

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ MOHAMED KHIDER – BISKRA  
FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE  
DÉPARTEMENT DE L'INFORMATIQUE



## THÈSE

Présentée pour l'obtention du diplôme de  
Doctorat LMD en Informatique

**Option :** Techniques de l'Image et de l'Intelligence Artificielle

---

# Rendu HDR interactif à partir de photographies non structurées

---

Présentée par : **Lechelek Loubna**

Soutenue le : **24/11/2019**

Devant le jury composé de :

CHERIF Foudil	Professeur	Université de Biskra	Président
BABAHENINI <i>M<sup>ed</sup></i> Chaouki	Professeur	Université de Biskra	Rapporteur
NINI Brahim	Professeur	Université d'Oum El Bouaghi	Examineur
KOUAHLA <i>M<sup>ed</sup></i> Nadjib	MCA	Université de Guelma	Examineur
DJEFFAL Abdelhamid	MCA	Université de Biskra	Examineur

---

## Résumé

Nous présentons une nouvelle méthode de rendu HDR interactif basé-image qui permet à partir de plusieurs séries de photographies non structurées de produire des images HDR. Les séries de photographies sont prises avec des conditions d'éclairage fixes et des niveaux d'expositions différents. La méthode s'appuie sur un nuage de points utilisé à la manière d'un proxy géométrique et sur l'ensemble de photographies d'origines, étalonnées en utilisant des méthodes de type structures à partir de mouvement. Dans un premier temps, nous construisons une carte de profondeur pour chaque nouveau point de vue en s'appuyant sur un algorithme de rendu basé point. La couleur de chaque pixel de la nouvelle vue est estimée par projection inverse sur les photographies d'origines, tandis que les problèmes d'occultations et de contributions de chacune des photographies sont gérés automatiquement en utilisant un modèle de cohérence épipolaire. Nous ré-itérons ce processus pour chaque nouveau point de vue et pour chaque exposition pour produire un ensemble d'images LDR bien alignées que nous fusionnons par la suite afin de produire l'image HDR. Contrairement aux approches précédentes traitant la reconstruction d'images HDR à partir de photos non alignées, notre système est plus robuste aux occultations. Il permet également une navigation virtuelle libre avec une fréquence d'affichage élevée.

**Mots-clés : rendu basé image, proxy basé point, rendu HDR, rendu interactif**

---

## Abstract

We present an interactive High Dynamic Range (HDR) image-based rendering system dedicated to manually acquired photographs. The series of photographs are taken with fixed lighting conditions and different exposure levels. Our system only relies on a point-based geometric proxy and the original photographs, calibrated using a standard structure from-motion process. First, a depth map is estimated for each new rendered viewpoint using a point based rendering algorithm. Second, pixel values are reconstructed from the original photographs, using a blending model that also handles occlusion. Our system can be used for producing HDR images from several series of unaligned photographs with different exposures. Unlike previous approaches dealing with reconstructing HDR images from non-aligned photos, our system is more robust to occlusions. It also allows free viewpoint navigation with a high refresh rate.

**Keywords :** Image-based rendering, Point-based proxy, high-dynamic range images, interactive rendering

## ملخص

في هذه الأطروحة، نقدم طريقة إظهار تفاعلية جديدة التي تسمح بإنتاج صور HDR (صور ذات مجال ديناميكي عالي) ، إنطلاقاً من مجموعات متعددة من الصور LDR (صور ذات مجال ديناميكي منخفض) غير منظمة. يتم التقاط سلسلة الصور في ظروف إضاءة ثابتة ومستويات عرض مختلفة. تعتمد الطريقة على مجموعة من النقاط التي تستخدم كمعلومة هندسية وعلى مجموعة من الصور الأصلية ، يتم معايرتها باستخدام طرق من صنف البنية إنطلاقاً من الحركة. أولاً ، نقوم بحساب بطاقة عمق لكل وجهة نظر جديدة استناداً إلى خوارزمية تجسيد قائمة على النقاط. يتم تقدير لون كل بكسل صورة جديدة من خلال الإسقاط العكسي على الصور الأصلية ، في حين أن مشكلات أخذ بعين الاعتبار الأجزاء المختلفة ومساهمة كل صورة من الصور يتم إدارتها تلقائياً باستخدام نموذج تناسق قطبي. نعيد تكرار هذه العملية لكل وجهة نظر جديدة ولكل عرض لإنتاج مجموعة من صور LDR المتوافقة و التي ندمجها بعد ذلك لإنتاج صورة HDR. على عكس الأساليب السابقة التي تتناول إنتاج صور HDR من صور LDR غير منظمة ، فإن نظامنا يعتبر أكثر متانة كما أنه يسمح بالتصفح الظاهري الحر مع ارتفاع وتيرة العرض.

---

## Remerciements

A l'issue de la rédaction de cette recherche, je suis convaincue que la thèse est loin d'être un travail solitaire. En effet, je n'aurais jamais pu réaliser ce travail doctoral sans le soutien d'un grand nombre de personnes dont la générosité, la bonne humeur, et l'intérêt manifesté à l'égard de ma recherche m'ont permis de progresser dans cette phase délicate de "*chercheur*".

Je souhaiterais tout d'abord, non seulement remercier, mais surtout dédier cette thèse à mes parents. Sans tous les sacrifices qu'ils ont consentis, leur soutien et leur amour, rien de tout cela n'aurait été possible.

Je tiens à remercier mon directeur de thèse, Mr.BABAHENINI Mohamed Chaouki pour la confiance qu'il m'a accordé en acceptant d'encadrer ce travail de doctorat, pour ses multiples conseils, sa patience et son soutien administratif.

Je voudrais également remercier mes encadrants : Mr. MENEVEAUX Daniel et Mr. Ribardière Mickael pour leur engagement et leur aide dans ce projet. Ils m'ont transmis leur savoir pour mener à bien cette thèse. Merci à Mickael pour son aide sans faille et pour ses apports précieux tout le long de cette thèse. Un grand merci à Daniel pour son accompagnement, ses encouragements quotidiens et ses nombreux retours. Merci aussi pour son aide sans arrêt et pour les nombreuses discussions scientifiques que nous avons eu. J'aimerais également lui dire à quel point j'ai apprécié sa grande disponibilité et son respect sans faille des délais serrés de relecture des documents que je lui ai adressé. Enfin, j'ai été extrêmement sensible à ses qualités humaines d'écoute et de compréhension tout au long de ce travail.

Je n'y manquerais pas à remercier également ROMUALD Perrot pour ses apports précieux à chaque fois que j'ai sollicité son aide ainsi que pour ses nombreuses connaissances techniques qui m'ont permis de progresser et ont répondu à plusieurs de mes préoccupations.

Je remercie également les membres de jury pour l'honneur qu'ils m'ont fait en acceptant de juger notre travail.

Au passage, je remercie mes frères, sœur et belle-sœur qui m'ont soutenu et m'ont appuyé moralement.

Enfin, merci à mes collègues et mes amis pour les moments inoubliables qu'on a passé ensemble.

---

## Publications

**LECHLEK, L.**, MENEVEAUX, D., RIBARDIÈRE, M., PERROT, R. et BAHENINI, C. (2019). Interactive HDR Image-Based Rendering from Unstructured LDR Photographs. *Computer and Graphics*.

---

## Communications

**Workshop** : état de l'art sur les techniques de rendu. Second workshop on images, Graphiques et vie artificielle. IGVA'2014, Université de Biskra.

**Journées doctoriales** : informatique théorique et appliquée (JDITA'2018), Biskra

# Table des matières

<b>Table des figures</b>	<b>xi</b>
<b>Liste des tableaux</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivations . . . . .	3
1.2 Contributions . . . . .	5
1.3 Plan de la thèse . . . . .	6
<b>2 État de l’art</b>	<b>7</b>
2.1 Rendu basé-image . . . . .	8
2.1.1 Interpolation d’images sans modèle 3D . . . . .	8
2.1.2 Rendu basé image avec géométrie explicite . . . . .	10
2.1.3 Discussion . . . . .	13
2.2 Reconstruction des images HDR à partir de photos prises à main levée	14
2.3 Rendu basé point . . . . .	16
2.3.1 Splatting . . . . .	16
2.3.2 Rendu par reconstruction dans l’espace image . . . . .	19
2.3.3 Discussion . . . . .	21
2.4 Reconstruction 3D . . . . .	22
2.4.1 Stéréo multi-vues . . . . .	23
2.4.2 Reconstruction de la surface . . . . .	23
2.4.3 Discussion . . . . .	25
<b>3 Rendu basé image interactif avec proxy basé point</b>	<b>27</b>
3.1 Présentation générale du système . . . . .	28
3.2 Modélisation du système . . . . .	30
3.2.1 Rendu des points . . . . .	31
3.2.2 Gestion des occultations . . . . .	36
3.2.3 Cohérence épipolaire . . . . .	39
3.3 Résultats . . . . .	43
3.4 Limites . . . . .	49
<b>4 Reconstruction d’images à haute gamme de dynamique (HDR)</b>	<b>51</b>
4.1 Présentation du système . . . . .	51
4.2 Méthode pour fusionner les images synthétisées en une image HDR .	54
4.3 Reproduction de tons " <i>tone mapping</i> " . . . . .	56

---

4.4	Les stratégies de reconstruction des images HDR . . . . .	57
4.5	Résultats . . . . .	58
4.6	Conclusion . . . . .	61
<b>5</b>	<b>Conclusion</b>	<b>63</b>
<b>A</b>	<b>Transformation des matrices de la caméra de l'espace réel vers le système de coordonnées Opengl</b>	<b>65</b>
	<b>Bibliographie</b>	<b>69</b>

# Table des figures

2.1	Déformation de vue [Seitz et Dyer, 1996] . . . . .	9
2.2	Rendu avec reconstruction multi-modale [Hedman <i>et al.</i> , 2016] . . . . .	11
2.3	Synthèse de profondeur [Chaurasia <i>et al.</i> , 2013] . . . . .	13
2.4	Reconstruction de l'image HDR à partir de photos tenue à main levée [Tomaszewska et Mantiuk, 2007] . . . . .	15
2.5	Reconstruction de l'image HDR par la méthode de Zimmer [Zimmer <i>et al.</i> , 2011] . . . . .	15
2.6	Figure de [Botsch <i>et al.</i> , 2005] : Rendu par splatting sur GPU . . . . .	17
2.7	Test d'appartenance au splat par lancer de rayon; . . . . .	18
2.8	Pipeline utilisé par [Rosenthal et Linsen, 2008] . . . . .	21
2.9	Reconstruction d'un maillage [Chaurasia, 2014] . . . . .	24
2.10	Reconstruction 3D de la boule de neige en utilisant openMVG et PMVS2	25
3.1	Architecture logicielle de notre système de rendu à base d'images interactif . . . . .	28
3.2	Points de vues et nuage de points . . . . .	30
3.3	Proxy géométrique et configurations possibles . . . . .	31
3.4	L'algorithme proposé par [Marroquim <i>et al.</i> , 2007] . . . . .	33
3.5	Corrections de la profondeur . . . . .	36
3.6	Gestion des occultations pour un nouveau point de vue $C$ . . . . .	38
3.7	L'effet de la prise en compte des occultations sur les nouvelles vues .	39
3.8	Cohérence épipolaire . . . . .	40
3.9	Exemples de scènes affichées avec notre système de rendu basé image interactif. . . . .	43
3.10	Nouvelles images synthétisées par notre système de navigation inter- actif, basé sur 22 photographies. . . . .	45
3.11	Les effets dépendant du point de vue . . . . .	46
3.12	Gain de notre méthode en terme de qualité . . . . .	47
3.13	Fiabilité des nuages de points par rapport aux maillages géométriques	48
3.14	Chillac visualisée en utilisant des photographies à différentes résolu- tions, avec l'occupation mémoire correspondante. . . . .	49
4.1	Notre système de reconstruction d'images HDR . . . . .	52
4.2	Reconstruction d'une image HDR à partir de 3 séries de photographies non structurées, ayant Chacune un niveau d'exposition fixe . . . . .	53

---

4.3	Trois images LDR avec des niveaux d'expositions différents produites par notre système de visualisation pour les deux scènes: statue et coquillages . . . . .	54
4.4	Comparaison entre deux stratégies de rendu HDR . . . . .	58
4.5	Rendu HDR interactif avec différents paramètres de tone-mapping . .	59
4.6	Trois échantillons de série de photos avec différentes expositions . . .	61
4.7	Trois nouvelles expositions synthétisées pour un même point de vue en utilisant notre système de visualisation . . . . .	62

# Liste des tableaux

3.1	Description des scènes, avec mémoire requise. . . . .	43
3.2	Temps de calcul pour une résolution de $512 \times 512$ . . . . .	44
4.1	Caractéristiques des scènes. . . . .	59
4.2	Temps de calcul des différentes parties de rendu HDR. . . . .	60



# Chapitre 1

## Introduction

L'un des objectifs des recherches menées jusqu'à présent en synthèse d'images est le rendu réaliste. La synthèse d'images est le procédé qui cherche à traiter un problème bien défini : étant donné un modèle géométrique précis décrivant l'ensemble des objets composant une scène, les différentes propriétés sur les matériaux qui les composent, l'éclairage.. etc, créer une image aussi proche que possible de la version réelle de la scène décrite. De nombreuses recherches sur la création de scènes très réalistes ont été effectuées, en utilisant des méthodes de rendu et de simulations ou en utilisant des images.

Les méthodes de rendu et de simulations nécessitent de nombreuses opérations préalables pour définir la géométrie de la scène, le positionnement des sources de lumière ou encore le paramétrage des matériaux avec leurs textures et la manière dont ils réfléchissent la lumière. Une autre difficulté concerne les calculs de simulation d'éclairages nécessaires pour approcher de manière précise les échanges lumineux à l'intérieur de l'environnement. Malheureusement ces calculs consistent à résoudre une équation intégrale du premier ordre, récursive infinie.

Les photographies peuvent être une alternative intéressante pour simplifier les traitements, et leur utilisation est arrivée très rapidement, en commençant par le placage de textures à partir de photos [Williams, 1983, Perlin, 1985]. Depuis deux décennies, elles ont été utilisées de manière plus avancée pour diverses applications, et en particulier pour la reconstruction automatique de la géométrie. Cette intuition de base associée à l'émergence des appareils photos a conduit à la conception de nouvelles

---

approches dites : "*approches à base d'images*" [Shum *et al.*, 2008], qui prennent ces dernières années une dimension significative dans le contexte du rendu réaliste.

Un grand nombre de méthodes, que l'on peut classer dans les approches à base d'images : les méthodes de ré-éclairage à base d'images et les méthodes de rendu basé images. Les premières ont pour objectifs de changer l'apparence des objets acquis tout en maintenant le point de vue [Hawkins *et al.*, 2005, Einarsson *et al.*, 2006]. Les secondes cherchent à changer le point de vue tout en maintenant l'apparence [Debevec *et al.*, 1996, Chaurasia *et al.*, 2013, Hedman *et al.*, 2018]. Ce mémoire de thèse se place dans le contexte de rendu basé image et notre objectif est de proposer une nouvelle méthode de visualisation interactive basée images, peu coûteuse en ressources de calcul, facile à utiliser (visualisation sur navigateur web) et qui nécessite peu de photographies en entrée prises à main levée.

Conceptuellement, il est possible d'obtenir un rendu physiquement correct d'un environnement existant à condition de pouvoir acquérir l'ensemble de ses caractéristiques à la fois géométriques, photométriques et radiométriques. Malheureusement, à partir de photographies, la reconstruction de maillages 3D précis reste un défi difficile à relever malgré les nombreuses avancées de ces dernières années en vision par ordinateur [Berger *et al.*, 2014]. Les imprécisions issues d'un modèle géométrique causent systématiquement des artefacts visuels dans les vues synthétisées [Vangorp *et al.*, 2011]. Notre objectif est de proposer un système de navigation bénéficiant pleinement du niveau de détail offert par les photographies, sans aucune reconstruction de maillage géométrique.

Nous proposons également un système de production automatique d'images à haute gamme de dynamique (ou HDR pour High Dynamic Range). Ces images ont l'avantage de stocker tous les détails de luminance d'un environnement quelle que soit l'intensité de son éclairage. Cependant, leur reconstruction nécessite plusieurs images prises d'un même point de vue [Mann et Picard, 1994, Debevec et Malik, 1997]. Lorsque les photographies sont prises à main levée, tout mauvais alignement conduit souvent à du flou ou un effet fantôme (dédoublage dans l'image HDR produite). Les travaux précédents traitant la reconstruction d'images HDR à partir de photos non structurées se sont focalisés sur des changements légers entre les exposi-

---

tions, en ignorant les occultations complexes et les gros mouvements de la caméra [Tomaszewska et Mantiuk, 2007, Zimmer *et al.*, 2011, Sen *et al.*, 2012]. Notre intérêt est de fournir un système permet de les reconstruire automatiquement et cela à partir de plusieurs séries d’images LDR, moins coûteuses et plus rapides, tout autour de l’objet et sans se préoccuper de la prise de vue.

## 1.1 Motivations

Les récentes avancées en vision par ordinateur ont permis d’automatiser le processus d’estimation des paramètres intrinsèques et extrinsèques de la caméra pour un ensemble de photographies prises à main levée. Ce processus d’étalonnage est très important car il permet à la fois de déterminer le positionnement des appareils photos, et de retrouver des informations de profondeurs pour produire par exemple des nuages de points correspondant à la surface des objets visualisés [Snavely *et al.*, 2006, Furukawa et Ponce, 2010]. L’intégralité de la chaîne de traitements est automatique, et plusieurs méthodes permettent ainsi de visualiser des monuments pris en photos par de nombreux touristes [Snavely *et al.*, 2006], ou après une étape de reconstruction géométrique dont l’objectif est de produire un maillage [Kazhdan *et al.*, 2006, Fuhrmann et Goesele, 2014].

Malheureusement, la reconstruction d’un maillage précis à partir d’un nuage de points reste encore un sujet de recherche actif car la triangulation produite n’est souvent pas alignée avec les arêtes ou les courbures des objets. Le processus de triangulation crée très souvent des surfaces qui n’existent pas sur l’objet réel (création de dôme autour de l’objet, remplissage de trous). Ce manque de précision conduit à des artefacts compliquant par conséquent l’exactitude du rendu des scènes. En outre, le processus de rendu peut également être affecté par la complexité de la scène dès que celle ci dépasse un certain seuil (plusieurs millions de triangles par maillage). Cela réduit de façon très importante les performances d’affichage, et requiert une place mémoire importante.

Face à ces problèmes récurrents, de nombreuses méthodes de rendu basé image intègrent l’incertitude des maillages dans le rendu pour réduire les artefacts dans

---

les vues synthétisées [Eisemann *et al.*, 2008, Pujades *et al.*, 2014, Nieto *et al.*, 2016]. D'autres proposent d'utiliser des structures de données dédiées pour alléger les coûts de calcul [Hedman *et al.*, 2016] ou même de remplacer carrément les maillages par des nuages de points épars associés à des approximations basées sur l'image [Chaurasia *et al.*, 2011, Chaurasia *et al.*, 2013]. Néanmoins, la complexité temporelle, la nécessité de faire intervenir l'utilisateur et la complexité des traitements rendent difficile l'emploi de ces approches pour la visualisation des environnements à partir de photographies.

Contrairement aux approches existantes, nous désirons dans cette thèse éviter le processus complexe de reconstruction d'un maillage géométrique, en exploitant directement les nuages de points produits par les méthodes de stéréo multi-vue comme proxy géométrique, dans un système de navigation interactif basé image. Nous utilisons un algorithme de rendu basé point pour construire pour chacun des points de vue des cartes de profondeurs, utilisées pour gérer les problèmes d'occultations et éviter les erreurs de parallaxe lors de la visualisation.

Nous désirons également fournir un système de production automatique d'images à haute gamme de dynamique (HDR) à partir de plusieurs séries de photographies non structurées, ayant chacune un niveau d'exposition fixe. Toutes les méthodes de rendu à base d'images utilisent des photos prises par des capteurs classiques (appareil photo ou caméra) dans le processus de rendu. Or, ces dispositifs d'acquisition ne permettent pas de fournir une image contenant toute la dynamique lumineuse d'un environnement en une seule prise de vue. Les images obtenues généralement présentent des zones sous-exposées ou sur-exposées pour lesquelles la quantité de la lumière véritablement présente dans l'environnement n'est pas définie. Dans cette thèse, l'un des facteurs les plus importants qui nous distingue des approches précédentes est l'emploi des images HDR dans le processus de rendu. L'utilisation de ces images pour le rendu basé image va permettre d'augmenter le réalisme des scènes visualisées et d'apporter une forte sensation d'immersion dans l'environnement lors de la navigation.

---

## 1.2 Contributions

**Un système de navigation interactif** Nous proposons une nouvelle méthode de rendu basé image qui permet à partir de photos non calibrées recouvrant la surface de l'objet acquis, de naviguer librement et générer des points de vue quelconques. La plupart des méthodes précédentes utilisent un maillage géométrique. Nous avons choisi d'utiliser directement le nuage de points car il est moins coûteux en mémoire qu'un maillage, il est beaucoup plus rapide à afficher et il correspond à des primitives géométriques plus fiables que les maillages (ils ne comportent pas d'arête erronée).

**Amélioration des silhouettes** La projection des points sur l'espace image sous la forme de simples pixels causent des trous dans l'image produite. Nous avons utilisé un algorithme de rendu basé point pour combler les éventuels trous et définir une valeur (profondeur, normale) pour chaque pixel de l'image. Le choix a été mis sur l'algorithme de Marroquim [Marroquim *et al.*, 2007] pour sa rapidité par rapport aux autres approches de rendu par splatting. Or, La reconstruction proprement dite étant effectuée exclusivement dans l'espace image, elle produit des artefacts de clignotement sur la silhouette de l'objet. De plus, l'algorithme échoue lorsque le point de vue est trop proche de la surface des objets. Nous proposons des passes de rendu supplémentaires permettant de corriger ces défauts, sans trop impacter les performances du système.

**Un système de pondération des photographies** Nous présentons un modèle épipolaire complet pour gérer la sélection des photographies pertinentes et la fusion des pixels au cours du processus de rendu.

**Un système de production d'images HDR** La contribution principale de cette thèse est de proposer une méthode qui permet de produire des images HDR à partir d'un jeu de données non structuré (les expositions sont prises par des caméras tenues à main levée), capturé à main levée par des caméras standards. Ce jeu de données correspond à plusieurs séries de photographies non HDR, prises avec des conditions d'illumination fixes et ayant chacune un niveau d'exposition constant. Les images

---

HDR produites sont ensuite utilisées par le système de visualisation pour synthétiser de nouvelles vues HDR, avec la possibilité de choisir la luminosité de l’environnement lors de la navigation.

## 1.3 Plan de la thèse

Le travail présenté dans ce mémoire est organisé de la manière suivante :

- Le chapitre 2 propose une étude bibliographique sur les sujets de rendu à base d’images, de la stéréo multi-vues, de rendu basé point et de construction d’images HDR à partir de photos prises à main levée.
- Le chapitre 3 présente l’architecture générale de notre méthode, les outils utilisés lors du pré-traitement ainsi que notre nouvelle approche de rendu basé image. Les passes de correction de la profondeur sont également détaillées dans cette partie.
- Le chapitre 4 expose notre nouveau système de création d’images HDR à partir de photographies non structurées, en s’appuyant sur notre méthode de rendu basé image.
- Après la conclusion et les perspectives que nous présentons dans le chapitre 5, un annexe (A) décrit la façon de transformer la matrice de projection de l’espace réel vers l’espace opengl (webgl).

# Chapitre 2

## État de l'art

De nombreux auteurs se sont intéressés à proposer des méthodes de rendu à base d'images pour la visualisation d'objets réels, d'effets complexes, ou encore pour naviguer virtuellement dans des environnements acquis. Les premières approches utilisent uniquement un ensemble de photographies pour représenter une scène. Des mécanismes d'interpolations entre les différents points de vue permettent ensuite de synthétiser de nouvelles images. Ces approches ont été proposées après la définition de la fonction plénoptique par Adelson et Bergen [Adelson *et al.*, 1991]. Le rendu d'un nouveau point de vue nécessite un échantillonnage dense de cette fonction, ce qui entraîne un stockage important de photographies. Pour cela, de nombreux auteurs se sont dirigés vers l'utilisation d'une représentation géométrique, car elle permet à la fois de gérer les problèmes d'occultations, et de synthétiser de nouvelles vues uniquement à partir de quelques dizaines d'images. Depuis, le rendu basé image a été associé à des méthodes de stéréo multi-vues [Goesele *et al.*, 2007, Furukawa et Ponce, 2010] et de reconstruction de surfaces [Kazhdan *et al.*, 2006, Fuhrmann et Goesele, 2014] pour récupérer de l'information géométrique sous forme de maillage, nuage de points ou cartes de profondeur.

Théoriquement, un rendu de haute qualité nécessite une reconstruction 3D précise de la scène. Malheureusement, la reconstruction d'un maillage fidèle reste un défi. Dans de nombreux cas, la triangulation génère des arêtes indésirables (parfois même perpendiculaires sur l'objet) et le modèle 3D résultant est maillé par des volumes supplémentaires, produisant des artefacts qui se répercutent par la suite sur l'image

---

synthétisée. Pour ces raisons, nous avons choisi d’exploiter directement les nuages de points produits par les méthodes de stéréo multi-vue dans une nouvelle méthode de navigation basée images dont les photos sont prises à main levée. Le nuage de points étant de nature épars, nous utilisons un algorithme de rendu basé point pour estimer une profondeur pour chaque pixel de l’image. Ces valeurs de profondeurs sont utilisées à la manière d’un proxy géométrique pour gérer les occultations lors de la synthèse de la nouvelle vue. Nous exploitons ce système par la suite pour reconstruire des images à haute gamme de dynamique à partir de plusieurs séries de photos prises à main levée. A ces fins, nous sommes amenés à utiliser des travaux de la stéréo multi-vues et du rendu basé point. Dans ce chapitre, nous donnons un aperçu des travaux existants dans chacun de ces domaines. Nous allons également se positionner par rapport à quelques travaux de rendu basé-image et de reconstruction d’images HDR à partir de photographies prises à main levée .

## 2.1 Rendu basé-image

Les méthodes de rendu basé-image sont devenues légion, après le développement de la modélisation plénoptique (*plenoptic modeling*) en 1995 [McMillan et Bishop, 1995]. Shum *et al.* [Shum *et al.*, 2008] choisissent de les classer selon le type de modèle géométrique qu’elles utilisent (géométrie implicite ou explicite). Dans cette thèse, nous classifions ces méthodes selon les données utilisées en entrée, en soulignant les différences par rapport à notre travail et nous discutons des contraintes de chacune d’elles.

### 2.1.1 Interpolation d’images sans modèle 3D

De nombreuses approches permettent de visualiser un objet ou un environnement à partir de deux ou plusieurs photographies à l’aide d’interpolations entre les points de vue, sans reconstruire explicitement un modèle 3D. Chen et Williams [Chen et Williams, 1993] construisent une carte de correspondance pour chaque paire d’images et synthétisent de nouvelles images par interpolation entre les deux cartes de correspondances les plus proches du nouveau point de vue.

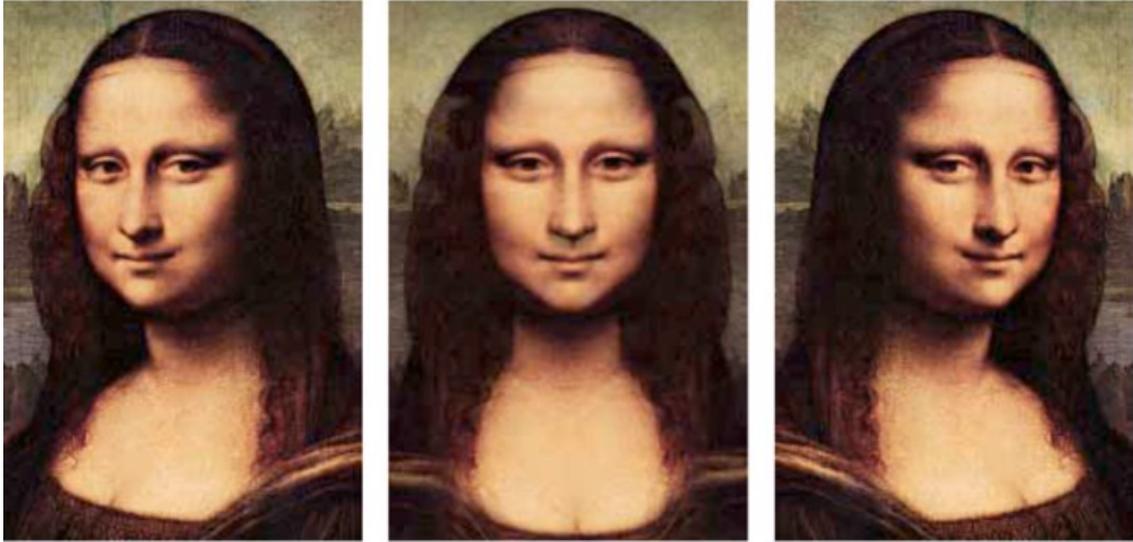


FIGURE 2.1 – Déformation de vue [Seitz et Dyer, 1996]

La vue centrale est générée par interpolation des vues sur les bords. Notez que la navigation est contrainte à un faible nombre de points de vue, ce qui constitue l’une des grandes limites de ces méthodes.

La méthode développée par Seitz et Dyer [Seitz et Dyer, 1996] permet de synthétiser de nouvelles images sur la droite passant par les centres optiques des deux vues originales. Les light fields [Levoy et Hanrahan, 1996] sont une représentation de la fonction plénoptique en 4D à l’aide de paires de plans parallèles (appelé les slabs). La visualisation d’un objet à partir du point de vue désiré s’effectue en calculant les coordonnées du rayon intersectant les deux plans, et en faisant l’interpolation des valeurs aux voisins. L’huillier et Quan [Lhuillier et Quan, 1999] utilisent un algorithme d’appariement quasi-dense pour associer les points et les régions entre les images et génèrent de nouvelles vues par interpolation effectuée entre les régions appariées. Stich et al. [Stich *et al.*, 2011] proposent une méthode d’interpolation perceptuellement correcte en divisant les images en entrée en régions homogènes. Ils cherchent ensuite des correspondances entre ces régions et calculent des matrices d’homographies locales. Les régions sont projetées et mélangées pour estimer la nouvelle vue sans reconstruire explicitement un modèle géométrique. Dans ces approches, la navigation reste contrainte à un faible nombre de points de vue pour conserver le réalisme des photographies (elle se limite à interpoler entre deux vues originales (figure 2.1)) et elle ne s’attelle pas à la configuration générique, dans laquelle les points de vues originales sont non structurés, ce que nous proposons de faire. De plus,

---

L'absence d'un modèle 3D, rend ces méthodes moins robustes face aux problèmes d'occultations.

### 2.1.2 Rendu basé image avec géométrie explicite

Les approches récentes utilisent une représentation géométrique pour synthétiser de nouvelles images à partir de n'importe quel point de vue et créer des systèmes de rendu permettant de naviguer librement dans des environnements acquis de manière fluide et réaliste. Les lumigraphes [Gortler *et al.*, 1996] était la première approche à proposer l'utilisation d'un maillage géométrique rudimentaire pour évaluer la profondeur lors de la génération de nouvelles vues. Cela a permis de réduire considérablement les problèmes de parallaxe et à améliorer fortement la qualité de rendu. Depuis lors, plusieurs méthodes de rendu basé image ont été développées avec différentes sortes de proxy dont on peut citer : les modèles architecturaux [Debevec *et al.*, 1996], les plans 3D [Heigl *et al.*, 1999] et les imposteurs [Popescu *et al.*, 2006, Sillion *et al.*, 1997].

L'arrivée des méthodes de structure à partir de mouvement [Snavely *et al.*, 2006, Snavely *et al.*, 2008] et de la stéréo multi-vue [Goesele *et al.*, 2007] ont permis de substituer l'utilisation des mires et des proxy modélisés manuellement par des nuages de points et des maillages reconstruits d'une manière quasi-automatique à partir d'un ensemble de photographies. Suite à ces avancées, de nombreuses applications de rendu basé image permettent aujourd'hui de naviguer librement dans des scènes intérieurs et extérieurs plus au moins complexes à partir de quelques dizaines d'images prises à main levée. Dans ce qui suit, nous exposons quelque travaux récents qui sont adaptés à la synthèse de point de vue libre "free viewpoint".

**Les méthodes de rendu basé image utilisant un maillage** Certaines approches emploient un maillage comme un proxy géométrique, utilisé pour établir des correspondances entre les photographies et la nouvelle vue. Par exemple, Buehler *et al.* [Gortler *et al.*, 1996] proposent de généraliser le concept des lumigraphes en exploitant directement les photographies étalonnées et un modèle simplifié de l'objet. Debevec *et al.* [Debevec *et al.*, 1996, Debevec *et al.*, 1998] proposent des textures dépendant du point de vue.

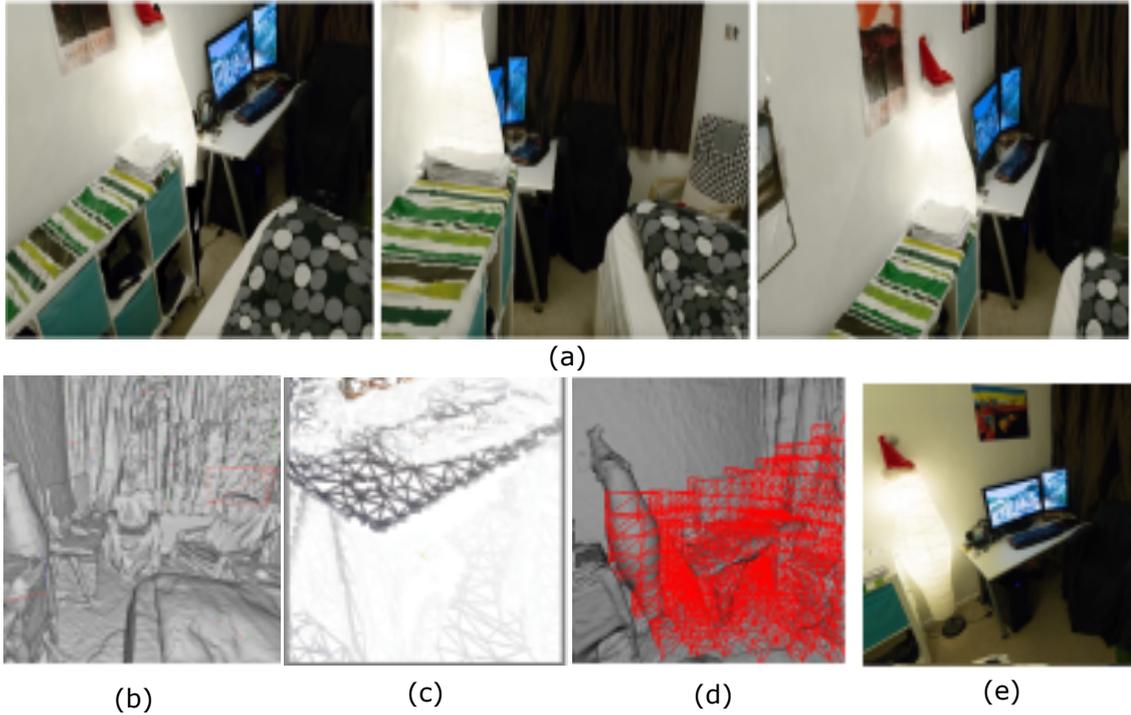


FIGURE 2.2 – Rendu avec reconstruction multi-modale [Hedman *et al.*, 2016]

(a) quelques images d'origines; (b) maillage globale reconstruit via un capteur de profondeur; (c) maillage simplifié généré par des méthodes de stéréo multi-vue; (d) l'emploi de structure de donnée "tiles" : la partie en rouge correspond au maillage visible par l'une des images pertinentes, ce maillage est valide pour le rendu de la nouvelle vue; (e) nouvelle vue résultante.

Eisemann *et al.* [Eisemann *et al.*, 2008] introduisent une passe de correction en se basant sur le flux optique pour réduire les artefacts dûs à la l'incertitude de la géométrie. Aliaga *et al.* [Aliaga *et al.*, 2003] proposent une nouvelle approche de navigation basé image dédiée aux scènes intérieurs dans laquelle ils utilisent un grand nombre de photographies pour compenser l'incertitude de la géométrie. Pujades *et al.* [Pujades *et al.*, 2014] présentent une approche différente pour synthétiser une nouvelle image à partir d'un ensemble de photos non structurées. Les images en entrée sont utilisées pour estimer un modèle 3D et la nouvelle vue est calculée en minimisant une fonction d'énergie qui prend en compte l'incertitude de la géométrie reconstruite et le bruit du capteur. Grégoire *et al.* [Nieto *et al.*, 2016] sont allés plus loin en rajoutant plus de contraintes sur les gradients de l'image pour réduire les artefacts visuels causés par les discontinuités dans les poids de mélange. Plus récemment, Goesele *et al.* [Goesele *et al.*, 2010] proposent un système d'interpolation

---

basé image, dans lequel ils utilisent diverses représentations géométriques, y compris des maillages dans les régions où la profondeur est correctement reconstruite. Hedman *et al.* [Hedman *et al.*, 2016] utilisent un maillage globale reconstruit via un capteur de profondeur, ainsi qu'un maillage simplifié pour chacune des images originales pour éviter les artefacts visuels pouvant se produire à cause du mauvais alignement entre le modèle globale et les bordures des objets sur les images. Pour alléger le temps de rendu, ils ont également proposé des structures de données dédiées pour ne visualiser que les maillages visibles par les caméras pertinentes (voir figure 2.2) . Dans [Hedman *et al.*, 2018], les auteurs sont allés plus loin en utilisant l'apprentissage en profondeur (Deep learning) pour calculer les poids de mélange.

**Les méthodes de rendu basé image utilisant un nuage de points** La vision par ordinateur a été développée avec différents types de systèmes d'acquisitions produisant souvent des nuages de points denses [Furukawa *et Ponce*, 2010, Song *et al.*, 2018, Wang *et al.*, 2015]. Les méthodes de type structure à partir de mouvement [Snavely *et al.*, 2006, Snavely *et al.*, 2008] offrent des moyens pour étalonner les caméras et générer des nuages de points à partir de photos prises à main levée.

Certaines approches consistent à visualiser directement les nuages de points, en association avec des textures. Par exemple : Yang *et al.* [Yang *et al.*, 2006] utilisent des textures dépendantes du point de vue avec un système de rendu basé points.

D'autres, choisissent d'employer les points comme une représentation grossière ; Chaurasia *et al.* [Chaurasia *et al.*, 2011] proposent d'utiliser un nuage de points comme proxy pour déformer les photos ainsi que la mise en place de contraintes sur les silhouettes des objets pour gérer les occultations. Dans [Chaurasia *et al.*, 2013], ils proposent d'utiliser un algorithme de synthèse de profondeur pour produire de nouveaux points dans les régions mal reconstruites et une sur-segmentation des images pour former des superpixels. Le rendu de la nouvelle vue s'effectue en déformant chaque superpixel séparément au lieu de déformer l'image entière, ce qui rend la navigation plus adaptée à un grand nombre de point de vue par rapport à [Chaurasia *et al.*, 2011].

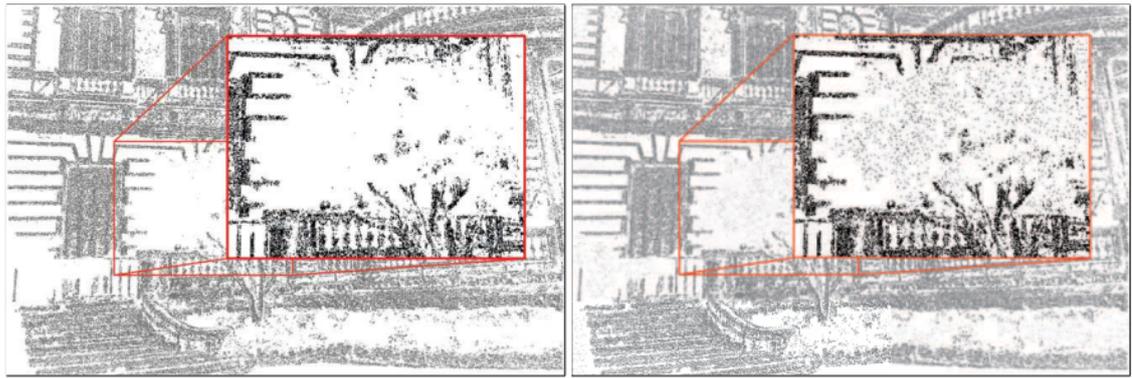


FIGURE 2.3 – Synthèse de profondeur [Chaurasia *et al.*, 2013]

Grâce à la synthèse de profondeur, de nouveaux points sont créés dans les régions mal reconstruites.

La méthode proposée par Ortiz-Cayon *et al.* [Ortiz-Cayon *et al.*, 2015] repose sur la même idée de la sur-segmentation des images en plusieurs régions (super-pixels). Ils ont expérimenté différents algorithmes de rendu basé image sur chaque région, et en gardant celui qui produit de meilleurs résultats. Ces méthodes ont prouvé leur robustesse dans la gestion des occultations, même avec des régions mal reconstruites, et cela grâce à la sur-segmentation des images et à l'extraction des silhouettes qui gèrent avec précision les discontinuités de profondeur. Ces méthodes se placent dans le contexte de la navigation libre "free viewpoint" dans des scènes urbaines. La différence par rapport à notre travail réside dans leur capacité à manipuler et gérer des scènes complexes grâce à l'étape de synthèse de profondeur qui permet de créer de l'information géométrique dans les régions où les méthodes de stéréo-multi-vue échouent (figure 2.3).

### 2.1.3 Discussion

La plupart des systèmes de navigation à bases d'images reposent sur des systèmes d'acquisitions évolués, nécessitant du matériel spécifique, et souvent avec l'intervention de l'utilisateur. Certaines d'entre elles reposent sur la reconstruction de maillages [Gortler *et al.*, 1996, Hedman *et al.*, 2016, Hedman *et al.*, 2018] ou nécessitent l'intervention de l'utilisateur [Chaurasia *et al.*, 2011] pour guider la reconstruction. D'autres approches dites variationnelles [Pujades *et al.*, 2014, Nieto *et al.*, 2016] permettent de compenser le manque de précision des maillages géométriques. Cependant,

---

elles sont coûteuses en temps de calcul. Notre objectif est de proposer un système de visualisation basé-image ne nécessitant pas la reconstruction d'un maillage des objets, ni aucune intervention de l'utilisateur, et cela à partir de quelques photographies prises à main levée. Cela se rapproche des concepts des lumigraphes non structurés, mais sans reconstruction de maillage géométrique ni dans l'espace objet, ni dans l'espace caméra et avec une gestion différente des occultations et des photographies.

## 2.2 Reconstruction des images HDR à partir de photos prises à main levée

Les premières approches de reconstruction d'images à haute gamme de dynamique (HDR) reposent sur un même principe introduit initialement par Mann et Picard [Mann et Picard, 1994] et Debevec et Malik [Debevec et Malik, 1997]. L'idée est de fusionner plusieurs images prises depuis un même point de vue, avec des temps d'exposition différents afin d'obtenir une nouvelle image contenant tous les niveaux de luminance des différentes expositions. Ces approches fonctionnent correctement si l'objet est capturé d'un même point de vue. Les photographes utilisent généralement des caméras montées sur des trépieds pour éviter les défauts d'alignement entre les expositions. En revanche, si les objets sont en mouvement ou la scène est capturée par des caméras à main levée, ces approches produisent des effets de floue et d'objets fantômes dans l'image HDR.

Certains auteurs se sont intéressés à proposer des méthodes de reconstruction d'images HDR à partir de photos prises par des caméras standards à main levée. La plupart d'entre eux sont basés sur l'idée d'aligner toutes les photographies en entrée sur l'une des expositions, appelée l'image référence (voir la figure 2.4). Par exemple, la méthode proposée par Tomaszewska *et al.* [Tomaszewska et Mantiuk, 2007] utilisent les descripteurs de formes robustes SIFT pour extraire les points d'intérêts "*feature-points*" et établir les correspondances entre les points produites. Ces correspondances sont ensuite utilisées pour exprimer l'ensemble d'images dans le système de coordonnées de l'image référence. Cette méthode est implémentée dans le logiciel

---

"*pfstools*"<sup>1</sup>. Son inconvénient majeur est qu'elle échoue en cas de mouvement de caméras plus complexes qu'une simple rotation .

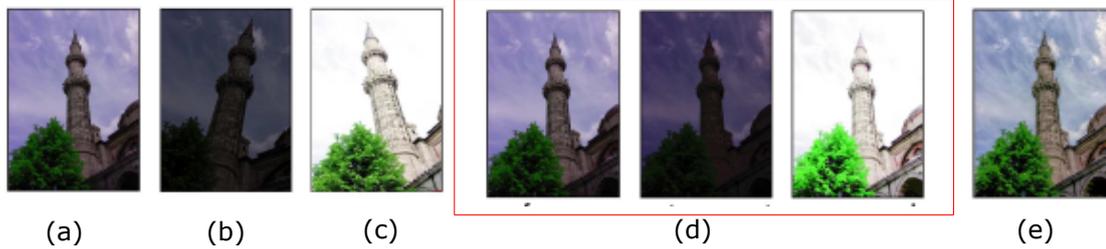


FIGURE 2.4 – **Reconstruction de l'image HDR à partir de photos tenue à main levée** [Tomaszewska et Mantiuk, 2007]

(a) l'image référence; (b) et (c) la même scène prise par caméra à main levée avec des niveaux d'expositions différents (d) l'alignement des images (b) et (c) sur l'image référence (a); (e) l'image HDR résultante.

Zimmer *et al.* [Zimmer *et al.*, 2011] utilisent le flux optique pour aligner les images. Leur méthode est basée sur le calcul d'un champ de vecteurs de déplacement dense permettant de décrire de mouvements arbitraires (rotation, translation, zoom..etc). Bien que, cette méthode est efficace par rapport aux autres, elle reste moins robuste aux mouvements provoqués par les objets dans la scène (voir figure 2.5). Plus récemment, Sen *et al.* [Sen *et al.*, 2012] ont remédié à ce problème en intégrant l'alignement et la reconstruction en utilisant une variante de l'algorithme PatchMatch. La reconstruction de l'image HDR correspond à la minimisation d'une équation d'énergie dont la solution optimale correspond à l'image référence dans ses régions correctement exposées et aux autres images LDR dans ses régions sous ou sur-exposées.



FIGURE 2.5 – **Reconstruction de l'image HDR par la méthode de Zimmer** [Zimmer *et al.*, 2011]

Des effets d'objets fantômes apparaissent sur l'image HDR résultante.

---

1. <http://pfstools.sourceforge.net/>

---

**Discussion** La reconstruction d’une image HDR à partir d’un ensemble d’expositions non alignées sur un même point de vue est devenue un sujet de recherche actif, en particulier dans les domaines de l’analyse d’images et de la vision par ordinateur. Bien que ces approches ont réussi dans la plupart des cas à produire une image HDR bien alignée et sans artefacts visuels, elles échouent dans certains cas tel que dans la gestion des occultations complexes et cela à cause de l’absence de l’information géométrique.

## 2.3 Rendu basé point

De nombreux auteurs se sont intéressés à reconstruire des maillages géométriques [Amenta et Bern, 1999, Kolluri *et al.*, 2004, Kazhdan *et al.*, 2006, Denker *et al.*, 2011, Fuhrmann et Goesele, 2014] à cause de leur forte connectivité. Cependant, le processus de visualisation peut être affecté par la complexité de la scène dès que celle-ci dépasse un certain seuil. Les nuages de points sont considérés parmi les représentations alternatives, récemment utilisées. Principalement, à cause de leur flexibilité et fiabilité. De nombreuses méthodes ont été proposées pour visualiser directement les nuages de points denses. Chaque point est considéré comme un élément de surface associé à une position, une orientation, une couleur (R,V,B) et un rayon correspondant à la surface sur laquelle le point s’étend. Le rayon est généralement calculé en fonction de la distance du point à ses plus proches voisins, de manière à ce que les surfaces des différents points se recouvrent et ainsi éviter l’apparition de trous durant la visualisation.

Diverses structures de données ont été proposées pour représenter les points. Nous allons nous intéresser uniquement aux méthodes de visualisation de nuages de points dans l’espace image et aux méthodes de visualisation par splatting. Les autres approches dites lancer de rayon se situent hors de la portée de ce manuscrit.

### 2.3.1 Splatting

Les méthodes de visualisation par splatting consistent à rendre chaque point par un petit élément de surface (disque orienté dans l’espace objet). L’intérêt de

ces méthodes réside dans leur capacités à reconstruire une surface lisse, sans trous sans avoir besoin d'un algorithme de reconstruction de surface et cela grâce au chevauchement des splats dans l'espace objet. Pfister *et al.* [Pfister *et al.*, 2000] étaient les premiers qui proposent d'utiliser les splats comme primitives de rendu. Un état de l'art complet sur les premières approches de splatting peut se trouver dans [Kobbelt et Botsch, 2004].

De nombreux auteurs ont exploité les capacités des cartes graphiques pour apporter plus de flexibilité aux algorithmes de splatting. La première implantation du splatting sur GPU a été proposée par Ren *et al.* [Ren *et al.*, 2002] qui représentent chaque point par un quadrilatère texturé. Les quatre sommets du quadrilatère sont calculés au niveau du vertex shader en utilisant un filtre de ré-échantillonnage de l'EWA (*pour elliptical weighted average*) splatting dans l'espace objet. Au moment de leur travaux, la passe de normalisation par la somme des poids étant n'était pas encore supportée par les cartes graphiques, les auteurs choisissent de pré-normaliser les contributions des splats.

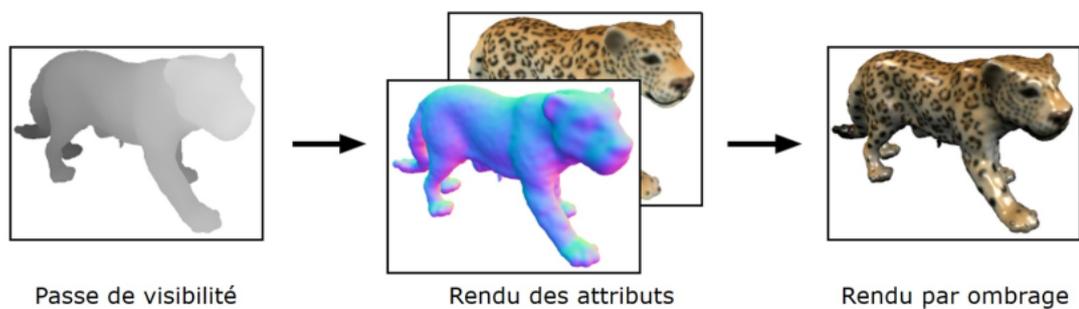


FIGURE 2.6 – Figure de [Botsch *et al.*, 2005] : Rendu par splatting sur GPU

Plus récemment, avec la possibilité d'affiner les développements sur carte graphique, la passe de normalisation des contributions des splats est devenue possible : les contributions sont accumulées dans un framebuffer au lieu de l'écran. La normalisation est donc réalisée au niveau du fragment shader par une simple division des coordonnées (R,G,B) par la composante alpha . De plus, l'avènement de la primitive *point* a permis la rasterisation d'un carré en utilisant un sommet par point au lieu de quatre sommets. La taille, la forme et l'orientation des splats sont approchés dans l'espace écran [Guennebaud et Paulin, 2003, Botsch et Kobbelt, 2003], puis envoyés

au fragment shader pour estimer la profondeur réelle de chaque fragment et interpoler correctement les splats voisins. Botsch *et al.* [Botsch *et al.*, 2005] proposent une approche qui exploite ces avancées. Le pipeline employé dans l'approche est illustré sur la figure 2.6. Il se compose de trois passes de rendu : la première consiste à rendre les splats dans une texture en activant le test de profondeur pour ne garder que les splats les plus proches à l'écran. Un rayon est ensuite lancé de l'espace écran vers l'espace objet (voir figure 2.7) pour déterminer si le fragment actuel fait partie ou pas du splat ainsi que pour ajuster sa valeur de profondeur comme décrit dans [Botsch *et al.*, 2004]. Dans une deuxième passe, les attributs des splats (couleur, vecteur normal) sont ensuite accumulés sur la base de la carte de profondeur produite et le test de mélange (blending). La normalisation des splats est alors effectuée dans un dernier temps sur la base des attributs accumulés. L'intérêt principal de cette approche est qu'elle permet une reconstruction sans aliasing, aboutissant à un résultat de haute qualité et cela grâce à l'utilisation d'une version avancée du filtre EWA.

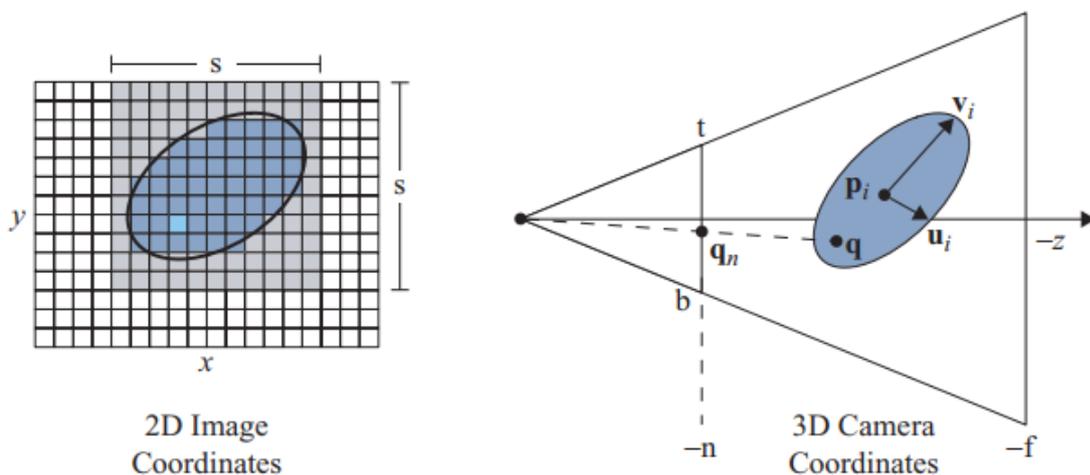


FIGURE 2.7 – **Test d'appartenance au splat par lancer de rayon ;** pour chaque fragment du carré, un rayon est lancé de l'espace écran vers l'espace objet, les paramètres locaux du point d'intersection avec le plan du splat sont calculés et utilisés pour déterminer si le fragment en question est valide ou pas, ainsi que pour ajuster sa valeur de profondeur [Gross et H. Pfister (eds.), 2007].

---

### 2.3.2 Rendu par reconstruction dans l'espace image

Pour accroître encore les performances de la visualisation, plusieurs auteurs proposent de travailler exclusivement dans l'espace image. Le principe consiste à projeter les points sur l'espace écran sous la forme de pixels. Cette étape correspond à la passe géométrique. Les attributs associées à chaque point (normale, profondeur, etc..) sont stockés dans des textures et utilisés pour combler les éventuels trous après la première passe. Une première difficulté consiste alors à classifier les pixels qui peuvent être soit valides (remplis) soit à reconstruire (trous ou pixels non visibles). Pour ce type de méthodes, deux stratégies sont généralement mises en œuvre : la première consiste à utiliser de l'information de profondeur pour gérer les occultations et éliminer les points non visibles. La seconde est que pour un pixel vide, de déterminer puis d'interpoler les pixels valides les plus proches.

L'intérêt principal de ces approches réside dans le fait qu'elles restent relativement indépendantes de la complexité géométrique de la scène car les traitements ne dépendent que de la résolution des textures résultantes à la suite de la passe géométrique. Le second intérêt est que certaines de ces approches ne s'appuient que sur une seule étape géométrique alors que les approches de rendu par splatting en exigent deux.

Grossman et Dally [Grossman et Dally, 1998] sont les premiers qui proposent l'idée de rendre les nuages de points dans l'espace image. Une hiérarchie de cartes de profondeur à résolution décroissante est utilisée pour gérer les problèmes de visibilité. Au moment du rendu, pour chaque partie de l'objet, le tampon de profondeur ayant la résolution suffisamment faible est choisi. Les attributs des points projetés (couleur, vecteur normal, profondeur) sont stockés dans le tampon d'image et le tampon de profondeur. Afin de détecter et reconstruire les trous dans l'image source, une comparaison des profondeurs de cette image avec les profondeurs des différents niveaux de l'hiérarchie est réalisée où chaque point est associé d'un coefficient entre 0 et 1 qui indique à quel degré le pixel est visible ou non.

Ce type d'approche, nommée pull-push est à l'origine inspirée de [Gortler *et al.*, 1996] pour la reconstruction de lumigraphes. Durant le pull, une pyramide d'images à résolution décroissante est construite en utilisant les coefficients pour moyennner les

---

pixels valides. Dans le push, l'image de faible résolution est utilisée pour combler les trous ou recalculer les pixels non visibles.

Les approches récentes adaptées aux cartes graphiques offrent des performances remarquables. Marroquim *et al.* [Marroquim *et al.*, 2007] proposent également une approche de rendu basée sur l'algorithme de pull-push. Contrairement à celle proposée par Grossman et Dally qui sépare le calcul de visibilité de la reconstruction d'images, Marroquim réunit les deux. Le principe consiste à projeter chaque point sur un seul pixel, les attributs du point (profondeur, la normale, les coordonnées du point projeté sur l'espace écran, etc..) sont stockés dans des textures. Cette étape correspond à la passe géométrique. Les textures produites de cette étape sont utilisées dans les deux passes pull et push pour combler les éventuels trous. Pour résoudre le problème de visibilité, un intervalle de profondeur est associé à chaque point (méthode décrite plus en détail dans la section 3.2.1 ). Malgré cette technique est basée sur un processus de splatting, elle se place parmi les techniques de rendu par traitement en espace image car la reconstruction des splats est effectuée exclusivement dans l'espace image à partir d'une représentation purement ponctuelle. Malgré cette méthode est plus rapide que les approches de rendu par splatting. Elle souffre néanmoins de quelques artefacts sous forme de clignotements lorsque le point de vue change. En 2008, Marroquim *et al.* [Marroquim *et al.*, 2008] ont amélioré la méthode en introduisant de petits noyaux (kernels) de  $5 \times 5$  dans l'hierarchie afin d'éviter les interpolations inutiles. Bien que ces améliorations réduisent considérablement les clignotements, la méthode est néanmoins coûteuse en terme de temps de calcul.

Rosenthal et Linsen [Rosenthal et Linsen, 2008] proposent une autre approche efficace de rendu à base de points évitant les étapes de pull-push. Le processus de reconstruction est réalisé en deux étapes en utilisant de filtres  $3 \times 3$  sur l'ensemble des pixels de l'image. Dans un premier temps, les filtres sont utilisés pour décider pour chaque pixel vide s'il doit être rempli ou non en fonction de son voisinage. Une fois, les trous pertinents sont remplis, une deuxième passe basée également sur des filtres est effectuée pour le retrait des parties cachées. Malgré sa simplicité, elle présente des inconvénients : la méthode n'est plus pertinente lorsque le point de vue est trop proche de la surface de l'objet car l'espacement local entre les points devient grand.

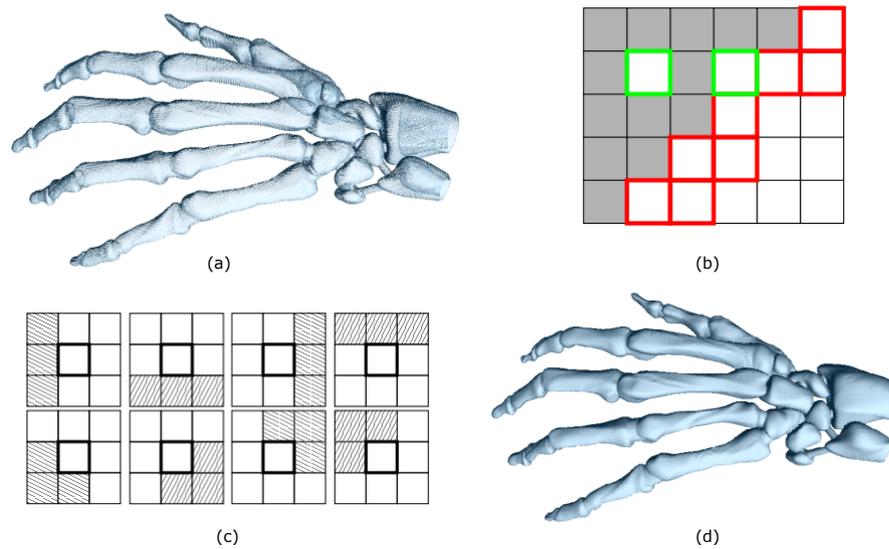


FIGURE 2.8 – Pipeline utilisé par [Rosenthal et Linsen, 2008]

(a) le nuage de points initial. (b) Zoom sur une partie de l’objet : le pixel en gris correspond à un pixel atteint par un point projeté (rempli) ; les pixels nécessitant d’être remplis sont encadrés par le vert car ils sont proches de pixels remplis, les autres (encadrés par le rouge) dépassent la silhouette et par conséquent ils ne devraient pas être remplis. (c) : masques utilisés pour tester si un pixel vide nécessite d’être rempli ou non (d) le résultat final après deux passes de remplissage (pixels vides et occultés).

De plus, sa nature itérative lui rend coûteuse dans certains cas et en particulier si la densité du nuage n’est pas uniforme, ce qui exigerait un grand nombre d’itérations pour combler les trous. Preiner *et al.* [Preiner *et al.*, 2012] proposent une autre approche qui permet de faire du splatting sur des nuage de points bruts. Seuls les positions des points 3D sont requis en entrée. Les rayons et les normales sont estimés exclusivement sur les GPU, après projection des points sur l’espace image. Le rendu des points est ensuite réalisé via la méthode par splatting de Botsch [Botsch *et al.*, 2005].

### 2.3.3 Discussion

Les méthodes de rendu par reconstruction dans l’espace image semblent plus intéressantes que les méthodes de rendu par splatting pour les raisons évoquées précédemment (temps de calcul moins coûteux). Néanmoins, ces méthodes ont chacune son lot de limites : par exemple, l’approche de Rosenthal [Rosenthal et Linsen, 2008] reste cantonné à la visualisation de nuages de densité uniforme. Dans la méthode de

---

Preiner [Preiner *et al.*, 2012], les normales sont estimées sur les GPU, alors que les méthodes de stéréo multi-vue permettent de produire de telles informations. De plus, l’approche adoptée pour visualiser les nuages reste coûteuse.

L’approche de reconstruction par pyramide d’images [Marroquim *et al.*, 2007] semble plus intéressante car elle reste la plus rapide et permet de prendre en compte de manière plus précise l’orientation des éléments de surfaces associés aux points 3D de l’environnement. Néanmoins, elle souffre de quelques artefacts de clignotements dus principalement à la nature de l’interpolation. Nous proposons deux passes de rendu supplémentaires permettant d’améliorer la qualité des cartes de profondeurs produites sans trop impacter les performances de l’algorithme.

## 2.4 Reconstruction 3D

De manière générale, les méthodes de rendu à base d’images nécessitent une étape d’étalonnage de chaque photographie. L’objectif est de déterminer les paramètres intrinsèques et extrinsèques : la distance focale, ainsi que la position et l’orientation de l’appareil-photo au moment de la prise de vue. Notre approche, de façon similaire aux autres systèmes de rendu à base d’images, nécessite de déterminer ces paramètres, et notre objectif est de simplifier au maximum ces traitements. De nombreuses méthodes existent dans la littérature, avec l’utilisation de mires ou de systèmes d’acquisition dédiés, mais l’arrivée de descripteurs performants et quasi invariants aux translations et aux rotations, tels que SIFT [Lowe, 2004], associés à des systèmes de type Structure from Motion permettent de simplifier radicalement cette étape d’étalonnage. Associés à des méthodes d’appariement telles que RANSAC, ils ont démontré leur robustesse pour de nombreuses bases d’images [Snavely *et al.*, 2006, Snavely *et al.*, 2008]. Hartley et Zisserman [Hartley et Zisserman, 2003] présentent de manière détaillée les différents algorithmes utilisés.

De nombreux frameworks de *Structure from Motion* ont été proposés et implémentés dans des logiciels dont on peut citer : le Bundler [Snavely *et al.*, 2006], Visual SFM [Wu *et al.*, ] et la bibliothèque openMVG [Moulon *et al.*, 2016a]. Nous avons utilisé la dernière car elle offre une implémentation complète d’une chaîne de

---

"Structure à partir de mouvement" et fournit des exporteurs pour des logiciels de stéréo multi-vue comme la *PMVS*, *openMVS* et *MVE*

### 2.4.1 Stéréo multi-vues

Une grande famille de stéréo multi-vues s'attelle à extraire de l'information géométrique sous forme de cartes de profondeurs, de nuages de points denses ou de maillages géométriques à partir d'un ensemble de photographies et des paramètres des appareils photo (obtenues par les méthodes de structure à partir de mouvement). Goesele *et al.* [Goesele *et al.*, 2007] proposent une méthode implémentée dans le logiciel MVE [Fuhrmann *et al.*, 2014] reposant sur la construction d'une carte de profondeur pour chacune des photographies. Ces cartes de profondeur sont ensuite fusionnées pour représenter l'intégralité de l'objet visionné et peuvent ensuite être utilisées pour construire un maillage triangulaire ou un nuage de points. Ainsi Furukawa et Ponce [Furukawa et Ponce, 2007] proposent une méthode implémentée dans les logiciels PMVS (*Patch-based Multi-View Stereo*) et PMVS2 pour générer un nuage de points orienté et dense "appelé souvent patches" à partir d'un ensemble de photographies calibrées. Cette méthode est ensuite améliorée [Furukawa et Ponce, 2010] en regroupant les photographies en clusters pour effectuer les calculs en parallèle. Cette extension est implémentée dans le logiciel CMVS (*Clustering Views for Multi-view Stereo*). Nous avons utilisé la PMVS2 car elle est très simple à exploiter et soumise à aucun traitement. De plus, elle produit des nuages de points moins denses et moins bruités par rapport à MVE et openMVS. Dans l'approche que nous avons du rendu basé image, nous exploitons directement le nuage de points produit pour calculer des cartes de profondeurs. Cependant, il est possible de continuer le processus de triangulation afin de reconstruire une surface sous la forme d'un maillage triangulaire.

### 2.4.2 Reconstruction de la surface

L'utilisation d'un nuage de points ou d'un maillage géométrique est avantageuse par rapport à l'utilisation des cartes de profondeur produites par les méthodes de stéréo multi-vues. D'abord, les nuages de points et les maillages ne produisent pas

---

de données redondantes comme les cartes de profondeurs. En outre, ils sont plus adaptés à la synthèse de point de vue libre "*free viewpoint*", un contexte dans lequel notre approche se place. Cependant, nous privilégions l'usage des nuages de points car la reconstruction d'un maillage parfait est vaine.

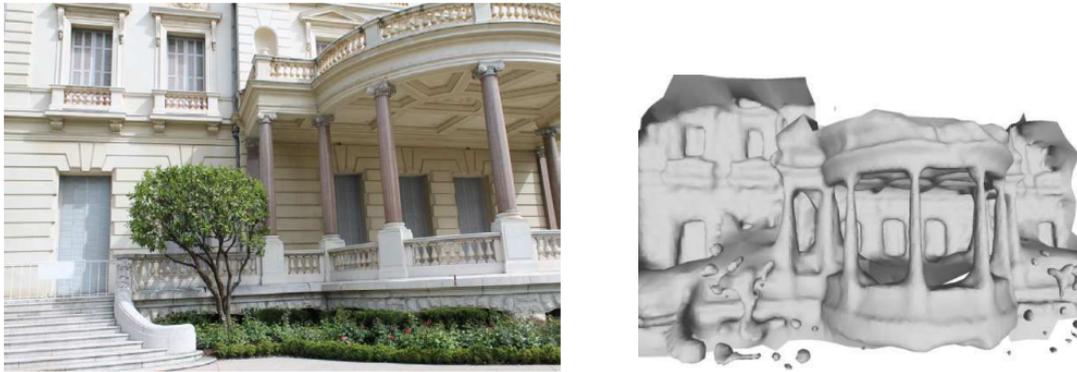


FIGURE 2.9 – **Reconstruction d'un maillage** [Chaurasia, 2014]

l'une des photographies d'origines. A droite : la reconstruction d'un maillage géométrique [Kazhdan *et al.*, 2006] à partir d'un nuage de points. La triangulation produite n'est pas adaptée aux arêtes et aux courbures de l'objet.

Les méthodes de reconstruction de surfaces peuvent être classées en deux catégories : les méthodes de reconstruction dites "de poisson" à partir d'un nuage de points et les méthodes de reconstruction à partir d'un ensemble de cartes de profondeur. Les méthodes faisant partie de la première catégorie [Kazhdan *et al.*, 2006, Kazhdan et Hoppe, 2013] utilisent la triangulation de Delaunay pour produire un maillage à partir d'un nuage de points orientés (chaque point est associé à un vecteur normal). Ces méthodes ne sont pas adaptées pour des scènes extérieures (voir figure 2.9) car elles créent souvent des surfaces qui n'existent pas (création de dôme autour de la scène, remplissage de trous), ce qui impacte négativement la qualité du rendu. Nous utilisons ces algorithmes dans nos expériences car notre objectif n'est pas de traiter des scènes extérieures complexes (des scènes urbaines). De plus, ces algorithmes ne nécessitent pas des nuages très denses par rapport aux autres logiciels comme openMVS. Dans nos expériences, nous avons utilisé les nuages produits par PMVS2 pour produire des maillages, utilisés pour implémenter les lumigraphs non structurés [Buehler *et al.*, 2001] à titre de comparaison. La reconstruction de poisson est également intégrée dans Meshlab, ce qui permet de reconstruire, visualiser, mani-

---

puler et simplifier des maillages texturés. Pour remédier au problème de remplissage de trous indésirables, on peut enlever manuellement les triangles suspects en utilisant Meshlab.

La deuxième classe d'algorithmes est basée sur la fusion des cartes de profondeur [Fuhrmann et Goesele, 2011, Fuhrmann et Goesele, 2014]. Les nuages de points obtenues à partir des cartes de profondeurs sont structurés car chaque paire de pixels adjacents dans la carte correspond à deux points connectés dans l'espace 3D. Ces techniques ne génèrent pas les défauts évoqués précédemment, mais elles nécessitent des cartes de profondeur très denses.



**FIGURE 2.10 – Reconstruction 3D de la boule de neige en utilisant openMVG et PMVS2**

ces algorithmes ont du mal à reconstruire les régions spéculaires.

### 2.4.3 Discussion

Nous avons succinctement présenté les différents algorithmes de reconstruction 3D qui permettent d'estimer de l'information géométrique servant au rendu basé images. Le choix entre ces méthodes dépend des besoins et du type de rendu que nous voulons. La recherche d'un "proxy géométrique" idéal est vaine car malgré ces algorithmes sont différents entre eux dans le principe, ils posent les mêmes problèmes dans certains cas bien particuliers : ils supposent tous que la scène est lambertienne. Les régions spéculaires ne peuvent pas être reconstruites car la luminance reçue par la caméra dépend essentiellement de la luminance provenant de la direction

---

miroir ce qui perturbe la mise en correspondance (voir la figure 2.10). De plus, ils produisent des cartes de profondeurs et des nuages de points très peu denses pour les régions non texturées (ayant une couleur unie). Les méthodes de reconstruction de maillage permettent dans certains cas de combler les trous et reconstruire les parties manquantes. Cependant, les surfaces résultantes sont souvent imprécises car le processus de triangulation cherche à lier les différents points mêmes s'ils sont distants, ce qui altère souvent la forme de l'objet et impacte par la suite gravement la qualité du rendu. Pour cela, nous avons choisi d'utiliser un algorithme de rendu basé point pour combler les espaces entre les points voisins à travers une interpolation de profondeur fiable et plus adaptée aux courbures de l'objet.

# Chapitre 3

## Rendu basé image interactif avec proxy basé point

Créer un système de navigation qui permet à partir de photos, de naviguer librement d'une manière fluide et réaliste dans des environnements acquis est devenu un thème de recherche très actif ces dernières années en rendu basé image. À ce titre, de nombreuses approches ont été proposées avec différents traitements et diverses représentations géométriques [Eisemann *et al.*, 2008, Chaurasia *et al.*, 2011, Chaurasia *et al.*, 2013]. Dans ce chapitre, nous présentons un système avec lequel on peut atteindre les mêmes objectifs, et cela à partir d'un ensemble de photographies prises à main levée. La solution proposée ne nécessite ni reconstruction de maillages ni de traitements complexes. Elle est basée sur un nuage de points utilisé comme proxy géométrique pour reconstruire une carte de profondeur pour chaque point de vue, à travers une méthode d'interpolation qui s'opère exclusivement dans l'espace image. Cela permet d'accélérer le rendu vu que l'interpolation dépend essentiellement de la résolution des images. Nous proposons ensuite des passes de corrections qui permettent de réduire les effets de clignotements sur la silhouette de l'objet, et qui sont dus principalement à la nature de l'interpolation, ce qui permet d'améliorer significativement la qualité de rendu. Enfin, nous présentons un système complet de cohérence épipolaire pour gérer les occultations et les contributions de chacune des photographies.

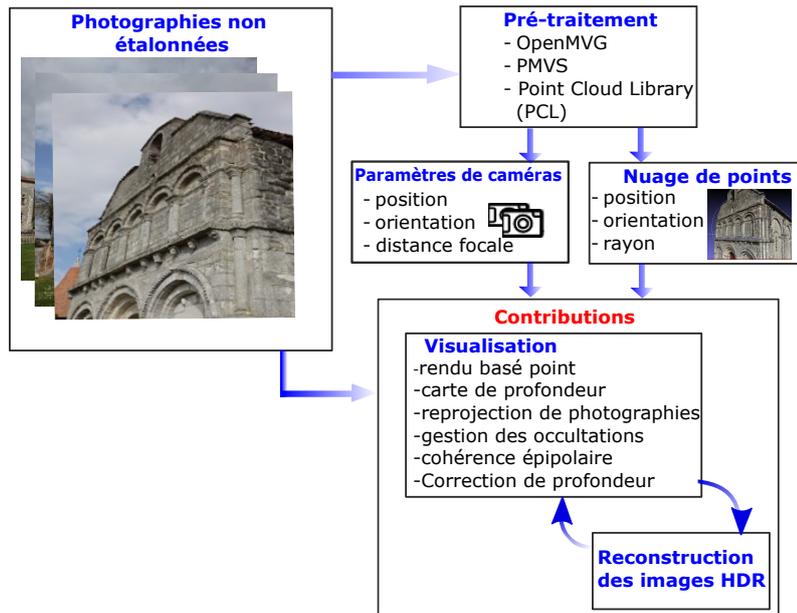


FIGURE 3.1 – Architecture logicielle de notre système de rendu à base d’images interactif

Le système permet une navigation virtuelle libre. En outre, avec plusieurs séries de photographies, dont l’exposition varie, il permet de gérer automatiquement la construction d’images à haute gamme de dynamique (images HDR), que nous expliquons en détail dans le chapitre suivant (chapitre 4).

### 3.1 Présentation générale du système

La figure 3.1 présente l’architecture logicielle du processus général, y compris notre système de visualisation (le noyau de ce travail).

Le traitement des photographies en entrée est réalisé à l’aide d’une chaîne de structure à partir de mouvement [Moulon *et al.*, 2016a] pour estimer les paramètres intrinsèques et extrinsèques de chaque prise de vue, suivi d’un algorithme de stéréo multi-vue [Furukawa et Ponce, 2010] pour produire un nuage de points dense correspondant à la surface de l’objet capturé.

Pour chaque photographie, un ensemble de descripteurs SIFT [Lowe, 2004] est déterminé. L’appariement est réalisé à l’aide d’une structure de donnée en Kd-arbre, et d’une approche de type plus proche voisins (Approximate nearest neighbor). Un pipeline de Structure from Motion incrémentale ou global [Moulon *et al.*, 2012]

---

permet d'estimer les paramètres intrinsèques (distance focale et facteurs de distorsion de l'objectif) et les paramètres extrinsèques (position et orientation) de la caméra pour chaque point de vue, c'est à dire la matrice de projection globale. La seconde étape de pré-traitement consiste à densifier le nuage de points produit par l'OpenMVG, en utilisant la PMVS, où chaque point est associé à une couleur (R,G,B) et à une orientation.

Les éléments de surface associés à chaque point sont déterminés en fonction de son voisinage. Un rayon est attribué à chaque point, de manière à ce que les éléments de surface se recouvrent. Pour calculer efficacement cette information, nous utilisons la PCL (Point cloud Library) [Rusu et Cousins, 2011]. Le nuage de points produit, les paramètres de la caméra ainsi que les photographies étalonnées sont ensuite directement utilisés pour la visualisation.

Lors de la visualisation, pour chaque nouveau point de vue, tous les points sont projetés sur l'espace image. Nous utilisons l'algorithme proposé par Marroquim [Marroquim *et al.*, 2007] pour définir une profondeur en chaque pixel. L'accent a été mis sur cet algorithme pour son efficacité. La reconstruction de la profondeur est fait exclusivement dans l'espace image, ce qui rend cet algorithme plus rapide par rapport aux autres approches de splatting. Ce processus est également appliqué aux photographies d'origines pour leur associer une carte de profondeur. La couleur de chaque pixel de l'image est estimée par projection inverse sur chaque photographie, avec la prise en compte des occultations et une pondération des couleurs des photographies pertinentes selon un modèle de cohérence épipolaire. L'interpolation proposée par Marroquim est une approximation. Les splats sont reconstruits dans l'espace image, ce qui produit des effets de clignotements sur la silhouette de l'objet. De plus, si le nuage de points n'est pas suffisamment dense, l'algorithme échoue lorsque nous rapprochons de la surface de l'objet. Ces défauts sont corrigés par des passes de rendu supplémentaires, avec un impact léger sur les performances de la visualisation. Avec plusieurs séries de photographies dont l'exposition varie, notre système permet de gérer automatiquement la construction d'images à haute gamme de dynamique (HDR).

## 3.2 Modélisation du système

La figure 3.2 illustre le résultat produit par l'étape d'étalonnage des photographies et de construction du nuage de points. Sur cette figure, la statue de Mirebeau est représentée par le nuage de points reconstruit ; la position des appareils photos (déterminé par le système d'étalonnage) est également illustré.



FIGURE 3.2 – Points de vues et nuage de points

Pour synthétiser une image LDR/HDR depuis un nouveau point de vue  $C$  (dont la matrice de projection est  $M$ ), notre système de visualisation consiste à : (i) reconstruire la profondeur et le vecteur normal de l'objet visualisé en chaque pixel ; (ii) À partir de la profondeur, calculer la position 3D correspondante ; (iii) déterminer la couleur (R,G,B) par projection inverse sur les photographies d'origine, avec la prise en compte des occultations et une pondération de la direction d'observation de chacune des caméras. Dans ce qui suit, nous décrivons en détail chacune de ces étapes, en mettant l'accent sur les différents critères qu'on doit prendre en compte pour bénéficier pleinement du niveau de détail offert par les photographies d'origines.

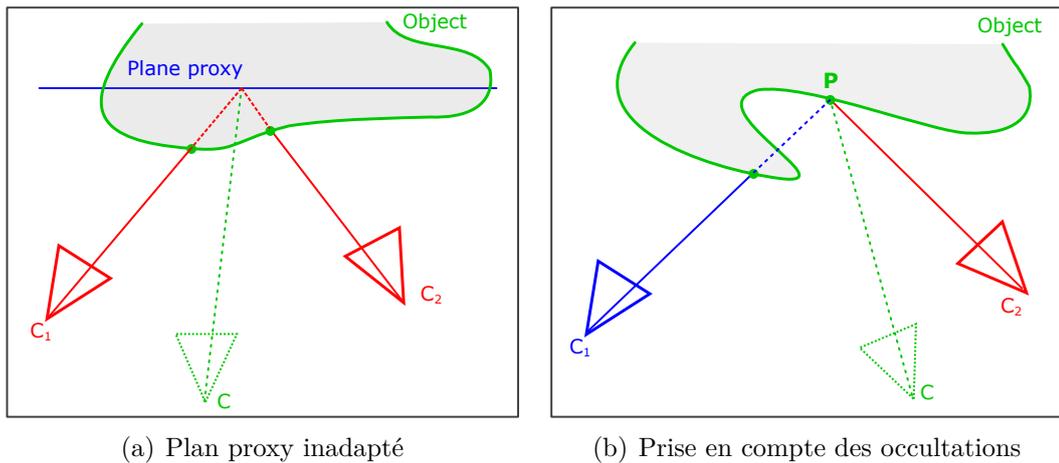


FIGURE 3.3 – **Proxy géométrique et configurations possibles**

(a) Lorsque le proxy diffère trop de l'objet visualisé, la reconstruction de la couleur (R,G,B) devient fautive pour la caméra  $C$ , cela produit des effets de lissage et conduit à une perte de précision importante; (b) La caméra  $C_1$  ne doit pas être prise en compte car elle ne visualise pas la même région que  $C$  pour le pixel considéré, les artefacts visibles sont des effets d'objets fantômes.

### 3.2.1 Rendu des points

Les méthodes de rendu à base d'images utilisent souvent des proxy géométriques pour éviter les erreurs de parallaxe lorsque l'utilisateur se déplace autour de l'objet [Gortler *et al.*, 1996, Vangorp *et al.*, 2011]. Un proxy sert à déterminer précisément la profondeur de la surface observée en un pixel de la nouvelle vue et à identifier correctement les pixels à utiliser sur les photographies d'origines.

La figure 3.3 illustre deux cas pour lesquelles seul un proxy précis peut déterminer correctement les pixels de l'image  $C$ . Une représentation géométrique trop simple conduit à des affichages d'objets fantômes et à des images floues. Notre approche repose sur un proxy basé point, avec une reconstruction de la profondeur par un algorithme de rendu basé point (aussi appelé : *pull push*).

#### Reconstruction de la profondeur

La première étape de l'algorithme consiste à projeter tous les points sur l'espace image, la profondeur des points, leur vecteurs normaux ainsi que leur rayons sont stockés dans un framebuffer à l'aide de deux textures  $T_{rn}$  et  $T_{zc}$ .  $T_{rn}$  stocke le rayon  $r$  associé à chaque point projeté ainsi que les coordonnées du vecteur normal  $N$ ;

---

$T_{zc}$  comporte la profondeur du point projeté, l'intervalle de profondeur ainsi que les coordonnées relatives du centre de l'ellipse dans l'espace image.

L'objectif est d'obtenir de cette représentation partielle la profondeur de chacun des pixels de l'objet visualisé. Pour cela, une pyramide d'images est construite et un algorithme de type pull push remplit les espaces entre les points projetés (pixels) par interpolation des échantillons les plus proches.

**Phase de remontée (pull)** Dans la phase de remontée, une série d'images à résolutions décroissantes est calculée progressivement en utilisant les informations de l'image de résolution supérieure pour remplir l'image de résolution inférieure. Un intervalle de profondeur est attribué à chaque pixel pour écarter les parties occultées : au niveau le plus détaillé, la borne inférieure est la profondeur du point et la borne supérieure est la profondeur du point à laquelle est ajoutée le rayon correspondant. Un pixel est considéré occulté si sa profondeur n'appartient pas à l'intervalle de profondeur du pixel le plus proche à l'écran. Dans les niveaux suivants de la pyramide, cet intervalle correspond aux valeurs de profondeurs minimum et maximum des pixels valides du niveau précédent. Un pixel est considéré valide si ces trois critères sont satisfaits : s'il est atteint par un point réellement visible, si l'ellipse de ce pixel recouvre le pixel à reconstruire et si sa profondeur est dans l'intervalle de profondeur du pixel ayant la plus petite valeur de profondeur. La région d'influence d'un pixel est limitée par une ellipse. Au niveau le plus détaillé, le centre de l'ellipse correspond aux coordonnées du point projeté sur l'espace écran ; l'axe majeur de l'ellipse fait deux fois le rayon projeté sur l'espace écran et l'axe mineur est égale à la longueur de l'axe majeur multiplié par la coordonnée  $z$  du vecteur normal projeté. L'orientation de l'ellipse est définie par le vecteur normal. La position du centre des ellipses varie relativement au niveau de la pyramide (figure 3.4.a). En pratique les ellipses sont utilisées seulement lors de la phase du push, cependant leur paramètres doivent être remontés pour permettre leur utilisation lors de l'étape suivante. Les attributs des pixels de l'image de résolution inférieure correspondent à une moyenne des valeurs des pixels valides de l'image de résolution supérieure.

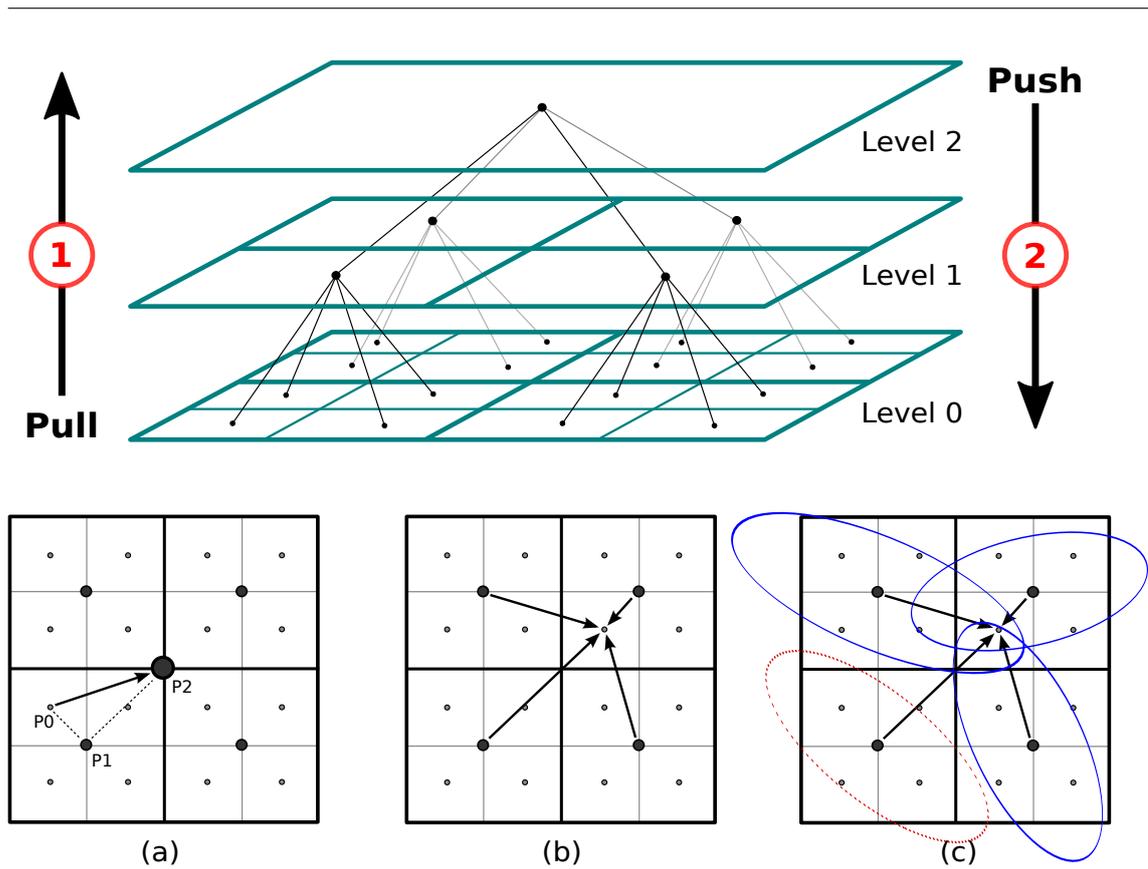


FIGURE 3.4 – L’algorithme proposé par [Marroquim *et al.*, 2007]

Top : La pyramide d’images illustrant (1) le pull moyenne les pixels valides de l’image de haute résolution pour calculer l’image de faible résolution, et (2) Le push propage les informations de l’image de faible résolution vers l’image de haute résolution (a) Le pull : la position du centre de l’ellipse varie relativement au niveau de la pyramide; (b-c) le push : les pixels du niveau supérieur (en gras) sont utilisés pour reconstruire les pixels du niveau inférieur. Dans certains cas l’ellipse ne recouvre pas le pixel à reconstruire.

**Phase de descente (push)** L’objectif est de propager les informations de l’image de plus faible résolution (celle qui correspond à la racine de la pyramide) vers l’image de plus haute résolution (le niveau le plus détaillé) afin de combler les trous ou de recalculer les pixels occultés. Un pixel est considéré occulté si sa profondeur n’est pas dans l’intervalle de profondeur du pixel correspondant dans le niveau supérieur. Les données sont propagées en un pixel à partir des quatre pixels valides (figure 3.4.b). Pour chacun de ces pixels, l’ellipse représentant sa zone d’influence est utilisée pour déterminer s’il contribue à la reconstruction ou non. Lorsque l’ellipse ne recouvre pas le pixel à reconstruire, ses paramètres ne sont pas propagés, et elle ne contribue pas à la reconstruction du pixel (figure 3.4.c).

---

Pour résumer, cette méthode est un atout grâce aux caractéristiques suivantes :

- Reconstruction relativement indépendante de la complexité géométrique de la scène.
- Reconstruction lisse des attributs grâce à l'utilisation de filtres.
- Possibilité d'appliquer un modèle d'éclairage (*deffered shading*) grâce à la reconstruction des normales de la surface. Dans le cadre de notre projet, cela permet d'écarter la contribution des caméras formant un angle razan (*grazing angle*) avec le nouveau point de vue lors de la visualisation .

En contrepartie, pour les nuages de points qui ne sont pas assez denses ou qui n'ont pas une densité uniforme, la méthode génère des clignotements sur la silhouette de l'objet lorsque le point de vue se déplace. De plus, elle échoue en cas d'agrandissement (lorsque la caméra est trop proche de la surface de l'objet). Cela dû à la nature approximative de l'interpolation qui reconstruit les splats exclusivement dans l'espace écran (si le point se déplace d'un pixel, il pourrait être interpolé différemment). En fait, le pull cherche toutes les combinaisons possibles, plus on se rapproche de l'objet, plus l'intervalle de profondeur associé à chaque splot (ellipse) devient large, vu qu'il est calculé en fonction de l'inverse de la distance entre le point et la caméra. Dans ce cas, les attributs des splats de la couche de la surface cachée peuvent être remontés et lors du push, ils sont interpolés avec les splats de la surface du premier plan (figure 3.5.g). Les améliorations proposées dans [Marroquim *et al.*, 2008] réduisent certains artefacts, mais le coût de calcul est relativement élevé.

Nous proposons une méthode plus simple permettant d'une part de réduire les artefacts de clignotement sur la silhouette de l'objet et d'autre part d'éviter les interpolations de profondeur indésirables. Elle est basée sur deux passes de rendu supplémentaires. Le principe est d'utiliser plusieurs tampons et de rendre chaque couche de surface dans un tampon différent :

- Dans la première passe, rendre les points avec une largeur correspondant à l'ellipse projetée sur l'espace écran. Appliquer le back-face culling comme premier test de rejet. Pour gérer le problème de projection de plusieurs splats

---

sur les mêmes pixels, activer le test de profondeur pour ne garder que les pixels les plus proches à l'écran (ayant la plus petite valeur profondeur)

- Stocker la carte de profondeur produite dans un framebuffer. Cette carte agit comme un masque binaire défini par les ellipses. Notez que dans ce cas, la profondeur est constante sur tous les pixels de l'ellipse pour un point donné.
- Dans la deuxième passe, utiliser la profondeur des points stockée dans  $T_{zc}$  pour classer les points en back-points et front-points : Comparer la profondeur de chaque point projeté avec la profondeur du pixel correspondant sur la carte produite dans la première passe. Ce point est copié dans un framebuffer "BackgroundFB" si la différence est très grande, cela veut dire que ce point est occulté par les splats. Sinon, il est copié dans un autre framebuffer appelé, "ForegroundFB".
- Appliquer le pull push séparément sur "BackgroundFB" et "ForegroundFB".
- Utiliser la carte de profondeur de la première passe pour éviter la reconstruction des pixels qui tremblotent sur la silhouette de l'objet. Si un pixel est reconstruit lors de la première passe, on le reconstruit lors du push. Sinon, on l'écarte par un simple discard.
- Fusionner les deux cartes de profondeurs produites. Notez que la carte de profondeur "BackgroundFB" est seulement utilisée pour combler certains trous.

Lors de la première passe, chaque point est projeté sur l'espace écran. La largeur du point (*pointSize*) est deux fois le rayon projeté  $r_{proj}$ , estimé de la même manière que celui de Marroquim [Marroquim *et al.*, 2007] :

$$r_{proj} = r \cdot \frac{f}{d_z} \cdot h$$
$$f = \frac{1}{2 \cdot \tan(\frac{fov}{2})}$$

Où  $d_z$  correspond à la distance entre le centre du point et la caméra,  $f$  est la longueur focale, calculée en fonction de l'angle de vue ( $fov$ ) et  $h$  correspond la hauteur du viewport.

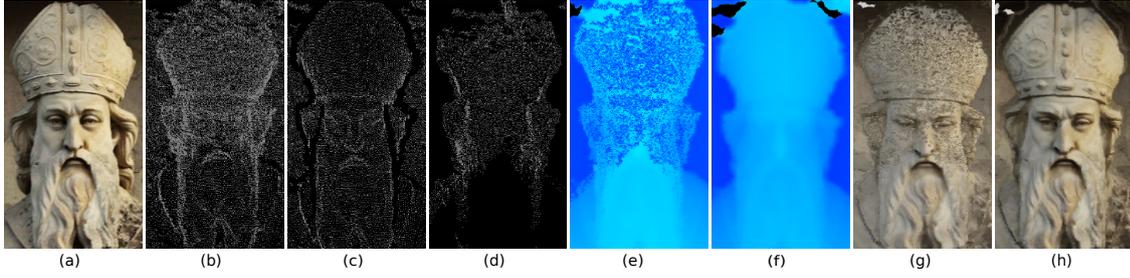


FIGURE 3.5 – **Corrections de la profondeur**

(a) Région sélectionnée sur l'une des photos d'origines; (b) Rendu du nuage de points correspondant; (c) et (d) Séparation du nuage de points en *front-points* et *back-points* respectivement en utilisant notre première passe; (e) Profondeur reconstruite par l'algorithme de Marroquim *et al.* [Marroquim *et al.*, 2007] (f) Profondeur reconstruite avec la prise en compte des corrections; (g) et (h) images finales synthétisées en utilisant les cartes de profondeurs (e) et (f) respectivement.

Un carré centré sur la position du point projeté, englobant l'ellipse correspondante est rastérisé. Pour chaque pixel de ce carré, le fragment shader détermine s'il fait partie ou non de l'ellipse. La longueur du demi grand axe  $a$  de l'ellipse correspond au rayon projeté  $r_{proj}$  et la magnitude du demi petit axis  $b$  correspond à la longueur du demi grand axe multiplié par la coordonnée  $z$  du vecteur normal projeté  $N_z$ . Un pixel  $(x,y)$  appartient à l'ellipse si :  $\frac{d_x^2}{a^2} + \frac{d_y^2}{b^2} \leq 1$ , avec  $d_x$  et  $d_y$  correspondent à la distance du pixel au centre du carré, exprimée dans le système de coordonnées de l'ellipse.

Ce processus est employé pour construire une carte de profondeur, associée à chaque nouveau point de vue lors de la visualisation. Il est également appliqué aux photographies d'origines, pour leur associer une carte de profondeur, utilisée pour gérer les occultations.

### 3.2.2 Gestion des occultations

Grâce à la reconstruction de la profondeur à partir des points pour un point de vue  $C$  donné, il est possible de déterminer la position 3D  $P$  de la surface visualisée à travers chaque pixel. Notre objectif est d'exploiter cette information supplémentaire pour résoudre les problèmes d'occultations décrits sur la figure (3.3.b).

Lors de la visualisation, à la manière de la méthode des cartes d'ombres (shadow mapping [Williams, 1978]), le point  $P$  est projeté sur les photographies d'origines

---

**Algorithm 1** Pseudo algorithm de notre nouvelle passe de rendu basé point

---

```
//Entrée :  
//N : vecteur normal projeté.  
//Cx,Cy : coordonnées du point projeté (centre du carré).  
//rproj : rayon projeté.  
//z : profondeur du point dans l'espace caméra.  
a ← rproj; //demi grand axe de l'ellipse  
b ← a * max (Nz , 0.5); // demi petit axe  
// orientation de l'ellipse  
l ← length (Nxy);  
if (l == 0) then  
    Ny = 0.0;  
else  
    Ny ← Ny /l;  
end if  
cosAngle = Ny;  
sinAngle = sqrt (1.0 - cosAngle * cosAngle);  
// la prise en compte du signe de Nx  
if (Nx > 0) then  
    sinAngle = -sinAngle;  
end if  
// Rotation du point  
diffx = (x - Cx);  
diffy = (y - Cy);  
dx = (diffx * cosAngle + diffy * sinAngle);  
dy = (-diffx * sinAngle + diffy * cosAngle);  
test = (dx2 /a2 + dy2 /b2);  
if (test > 1.0) then  
    discard;  
else  
    color = z;  
end if
```

---

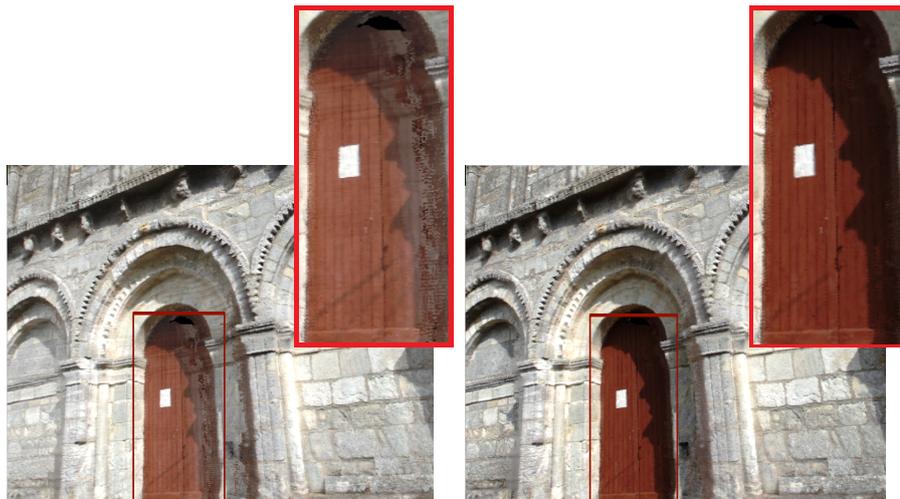


FIGURE 3.6 – **Gestion des occultations pour un nouveau point de vue  $C$**

A gauche : Toutes les photographies sont utilisées pour produire l'image, et des effets d'objets fantômes apparaissent. A droite : même image avec la prise en compte des occultations, quelques photographies ont été écartées ;

(définies par  $C_k$ ), et la couleur finale des pixels de  $C$  est estimée grâce à un système de cohérence épipolaire, qui combine les valeurs des pixels qui sont considérés valides en terme de profondeur. Pour cela, plusieurs critères doivent être satisfaits :

- Le point  $P$  doit être projeté dans le champ de vision de la caméra  $C_k$  ;
- La profondeur correspondant au point  $P$  projeté sur chaque image  $C_k$  est comparée à la profondeur effective de l'image. Lorsque la profondeur coïncide, la valeur du pixel correspondant est considérée valide pour le pixel reconstruit sur l'image  $C$  ; dans le cas contraire, l'image  $C_k$  est rejetée pour ce pixel ;
- L'angle entre la direction d'observation (noté :  $\mathbf{V}_k$  pour la caméra  $C_k$ ) et le vecteur normal  $\mathbf{N}_p$  du point  $P$  doit être inférieur à  $\pi/2$ . Cela permet d'éliminer la contribution des photographies formant un angle razan avec le point de vue  $C$  :  $\mathbf{N}_p \cdot \mathbf{V}_k < \lambda$ , avec  $\lambda$  est fixée à  $10^{-4}$  dans notre implémentation et  $\mathbf{N}_p$  est estimé à partir du nuage de point, pendant le pull push.

Les figures 3.6 et 3.7 illustrent la différence obtenue lorsque les occultations sont prises en compte ou non.

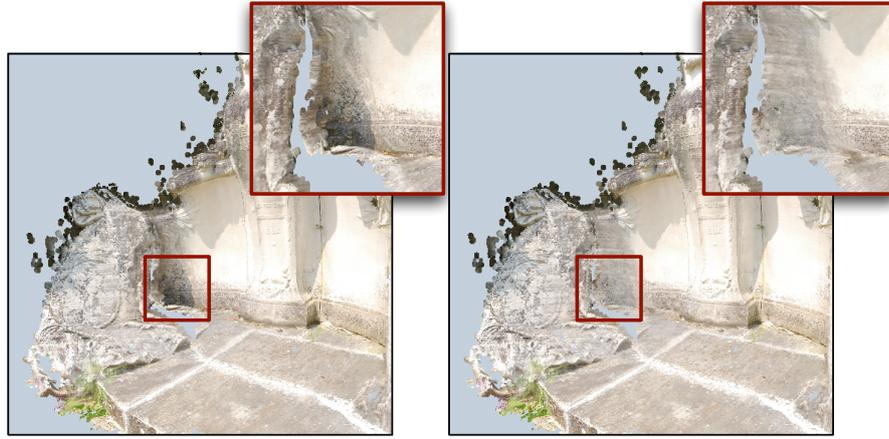


FIGURE 3.7 – L’effet de la prise en compte des occultations sur les nouvelles vues

A gauche : l’image montre la reconstruction avec la prise en compte des occultations, l’ombre est reproduit sur la nouvelle vue. A droite : même image sans prise en compte des occultations, un effet de lissage est apparait car certains points de vue ne devraient pas être pris en compte ;

### 3.2.3 Cohérence épipolaire

De nombreux systèmes de reconstruction échouent pour des matériaux très brillants, car l’analyse des descripteurs n’est plus adaptée ; la luminance reçue par une caméra dépend essentiellement de la luminance provenant de la direction miroir, dont les variations sont très dépendantes du point de vue. Cependant, ces systèmes restent robustes pour des objets non parfaitement lambertiens. Les approches de rendu basé images prennent en compte la déviation angulaire entre les directions d’observations pour reproduire les effets dépendantes du point de vue et préserver un maximum de détails. Notre modèle de cohérence épipolaire est inspiré des lumigraphes non structurées [Buehler *et al.*, 2001]. Cependant, au lieu de fixer un seuil, nous choisissons d’utiliser toutes les photographies qui peuvent contribuer à la construction du pixel  $C$ . Cela permet d’éviter les effets de clignotement sous forme de pixels noirs lorsque le point de vue change. De plus, plutôt que des pénalités, nous préférons introduire une pondération permettant de donner plus d’importance aux photographies les plus pertinentes :

$$I(i, j) = \frac{1}{\sum_{k=1}^N \langle \cos \theta_k \rangle} \sum_{k=1}^N I_k(u_k, v_k) \langle \cos \theta_k \rangle, \quad (3.1)$$

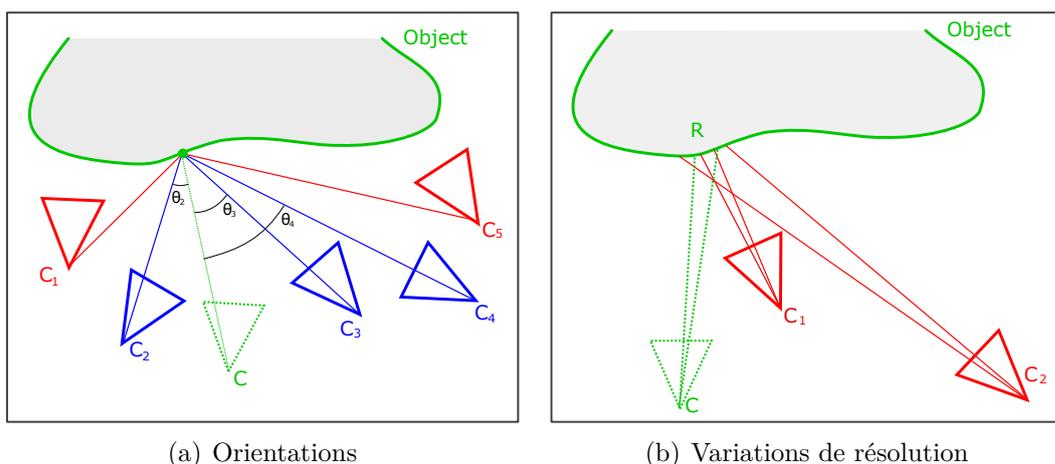


FIGURE 3.8 – **Cohérence épipolaire**

(a) L'apparence associée à  $P$  observé depuis une caméra  $C$  peut être reconstruite à partir de nombreux points de vues ; les effets dépendants du point de vue dépendent de l'angle d'observation (b) Les positions relatives des caméras introduisent également des différences de résolution, de sorte que les distances doivent également être prises en compte.

avec  $I(i, j)$  le pixel courant de la nouvelle vue  $C$  correspondant au point  $P$  de l'objet visualisé,  $I_k(u_k, v_k)$  le pixel correspondant à la projection du point  $P$  sur l'image  $I_k$ , et  $\theta_k$  est l'angle entre les directions d'observations  $\mathbf{V}$  et  $\mathbf{V}_k$  du nouveau point de vue et de la photographie d'origine respectivement figure (3.8.a).

$$\langle \cos \theta_k \rangle = \max(0, \text{dot}(\mathbf{V}, \mathbf{V}_k)), \quad (3.2)$$

avec  $\mathbf{V}_k = \frac{C_k^c - P}{\|C_k^c - P\|}$ ,  $C_k^c$  correspond au centre de projection de la caméra  $C_k$ . Cette pondération permet de prendre en compte des objets non lambertiens, par exemple légèrement brillant.

La distance entre l'objet et la caméra est un autre critère important car elle permet de reproduire certains détails sur la nouvelle image lorsque le nouveau point de vue se rapproche de l'objet. L'objectif est d'associer un poids plus élevé aux photographies d'origines qui sont proches du nouveau point de vue, ce qui fournit d'une manière adaptative une définition d'une texture adéquate, avec de détails plus nets et plus précis. En pratique, la valeur  $(R, G, B)$  d'un pixel correspond à l'intégrale de la luminance réfléchiée par les objets observés à travers le pixel. La figure (3.8.b) illustre l'observation d'une même région  $\mathcal{R}$  avec deux photographies

---

$C_1$  et  $C_2$  à travers les deux pixels  $I_1(u_1, v_1)$  et  $I_2(u_2, v_2)$  respectivement. L'angle solide correspondant aux pixels ne recouvre pas parfaitement la même surface de l'objet et par conséquent, la luminance réfléchiée par une même surface et dans la même direction peuvent ne pas être identique (3.8.b) car la surface observée n'est pas exactement la même.

Pour prendre en compte cette différence de résolution, nous prenons en compte la distance entre le point observé  $P$  et chacune des positions des photographies, en introduisant la pondération suivante :

$$I(i, j) = \frac{1}{\sum_{k=1}^N \frac{1}{\delta(C, P, C_k) + 1}} \sum_{k=1}^N \frac{I_k(u, v)}{\delta(C, P, C_k) + 1}, \quad (3.3)$$

avec  $\delta(C, P, C_k)$  est la différence de distance entre le point  $P$  et les centres de projection des caméras  $C^c$  et  $C_k^c$ .

$$\delta(C, P, C_k) = \text{abs}(\|C^c - P\| - \|C_k^c - P\|), \quad (3.4)$$

avec  $C^c$  et  $C_k^c$  correspondent aux centres de projection de  $C$  et  $C_k$  respectivement.

La profondeur et la gestion des occultations permet de déterminer pour chaque pixel de l'image  $C$ , l'ensemble des photographies valides et des pixels correspondants. Pour la couleur finale des pixels de  $C$ , notre méthode consiste à combiner les valeurs des pixels à l'aide des poids donnés, pour prendre en compte les priorités d'orientation et de résolution :

$$I(i, j) = \frac{1}{\sum_{k=1}^N \frac{\langle \cos \theta_k \rangle}{\delta(C, P, C_k) + 1}} \sum_{k=1}^N \frac{I_k(u, v) \langle \cos \theta_k \rangle}{\delta(C, P, C_k) + 1}. \quad (3.5)$$

Pour éviter les changements conversions d'espace couleur, tous les calculs d'accumulation et de moyenne couleur sont réalisés dans l'espace de couleur RVB standard. Pour la construction d'images à haute gamme de dynamique (HDR), l'ensemble du processus de visualisation décrit ci-dessus est appliqué pour chaque série d'images. L'image finale correspond à la combinaison par pixel des niveaux de luminance (tone-mapping) ;

---

**Algorithm 2** Pseudo algorithme pour la gestion d'occultations et la cohérence épipolaire

---

```
//Entrée :
//DMk,Mk : carte de profondeur et matrice de projection associés à Ck
//DM,M : carte de profondeur et matrice de projection associés à C.
//NP : vecteur normal associé à P
//V, Vk : directions d'observations correspondant à C et Ck respectivement ;
//Cc, Ckc : centres de projections de C et Ck respectivement ;
// récupérer la profondeur de C
Depth ← tex2D (DM , texCoord.xy);
// calculer les coordonnées dans l'espace NDC
clipPos ← (texCoord.xy , Depth) * 2.0 - 1.0;
// projection inverse dans l'espace objet.
P ← inverse(M) * clipPos;
// projection sur la caméra Ck
texPos ← P * Mk
if (NP.Vk < 10-4) then
    // supprimer la caméras vues sur des angles razan
    discard;
end if
if (texPos.xy > 1.0 || texPos.xy < 0.0) then
    // Si le point est projeté hors du champs de vision de Ck
    discard;
end if
// récupérer la profondeur de Ck
Depthk ← tex2D (DMk , texPos.xy);
if (Depthk < texPos.z * depthTreshold) then
    discard; //le point P est occulté de Ck
end if
// sinon le point est visible depuis Ck. Activer la fonction de mélange.
// déviation angulaire
cos θk ← abs(V.Vk);
//résolution
δ(C, P, Ck) ← abs(||Cc - P|| - ||Ckc - P||);
SumColor ← (Ik(u, v) * cos θk) / δ(C, P, Ck);
SumWeight ← 1 / (cos θk / δ(C, P, Ck));
// la couleur finale de C
I(i, j) ← SumColor * SumWeight;
```

---

### 3.3 Résultats

Nous avons testé notre méthode sur divers jeux de données. La figure 3.9 et le tableau 3.1 illustrent les quatre scènes LDR utilisées pour cette thèse (voir aussi la vidéo<sup>1</sup>) avec : le nombre de points issus des étapes d'étalonnage et de reconstruction, le nombre total de photographies utilisées durant la visualisation ainsi que l'occupation mémoire.

Tableau 3.1 – Description des scènes, avec mémoire requise.

Scène	# points	# img	Sommets	Textures
Cathedral	2,901,019	33	162 Mo	661 Mo
Church	1,233,795	20	69 Mo	401 Mo
Mirebeau	2,172,103	22	119 Mo	441 Mo
Bell	503,712	118	28 Mo	85 Mo



(a) Church



(b) Cathedral



(c) Mirebeau



(d) Bell

FIGURE 3.9 – Exemples de scènes affichées avec notre système de rendu basé image interactif.

1. <https://vimeo.com/307643834>

---

## Performance

La station utilisée pour tester nos scènes est équipée d'un processeur Intel Core I7 930 et d'une carte graphique NVIDIA GeForce 750 GTX. Le programme a été développé en Javascript, avec le WebGL 1.0. Les photographies utilisées étant d'une grande résolution (de  $1500 \times 1000$  à  $5600 \times 3700$ ), nous utilisons la compression de textures proposée par WebGL pour limiter l'occupation mémoire. Toutes les étapes de reconstruction sont réalisées sur le processeur central, mais notre système de visualisation est intégralement exécuté sur la carte graphique.

Le tableau 3.2 présente les temps de calcul des différentes parties de la méthode de rendu ainsi que du nombre d'images par secondes (fps) lors de la visualisation pour la résolution  $512 \times 512$  pixels. La colonne *Initialisation* correspond à l'étape de projection du nuages de points sur les photographies d'origines et à l'étape de reconstruction de la profondeur. La colonne *Visualisation* correspond à la projection des points sur le nouveau point de vue, la nouvelle passe de splatting, la reconstruction de la profondeur et à la reprojexion sur les photographies. Les performances dépendent essentiellement de l'étape de rasterisation (c'est à dire du nombre de points pour le proxy géométrique) et du nombre d'images. Le temps de reconstruction est constant puisqu'il ne dépend que de la résolution de l'image.

<i>Scène</i>	<i>Initialisation</i> (ms)	<i>Visualisation</i> (ms)	<i>FPS</i>
Church	100	15	66
Cathedral	167	17	58
Mirebeau	91	13	76
Bell	100.5	8	111

Tableau 3.2 – Temps de calcul pour une résolution de  $512 \times 512$ .

La figure 3.10 montre que notre système est capable de générer de nouveaux points de vue qui sont largement différents des points de vue d'origines même dans le cas d'un petit nombre de photographies. Ainsi, grâce à notre système de cohérence épipolaire qui prend en compte les déviations angulaires, les effets dépendant du point de vue (les variations de la lumière) sont reproduits lorsque l'observateur se déplace autour de l'objet (voir l'exemple de la cloche sur la figure 3.11).



FIGURE 3.10 – Nouvelles images synthétisées par notre système de navigation interactif, basé sur 22 photographies.



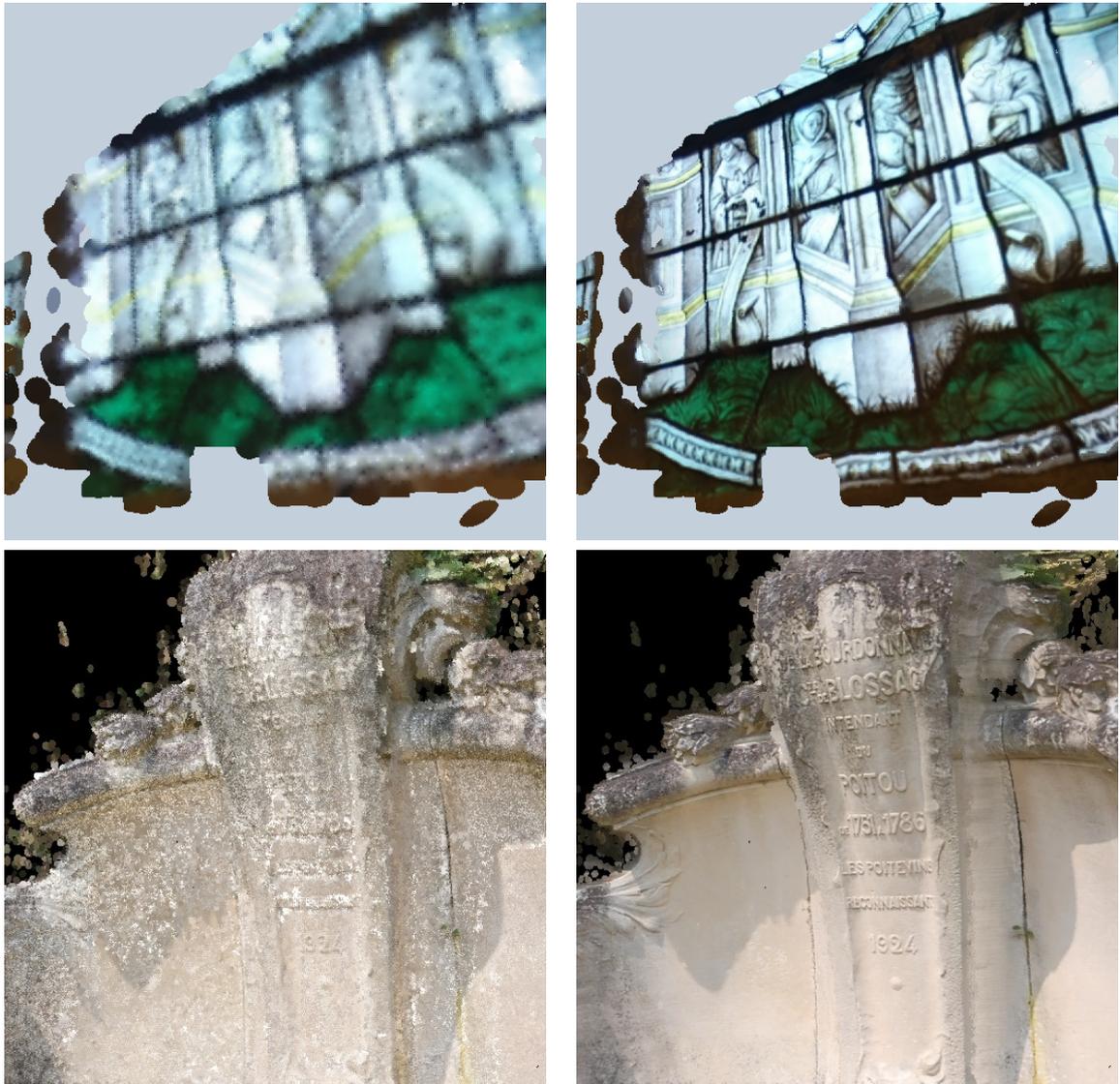
**FIGURE 3.11 – Les effets dépendant du point de vue**

les zones lumineuses se déplacent en fonction de l'observateur. En haut : deux zones de l'objet sont encerclées. La lumière visible sur les régions jaune et bleue disparaît sur l'image de droite (le texte en haut de la cloche peut être utilisé comme référence). En bas : la lumière dans la région jaune se déplace à droite en fonction du point de vue.

La figure 3.12 montre le gain de notre méthode par rapport à un système de rendu basé point pour lequel la couleur de chaque pixel est interpolée uniquement à partir des couleurs des points voisins de l'objet. notre méthode bénéficie à la fois de

---

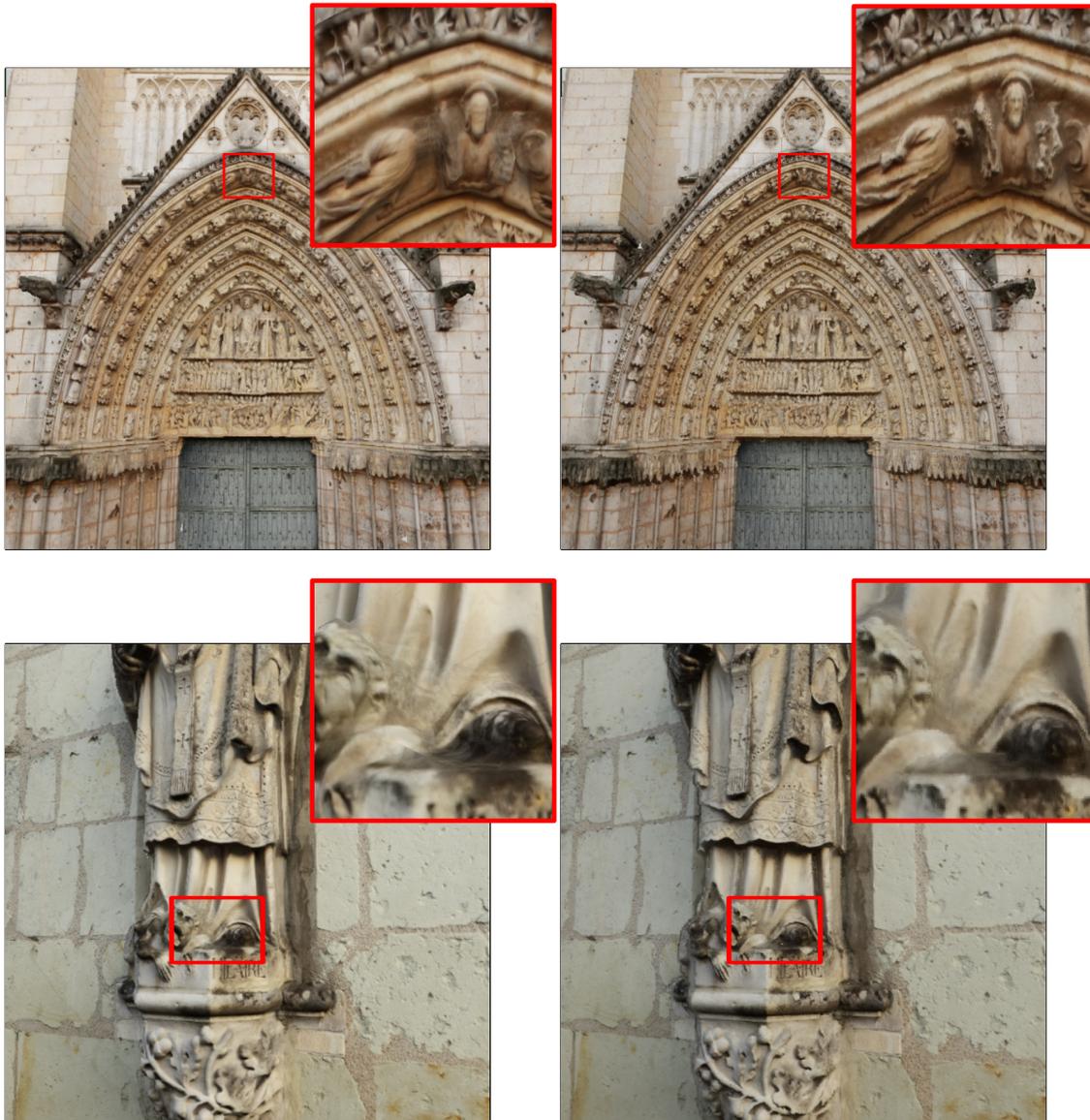
la rapidité de la projection du nuage de points avec le processus de rasterisation et de la haute qualité de l'image produite vu que la texture n'est pas fixée à priori sur le modèle 3D (Les photographies d'origines sont mélangées en fonction de la position de l'observateur).



**FIGURE 3.12 – Gain de notre méthode en terme de qualité**  
Rendu basé-points (à gauche) ; notre approche de rendu à base d'images (à droite).

Nous avons également comparé notre méthode avec les lumigraphs non structurés [Buehler *et al.*, 2001] auxquels nous avons rajouté notre système de gestion d'occultations. Nous avons reconstruit des maillages triangulaires pour les deux scènes *Mirebeau* (2.174 million triangles) et *Cathédrale* (1.875 million triangles) à partir du nuages de points correspondants en utilisant la méthode de reconstruction

de surfaces proposée par Kazhdan *et al.* [Kazhdan *et al.*, 2006]. La figure 3.13 illustre des artefacts visuels (effets d'objets fantômes) sur les images produites même avec la prise en compte des occultations car le proxy utilisé pour construire les cartes de profondeur est erroné.



**FIGURE 3.13 – Fiabilité des nuages de points par rapport aux maillages géométriques**

Comparaison de notre approche (à droite) avec les lumigraphes non structurées (à gauche) sur la cathédrale et Mirebeau (de haut en bas).



(a) 5616×3744 pixels (original), 456 Mo

(b) 1/4 size, 297 Mo



(c) 1/16 size, 257 Mo

FIGURE 3.14 – Chillac visualisée en utilisant des photographies à différentes résolutions, avec l’occupation mémoire correspondante.

### 3.4 Limites

Notre mise en oeuvre nécessite de stocker toutes les photographies sur le GPU, potentiellement avec une place mémoire importante (en fonction du nombre d’images et de leur résolution). Cela peut être un frein pour la visualisation lorsque le nombre de photographies devient élevé. Pour permettre de réduire les coûts en mémoire, nous avons déjà utilisé la compression de textures car elle permet de diviser par

---

quatre l'occupation mémoire des photographies. Une autre possibilité est de réduire la résolution des photographies d'origines, mais cela peut nuire à la qualité de la visualisation (voir la figure 3.14). Nous avons pu stocker jusqu'à un maximum de 200 photographies en mémoire de la carte vidéo. Nous pensons qu'il est également possible de réaliser, en cas de besoin un système de chargement dépendant du point de vue pour réduire le nombre de photographies en mémoire selon la position et l'orientation du point de vue. Cependant, la mise en œuvre d'un tel système peut affecter les performances lors des échanges de données entre la mémoire centrale et le processeur graphique.

La deuxième limite concerne les clignotements sur la silhouette de l'objet. Nous avons proposé des passes de rendu supplémentaires pour réduire ces effets. Ces passes nous procurent des résultats très satisfaisants, mais elles ne prennent en charge que les clignotements observés sur l'arrière plan (background). Elles ne tiennent pas compte des parties de l'objet qui se projettent sur le nuage de points lui-même.

Le chapitre suivant montre comment peut-on exploiter notre système de visualisation pour construire des images à haute gamme de dynamique, à partir de plusieurs séries d'images LDR, prises à main levée.

# Chapitre 4

## Reconstruction d'images à haute gamme de dynamique (HDR)

La reconstruction d'images à haute gamme de dynamique (HDR) nécessite plusieurs prises de vue pour chaque position d'observation [Debevec et Malik, 1997]. Dans ce chapitre, nous montrons que notre système de visualisation (chapitre précédent, 3) permet de les reconstruire d'une manière automatique à partir de plusieurs séries indépendantes de photographies non HDR, prises à main levée par des caméras standards ; chaque série ayant un niveau d'exposition fixe. L'idée est d'exploiter ce système pour produire pour chaque exposition, une nouvelle image LDR alignée sur un point de vue donné en se basant sur les séries de photographies non structurées. Nous discutons dans ce chapitre du processus complet de reconstruction d'images HDR en utilisant notre méthode de rendu basé image ainsi que des stratégies de reconstruction envisagées, en mettant l'accent sur les avantages et les limites de chacune d'elles.

### 4.1 Présentation du système

La figure 4.1 présente l'architecture logicielle de notre méthode de reconstruction d'images HDR à l'aide de notre système de visualisation.

Avec plusieurs séries de photographies non structurées dont l'exposition varie. Les processus d'étalonnage et de production de nuage de points [Furukawa et Ponce, 2010,

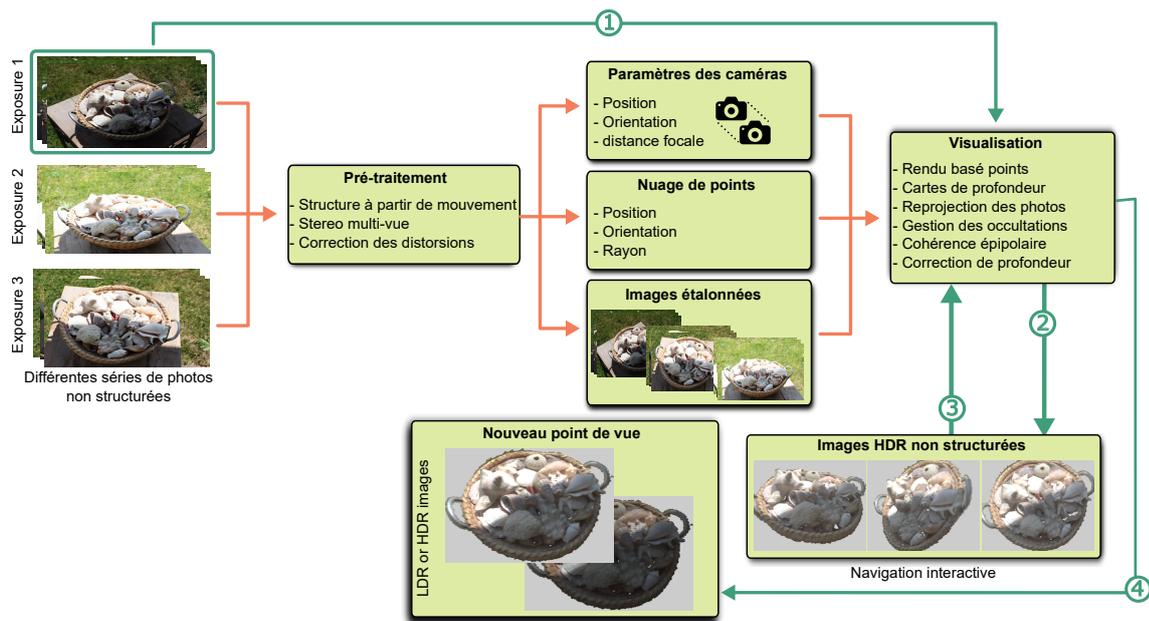


FIGURE 4.1 – Notre système de reconstruction d’images HDR

plusieurs séries de photographies LDR non structurées, ayant chacune un niveau d’exposition fixe sont utilisées en entrée du pré-traitement ; Le résultat produit est un nuage de points dense ainsi qu’une matrice de projection pour chacune des photographies ; (1) Les points de vue de l’une des séries de photographies sont utilisées pour aligner les images des autres séries (2) ; Grâce à notre processus de rendu. Les images LDR alignées sont utilisées pour produire des images HDR. (3) Les images HDR produites peuvent être utilisées en entrée de notre système de rendu pour générer de nouvelles vues HDR quelconques (4). Notez que notre système permet de reproduire les effets dépendants du point de vue et d’améliorer les régions sous-exposées ou sur-exposées grâce au processus de reconstruction des images HDR

Moulon *et al.*, 2016b] (voir section 3.1 du chapitre 3) sont appliqués sur l’ensemble de photographies afin de l’étalonner et produire un nuage de points unique. Pour un point de vue donné  $C$ , et pour chaque exposition, une image LDR peut être synthétisée à l’aide de notre système de visualisation. Cela produit un ensemble de nouvelles images LDR alignées sur  $C$  (voir la figure 4.2). La fusion de ces images produit l’image HDR à afficher. Ce processus peut être effectué à la volée, mais de meilleures performances peuvent être obtenues lors d’une étape de pré-calcul dont l’objectif est de reconstruire une unique série d’images HDR alignées sur les points de vue de l’une des séries de photographies (voir section 4.4). Ainsi, les images HDR résultantes peuvent ensuite être utilisées pour la navigation interactive, avec une exposition définie par l’utilisateur.

Idéalement, chaque série de photographies doit bien couvrir la surface de l'objet à visualiser. Pour cela, nous avons choisi de sélectionner arbitrairement les points de vue de l'une des séries de photographies LDR pour construire les images HDR.

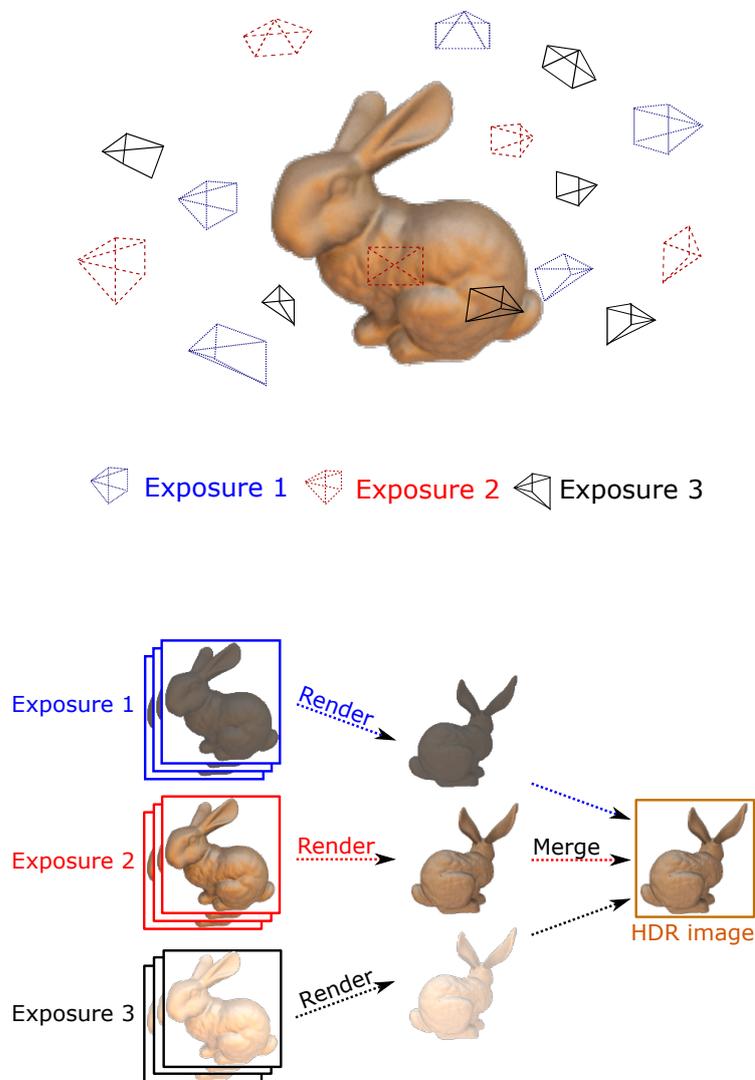


FIGURE 4.2 – **Reconstruction d'une image HDR à partir de 3 séries de photographies non structurées, ayant Chacune un niveau d'exposition fixe**

Pour un point de vue donné, une image est construite grâce à notre pipeline de rendu.

Soit  $M$  le nombre de séries de photographies  $\{S^1, S^2, \dots, S^M\}$ , ayant chacune un niveau d'exposition fixe. Notre objectif est de reconstruire une série de  $K$  images HDR alignées sur un ensemble de  $K$  points de vue  $\{C_{i \in [1..K]}^{new}\}$ .

Pour chaque point de vue  $C_i^{new}$ ,  $M$  images LDR  $\{I_i^1, I_i^2, \dots, I_i^M\}$  sont synthétisées à l'aide de notre pipeline de rendu (voir les exemples illustrés sur la figure 4.3).

La fusion de ces  $M$  images produit l'image HDR  $I_i^H$ . Les  $K$  photographies de la série choisie peuvent être alors remplacées par les  $K$  images HDR, les autres séries deviennent alors inutiles.

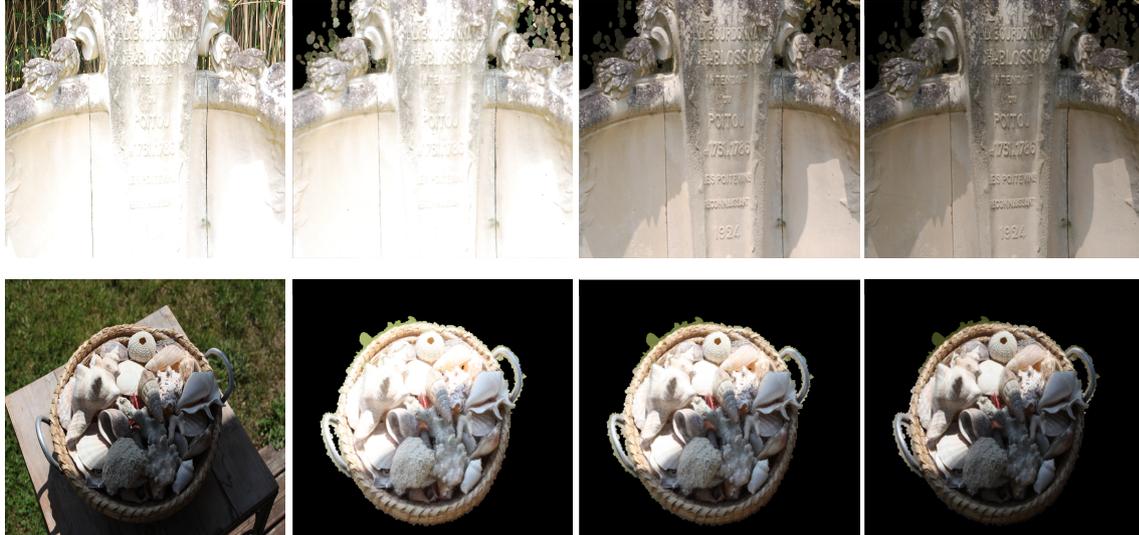


FIGURE 4.3 – Trois images LDR avec des niveaux d'expositions différents produites par notre système de visualisation pour les deux scènes : statue et coquillages

les trois images sont bien alignées sur le point de vue de la photographie originale.

## 4.2 Méthode pour fusionner les images synthétisées en une image HDR

Nous avons utilisé le processus standard de fusion proposé par Debevec et Malik [Debevec et Malik, 1997] pour construire l'image HDR  $I_i^H$  pour le point de vue  $C_i^{new}$ . Néanmoins, tout autre algorithme de fusion pourrait également être utilisé.

La méthode de Debevec [Debevec et Malik, 1997] se base sur le principe de réciprocité. Chaque caméra transforme l'énergie lumineuse qu'elle reçoit via une fonction qui s'appelle la fonction de réponse  $f$ . L'application de l'inverse de la fonction  $f$  sur l'image reproduit l'éclairage réel. Les auteurs proposent dans un premier temps une méthode pour déduire cette fonction, ils reconstruisent ensuite l'image HDR en combinant les images en entrée transformées, par  $f^{-1}$ , en fonction du temps d'exposition qui leur est associé, . L'image HDR ainsi créée est codée en flottants.

---

La plage est donc bien supérieure par rapport à celle utilisée pour les images LDR et les couleurs sont bien représentées peu importe claires ou sombres.

Théoriquement, la couleur de chaque pixel  $I(x, y)$  d'une exposition est obtenue par le produit de l'énergie lumineuse  $E(x, y)$  par le temps d'exposition  $\Delta t$ . L'image étant numérique, les valeurs obtenues sont transformées par la fonction de réponse  $f$  et représentées par des pixels :  $I(x, y) = f(E(x, y) \cdot \Delta t)$ .

L'image HDR correspond à l'estimation de l'énergie  $E$  au pixel  $(x, y)$  :

$$E(x, y) = \frac{f^{-1}(I(x, y))}{\Delta t}. \quad (4.1)$$

Pour déterminer  $E(x, y)$  en chaque point de l'image, il faut déterminer d'abord la fonction  $g$ , le logarithme népérien de la fonction inverse  $f^{-1}$  :

$$g(I(x, y)) = \ln(f^{-1}(I(x, y))) = \ln(E(x, y)) + \ln(\Delta t). \quad (4.2)$$

Les inconnus de cette équation sont  $g$  et  $E(x, y)$ . Une décomposition en valeurs singulière permet de résoudre le système d'équations. Ce système étant largement sur-déterminé, les auteurs choisissent de sélectionner quelques pixels de l'image. Les pixels choisis doivent être bien répartis dans l'image et représentent une grande variété de couleurs. Ceci donne un système de  $M * N$  équations,  $M$  étant le nombre d'expositions et  $N$  le nombre de pixels choisis. Ils ont introduit également une fonction de pondération  $w$  pour ne pas prendre en compte les valeurs proches des bornes. Cette fonction vaut :  $w = z$  pour  $z$  appartenant à l'intervalle  $[0-127]$  et  $w = 256 - z$  pour  $z$  appartenant à  $[128-255]$ .

L'équation résolue par décomposition en valeurs singulières est :

$$O = \sum_{(x,y)}^N \sum_i^M (w(I^i(x, y)) [g(I^i(x, y)) - \ln(E(x, y)) - \ln(\Delta t)])^2 + \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2. \quad (4.3)$$

---

où  $Z_{min} = 0$  et  $Z_{max} = 255$

$$g''(z) = g(z - 1) - 2g(z) + g(z + 1) \quad (4.4)$$

L'image de luminance correspondant à l'image HDR, peut être ensuite reconstruite en appliquant l'équation suivante pour chacun des pixels  $(x,y)$  :

$$\ln(E_{(x,y)}) = \frac{\sum_{i=1}^M w(I^i(x,y)) [g(I^i(x,y)) - \ln(\Delta t)]}{\sum_{i=1}^M w(I^i(x,y))}, \quad (4.5)$$

### 4.3 Reproduction de tons "*tone mapping*"

Les images HDR ont une plage dynamique très grande comparant à la plage dynamique de l'écran qui est beaucoup plus faible. Pour cela, de nombreux opérateurs de reproduction de tons ont été proposés pour compresser la luminance des images HDR et les afficher sur des écrans LDR. Nous avons choisi celui de Reinhard [Reinhard *et al.*, 2002], car il est fréquemment utilisé dans les applications temps réel.

Soit  $I^H$  l'image HDR et  $N$  le nombre total de pixels. Reinhard converti d'abord la valeur (R,G,B) de chaque pixel en une luminance  $L$  :

$$L = 0.212 * I_R^H + 0.715 * I_G^H + 0.072 * I_B^H. \quad (4.6)$$

Il estime ensuite la luminance moyenne de la scène  $L_a$  par l'équation suivante :

$$L_a = \frac{1}{N} \exp\left(\sum_{(x,y)} \log(\epsilon + L(x,y))\right), \quad (4.7)$$

Où  $L(x,y)$  représente la luminance de la scène pour le pixel  $(x,y)$  et  $\epsilon$  est une valeur très faible pour éviter  $\log(0)$ .

La valeur de chaque pixel de l'image est ensuite mise à l'échelle, puis réduite à la plage  $[0,1]$  pour compresser les luminances élevées :

$$L_s(x,y) = \frac{k}{L_a} L(x,y), \quad (4.8)$$

---


$$L_d(x, y) = \frac{L_s(x, y)}{1 + L_s(x, y)}. \quad (4.9)$$

où  $L_s(x, y)$  est la luminance escaladé ,  $k = 0.18$  par défaut et  $L_d(x, y)$  correspond à la luminance compressée .

La couleur finale de chaque pixel de l'image HDR est donnée en fonction de *gamma* et *l'exposition*, deux paramètres à travers lesquels l'utilisateur peut déterminer tous les niveaux de luminance contenus dans le jeu d'images :

$$I(x, y) = (\text{exposure} * L_d(x, y))^{1.0/\text{gamma}}. \quad (4.10)$$

La luminance moyenne  $L_a$  étant calculée indépendamment pour chaque nouveau point de vue, des effets de clignotements peuvent apparaître lors de la navigation. En pratique, nous n'avons pas rencontré ces effets. Néanmoins, au cas où de telles variations apparaissent, une valeur fixe de  $L_a$  pourrait être estimée à partir de toutes les images HDR.

## 4.4 Les stratégies de reconstruction des images HDR

Deux stratégies de reconstruction peuvent être envisagées : (i) soit à la volée (lors de la visualisation), mais cela requiert des ressources de calcul supplémentaires vu que le nombre de séries augmente en fonction des niveaux d'expositions choisis ; (ii) soit lors d'une étape de pré-calcul dont l'objectif est de reconstruire une unique série de nouvelles images HDR alignées sur les points de vues de l'une des séries des photographies. Nous avons choisi la seconde option pour éviter de multiplier les traitements lors de la visualisation et augmenter par conséquent les performances du système.

La figure 4.4 montre la différence en terme de qualité entre les deux approches de reconstruction. la première effectue le processus entier à la volée ; Cela préserve mieux les détails mais prends plus de temps et requiert un espace mémoire supplémentaire sur les GPU, car toutes les photographies d'origines doivent être stockées.



**FIGURE 4.4 – Comparaison entre deux stratégies de rendu HDR**

à gauche : l'image HDR est reconstruite à la volée, y inclut l'étape de reproduction de tons ; les détails de l'image produite sont plus nets. A droite : l'image HDR est reconstruite après une étape de pré-calcul ; la qualité de l'image résultante est légèrement diminuée.

La seconde approche privilégie les performances et l'espace mémoire, mais les images produites sont légèrement floues à causes des deux phases de projections (l'une pour la production d'une unique série d'images HDR alignée, et l'autre pour le rendu de la nouvelle vue). Le temps de calcul de la première approche correspond à celui de la seconde multiplié par le nombre de séries auquel on rajoute le temps de reproduction de tons qui ne peut pas être négligé. La stratégie choisie est plutôt un compromis entre qualité, performance et espace mémoire.

## 4.5 Résultats

Nous avons testé notre approche sur trois jeux de données différents. Chacun est composé de trois ou quatre séries de photographies non structurées, ayant chacune un niveau d'exposition fixe. Le tableau 4.1 et la figure 4.5 présentent les trois scènes utilisées, avec : le nombre de points issus des étapes de pré-traitements, le nombre de séries, l'ouverture (le diaphragme) utilisé pour contrôler l'exposition lors de la capture, ainsi que le nombre de photographies employées pour les deux étapes de pré-calcul et de visualisation.

Scene	# series	# images		# 3D points	diaphragme (f-stops)			
		HDR	Visu		expo1	expo2	expo3	expo4
Statue	4	160	58	3'819'846	1.4	4.0	8.0	11
Seashell	3	44	20	222'228	2.0	4.0	8.0	-
Horse	3	55	23	117'891	2.0	4.0	8.0	-

Tableau 4.1 – Caractéristiques des scènes.

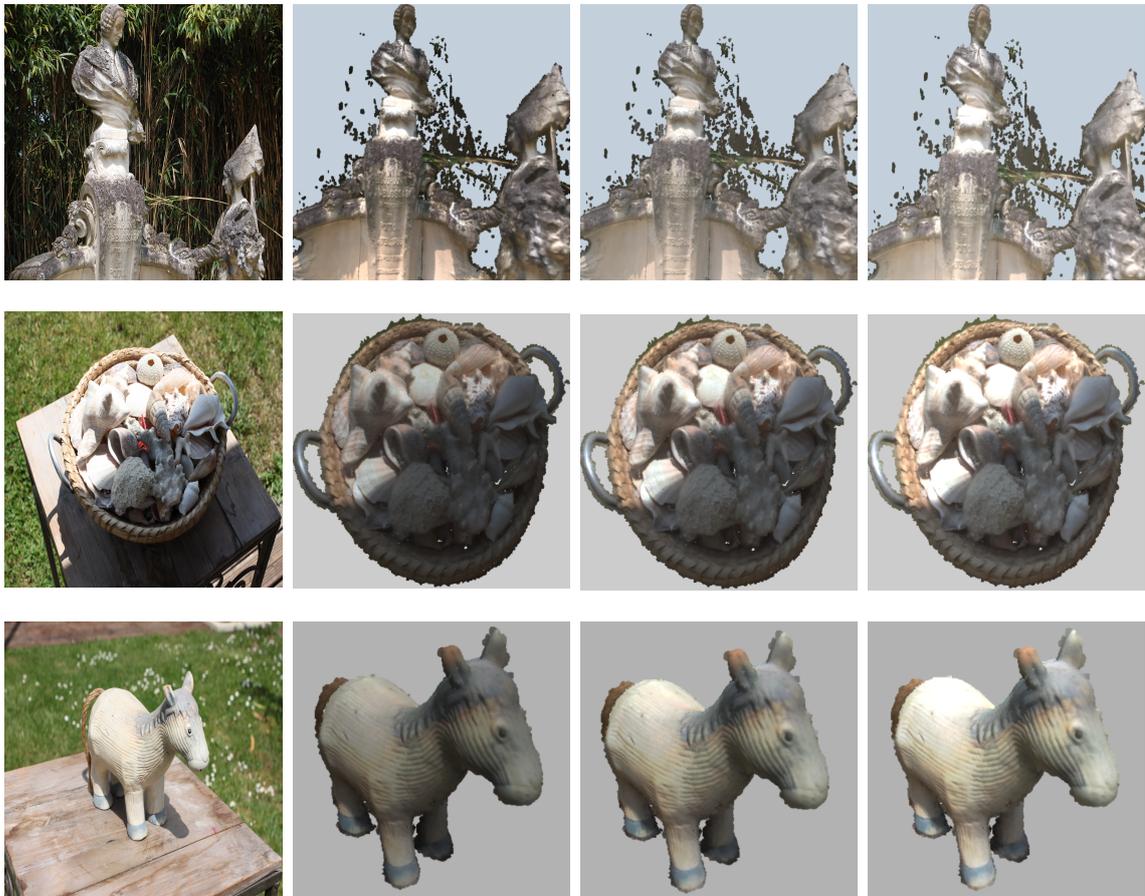


FIGURE 4.5 – Rendu HDR interactif avec différents paramètres de tone-mapping

Chaque ligne montre l'une des photographies d'origine associée à trois expositions différentes issues de notre système de rendu. *Statue* (première ligne), *Seashell* (deuxième ligne); *Horse* (troisième ligne).

Lors de l'étape de pré-calcul, l'utilisateur sélectionne arbitrairement des points de vue à partir d'une ou de plusieurs séries, ensuite pour chaque point de vue et pour

chaque exposition, une image LDR est calculée par notre système de visualisation (voir section 4.1). Les images résultantes sont ensuite transférées vers le CPU puis fusionnées en HDR (stockées sous le format .exr) en utilisant l'application HDR tools [Cotter, 2011].

Chaque image HDR subit une reproduction de tons via l'opérateur de Reinhard [Reinhard *et al.*, 2002]. L'image LDR résultante est accompagnée d'un ratio obtenu en divisant les valeurs de luminances de l'image HDR originale par les valeurs de luminance de l'image compressée (ayant subi une opération de réduction de tons). L'image LDR est ensuite stockée dans fichier PNG où le ratio est encodé puis stocké dans la composante alpha [Banterle, 2013]. L'image PNG obtenue est compressée en utilisant l'algorithme de compression DXT5 qui est largement utilisée par les cartes graphiques ensuite stockée sous le format (.DDS) et chargée sur le GPU. Le décodage et la restitution de l'image HDR est effectuée au niveau du fragment shader en utilisant l'équation.

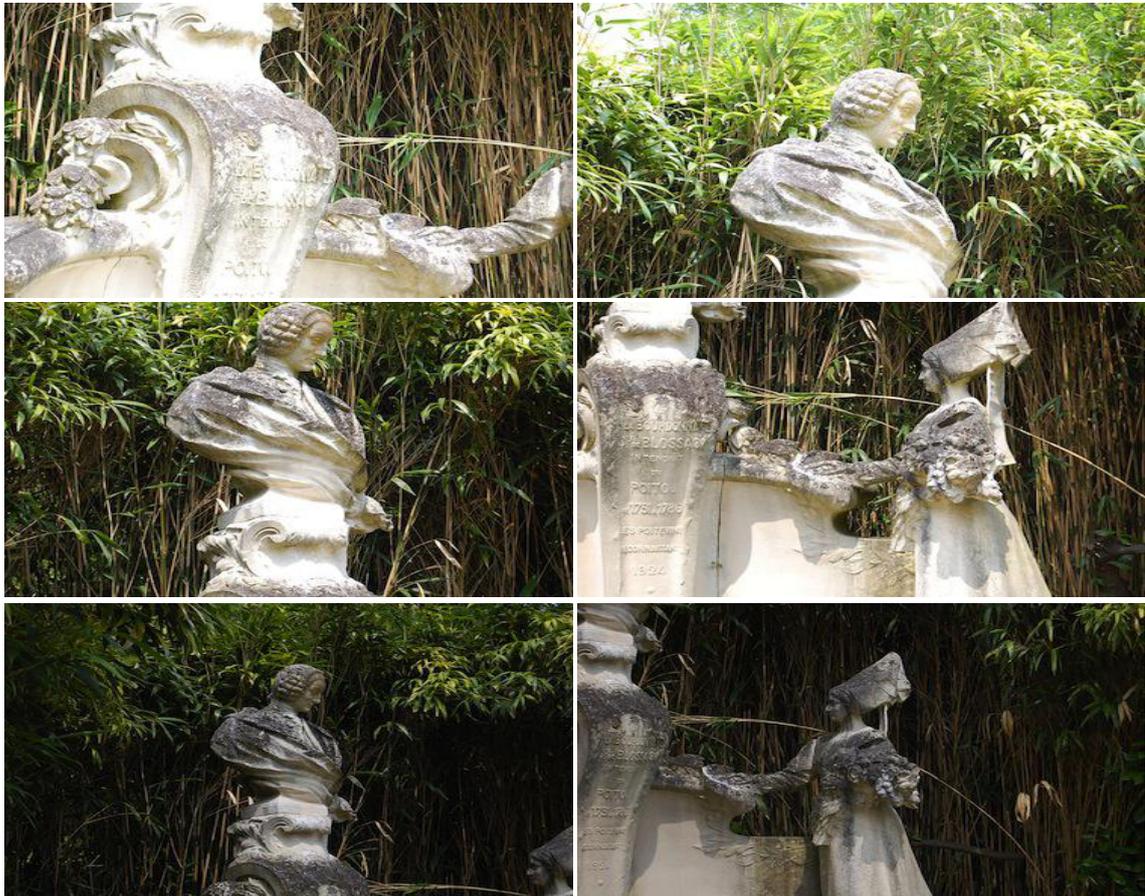
Le tableau 4.2 présente le temps de calcul des différentes parties de la méthode de rendu HDR ainsi que le nombre d'images par secondes pour une résolution égale à 512\*512 lors de la visualisation. La colonne *initialisation* correspond à l'étape de calcul des cartes de profondeurs pour la série d'images HDR. La colonne *visualisation* correspond au temps nécessaire pour générer une nouvelle image HDR. Quant à la colonne *pré calcul* correspond au temps de calcul nécessaire pour générer une unique série d'images HDR à partir de plusieurs séries de photos LDR, sans prendre en compte le temps de la fusion et de la réduction de tons qui se sont effectuées sur le CPU.

Scene	Initialisation (ms)	Visualisation (ms)	FPS	Pré-calcul (ms)
Statue	335	22	45	1500
Seashell	28	6	166	105
Horse	17	5	200	70

Tableau 4.2 – Temps de calcul des différentes parties de rendu HDR.

---

La figure 4.6 montre trois échantillons de séries de photos avec différentes expositions utilisées en entrée de notre système de rendu pour produire pour chaque exposition une image LDR pour un point de vue donné (voir figure 4.7)



**FIGURE 4.6 – Trois échantillons de série de photos avec différentes expositions**

chaque rangée présente deux photographies acquise avec un niveau d'exposition différent

## 4.6 Conclusion

Nous avons présenté un nouveau système de production d'images à haute gamme de dynamique (HDR) dédié aux photographies prises à main levée. Le système s'appuie sur une méthode de rendu basé image ce qui le rend plus robuste aux occultations par rapport aux autres méthodes de l'état de l'art. Il permet également



FIGURE 4.7 – Trois nouvelles expositions synthétisées pour un même point de vue en utilisant notre système de visualisation

une navigation virtuelle libre basé sur la série d'images HDR construites, avec la possibilité de changer la luminosité de l'environnement lors de la navigation.

# Chapitre 5

## Conclusion

Nous avons présenté dans cette thèse une nouvelle méthode de visualisation permettant de naviguer librement de manière interactive à l'intérieur d'environnements acquis à partir de photographies prises à main levée. L'étape de pré-traitement consiste à étalonner l'ensemble des caméras correspondant aux photographies et reconstruire un nuage de points de l'objet visualisé. Le nuage de points est utilisé à la manière d'un proxy géométrique. L'ensemble de notre système est exécuté sur le processeur graphique en tenant compte des occultations, de l'orientation des points de vue et la précision de la résolution à l'aide d'un système de pondération des photographies. Ce travail combine deux contributions : une en rendu basé points et l'autre en rendu basé images. La première contribution consiste à rajouter deux passes de rendu par splatting à l'algorithme de rendu basé points en vue d'une part de réduire les clignotements sur la silhouette de l'objet et d'autre part d'éviter les interpolations de profondeur indésirables lorsqu'on se rapproche de la surface de l'objet. Les corrections apportées ont montré leur robustesse aux artefacts de profondeur par rapport à l'algorithme de rendu basé point original.

La deuxième contribution présentée montre que notre système peut être employé pour reconstruire des images HDR lors de la visualisation à partir de plusieurs séries de prises de vues non alignées (ayant chacune une exposition fixe). Les images HDR produites donnent des informations plus précises des luminances de l'environnement ce qui permet d'augmenter le réalisme des scènes rendues. Le système est par ailleurs plus robuste aux occultations que les méthodes similaires de l'état de l'art.

---

Enfin, notre approche ne nécessite aucune intervention de l'utilisateur durant le processus de reconstruction. Elle offre les avantages des méthodes de visualisation libre à base d'images, tout en évitant les problèmes habituels inhérents à la reconstruction de maillages pour les proxy géométriques. De plus, la combinaison de plusieurs expositions pour le rendu basé image dans le but d'apporter plus de réalisme et d'immersion aux scènes visualisées est une idée très intéressante et originale en soi. Dans le futur, Nous envisagerons d'utiliser cette méthode pour combiner des objets réels et virtuels de manière interactive. Pour cela, une connaissance de l'environnement lumineux en association avec les images HDR produites par notre système est nécessaire pour estimer les informations de réflectance et modifier les conditions d'éclairage des objets acquis.

# Annexe A

## Transformation des matrices de la caméra de l'espace réel vers le système de coordonnées Opengl

Dans la section 2.4, nous indiquons que nous avons utilisé la bibliothèque OpenMVG [Moulon *et al.*, 2016a] pour étalonner les caméras et estimer les matrices intrinsèque et extrinsèque (c'est à dire la matrice de projection globale) pour chaque prise de vue. Dans cette partie, nous décrivons les étapes de conversion de ces matrices vers l'espace Opengl (voire webgl). Les nouvelles matrices résultantes (dites ModelView et Perspective) sont utilisées lors du rendu pour projeter le nuage de points sur les photographies d'origines.

La matrice de projection globale, que l'on note  $P$  est une matrice de  $3 \times 4$ , qui se compose de plusieurs paramètres :

$$P_{3 \times 4} = K[R|t] = \begin{bmatrix} f & c_u \\ f & c_v \\ & 1 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

---

où la distance focale  $f$  et les points principales  $c_u$  et  $c_v$  représentent les paramètres intrinsèques.

Les paramètres extrinsèques sont la rotation de la caméra et le vecteur de translation, notés respectivement par  $R$  et  $t$ .

Ces paramètres sont stockés par l'openMVG dans un fichier sous format binaire ou Json. Lorsqu'on utilise la PMVS, la matrice de rotation, la matrice intrinsèque et le vecteur de translation sont généralement pré-multipliés et le résultat est stocké dans un fichier texte sous la forme d'une matrice  $P_{3 \times 4}$  de valeurs flottantes. Dans ce qui suit nous montrons les étapes nécessaires pour décomposer cette matrice et d'en déduire la matrice modelView et la matrice de projection perspective, en tenant compte de la convention retenue pour l'orientation de repère en Opengl (l'axe Z positif est orienté vers l'arrière, l'axe X vers la droite et l'axe Y vers le haut).

Soit  $P_{3 \times 4}$  la matrice de projection globale, composée de 12 valeurs flottantes :

$$P_{3 \times 4} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

La première étape est d'exprimer la matrice dans l'espace des coordonnées homogènes, en rajoutant le vecteur (0,0,0,1). Nous multiplions ensuite les éléments par (-1) pour inverser la direction des axes. La matrice (P) devient :

$$P_{4 \times 4} = \begin{bmatrix} -p_{11} & -p_{12} & -p_{13} & -p_{14} \\ -p_{21} & -p_{22} & -p_{23} & -p_{24} \\ -p_{31} & -p_{32} & -p_{33} & -p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pour extraire la matrice ModelView ( $MV$ ) de la matrice de projection globale ( $P$ ), il faut multiplier celle-ci par l'inverse de la matrice intrinsèque :  $MV = inv(M) * P$ .

Soit  $l$  et  $h$ , respectivement la largeur et la hauteur de l'image. La matrice intrinsèque  $M$  est donnée par :

---


$$M = \begin{bmatrix} -f & 0 & c_1 & 0 \\ 0 & f & c_2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

tels que :  $c_1 = (0.5 * l) - 0.5$  et  $c_2 = (0.5 * h) - 0.5$  correspondent aux coordonnées du centre de l'image.

La deuxième étape consiste à calculer la matrice de projection perspective  $P_s$ . Pour se faire, on introduit le *near* et le *far*, la profondeur en valeur absolue selon l'axe Z, des plans near et far. Il faut tenir compte que le point de coordonnées (0,0) en OpenGL se situe dans le coin inférieur gauche de l'écran.

$$P_s = \begin{bmatrix} (2 * f)/l & 0 & 1 - (2 * c_1)/l & 0 \\ 0 & (2 * f)/h & -1 + (2 * c_2)/h & 0 \\ 0 & 0 & near & far \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



# Bibliographie

- [Adelson *et al.*, 1991] ADELSON, E. H., BERGEN, J. R. *et al.* (1991). The plenoptic function and the elements of early vision. [2](#)
- [Aliaga *et al.*, 2003] ALIAGA, D., A. FUNKHOUSER, T., YANOVSKY, D. *et* CARLBOM, I. (2003). Sea of images : A dense sampling approach for rendering large indoor environments. *IEEE Computer Graphics and Applications*, 23:22–30. [2.1.2](#)
- [Amenta *et* Bern, 1999] AMENTA, N. *et* BERN, M. (1999). Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504. [2.3](#)
- [Banterle, 2013] BANTERLE, F. (2013). Spidergl. [4.5](#)
- [Berger *et al.*, 2014] BERGER, M., TAGLIASACCHI, A., SEVERSKY, L. M., ALLIEZ, P., LEVINE, J. A., SHARF, A. *et* SILVA, C. T. (2014). State of the Art in Surface Reconstruction from Point Clouds. In LEFEBVRE, S. *et* SPAGNUOLO, M., éditeurs : *Eurographics 2014 - State of the Art Reports*. [1](#)
- [Botsch *et al.*, 2005] BOTSCH, M., HORNUNG, A., ZWICKER, M. *et* KOBBELT, L. (2005). High-quality surface splatting on today’s gpus. In *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005.*, pages 17–141. IEEE. ([document](#)), [2.6](#), [2.3.1](#), [2.3.2](#)
- [Botsch *et* Kobbelt, 2003] BOTSCH, M. *et* KOBBELT, L. (2003). High-quality point-based rendering on modern gpus. In *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings.*, pages 335–343. IEEE. [2.3.1](#)
- [Botsch *et al.*, 2004] BOTSCH, M., SPERNAT, M. *et* KOBBELT, L. (2004). Phong splatting. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, pages 25–32. Eurographics Association. [2.3.1](#)
- [Buehler *et al.*, 2001] BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S. *et* COHEN, M. (2001). Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432. ACM. [2.4.2](#), [3.2.3](#), [3.3](#)
- [Chaurasia, 2014] CHAURASIA, G. (2014). Algorithms and perceptual analysis for interactive free viewpoint image based-navigation. ([document](#)), [2.9](#)
- [Chaurasia *et al.*, 2013] CHAURASIA, G., DUCHENE, S., SORKINE-HORNUNG, O. *et* DRETTAKIS, G. (2013). Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):30. ([document](#)), [1](#), [1.1](#), [2.1.2](#), [2.3](#), [3](#)
- [Chaurasia *et al.*, 2011] CHAURASIA, G., SORKINE, O. *et* DRETTAKIS, G. (2011). Silhouette-aware warping for image-based rendering. In *Computer Graphics Forum*, volume 30, pages 1223–1232. Wiley Online Library. [1.1](#), [2.1.2](#), [2.1.3](#), [3](#)

- 
- [Chen et Williams, 1993] CHEN, S. E. et WILLIAMS, L. (1993). View interpolation for image synthesis. *In Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288. ACM. [2.1.1](#)
- [Cotter, 2011] COTTER, A. (2011). Hdr tools. [4.5](#)
- [Debevec et al., 1998] DEBEVEC, P., YU, Y. et BORSHUKOV, G. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. *In Rendering Techniques' 98*, pages 105–116. Springer. [2.1.2](#)
- [Debevec et Malik, 1997] DEBEVEC, P. E. et MALIK, J. (1997). Recovering high dynamic range radiance maps from photographs. *In Annual Conference on Computer Graphics*, pages 369–378. [1](#), [2.2](#), [4](#), [4.2](#)
- [Debevec et al., 1996] DEBEVEC, P. E., TAYLOR, C. J. et MALIK, J. (1996). Modeling and rendering architecture from photographs : A hybrid geometry-and image-based approach. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM. [1](#), [2.1.2](#), [2.1.2](#)
- [Denker et al., 2011] DENKER, K., LEHNER, B. et UMLAUF, G. (2011). Real-time triangulation of point streams. *Engineering with Computers*, 27(1):67–80. [2.3](#)
- [Einarsson et al., 2006] EINARSSON, P., CHABERT, C.-F., JONES, A., MA, W.-C., LAMOND, B., HAWKINS, T., BOLAS, M. T., SYLWAN, S. et DEBEVEC, P. E. (2006). Relighting human locomotion with flowed reflectance fields. *Rendering techniques*, 2006:17th. [1](#)
- [Eisemann et al., 2008] EISEMANN, M., DE DECKER, B., MAGNOR, M., BEKAERT, P., DE AGUIAR, E., AHMED, N., THEOBALT, C. et SELLENT, A. (2008). Floating textures. *In Computer graphics forum*, volume 27, pages 409–418. Wiley Online Library. [1.1](#), [2.1.2](#), [3](#)
- [Fuhrmann et Goesele, 2011] FUHRMANN, S. et GOESELE, M. (2011). Fusion of depth maps with multiple scales. *In ACM Transactions on Graphics (TOG)*, volume 30, page 148. ACM. [2.4.2](#)
- [Fuhrmann et Goesele, 2014] FUHRMANN, S. et GOESELE, M. (2014). Floating scale surface reconstruction. *ACM Transactions on Graphics (TOG)*, 33(4):46. [1.1](#), [2](#), [2.3](#), [2.4.2](#)
- [Fuhrmann et al., 2014] FUHRMANN, S., LANGGUTH, F. et GOESELE, M. (2014). Mve-a multi-view reconstruction environment. *In GCH*, pages 11–18. [2.4.1](#)
- [Furukawa et Ponce, 2007] FURUKAWA, Y. et PONCE, J. (2007). Accurate, dense, and robust multi-view stereopsis. volume 32. [2.4.1](#)
- [Furukawa et Ponce, 2010] FURUKAWA, Y. et PONCE, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376. [1.1](#), [2](#), [2.1.2](#), [2.4.1](#), [3.1](#), [4.1](#)
- [Goesele et al., 2010] GOESELE, M., ACKERMANN, J., FUHRMANN, S., HAUBOLD, C., KLOWSKY, R., STEEDLY, D. et SZELISKI, R. (2010). Ambient point clouds for view interpolation. *In ACM Transactions on Graphics (TOG)*, volume 29, page 95. ACM. [2.1.2](#)
- [Goesele et al., 2007] GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H. et SEITZ, S. M. (2007). Multi-view stereo for community photo collections. *In*

- 
- Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE. [2](#), [2.1.2](#), [2.4.1](#)
- [Gortler *et al.*, 1996] GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R. et COHEN, M. F. (1996). The lumigraph. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM. [2.1.2](#), [2.1.2](#), [2.1.3](#), [2.3.2](#), [3.2.1](#)
- [Gross et H. Pfister (eds.), 2007] GROSS, M. et H. PFISTER (EDS.), M. K. (2007). *Point-Based Graphics*. Elsevier. [2.7](#)
- [Grossman et Dally, 1998] GROSSMAN, J. P. et DALLY, W. J. (1998). Point sample rendering. *In Rendering Techniques' 98*, pages 181–192. Springer. [2.3.2](#)
- [Guennebaud et Paulin, 2003] GUENNEBAUD, G. et PAULIN, M. (2003). Efficient screen space approach for hardware accelerated surfel rendering. *In VMV*, volume 20003, pages 1–10. [2.3.1](#)
- [Hartley et Zisserman, 2003] HARTLEY, R. et ZISSERMAN, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press. [2.4](#)
- [Hawkins *et al.*, 2005] HAWKINS, T., EINARSSON, P. et DEBEVEC, P. (2005). A dual light stage. *In Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, pages 91–98. Eurographics Association. [1](#)
- [Hedman *et al.*, 2018] HEDMAN, P., PHILIP, J., PRICE, T., FRAHM, J.-M., DRETTAKIS, G. et BROSTOW, G. (2018). Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings)*, 37(6). [1](#), [2.1.2](#), [2.1.3](#)
- [Hedman *et al.*, 2016] HEDMAN, P., RITSCHER, T., DRETTAKIS, G. et BROSTOW, G. (2016). Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 35(6):231. ([document](#)), [1.1](#), [2.2](#), [2.1.2](#), [2.1.3](#)
- [Heigl *et al.*, 1999] HEIGL, B., KOCH, R., POLLEFEYS, M., DENZLER, J. et VAN GOOL, L. (1999). Plenoptic modeling and rendering from image sequences taken by a hand-held camera. *In Mustererkennung 1999*, pages 94–101. Springer. [2.1.2](#)
- [Kazhdan *et al.*, 2006] KAZHDAN, M., BOLITHO, M. et HOPPE, H. (2006). Poisson surface reconstruction. *In Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70. Eurographics Association. [1.1](#), [2](#), [2.3](#), [2.9](#), [2.4.2](#), [3.3](#)
- [Kazhdan et Hoppe, 2013] KAZHDAN, M. et HOPPE, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29. [2.4.2](#)
- [Kobbelt et Botsch, 2004] KOBBELT, L. et BOTSCH, M. (2004). A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814. [2.3.1](#)
- [Kolluri *et al.*, 2004] KOLLURI, R., SHEWCHUK, J. R. et O'BRIEN, J. F. (2004). Spectral surface reconstruction from noisy point clouds. *In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 11–21. ACM. [2.3](#)

- 
- [Levoy et Hanrahan, 1996] LEVOY, M. et HANRAHAN, P. (1996). Light field rendering. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM. [2.1.1](#)
- [Lhuillier et Quan, 1999] LHUILLIER, M. et QUAN, L. (1999). Image interpolation by joint view triangulation. *In Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 139–145. IEEE. [2.1.1](#)
- [Lowe, 2004] LOWE, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110. [2.4](#), [3.1](#)
- [Mann et Picard, 1994] MANN, S. et PICARD, R. W. (1994). Being ‘undigital’ with digital cameras : extending dynamic range by combining differently exposed pictures. [1](#), [2.2](#)
- [Marroquim *et al.*, 2007] MARROQUIM, R., KRAUS, M. et CAVALCANTI, P. R. (2007). Efficient point-based rendering using image reconstruction. *In Proceedings Symposium on Point-Based Graphics*, pages 101–108. ([document](#)), [1.2](#), [2.3.2](#), [2.3.3](#), [3.1](#), [3.4](#), [3.2.1](#), [3.5](#)
- [Marroquim *et al.*, 2008] MARROQUIM, R., OLIVEIRA, A. et CAVALCANTI, P. R. (2008). High quality image reconstruction of point models. *In 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 297–304. IEEE. [2.3.2](#), [3.2.1](#)
- [McMillan et Bishop, 1995] MCMILLAN, L. et BISHOP, G. (1995). Plenoptic modeling : An image-based rendering system. *In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 39–46. ACM. [2.1](#)
- [Moulon *et al.*, 2012] MOULON, P., MONASSE, P. et MARLET, R. (2012). Adaptive structure from motion with a contrario model estimation. *In Asian Conference on Computer Vision*, pages 257–270. Springer. [3.1](#)
- [Moulon *et al.*, 2016a] MOULON, P., MONASSE, P., PERROT, R. et MARLET, R. (2016a). Openmvg : Open multiple view geometry. *In International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer. [2.4](#), [3.1](#), [A](#)
- [Moulon *et al.*, 2016b] MOULON, P., MONASSE, P., PERROT, R. et MARLET, R. (2016b). Openmvg : Open multiple view geometry. *In International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer. [4.1](#)
- [Nieto *et al.*, 2016] NIETO, G., DEVERNAY, F. et CROWLEY, J. (2016). Variational image-based rendering with gradient constraints. *In 3D Imaging (IC3D), 2016 International Conference on*, pages 1–8. IEEE. [1.1](#), [2.1.2](#), [2.1.3](#)
- [Ortiz-Cayon *et al.*, 2015] ORTIZ-CAYON, R., DJELOUAH, A. et DRETTAKIS, G. (2015). A bayesian approach for selective image-based rendering using superpixels. *In International Conference on 3D Vision-3DV*. [2.1.2](#)
- [Perlin, 1985] PERLIN, K. (1985). An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296. [1](#)
- [Pfister *et al.*, 2000] PFISTER, H., ZWICKER, M., VAN BAAR, J. et GROSS, M. (2000). Surfels : Surface elements as rendering primitives. *In Proceedings of the*

- 
- 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co. [2.3.1](#)
- [Popescu *et al.*, 2006] POPESCU, V., MEL, C., DAUBLE, J. et SACKS, E. (2006). Reflected-scene impostors for realistic reflections at interactive rates. *In Computer Graphics Forum*, volume 25, pages 313–322. Wiley Online Library. [2.1.2](#)
- [Preiner *et al.*, 2012] PREINER, R., JESCHKE, S. et WIMMER, M. (2012). Autosplats : Dynamic point cloud visualization on the gpu. *In EGPGV*. [2.3.2](#), [2.3.3](#)
- [Pujades *et al.*, 2014] PUJADES, S., DEVERNAY, F. et GOLDLUECKE, B. (2014). Bayesian view synthesis and image-based rendering principles. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3906–3913. [1.1](#), [2.1.2](#), [2.1.3](#)
- [Reinhard *et al.*, 2002] REINHARD, E., STARK, M., SHIRLEY, P. et FERWERDA, J. (2002). Photographic tone reproduction for digital images. *ACM transactions on graphics (TOG)*, 21(3):267–276. [4.3](#), [4.5](#)
- [Ren *et al.*, 2002] REN, L., PFISTER, H. et ZWICKER, M. (2002). Object space ewa surface splatting : A hardware accelerated approach to high quality point rendering. *In Computer Graphics Forum*, volume 21, pages 461–470. Wiley Online Library. [2.3.1](#)
- [Rosenthal et Linsen, 2008] ROSENTHAL, P. et LINSEN, L. (2008). Image-space point cloud rendering. *In Proceedings of Computer Graphics International*, pages 136–143. [\(document\)](#), [2.3.2](#), [2.8](#), [2.3.3](#)
- [Rusu et Cousins, 2011] RUSU, R. B. et COUSINS, S. (2011). 3D is here : Point Cloud Library (PCL). *In IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China. [3.1](#)
- [Seitz et Dyer, 1996] SEITZ, S. M. et DYER, C. R. (1996). View morphing. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30. ACM. [\(document\)](#), [2.1](#), [2.1.1](#)
- [Sen *et al.*, 2012] SEN, P., KALANTARI, N. K., YAESOUBI, M., DARABI, S., GOLDMAN, D. B. et SHECHTMAN, E. (2012). Robust Patch-Based HDR Reconstruction of Dynamic Scenes. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH Asia 2012)*, 31(6):203 :1–203 :11. [1](#), [2.2](#)
- [Shum *et al.*, 2008] SHUM, H.-Y., CHAN, S.-C. et KANG, S. B. (2008). *Image-based rendering*. Springer Science & Business Media. [1](#), [2.1](#)
- [Sillion *et al.*, 1997] SILLION, F., DRETTAKIS, G. et BODELET, B. (1997). Efficient impostor manipulation for real-time visualization of urban scenery. *In Computer Graphics Forum*, volume 16. Wiley Online Library. [2.1.2](#)
- [Snavely *et al.*, 2008] SNAVELY, N., GARG, R., SEITZ, S. M. et SZELISKI, R. (2008). Finding paths through the world’s photos. *ACM Transactions on Graphics (SIGGRAPH 2008)*, 27(3):11–21. [2.1.2](#), [2.1.2](#), [2.4](#)
- [Snavely *et al.*, 2006] SNAVELY, N., SEITZ, S. M. et SZELISKI, R. (2006). Photo tourism : Exploring photo collections in 3d. *In SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA. ACM Press. [1.1](#), [2.1.2](#), [2.1.2](#), [2.4](#)

- 
- [Song *et al.*, 2018] SONG, L., LI, X., YANG, Y.-g., ZHU, X., GUO, Q. et LIU, H. (2018). Structured-light based 3d reconstruction system for cultural relic packaging. *Sensors*, 18(9):2981. [2.1.2](#)
- [Stich *et al.*, 2011] STICH, T., LINZ, C., WALLRAVEN, C., CUNNINGHAM, D. et MAGNOR, M. (2011). Perception-motivated interpolation of image sequences. *ACM Transactions on Applied Perception (TAP)*, 8(2):11. [2.1.1](#)
- [Tomaszewska et Mantiuk, 2007] TOMASZEWSKA, A. et MANTIUK, R. (2007). Image registration for multi-exposure high dynamic range image acquisition. In *WSCG (document)*, 1, [2.2](#), [2.4](#)
- [Vangorp *et al.*, 2011] VANGORP, P., CHAURASIA, G., LAFFONT, P.-Y., FLEMING, R. W. et DRETTAKIS, G. (2011). Perception of visual artifacts in image-based rendering of façades. In *Computer Graphics Forum*, volume 30, pages 1241–1250. Wiley Online Library. [1](#), [3.2.1](#)
- [Wang *et al.*, 2015] WANG, J., ZHANG, J. et XU, Q. (2015). Research on 3d laser scanning technology based on point cloud data acquisition. *ICALIP 2014 - 2014 International Conference on Audio, Language and Image Processing, Proceedings*, pages 631–634. [2.1.2](#)
- [Williams, 1978] WILLIAMS, L. (1978). Casting curved shadows on curved surfaces. In *ACM Siggraph Computer Graphics*, volume 12, pages 270–274. ACM. [3.2.2](#)
- [Williams, 1983] WILLIAMS, L. (1983). Pyramidal parametrics. In *Acm siggraph computer graphics*, volume 17, pages 1–11. ACM. [1](#)
- [Wu *et al.*, ] WU, C. *et al.* Visualsfm : A visual structure from motion system. [2.4](#)
- [Yang *et al.*, 2006] YANG, R., GUINNIP, D. et WANG, L. (2006). View-dependent textured splatting. *The Visual Computer*, 22(7):456–467. [2.1.2](#)
- [Zimmer *et al.*, 2011] ZIMMER, H., BRUHN, A. et WEICKERT, J. (2011). Freehand hdr imaging of moving scenes with simultaneous resolution enhancement. In *Computer Graphics Forum*, volume 30, pages 405–414. Wiley Online Library. [\(document\)](#), [1](#), [2.2](#), [2.5](#)

---