $\mathbf{X}$ par $\bar{\mathbf{X}}$ dans les équations précédentes. Les covariances de deux signaux convolués s'additionnant, il nous reste à calculer la matrice de covariance issus de la propagation de l'incertitude sur le

point 3D $\mathbf{X}$ pour un rayon $\mathbf{r}$ donné. On l'obtient grâce à la formule

$$\mathbf{\Sigma}_{\mathbf{x}'\mathbf{x}'} = \frac{\partial \mathbf{x}'}{\partial \mathbf{X}} \mathbf{\Sigma}_{\mathbf{X}\mathbf{X}} \frac{\partial \mathbf{x}'^{\top}}{\partial \mathbf{X}}. \tag{7.13}$$

De la même manière que précédemment, le jacobien $\frac{\partial \mathbf{x}'}{\partial \mathbf{X}}$ s'obtient par dérivation chaînée, selon le schéma de propagation suivant :

$$\mathbf{X} \underset{(a)}{\longrightarrow} \mathbf{X}' \underset{(b)}{\longrightarrow} \mathbf{x}'. \tag{7.14}$$

(b) est identique à (c) dans la section précédente :

$$\text{(b)} \quad \frac{\partial \mathbf{x}'}{\partial \mathbf{X}} = \mathbf{J}_e(\tilde{\mathbf{x}}')\mathbf{K}_u\mathbf{R}_u \frac{\partial \mathbf{X}'}{\partial \mathbf{X}}. \tag{7.15}$$

En reprenant l'équation (7.8), nous exprimons la première dérivée

$$\text{(a)} \quad \frac{\partial \mathbf{X}'}{\partial \mathbf{X}} = \frac{\mathbf{r}\mathbf{n}^{\top}}{\mathbf{n}^{\top}\mathbf{r}}. \tag{7.16}$$

(a) et (b) mis bout-à-bout, on a

$$\frac{\partial \mathbf{x}'}{\partial \mathbf{X}} = \mathbf{J}_e(\tilde{\mathbf{x}}')\mathbf{K}_u\mathbf{R}_u \frac{\mathbf{r}\mathbf{n}^{\top}}{\mathbf{n}^{\top}\mathbf{r}}. \tag{7.17}$$

7.2.5 Propagation de l'incertitude de mesure du point image pour une géométrie aléatoire

Nous avons donc deux matrices de covariances données par les formules de propagation 7.3 et 7.13, que nous appelons respectivement $\mathbf{\Sigma}_{res}$ et $\mathbf{\Sigma}_{geo}$ (figure 7.2). Par convolution des signaux correspondants, nous pouvons les additionner pour trouver la matrice de covariance associée à l'incertitude sur le point $\mathbf{x}'$ projeté :

$$\mathbf{\Sigma}_{\mathbf{x}'\mathbf{x}'} = \mathbf{\Sigma}_{res} + \mathbf{\Sigma}_{geo} = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}} \mathbf{\Sigma}_{\mathbf{x}\mathbf{x}} \frac{\partial \mathbf{x}'^{\top}}{\partial \mathbf{x}} + \frac{\partial \mathbf{x}'}{\partial \mathbf{X}} \mathbf{\Sigma}_{\mathbf{X}\mathbf{X}} \frac{\partial \mathbf{x}'^{\top}}{\partial \mathbf{X}}. \tag{7.18}$$

Ces deux termes rappellent les poids de rendu utilisés par Pujades *et al.* (2014) :

- Le terme de résolution $\mathbf{\Sigma}_{res}$ pénalise les courtes focales car elle restituent moins de détails de l'objet filmé. En effet plus la distance focale de la caméra est courte, plus le « cône d'incertitude » projeté sera large. Le terme de résolution pénalise aussi les caméras qui regardent la surface de biais, ou qui sont éloignées de la scène, car la projection de l'incertitude sur la surface est grande. Cette covariance est en général de rang plein.
- Le terme de géométrie $\mathbf{\Sigma}_{geo}$ est proportionnel au produit scalaire entre le rayon incident et la normale à la surface, et pénalise donc les caméras éloignées de la vue cible. Cependant c'est aussi une fonction croissante de l'incertitude sur la géométrie, qui est faible lorsque les caméras de prise de vue sont bien écartées

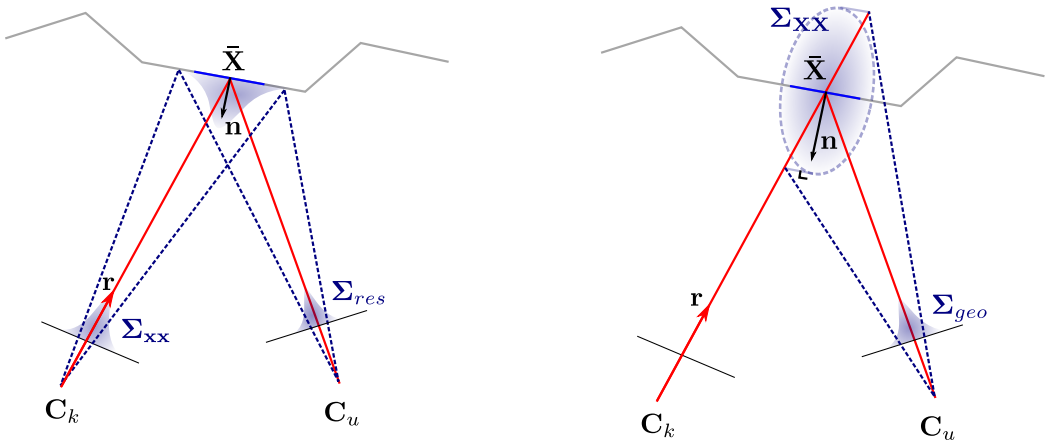


Fig. 7.2 – *Illustration du terme de résolution $\boldsymbol{\Sigma}_{res}$ (à gauche) et du terme de géométrie $\boldsymbol{\Sigma}_{geo}$ (à droite). $\boldsymbol{\Sigma}_{res}$ est obtenu par propagation de l'incertitude de mesure $\boldsymbol{\Sigma}_{\mathbf{xx}}$ du point $\mathbf{x}$ associé au rayon $\mathbf{r}$, pour une géométrie donnée $(\bar{\mathbf{X}}, \mathbf{n})$. Nous avons aussi représenté l'incertitude intermédiaire projetée sur l'élément de surface. $\boldsymbol{\Sigma}_{geo}$ est obtenu par propagation de l'incertitude $\boldsymbol{\Sigma}_{\mathbf{XX}}$ sur le point 3D triangulé $\mathbf{X}$, pour une mesure 3D donnée c'est-à-dire un rayon $\mathbf{r}$ fixe. Cette incertitude est d'abord projetée sur le rayon incident $\mathbf{r}$ à la surface perpendiculairement à la normale $\mathbf{n}$, puis projetée sur la vue cible.*

(Beder et Steffen, 2006). La minimisation de ce terme revient à trouver un compromis entre une bonne reconstruction de la géométrie et une faible reprojection de l'incertitude. Cette covariance est en général de rang 1 car elle est d'abord projetée sur le sous-espace vectoriel de dimension 1 défini par le rayon indicent $\mathbf{r}$, avant d'être projetée sur la vue cible.

Intuitivement, si les caméras sont contraintes à se déplacer sur un axe et que tous les autres paramètres excepté l'entraxe sont fixes, on imagine qu'il existe un minimum global à la fonction de coût. En effet $\boldsymbol{\Sigma}_{res}$ augmente avec l'entraxe (car la surface est vue de plus en plus de biais), mais en même temps une augmentation de l'entraxe permet une triangulation plus précise, et donc diminue l'incertitude $\boldsymbol{\Sigma}_{geo}$. Cette intuition est confirmée par les simulations dans la section suivante.

7.2.6 Simulation pour un point 3D vu par deux caméras sources

Nous avons cherché à exprimer de manière formelle la matrice de covariance $\boldsymbol{\Sigma}_{\mathbf{x'x'}}$ pour une scène réduite à un élément de surface, visible dans deux vues sources. Cette matrice est la somme de deux matrices de covariance, $\boldsymbol{\Sigma}_{\mathbf{x'x'},1}$ et $\boldsymbol{\Sigma}_{\mathbf{x'x'},2}$ provenant respectivement des vues 1 et 2. Nous nous plaçons dans un espace plan (X, Y) ; les matrices de covariances sont les variances scalaires $\sigma^2_{\mathbf{x'},1}$ et $\sigma^2_{\mathbf{x'},2}$. Les positions des caméras sources sont contraintes sur un axe, paramétrées par leur entraxe. La vue cible est positionnée sur l'axe médian entre les deux vues sources, à une distance du point 3D dix fois moindre que les caméras sources. Les distances focales sont identiques et constantes, il en est de même pour les autres paramètres intrinsèques. Toutes les caméras sont orientées dans la même direction, seul leur centre optique diffère.

En 2D, la covariance (scalaire) est une fraction rationnelle de l'entraxe. L'entraxe

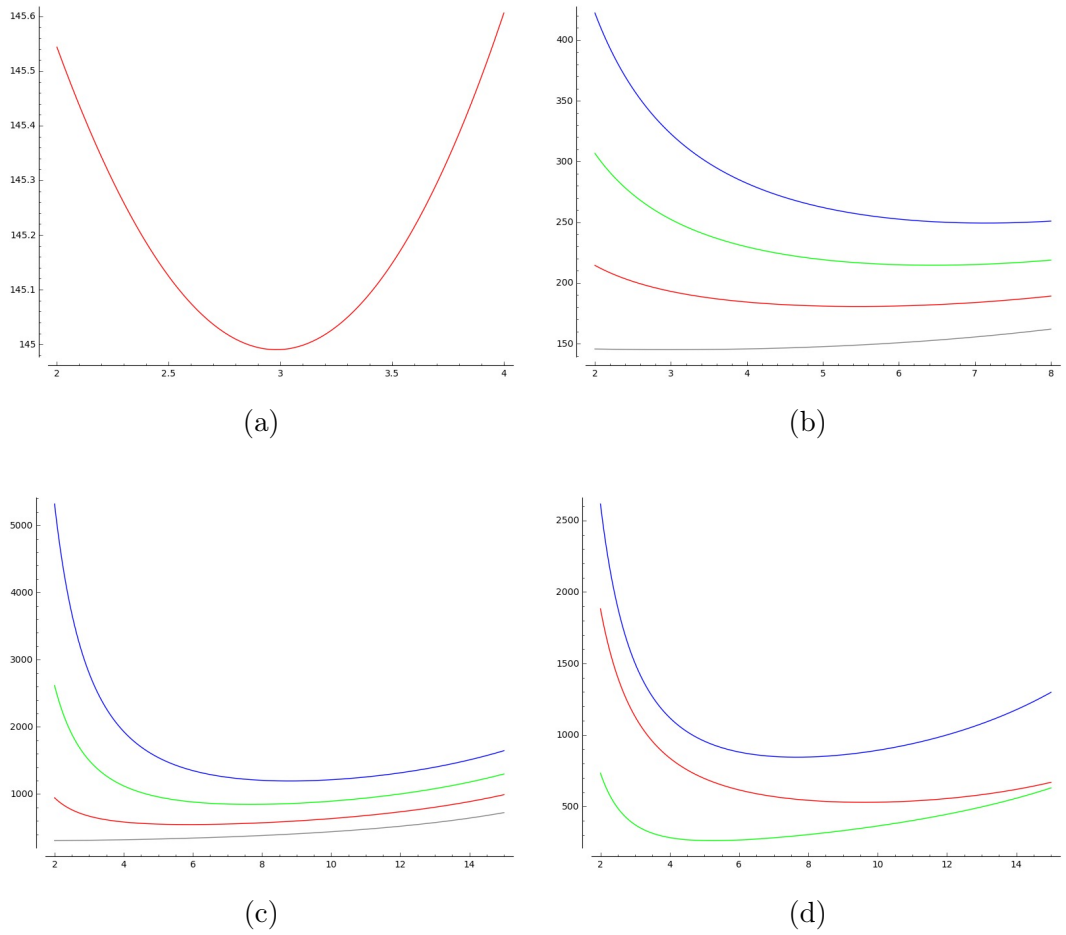


Fig. 7.3 – *Évolution de la fonction de coût en fonction de l'entraxe entre deux caméras sources observant un point 3D* $\mathbf{X}$. *(a) Variance (scalaire)* $\sigma^2_{\mathbf{x}',1}$ *issue de la vue source 1. (b) Cette même variance pour différentes positions de* $\mathbf{X}$, *de « centré entre les deux vues sources » (gris) à « très excentré » (bleu). (c) Même chose avec la fonction de coût globale, somme des variances des vues sources* $\sigma^2_{\mathbf{x}',1}$ *et* $\sigma^2_{\mathbf{x}',2}$. *(d) Comparaison de la variance* $\sigma^2_{\mathbf{x}',1}$ *issue de la vue 1 (en rouge) avec la variance* $\sigma^2_{\mathbf{x}',2}$ *issue de la vue 2 (en vert). La fonction de coût (en bleu) est la somme des deux.*

étant strictement positif, il y a deux pôles :

- 0 : les deux vues ont la même position, la variance du point 3D triangulé est donc infinie, tout comme sa reprojection $\mathbf{\Sigma}_{geo}$ sur la vue cible.
- un pôle qui dépend de l'inclinaison de l'élément de surface, qui correspond à l'endroit où l'une des vues sources voit la surface de biais ; $\mathbf{\Sigma}_{res}$ est alors infinie.

Si le point 3D se situe exactement sur l'axe médian entre les deux caméras sources, alors il n'y a pas de pôle 0, seulement le pôle qui correspond à $\mathbf{\Sigma}_{res}$ infinie.

Les figures 7.3 et 7.4 décrivent l'évolution de la fonction de coût en fonction de l'entraxe entre les deux vues sources. Nous voyons (figure 7.3a) que pour une vue source donnée, la variance a un minimum, ici pour un entraxe de 3. Ce minimum (figure 7.3b) sera d'autant plus grand que le point 3D est excentré (par rapport à l'axe médian entre les deux vues sources). Ce minimum n'est pas le même pour l'autre vue (figure 7.3d) (car le point 3D est excentré et l'élément de surface n'est pas

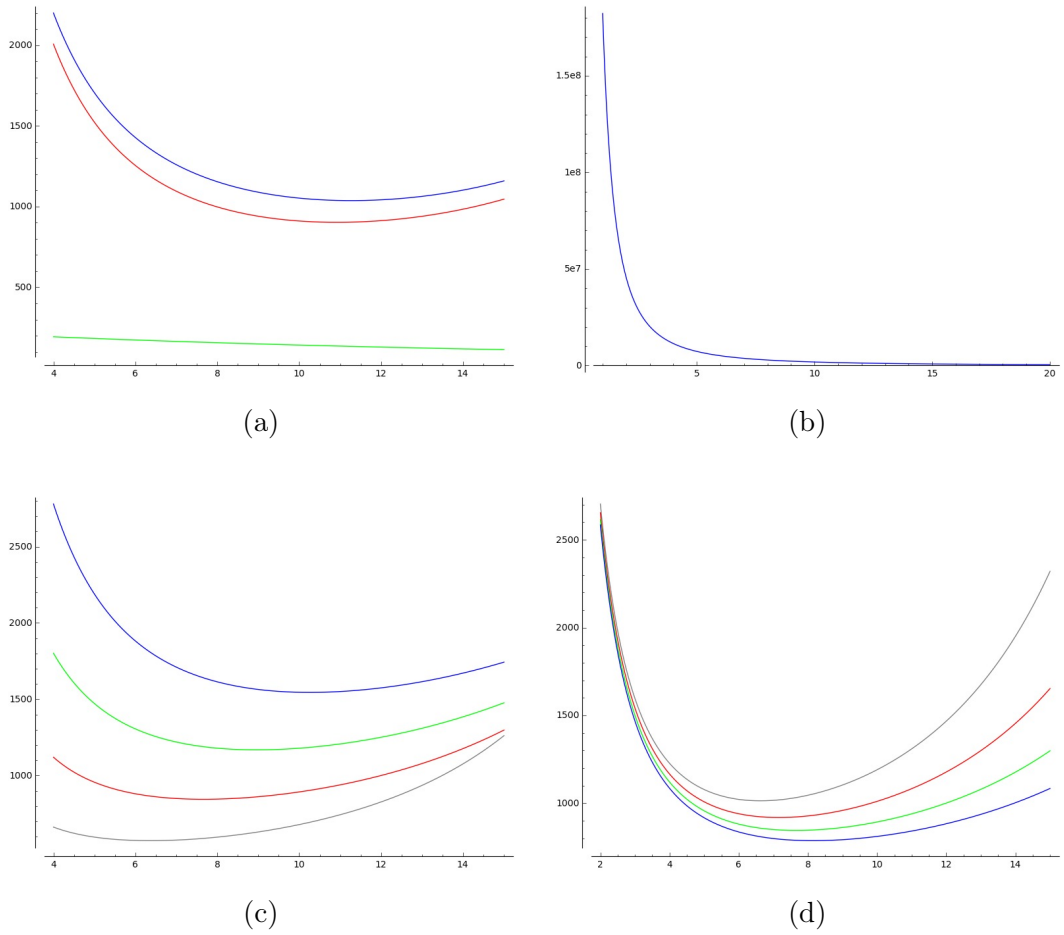


Fig. 7.4 – *Évolution de la fonction de coût en fonction de l'entraxe entre deux caméras sources observant un point 3D* $\mathbf{X}$. *(a) Fonction de coût globale (en bleu), décomposée en la somme des termes de résolution (en rouge) et la somme des termes de géométrie (en vert). (b) Déterminant de la matrice de covariance* $\mathbf{\Sigma_{XX}}$. *(c) En faisant varier la distance des caméras sources au point 3D, de faible (gris) à grande (en bleu). (d) En faisant varier l'orientation de l'élément de surface, de fronto-parallèle aux vues sources (en bleu) à oblique (en gris).*

fronto-parallèle, donc la scène n'est pas symétrique) mais il existe, donc les mêmes conclusions sont à tirer de la fonction de coût (figure 7.3c). Si l'on décompose la fonction de coût en la somme des termes de résolution et des termes de géométrie (figure 7.4a), on remarque que la somme des termes de géométrie est une fonction décroissante de l'entraxe, contrairement à la somme des termes de résolution qui atteint un minimum (qui dépend de l'orientation de la surface et de la distance des vues sources à la scène, d'après les figures 7.4c-d). Cela est dû au fait que la covariance (ou plus rigoureusement son déterminant) du point 3D triangulé est une fonction strictement décroissante de l'entraxe (figure 7.4b).

7.3 Se rapprocher virtuellement : rendu d'objets lointains

7.3.1 Motivations

Une des applications principales des algorithmes de rendu basé image décrits précédemment est la vision à longue distance. Alors que la plupart des algorithmes de rendu se contentent souvent de démontrer leur performance par la synthèse d'un point de vue *à côté* des vues sources, nous proposons de générer un point de vue *devant* celles-ci. Pour une scène constituée d'un *sujet* et d'un *arrière-plan*, le problème de vision à longue distance consiste à générer un point de vue virtuellement plus proche du sujet que les vues sources utilisée pour sa synthèse. Il s'agit d'un problème qui est loin d'être trivial, car même si les vues sources peuvent partager quelques rayons avec la vue cible, la plupart sont à reconstruire, non seulement leur géométrie (position, direction), mais aussi leur radiance. La vision à longue distance rencontre donc les mêmes difficultés que n'importe quel synthèse de point de vue, en particulier la nécessité de bien modéliser la scène, soit par la reconstruction d'un *proxy géométrique* précis, soit par la bonne modélisation de l'espace plénoptique, par exemple dans le cas de présence de surfaces réfractives ou transparentes. La figure 7.5 illustre les artefacts causés par le rendu d'une vue plus proche d'un sujet dont la qualité du modèle géométrique est insuffisante. Il n'est pas nécessaire de bien modéliser la scène si les rayons reconstruits passent par le centre optique d'une caméra source, et que l'algorithme de rendu possède la propriété de *cohérence épipolaire* (Buehler *et al.*, 2001). Mais si aucune de ces deux conditions n'est vérifiée, alors le rendu est grandement affecté.

Synthétiser un point de vue plus proche du sujet n'est pas équivalent à zoomer sur ce sujet, comme l'illustre la figure 7.6. Dans les deux cas la portion du sujet vue à l'image est identique, mais l'arrière-plan n'a pas la même taille. Pour zoomer nous avons simplement augmenté la distance focale d'une des vues sources, ce qui ne fait que réduire le champ de vue, augmentant ainsi la taille du sujet mais aussi l'arrière-plan par la même occasion. La figure de gauche est le scénario souhaité, où l'arrière-plan reste en proportion raisonnable, tel qu'il est vu ou presque dans les images sources. Le zoom peut occasionner de nombreux effets indésirables dans la production de contenu stéréoscopique 3D, comme la perte de relief de l'image (la *rondeur*) ou la divergence occulaire (Pujades et Devernay, 2013). Dans ce contexte on préfèrera donc générer une paire de vues stéréoscopique plus proche du sujet au lieu d'augmenter la distance focale.

La différence entre le zoom et la synthèse de point de vue se fait principalement en termes de rayons reconstruits. Sur la figure 7.6 nous avons représenté trois rayons optiques $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ qu'il faut reconstruire pour générer la vue cible. Dans le cas de la synthèse de point de vue, seul $\mathbf{r}_2$ est déjà échantillonné par les caméras sources (la vue $\mathbf{C}_2$), les autres doivent être reconstruits à partir de notre modèle de la scène et d'un algorithme de rendu basé image. Si le modèle de *point visuel* est trop imprécis ($3\mathrm{g}+3\mathrm{p}$ pour la boule de *tarot*) le rayon $\mathbf{r}_1$ est mal reconstruit : c'est ce qui explique pourquoi la partie supérieure gauche de la sphère est moins bien reconstruite que la

Fig. 7.5 – *Synthèse d'un point de vue virtuellement plus proche de la sphère du jeu de données* tarot. *Les rayons issus des surfaces diffuses (les cartes de tarot) sont bien reconstruits ; il en est de même pour la partie inférieure droite de la sphère, dont les rayons à reconstruire sont identiques à ceux échantillonnés par les vues sources. Mais la partie supérieure gauche n'est pas bien modélisée du fait des réfractions, ce qui crée des artefacts de rendu. En effet nous avons employé* $3\mathrm{g} + 3\mathrm{p}$*, le modèle simple du point 3D triangulé à 3 paramètres.*

partie inférieure droite sur la figure 7.6. Au contraire le modèle lambertien $3\mathrm{g} + 3\mathrm{p}$ suffisant à décrire l'arrière-plan composé de cartes à la surface diffuse, le rayon $\mathbf{r}_3$ est bien reconstruit et aucun artefact n'est visible.

Notons que le zoom et la synthèse de vue ne sont pas incompatibles. Dans ce but nous conseillons de générer une vue ayant une plus grande distance focale à partir des vues sources plutôt que de simplement rogner les images. En effet l'apport d'information des autres points de vue permet de compenser la résolution d'image (en pixel) de la caméra qui devient insuffisante dès lors que l'image du sujet grossit. Cela provient du fait que les rayons $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ sur la figure 7.6 ne sont en général pas exactement ceux qui ont été capturés par la vue source $\mathbf{C}_2$, même s'ils passent tous par son centre optique (qui est le même que celui de $\mathbf{C}_u$) : la résolution d'image peut être différente entre $\mathbf{C}_2$ et $\mathbf{C}_u$, mais surtout leur champ de vue sont différents (car la distance focale de $\mathbf{C}_2$ est inférieure à celle de $\mathbf{C}_u$). Les rayons $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ sont donc *interpolés* à partir des rayons capturés par $\mathbf{C}_2$. Cette interpolation gagne en précision si d'autres vues contribuent (c'est un problème voisin de celui de la super-résolution).

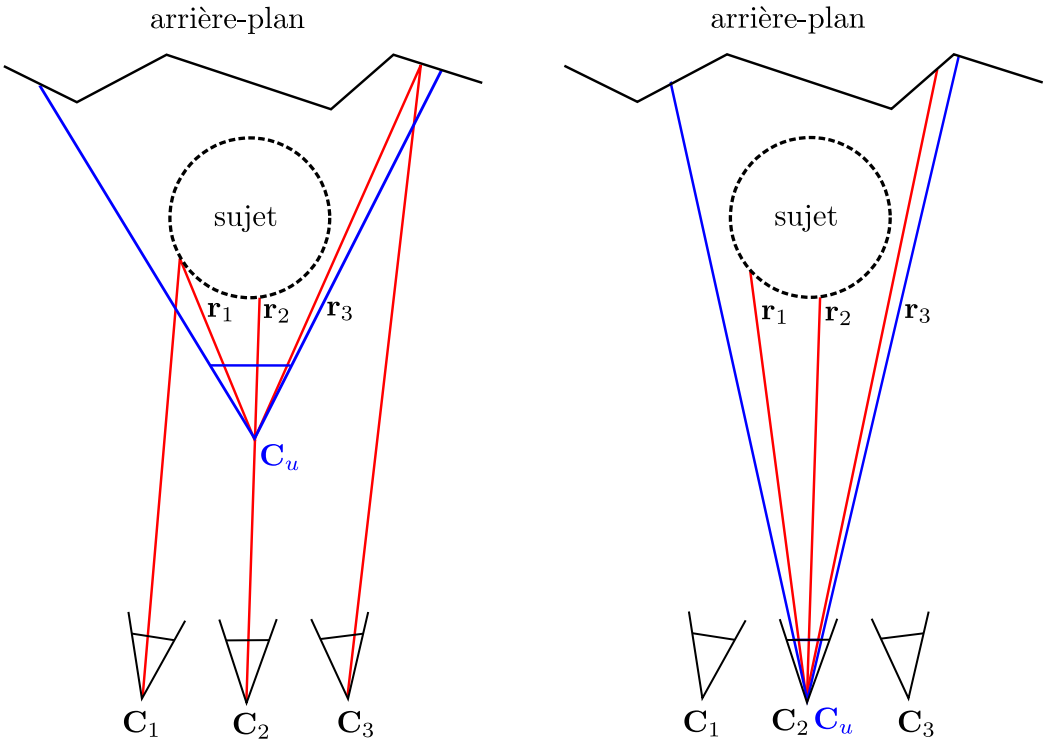


Fig. 7.6 – *Différence entre synthétiser un point de vue $\mathbf{C}_u$ plus proche du sujet (à gauche) à partir des vues sources $\mathbf{C}_1$, $\mathbf{C}_2$ et $\mathbf{C}_3$, et zoomer sur le sujet (à droite). Dans les deux cas de figure la même portion du sujet est vue à l'image ; on peut le remarquer par le champ de vue cible illustré en bleu. Trois rayons optiques $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ à reconstruire ou interpoler ont été représentés en rouge.*

7.3.2 Expérience

Nous avons voulu comparer expérimentalement le rendu d'une vue plus proche du sujet avec le rendu d'une vue statique, le grossissant simplement. Pour cela nous avons utilisé l'algorithme de rendu par éclaboussure après avoir modélisé les *points visuels* de la scène *tarot* comme dans le chapitre 6. Une première série d'images (figure 7.7) est générée pour un point de vue qui s'éloigne progressivement des vues sources pour se rapprocher de la sphère. Une autre série d'images est générée pour un point de vue qui reste identique à la vue source centrale de la grille, exceptée la distance focale que nous avons progressivement augmenté, de telle sorte que la sphère occupe approximativement la même place à l'image. Les deux séries d'images ont été générées avec le même algorithme et le même modèle de *point visuel*, celui du point lambertien (le plus simple). Comme attendu, on constate que lors de la synthèse du point de vue se rapprochant, l'arrière-plan conserve ses dimensions alors que la sphère prend de plus en plus de place à l'image : on donne l'illusion de la parallaxe, et donc de se rapprocher virtuellement du sujet. Au contraire, le zoom simulé est équivalent à rogner les images, les proportions entre les cartes en arrière-plan et la sphère-sujet n'évoluant pas d'image en image.

La synthèse de point de vue a ses limites (figure 7.8). Nous avons poussé l'expérience jusqu'à se rapprocher suffisamment pour que les premiers artefacts de rendu

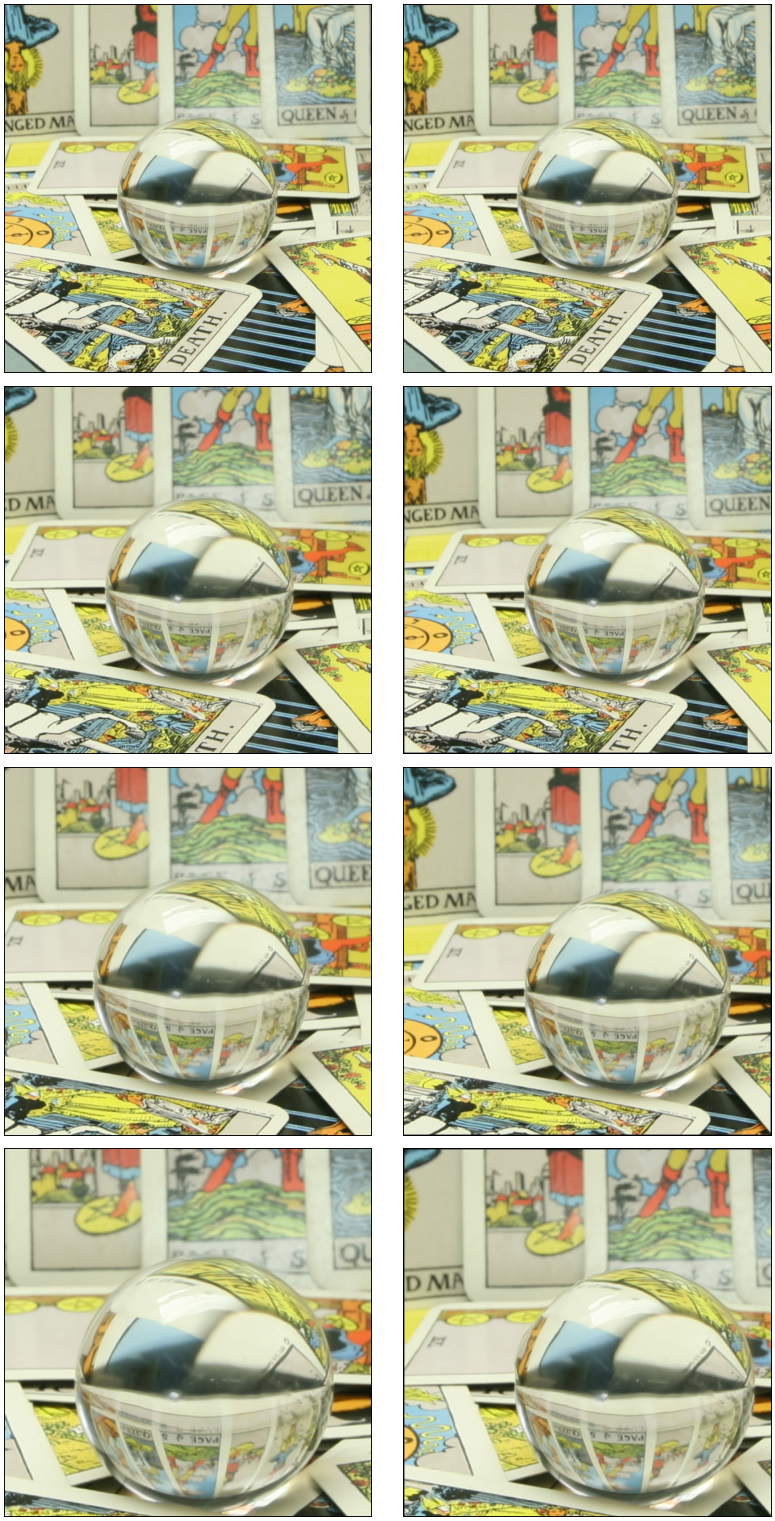


Fig. 7.7 – *Comparaison entre la synthèse d'un point de vue réduisant son champ de vue (première colonne) et celle d'un point de vue se rapprochant du sujet.*

Fig. 7.8 – *Artefacts de rendu. Nous comparons la synthèse d'un point de vue réduisant son champ de vue (à gauche) et celle d'un point de vue se rapprochant du sujet (à droite).*

apparaissent. Il s'agit d'artefacts liés à la modélisation de la scène, et non à la méthode de rendu elle-même, car le rendu du point de vue statique est quasi-parfait. Le modèle du point lambertien est insuffisant pour reconstruire les rayons issus du bord supérieur gauche de la sphère transparente. Il est donc nécessaire de choisir un modèle de *point visuel* dont la complexité est adaptée à celle de la surface observée, comme il a été montré dans le chapitre 6.

7.4 Voir l'invisible : rendu d'objets camouflés

Imaginons que nous souhaitons voir un objet partiellement caché derrière un camouflage. Si nous prenons un cliché de cet objet, des parties seront visibles, d'autres seront occultées par le camouflage. La fraction des rayons émanant de l'objet occulté qui rencontrent le camouflage sont perdus, mais peuvent être interpolés via les algorithmes présentés dans cette thèse si l'échantillonnage du champ de lumière est suffisamment dense. Ainsi le problème de vision d'objets camouflés n'est qu'un sous-problème de la reconstruction de champ de lumière, avec la difficulté supplémentaire que la mise en correspondance d'une image à l'autre est plus difficile car les images sont « polluées » par le camouflage. D'autre part le champ de lumière échantillonné est beaucoup plus épars car de nombreux rayons sont bloqués par le camouflage.

Un peu de poussière sur un objectif ne gâche pas une photographie : au pire le sujet sera un peu flou. Gu *et al.* (2009) proposent un modèle physique de formation de l'image perturbée par la présence d'obstacles, dans le but de corriger les artefacts dus à la saleté sur l'objectif, ou des obstacles fins qui prennent une toute petite place à l'image. Cela est possible sous l'hypothèse que l'objet occultant ou camouflage prend une place négligeable comparée à la taille de l'objectif ; autrement dit si une large partie des rayons émanant de l'objet (partiellement occulté) sont

captés par l'appareil (et non bloqués par l'objet occultant). De la même manière un ensemble de vues (structuré ou non), ou un réseaux de micro-lentilles dans une caméra plénoptique, forment ce que l'on appelle une *ouverture synthétique*, simulant virtuellement l'objectif d'une caméra. C'est le nombre de vues et leur entraxe qui déterminent la taille de l'ouverture synthétique, et par ailleurs la profondeur de champ. Ainsi, plus le nombre de points de vue différents est élevé, plus la fraction du nombre de rayons occultés sur le nombre de rayons capturés est faible ; il est alors possible de reconstruire une image satisfaisante de l'arrière-plan.

7.4.1 État de l'art

Refocaliser derrière un camouflage La vision d'objets camouflés à l'aide d'une grille de caméras trouve son fondement dans les travaux de Vaish *et al.* (2004, 2006). La grille de caméras est d'abord calibrée à l'aide de la parallaxe (Vaish *et al.*, 2004). Le but est de faire la mise au point derrière un buisson pour faire apparaître les étudiants cachés. Aucune reconstruction 3D de la scène occultée n'est faite : le *proxy géométrique* est un plan de refocalisation, parallèle à la grille de caméras, situé au niveau des étudiants. Le même principe est utilisé par Hong et Javidi (2005); Javidi *et al.* (2006); Hwang *et al.* (2007), mais avec un dispositif de visualisation 3D avec projection et réseau de micro-lentilles. Hwang *et al.* (2007) généralise la méthode à n'importe quel point de vue. Quel que soit le dispositif employé, l'objectif est de combiner les images sources entre elles via le plan de refocalisation, ce qui a pour effet de flouter tous les objets qui ne sont pas situés dans un voisinage du plan de refocalisation (dont l'objet occultant), et de faire apparaître avec netteté ceux qui le sont (l'objet occulté). La profondeur de champ (intervalle de profondeur le long duquel les objets sont nets) dépend du nombre de points de vue et de leur entraxe : l'ensemble des vues sources constitue l'*ouverture virtuelle*.

Reconstruire des objets camouflés par balayage de plan La plupart des méthodes classiques de MVS échouent à reconstruire des objets occultés par des camouflages (Furukawa et Hernández, 2015). On utilise alors généralement une méthode par plan de balayage : un plan parallèle à la grille de caméras se déplace le long de l'axe des profondeurs ; il sert de plan de reprojection des images sources et on calcule un coût pour chaque pixel à chaque profondeur ; la profondeur la plus plausible est celle qui a le coût le plus faible. On distingue les méthodes *shape from stereo* et *shape from focus*. *Shape from stereo* sélectionne la profondeur à laquelle l'image refocalisée (rendue avec le plan de refocalisation) a la variance sur toutes les images sources la plus faible ; *shape from focus* sélectionne la profondeur à laquelle l'image refocalisée est la plus nette (la netteté est définie comme la norme au carré du gradient de l'image synthétisée). Le principal problème de ces méthodes *stereo* et *focus* est que la fonction de coût calculée pour chaque pixel de l'image cible est polluée par la couleur des rayons issus de l'objet occultant. Ainsi les rayons issus du camouflage ont autant d'importance que les rayons issus de l'objet occulté, ce qui perturbe la reconstruction. Vaish *et al.* (2006) proposent deux nouvelles fonctions de coût, l'une basée sur la médiane des couleurs, l'autre sur l'entropie, moins sensible à la pollution

de l'objet occultant ; la couleur de rendu est le mode de la distribution. Cependant leur méthode échoue quand même lorsque le camouflage couvre plus de la moitié des images sources : c'est la couleur du camouflage qui prédomine, ce qui fausse la profondeur estimée et la couleur de rendu.

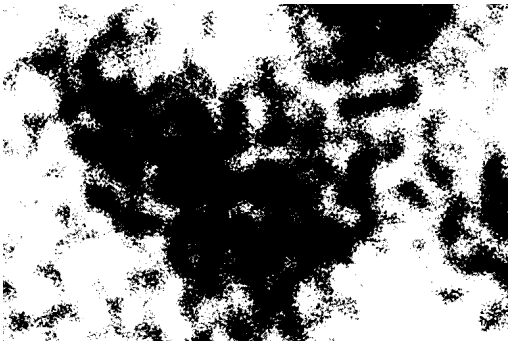
Reconstruire des objets camouflés par MVS Xiao *et al.* (2014) généralise le problème de MVS à des images sources pertubées par des objets occultant. Comme dans la plupart des algorithmes de MVS, la fonction de coût est composée d'un terme d'attache aux données et d'un terme de lissage. Le critère de photo-cohérence du terme d'attache aux données est construit à l'aide d'un *K-mean clustering* qui sélectionne les candidats les plus plausibles pour la mise en correspondance des images. La méthode de résolution du problème est itérative, et procure à chaque itération une estimation de l'image de référence débarrassée du bruit de l'objet occultant. Cette image est ensuite utilisée dans le calcul du terme de lissage pour l'itération suivante. Notez qu'à chaque itération l'algorithme optimise globalement l'estimation courante de la profondeur avec une méthode de *graph-cut*. La méthode de Xiao *et al.* (2014) réalise de bien meilleures performances que celle de Vaish *et al.* (2006) avec l'entropie ou tout autre méthode de stéréo (*shape from focus* ou *shape from stereo* sur les mêmes jeux de données.

7.4.2 Approche proposée

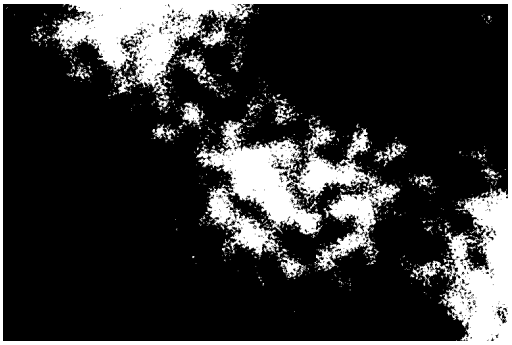
Bien que l'algorithme de Xiao *et al.* (2014) surpasse les algorithme de reconstruction d'objets occultés, il n'est pas exempt de limites : comme les autres sa performance décroit largement lorsque plus de la moitié de chaque image source est occultée. Même en deça, la reconstruction n'est jamais parfaite et les rayons optiques issus de l'objet occultant polluent le rendu de l'arrière-plan. Nous proposons donc de supprimer au préalable les pixels de l'objet occultant des images sources, par une méthode de *matting*, par exemple l'algorithme de Chen *et al.* (2013), ou une méthode multi-vues comme celle de Joshi *et al.* (2006). N'étant pas perturbée par le bruit issu l'objet occultant, la reconstruction de la géométrie de l'arrière-plan (et donc par transitivité le rendu de la vue cible) est améliorée.

Cependant le problème du rendu n'est pas résolu par un masquage des objets occultants dans les images sources (figure 7.10). En effet même avec une bonne reconstruction (c'est-à-dire un bon *proxy géométrique*) l'image de référence contient des trous si l'obstacle (buisson dense par exemple) occulte une trop grande proportion des images sources. Des zones entières de l'image cible peuvent rester inconnues par manque d'information contenue dans les images sources. D'autre par la suppression de pans entiers des images occultées contribue à raréfier les contributions de chaque vue à l'image sources, ce qui exacerbe l'inhomogénéité du rendu final. Alors que Thonat *et al.* (2016) proposent une méthode d'*inpainting* multi-vues, nous suggérons l'emploi de notre méthode directe de rendu multi-résolution pour résoudre simultanément le problème d'inhomogénéité des contributions éparses et du remplissage des zones où il manque de l'information.

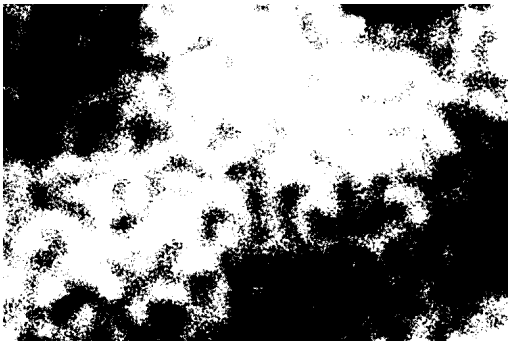
Nous nous attelons spéfcifiquement au problème de rendu, étant donné un *proxy*



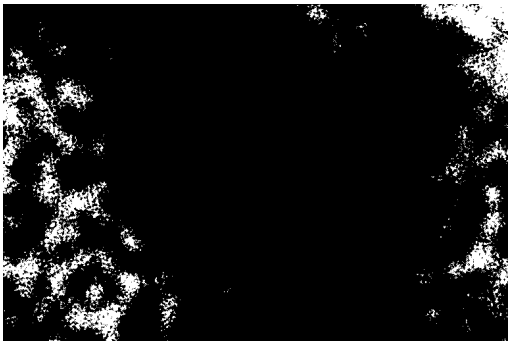
$Z = 0.0$ et $T = 0.50$



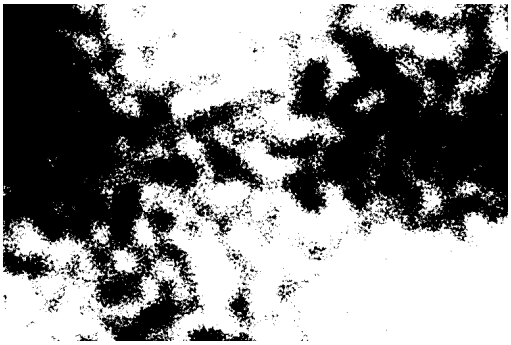
$Z = 56.0$ et $T = 0.60$



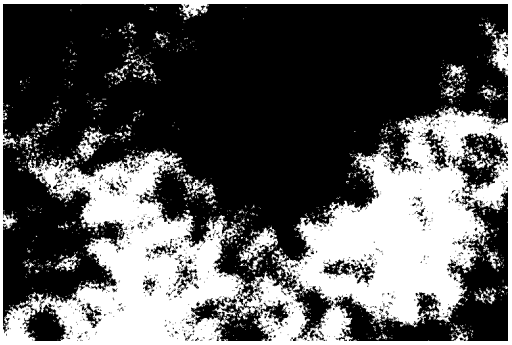
$Z = 1.0$ et $T = 0.50$



$Z = 57.0$ et $T = 0.60$

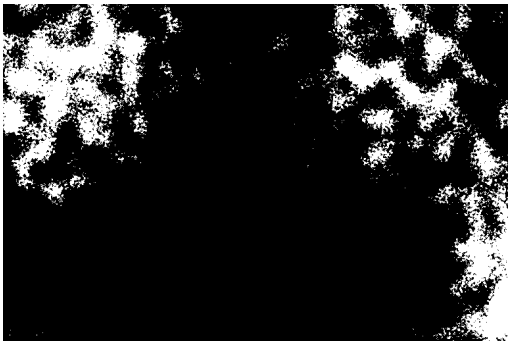


$Z = 2.0$ et $T = 0.50$



$Z = 58.0$ et $T = 0.60$

$Z = 3.0$ et $T = 0.50$



$Z = 59.0$ et $T = 0.60$

Fig. 7.9 – *Quelques-uns des masques utilisés pour simuler un camouflage aléatoire. Chaque masque est défini par une profondeur Z de bruit FBM utilisé pour le générer et un seuil T permettant de contrôler le pourcentage de pixels occultés.*

Fig. 7.10 – *Artefacts de rendu générés par la sparsité des images sources. Des zones entières ne sont pas reconstruites par l'algorithme de rendu car elles sont masquées par le camouflage dans toutes les vues sources. Du fait de la sparsité élevée des contributions, le rendu n'est pas homogène, comme le soulignent les zones cerclées de rouge. Cela nuit au principe de* continuité *énoncé par (Buehler* et al.*, 2001).*

géométrique que l'on suppose estimé de façon itérative comme le proposent Xiao *et al.* (2014) par exemple. Nous supposons que l'occultant a été retiré des images sources grâce au *matting*, en découlent des masques (figure 7.9) qui donnent les contributions de chaque pixel de chaque image source, de 0 (objet occultant, ce pixel ne contribue pas) à 1 (arrière-plan, ce pixel contribue pleinement).

7.4.3 Expériences

Nous avons conduit un certain nombre d'expériences sur les jeux de données *fountain* et *herzjesu*, où l'objectif était de faire le rendu de la vue centrale en utilisant alternativement une méthode naïve ou notre méthode de rendu multi-échelle. Nous espérons que notre méthode de rendu multi-échelle permette de résoudre à la fois le problème de trous dus à la sparsité d'information disponible dans les images sources, et le problème d'inhomogénéité des contributions. La méthode naïve quant à elle consiste simplement à effectuer un rendu par éclaboussure des vues sources. Nous supposons que la géométrie a été reconstruite avec précision : pour cela nous utilisons les cartes de profondeurs prises pour les expériences précédentes comme *proxy géométrique*.

Chaque image source est occultée par un masque qui simule l'application d'un algorithme de *matting* afin de se débarrasser des pixels de l'objet occultant l'arrière-plan. Nous avons généré ces masques de manière aléatoire en utilisant un bruit FBM (*Fractal Brownian Motion*, `https://thebookofshaders.com/13/`) avec un seuillage pour en tirer une image binaire (0 pour les pixels occultée et 1 pour les pixels de l'arrière-plan visibles). Nous faisons varier deux paramètres lorsque nous générons ces masques : le seuil T et la profondeur de bruit Z. La profondeur de bruit Z est incrémentée de 1 d'un masque à l'autre afin de servir de graine aléatoire. Le seuil

58% d'occultation, rendu naïf

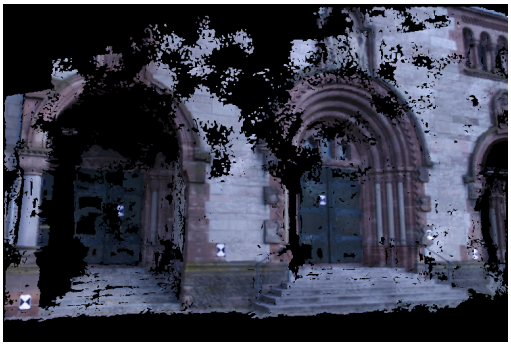
58% d'occultation, rendu multi-échelles

84% d'occultation, rendu naïf

84% d'occultation, rendu multi-échelles

55% d'occultation, rendu naïf

55% d'occultation, rendu multi-échelles



81% d'occultation, rendu naïf

81% d'occultation, rendu multi-échelles

Fig. 7.11 – *Images résultats sur les jeux de données* fountain *et* herzjesu.

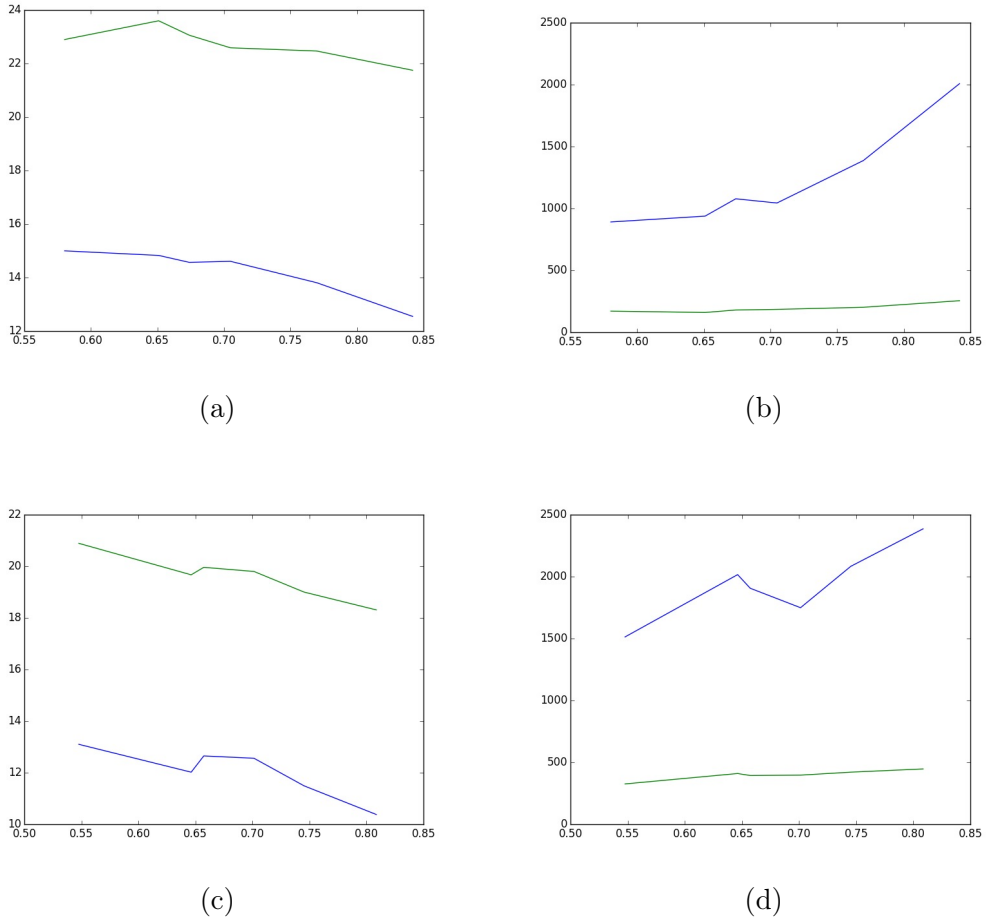


Fig. 7.12 – *Évolution de la qualité du rendu en fonction du pourcentage moyen de pixels occultés dans les images sources, de 0 (pas d'occultation) à 1 (images totalement occultées). Nous avons représenté en rouge les résultats numériques du rendu naïf, et en bleu ceux du rendu multi-échelles. (a) PSNR du jeu* fountain. *(b) DSSIM du jeu* fountain. *(c) PSNR du jeu* herzjesu. *(d) DSSIM du jeu* herzjesu.

T permet de contrôler la proportion de pixels occultés dans l'image ; nous le faisons varier de 0.50 à 0.60 en incrémentant de 0.02 pour produire 6 séries de masques avec des taux d'occultation différents. Pour indication un seuil $T = 0.50$ correspond environ à un taux d'occultation de $55 - 60\%$, et $T = 0.60$ à $80 - 85\%$. La figure 7.9 illustre certains de ces masques binaires générés avec du bruit FBM. Notons que notre algorithme fonctionne tout aussi bien avec des masques en niveaux de gris ; il est donc possible d'avoir des masques donnant la proportion du pixel, sur une échelle allant de 0 à 1, qui appartient à l'arrière-plan. Dans l'algorithme de rendu, ces masques multiplient les poids de contribution de chaque pixel de chaque vue dans le rendu.

La figure 7.11 montre quelques résultats de rendu naïf et multi-échelles, pour les seuils minimaux et maximaux testés, $T = 0.50$ et $T = 0.60$. Le rendu multi-échelles génère chaque pixel dans une continuité de tons, ce qui participe à l'homogénéisation

de l'image. En outre les trous sont remplis ; il n'est donc pas nécessaire d'employer une méthode d'*inpainting*. Notons que l'homogénéisation des couleurs a pour effet indésirable de faire « pâlir » l'image, qui parait plus fade que l'originale. Un autre problème est que les zones des images sources occultées procurent des faux contours qui viennent perturber le mélange de laplaciens, créant des effets de halos similaires à du *ringing*, comme à droite de l'image du jeu *fountain* par exemple. Une solution consisterait à éroder les zones positives du masque en attribuant à leur contour des valeurs intermédiaires strictement inférieures à 1 afin de limiter l'influence de ces contours d'occultation dans le rendu de laplaciens. La figure 7.12 vient appuyer ces résultats visuels par des mesures numériques, prises en comparant les images synthétisées avec les originales. Concluons que l'efficacité de la méthode multi-échelles peut être remise en cause lorsque le pourcentage d'occultation devient trop important (80%).

7.5 Conclusion

Dans ce chapitre nous avons abordé trois sujets de recherche qui entrent dans le cadre d'application de la vision à longue distance : la synthèse d'un point de vue plus proche du sujet, débarrassé d'un éventuel camouflage, et le placement des caméras sources contraintes qui optimise cette synthèse de point de vue.

Concernant le placement des caméras, il reste à faire des expériences sur des scènes réelles dont le *proxy géométrique* a été préalablement reconstruit. Elles consisteraient à la fois à trouver le placement optimal de caméras qui minimise la fonction de coût, à l'aide l'un algorithme génétique par exemple, et à comparer le rendu basé sur le ou les placements trouvés, avec le rendu basé sur d'autres configurations. La méthode est validée si le rendu basé sur le placement « optimal » est systématiquement meilleur. Il serait intéressant de comparer le placement optimal trouvé pour différents *proxys géométriques*, afin de savoir s'il dépend beaucoup de la reconstruction 3D pour des scènes classiques du type « sujet plus arrière-plan ». Si pour des *proxys géométriques* simples (un plan pour le sujet et un arrière-plan par exemple) le placement de caméra optimal estimé est le même que pour un *proxy géométrique* plus complexe (ce que nous intuitionnons), on peut alors s'affranchir de la reconstruction pour le calcul du placement idéal.

Le placement optimal de caméras ayant été calculé, il faut maintenant résoudre le problème de synthèse de vue. L'approche proposée consiste à utiliser les techniques de rendu basé image présentées dans cette thèse pour synthétiser un nouveau point de vue, c'est-à-dire reconstruire les rayons optiques manquants (orientation et couleur) dans le but de générer une nouvelle image. Cette approche tranche avec celle qui consiste simplement à « zoomer » sur le sujet, c'est-à-dire réduire le champ de vue de l'une des vues sources, ce qui revient à densifier l'échantillonnage de l'espace plénoptique. Contrairement au zoom, la synthèse de point de vue plus proche demande de reconstruire de nouveaux rayons à partir de rayons sources qui pour la plupart ne passent pas par le centre optique de la nouvelle vue. Cette approche demande donc une modélisation précise des *points visuels*, sans quoi les rayons sont mal reconstruits et des artefacts de rendu dégradent l'image que l'on obtient.

Enfin le sujet observé peut être camouflé derrière un buisson ou autre objet occultant. Le problème de la reconstruction d'un *proxy géométrique* dans ce contexte particulier est déjà traité dans quelques publications. Cependant, la qualité du rendu ne dépend pas exclusivement de la qualité de la reconstruction. Nous proposons de résoudre à la fois le problème de remplissage des trous et d'homogénéisation des couleurs de l'image en générant un nouveau point de vue à l'aide de notre méthode d'IBR multi-échelle, après s'être débarrassé de l'objet occultant dans les images à l'aide d'un algorithme de *matting*. De la même manière il serait intéressant d'étudier l'impact du *matting* sur la reconstruction du *proxy géométrique*. En effet nous pouvons imaginer que sans les couleurs de l'objet occultant (le vert du buisson par exemple) qui polluent les images sources, la reconstruction serait de bien meilleure qualité.

8 Conclusion

8.1 Résumé

L'objectif de cette thèse est d'utiliser un dispositif multi-vues pour se rapprocher virtuellement d'un sujet, potentiellement camouflé, afin de pouvoir l'observer sans perte de relief. Cela permettrait par exemple de pouvoir tourner un film 3D stéréoscopique sur un sujet distant sans avoir à « zoomer », ce qui crée de la divergence oculaire et de la perte de *rondeur*. Synthétiser un nouveau point de vue consiste à reconstruire des rayons optiques en se basant sur un certain échantillonnage de l'espace plénoptique, l'espace des rayons géométriques et des couleurs. Notre approche a consisté à faire appel au rendu basé image. Nous avons disséqué les méthodes de rendu basé image pour en extraire leur base commune : une approche bicéphale où les uns s'évertuent à reconstruire la géométrie de la scène avec précision tandis que les autres la prennent pour acquise et tente d'en restituer une vue de qualité. Nous avons proposé une méthode (Nieto *et al.*, 2017a) pour reconstruire le champ de lumière qui vient se substituer aux méthodes classiques de MVS. Nous avons cherché à modéliser des *points visuels* qui sont ensuite rendus par des méthodes directes. En outre nous avons proposé une nouvelle méthode de rendu variationnel (Nieto *et al.*, 2016b,a, 2017b) et une méthode de rendu direct multi-échelles dont le but est de résoudre les problèmes d'artefacts de discontinuité liées aux imperfections du modèle géométrique. Nous espérons que ces contributions dans le domaine du rendu basé images seront utiles dans la production de dispositifs de capture optimaux pour la vision à longue distance.

En résumé nos contributions sont :

- Une chaîne logicielle complète de rendu basé image, de la modélisation de l'espace plénoptique à la génération d'une nouvelle vue de la scène.
- Un aperçu des différentes manières de calculer les fonctions de déformation à partir des modèles géométriques estimés par les principales méthodes de MVS.
- Un point sur la déformation de l'espace d'échelle par les fonctions de déformation ; en découle un algorithme de rendu basé image multi-échelle.
- Une méthode variationnelle de rendu basé image qui surpasse les méthodes

de l'état de l'art. L'idée clé est d'imposer des contraintes sur le gradient de l'image solution, afin de corriger les artefacts dus à l'imperfection des fonctions de déformation.
– Une approche originale pour modéliser l'espace plénoptique, afin de reconstruire le champ de lumière. Cette méthode s'affranchit de la reconstruction explicite d'un *proxy géométrique*, et permet d'extrapoler avec précision de nouvelles vues de scènes contenant des réfractions et transparences.
– L'application de nos méthodes de rendu basé image au problème de vision à longue distance d'objets camouflés.
– Des pistes de réflexion sur le placement de caméras contraintes qui optimise le rendu d'une nouvelle vue.

8.2 Travaux futurs

Méthodes de rendu La méthode variationnelle pourrait être retravaillée pour optimiser directement les gradients de la vue cible, plutôt que les intensités ; puis l'intensité de l'image pourrait être reconstruite en résolvant l'équation de Poisson. Cela devrait complètement enlever les effets de halo et autres artefacts dus à l'imperfection du *proxy géométrique* dans l'image synthétisée. En outre nous pourrions combiner l'approche variationnelle avec l'approche multi-échelle, en imaginant un processus d'optimisation multi-résolution pour générer une nouvelle image : une première solution de faible résolution serait d'abord trouvée en résolvant l'équation à une échelle donnée, qui servirait ensuite d'initialisation à une résolution supérieure. Cela requerrait la formulation de l'énergie à différents niveaux de résolution, mais nous avons déjà fourni les outils mathématiques nécessaires dans le chapitre sur les transformations d'espace d'échelle.

Modélisation de l'espace plénoptique Les modèles complexes de *point visuel* apportent des résultats satisfaisants, mais ils pourraient être enrichis en modélisant par exemple des comportements non-linéaires de la lumière. Une autre extension serait d'incorporer une dimension temporelle dans nos modèles de points visuels, et de calculer des congruences de droites qui varient avec le temps. Cela pourrait être utilisé afin de synthétiser des vues à partir de vidéos asynchrones ou d'ensembles de photographies prises à des endroits et à des moments différents.

Vision à longue distance Le problème du placement de caméras qui optimise le rendu d'une nouvelle vue est un sujet original qui mérite beaucoup d'intérêt, en particulier pour la synthèse d'un point de vue proche du sujet. Cela est d'autant plus vrai que le rendu basé image est rarement utilisé pour synthétiser un point de vue plus proche de la scène. Nous avons vu que cela produisait de nombreux artefacts, car contrairement à la simulation d'un zoom, ou à l'interpolation de point de vue, les rayons optiques à reconstruire sont presque tous inédits. De nombreux travaux restent à entreprendre dans le domaine de la modélisation des *points visuels* pour résoudre le problème des artefacts de rendu. Outre la modélisation de la scène,

le rendu d'objets occultés à proprement parler pourrait être largement amélioré en supprimant le camouflage des images sources par un algorithme de *matting* multi-vues.

A Formulaire de calcul matriciel

Voici quelques formules usuelles de dérivation de matrices et de vecteurs que nous utilisons tout au long de ce manuscrit. Par convention nous avons choisi nos notations de la manière *denominator layout*[1]. Ainsi, si nous dérivons le vecteur $\mathbf{y}$ à n éléments par rapport au vecteur $\mathbf{x}$ à m éléments, $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ est représenté par une matrice de taille $n \times m$ (n lignes et m colonnes). Pour plus de souplesse d'application de ces formules, les vecteurs sont considérés comme des matrices à une colonne, et les scalaires des matrices à une colonne et à une ligne. De cette manière il est possible d'utiliser la même formule pour différents objets (matrices, vecteurs ou scalaires) à condition que les contraintes de tailles soient respectées lors des opérations élémentaires de multiplication, transpose, addition, etc.

Nous dérivons par rapport à la variable scalaire x ou vectorielle $\mathbf{x}$ des expressions composées des objets suivants :

- constante vectorielle : $\mathbf{a}$
- fonction vectorielle : $\mathbf{u}(.)$, $\mathbf{v}(.)$
- constante matricielle : $\mathbf{A}$
- fonction matricielle : $\mathbf{U}(.)$

A.1 Produit scalaire

A.1.1 Par un scalaire

$$\frac{\partial}{\partial x}(\mathbf{u} \cdot \mathbf{v}) = \mathbf{u} \cdot \frac{\partial \mathbf{v}}{\partial x} + \frac{\partial \mathbf{u}}{\partial x} \cdot \mathbf{v}. \tag{A.1}$$

On utilisera plutôt la notation matricielle $\frac{\partial}{\partial x}(\mathbf{u}^\intercal \mathbf{v}) = \mathbf{u}^\intercal \frac{\partial \mathbf{v}}{\partial x} + \frac{\partial \mathbf{u}^\intercal}{\partial x} \mathbf{v}$.

On a aussi pour $\mathbf{A}$ une matrice constante $\frac{\partial}{\partial x}(\mathbf{u}^\intercal \mathbf{A} \mathbf{v}) = \mathbf{u}^\intercal \mathbf{A} \frac{\partial \mathbf{v}}{\partial x} + \frac{\partial \mathbf{u}^\intercal}{\partial x} \mathbf{A} \mathbf{v}$. (A.2)

1. https://en.wikipedia.org/wiki/Matrix_calculus

En particulier si $\mathbf{A}$ est symétrique $\frac{\partial}{\partial x}(\mathbf{u}^\intercal \mathbf{A}\mathbf{u}) = 2\mathbf{u}^\intercal \mathbf{A}\frac{\partial \mathbf{u}}{\partial x}$. (A.3)

A.1.2 Par un vecteur

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{u}^\intercal \mathbf{v}) = \mathbf{u}^\intercal \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}^\intercal}{\partial \mathbf{x}}\mathbf{v}. \tag{A.4}$$

En particulier pour un vecteur constant $\mathbf{a}$, $\frac{\partial}{\partial \mathbf{x}}(\mathbf{a}^\intercal \mathbf{v}) = \mathbf{a}^\intercal \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$. (A.5)

On en déduit que $\frac{\partial}{\partial \mathbf{x}}(\mathbf{v}[i]) = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}[i]$ (A.6)

$\mathbf{A}$ une matrice constante, $\frac{\partial}{\partial \mathbf{x}}(\mathbf{u}^\intercal \mathbf{A}\mathbf{v}) = \mathbf{u}^\intercal \mathbf{A}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^\intercal \mathbf{A}^\intercal \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$. (A.7)

Si $\mathbf{A}$ est symétrique, $\frac{\partial}{\partial \mathbf{x}}(\mathbf{u}^\intercal \mathbf{A}\mathbf{u}) = \mathbf{u}^\intercal (\mathbf{A} + \mathbf{A}^\intercal)\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ et (A.8)

$$\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\intercal \mathbf{A}\mathbf{x}) = 2\mathbf{x}^\intercal \mathbf{A}. \tag{A.9}$$

A.2 Produit matriciel

$$\frac{\partial}{\partial x}(\mathbf{U}\mathbf{v}) = \mathbf{U}\frac{\partial \mathbf{v}}{\partial x} + \frac{\partial \mathbf{U}}{\partial x}\mathbf{v}. \tag{A.10}$$

A.3 Inverse

$$\frac{\partial}{\partial x}(\mathbf{U}^{-1}) = -\mathbf{U}^{-1}\frac{\partial \mathbf{U}}{\partial x}\mathbf{U}^{-1} \tag{A.11}$$

B

Pyramide laplacienne

Le concept de pyramide laplacienne fut introduit par Burt et Adelson (1983a) dans le but d'encoder les images pour faciliter le travail d'analyse et de compression. Le construction de la pyramide s'appuie sur l'utilisation de filtres appelés filtres HDC (*Hierarchical Discrete Correlation*) (Burt, 1981) qui, appliqués successivement, approchent un filtre gaussien (passe-bas). Ces pyramides furent ensuite utilisées à des fins de mélange de vues multi-résolutions, ou multi-échelles (Burt et Adelson, 1983b) ; c'est ce cas d'application qui nous intéresse. Avant d'aborder la notion de pyramide laplacienne, il est nécessaire de présenter la pyramide gaussienne.

B.1 Pyramide gaussienne

La pyramide gaussienne (I_n) permet de visualiser une image I à plusieurs niveaux de résolution. Le niveau initial $n = 0$ est appelé $I_0 = I$; c'est le niveau de résolution maximale dans la pyramide. Tous les niveaux supérieurs, c'est-à-dire pour $n > 0$, sont de résolution inférieure. Les niveaux supérieurs de la pyramide sont successivement calculés à l'aide d'un filtre HDC. Dans cette thèse nous utilisons un filtre impair, aussi appelé `Odd HDC`. Le noyau 5×5 du filtre est symétrique et séparable. Chaque filtre 1D se présente sous la forme $w = [0.25 - 0.50a, 0.25, a, 0.25, 0.25 - 0.50a]$, avec a que l'on prend égal à 0.4 pour approximer une gaussienne (figure B.1). Cette opération est accompagnée d'un sous-échantillonnage optionnel par un facteur 2. Pour tout niveau n, le niveau $n + 1$ s'obtient par application de ce filtre, une opération de réduction appelée `reduce` :

$$I_{n+1} = \texttt{reduce}(I_n). \tag{B.1}$$

I_n est donc l'image I à laquelle on a appliqué le filtre n fois. Pour n qui tend vers l'infini, ce filtre approxime une gaussienne. On désignera alors de manière équivalente un niveau de résolution par l'entier n ou par l'écart-type σ du filtre gaussien appliqué, comme c'est le cas dans le chapitre 4. Une pyramide d'images du jeu de données *fountain* est illustrée sur la figure B.2.

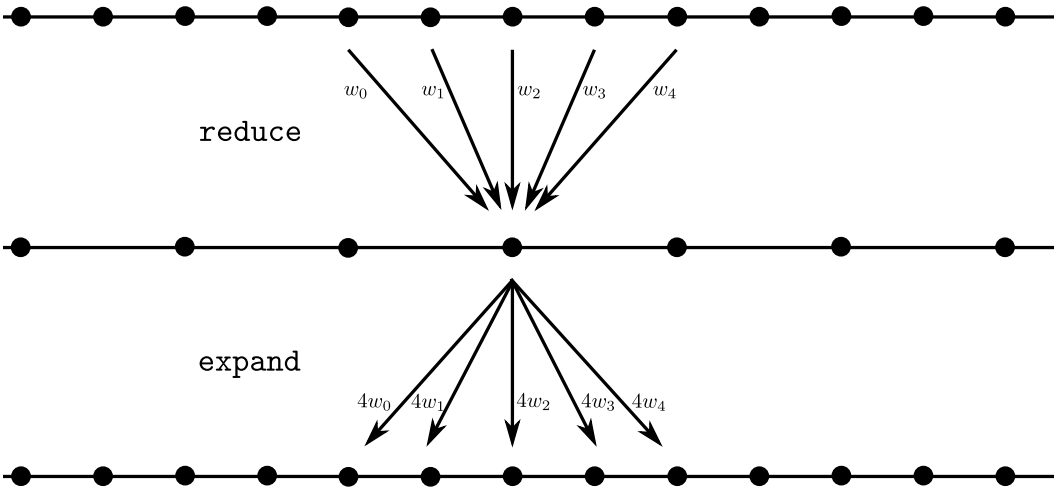


Fig. B.1 – *Détails d'implémentation des opérations* **`reduce`** *et* **`expand`**, *pour un noyau 1D* $w = [0.25 - 0.50a, 0.25, a, 0.25, 0.25 - 0.50a]$. *L'image* I_n *à un niveau de résolution* n *est réduite : on obtient alors une image* $I_{n+1} = reduce(I_n)$ *à un niveau de résolution* $n + 1$ *(la résolution est inférieure). Notez que l'image est simultanément sous-échantillonnée d'un facteur 2. L'opération* **`expand`** *sur-échantillonne l'image précédemment réduite. L'image finale est soustraite à* I_n *pour obtenir une approximation du laplacien* ΔI_n *de* I *au niveau de résolution* n.

B.2 Pyramide laplacienne

La pyramide laplacienne (ΔI_n) représente le laplacien de l'image à plusieurs niveaux de résolution. Sa construction est basée sur le constat qu'une différence de gaussiennes (DoG) approxime le laplacien. En effet le filtre gaussien étant un filtre passe-bas, une différence de deux niveaux de résolution I_n et I_{n+1} de la même image I conserve dans l'image une certaine bande de fréquence qui dépend de n. Ce sont ces mêmes fréquences qui sont préservées par un laplacien calculé au niveau de résolution n, qui agit comme un filtre passe-bande. Si deux niveaux consécutifs n'ont pas la même résolution d'image (en pixels), alors il est nécessaire d'appliquer

Fig. B.2 – *Pyramide gaussienne, aussi appelée pyramide d'images, constituée ici de trois niveaux de résolution,* $n = 0$, $n = 1$ *et* $n = 2$.

Fig. B.3 – *Pyramide laplacienne, constituée ici de trois niveaux de résolution, $n = 0$, $n = 1$ et $n = 2$.*

l'opération `expand` qui sur-échantillonne l'image précédemment réduite ; sinon une simple soustraction suffit. Un niveau n de la pyramide laplacienne s'obtient comme suit :

$$\Delta I_n = I_n - \texttt{expand}(I_{n+1}). \tag{B.2}$$

Chaque laplacien contient les détails de l'image à une certaine résolution. La résolution d'image (en pixels) du laplacien ΔI_n est identique à celle de I_n. La pyramide laplacienne d'une image de *fountain* est illustrée dans la figure B.3. La dernière image d'une pyramide de hauteur n_h, notée I_{n_h} et appelée la fondamentale, sert à amorcer le processus de reconstruction de l'image initiale I à partir des laplaciens. Nous appelons cette opération l'« effondrement » de la pyramide. Soit une pyramide laplacienne $((\Delta I_n), I_{n_h})$; on retrouve l'image initiale par application successive pour $n > 0$ de l'opération inverse de (B.2) :

$$I_{n-1} = \Delta I_{n-1} + \texttt{expand}(I_n). \tag{B.3}$$

Si les images ont la même résolution d'image (en pixels), alors l'effondrement est simplement une somme des laplaciens à tous les niveaux de résolution, à laquelle on ajoute la fondamentale.

B.3 Mélange laplacien

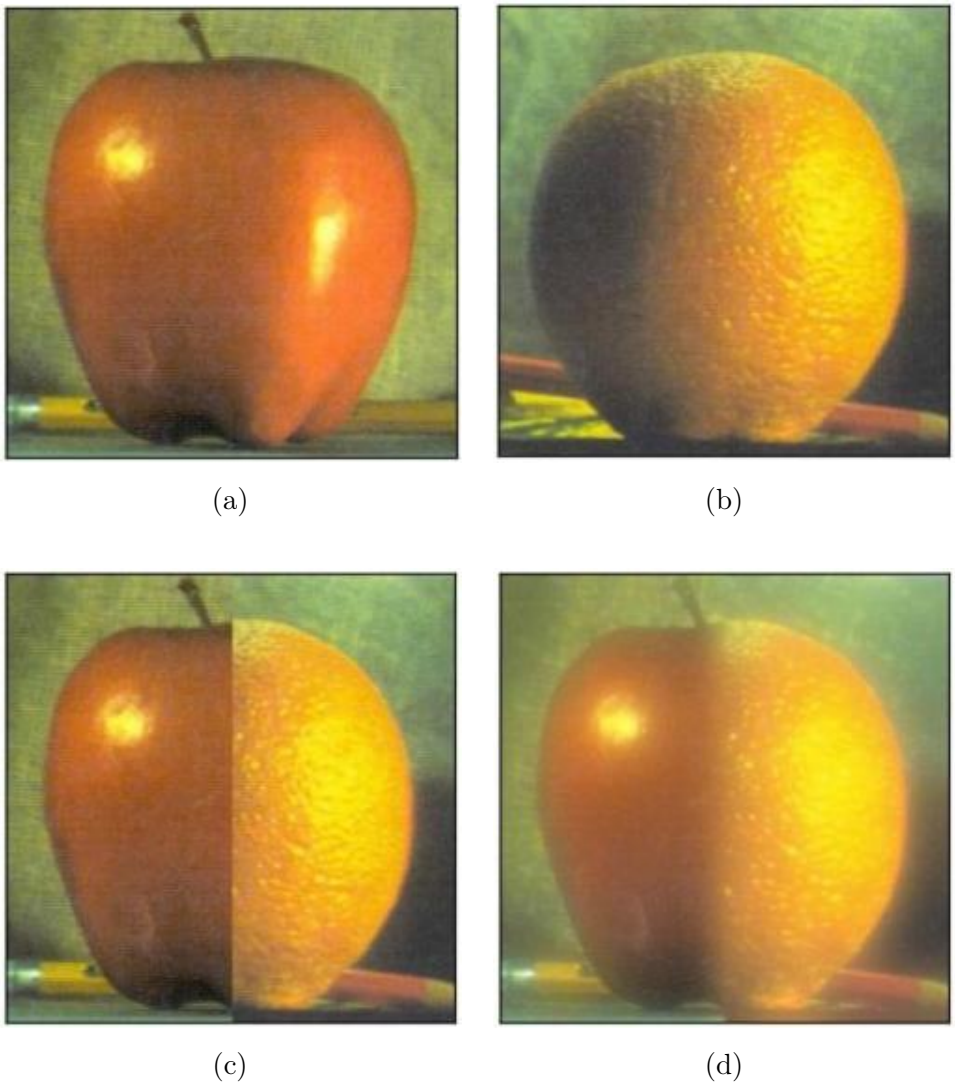
L'application qui nous intéresse est le mélange laplacien, car il permet de combiner plusieurs images à toutes les fréquences. Soient I_1 et I_2 deux images, comme la pomme et l'orange sur la figure B.4. On note $((\Delta I_{1,n}), I_{1,n_h})$ et $((\Delta I_{2,n}), I_{2,n_h})$ leur pyramide laplacienne, ω_1 et ω_2 leur poids de mélange et $(\omega_{1,n})$ et $(\omega_{2,n})$ les pyramides gaussiennes de ces derniers. Après avoir décomposé chaque image en sa pyramide laplacienne, on combine les laplaciens de chaque niveau n en un seul ΔI_n de la manière suivante :

$$\Delta I_n = \omega_{1,n}\Delta I_{1,n} + \omega_{2,n}\Delta I_{2,n}. \tag{B.4}$$

La fondamentale de la nouvelle pyramide est obtenue selon la même combinaison :

$$I_{n_h} = \omega_{1,n_h} \Delta I_{1,n_h} + \omega_{2,n_h} \Delta I_{2,n_h}. \tag{B.5}$$

Enfin la pyramide est effondrée et on obtient un mélange lisse I.



(a) (b)

(c) (d)

Fig. B.4 – *Mélange laplacien. (a) Pomme. (b) Orange. (c) Juxtaposition. (d) Mélange laplacien.*

Bibliographie

Adato, Y., Vasilyev, Y., Zickler, T. et Ben-Shahar, O. "Shape from Specular Flow". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2054–2070 (2010). ISSN 0162-8828. doi:10.1109/TPAMI.2010.126. 98

Adelson, E.H. et Wang, J.Y.A. "Single Lens Stereo with a Plenoptic Camera". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106 (1992). ISSN 0162-8828. doi:10.1109/34.121783. 8, 10

Adelson, E.H. et Bergen, J.R. "The plenoptic function and the elements of early vision". Dans *Computational models of visual processing*, 1(2):3–20 (1991). 7, 8, 99

Agarwal, S., Mierle, K. et Others. "Ceres Solver". Dans (2017). 114

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D. et Cohen, M. "Interactive Digital Photomontage". Dans "ACM SIGGRAPH 2004 Papers", SIGGRAPH '04, pages 294–302. ACM, New York, NY, USA (2004). doi:10.1145/1186562.1015718. 85

Alterman, M., Schechner, Y.Y. et Swirski, Y. "Triangulation in random refractive distortions". Dans "IEEE International Conference on Computational Photography (ICCP)", pages 1–10 (2013). doi:10.1109/ICCPhot.2013.6528314. 100

Baker, S. et Kanade, T. "Limits on super-resolution and how to break them". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183 (2002). ISSN 0162-8828. doi:10.1109/TPAMI.2002.1033210. 79, 80

Beck, A. et Teboulle, M. "A fast Iterative Shrinkage-Thresholding Algorithm with application to wavelet-based image deblurring". Dans "IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009", pages 693–696 (2009). doi:10.1109/ICASSP.2009.4959678. 87

Beder, C. et Steffen, R. "Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence". Dans K. Franke, K.R. Müller,

B. Nickolay et R. Schäfer, editeurs, "Pattern Recognition", Numéro 4174 dans Lecture Notes in Computer Science, pages 657–666. Springer Berlin Heidelberg (2006). ISBN 978-3-540-44412-1 978-3-540-44414-5. 121, 126

Bishop, T. et Favaro, P. "The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):972–986 (2012). ISSN 0162-8828. doi:10.1109/TPAMI.2011.168. 79

Bishop, T., Zanetti, S. et Favaro, P. "Light field superresolution". Dans "2009 IEEE International Conference on Computational Photography (ICCP)", pages 1–9 (2009). doi:10.1109/ICCPHOT.2009.5559010. 79

Botsch, M., Hornung, A., Zwicker, M. et Kobbelt, L. "High-quality surface splatting on today's GPUs". Dans "Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings", pages 17–141 (2005). doi:10.1109/PBG.2005.194059. 56

Bredies, K., Kunisch, K. et Pock, T. "Total Generalized Variation". Dans *SIAM Journal on Imaging Sciences*, 3(3):492–526 (2010). doi:10.1137/090769521. 83

Buehler, C., Bosse, M., McMillan, L., Gortler, S. et Cohen, M. "Unstructured Lumigraph Rendering". Dans "Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '01, pages 425–432. ACM, New York, NY, USA (2001). ISBN 1-58113-374-X. doi:10.1145/383259.383309. 17, 21, 51, 52, 60, 78, 80, 93, 99, 112, 113, 129, 137

Burt, P. et Adelson, E. "The Laplacian Pyramid as a Compact Image Code". Dans *IEEE Transactions on Communications*, 31(4):532–540 (1983a). ISSN 0090-6778. doi:10.1109/TCOM.1983.1095851. 64, 69, 149

Burt, P.J. "Fast filter transform for image processing". Dans *Computer Graphics and Image Processing*, 16(1):20–51 (1981). ISSN 0146-664X. doi:10.1016/0146-664X(81)90092-7. 41, 63, 69, 149

Burt, P.J. et Adelson, E.H. "A Multiresolution Spline with Application to Image Mosaics". Dans *ACM Trans. Graph.*, 2(4):217–236 (1983b). ISSN 0730-0301. doi:10.1145/245.247. 149

Butler, D.J., Wulff, J., Stanley, G.B. et Black, M.J. "A Naturalistic Open Source Movie for Optical Flow Evaluation". Dans A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato et C. Schmid, editeurs, "Computer Vision – ECCV 2012", Numéro 7577 dans Lecture Notes in Computer Science, pages 611–625. Springer Berlin Heidelberg (2012). ISBN 978-3-642-33782-6 978-3-642-33783-3. 25

Chai, J.X., Tong, X., Chan, S.C. et Shum, H.Y. "Plenoptic Sampling". Dans "Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '00, pages 307–318. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000). ISBN 978-1-58113-208-3. doi:10.1145/344779.344932. 99

Chambolle, A. "An Algorithm for Total Variation Minimization and Applications". Dans *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97 (2004). ISSN 0924-9907, 1573-7683. doi:10.1023/B:JMIV.0000011325.36760.1e. 83

Chaurasia, G., Duchene, S., Sorkine-Hornung, O. et Drettakis, G. "Depth Synthesis and Local Warps for Plausible Image-based Navigation". Dans *ACM Trans. Graph.*, 32(3):30:1–30:12 (2013). ISSN 0730-0301. doi:10.1145/2487228.2487238. 17, 52

Chen, Q., Li, D. et Tang, C.K. "KNN Matting". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2175–2188 (2013). ISSN 0162-8828. doi:10.1109/TPAMI.2013.18. 135

Cho, T.S., Zitnick, C.L., Joshi, N., Kang, S.B., Szeliski, R. et Freeman, W.T. "Image Restoration by Matching Gradient Distributions". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):683–694 (2012). ISSN 0162-8828. doi:10.1109/TPAMI.2011.166. 83

Connolly, T.J. et Lane, R.G. "Gradient methods for superresolution". Dans ", International Conference on Image Processing, 1997. Proceedings", volume 1, pages 917–920 vol.1 (1997). doi:10.1109/ICIP.1997.648116. 78

Cristóbal, G., Gil, E., Šroubek, F., Flusser, J., Miravet, C. et Rodríguez, F.B. "Superresolution imaging: A survey of current techniques". volume 7074, pages 70740C–70740C–18 (2008). doi:10.1117/12.797302. 78

Davis, A., Levoy, M. et Durand, F. "Unstructured Light Fields". Dans *Computer Graphics Forum*, 31(2pt1):305–314 (2012). ISSN 1467-8659. doi:10.1111/j.1467-8659.2012.03009.x. 17, 120

Einarsson, P., Chabert, C.F., Jones, A., Ma, W.C., Lamond, B., Hawkins, T., Bolas, M., Sylwan, S. et Debevec, P. "Relighting Human Locomotion with Flowed Reflectance Fields". Dans "Proceedings of the 17th Eurographics Conference on Rendering Techniques", EGSR '06, pages 183–194. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006). ISBN 978-3-905673-35-7. doi:10.2312/EGWR/EGSR06/183-194. 99

Faramarzi, E., Rajan, D. et Christensen, M. "Unified Blind Method for Multi-Image Super-Resolution and Single/Multi-Image Blur Deconvolution". Dans *IEEE Transactions on Image Processing*, 22(6):2101–2114 (2013). ISSN 1057-7149. doi:10.1109/TIP.2013.2237915. 78

Fitzgibbon, A., Wexler, Y. et Zisserman, A. "Image-based rendering using image-based priors". Dans "Ninth IEEE International Conference on Computer Vision, 2003. Proceedings", pages 1176–1183 vol.2 (2003). doi:10.1109/ICCV.2003.1238625. 79

Flynn, J., Neulander, I., Philbin, J. et Snavely, N. "DeepStereo: Learning to Predict New Views from the World's Imagery". Dans *arXiv:1506.06825 [cs]* (2015). 79

Fuhrmann, S. et Goesele, M. "Fusion of Depth Maps with Multiple Scales". Dans "Proceedings of the 2011 SIGGRAPH Asia Conference", SA '11, pages 148:1–148:8. ACM, New York, NY, USA (2011). ISBN 978-1-4503-0807-6. doi:10.1145/2024156.2024182. 46

Fuhrmann, S. et Goesele, M. "Floating Scale Surface Reconstruction". Dans *ACM Trans. Graph.*, 33(4):46:1–46:11 (2014). ISSN 0730-0301. doi:10.1145/2601097.2601163. 45, 46, 47, 48, 53, 87

Fuhrmann, S., Langguth, F. et Goesele, M. "MVE - A Multi-View Reconstruction Environment". Dans "Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH)", (2014). 24, 38, 58, 73, 87

Furukawa, Y., Curless, B., Seitz, S.M. et Szeliski, R. "Towards Internet-scale multi-view stereo". Dans "2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition", pages 1434–1441 (2010). doi:10.1109/CVPR.2010.5539802. 31, 32

Furukawa, Y. et Ponce, J. "Accurate, Dense, and Robust Multiview Stereopsis". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376 (2010). ISSN 0162-8828. doi:10.1109/TPAMI.2009.161. 31, 32, 34, 46, 48, 87

Furukawa, Y. et Hernández, C. "Multi-View Stereo: A Tutorial". Dans *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148 (2015). ISSN 1572-2740, 1572-2759. doi:10.1561/0600000052. 134

Geiger, A. "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite". Dans "Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", CVPR '12, pages 3354–3361. IEEE Computer Society, Washington, DC, USA (2012). ISBN 978-1-4673-1226-4. 25

Geman, S. et Geman, D. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". Dans *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741 (1984). ISSN 0162-8828. doi:10.1109/TPAMI.1984.4767596. 79

Georgeiv, T., Zheng, K.C., Curless, B., Salesin, D., Nayar, S. et Intwala, C. "Spatio-angular Resolution Tradeoffs in Integral Photography". Dans "Proceedings of the 17th Eurographics Conference on Rendering Techniques", EGSR '06, pages 263–272. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006). ISBN 978-3-905673-35-7. doi:10.2312/EGWR/EGSR06/263-272. 99

Gershun, A. "The Light Field". Dans *Journal of Mathematics and Physics*, 18(1-4):51–151 (1939). ISSN 1467-9590. doi:10.1002/sapm193918151. 7

Goesele, M., Snavely, N., Curless, B., Hoppe, H. et Seitz, S. "Multi-View Stereo for Community Photo Collections". Dans "IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007", pages 1–8 (2007). doi:10.1109/ICCV.2007.4408933. 38, 39, 40, 46, 47, 48, 49, 50, 52

Goldluecke, B. et Cremers, D. "Superresolution texture maps for multiview reconstruction". Dans "2009 IEEE 12th International Conference on Computer Vision", pages 1677–1684 (2009). doi:10.1109/ICCV.2009.5459378. 78, 79

Goldluecke, B. et Cremers, D. "An approach to vectorial total variation based on geometric measure theory". Dans "2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pages 327–333 (2010). doi:10.1109/CVPR.2010.5540194. 83

Goldluecke, B., Strekalovskiy, E. et Cremers, D. "The natural vectorial total variation which arises from geometric measure theory". Dans *SIAM Journal on Imaging Sciences* (2012). 87

Gortler, S.J., Grzeszczuk, R., Szeliski, R. et Cohen, M.F. "The Lumigraph". Dans "Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '96, pages 43–54. ACM, New York, NY, USA (1996). ISBN 978-0-89791-746-9. doi:10.1145/237170.237200. 52, 63, 87, 99, 113

Greene, N. et Heckbert, P. "Creating Raster Omnimax Images from Multiple Perspective Views Using the Elliptical Weighted Average Filter". Dans *IEEE Computer Graphics and Applications*, 6(6):21–27 (1986). ISSN 0272-1716. doi:10.1109/MCG.1986.276738. 55

Gross, M. et Pfister, H. *Point-Based Graphics*. Morgan Kaufmann (2011). ISBN 978-0-08-054882-1. 58

Gu, J., Ramamoorthi, R., Belhumeur, P. et Nayar, S. "Removing Image Artifacts Due to Dirty Camera Lenses and Thin Occluders". Dans "ACM SIGGRAPH Asia 2009 Papers", SIGGRAPH Asia '09, pages 144:1–144:10. ACM, New York, NY, USA (2009). ISBN 978-1-60558-858-2. doi:10.1145/1661412.1618490. 133

Gurdan, T., Oswald, M.R., Gurdan, D. et Cremers, D. "Spatial and Temporal Interpolation of Multi-view Image Sequences". Dans X. Jiang, J. Hornegger et R. Koch, editeurs, "Pattern Recognition", Numéro 8753 dans Lecture Notes in Computer Science, pages 305–316. Springer International Publishing (2014). ISBN 978-3-319-11751-5 978-3-319-11752-2. doi:10.1007/978-3-319-11752-2_24. 52

Haner, S. et Heyden, A. "Covariance Propagation and Next Best View Planning for 3D Reconstruction". Dans A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato et C. Schmid, editeurs, "Computer Vision – ECCV 2012", Lecture Notes in Computer Science, pages 545–556. Springer Berlin Heidelberg (2012). ISBN 978-3-642-33708-6 978-3-642-33709-3. 121

Hardie, R., Barnard, K. et Armstrong, E. "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images". Dans *IEEE Transactions on Image Processing*, 6(12):1621–1633 (1997). ISSN 1057-7149. doi:10.1109/83.650116. 79, 80, 81

Hardie, R.C., Barnard, K.J., Bognar, J.G., Armstrong, E.E. et Watson, E.A. "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system". Dans *Optical Engineering*, 37(1):247–260 (1998). ISSN 0091-3286. doi:10.1117/1.601623. 79

Harmeling, S., Sra, S., Hirsch, M. et Scholkopf, B. "Multiframe blind deconvolution, super-resolution, and saturation correction via incremental EM". Dans "2010 17th IEEE International Conference on Image Processing (ICIP)", pages 3313–3316 (2010). doi:10.1109/ICIP.2010.5651650. 78

Harris, C. et Stephens, M. "A combined corner and edge detector". Dans "In Proc. of Fourth Alvey Vision Conference", pages 147–151 (1988). 32

Hartley, R. et Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press (2003). ISBN 978-0-521-54051-3. 29, 30

Heber, S. et Pock, T. "Shape from Light Field Meets Robust PCA". Dans D. Fleet, T. Pajdla, B. Schiele et T. Tuytelaars, editeurs, "Computer Vision –

ECCV 2014", Numéro 8694 dans Lecture Notes in Computer Science, pages 751–767. Springer International Publishing (2014). ISBN 978-3-319-10598-7 978-3-319-10599-4. doi:10.1007/978-3-319-10599-4_48. 99

Heber, S. et Pock, T. "Convolutional Networks for Shape From Light Field". pages 3746–3754 (2016). 99

Heckbert, P.S. "Fundamentals of Texture Mapping and Image Warping". Rapport technique UCB/CSD-89-516, EECS Department, University of California, Berkeley (1989). 51

Hess-Flores, M., Recker, S. et Joy, K. "Uncertainty, Baseline, and Noise Analysis for L1 Error-Based Multi-view Triangulation". Dans "2014 22nd International Conference on Pattern Recognition (ICPR)", pages 4074–4079 (2014). doi:10.1109/ICPR.2014.698. 121

Heuel, S. *Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction*. Springer Science & Business Media (2004). ISBN 978-3-540-22029-9. 23

Hong, S.H. et Javidi, B. "Three-dimensional visualization of partially occluded objects using integral imaging". Dans *Journal of Display Technology*, 1(2):354–359 (2005). ISSN 1551-319X. doi:10.1109/JDT.2005.858879. 134

Hwang, Y.S., Hong, S.H. et Javidi, B. "Free View 3-D Visualization of Occluded Objects by Using Computational Synthetic Aperture Integral Imaging". Dans *Journal of Display Technology*, 3(1):64–70 (2007). ISSN 1551-319X. doi:10.1109/JDT.2006.890702. 134

Iffa, E., Wetzstein, G. et Heidrich, W. "Light field optical flow for refractive surface reconstruction". volume 8499, pages 84992H–84992H–8 (2012). doi:10.1117/12.981608. 100

Javidi, B., Ponce-Díaz, R. et Hong, S.H. "Three-dimensional recognition of occluded objects by using computational integral imaging". Dans *Optics Letters*, 31(8):1106 (2006). ISSN 0146-9592, 1539-4794. doi:10.1364/OL.31.001106. 134

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E. et Aanaes, H. "Large Scale Multi-view Stereopsis Evaluation". Dans "2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pages 406–413 (2014). doi:10.1109/CVPR.2014.59. 25

Joshi, N., Matusik, W. et Avidan, S. "Natural Video Matting Using Camera Arrays". Dans "ACM SIGGRAPH 2006 Papers", SIGGRAPH '06, pages 779–786. ACM, New York, NY, USA (2006). ISBN 978-1-59593-364-5. doi:10.1145/1179352.1141955. 135

Kalantari, N.K., Wang, T.C. et Ramamoorthi, R. "Learning-based View Synthesis for Light Field Cameras". Dans *ACM Trans. Graph.*, 35(6):193:1–193:10 (2016). ISSN 0730-0301. doi:10.1145/2980179.2980251. 79

Kazhdan, M. "Reconstruction of Solid Models from Oriented Point Sets". Dans "Proceedings of the Third Eurographics Symposium on Geometry Processing", SGP '05. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2005). ISBN 978-3-905673-24-1. 44, 45

Kazhdan, M., Bolitho, M. et Hoppe, H. "Poisson Surface Reconstruction". Dans "Proceedings of the Fourth Eurographics Symposium on Geometry Processing", SGP '06, pages 61–70. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006). ISBN 3-905673-36-3. 20, 44, 45, 46, 48, 50, 87

Kazhdan, M. et Hoppe, H. "Screened Poisson Surface Reconstruction". Dans *ACM Trans. Graph.*, 32(3):29:1–29:13 (2013). ISSN 0730-0301. doi:10.1145/2487228.2487237. 44, 45, 46, 49

Kopf, J., Cohen, M.F., Lischinski, D. et Uyttendaele, M. "Joint Bilateral Upsampling". Dans "ACM SIGGRAPH 2007 Papers", SIGGRAPH '07. ACM, New York, NY, USA (2007). doi:10.1145/1275808.1276497. 33, 34, 40, 47

Kopf, J., Cohen, M.F. et Szeliski, R. "First-person Hyper-lapse Videos". Dans *ACM Trans. Graph.*, 33(4):78:1–78:10 (2014). ISSN 0730-0301. doi:10.1145/2601097.2601195. 64, 85

Kopf, J., Langguth, F., Scharstein, D., Szeliski, R. et Goesele, M. "Image-based Rendering in the Gradient Domain". Dans *ACM Trans. Graph.*, 32(6):199:1–199:9 (2013). ISSN 0730-0301. doi:10.1145/2508363.2508369. 17, 52, 64, 85, 94

Kushal, A., Self, B., Furukawa, Y., Gallup, D., Hernandez, C., Curless, B. et Seitz, S.M. "Photo Tours". Dans "Visualization Transmission 2012 Second International Conference on 3D Imaging, Modeling, Processing", pages 57–64 (2012). doi:10.1109/3DIMPVT.2012.62. 52

Langer, M.S. "Surface Visibility Probabilities in 3D Cluttered Scenes". Dans D. Forsyth, P. Torr et A. Zisserman, editeurs, "Computer Vision – ECCV 2008", Numéro 5302 dans Lecture Notes in Computer Science, pages 401–412. Springer Berlin Heidelberg (2008). ISBN 978-3-540-88681-5 978-3-540-88682-2. 36

Levin, A., Zomet, A. et Weiss, Y. "Learning how to inpaint from global image statistics". Dans "Ninth IEEE International Conference on Computer Vision, 2003. Proceedings", pages 305–312 vol.1 (2003). doi:10.1109/ICCV.2003.1238360. 85

Levoy, M. "Light Fields and Computational Imaging". Dans *Computer*, 39(8):46–55 (2006). ISSN 0018-9162. 9, 10

Levoy, M. et Hanrahan, P. "Light Field Rendering". Dans "Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '96, pages 31–42. ACM, New York, NY, USA (1996). ISBN 978-0-89791-746-9. doi:10.1145/237170.237199. 7, 10, 99, 101

Linz, C., Lipski, C. et Magnor, M.A. "Multi-image Interpolation Based on Graph-cuts and Symmetric Optical Flow". Dans "ACM SIGGRAPH 2010 Posters", SIGGRAPH '10, pages 129:1–129:1. ACM, New York, NY, USA (2010). ISBN 978-1-4503-0393-4. doi:10.1145/1836845.1836983. 52

Lipski, C., Klose, F. et Magnor, M. "Correspondence and Depth-Image Based Rendering a Hybrid Approach for Free-Viewpoint Video". Dans *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):942–951 (2014). ISSN 1051-8215. doi:10.1109/TCSVT.2014.2302379. 17, 52

Lipski, C., Linz, C., Berger, K., Sellent, A. et Magnor, M. "Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time". Dans *Computer Graphics Forum*, 29(8):2555–2568 (2010). 17, 52

Lorensen, W.E. et Cline, H.E. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". Dans "Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '87, pages 163–169. ACM, New York, NY, USA (1987). ISBN 978-0-89791-227-3. doi:10.1145/37401.37422. 45

Maeno, K., Nagahara, H., Shimada, A. et Taniguchi, R.I. "Light Field Distortion Feature for Transparent Object Recognition". pages 2786–2793 (2013). 100

Mahajan, D., Huang, F.C., Matusik, W., Ramamoorthi, R. et Belhumeur, P. "Moving Gradients: A Path-based Method for Plausible Image Interpolation". Dans "ACM SIGGRAPH 2009 Papers", SIGGRAPH '09, pages 42:1–42:11. ACM, New York, NY, USA (2009). ISBN 978-1-60558-726-4. doi:10.1145/1576246.1531348. 52

McCann, J. et Pollard, N.S. "Real-time Gradient-domain Painting". Dans "ACM SIGGRAPH 2008 Papers", SIGGRAPH '08, pages 93:1–93:7. ACM, New York, NY, USA (2008). ISBN 978-1-4503-0112-1. doi:10.1145/1399504.1360692. 85

Menze, M. et Geiger, A. "Object scene flow for autonomous vehicles". Dans "2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pages 3061–3070 (2015). doi:10.1109/CVPR.2015.7298925. 25

Mitra, K. et Veeraraghavan, A. "Light field denoising, light field superresolution and stereo camera based refocussing using a GMM light field patch prior". Dans "2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops", pages 22–28 (2012). doi:10.1109/CVPRW.2012.6239346. 79

Moulon, P., Monasse, P. et Marlet, R. "La bibliothèque openMVG : open source Multiple View Geometry" (2013). 23, 24, 113

Mücke, P., Klowsky, R. et Goesele, M. *Surface Reconstruction from Multi-Resolution Sample Points*. The Eurographics Association (2011). ISBN 978-3-905673-85-2. 46

Nieto, G., Devernay, F. et Crowley, J. "Variational image-based rendering with gradient constraints". Dans "2016 International Conference on 3D Imaging (IC3D)", pages 1–8 (2016a). doi:10.1109/IC3D.2016.7823449. vii, 3, 49, 97, 143

Nieto, G., Devernay, F. et Crowley, J. "Linearizing the Plenoptic Space". Dans "CVPR Workshops (LF4CV)", (2017a). vii, 2, 143

Nieto, G., Devernay, F. et Crowley, J. "Rendu basé image avec contraintes sur les gradients". Dans "Numéro Spécial de La Revue Traitement Du Signal", (2017b). vii, 3, 143

Nieto, G., Devernay, F. et Crowley, J. "Rendu basé image avec contraintes sur les gradients". Dans "Reconnaissance Des Formes et l'Intelligence Artificielle, RFIA 2016", Clermont-Ferrand, France (2016b). vii, 3, 143

Nieto, G.A., Devernay, F. et Crowley, J.L. "Placement optimal de caméras contraintes pour la synthèse de nouvelles vues". Dans "Journées Francophones Des

Jeunes Chercheurs En Vision Par Ordinateur", page 2. Amiens, France (2015). vii, 2

Olague, G. et Mohr, R. "Optimal camera placement for accurate reconstruction". Dans *Pattern Recognition*, 35(4):927–944 (2002). ISSN 0031-3203. doi:10.1016/S0031-3203(01)00076-0. 121, 122

Ortiz-Cayon, R., Djelouah, A. et Drettakis, G. "A Bayesian Approach for Selective Image-Based Rendering using Superpixels". Dans "3D Vision (3DV), International Conference On", IEEE (2015). 52

Park, T.J., Shin, S.Y. et Lee, S. "Optical Flow Rendering". Dans *Computer Graphics Forum*, 17(3):75–85 (1998). ISSN 1467-8659. doi:10.1111/1467-8659.00255. 99

Pérez, P., Gangnet, M. et Blake, A. "Poisson Image Editing". Dans "ACM SIGGRAPH 2003 Papers", SIGGRAPH '03, pages 313–318. ACM, New York, NY, USA (2003). ISBN 978-1-58113-709-5. doi:10.1145/1201775.882269. 64, 85

Pfister, H., Zwicker, M., van Baar, J. et Gross, M. "Surfels: Surface Elements As Rendering Primitives". Dans "Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '00, pages 335–342. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000). ISBN 1-58113-208-5. doi:10.1145/344779.344936. 56

Ponce, J., Sturmfels, B. et Trager, M. "Congruences and Concurrent Lines in Multi-View Geometry". Dans *arXiv:1608.05924 [cs, math]* (2016). 98

Pottmann, H. et Wallner, J. "Line Congruences and Line Complexes". Dans "Computational Line Geometry", Mathematics and Visualization, pages 423–496. Springer Berlin Heidelberg (2010). ISBN 978-3-642-04017-7 978-3-642-04018-4. doi:10.1007/978-3-642-04018-4_7. 102

Pujades, S., Devernay, F. et Goldluecke, B. "Bayesian View Synthesis and Image-Based Rendering Principles". Dans "2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pages 3906–3913 (2014). doi:10.1109/CVPR.2014.499. 42, 61, 78, 79, 88, 89, 93, 95, 97, 125

Pujades, S. et Devernay, F. "Utilisation des longues focales lors des prises de vues stéréoscopiques". Dans "Orasis, Congrès Des Jeunes Chercheurs En Vision Par Ordinateur", Cluny, France (2013). 129

Recker, S., Hess-Flores, M., Duchaineau, M.A. et Joy, K.I. "Visualization of Scene Structure Uncertainty in a Multi-View Reconstruction Pipeline." Dans "VMV", pages 183–190 (2012). 121

Ren, L., Pfister, H. et Zwicker, M. "Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering". Dans *Computer Graphics Forum*, 21(3):461–470 (2002). ISSN 1467-8659. doi:10.1111/1467-8659.00606. 56

Rumpler, M., Irschara, A. et Bischof, H. "Multi-view stereo: Redundancy benefits for 3d reconstruction". Dans "Proceedings of the 35th Workshop of the Austrian Association for Pattern Recognition, AAPR/OAGM", (2011). 121

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X. et Westling, P. "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth". Dans X. Jiang, J. Hornegger et R. Koch, editeurs, "Pattern Recognition", Numéro 8753 dans Lecture Notes in Computer Science, pages 31–42. Springer International Publishing (2014). ISBN 978-3-319-11751-5 978-3-319-11752-2. 25

Scharstein, D. et Szeliski, R. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". Dans *International Journal of Computer Vision*, 47(1-3):7–42 (2002). ISSN 0920-5691, 1573-1405. doi:10.1023/A:1014573219977. 25

Schöps, T., Schönberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M. et Geiger, A. "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos". Dans , (2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)) (2017). 25

Schwarz, G. "Estimating the Dimension of a Model". Dans *The Annals of Statistics*, 6(2):461–464 (1978). ISSN 0090-5364, 2168-8966. doi:10.1214/aos/1176344136. 109

Seitz, S., Curless, B., Diebel, J., Scharstein, D. et Szeliski, R. "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms". Dans "2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition", volume 1, pages 519–528 (2006). doi:10.1109/CVPR.2006.19. 25

Shen, H., Zhang, L., Huang, B. et Li, P. "A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution". Dans *IEEE Transactions on Image Processing*, 16(2):479–490 (2007). ISSN 1057-7149. doi:10.1109/TIP.2006.888334. 79

Shum, H.Y., Chan, S.C. et Kang, S.B. *Image-Based Rendering*. Springer Science & Business Media (2008). 2, 17, 18, 51, 120

Sinha, S.N., Kopf, J., Goesele, M., Scharstein, D. et Szeliski, R. "Image-based Rendering for Scenes with Reflections". Dans *ACM Trans. Graph.*, 31(4):100:1–100:10 (2012). ISSN 0730-0301. doi:10.1145/2185520.2185596. 17, 52

Smith, S.M. et Brady, J.M. "SUSAN—A New Approach to Low Level Image Processing". Dans *International Journal of Computer Vision*, 23(1):45–78 (1997). ISSN 0920-5691, 1573-1405. doi:10.1023/A:1007963824710. 33

Snavely, N., Seitz, S.M. et Szeliski, R. "Photo Tourism: Exploring Photo Collections in 3D". Dans "ACM SIGGRAPH 2006 Papers", SIGGRAPH '06, pages 835–846. ACM, New York, NY, USA (2006a). ISBN 978-1-59593-364-5. doi:10.1145/1179352.1141964. 23, 24

Snavely, N., Seitz, S.M. et Szeliski, R. "Photo Tourism: Exploring Photo Collections in 3D". Dans "ACM SIGGRAPH 2006 Papers", SIGGRAPH '06, pages 835–846. ACM, New York, NY, USA (2006b). ISBN 1-59593-364-6. doi:10.1145/1179352.1141964. 52

Snavely, N., Seitz, S.M. et Szeliski, R. "Modeling the World from Internet Photo Collections". Dans *International Journal of Computer Vision*, 80(2):189–210 (2008). ISSN 0920-5691, 1573-1405. doi:10.1007/s11263-007-0107-3. 23, 24

Sroubek, F., Cristobal, G. et Flusser, J. "A Unified Approach to Superresolution and Multichannel Blind Deconvolution". Dans *IEEE Transactions on Image Processing*, 16(9):2322–2332 (2007). ISSN 1057-7149. doi:10.1109/TIP.2007.903256. 78

Strecha, C., von Hansen, W., Van Gool, L., Fua, P. et Thoennessen, U. "On benchmarking camera calibration and multi-view stereo for high resolution imagery". Dans "IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008", pages 1–8 (2008). doi:10.1109/CVPR.2008.4587706. 19, 24, 28, 73, 86, 88

Sulc, A., Alperovich, A., Marniok, N. et Goldluecke, B. "Reflection Separation in Light Fields based on Sparse Coding and Specular Flow". Dans "Vision, Modelling and Visualization (VMV)", (2016). 100

Sun, J., Sun, J., Xu, Z. et Shum, H.Y. "Image super-resolution using gradient profile prior". Dans "IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008", pages 1–8 (2008). doi:10.1109/CVPR.2008.4587659. 83

Takahashi, K. "Theory of Optimal View Interpolation with Depth Inaccuracy". Dans K. Daniilidis, P. Maragos et N. Paragios, editeurs, "Computer Vision – ECCV 2010", Numéro 6314 dans Lecture Notes in Computer Science, pages 340–353. Springer Berlin Heidelberg (2010). ISBN 978-3-642-15560-4 978-3-642-15561-1. doi:10.1007/978-3-642-15561-1_25. 52

Thonat, T., Shechtman, E., Paris, S. et Drettakis, G. "Multi-View Inpainting for Image-Based Scene Editing and Rendering" (2016). 135

Vaish, V., Levoy, M., Szeliski, R., Zitnick, C. et Kang, S.B. "Reconstructing Occluded Surfaces Using Synthetic Apertures: Stereo, Focus and Robust Measures". Dans "2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition", volume 2, pages 2331–2338 (2006). doi:10.1109/CVPR.2006.244. 134, 135

Vaish, V., Wilburn, B., Joshi, N. et Levoy, M. "Using plane + parallax for calibrating dense camera arrays". Dans "Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004", volume 1, pages I–2–I–9 Vol.1 (2004). doi:10.1109/CVPR.2004.1315006. 134

Vanhoey, K., Sauvage, B., Génevaux, O., Larue, F. et Dischler, J.M. "Robust Fitting on Poorly Sampled Data for Surface Light Field Rendering and Image Relighting". Dans *Computer Graphics Forum*, 32(6):101–112 (2013). ISSN 1467-8659. doi:10.1111/cgf.12073. 84

Venkataraman, K., Lelescu, D., Duparré, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R. et Nayar, S. "PiCam: An Ultra-thin High Performance Monolithic Camera Array". Dans *ACM Trans. Graph.*, 32(6):166:1–166:13 (2013). ISSN 0730-0301. doi:10.1145/2508363.2508390. 10

Waechter, M., Beljan, M., Fuhrmann, S., Moehrle, N., Kopf, J. et Goesele, M. "Virtual Rephotography: Novel View Prediction Error for 3D Reconstruction". Dans *arXiv:1601.06950 [cs]* (2016). 120

Wang, Z., Bovik, A.C., Sheikh, H.R. et Simoncelli, E.P. "Image quality assessment: From error visibility to structural similarity". Dans *IEEE Transactions on Image Processing*, 13(4):600–612 (2004). ISSN 1057-7149. doi:10.1109/TIP.2003.819861. 26, 88, 89

Wanner, S. et Goldluecke, B. "Globally consistent depth labeling of 4D light fields". Dans "2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", pages 41–48 (2012a). doi:10.1109/CVPR.2012.6247656. 12, 88

Wanner, S. et Goldluecke, B. "Spatial and Angular Variational Super-Resolution of 4D Light Fields". Dans A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato et C. Schmid, editeurs, "Computer Vision – ECCV 2012", Numéro 7576 dans Lecture Notes in Computer Science, pages 608–621. Springer Berlin Heidelberg (2012b). ISBN 978-3-642-33714-7 978-3-642-33715-4. 15, 60, 78, 79, 80, 88, 89, 93

Wanner, S. et Goldluecke, B. "Variational Light Field Analysis for Disparity Estimation and Super-Resolution". Dans *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):606–619 (2014). ISSN 0162-8828. doi:10.1109/TPAMI.2013.147. 87, 97, 99, 100

Wanner, Sven, Meister, Stephan et Goldluecke, Bastian. "Datasets and Benchmarks for Densely Sampled 4D Light Fields". Dans (2013). doi:10.2312/PE.VMV.VMV13.225-226. 25, 88, 89

Westover, L. "Footprint Evaluation for Volume Rendering". Dans "Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '90, pages 367–376. ACM, New York, NY, USA (1990). ISBN 0-89791-344-2. doi:10.1145/97879.97919. 51

Wetzstein, G., Roodnick, D., Heidrich, W. et Raskar, R. "Refractive shape from light field distortion". Dans "2011 International Conference on Computer Vision", pages 1180–1186 (2011). doi:10.1109/ICCV.2011.6126367. 10, 100

Weyrich, T., Heinzle, S., Aila, T., Fasnacht, D.B., Oetiker, S., Botsch, M., Flaig, C., Mall, S., Rohrer, K., Felber, N., Kaeslin, H. et Gross, M. "A Hardware Architecture for Surface Splatting". Dans "ACM SIGGRAPH 2007 Papers", SIGGRAPH '07. ACM, New York, NY, USA (2007). doi:10.1145/1275808.1276490. 56

Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M. et Levoy, M. "High Performance Imaging Using Large Camera Arrays". Dans "ACM SIGGRAPH 2005 Papers", SIGGRAPH '05, pages 765–776. ACM, New York, NY, USA (2005). doi:10.1145/1186822.1073259. 9, 25, 100, 113

Wolff, K., Kim, C., Zimmer, H., Schroers, C., Botsch, M., Sorkine-Hornung, O. et Sorkine-Hornung, A. "Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction". Dans "2016 Fourth International Conference on 3D Vision (3DV)", pages 118–127 (2016). doi:10.1109/3DV.2016.20. 47

Xiao, Z., Wang, Q., Si, L. et Zhou, G. "Reconstructing scene depth and appearance behind foreground occlusion using camera array". Dans "2014 IEEE

International Conference on Image Processing (ICIP)", pages 41–45 (2014). doi: 10.1109/ICIP.2014.7025007. 135, 137

Zhou, P., Yu, L. et Zhong, G. "The non-Lambertian reflection in plenoptic sampling". Dans "2013 IEEE International Conference on Image Processing", pages 2154–2157 (2013). doi:10.1109/ICIP.2013.6738444. 99

Zhou, P., Yu, L. et Pak, C. "The Spectrum Broadening in the Plenoptic Function". Dans "Proceedings of International Conference on Internet Multimedia Computing and Service", ICIMCS '14, pages 130:130–130:135. ACM, New York, NY, USA (2014). ISBN 978-1-4503-2810-4. doi:10.1145/2632856.2632933. 99

Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S. et Szeliski, R. "High-quality Video View Interpolation Using a Layered Representation". Dans "ACM SIGGRAPH 2004 Papers", SIGGRAPH '04, pages 600–608. ACM, New York, NY, USA (2004). doi:10.1145/1186562.1015766. 52

Zomet, A., Levin, A., Peleg, S. et Weiss, Y. "Seamless image stitching by minimizing false edges". Dans *IEEE Transactions on Image Processing*, 15(4):969–977 (2006). ISSN 1057-7149. doi:10.1109/TIP.2005.863958. 64, 85

Zwicker, M., Pfister, H., van Baar, J. et Gross, M. "Surface Splatting". Dans "Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques", SIGGRAPH '01, pages 371–378. ACM, New York, NY, USA (2001). ISBN 978-1-58113-374-5. doi:10.1145/383259.383300. 56