

Reconstruction par étalage gaussien 3D de phénomènes gazeux pour les effets visuels

par

Vincent ALIX-JOLY

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION
M. Sc. A.

MONTREAL, LE 9 JUIN 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Vincent Alix-Joly, 2026



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Eric Paquette, directeur de mémoire

Département de génie logiciel et des technologies de l'information, École de technologie supérieure

M. Adrien Gruson, président du jury

Département de génie logiciel et des technologies de l'information, École de technologie supérieure

M. Stéphane Coulombe, examinateur externe

Département de génie logiciel et des technologies de l'information, École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 11 MAI 2026

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je remercie d'abord mon directeur de recherche, Eric Paquette, de m'avoir guidé à travers ma maîtrise. Il n'y aurait tout simplement pas de mémoire devant vos yeux sans sa supervision libre et constructive, sans oublier son support pédagogique, financier et administratif.

Je remercie mes collègues de laboratoires, devenus amis, de m'avoir partagé leurs connaissances et d'avoir cultivé un environnement de travail collaboratif, cordial et chaleureux.

Enfin, je tiens à remercier ma conjointe, mes parents et mes sœurs pour leur soutien indéfectible, tout au long de mon parcours universitaire. J'ignore si j'aurais pu profiter d'autant d'opportunités s'ils n'avaient pas été là pour me pousser à me dépasser.

À la mémoire de Pinpin.

Reconstruction par étalage gaussien 3D de phénomènes gazeux pour les effets visuels

Vincent ALIX-JOLY

RÉSUMÉ

Le rendu de volumes transparents tels la fumée, le feu, la vapeur et les nuages requiert des temps de calcul prohibitifs. Au contraire, les méthodes basées sur l'étalage gaussien 3D permettent, par l'utilisation d'une représentation géométrique nouvelle, de capturer la radiance d'un sujet, puis de la rendre sous des vues nouvelles à des vitesses interactives et à des fidélités pratiquement inatteignables par les méthodes antérieures. Or, les efforts de recherche se sont marginalement intéressés au problème de la capture de volumes et de l'utilisation de celles-ci dans un contexte de création d'effets spéciaux. Nous pensons pouvoir tirer profit de l'étalage gaussien 3D pour transcoder les volumes en une représentation plus rapide pour le rendu. Le présent mémoire propose une méthode pour ledit transcodage de volumes transparents en gaussiennes 3D étalées. Notre méthode permet de circonscrire à la phase de prétraitement les opérations prohibitives propres à l'édition d'effets visuels volumiques, telles le transport lumineux, et s'articule autour d'améliorations à la robustesse de l'initialisation des gaussiennes 3D et de modifications à la boucle d'optimisation. L'amélioration de la robustesse de l'initialisation est implémentée d'une part grâce à l'extraction de la disposition des caméras depuis l'environnement de rendu des volumes, et d'autre part grâce à la conversion de l'information de densité des volumes préalablement simulés depuis leur format de stockage. Les modifications à l'optimisation consistent en des algorithmes de régularisation de position et d'anisotropie, en l'implémentation d'un noyau d'étalage quadratique comme alternative au noyau gaussien, et en un entraînement multifond. Nous testons aussi d'autres travaux de recherche en reconstruction, modélisation et rendu gaussiens pour mettre à l'épreuve leurs conclusions dans notre contexte. Nos résultats montrent que nos méthodes permettent de reconstruire les sujets volumiques à de meilleures qualités photométriques que les méthodes de reconstruction gaussienne existantes. Nos méthodes produisent des reconstructions plus géométriquement cohésives et mieux séparées de leur environnement d'entraînement. Les modèles ainsi reconstruits sont conséquemment utilisables dans des situations de composition avec d'autres modèles gaussiens et polygonaux et avec des fonds inconnus tels des cartes d'environnement présentant des conditions d'éclairage similaires à celles vues à l'entraînement.

Mots-clés: étalage gaussien, 3DGS, volume transparent, débiaisement multifond, effets visuels, VFX, transcodage, nuage, vapeur, feu, fumée

3D Gaussian Splatting-Based Reconstruction of Gaseous Phenomena for Visual Effects

Vincent ALIX-JOLY

ABSTRACT

Rendering volumetric phenomena such as smoke, fire, vapour, and clouds require prohibitive computation times. Meanwhile, 3D gaussian splatting-based (3DGS) methods are an emergent family of differentiable rendering algorithms which allow to capture and subsequently render a subject's radiance from novel viewpoints at unmatched interactive framerates and quality. However, only a few projects researched the possibility of leveraging 3DGS in the context of volumetric visual effects (VFX) productions. We think that 3DGS can be used to transcode (also known as baking) volumes to a 3DGS representation to accelerate post-authoring renders. The present master's thesis proposes such a method. Our method allows to circumscribe the computationally prohibitive light transport simulation to the volumetric VFX authoring step. We achieve this by proposing modifications to the 3DGS method in the initialization and optimization steps. We improve the initialization by leveraging properties of standard volumetric storage formats and of the camera parametrization used in VFX authoring software; this addresses the issues that arise when using photogrammetric registration, as proposed by the 3DGS reference implementation, in our volumetric context. To specifically improve the reconstruction of low opacity volumetric subjects, we propose anisotropy and position regularizations, an implementation of the quadratic splatting kernel as an alternative to the gaussian kernel, and a multi-background rendering and evaluation algorithm. We test the conclusions of other gaussian reconstruction, modelization, and rendering research endeavours when applied in our context. Our results show that our method allows the reconstruction of volumetric subjects with greater photometric accuracy than existing gaussian reconstruction methods. Our method produces gaussian models which are more geometrically cohesive and are better separated from their training environment. Consequently, the models are usable in a compositing workflow. Our gaussian models' transparency correctly encode their respective ground-truth's transparency, allowing them to be composited with other gaussian models, with polygonal models, and with environment maps featuring similar lighting conditions as the ground truth, as demonstrated by our results.

Keywords: 3D gaussian splatting, 3DGS, transparent volume, multi-background debias, visual effects, VFX, bake, baking, transcode, cloud, vapor, fire, smoke

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 ÉTAT DE L'ART	5
1.1 Étalage gaussien	5
1.1.1 Rendu	6
1.1.2 Optimisation	8
1.1.3 Densification	9
1.1.4 Noyaux statistiques alternatifs	12
1.2 Grilles volumiques	14
1.3 Gaussiennes volumiques	16
1.3.1 Rendu de volumes participatifs modélisés par primitives gaussiennes	17
1.3.2 Approximation de volumes par gaussiennes hiérarchiques éparses	18
1.4 Orientations du mémoire	20
CHAPITRE 2 CONTRIBUTIONS	23
2.1 Assemblage des scènes de test	23
2.2 Découplage photogrammétrique	25
2.3 Algorithmes d'initialisation volumique	28
2.3.1 Échantillonnage aléatoire et élagage basé sur la visibilité	28
2.3.2 Échantillonnage du gradient de la densité	32
2.3.3 Conversion hiérarchique de grille de densité	33
2.4 Paramétrisation de l'implémentation de référence	36
2.5 Régularisations	38
2.5.1 Régularisation de position	38
2.5.2 Régularisation d'anisotropie	40
2.6 Étalage quadratique	41
2.7 Optimisation multifond	42
CHAPITRE 3 RÉSULTATS	45
3.1 Entraînement multifond	45
3.2 Algorithme d'initialisation	48
3.3 Algorithme de densification	53
3.4 Configuration des harmoniques sphériques	57
3.5 Régularisations	60
3.6 Noyaux d'étagage	63
3.7 Niveau de détail maximal reconstituable	65
3.8 Composition de scène	71
CONCLUSION ET RECOMMANDATIONS	77

ANNEXE I	CONVERSION DU RÉFÉRENTIEL <i>BLENDER</i> AU RÉFÉRENTIEL <i>COLMAP</i> PAR <i>PHOTOGRAMMETRY IMPORTER</i>	81
ANNEXE II	ÉCHANTILLONNEURS DE GRADIENT DE CHAMP DE DENSITÉ VDB	83
ANNEXE III	CONVERSION HIÉRARCHIQUE DE GRILLE DE DENSITÉ VDB	85
ANNEXE IV	COMPARAISON NUMÉRIQUE DES NOYAUX D'ÉTALAGE	87
LISTE DE RÉFÉRENCES		89

LISTE DES TABLEAUX

	Page
Tableau 3.1	Qualité photométrique sur fond inconnu selon les fonds d'entraînement 48
Tableau 3.2	Qualité photométrique moyenne selon la méthode d'initialisation 52
Tableau 3.3	Qualité photométrique moyenne selon l'algorithme de densification ... 55
Tableau 3.4	Qualité photométrique moyenne selon le degré d'harmoniques sphériques 59
Tableau 3.5	Coûts de stockage moyens selon le degré d'harmoniques sphériques ... 59
Tableau 3.6	Qualité photométrique moyenne selon la régularisation et sa force 62
Tableau 3.7	Meilleur noyau selon la configuration de la scène reconstruite 65
Tableau 3.8	Qualité photométrique moyenne selon la disposition de caméras 67
Tableau 3.10	Occurences d'erreur par canal en reconstruction monomurale 70
Tableau 3.11	Occurences d'erreur par canal en reconstruction tétramurale 70
Tableau 3.12	Qualité photométrique moyenne en situation de composition 72
Tableau IV-1	Qualité photométrique moyenne selon le noyau d'étagage 87

LISTE DES FIGURES

	Page
Figure 1.1	Contribution des noyaux d'étalement 1D selon la distance à la moyenne ... 13
Figure 1.2	Schématisation d'un disque de densité unitaire encodé en VDB 16
Figure 1.3	Schéma des méthodes proposées assemblées en pipeline 22
Figure 2.1	Scènes et disposition de caméras tétramurale 24
Figure 2.2	Scène <i>nuage de poussière</i> en disposition de caméras monomurale 25
Figure 2.3	Référence et reconstruction partielle de <i>foyer</i> avec environnement de support 27
Figure 2.4	Points résultants de l'initialisation aléatoire 30
Figure 2.5	Points résultants de l'élagage par test de visibilité minimale 32
Figure 2.6	Points résultats de l'échantillonnage du gradient de la densité 33
Figure 2.7	Surajustement par lentilles gaussiennes 39
Figure 2.8	Opacité des noyaux 3D selon la distance à leur centre 42
Figure 3.1	Reconstruction sur fond connu selon les fonds d'entraînement 47
Figure 3.2	Reconstruction selon l'algorithme d'initialisation 51
Figure 3.3	Reconstructions de <i>nuage atmosphérique</i> selon l'algorithme de densification 55
Figure 3.4	Reconstructions de <i>théière</i> selon l'algorithme de densification 56
Figure 3.5	Reconstructions de <i>nuage de poussière</i> selon le degré d'harmoniques sphériques 58
Figure 3.6	Reconstructions de <i>nuage de poussière</i> selon la régularisation 61
Figure 3.7	Reconstructions de <i>nuage de poussière</i> selon la régularisation 62
Figure 3.8	Reconstructions de <i>théière</i> selon le noyau d'étalement 64
Figure 3.9	Reconstructions de <i>nuage de poussière</i> selon la disposition de caméra .. 67

Figure 3.10	Différence de luminance des rendus et de la référence	68
Figure 3.11	Répartition de la différence des rendus et de la référence	70
Figure 3.12	Reconstructions composées de <i>théière</i> selon la combinaison des méthodes d'entraînement	73
Figure 3.13	Erreur carrée des reconstructions composées de <i>théière</i> selon la combinaison des méthodes	74
Figure 3.14	Rendus quadruples composés avec polygones et carte d'environnement .	75

LISTE DES ALGORITHMES

	Page
Algorithme 2.1	Initialisation de nuage de points à positions aléatoires 30
Algorithme 2.2	Élagage de points basé sur la visibilité minimale 31
Algorithme 2.3	Échantillonnage du gradient de la grille de densité 34
Algorithme 2.4	Conversion hiérarchique de grille en gaussiennes 35
Algorithme II-1	Échantillonneur de gradient de densité 84
Algorithme III-1	Conversion hiérarchique à basse résolution de grille en boîtes 85
Algorithme III-2	Conversion hiérarchique à résolution moyenne de grille en boîtes .. 86
Algorithme III-3	Conversion hiérarchique à haute résolution de grille en boîtes 86

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

3DGS	<i>3D Gaussian Splatting for Real-Time Radiance Field Rendering</i> (Étalage gaussien 3D) (Kerbl, Kopanas, Leimkuehler & Drettakis, 2023)
3DGS-MCMC	<i>3D Gaussian Splatting as Markov Chain Monte Carlo</i> (Densification par gradient stochastique pour 3DGS) (Kheradmand <i>et al.</i> , 2024)
AbsGrad	<i>AbsGS : Recovering Fine Details for 3D Gaussian Splatting</i> (Densification par gradient homodirectionnel pour 3DGS) (Ye, Li, Liu, Qiao & Dou, 2024)
DSYG	<i>Don't Splat your Gaussians : Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media</i> (Rendu de volume participatif modélisé par primitives gaussiennes) (Condor <i>et al.</i> , 2025)
LPIPS	<i>Learned Perceptual Image Patch Similarity</i> (Similarité perceptuelle apprise de fragments d'image) (Zhang, Isola, Efros, Shechtman & Wang, 2018)
MSE	<i>Mean Square Error</i> (Moyenne du carré des erreurs)
NeRF	<i>Neural Radiance Fields for View Synthesis</i> (Champ de radiance neuronal) (Mildenhall <i>et al.</i> , 2021)
OpenVDB	Implémentation de référence des travaux présentés dans l'article <i>VDB : High-Resolution Sparse Volumes with Dynamic Topology</i> (Museth, 2013)
Pixel	<i>Picture element</i> (Unité de base d'une grille bidimensionnelle décrivant une image)
PSNR	<i>Peak Signal to Noise Ratio</i> (Ratio maximal du signal et du bruit)
SSIM	<i>Structural Similarity Index Measure</i> (Index de similarité structurelle) (Wang, Bovik, Sheikh & Simoncelli, 2004)
Voxel	<i>Volume element</i> (Unité de base d'une grille tridimensionnelle décrivant un volume)

LISTE DES SYMBOLES ET UNITÉS DE MESURE

x	Quantité scalaire
$\mathbf{v}$	Quantité vectorielle
$f(\mathbf{x})$	Fonction scalaire opérant sur un vecteur
$\mathbf{f}(\mathbf{v})$	Fonction vectorielle
M	Quantité matricielle
$\hat{\mathbf{q}}$	Quaternion hamiltonien unitaire
$\mathbb{I}$	Matrice identité
$F(x)$	Fonction matricielle
$\nabla f(\mathbf{x})$	Gradient d'une fonction
$\ \mathbf{x}\ $	Norme euclidienne d'une quantité vectorielle
$ \mathbf{x} $	Valeur absolue d'une quantité
$\mathbf{x} \odot \mathbf{y}$	Produit hamiltonien (composant-à-composant)
$\bar{x}$	Moyenne d'une variable
$\#\mathbf{v}$	Nombre d'éléments d'un vecteur
$\#M$	Nombre de colonnes d'une matrice
$\mathbf{v}_i$	i -ième élément d'un vecteur
$M_{ij}, M_{i;j}$	Élément situé à la j -ième colonne de la i -ième rangée d'une matrice
$M_{[i_1;i_2][j_1;j_2]}$	Sous-bloc de la matrice délimité inclusivement par les i_1 -ième et i_2 -ième rangées, et par les j_1 -ième et j_2 -ième colonnes
$[x]_i$	Quantité vectorielle de dimension i dont tous les éléments ont la valeur x
$[x]_{ij}$	Quantité matricielle de dimension i par j dont tous les éléments ont la valeur x

INTRODUCTION

Le rendu de volumes est une tâche ayant suscité des efforts de recherche constants dans les dernières années. Les équations permettant de simuler le comportement des volumes et le comportement de la lumière à travers ceux-ci sont connues, ce qui rend le rendu volumique omniprésent dans les productions cinématographiques. Or, ces mêmes équations sont prohibitives en temps de calcul, ce qui les rend extrêmement difficiles d'utilisation dans les applications interactives. Parmi les phénomènes difficiles à rendre, certains, dont le feu, la fumée et la vapeur, sont approximés grossièrement par des émetteurs de particules. D'autres, tels les nuages, sont simulés sommairement par des grilles volumiques rendues à basse résolution par lancer de rayon. Les deux cas souffrent de lacunes importantes : l'illusion créée par les émetteurs de particules se brise dès qu'on remarque les textures répétées ; le lancer de rayons dans des grilles volumiques peut facilement occuper la moitié du temps consacré au rendu de chaque image à chaque pas de temps si la résolution de la grille et le nombre de rayons lancés ne sont pas méticuleusement calibrés.

Des recherches antérieures ont déjà permis d'améliorer la faisabilité du rendu volumique. La méthode de référence contemporaine pour le stockage, la compression, la simulation et l'interrogation des propriétés des fluides, dont la fumée et les nuages, est la librairie *OpenVDB* (Museth, 2013) (les volumes encodés au format *OpenVDB* seront aussi appelés « VDB » ci-après). Or, cette représentation volumique n'encode habituellement que les propriétés physico-optiques des volumes, telle la densité, ce qui maintient le besoin d'avoir recours à un algorithme de rendu. Cependant, les travaux de Kerbl *et al.* (2023) ont mené à la création d'une nouvelle méthode : le « *3D Gaussian Splatting* (étalage gaussien 3D, ci-après 3DGS) ». La méthode 3DGS permet de modéliser la radiance d'une scène à l'aide d'ellipsoïdes translucides. L'apparence et le nombre d'ellipsoïdes sont contrôlés par une boucle d'optimisation ayant pour seul intrant des photographies de la scène à imiter. La radiance ainsi reconstruite peut alors être réévaluée à partir de positions et d'orientations jamais observées dans les caméras et images

ayant formé le corpus d’entraînement. De plus, des travaux antérieurs ont déjà mis à profit les primitives gaussiennes en tant que primitive volumique pour le lancer de rayon (Condor *et al.*, 2025) et comme format intermédiaire au rendu de volumes VDB (Sharma & Schmalstieg, 2025).

Nos travaux ont pour objectif le développement d’une méthode de « *baking* (transcodage en amont) » des volumes translucides en gaussiennes. Les travaux antérieurs confirment que les gaussiennes sont capables d’encoder les propriétés physico-optiques des volumes, mais nous pensons que cette capacité peut être améliorée de sorte à circonscrire le lancer de rayon à l’étape de création des effets spéciaux. Ainsi, les effets volumiques pourraient être rendus à l’écran sous un meilleur compromis de qualité visuelle, ce qui sera mesuré objectivement par des indices de qualité photométrique, tout en maintenant stable la part de temps alloué à ces effets parmi le budget de rendu d’une trame de l’application. Lorsqu’utilisées de concert, nos méthodes devraient permettre le rendu de volumes transparents à des vitesses interactives sur du matériel accessible au grand public.

Pour arriver à cette fin, le présent mémoire comportera trois chapitres. Le premier présentera une revue de la littérature contemporaine du domaine de l’étalage gaussien, du rendu volumique et de l’intersection de ces deux derniers ; le chapitre creusera notamment les articles évoqués dans la présente introduction. Le second chapitre présentera le travail effectué et nos contributions à l’objectif poursuivi ; nos contributions comportent des scènes de test représentatives, des méthodes d’export des données photogrammétriques entre l’éditeur 3D et le moteur d’optimisation 3DGS, des algorithmes d’initialisation de la reconstruction volumique par l’échantillonnage des données photogrammétriques et volumiques exportées par l’éditeur 3D, des régularisations de position et d’élongation pour réduire la prépondérance de certains artefacts visuels et une méthode d’optimisation multifond augmentant la convergence des modèles et la reconstruction des détails à faible contraste. Le troisième chapitre comparera les performances qualitatives et quantitatives des méthodes proposées à celles formant l’état de l’art en situation de reconstruction

d'effet visuel volumique. Nous mesurerons la performance de l'optimisation multifond, des algorithmes d'initialisation, de l'algorithme de densification, de la paramétrisation de l'encodage des couleurs, des régularisations et du noyau d'étalement, avant d'étudier le niveau de détail maximal reconstituable par nos méthodes et la cohésivité des modèles reconstruits en situation de composition avec des modèles et des environnements jamais vus pendant l'optimisation. Le mémoire sera conclu par une discussion sur les résultats obtenus et une recommandation quant auxquelles des méthodes proposées sont les plus influentes sur le succès du transcoding d'un effet volumique en gaussiennes 3D rasterisées.

CHAPITRE 1

ÉTAT DE L'ART

Le domaine des effets spéciaux s'inspire d'avancées majeures en simulation de fluide et en rendu photoréaliste. Alors que les studios orchestrent des productions toujours plus visuellement ambitieuses, ceux-ci sont aussi à la recherche d'économies de coûts permises par des techniques plus rapides, donc plus économes. Les volumes translucides sont omniprésents dans les productions cinématographiques. Par contre, ils sont coûteux en temps de calcul, et sont donc bien souvent grossièrement approximatés dans les applications nécessitant des rendus à vitesses interactives, comme les jeux vidéos. Conséquemment, le présent chapitre présentera l'état de l'art dans le domaine du rendu de volumes translucides, ainsi que certains développements récents prometteurs pour accélérer le rendu de tels phénomènes.

1.1 Étalage gaussien

L'article *3D Gaussian Splatting for Real-Time Radiance Field Rendering* (3DGS) (Kerbl *et al.*, 2023) est devenu incontournable dans le domaine du rendu et de la reconstruction 3D dès sa publication. Leur méthode permet, à partir d'un système de caméra calibré et d'images, de reconstruire l'environnement vu par ces images à l'aide de gaussiennes 3D. Les gaussiennes 3D sont analogues à des ellipsoïdes de plus en plus translucides à mesure que l'on s'éloigne de leur centre respectif. Les gaussiennes sont ajoutées, retirées, placées, déformées et recolorées grâce à une boucle de descente du gradient dirigée par un signal de qualité photométrique. Contrairement à ses méthodes contemporaines de la famille des champs de radiance neuronaux (NeRF (Mildenhall *et al.*, 2021)), 3DGS n'est pas basée sur les réseaux de neurones et n'utilise pas de « *ray marching* (marche de rayon) » volumique, ce qui lui permet une vitesse de rendu, d'entraînement, et une capture du détail nettement supérieures.

La méthode descend des méthodes de rendu et de reconstruction utilisant le point comme primitive géométrique. Or, cette famille de méthodes s'était auparavant butée à des lacunes attribuables à la non-différenciabilité du point. Pour contourner ce problème, 3DGS propose

d'adopter la gaussienne 3D comme primitive de rendu optimisable. Les gaussiennes 3D y sont définies comme une sorte de point déformable, chaque gaussienne possédant une position, une couleur, une opacité, une taille et une rotation. Ces deux dernières propriétés sont alors combinées pour former la covariance de la gaussienne et donc sa forme dans l'espace. L'utilisation de la gaussienne à des fins de rendu est inspirée des travaux antérieurs de Zwicker, Pfister, van Baar & Gross (2001), où la gaussienne est utilisée comme un filtre d'échantillonnage pour étaler les points disposés sur une surface (par exemple, une capture produite par lidar) et leur donner l'illusion de la continuité.

1.1.1 Rendu

Formellement, Kerbl *et al.* (2023) définissent géométriquement une gaussienne 3D $\mathbf{g}$ par sa position 3D $\boldsymbol{\mu}$, son orientation encodée par un quaternion hamiltonien unitaire $\hat{\mathbf{q}}$, sa taille dans les trois axes $\mathbf{s}$, sa couleur $\mathbf{c}$ et son opacité en son centre o . Les équations

$$\mathbf{g} = \left\{ \boldsymbol{\mu} \quad \hat{\mathbf{q}} \quad \mathbf{s} \quad \mathbf{c} \quad o \right\} \Big| \hat{\mathbf{q}} = \begin{bmatrix} \hat{\mathbf{q}}_w & \hat{\mathbf{q}}_i & \hat{\mathbf{q}}_j & \hat{\mathbf{q}}_k \end{bmatrix}, \|\hat{\mathbf{q}}\| = 1 \quad (1.1a)$$

$$G = \left\{ \mathbf{g}_1 \quad \mathbf{g}_2 \quad \dots \quad \mathbf{g}_i \right\} \quad (1.1b)$$

$$R_i = \begin{bmatrix} 1 - 2(\hat{\mathbf{q}}_j^2 + \hat{\mathbf{q}}_k^2) & 2(\hat{\mathbf{q}}_i \hat{\mathbf{q}}_j - \hat{\mathbf{q}}_k \hat{\mathbf{q}}_w) & 2(\hat{\mathbf{q}}_i \hat{\mathbf{q}}_k + \hat{\mathbf{q}}_j \hat{\mathbf{q}}_w) \\ 2(\hat{\mathbf{q}}_i \hat{\mathbf{q}}_j + \hat{\mathbf{q}}_k \hat{\mathbf{q}}_w) & 1 - 2(\hat{\mathbf{q}}_i^2 + \hat{\mathbf{q}}_k^2) & 2(\hat{\mathbf{q}}_j \hat{\mathbf{q}}_k - \hat{\mathbf{q}}_i \hat{\mathbf{q}}_w) \\ 2(\hat{\mathbf{q}}_i \hat{\mathbf{q}}_k - \hat{\mathbf{q}}_j \hat{\mathbf{q}}_w) & 2(\hat{\mathbf{q}}_j \hat{\mathbf{q}}_k + \hat{\mathbf{q}}_i \hat{\mathbf{q}}_w) & 1 - 2(\hat{\mathbf{q}}_i^2 + \hat{\mathbf{q}}_j^2) \end{bmatrix} \quad (1.1c)$$

$$S_i = \text{diag}(\mathbf{s}_i) \quad (1.1d)$$

$$\Sigma_i = R_i S_i S_i^\top R_i^\top \quad (1.1e)$$

présentent comment les gaussiennes sont agencées en un modèle G et présentent la formation de la matrice de covariance 3D Σ_i de chacune des gaussiennes $\mathbf{g}_i$ à partir de la forme matricielle de leurs paramètres d'orientation R_i et de taille S_i .

Comme l'étagage gaussien est une technique de rasterisation, chaque primitive doit être projetée à l'écran dans le but de calculer sa contribution aux pixels de l'image. La projection de la

gaussienne par l'étalage à l'image de sa matrice de covariance 3D fut initialement proposée par Zwicker *et al.* (2001). Ladite projection est présentée à l'équation (1.2) (adaptée de Chen *et al.* (2024)). Les équations

$$\Sigma'_i = (JW\Sigma W^\top J^\top)_{[1;2][1;2]} \quad (1.2a)$$

$$\mu'_i = PW \mu_i \quad (1.2b)$$

$$\mathcal{G}_i(\mathbf{x}') = \exp\left(\frac{-1}{2}(\mathbf{x}' - \mu'_i)^\top \Sigma'^{-1}_i (\mathbf{x}' - \mu'_i)\right) \quad (1.2c)$$

présentent comment calculer la contribution $\mathcal{G}_i$ de la i -ième gaussienne en le pixel de position 2D $\mathbf{x}'$ grâce aux projections du centre μ_i et de la covariance Σ'_i à l'espace-image. La matrice W encode la projection monde-caméra, P encode la projection caméra-image, et J est formée par le jacobien de P .

Finalement, la couleur de chaque pixel est calculée en espace-image avec un processus de mélange alpha (*alpha-blending*) selon l'équation

$$\alpha_k(\mathbf{x}') = o_k \mathcal{G}_k(\mathbf{x}') \quad (1.3a)$$

$$C(\mathbf{x}') = \sum_{i=1}^{\#G} \mathbf{c}_i \alpha_i(\mathbf{x}') \prod_{j=1}^{i-1} (1 - \alpha_j(\mathbf{x}')) . \quad (1.3b)$$

Le processus présenté à l'équation (1.3a) (adaptée de Chen *et al.* (2024)), consiste à itérer sur les gaussiennes triées selon la distance séparant leur centre μ_i du centre optique de la caméra les observant, à évaluer leur contribution $\mathcal{G}$ en le pixel $\mathbf{x}'$ pondérée par leur opacité centrale o_i , de sorte à former leur opacité en le pixel α . On boucle alors sur les opacités α pour calculer leur produit avec la couleur des gaussiennes correspondantes et avec la transparence $1 - \alpha$ des j -gaussiennes plus proches de la caméra, avant de sommer ces produits pour obtenir la couleur finale C en le pixel de position $\mathbf{x}'$.

1.1.2 Optimisation

Tel que mentionné plus tôt, cette chaîne d'opérations est entièrement différentiable. La méthode est capable de retracer la contribution des gaussiennes 3D vis-à-vis de chaque pixel, ce qui permet d'optimiser leurs paramètres en 3D (position, rotation, taille, couleur et opacité), de sorte à minimiser une fonction de perte photométrique. La minimisation de cette perte photométrique entre un rendu du nuage de gaussiennes et une image de référence prise par la même caméra dans la même orientation permet d'ajuster progressivement les paramètres des gaussiennes jusqu'à ce qu'elles imitent presque parfaitement la référence depuis tous les points de vue du jeu de données de référence. Une reconstruction qui a bien généralisé peut même imiter fidèlement la référence dans des vues nouvelles, jamais vues à l'entraînement. La fonction de perte photométrique proposée par Kerbl *et al.* (2023) suit l'équation

$$L_k = (1 - \lambda_s)L_{1;k}(C_k, R_k) + \lambda_s L_{S;k}(C_k, R_k), \quad (1.4)$$

où k désigne la caméra ayant produit le rendu C_k et la référence R_k , L_k désigne la perte totale, $L_{1;k}$ désigne la différence absolue pixel à pixel moyenne observée de la vue k (aussi appelée perte L1 dans la littérature), $L_{S;k}$ désigne la perte structurelle observée de la vue k et λ_s est un hyperparamètre scalaire décrivant la pondération des pertes. Les travaux de Ye *et al.* (2025) recommandent de donner à λ_s la valeur de 0,2.

Le coefficient $L_{1;k}$ de la fonction de perte L_k est décrit par l'équation

$$L_{1;k} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |C_k(\mathbf{x}') - R_k(\mathbf{x}')| \Big|_{\mathbf{x}' = \begin{bmatrix} i \\ j \end{bmatrix}}. \quad (1.5)$$

Celle-ci décrit le calcul de la moyenne absolue des différences pixel-à-pixel entre le rendu produit C_k et la référence R_k en la même vue de référence d'index k .

Le coefficient $L_{S;k}$ de la fonction de perte L_k est décrit par l'équation

$$L_{S;k} = \text{SSIM}(C_k, R_k) . \quad (1.6)$$

Le « *structural similarity index measure* (index de similarité structurelle) », aussi connu sous l'acronyme SSIM, a originellement été proposé par Wang *et al.* (2004). L'algorithme opère autour de chaque pixel une convolution de la différence des pixels avoisinants par rapport à la référence jusqu'à l'obtention d'un index scalaire. La convolution apporte de la compréhension structurelle, autrement invisible à une simple différence pixel à pixel.

1.1.3 Densification

Le secret d'une reconstruction compacte et sans compromis de qualité est d'allouer intelligemment les ressources disponibles. À cette fin, 3DGS propose une technique de contrôle adaptatif, tirant profit de sa différenciabilité de bout-en-bout. Dans sa configuration par défaut, 3DGS opère une densification à chaque $p_{\text{densif}} = 100$ itérations. Pour déterminer si une gaussienne doit être densifiée, la méthode somme à chaque itération la dérivée partielle de la fonction de perte photogrammétrique L_k vis-à-vis de la moyenne de la gaussienne μ_i , avant de moyenniser cette perte en divisant la somme par le nombre d'itérations par période de densification pour former le gradient de la perte $\nabla L(\mu_i)$. Cette opération est décrite par l'équation

$$\nabla L(\mu_i) = \frac{1}{p_{\text{densif}}} \sum_{k=1}^{p_{\text{densif}}} \frac{\partial L_k}{\partial \mu_i} . \quad (1.7)$$

Or, les travaux de Ye *et al.* (2024) démontrent que cette formulation sous-évalue le gradient du signal par rapport à la position des gaussiennes à cause de la collision des gradients. En effet, comme toutes les dérivées partielles $\frac{\partial L_k}{\partial \mu_i}$ sont d'orientations différentes, plusieurs de ces dérivées s'annulent lors du calcul de la somme. Pour contrecarrer cet effet, Ye *et al.* (2024)

proposent « *AbsGrad* (le gradient homodirectionnel) », d'équation

$$\nabla \hat{L}(\mu_i) = \frac{1}{p_{\text{densif}}} \sum_{k=1}^{p_{\text{densif}}} \left| \frac{\partial L_k}{\partial \mu_i} \right|. \quad (1.8)$$

Additionner les valeurs absolues des dérivées partielles permet d'aligner leurs orientations avant d'en calculer la somme ; les dérivées ne s'annulent plus.

Une fois le gradient de la perte $\nabla L(\mu_i)$ ou $\nabla \hat{L}(\mu_i)$ calculé pour chaque gaussienne, l'algorithme de densification marque les gaussiennes comme sous-reconstruites et à densifier lorsqu'elles excèdent le seuil de densification τ_{densif} . Les gaussiennes les plus petites, identifiées comme telles si la norme de leur taille $\|s_i\|$ est inférieure au seuil τ_{div} , sont doublées, tandis que les gaussiennes les plus grandes, identifiées comme telles si la norme de leur taille $\|s_i\|$ excède le seuil τ_{div} , sont divisées en deux. Les gaussiennes densifiées se voient distancées légèrement pour éviter qu'elles ne se chevauchent et qu'elles ne subissent la même optimisation. Cette série d'opération est décrite par le système d'équation

$$A_i \leftarrow \begin{cases} \{G_i\} & \text{si } \|\nabla L(\mu_i)\| < \tau_{\text{densif}} \\ \{G_i \quad G_i\} & \text{si } \|\nabla L(\mu_i)\| \geq \tau_{\text{densif}} \text{ et } s_i < \tau_{\text{div}} \\ \{G_i \quad G_i \mid \frac{s_i}{2}\} & \text{si } \|\nabla L(\mu_i)\| \geq \tau_{\text{densif}} \text{ et } s_i \geq \tau_{\text{div}} \end{cases} \quad (1.9a)$$

$$G \leftarrow \{\mathbf{g} \mid \forall i \in \{1, 2, \dots, \#G\}, \mathbf{g} \in A_i\}, \quad (1.9b)$$

où A_i décrit le sous-ensemble produit par la densification d'une des gaussiennes sur lequel on itère pour mettre à jour l'ensemble de gaussiennes G qui forme la modélisation de la scène.

En remplaçant $\nabla L(\mu_i)$ par $\nabla \hat{L}(\mu_i)$ dans l'équation (1.9), la contribution des gaussiennes est évaluée plus justement vis-à-vis des pixels. Conséquemment, la contribution estimée par gradient homodirectionnel est plus élevée que celle évaluée par 3DGS, ce qui cause une hausse du nombre de gaussiennes pour un même seuil τ_{densif} . Dans leurs expériences, Ye *et al.* (2024) doublent le seuil pour arriver à un nombre de gaussiennes semblable à 3DGS et le quadruplent pour en obtenir approximativement 40 % moins, tandis que la librairie *gsplat* (Ye *et al.*, 2025) utilise

le triple pour en obtenir 20 % moins. Malgré cette diminution du nombre de gaussiennes, les reconstructions usant du gradient homodirectionnel performant mieux que celles faites avec le gradient originel grâce à une allocation plus intelligente des primitives ; la densification 3DGS surestime la densité nécessaire aux frontières des objets tout en délaissant les arrière-plans, ce qui est corrigé par la densification *AbsGrad*.

L’expression de la densification en termes de moyenne du gradient et d’heuristiques, comme le proposent les travaux de Kerbl *et al.* (2023) et de Ye *et al.* (2024), n’est pas la seule approche existante : l’article *3D Gaussian Splatting as Markov Chain Monte Carlo* (Kheradmand *et al.*, 2024) (surnommé 3DGS-MCMC dans la littérature et ci-après) propose de formuler la densification comme une descente de gradient stochastique. La méthode fonctionne en téléportant les gaussiennes contribuant peu à l’image en leur ajoutant un terme de bruit proportionnel à leur opacité respective. Les gaussiennes contribuant peu sont ainsi utilisées pour explorer l’espace, et permettent aux gaussiennes opaques de rester en place pour favoriser la qualité. L’article rapporte des gains de qualité substantiels par rapport à 3DGS (Kerbl *et al.*, 2023) sur les jeux de données de référence *Mip-NeRF360* (Barron, Mildenhall, Verbin, Srinivasan & Hedman, 2022), *Tanks & Temples* (Knapitsch, Park, Zhou & Koltun, 2017) et *Deep Blending* (Hedman *et al.*, 2018) en initialisation aléatoire et photogrammétrie. Ces résultats indiquent que la méthode est plus robuste face aux initialisations sous-contraintes.

Par contre, toutes ces formulations de la densification sont optimisées pour la reconstruction de scènes naturelles. Ainsi, la performance lors de la reconstruction d’un seul objet sans arrière-plan n’est pas étudiée dans ces méthodes. Les articles se gardent aussi de quantifier la performance de leurs méthodes lors de reconstruction de volumes translucides. Le présent mémoire comparera ces méthodes afin de sélectionner la plus appropriée à notre cas d’utilisation. Par exemple, la variable de décision de 3DGS-MCMC, l’opacité, est possiblement systématiquement plus faible lors de reconstruction de volumes translucides que lors de la reconstruction de surfaces opaques. Conséquemment, il est fortement probable que l’algorithme ne puisse jamais stabiliser la position des gaussiennes. De même, 3DGS et *AbsGrad* supposent que les gaussiennes très transparentes

ne contribuent pas à l'image ; elles sont donc détruites périodiquement, et potentiellement trop agressivement lors de la reconstruction de sujets volumiques translucides.

1.1.4 Noyaux statistiques alternatifs

Par ailleurs, la gaussienne n'est pas la seule distribution statistique à pouvoir être étalée. Les travaux de Chen *et al.* (2024) ont démontré l'efficacité des noyaux linéaires sur plusieurs jeux de données de référence. Pour permettre la polymorphie de l'étalage vis-à-vis du noyau utilisé, la méthode propose de refactoriser le noyau gaussien $\mathcal{G}(\mathbf{x}'_i)$, présenté précédemment à l'équation (1.2), en termes de distance de Mahalanobis D_m selon l'équation

$$D_m = \sqrt{(\mathbf{x}' - \boldsymbol{\mu}'_i)^\top \boldsymbol{\Sigma}'^{-1} (\mathbf{x}' - \boldsymbol{\mu}'_i)} \quad (1.10a)$$

$$\mathcal{G}(\mathbf{x}') = \exp\left(\frac{-D_m^2}{2}\right). \quad (1.10b)$$

Cette formulation rend possible la comparaison des noyaux gaussien (équation (1.11a)), linéaire (équation (1.11c)), laplacien (équation (1.11d)), quadratique (équation (1.11b)) et cosinus relevé (équation (1.11e)). Pour maintenir la structure de l'algorithme d'optimisation, chaque noyau est réaligné sur la gaussienne par un coefficient λ , de sorte que tous les noyaux occupent un volume semblable. L'équation (1.11) présente la contribution des noyaux en fonction de la distance à la moyenne et illustre comment certains, par leur frontière finie, pourraient être utilisés pour mieux encoder des modèles à frontières nettes.

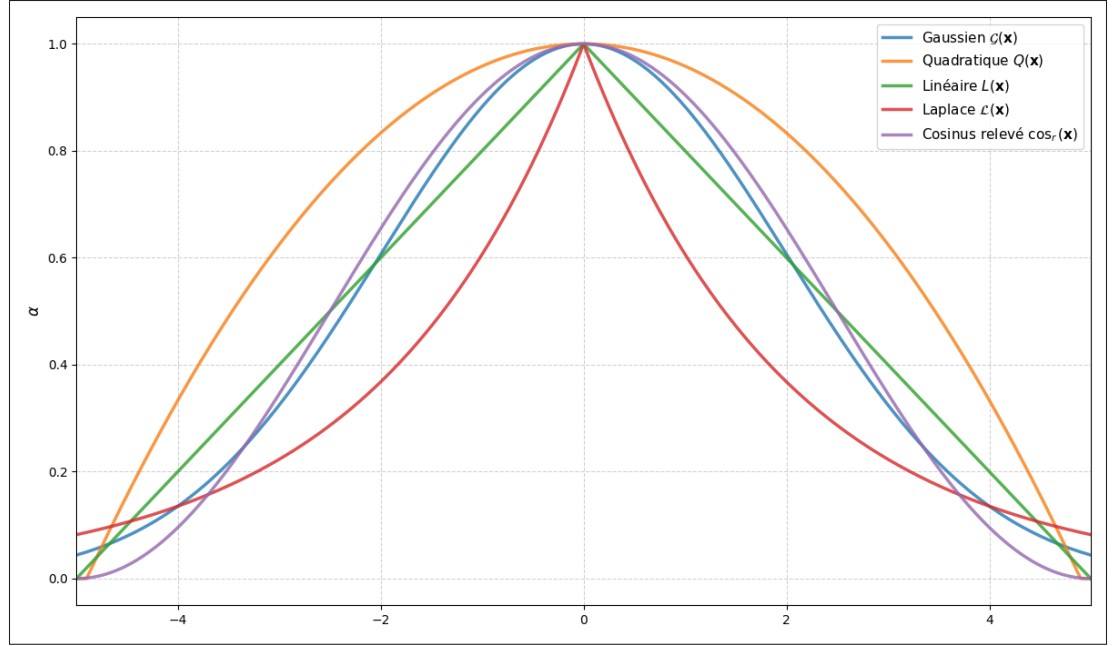


Figure 1.1 Contribution des noyaux d'étalement 1D selon la distance à la moyenne

$$\mathcal{G}(\mathbf{x}') = \exp\left(\frac{-D_m^2}{2}\right) \quad (1.11a)$$

$$Q(\mathbf{x}') = \max\left(0; 1 - \lambda D_m^2\right) \quad \lambda = \frac{1}{6} \quad (1.11b)$$

$$L(\mathbf{x}') = \max\left(0; 1 - \lambda D_m\right) \quad \lambda = \frac{2}{5} \quad (1.11c)$$

$$\mathcal{L}(\mathbf{x}') = \exp(-D_m) \quad (1.11d)$$

$$\cos_r(\mathbf{x}') = \begin{cases} \frac{1}{2}[1 + \cos(1 + \pi\lambda D_m)] & \text{si } \lambda D_m \leq 1 \\ 0 & \text{sinon} \end{cases} \quad \lambda = \frac{2}{5} \quad (1.11e)$$

De plus, les distributions présentées à l'équation (1.11) dont le gradient est linéaire voient leur gradient respectif être pondéré par la distance entre le pixel évalué et le centre du noyau, de sorte à lisser le gradient à et éviter les discontinuités. La figure 1.1 illustre la contribution unidimensionnelle des différents noyaux alignés en fonction de la distance à la moyenne, et illustre le besoin de lisser le gradient des noyaux à support fini.

Les expériences comparatives menées par Chen *et al.* (2024) ont démontré un gain de qualité lors de l'utilisation de la distribution linéaire sur les jeux de données de référence *Mip-NeRF360* (Barron *et al.*, 2022), *Tanks & Temples* (Knapitsch *et al.*, 2017) et *Deep Blending* (Hedman *et al.*, 2018). Par contre, quoique la distribution linéaire permette une amélioration qualitative dans les zones de géométrie à haute fréquence, à transitions nettes et dans les arrière-plans, elle n'offre pas de gain supplémentaire aux avant-plans. Néanmoins, l'utilisation de distributions à support fini a pour bénéfice d'accélérer l'évaluation du gradient sans compromettre la précision du rendu ou du gradient. Au contraire, les distributions à support infini, comme la gaussienne et la laplacienne, sont généralement tronquées après un certain nombre d'écarts-types pour des raisons de performance.

En somme, 3DGS, AbsGrad, 3DGS-MCMC et les différents noyaux forment un champ de possibilités intéressant pour former la base des méthodes qui seront proposées aux chapitres suivants. Par contre, aucune de ces méthodes de base n'a été formellement testée en situation de reconstruction volumique. La prochaine section s'attarde à ce problème.

1.2 Grilles volumiques

La librairie de référence pour la manipulation et le stockage des volumes est *OpenVDB* (Museth, 2013). Celle-ci encode un volume de taille théoriquement infinie dans un arbre éparé composé de nœuds décrivant des sous-volumes denses progressivement plus petits, jusqu'à ce que les volumes soient unitaires. Ces unités de volume sont appelées « voxels ». Le premier nœud est appelé la racine, et cette racine contient une valeur de fond par défaut et une liste de nœuds moyens. Pour que la racine puisse décrire un espace infini, elle détient une liste de longueur quelconque et pouvant croître de ses enfants directs, ces derniers composant le premier niveau de nœuds moyens. Chaque nœud moyen décrit son sous-espace avec une liste de longueur fixe dont les éléments sont soit des sous-nœuds, soit des valeurs si le sous-espace correspondant est parfaitement uniforme. Ces sous-nœuds sont soit d'autres nœuds moyens plus petits, soit des nœuds-feuilles. Les nœuds-feuilles contiennent toujours tous leurs voxels. Les voxels sont soit actifs, auquel cas ils contiennent une valeur, soit inactifs, auquel cas ils prennent la valeur de fond.

Les nœuds encodent leurs enfants directs avec des masques de bits pour encoder l'information de façon compacte, ce qui rend l'information rapidement interprétable lorsque vient le besoin d'itérer sur l'arbre de haut en bas et inversement. Les pans de l'espace sont stockés de façon contiguë, ce qui permet des accès prévisibles à la mémoire, donc rapides, par le processeur.

La configuration typique utilise un nœud racine, deux niveaux de nœuds moyens et un niveau de nœuds-feuilles contenant les voxels et leurs valeurs ; les deux niveaux moyens et le niveau feuille décrivent des espaces denses d'un volume de 32, 16 et 8 voxels de côté, respectivement. Pour la simulation des gaz, seul l'encodage de la densité est nécessaire, chaque voxel prend alors pour valeur un unique nombre à virgule flottante. La valeur de fond est nulle, donc tous les voxels sans densité sont compressés. La figure 1.2 présente un cas simplifié d'un disque de densité unitaire partout à l'intérieur d'un cercle au rayon de 96 unités, et de densité nulle partout ailleurs. Les carrés y représentent les trois niveaux de nœuds et le quatrième niveau formé de voxels. Le nœud racine stocke l'ensemble des nœuds illustrés sans être illustré lui-même. Les chiffres écrits en blanc représentent les nombres qui sont stockés dans le VDB. La figure montre que les zones de densité uniforme sont efficacement compressées par une seule valeur, tandis que les zones frontières sont précisément discrétisées par des voxels. Tout le volume à l'extérieur des nœuds schématisés prend la valeur de fond nulle et ne consomme aucun espace de stockage, puisqu'il ne forme pas de cellules.

Quoique la librairie expose des opérateurs utiles au rendu des volumes, une grille VDB n'encode pas l'apparence du volume qu'elle décrit. De plus, puisqu'elle tire ses origines dans l'industrie cinématographique, elle est principalement adaptée au rendu par lancer de rayons. Dans un tel moteur de rendu, chaque pixel doit lancer des rayons pour tester s'il peut toucher des sources de lumière à travers le volume par l'intégration de tests d'intersection ou de transmission stochastique. Comme ce parcours est coûteux en temps de calcul, nous proposerons une approche de transcodage de l'apparence des volumes en gaussiennes 3D. Cette approche sera hors ligne, de sorte à circonscrire le calcul de l'éclairage. Ainsi, son affichage en ligne et illuminé sera direct, au coût de ne pas pouvoir modifier l'éclairage sans réentraînement.

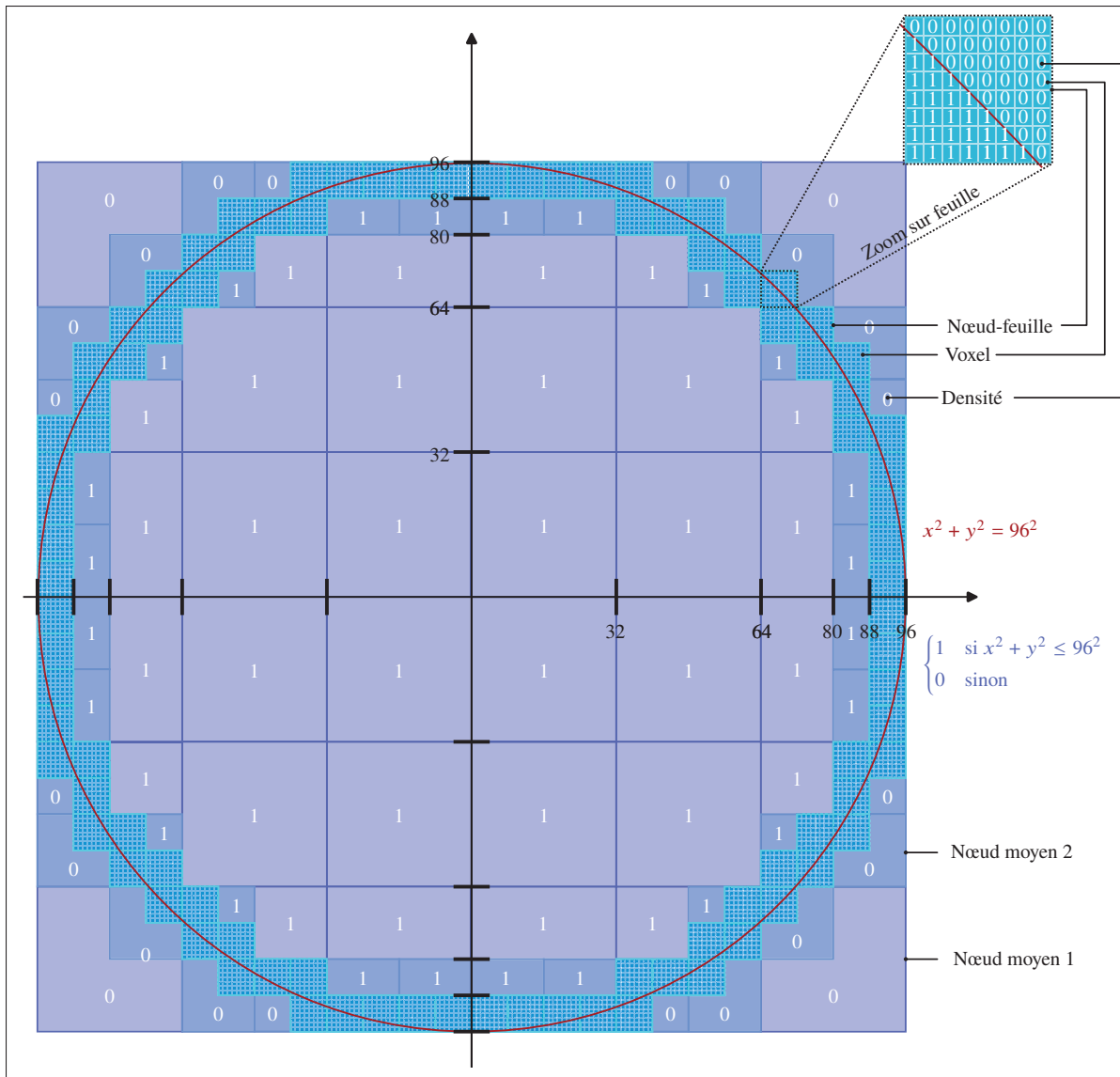


Figure 1.2 Schématisation d'un disque de densité unitaire encodé en VDB

1.3 Gaussiennes volumiques

Suite au succès instantané de 3DGS, plusieurs efforts de recherche ont mis à profit le rendu et la reconstruction par gaussiennes. Il s'avère que l'idée fondamentale de 3DGS, la modélisation à l'aide de primitives éparses et différenciables, est désirable et adaptable dans d'autres méthodes de rendu. En l'occurrence, nous présenterons ici deux méthodes complémentaires de modélisation

volumique gaussienne, soit le lancer de rayon à travers des primitives gaussiennes et la conversion de VDB en gaussiennes.

1.3.1 Rendu de volumes participatifs modélisés par primitives gaussiennes

Le lancer de rayon à travers des volumes gaussiens a été proposé par Condor *et al.* (2025) dans l'article *Don't Splat your Gaussians : Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media* (surnommé ci-après DSYG). Leurs travaux réutilisent la définition spatiale des gaussiennes proposée par Kerbl *et al.* (2023) dans 3DGS, mais remplacent le processus de rendu par rasterisation et par mélange alpha par une simulation de transport lumineux. Dans cette simulation, chaque gaussienne encode un milieu homogène du point de vue de ses propriétés physico-optiques. Tout comme 3DGS, la méthode proposée produit des rendus différenciables, ce qui permet de procéder à des reconstructions.

De plus, similairement à Chen *et al.* (2024), Condor *et al.* (2025) ont expérimenté avec l'utilisation d'un noyau statistique alternatif, soit le noyau quadratique d'Epanechnikov. Les deux articles ont soulevé les propriétés intéressantes des noyaux statistiques à support fini pour des raisons de vitesse de calcul. Or, les noyaux finis ne permettent que d'accélérer la rasterisation, déjà rapide (Chen *et al.*, 2024), tandis que le noyau d'Epanechnikov utilisé en lancer de rayon permet de réduire les chevauchements des noyaux le long du rayon, ce qui permet d'utiliser un chemin de code rapide dans l'intégrateur. Malgré ces gains d'efficacité, les auteurs soulignent que les effets de fumée et de nuage bénéficient des transitions douces permises par le noyau gaussien, et recommandent donc ce dernier noyau dans de tels cas.

Par ailleurs, le lancer de rayon permet d'éviter une lacune de la rasterisation 3DGS soulevée par Radl *et al.* (2024) et expliquée par le tri des gaussiennes avant le rendu. Puisque, dans 3DGS, les gaussiennes doivent être mélangées de l'avant vers l'arrière pour former l'image, celles-ci doivent être triées selon leur distance à la caméra les observant, et cette distance n'est calculée qu'avec la position de la gaussienne, sans égard pour la covariance de la gaussienne. Cette simplification cause des discontinuités dans l'ordre des gaussiennes, même sous un léger

changement de l'orientation de la caméra produisant l'image. Ce phénomène est particulièrement apparent dans un flux vidéo, alors que les gaussiennes passent l'une devant l'autre, causant ainsi des scintillements.

Pour démontrer la capacité du moteur de rendu différenciable DSYG dans les tâches de reconstructions permises par 3DGS, Condor *et al.* (2025) proposent un algorithme d'initialisation mieux adapté aux champs de densité. Cet algorithme, inspiré des travaux de Jakob, Regg & Jarosz (2011), se compose du calcul de l'espérance-maximisation du nuage de points à un certain budget de primitives, suivi d'un élagage des gaussiennes chevauchantes, basé sur la divergence de Kullback-Leibler, additionné d'une régularisation anisotrope. Cette approche produit une initialisation riche en détail, mais est prohibitive en temps de calcul lorsqu'utilisée sur des jeux de données de test, qui sont déjà de haute résolution. Par exemple, ce calcul prend de multiples heures sur le jeu de données de référence de nuage atmosphérique (Walt Disney Animation Studios, 2017), tel que rapporté par Condor *et al.* (2025).

1.3.2 Approximation de volumes par gaussiennes hiérarchiques éparses

La seconde méthode, l'approximation de volumes par des gaussiennes éparses, a été proposée par Sharma & Schmalstieg (2025). Les auteurs y proposent une famille d'algorithmes visant à approximer un volume *OpenVDB* (Museth, 2013) par un nuage de gaussiennes à des fins de visualisation scientifique, sous l'hypothèse qu'il est plus simple et rapide de transformer un volume en gaussiennes et de rendre ces gaussiennes que de rendre le volume lui-même. La méthode tire parti de la structure hiérarchique des grilles *OpenVDB* pour essayer de comprendre le détail local du volume tout en permettant de compresser les zones uniformes. L'article propose une famille d'algorithmes déclinés selon le niveau de détail requis.

Le premier niveau est le niveau faible. Dans cette configuration, l'algorithme itère sur tous les nœuds-feuilles de la grille *OpenVDB*. Pour chaque nœud-feuille actif, l'algorithme calcule la boîte englobante minimale englobant tous les voxels actifs dudit nœud-feuille. L'algorithme

crée alors, pour chaque boîte englobante calculée, une gaussienne dont les axes sont de même longueur que les axes de la boîte.

Le second niveau est le niveau moyen. Tout comme au niveau faible, l'algorithme itère sur tous les nœuds-feuilles ayant au moins un voxel actif. Pour chaque nœud-feuille dont tous les voxels sont actifs (le nœud-feuille est dit dense), huit gaussiennes positionnées et dimensionnées de sorte à couvrir les huit sous-blocs de quatre voxels de côtés composant le nœud-feuille sont créées. Pour chaque nœud-feuille non dense (aussi dit éparse), l'espace est séparé en 64 sous-blocs de deux voxels de côté. Pour chacun de ces 64 sous-blocs, soit une gaussienne ajustée est créée si le sous-bloc est dense, soit une gaussienne par voxel actif est créée dans le cas-contraire.

Le troisième et dernier niveau est le niveau élevé. Ce niveau est identique au niveau moyen, à l'exception du partitionnement des nœuds-feuilles denses : le niveau moyen les partitionne en sous-blocs de quatre voxels de côté, tandis que le niveau élevé les partitionne en sous-blocs de deux voxels de côté.

Les niveaux moyen et élevé allouent plus de gaussiennes dans les zones non uniformes du volume dans le but de préserver la densité d'information visuelle présente en ces endroits. Or, le stockage des gaussiennes tend à occuper beaucoup plus de mémoire que le stockage de la grille hiérarchique *OpenVDB*. D'une part, les gaussiennes ont simplement plus de propriétés à stocker : les grilles VDB destinées à être rendues par lancer de rayon n'ont besoin de stocker que leur densité, puisque la loi de Beer-Lambert permet de reconstruire la transmission lumineuse en intégrant la densité le long d'un rayon lancé à travers la grille, tandis que les gaussiennes doivent stocker leur position, leur taille, leur rotation, leur couleur et leur opacité centrale. D'autre part, les gaussiennes sont initialisées ici naïvement par des heuristiques, sans étape d'optimisation ; il existe donc une probabilité qu'il y ait trop de gaussiennes pour l'information visuelle présente, et donc que des gaussiennes soient stockées inutilement. Néanmoins, les gaussiennes volumiques ainsi produites peuvent s'intégrer dans un moteur de lancer de rayon, ce que les auteurs démontrent en créant un moteur de rendu basé sur *NVIDIA Optix*.

Finalement, leurs travaux suggèrent que, bien que la manipulation géométrique des gaussiennes soit tout à fait possible, les heuristiques simples proposées dans ces travaux ne semblent pas être capables de se substituer à un processus d’optimisation de l’apparence, du nombre et de la disposition des gaussiennes. En particulier, leur méthode est incapable de reproduire la couleur des volumes, puisque leurs rendus de fumée ne présentent que des couleurs uniformes ou de fausses couleurs. Nous pensons que de développer une étape d’optimisation de l’apparence basée sur 3DGS permettrait d’augmenter l’utilisabilité de la méthode de conversion de VDB proposée dans la présente section. Cette hypothèse est soutenue par les travaux de Celarek, Kopanas, Drettakis, Wimmer & Kerbl (2025), qui ont comparé les méthodes de rendu gaussien par étalage et par lancer de rayon dans le cadre d’une tâche de reconstruction de volume translucide, et ont conclu qu’avec suffisamment de gaussiennes, l’étalage est supérieur en qualité.

1.4 Orientations du mémoire

En bref, 3DGS a imposé la gaussienne comme primitive de rendu différenciable incontournable. Des travaux subséquents ont décroïsonné la méthode en montrant que la gaussienne était adaptée au rendu de volumes translucides. Par contre, cette démonstration s’est articulée grâce à un changement profond à l’algorithme de rendu, soit le passage de la rasterisation au lancer de rayon. Cette dernière technique est historiquement prohibitive en temps de calcul et en matériel graphique, bien que des avancements récents ont rendu la technique utilisable en temps réel sur du matériel de grade consommateur, quoiqu’à grand coût énergétique, en particulier sur les appareils mobiles.

Or, la revue de littérature réalisée dans le cadre de ce mémoire a révélé que la reconstruction de volumes translucides fut en bonne partie ignorée par la recherche en photogrammétrie. Du côté du rendu différenciable et de 3DGS, la littérature s’est articulée autour du lancer de rayon pour reconstruire par rendu différenciable les phénomènes lumineux volumiques. Notre revue n’a pas permis d’identifier une implémentation accessible publiquement d’un algorithme d’étalage gaussien spécialement adapté à la reconstruction de volumes, les modifications à l’étalage

gaussien proposées par Celarek *et al.* (2025) se limitant à certaines reparamétrisations visant la comparabilité avec les méthodes de lancer de rayon gaussien.

Par conséquent, le présent mémoire proposera une série de méthodes complémentaires et de modifications à 3DGS s’articulant autour de la tâche de reconstruction de la radiance de volumes participatifs par rastérisation gaussienne différenciable. Les méthodes proposées se concentreront sur la reconstruction de volumes synthétiques de sorte à pouvoir les inscrire dans un contexte de production d’effets spéciaux interactifs pour appareils grand public. Nous nous reposons sur les conclusions de Celarek *et al.* (2025) et visons à circonscrire la simulation et le rendu de fluides en amont de nos méthodes tout en minimisant la perte de qualité photométrique. La figure 1.3 résume graphiquement les sujets abordés, les contributions et les validations des conclusions de la littérature dans de nouveaux contextes qui seront présentés dans le ce mémoire. Tel que le précise la légende, les sujets auxquels nous avons contribué directement sont mis en évidence en bleu, tandis que les validations sont mises en évidence en rouge.

Les articles cités dans la présente section seront mis à profit de différentes manières. L’implémentation servant de base à nos travaux sera *gsplat* (Ye *et al.*, 2025), puisqu’il s’agit de l’implémentation de 3DGS la plus configurable et extensible disponible publiquement. Elle supporte plusieurs extensions à 3DGS potentiellement utiles à notre cas d’utilisation, dont *AbsGrad* (Ye *et al.*, 2024) et *3DGS-MCMC* (Kheradmand *et al.*, 2024). Les travaux de Condor *et al.* (2025) ainsi que ceux de Sharma & Schmalstieg (2025) inspireront nos méthodes d’initialisation, tout particulièrement pour la conversion directe de grilles volumiques *OpenVDB* (Museth, 2013) en gaussiennes. Nous réaliserons certains tests manquants en ce qui a trait à la performance de leurs méthodes en situation de rastérisation et, spécifiquement aux méthodes de Sharma & Schmalstieg (2025), d’optimisation. Nous verrons aussi à confirmer ou infirmer les conclusions de Chen *et al.* (2024), ainsi que celles de Condor *et al.* (2025), selon qui l’utilisation d’un noyau à support fini compromet trop les transitions douces requises pour la fumée malgré des gains théoriques et souhaitables de performance.

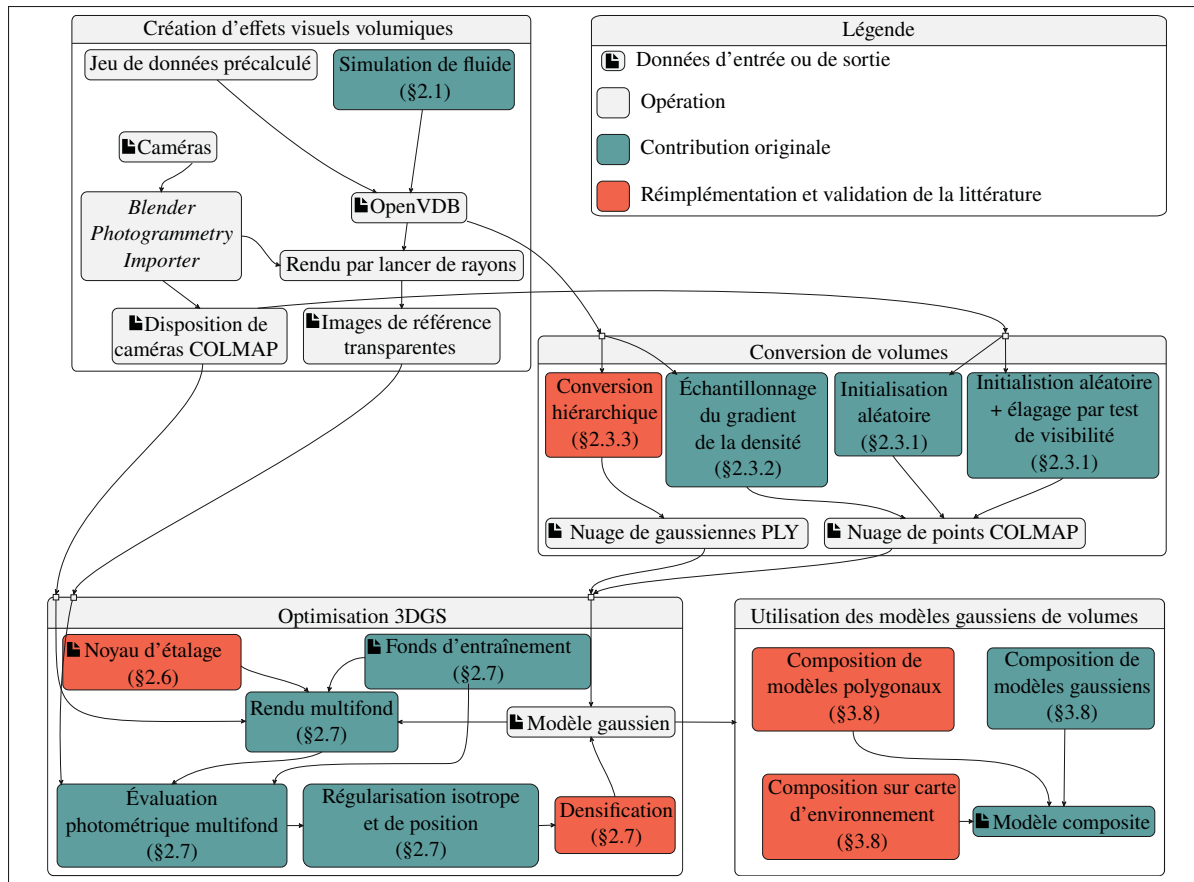


Figure 1.3 Schéma des méthodes proposées assemblées en pipeline

CHAPITRE 2

CONTRIBUTIONS

Le présent chapitre présentera chronologiquement le travail accompli dans le cadre du présent mémoire et les contributions visant à permettre le transcodage des volumes par 3DGS issues dudit travail. Ce chapitre couvrira l’assemblage des scènes de test, le découplage du processus photogrammétrique par l’export des scènes de test, nos algorithmes d’initialisation volumique, notre recherche de la paramétrisation optimale du moteur 3DGS, nos régularisations visant à amoindrir les artefacts visuels fréquents dans nos transcodages, notre réimplémentation de l’étalage quadratique et notre méthode d’entraînement multifond.

2.1 Assemblage des scènes de test

Les phénomènes visuels translucides sont innombrables. Parmi ces phénomènes, le feu, la vapeur d’eau et les nuages apparaissent dans plusieurs productions vidéoludiques et cinématographiques. Le feu et la vapeur d’eau sont représentés dans notre jeu de données sous la forme d’un foyer d’extérieur et d’une théière (assemblées et simulés dans le cadre du présent mémoire), tandis que les nuages y sont représentés par un nuage de poussière (JangaFX, 2020) et par un nuage atmosphérique (Walt Disney Animation Studios, 2017). Ces phénomènes forment la base des scènes constituant le jeu de données. Ces scènes présentent une diversité de niveaux de détail des simulations, de degrés d’opacité et de tailles relatives à la constellation de caméras les observant. Des scènes *foyer*, *théière* et *nuage de poussière*, nous choisissons la trame 47 comme la plus visuellement intéressante, puisque les scènes y atteignent un plateau de couverture spatiale, ce qui permet de capturer dans les rendus la diversité des phénomènes. Du jeu de données *nuage atmosphérique*, statique, les grilles à pleine et à quart de résolution sont respectivement utilisées pour produire des rendus et pour conversion par nos méthodes ; la grille de quart de résolution est la plus détaillée ne causant pas de dépassement de capacité matérielle lorsque chargée par nos méthodes.

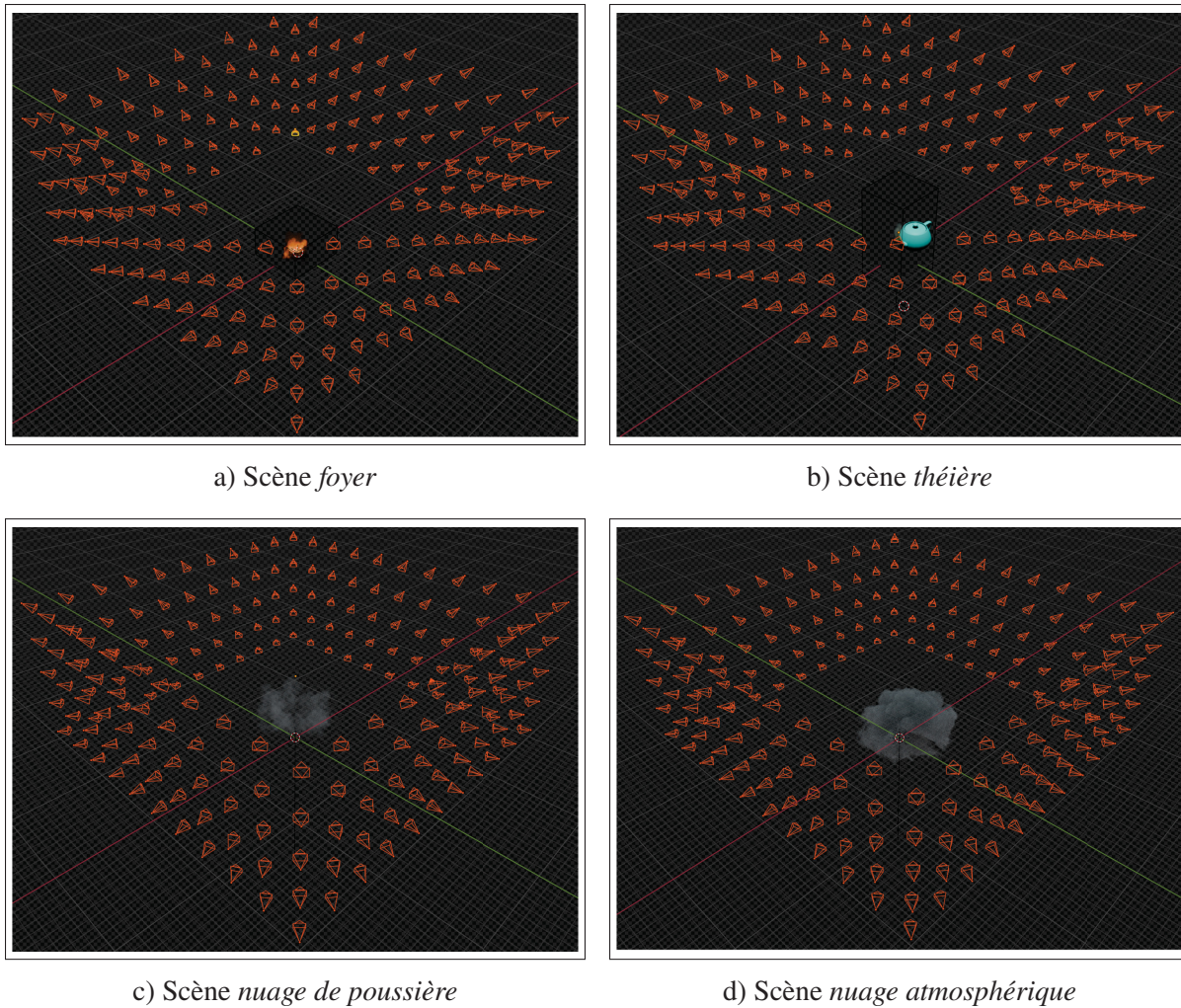


Figure 2.1 Scènes et disposition de caméras tétramurale

De plus, chaque scène vient sous deux configurations, soit la présence et l’absence d’un objet de support. Ces configurations visent à permettre l’évaluation de l’encodage de la transparence par les gaussiennes. La figure 2.1 illustre ces scènes et la disposition des caméras les observant. Les caméras sont disposées le long des « murs » d’une boîte invisible entourant la scène dans le but d’imiter ce qui se ferait dans un laboratoire de capture de mouvements. Les caméras y sont orientées de sorte à voir le sujet volumique sous plusieurs angles. Cette configuration « tétramurale » sera la principale disposition de caméras mise à l’épreuve dans nos tests. Nous sommes aussi capables de sélectionner le premier de ces quatre murs, de sorte à voir le sujet

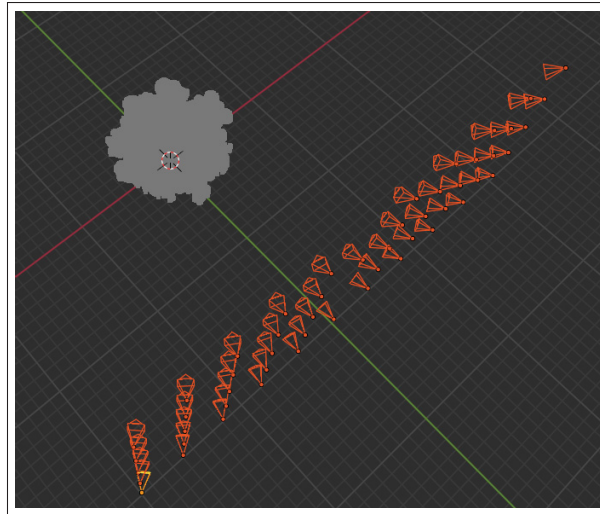


Figure 2.2 Scène *nuage de poussière* en disposition de caméras monomurale

d'un seul côté. Cette configuration « monomurale » sera utilisée en section 3.7 pour caractériser la difficulté du problème de reconstruction. Cette disposition est illustrée à la figure 2.2. Pour ces deux dispositions de caméras, nous réservons toutes les caméras dont l'identifiant est un multiple de huit afin de former l'ensemble de caméras de validation, et laissons les autres former l'ensemble de caméras d'entraînement. Toutes nos scènes et configurations furent créées ou assemblées à l'aide de *Blender* (Blender Online Community, 2026), un éditeur 3D libre et gratuit.

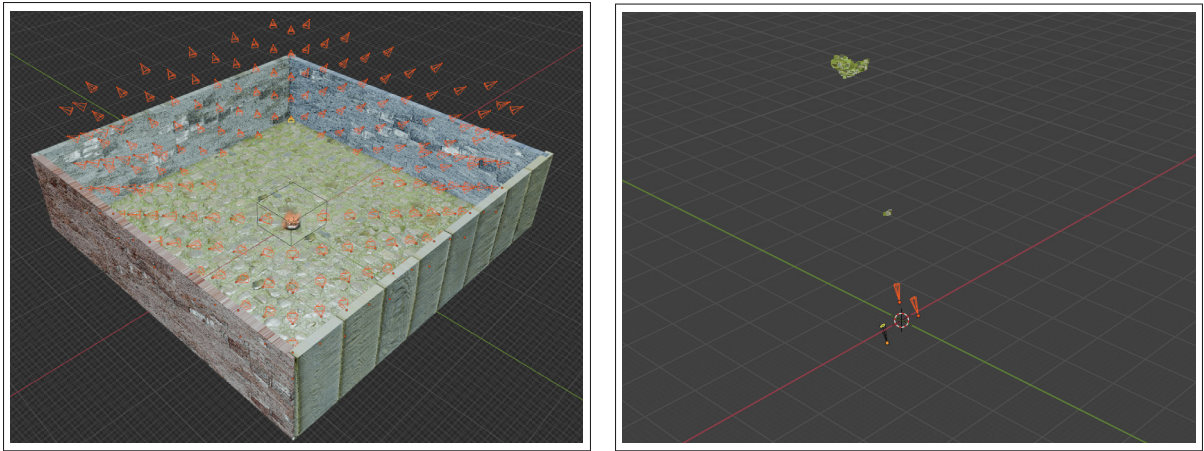
2.2 Découplage photogrammétrique

Comme mentionné dans la section 1.1, le succès de la reconstruction 3DGS dépend d'un point d'initialisation photogrammétrique composé d'un jeu d'images, du système de caméra calibré ayant produit lesdites images et d'un nuage de point 3D ayant été triangulé à partir des images et du système de caméra. La création de l'initialisation photogrammétrique est orthogonale à la contribution de 3DGS ; l'article cite le logiciel de photogrammétrie *COLMAP* (Schönberger & Frahm, 2016 ; Schönberger, Zheng, Pollefeys & Frahm, 2016b ; Schönberger, Price, Sattler, Frahm & Pollefeys, 2016a) comme point de départ de leurs

reconstructions sur scènes naturelles. Conséquemment, le format de fichier *COLMAP* est devenu une norme officieuse dans la communauté 3DGS, de sorte que la plupart des implémentations, dont l’implémentation de référence *3DGS-INRIA* (Kerbl *et al.*, 2023) et *gsplat* (Ye *et al.*, 2025), l’utilisent comme format d’initialisation principal. Ainsi, il apparaît nécessaire d’exporter nos scènes de *Blender* au format COLMAP.

Nous avons d’abord testé le flux de travail proposé par Kerbl *et al.* (2023). Cette approche est basée sur *COLMAP* et sa fonctionnalité de « *sparse reconstruction* (reconstruction partielle) ». En somme, nous produisons les rendus de la scène synthétique et les utilisons comme intrant à l’algorithme de reconstruction partielle. L’algorithme tente alors de régresser conjointement le système de caméra ayant observé la scène et la position 3D des points saillants de la scène. Or, nos expériences ont montré que nos scènes affichaient une performance médiocre avec cette approche. En effet, les volutes de vapeur offrent peu de points saillants, ce qui sous-contraint l’optimisation et mène *COLMAP* à reconstruire des scènes dégénérées. Ce problème peut être amoindri en ajoutant un environnement de support à nos simulations de vapeur (par exemple, quatre murs de briques mates de couleurs différentes et un sol gazonné, tel qu’illustré à la figure 2.3b). Cet environnement permettrait idéalement à l’algorithme d’identifier plus de points saillants, et permettrait donc à l’algorithme de compléter la reconstruction photogrammétrique partielle nécessaire à l’initialisation du moteur 3DGS.

La technique est fragile, puisque la reconstruction échoue souvent même dans des conditions optimales, telles qu’illustrées à la figure 2.3, et reste nuisible au reste du pipeline de reconstruction 3DGS dans les autres cas. Malgré l’augmentation de l’information géométrique d’une scène de référence à l’aide d’un environnement et d’un objet de support, comme présenté à la figure 2.3a, l’extraction échoue à identifier des points et donc à régresser conjointement la position desdits points et des caméras, ce qui produit une reconstruction dégénérée, comme présenté à la figure 2.3b. La technique force la reconstruction de parties de l’environnement inutiles au cas d’utilisation à l’étude, sous-échantillonne le vrai sujet de la reconstruction parmi les données d’entrées et ajoute une source d’ambiguïtés dans la radiance. Elle crée aussi, même lors du succès de la reconstruction partielle, un référentiel de caméra disjoint de celui utilisé pour créer

a) Scène *foyer* dans *Blender*

b) Reconstruction partielle dégénérée

Figure 2.3 Référence et reconstruction partielle de *foyer* avec environnement de support

la scène synthétique, ce qui est un obstacle à l'étude du comportement des gaussiennes sous la transformation géométrique.

Considérant l'inefficacité de l'approche photogrammétrique, et considérant que l'injection d'environnement de support dépend déjà fondamentalement de la nature synthétique de nos scènes, nous proposons de tirer pleinement parti de ladite nature synthétique afin de se débarrasser de la photogrammétrie. Cette approche repose dans un premier temps sur l'export de la paramétrisation des caméras de *Blender*, puis sur l'interprétation des données volumétriques fournies par le fichier *OpenVDB* produit par *Blender*.

Pour implémenter la conversion de *Blender* à *COLMAP*, nous utilisons le plugiciel *Photogrammetry Importer*, développé par Bullinger, Bodensteiner & Arens (2021). Ce plugiciel ajoute à *Blender* la capacité d'importer et d'exporter une foule de formats photogrammétriques, dont *COLMAP*. C'est le plugiciel qui s'occupe de convertir les notations vectorielles, le système d'axes, et le modèle physico-optique de caméra. Le détail de ces opérations est présenté à l'annexe I. Nos approches d'initialisation volumique seront présentées à la prochaine section.

2.3 Algorithmes d’initialisation volumique

L’initialisation de 3DGS est critique à son succès : 3DGS ne peut fonctionner en l’absence de points initiaux, puisque la méthode ne peut pas diviser et densifier des gaussiennes pour densifier l’espace si ces gaussiennes n’existent pas. Nos outils photogrammétriques que sont *COLMAP* et *Blender Photogrammetry Importer* étant incapables de manipuler des volumes, nous avons implémenté trois groupes de méthodes permettant de convertir un volume en données d’entrées pour 3DGS. Ces méthodes fonctionnent parallèlement et grâce à *Blender Photogrammetry Importer*, c’est-à-dire que le plugiciel est toujours responsable d’exporter le système de caméras au bon format ; nos méthodes ont pour unique rôle de former un nuage de point correspondant à la scène en se servant optionnellement des informations converties par *Blender Photogrammetry Importer*. Ces méthodes seront présentées dans les sous-sections suivantes.

2.3.1 Échantillonnage aléatoire et élagage basé sur la visibilité

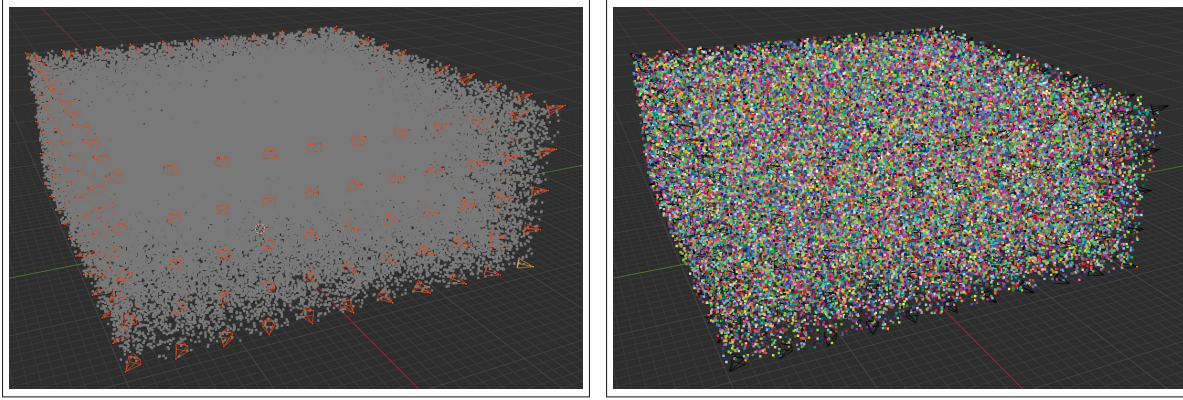
Le point de départ de nos contributions en initialisation de scènes volumiques est l’une des conclusions de Kerbl *et al.* (2023), confirmée par Kheradmand *et al.* (2024), selon laquelle un modèle gaussien de bonne qualité peut émerger d’une initialisation purement aléatoire. En effet, dans le type de scène à l’étude dans leurs travaux, soit une scène naturelle pleinement opaque et observée par un nuage de caméras disposées en orbite autour du sujet pour le voir sous tous ses angles, leurs études ablatives démontrent que le remplacement de points initiaux photogrammétriques par des points disposés aléatoirement dans la scène ne compromet pas la qualité de reconstruction du sujet principal ; le remplacement ne compromet que la qualité des arrière-plans. C’est un compromis tout à fait tolérable dans notre cas, puisque, dans l’idéal, nos reconstructions ne devraient même pas présenter d’arrière-plan.

Partant de ce constat, nous avons réimplémenté un algorithme de création de nuages de point uniformément aléatoire, dont le seul intrant est la position des caméras observant la scène. À la différence des implémentations actuelles, dont Ye *et al.* (2025), la création du nuage de point est un script antérieur au lancement du moteur 3DGS. Ce découplage de l’initialisation et du

chargement permet aussi d'utiliser le même nuage de points dans plusieurs expériences, ce qui améliore la répétabilité de ces dernières. Notre implémentation permet aussi de contrôler la couleur des points. L'algorithme fonctionne en recomposant la position des caméras, en sélectionnant leurs coordonnées de position minimales et maximales dans les trois axes spatiaux, en formant une boîte englobante alignée aux axes avec les coordonnées minimales et maximales, puis en instanciant des points colorés positionnés selon une distribution uniforme bornée par la boîte englobante. Cette implémentation repose cependant sur l'hypothèse selon laquelle chaque caméra est orientée vers le sujet de la reconstruction, comme c'est le cas pour nos dispositions tétramurale et monomurale. L'algorithme produit un nuage de points qui peut être encodé au format *COLMAP*, de sorte à pouvoir former une initialisation complète pour 3DGS lorsque ledit nuage est combiné avec les fichiers de paramétrisation des caméras. Le détail de cette opération est présenté à l'algorithme 2.1, tandis que la figure 2.4 illustre les points issus de l'exécution de cet algorithme.

Cependant, la méthode d'initialisation par une distribution de probabilité uniforme présentée ici a perdu certaines propriétés géométriques intéressantes par rapport à une initialisation photogrammétrique réussie. La première est que notre initialisation uniforme fait apparaître de la géométrie qui n'existe pas dans les images ; les efforts d'optimisation de la scène par 3DGS peuvent alors être mal distribués. La seconde est que certains points sont créés dans des régions de l'espace qui ne sont observées que par très peu de caméras ; les algorithmes de densification de 3DGS peuvent alors être incapables de modifier ou élaguer les gaussiennes issues de ces points parce qu'elles ne reçoivent pas de gradients. Le premier problème peut être résolu en tirant parti de l'information géométrique des grilles volumiques (sections 2.3.2 et 2.3.3), tandis que le second peut l'être en reconstruisant le champ de vision des caméras à partir de leurs paramètres intrinsèques. Nous proposons, à l'algorithme 2.2, un post-traitement s'ajoutant à l'algorithme 2.1 qui retire du nuage de points ceux qui sont visibles par trop peu de caméras.

L'algorithme d'élagage par test de visibilité minimale proposé à l'algorithme 2.2 compte le nombre de caméras pouvant voir chaque point, et élague lesdits points dont le compte est sous un seuil choisi. L'identification de la visibilité d'un point par rapport à une caméra est implémentée



a) Cent-mille points gris à position aléatoire

b) Cent-mille points à position et couleurs aléatoires

Figure 2.4 Points résultants de l'initialisation aléatoire

Algorithme 2.1 Initialisation de nuage de points à positions aléatoires

Entrées : Nombre de points à créer $p \in \mathbb{N}$;
 Extrinsèques de caméras $E \in \{\{\hat{\mathbf{q}}_1, \mathbf{t}_1\}, \dots, \{\hat{\mathbf{q}}_n, \mathbf{t}_n\}\}$;
 Type de sélecteur de couleur $c_t \in \{\text{aléatoire}, \text{unique}\}$;
 Si $c_t = \text{unique}$: couleur $c_{in} \in [0; 1]_3$
Sorties : Nuage de points $P = \{\{\mathbf{x}_1, \mathbf{c}_1\}, \dots, \{\mathbf{x}_p, \mathbf{c}_p\}\}$

```

1  $W \leftarrow \{-\text{quat2mat}(\hat{\mathbf{q}})^\top \mathbf{t} \mid \forall \hat{\mathbf{q}}, \mathbf{t} \in E\};$           /* Centre optique des caméras */
2  $\mathbf{b}_{min}, \mathbf{b}_{max} \leftarrow \min(W), \max(W);$ 
3 suivant  $c_t$  faire
4   | cas où aléatoire faire
5   |    $P \leftarrow \{\text{aléatoire}(\mathbf{b}_{min}, \mathbf{b}_{max}), \text{aléatoire}([0]_3, [1]_3)\}_p;$ 
6   | cas où unique faire
7   |    $P \leftarrow \{\text{aléatoire}(\mathbf{b}_{min}, \mathbf{b}_{max}), c_{in}\}_p;$ 
8 fin d'alternative
```

en projetant la position 3D du point vers l'espace-image, puis en testant si la position projetée tombe en dehors du rectangle-image formé de par la résolution de la caméra. Tel qu'évoqué, l'algorithme requiert de connaître les intrinsèques de chaque caméra (résolution et longueur focale) pour reconstruire leur pyramide de vue respective, ce qui le différencie de l'algorithme 2.1, mais aussi de la littérature existante en reconstruction 3D basée sur l'optimisation. Le post-

Algorithme 2.2 Élagage de points basé sur la visibilité minimale

Entrées : Visibilité minimale pour retenir un point v_{min} ;
 Extrinsèques de caméras $E = \{\{\hat{\mathbf{q}}_1, \mathbf{t}_1\}, \dots, \{\hat{\mathbf{q}}_n, \mathbf{t}_n\}\}$;
 Intrinsèques de caméras $I = \{\{\mathbf{r}_1, \mathbf{f}_1\}, \dots, \{\mathbf{r}_n, \mathbf{f}_n\}\}$;
 Nuage de points $P = \{\{\mathbf{x}_1, \mathbf{c}_1\}, \dots, \{\mathbf{x}_p, \mathbf{c}_p\}\}$
Sorties : Nuage de points élagué $Q = \{\{\mathbf{x}_1, \mathbf{c}_1\}, \dots, \{\mathbf{x}_q, \mathbf{c}_q\}\}$

```

1  $\mathbf{v} \leftarrow [0]_p$  ; /* Vecteur de visibilité des points */
2 pour  $\hat{\mathbf{q}}, \mathbf{t}, \mathbf{r}, \mathbf{f} \in E, I$  faire
3    $R \leftarrow \text{quat2rot}(\hat{\mathbf{q}})$ ;
4    $\mathbf{w} \leftarrow -R^T \mathbf{t}$ ;
5   pour  $\mathbf{x}, \mathbf{c} \in P$  faire
6      $\mathbf{x}' \leftarrow R(\mathbf{x} - \mathbf{w})$ ; /* translation et rotation vers l'espace-caméra */
7      $\mathbf{x}' \leftarrow \mathbf{f} \odot \frac{\mathbf{x}'_{xy}}{\mathbf{x}'_z}$ ; /* correction de perspective */
8      $v \leftarrow v + (\mathbf{x}'_z > 0 \ \& \ \prod_{i=1}^2 (2|\mathbf{x}'_i| < \mathbf{r}_i))$ ; /* devant la caméra et dans le rectangle-image */
9   fin pour
10 fin pour
11  $Q \leftarrow \{P_i \mid \forall i \in \{1, 2, \dots, p\}, v_i \geq v_{min}\}$ ;

```

traitement a aussi pour intrant un nuage de points tel que produit par l'algorithme 2.1, mais est tout à fait agnostique à sa provenance. Il fut choisi que l'élagage basé sur la visibilité minimale retranche simplement les points insuffisamment visibles sans en rajouter, et ce, dans le but d'étudier l'impact de points inutiles parce qu'ils contribuent peu et reçoivent peu de gradients. La figure 2.5 illustre le nuage de points résultant de l'élagage décrit dans la présente section en fonction du seuil de visibilité minimale utilisé. Nos résultats, présentés à la section 3.2, montrent que l'augmentation du seuil influe positivement sur la fidélité photométrique des modèles reconstruits, quoique cette corrélation positive s'estompe et forme un plateau passé le seuil de dix.

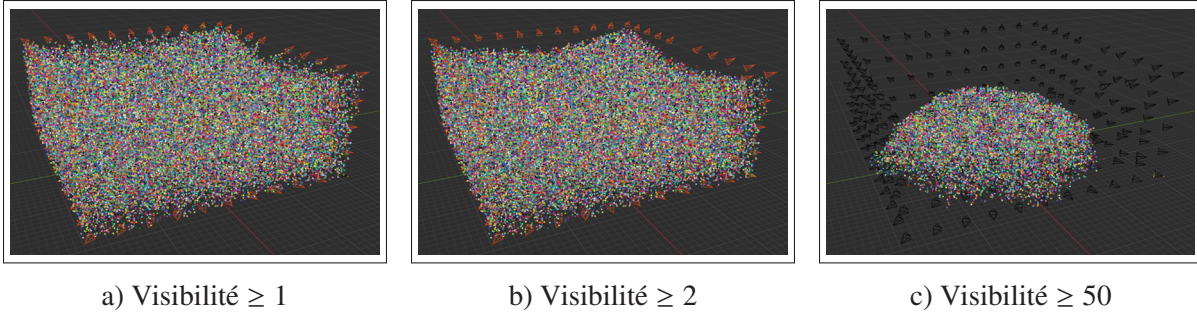


Figure 2.5 Points résultants de l'élagage par test de visibilité minimale

2.3.2 Échantillonnage du gradient de la densité

Tel qu'évoqué en sous-section 2.3.1, l'échantillonnage aléatoire peut inventer de la géométrie dans des zones de densité nulle, ce que 3DGS doit compenser en l'élaguant à l'optimisation. Pour corriger ce problème, nous cherchons à tirer parti de l'information géométrique présente dans nos simulations de fluides. Or, les expériences réalisées dans le cadre du présent mémoire montrent que les volutes de fumée et de vapeur sont les endroits dans les images affichant la plus grande erreur par rapport aux rendus de référence. De plus, les volutes sont aussi les endroits affichant les variations d'apparence et de gradient les plus franches. Il en découle que cette variation d'apparence est largement attribuable à une variation locale dans la densité. Conséquemment, nous formulons l'hypothèse que les volutes coïncident spatialement avec les endroits de haute variation du gradient de la grille de densité. Nous espérons que le suréchantillonnage des volutes initialisera des gaussiennes déjà proches de leur position optimale, réduisant l'espace d'exploration et d'optimisation et libérant des itérations pour le raffinement de l'apparence.

La méthode proposée dans la présente section tire parti du fait que *Blender* stocke chaque pas de temps de la simulation comme un fichier *OpenVDB* contenant une grille de densité. Par la suite, notre méthode emploie les bibliothèques *fVDB* (Williams *et al.*, 2024) et *PyTorch* (Ansel *et al.*, 2024) pour calculer le gradient spatial de la grille par différence finie de densité et calculer la norme de ce gradient. Le gradient spatial en chaque cellule est alors utilisé comme poids dans une fonction d'échantillonnage décidant du nombre de points alloué en chaque

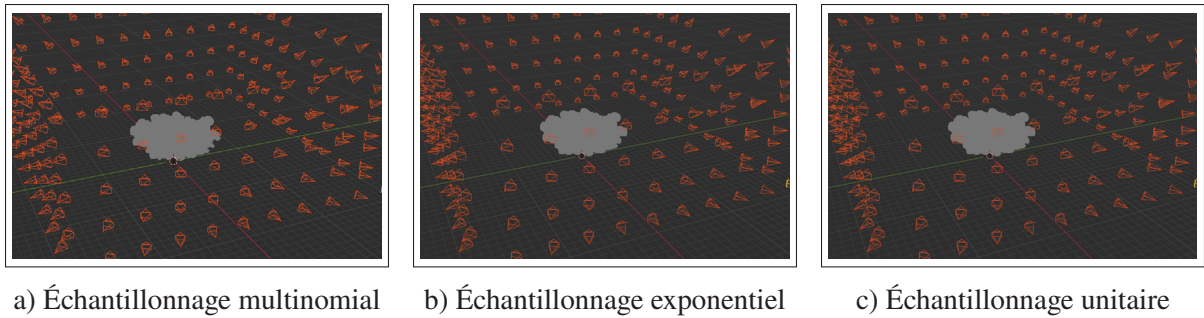


Figure 2.6 Points résultats des variantes de l'échantillonnage du gradient de la densité

cellule. Les points sont distribués dans leur cellule respective selon une distribution aléatoire uniforme pour éviter qu'ils ne se chevauchent, sauf s'ils sont seuls en leur cellule, auquel cas ils y sont positionnés au centre. Cette méthode est présentée à l'algorithme 2.3. De plus, trois stratégies d'allocation des échantillons furent implémentées et testées : l'échantillonnage multinomial, l'échantillonnage par interpolation exponentielle, et l'échantillonnage unitaire. Ces différentes implémentations cherchent à différencier l'apport seul d'une initialisation géométriquement robuste de celui du sous-échantillonnage et du suréchantillonnage. La sélection et le fonctionnement des fonctions d'échantillonnages sont détaillés en annexe à l'algorithme II-1. Les points résultants de l'exécution de ces trois variantes de l'algorithme d'échantillonnage du gradient de la densité sont illustrés à la figure 2.6.

2.3.3 Conversion hiérarchique de grille de densité

Quoique la méthode d'initialisation volumique présentée à l'algorithme 2.3 peut déployer une précision géométrique supérieure à celle permise par la méthode d'initialisation aléatoire (algorithme 2.1) augmentée de l'élitage basé sur la visibilité (algorithme 2.2), cette première emploie encore le point comme primitive géométrique à cause de la nécessité de se conformer au format *COLMAP*.

Or, les travaux de Sharma & Schmalstieg (2025) et de Condor *et al.* (2025) ont montré que l'utilisation de la gaussienne dès l'initialisation, comparativement à la conversion des points

Algorithme 2.3 Échantillonnage du gradient de la grille de densité

Entrées : Grille de densité $D \in \mathbb{R}^{x,y,z}$;
 Type d'échantillonneur s_t et ses paramètres P_s ;
 Amplitude du bruit ajouté aux points $b \in [0; 0,5[$;
 Transformation $\mathbf{t}, R, \mathbf{s}$ positionnant le volume dans la scène;
Sorties : Nuage de points $P \in \{\{\mathbf{x}_1, \mathbf{c}_1\}, \dots, \{\mathbf{x}_p, \mathbf{c}_p\}\}$

```

1  $\nabla D \leftarrow \text{gradientTrilinéaire}(D, \text{centroïdes}(D));$ 
2  $\mathbf{m} \leftarrow \text{échantillonner}(s_t, P_s, \|\nabla D\|);$  /* voir algorithme II-1 */
3  $C_m \leftarrow \text{colonnes}_{\mathbf{m}}(\text{centroïdes}(D));$  /* sélection par le vecteur d'indices  $m$  */
4  $B \leftarrow (\text{espaceVolume} \circ \text{aléatoire})(-b, b, \text{dim}(C_m));$  /* bruit uniforme */
5  $C_m \leftarrow C_m + B;$ 
6  $P \leftarrow \{\{R(\mathbf{s} \odot \mathbf{c}) + \mathbf{t}, [0,5]_3\} \mid \forall \mathbf{c} \in C_m\};$ 

```

en gaussiennes au chargement de 3DGS, permet de guider plus tôt l'apparence de la scène, de mieux capturer les structures anisotropes et de faire converger la reconstruction plus tôt. Dans le but de vérifier les conclusions de ces auteurs, la présente section présentera une réimplémentation de la conversion hiérarchique de grilles *OpenVDB* en gaussiennes d'abord proposée par Sharma & Schmalstieg (2025).

L'implémentation proposée ici est écrite en C++ et emploie la librairie *OpenVDB*, seule librairie permettant d'itérer sur les nœuds des différents niveaux de la grille hiérarchique tout en permettant de différencier les voxels actifs des voxels inactifs. En comparaison, *fVDB*, mentionnée plus haut, ne permet que d'itérer sur les voxels actifs. La présente implémentation permet de mettre à l'épreuve les trois niveaux de détail proposés par Sharma & Schmalstieg (2025), soit le niveau bas, le niveau moyen et le niveau élevé. Par contre, notre réimplémentation n'inclut pas les filtres interneœuds proposés dans l'article, puisque ces derniers sont d'efficacité limitée, de l'aveu des auteurs. L'algorithme 2.4 présente le comportement commun aux trois niveaux de détails, tandis que III-1, III-2 et III-3 présentent le comportement unique de chaque niveau de détail.

Les trois niveaux de détails encodent les gaussiennes au format *PLY*. Ce format fut choisi parce qu'il est déjà utilisé dans la communauté 3DGS pour encoder la reconstruction complète et la

Algorithme 2.4 Conversion hiérarchique de grille en gaussiennes

Entrées : Grille de densité $D \in \mathbb{R}^{x \times y \times z}$;
Niveau de détail $d \in \{\text{faible}, \text{moyen}, \text{élevé}\}$;
Transformation t, R, s positionnant le volume dans la scène
Sorties : Gaussiennes $G \in \{\{\mu_1, \hat{q}_1, s_1, c_1, o_1\}, \dots, \{\mu_g, \hat{q}_g, s_g, c_g, o_g\}\}$

```

1  suivant  $d$  faire
2      cas où faible faire
3           $B_{min}, B_{max}, d \leftarrow \text{grouperBoîtesEnglobantes}(D);$           /* voir III-1 */
4      cas où moyen faire
5           $B_{min}, B_{max}, d \leftarrow \text{grouperBlocsDenses4Ou2}(D);$           /* voir III-2 */
6      cas où élevé faire
7           $B_{min}, B_{max}, d \leftarrow \text{grouperBlocsDenses2}(D);$           /* voir III-3 */
8  fin d'alternative
9  pour  $i \in \{1, \dots, \#d\}$  faire
10      $\mu_i \leftarrow \frac{1}{2}(\text{col}_i(B_{min}) + \text{col}_i(B_{max}));$ 
11      $\mu_i \leftarrow R(G_{i,x} \odot s) + t;$ 
12      $s_i \leftarrow \frac{1}{2}(\text{col}_i(B_{max}) - \text{col}_i(B_{min}));$ 
13      $s_i \leftarrow \text{logit}(R(G_{i,s} \odot s));$           /* espace logit utilisé par 3DGS pour
        assurer une optimisation lisse */
14      $c_i \leftarrow [0,5]_3;$ 
15      $o_i \leftarrow \text{logit}(f(d_i));$           /* Tel qu'implémenté,  $f(d_i) = 0,6$ ; un point
        d'extension pour travaux futurs serait l'utilisation de la loi
        de Beer-Lambert */
16 fin pour

```

charger dans un logiciel de visualisation ; il permet donc déjà d'exprimer toutes les propriétés des gaussiennes (position, taille, rotation, couleur et opacité).

De plus, séparément de l'algorithme de conversion des grilles de densité en gaussiennes, l'implémentation de 3DGS utilisée pour exécuter les reconstructions dut aussi être modifiée. Puisque *gsplat* ne sait interpréter que les reconstructions partielles *COLMAP* pour initialiser la reconstruction gaussienne, il fut nécessaire de lui ajouter un module de chargement direct de gaussiennes. Le module, lorsqu'activé, charge les caméras depuis la reconstruction partielle *COLMAP*, mais court-circuite le chargement du fichier de points et la conversion des points en gaussiennes, notamment les étapes de dimensionnement des gaussiennes par la distance moyenne aux gaussiennes voisines.

En somme, les méthodes présentées ici permettent l’export de disposition et de paramétrisation de caméras utilisé dans l’environnement de simulation et de rendu *Blender*, et la production d’un nuage de points robuste pour l’initialisation de l’optimisation 3DGS. Le chapitre 3 fera la démonstration que les méthodes proposées permettent des reconstructions de volumes jusqu’ici impossibles en utilisant l’étalage gaussien.

2.4 Paramétrisation de l’implémentation de référence

Nos travaux sur 3DGS sont basés sur l’implémentation *gsplat* (Ye *et al.*, 2025). Il s’agit d’une implémentation libre d’utilisation basée sur *PyTorch* et *CUDA*, et suivant une architecture modulaire. Cette implémentation est aussi régulièrement maintenue et intègre périodiquement des rétroportages des articles d’amélioration à 3DGS, tant du côté du module de rendu que de celui de l’optimisation photométrique. Ces propriétés expliquent la décision de baser nos méthodes sur *gsplat*.

Nos expériences nous ont menés à utiliser l’algorithme de densification basé sur le gradient homodirectionnel (Ye *et al.*, 2024) fournie par *gsplat*. Nous avons sélectionné cette variante de la méthode comme implémentation de référence et de base à la suite d’une série d’expériences comparant objectivement les reconstructions permises par les densifications 3DGS (Kerbl *et al.*, 2023), 3DGS-MCMC (Kheradmand *et al.*, 2024) et AbsGrad (Ye *et al.*, 2024), ainsi que celles permises par la désactivation de la densification. Les résultats de ces expériences seront présentés en section 3.3. Selon nos résultats, AbsGrad produit des modèles de qualité photométrique légèrement supérieure à 3DGS, tandis que 3DGS-MCMC et la densification désactivée produisent au mieux des modèles flous et échouent souvent à produire un modèle. Comme 3DGS-MCMC utilise une faible opacité des gaussiennes comme signal d’exploration, l’algorithme ne parvient jamais à stabiliser leur position en l’absence d’un volume opaque, ce qui résulte au mieux en la reconstruction de modèles flous et au pire en l’absence de modèle.

Par ailleurs, dans sa configuration et ses implémentations de référence (Kerbl *et al.*, 2023 ; Ye *et al.*, 2025), 3DGS encode la couleur de chaque gaussienne dans une harmonique sphérique

réelle de degré trois afin de permettre la reconstruction d'effets lumineux anisotropes, telle la réflexion spéculaire directionnelle. Or, une harmonique de degré trois requiert seize fois plus de coefficients que celle de degré zéro, soit quarante-huit coefficients contre trois. En fait, le nombre de coefficients suit la règle $(l + 1)^2$, où l est le degré d'harmoniques. Accroître le nombre de coefficients accroît quadratiquement les besoins en stockage, en vitesse de communication entre le processeur central et le processeur graphique, ainsi qu'en vitesse de calcul. N'utiliser que des harmoniques de degré zéro est équivalent à un encodage RVB isotrope et ne modélise donc que la composante lambertienne de la couleur des surfaces. Cette perte de pouvoir de représentation a déjà été quantifiée dans les études ablatives de Kerbl *et al.* (2023), mais la perte ne se traduit que par une légère baisse de qualité photométrique par rapport à la reconstruction par harmoniques de degré trois. En contrepartie, l'utilisation d'harmoniques de degré zéro réduit le risque de créer des artefacts colorés dans les reconstructions qui n'apparaissent que dans certaines vues (Hull *et al.*, 2025).

Considérant l'objectif premier du présent mémoire, soit le transcodage des effets volumiques en gaussiennes 3D à des fins de modélisation additive, les harmoniques sphériques de degrés supérieurs soulèvent un problème d'ingénierie. Les scènes transcodées sont destinées à subir des transformations rigides dans le cadre d'un processus de modélisation additive et d'animation. Or, les harmoniques sphériques sont précisément utilisées pour encoder la directionnalité de la radiance des surfaces. Si ces surfaces subissent une rotation par l'intermédiaire de la rotation de la scène parente, il est nécessaire de faire subir le même pivot aux harmoniques pour préserver la cohérence de l'apparence desdites surfaces.

Quoique la littérature existante montre qu'il est possible de procéder à la rotation d'harmoniques réelles (Ivanic & Ruedenberg, 1996 ; Pinchon & Hoggan, 2007), les travaux en animation de modèles gaussiens suggèrent plutôt de n'utiliser que des harmoniques de degré zéro (Wang *et al.*, 2024 ; Keetha *et al.*, 2024). Compte tenu de ces recommandations et observations, et étant donné que les phénomènes à l'étude dans le cadre du présent mémoire sont majoritairement dénués de reflets spéculaires ou iridescents, nous avons cherché à quantifier l'apport des harmoniques de

degrés supérieurs à la qualité de la reconstruction avant de chercher à implémenter des solutions complexes. Cette expérience sera présentée à la section 3.4.

2.5 Régularisations

La reconstruction 3DGS s’articule autour d’une boucle de minimisation d’une fonction de perte par descente de gradient. L’un des défauts caractéristiques de la descente de gradient est le blocage de l’optimisation dans un minimum local, ce qui est aggravé par une initialisation trop éloignée de la solution optimale. Or, nos initialisations aléatoires naïves constituent un exemple d’initialisation éloignée. L’une des manifestations les plus fréquentes de ce blocage dans les scènes étudiées dans le cadre du présent mémoire est la formation de « gaussiennes-lentilles » : un agglutinement de gaussiennes devant les caméras d’entraînement pour compenser les erreurs dans la portion de gaussiennes visible par toutes les caméras. Ces gaussiennes sont parfois colorées, ce qui fausse la couleur de la portion centrale de la reconstruction. Les gaussiennes-lentilles constituent un cas de surajustement, soit une situation où la scène est surentraînée à être rendue depuis les caméras d’entraînement, au sacrifice de la qualité lors de la génération de vues nouvelles. La figure 2.7 illustre ce surajustement par lentilles gaussiennes : la sous-figure 2.7d montre que des gaussiennes approximent le nuage observé en la référence (sous-figure 2.7c), mais la reconstruction de la vue de validation adjacente 32 (sous-figure 2.7b) ne présente pas ce même nuage. On voit aussi en arrière-plan de la reconstruction des vues de validation (2.7b) et d’entraînement (2.7d) des taches grises disposées en grille régulière : ce sont toutes des lentilles gaussiennes disposées devant les autres caméras d’entraînement.

2.5.1 Régularisation de position

Pour combattre ce défaut d’optimisation, nous proposons d’introduire un nouveau terme de régularisation à la fonction de perte photométrique visant à tirer parti de la disposition des caméras dans la scène, c’est-à-dire que leur axe focal convergent vers le sujet central. Comme les gaussiennes bien optimisées devraient occuper une région délimitée de l’espace, nous proposons à l’équation (2.1) suivante, un terme de régularisation basé sur l’erreur quadratique moyennée

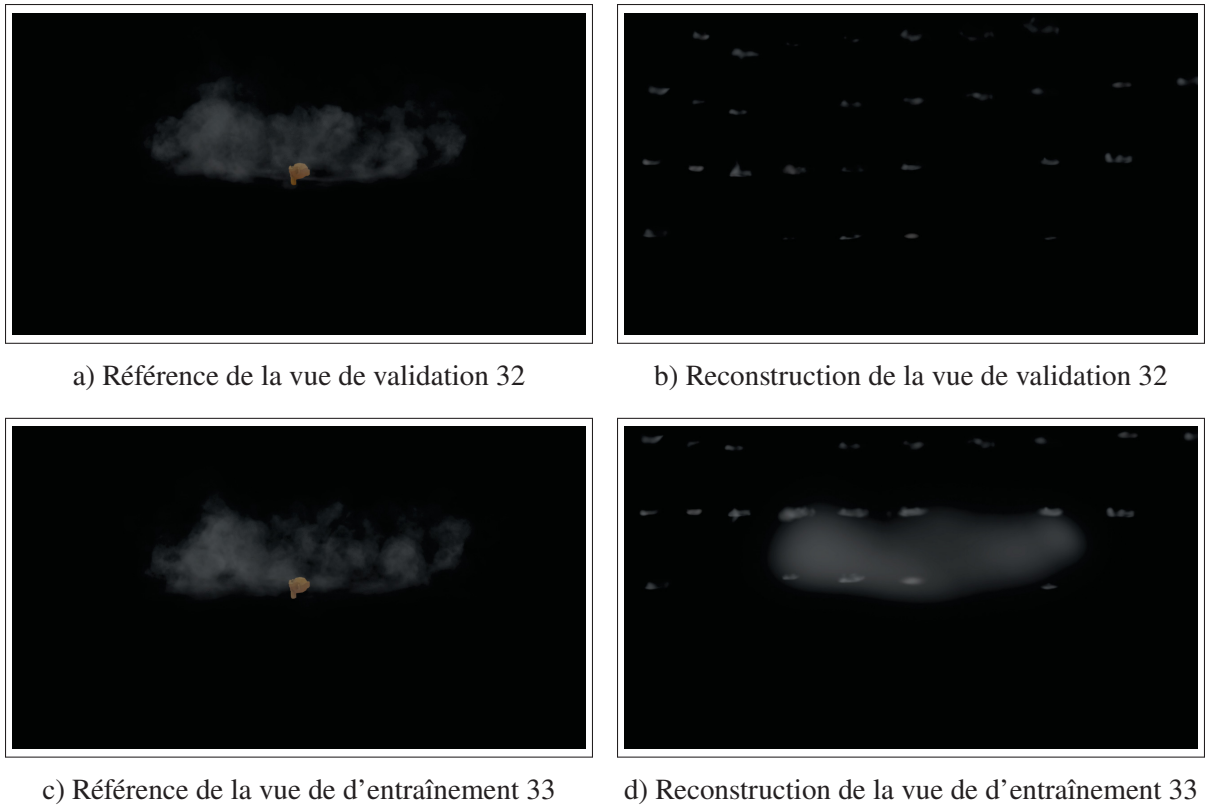


Figure 2.7 Surajustement par lentilles gaussiennes dans la scène *nuage de poussière*

sur toutes les gaussiennes et les trois composantes spatiales :

$$\bar{\mu} = \frac{1}{\#G} \sum_{i=1}^{\#G} \mu_i \quad (2.1a)$$

$$L_p = \frac{\lambda_p}{3\#G} \sum_{i=1}^{\#G} (\mu_i - \bar{\mu})^\top (\mu_i - \bar{\mu}) . \quad (2.1b)$$

Celui-ci fonctionne en pénalisant les gaussiennes qui s'éloignent du centroïde du nuage. Le coefficient de régularisation de position L_p est formé par la moyenne des composantes de la distance carrée moyenne de chaque centre de gaussienne μ_i par rapport au centroïde du nuage de gaussiennes $\bar{\mu}$, pondérée par la force de régularisation λ_p .

2.5.2 Régularisation d'anisotropie

Un autre des artefacts visuels présent dans les scènes de vapeur se manifeste sous la forme de gaussiennes pointues, trop anisotropes. Ces gaussiennes apparaissent comme de longs pics qui rompent la continuité de l'apparence des nuages et autres volumes translucides. Notre implémentation de référence, *gsplat* (Ye *et al.*, 2025), permet déjà de calculer un coefficient de régularisation de la taille des gaussiennes par le calcul de la moyenne des tailles dans toutes les dimensions. Quoique cette approche permette d'arrondir les gaussiennes, celle-ci présente l'effet secondaire de faire rapetisser les gaussiennes. Or, la taille des gaussiennes est un élément important de leur pouvoir de représentation, et aide à exprimer de grands pans de l'espace de façon compacte.

Pour permettre d'arrondir les gaussiennes sans restreindre leur taille, nous proposons un coefficient de réduction de l'anisotropie inspiré des travaux de Xie *et al.* (2024) :

$$L_{\text{aniso}} = \frac{\lambda_{\text{aniso}}}{\#G} \sum_{i=1}^{\#G} \frac{\max(\mathbf{s}_i)}{\min(\mathbf{s}_i)} - 1. \quad (2.2)$$

La régularisation L_{aniso} consiste à calculer l'élongation de chaque gaussienne. Cette élongation est calculée par le rapport de la longueur du plus grand et du plus petit axe de chaque gaussienne, la longueur de chaque axe étant indiquée par les composants de $\mathbf{s}$. Le rapport de longueur des axes est soustrait de un pour ne pas pénaliser les gaussiennes proches d'être parfaitement rondes avant d'être moyenné. Le coefficient d'anisotropie est alors ajouté à la fonction de perte totale, présentée à l'équation (1.4), avant que cette dernière ne soit rétropropagée. Notre approche se différencie de celle de Xie *et al.* (2024) par la façon dont l'objectif est formulé : la nôtre incite, avec un paramètre de force, les gaussiennes à devenir de plus en plus rondes, tandis que la leur pénalise les gaussiennes dont le ratio d'axe dépasse un paramètre de seuil.

Les régularisations présentées aux équations (2.1) et (2.2) se sont révélées inefficaces face à l'objectif premier, c'est-à-dire d'améliorer la qualité perçue des volumes reconstruits. Quoique la régularisation de position peut aider à réduire le nombre d'artefacts flottants près des caméras, elle

nuit à l’agencement des gaussiennes en objets cohérents. Quant à la régularisation d’anisotropie, elle tend à rendre les gaussiennes en moyenne plus rondes, mais aussi à accentuer l’anisotropie des gaussiennes les plus anisotropes. Globalement, ces régularisations sont inefficaces ou nuisibles, ce qui sera analysé numériquement dans la section 3.5.

2.6 Étalage quadratique

Comme mentionné au chapitre 1, la gaussienne n’est pas le seul noyau statistique utilisable dans un processus de rendu par étalage. Tel que relevé par Chen *et al.* (2024) et Condor *et al.* (2025), le support fini du noyau quadratique pourrait permettre des gains de performance et de mieux encoder les frontières nettes du sujet à reconstruire. La figure 2.8 illustre l’opacité des noyaux gaussiens et quadratiques 3D. On y voit que le noyau quadratique est plus opaque, plus loin de son centre, mais que la transition vers la transparence totale est beaucoup plus abrupte.

Or, les travaux de Condor *et al.* (2025) avaient déjà testé le noyau quadratique à la tâche du rendu volumique, mais avaient conclu que ce noyau produisait des rendus de fumée et de nuages de moindre qualité que le noyau gaussien. Toutefois, leur moteur de rendu étant basé sur le lancer de rayons, il est possible que ce compromis n’existe pas en rasterisation. Dans le but de confirmer ou d’infirmier les résultats rapportés par ces deux articles, nous avons réimplémenté l’étalage quadratique dans *gsplat* en modifiant les chemins de code *PyTorch* et *CUDA* appropriés pour le rendu et le calcul des gradients, et exposé le choix du noyau parmi les options de l’outil. Notre implémentation est inspirée des travaux de Chen *et al.* (2024). Tout comme pour le noyau gaussien, la boucle d’optimisation entière est exécutée avec le noyau quadratique comme primitive d’étalage. Les noyaux seront comparés à la section 3.6 sur la base de la qualité photométrique des reconstructions qu’ils permettent.

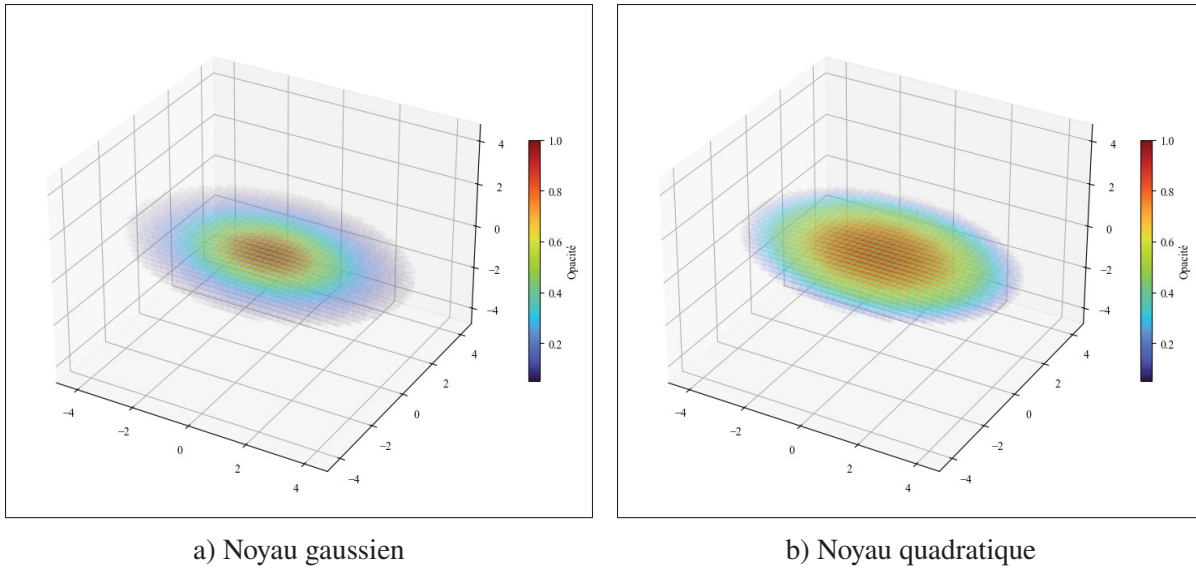


Figure 2.8 Opacité des noyaux gaussien et quadratique selon la distance à leur centre

2.7 Optimisation multifond

Les performances de reconstruction sur nos jeux de données sont sensibles à l’initialisation, comme montré en section 3.2. Ces résultats sont contraires à ceux rapportés par Kerbl *et al.* (2023), selon qui l’utilisation de points photogrammétriques robustes n’avait pour seul effet que l’amélioration de la netteté des arrière-plans. Dans nos scènes, l’utilisation d’une initialisation inappropriée peut soit empêcher l’optimisation d’arriver à terme, soit empêcher la reconstruction de converger en un modèle. C’est la raison principale ayant motivé le développement des méthodes d’initialisation par échantillonnage de volumes présentées en section 2.3. Autrement dit, actuellement, sans initialisation robuste, il est impossible de réaliser la reconstruction gaussienne d’un phénomène volumique, tel que soulevé au chapitre 2. Il existe donc un besoin pour une méthode qui permet de compenser les lacunes dans les données d’initialisation de la reconstruction.

Par ailleurs, l’un des défauts fréquents dont souffrent nos reconstructions de volumes translucides est le confondement de la reconstruction et du fond d’entraînement. Ce défaut se manifeste dans les zones de faible contraste entre le modèle et le fond. Par exemple, les ombres nettes du

dessous des objets de support causées par l'éclairage directionnel purement vertical ne sont pas encodées par des gaussiennes, mais plutôt par la couleur de fond noire. La couleur de fond paraît à l'écran telle quelle parce que l'optimisation n'a jamais eu besoin de la corriger avec des gaussiennes pour bien imiter l'apparence cible. Or, ce défaut est un surajustement à l'environnement d'entraînement qui empêche les modèles d'être réutilisés dans de nouveaux environnements ou en contexte de modélisation additive.

Pour corriger les lacunes soulevées ici, nous proposons une méthode d'entraînement multifond. Nous profitons de la nature synthétique de nos scènes pour en produire des rendus sur fond transparent. Nos rendus sur fond transparent peuvent alors être chargés par 3DGS et être composés par mélange alpha sur des fonds de nombre et de couleur quelconques. De même, l'étalage des gaussiennes à chaque itération d'entraînement est réalisé dans une texture transparente avant d'être composé sur les mêmes fonds que les rendus de référence. Pour atteindre cet objectif, le signal de perte photogrammétrique L_k , présenté précédemment à l'équation (1.4), se voit reformulé selon l'équation

$$L_{k;f} = (1 - \lambda_s)L_{1;k}(C_{k;f}, R_{k;f}) + \lambda_s L_{S;k}(C_{k;f}, R_{k;f}) \quad (2.3a)$$

$$L_k = \frac{1}{\#F} \sum_{f \in F} L_{k;f} . \quad (2.3b)$$

L'équation présente comment, pour chaque fond f , on compose l'image formée par l'étalage et le rendu de référence sur ledit fond pour former respectivement $C_{k;f}$ et $R_{k;f}$, avant de calculer la perte photométrique sur ledit fond, notée $L_{k;f}$. Tous les termes de perte photométriques par fond $L_{k;f}$ sont moyennés pour former la perte photométrique L_k , avant que cette dernière ne soit rétropropagée.

L'étude ablative de notre méthode d'optimisation multifond est présentée à la section 3.1. L'optimisation multifond est notre contribution la plus efficace et efficiente. Elle permet les meilleurs gains de fidélité photométriques et le meilleur débiaisement des modèles par rapport à leur environnement d'entraînement. Elle permet de multiplier la quantité d'information géométrique fournie par les images de référence sans demander plus de données d'entrée. De

plus, comme le temps d'entraînement est algorithmiquement dominé par le temps requis pour étaler les gaussiennes, l'ajout de fonds d'entraînement par simple copie de texture et mélange alpha ne coûte presque rien.

En somme, le présent chapitre a proposé le processus de recherche et les contributions qui en ont découlé. Les contributions s'organisent autour de l'interopération des données photogrammétriques et volumiques de l'éditeur 3D et du moteur 3DGS, ainsi qu'autour de la spécialisation du moteur 3DGS à l'optimisation de scènes volumiques. Plus précisément, les méthodes contribuées incluent l'export de la paramétrisation des caméras, la conversion des volumes en gaussiennes prenant en compte les caméras et la structure des volumes, des régularisations géométriques visant à réduire la fréquence de certains artefacts ainsi que l'optimisation multifond. Nous pensons que, lorsque combinées, nos modifications à la boucle d'optimisation de 3DGS produisent de meilleures reconstructions de volumes translucides que les méthodes précédentes. Le chapitre 3 mesurera les gains de fidélité permis par nos méthodes.

CHAPITRE 3

RÉSULTATS

Le présent chapitre présentera les expériences menées pour mesurer l’efficacité des méthodes proposées au chapitre 2. Lesdites expériences seront ordonnées selon leur contribution au succès de la tâche visée, c’est-à-dire la composition de scènes gaussiennes à sujet volumique avec des modèles polygonaux dans des environnements jamais vus à l’entraînement. Le chapitre sera conclu par la quantification du niveau de détail maximal pouvant être reconstruit par nos méthodes. Nous utiliserons principalement des indices photométriques pour comparer quantitativement nos méthodes.

Nous utiliserons le LPIPS (Zhang *et al.*, 2018), le MSE, le PSNR et le SSIM (Wang *et al.*, 2004). Le LPIPS est la « *Learned Perceptual Image Patch Similarity* (Similarité perceptuelle apprise de fragments d’image) » ; son algorithme projette, à l’aide d’un petit réseau de neurones préentraîné, deux images en espace vectoriel latent, et calcule l’indice LPIPS à l’aide de la distance entre les deux vecteurs, et plus le LPIPS approche de zéro, plus les images sont semblables. Le MSE est le « *mean square error* (moyenne du carré des erreurs) » ; il se calcule en soustrayant les images et en moyennant le carré de leur erreur, deux images ayant un MSE approchant de zéro sont très semblables. Le PSNR est le « *Peak Signal to Noise Ratio* (Ratio maximal du signal et du bruit) » ; il est formé par la conversion en décibels du MSE, et indique une meilleure reconstruction à mesure que l’indice croît vers l’infini positif et s’éloigne de zéro. Finalement, le SSIM est le *Structural Similarity Index Measure* (Index de similarité structurelle). Il se calcule, pour chaque pixel, par la convolution des différences des pixels adjacents à l’aide d’une convolution ; l’index est borné par zéro et un, et deux images sont similaires si leur SSIM s’approche de un.

3.1 Entraînement multifond

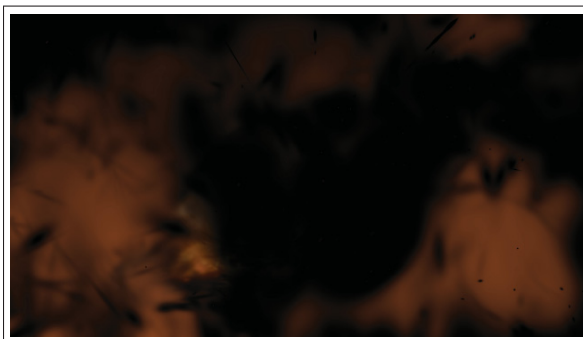
La présente section compare l’apport de la méthode d’entraînement multifond, présentée à la section 2.7, à celui de l’entraînement monofond constituant l’état de l’art. Pour mesurer cet apport, nous avons formulé une expérience dans laquelle nous avons reconstruit toutes

nos scènes, avec et sans objet de support, en faisant varier l’algorithme d’initialisation et les fonds d’entraînement. Nous utilisons le monofond noir pour imiter le processus d’entraînement antérieur à nos modifications, et un bifond noir et blanc pour permettre aux détails contrastant peu avec l’un des fonds d’être vus par l’autre sans favoriser de composante primaire de la couleur. Nous utilisons ici les suggestions de la littérature pour sélectionner l’algorithme de densification (AbsGrad) et le degré d’harmoniques sphériques (degré zéro) et mettrons à l’épreuve ces choix dans les sections 3.3 et 3.4. Nous présenterons dans la présente section les résultats qualitatifs et quantitatifs de cette expérience, dans cet ordre.

La figure 3.1 présente une comparaison qualitative de la scène *foyer* en fonction de la méthode d’initialisation et des fonds d’entraînement utilisés. Cette figure montre globalement que l’entraînement bifond aide à produire des modèles plus détaillés, plus cohésifs et plus découplés des fonds d’entraînement. La figure 3.1a montre que, lors de l’entraînement sur monofond noir, l’information géométrique fournie par l’initialisation aléatoire est insuffisante à la formation d’un modèle cohérent ; la figure 3.1c montre que l’entraînement sur bifond supplémente suffisamment l’information géométrique pour permettre au modèle de converger. De plus, l’entraînement bifond permet de mieux capturer les détails à faible contraste, comme les volutes de fumée noire, même lorsque la reconstruction est initialisée avec des données volumiques robustes, tel qu’illustré aux figures 3.1b et 3.1c. La figure 3.1 montre aussi que les modèles produits par l’entraînement bifond ont une apparence plus cohérente dans des environnements inconnus de l’entraînement. Par exemple, le modèle présenté à la figure 3.1b est surentraîné à être rendu sur fond noir et n’a donc jamais appris la frontière nette du bas du foyer ; ce défaut n’est pas présent dans les figures 3.1c et 3.1d.

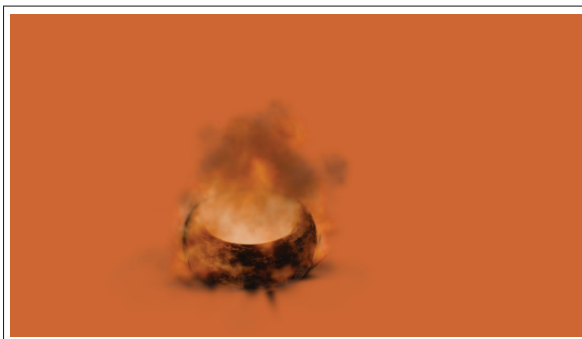
Le tableau 3.1 présente la qualité photogrammétrique moyenne en vue de validation des reconstructions composées sur un fond jamais vu à l’entraînement. Le tableau abonde dans le même sens que la figure 3.1, puisque, selon toutes les métriques, les reconstructions par entraînement multifond produisent des modèles plus fidèles à leur référence, peu importe l’initialisation utilisée. Nous en déduisons que l’entraînement bifond a au moins autant, sinon plus, d’impact sur la qualité de reconstruction que la robustesse géométrique de l’initialisation.

**Initialisation à position et couleur aléatoires,
visibilité minimale de 20**

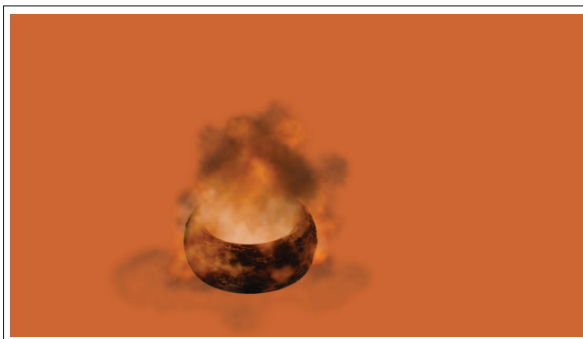


a) Monofond noir

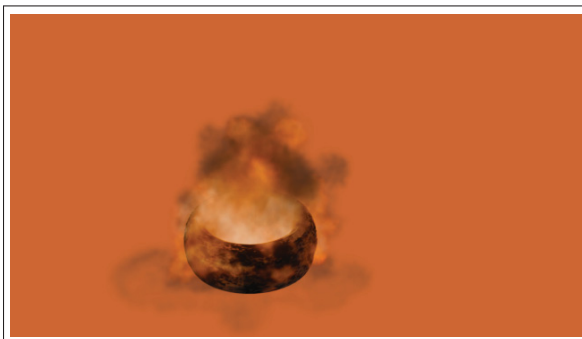
**Initialisation par échantillonnage
multinomial du gradient de densité**



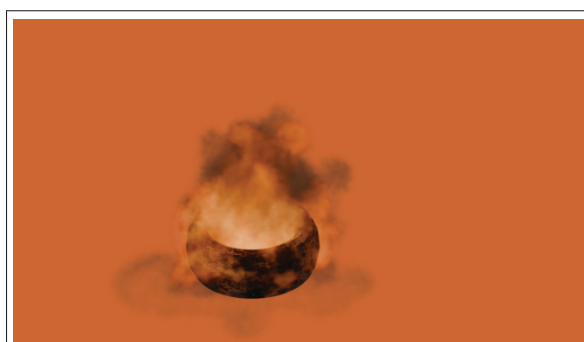
b) Monofond noir



c) Bifond noir et blanc



d) Bifond noir et blanc



e) Référence

Figure 3.1 Reconstruction de la scène *foyer* sur fond inconnu (orange) selon les fonds d'entraînement et l'initialisation

Considérant son efficacité, nous utiliserons l'entraînement bifond dans toutes les expériences suivantes.

Tableau 3.1: Qualité photométrique moyenne de toutes les vues de validation de toutes les scènes composées sur fond inconnu (orange) selon les fonds d’entraînement et l’algorithme d’initialisation

Algorithme d’initialisation	Fond(s) d’entraînement	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
100 k points à position et couleur aléatoires, visibilité minimale de 20	Noir	0,764 446	17 989,4	5,722 47	0,277 034
	Noir et blanc	0,071 256 1	178,292	36,971 4	0,942 310
Échantillonnage multinomial du gradient de la densité	Noir	0,081 684 4	681,672	32,990 6	0,937 980
	Noir et blanc	0,034 437 6	21,108 8	41,819 0	0,982 289

↓,↑: Sens d’optimisation de la métrique

Gradient rouge-vert: Gradient des pires aux meilleurs résultats

3.2 Algorithme d’initialisation

Cette section présentera l’impact des méthodes d’initialisation présentées à la section 2.3. Similairement au protocole de test présenté à la section 3.1, nous avons formé un jeu de données de test comportant nos scènes *foyer*, *théière* et *nuage de poussière* dans leurs variantes avec et sans objet de support. Nous avons reconstruit ce jeu de données en utilisant chacun de nos algorithmes d’initialisation et leurs variantes, mais en utilisant toujours l’optimisation bifond noir et blanc, la densification AbsGrad et des harmoniques sphériques de degré zéro. Nous présenterons d’abord une évaluation qualitative de ces méthodes, puis une évaluation quantitative des indices photométriques.

La figure 3.2 présente les reconstructions de la scène *nuage de poussière* par tous les algorithmes d’initialisation et leurs variantes respectives. Les figures présentant les algorithmes robustes (3.2f, 3.2h, 3.2i, 3.2j et 3.2k) montrent des modèles cohésifs à l’apparence plausible. Les figures présentant les algorithmes purement aléatoires échouent à produire un modèle utilisable selon des modes différents : l’algorithme présenté à la figure 3.2a a formé des lentilles gaussiennes devant les caméras d’entraînement, lesdites lentilles se manifestent dans la figure comme des

amas gris ; l'algorithme présenté à la figure 3.2b a pu produire un modèle, mais a mal appris sa couleur et corrige la couleur avec des lentilles de gaussiennes rouges situées devant les caméras d'entraînement. Dans nos expériences, seules les initialisations à points de couleur aléatoire ont montré un mauvais apprentissage de la couleur. Nous en concluons que l'initialisation à points gris est résistante à cette défaillance. Par contre, comme les deux méthodes échouent à produire un modèle cohérent, il est impossible de conclure définitivement sur laquelle des deux distributions de couleur est la meilleure lorsque tous les autres facteurs sont égaux. Les figures présentant l'élagage par test de visibilité à faible seuil (3.2c, 3.2d et 3.2e) ont échoué de la même manière que 3.2a. L'échec de reconstruction à faible seuil de visibilité indique que, dans ces cas, la qualité des points est plus importante que leur nombre. La figure 3.2g affiche une performance plus mauvaise que les figures 3.2f et 3.2h, ce qui est aberrant à premier abord. Nonobstant cette dernière observation, la figure 3.2 montre une tendance claire : les algorithmes basés sur l'interprétation des données issues de fichiers *OpenVDB* affichent de meilleures performances, mais il est néanmoins possible de s'approcher de cette qualité en utilisant des points aléatoires si on procède à un test de visibilité rigoureux.

Le tableau 3.2 montre les résultats quantitatifs de l'expérience décrite en introduction de la présente section. Le tableau présente la qualité photométrique moyenne sur vues de validation et sur toutes les expériences selon l'algorithme d'initialisation. Les meilleurs algorithmes du point de vue de toutes les métriques sont les conversions hiérarchiques VDB à détail faible ou moyen et l'échantillonnage unitaire ou multinomial du gradient de la densité. Ces résultats indiquent que la reconstruction bénéficie peu ou pas de l'utilisation des informations de densité, puisque ces trois méthodes ne la prennent pas en compte pour distribuer les échantillons : ils ne prennent en compte que la position des voxels et des nœuds-feuilles actifs. Ils montrent aussi, par la performance plus faible de la conversion hiérarchique à détail élevé, qu'utiliser trop de primitives pour l'initialisation est nuisible, même si lesdites primitives sont allouées en fonction de données volumiques robustes. C'est donc que le positionnement robuste des primitives initiales est le facteur dominant de la qualité photométrique des modèles rendus. Tout comme relevé dans l'examen qualitatif, les reconstructions purement aléatoires affichent la pire

performance, mais l'élagage par test de visibilité augmente la qualité de ces reconstructions à mesure que l'on hausse le seuil de visibilité, quoiqu'on observe un plateau lorsque le seuil excède dix. Le tableau montre aussi qu'une visibilité minimale de vingt affiche de moins bons résultats que les visibilités minimales de dix et de cinquante, ce qui suggère que la mauvaise reconstruction soulevée lors de l'examen qualitatif n'est pas qu'une donnée aberrante due aux éléments aléatoires de l'algorithme d'optimisation. Il est plus probable que cette différence soit imputable à des erreurs dans la préparation des données de certaines scènes. Les résultats du tableau 3.2 permettent aussi d'affirmer numériquement que nos méthodes robustes basées sur l'interprétation de VDB performant mieux que nos méthodes naïves. Ces dernières peuvent néanmoins opérer des reconstructions de qualité comparable à celles des méthodes robustes à condition qu'elles soient combinées avec un élague par test de visibilité minimale à seuil d'au moins dix.

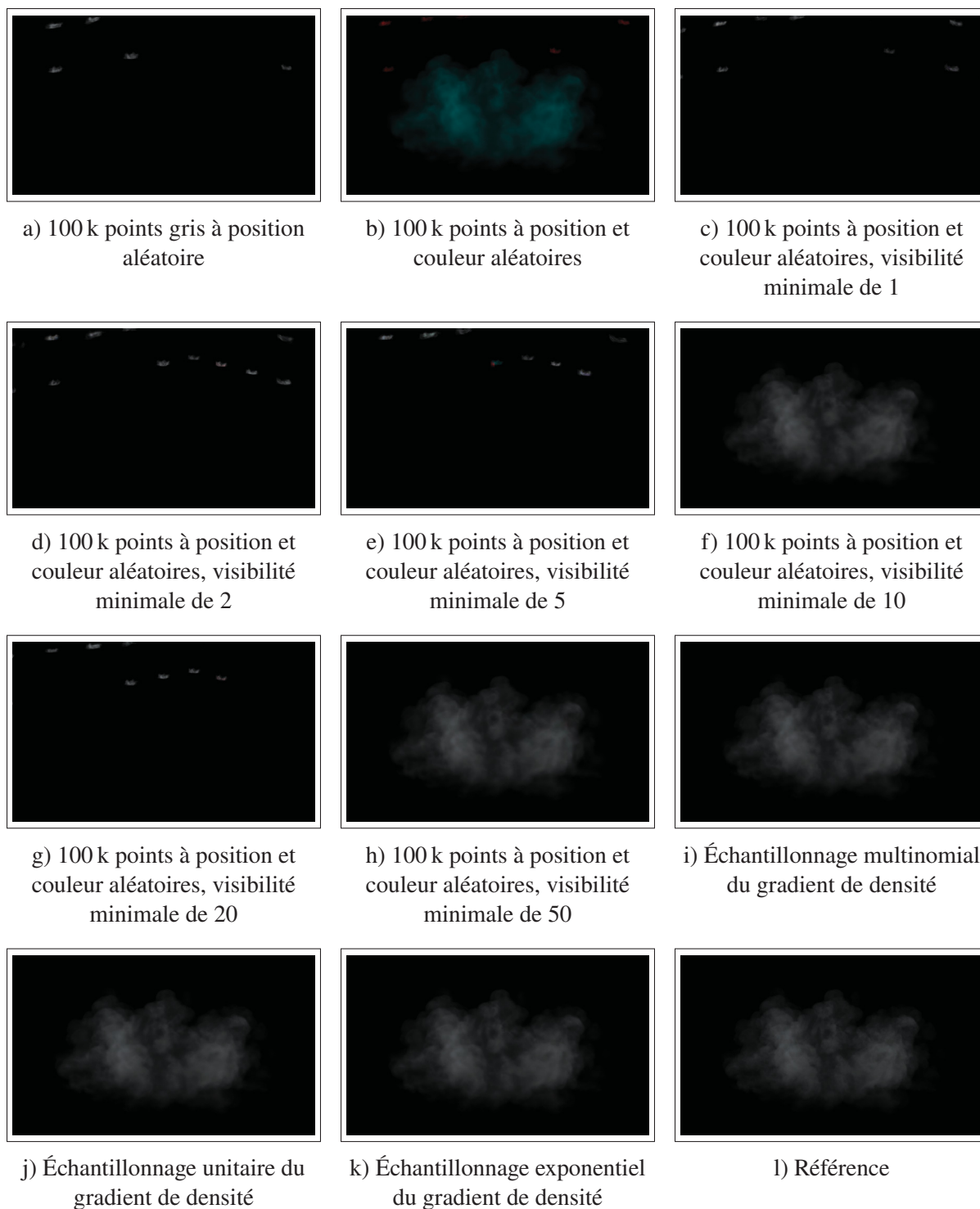


Figure 3.2 Reconstruction de la scène *nuage de poussière* selon l'algorithme d'initialisation

Tableau 3.2: Qualité photométrique moyenne de toutes les vues de validation des scènes *théière*, *foyer* et *nuage de poussière* selon la méthode d'initialisation

Algorithme d'initialisation	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
100 k points gris à position aléatoire	0,296 131	4 582,60	22,058 6	0,462 102
100 k points à position et couleur aléatoires	0,282 229	4 822,41	22,966 6	0,489 405
100 k points à position et couleur aléatoires, visibilité minimale de 1	0,204 362	1 753,04	25,719 8	0,480 673
100 k points à position et couleur aléatoires, visibilité minimale de 2	0,072 595 8	90,741 6	40,052 2	0,913 653
100 k points à position et couleur aléatoires, visibilité minimale de 5	0,045 520 1	68,013 7	47,718 5	0,954 066
100 k points à position et couleur aléatoires, visibilité minimale de 10	0,008 048 14	0,630 417	52,319 1	0,997 339
100 k points à position et couleur aléatoires, visibilité minimale de 20	0,041 434 5	64,508 8	48,363 0	0,956 736
100 k points à position et couleur aléatoires, visibilité minimale de 50	0,008 092 62	0,620 946	52,934 2	0,997 319
Échantillonnage multinomial du gradient de la densité	0,006 635 38	0,577 454	53,914 3	0,997 953
Échantillonnage unitaire du gradient de la densité	0,006 504 64	1,104 56	52,574 1	0,997 989
Échantillonnage exponentiel du gradient de la densité	0,006 524 80	0,566 661	53,219 8	0,997 982
Conversion hiérarchique à détail faible	0,006 360 69	0,568 357	53,346 4	0,998 020
Conversion hiérarchique à détail moyen	0,006 400 66	0,567 638	53,352 7	0,998 007
Conversion hiérarchique à détail élevé	0,006 578 57	1,125 70	52,538 7	0,997 929

3.3 Algorithme de densification

Nous mesurons dans le présent test l’apport de l’algorithme de densification à la cohérence géométrique et à la fidélité visuelle des modèles reconstruits. Pour ce faire, nous avons assemblé un jeu de données comportant toutes nos scènes, avec et sans objet de support, et les avons toutes reconstruites avec les trois algorithmes de densification et le témoin présentés à la section 2.4. Pour comparer justement les méthodes de densification, nous avons utilisé la même méthode d’initialisation pour toutes les scènes, soit l’échantillonnage multinomial du gradient de densité. Cette méthode fut choisie pour la qualité des modèles produits, mais aussi pour le contrôle qu’elle permet sur le nombre de points instanciés. Nous utilisons cette dernière capacité pour fixer le nombre de points initiaux à cent-mille. Nous utilisons également l’optimisation par bifond noir et blanc et des harmoniques sphériques de degré zéro. Nous évaluerons les résultats de cette expérience de manière qualitative et quantitative, dans cet ordre.

Les figures 3.3 et 3.4 présentent les scènes *nuage atmosphérique* et *théière* reconstruites par les densifications 3DGS, AbsGrad, 3DGS-MCMC et par la densification désactivée servant de témoin. Parmi les quatre algorithmes de densification, seuls 3DGS et AbsGrad n’échouent jamais à créer un modèle cohérent et net. La densification 3DGS-MCMC tend, dans les scènes sans objet de support comme les scènes *nuage de poussière* et *nuage atmosphérique*, à produire des modèles flous ou à carrément échouer à en produire un, alors que les trois autres densifications réussissent à arriver à terme. La figure 3.3c présente une telle situation. Dans les scènes où l’initialisation volumique ne couvre pas l’objet de support, telle la scène *théière*, la densification désactivée n’arrivera pas à placer et raffiner l’apparence des gaussiennes de l’objet de support, laissant l’objet trop flou pour être utilisable. Une telle situation est observable à la figure 3.4d. Dans les scènes avec un grand objet de support opaque, 3DGS-MCMC réussit à converger en un modèle utilisable. Nous pensons que le fonctionnement de 3DGS-MCMC est incompatible avec nos scènes très peu opaques : comme 3DGS-MCMC fonctionne par la téléportation stochastique des gaussiennes selon un bruit inversement proportionnel à leur opacité centrale pour favoriser l’exploration, les gaussiennes sont continuellement téléportées et n’apprennent jamais le modèle central. Cette situation est amoindrie dans les scènes présentant un objet de support opaque

couvrant plus d'aire à l'image que le volume translucide. Comme les méthodes 3DGS et AbsGrad ne téléportent pas les gaussiennes selon le gradient de leur contribution visuelle, elles seraient capables de rester en place et de se raffiner.

Le tableau 3.3 présente les mesures photométriques de la qualité des modèles en fonction de l'algorithme de densification utilisé et de la présence d'un objet de support dans la scène. Le tableau met en lumière la performance supérieure de 3DGS et d'AbsGrad, avec un léger avantage pour ce dernier en l'absence d'objet de support. La densification 3DGS-MCMC est toujours la pire méthode au sein d'une catégorie de présence d'objet. La désactivation ne performe dans le même ordre de grandeur que 3DGS et AbsGrad qu'en l'absence d'objet de support, ce qui est conforme aux observations soulevées lors de la comparaison qualitative. Considérant qu'AbsGrad est la méthode performant le mieux en reconstruction de sujets purement volumiques, ces reconstructions étant l'objectif et le sujet principal du présent mémoire, nous retenons AbsGrad comme l'algorithme de densification de prédilection pour le reste des tests que nous effectuerons.

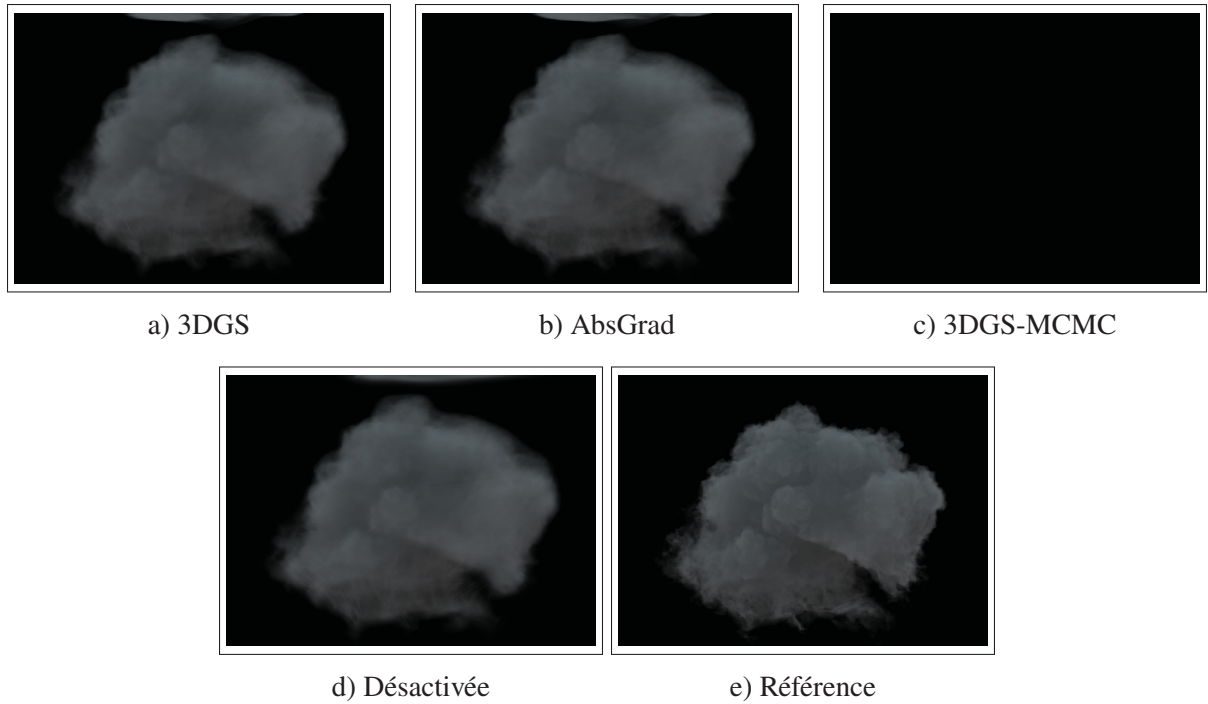


Figure 3.3 Reconstructions de *nuage atmosphérique* selon l’algorithme de densification

Tableau 3.3: Qualité photométrique moyenne de toutes les vues de validation de toutes les scènes selon l’algorithme de densification et la présence d’objet de support

Présence d’objet de support	Algorithme de densification	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Non	3DGS	0,069 052 4	112,072	42,630 9	0,962 484
	AbsGrad	0,065 851 0	112,088	42,921 3	0,962 734
	3DGS-MCMC	0,169 968	3 481,57	33,310 1	0,860 645
	Désactivée	0,071 961 4	109,944	41,972 3	0,961 902
Oui	3DGS	0,009 497 27	1,576 13	46,899 6	0,995 435
	AbsGrad	0,009 410 28	1,531 22	46,980 4	0,995 476
	3DGS-MCMC	0,037 311 5	8,184 41	41,982 3	0,983 999
	Désactivée	0,021 411 9	35,171 7	39,352 2	0,990 146



Figure 3.4 Reconstructions de *théière* selon l'algorithme de densification

3.4 Configuration des harmoniques sphériques

Cette section présentera une comparaison des reconstructions permises par l'utilisation d'harmoniques sphériques de différents degrés. Nous testerons l'hypothèse formulée à la section 2.4 selon laquelle nos scènes bénéficient peu du pouvoir de représentation anisotrope offert par les harmoniques de degrés supérieurs. Pour ce faire, nous avons reconstruits nos scènes *théière*, *foyer* et *nuage de poussière*, avec et sans objet de support, en utilisant des harmoniques de degré zéro, un, deux et trois, et en utilisant l'initialisation aléatoire à cent-mille points gris et l'initialisation par échantillonnage multinomial du gradient de la densité. Toutes les reconstructions utilisent la densification AbsGrad et sont composées sur un bifond noir et blanc.

La figure 3.5 présente les reconstructions de la scène *nuage de poussière* en fonction du degré d'harmoniques sphériques utilisé pour encoder la couleur des gaussiennes, et lorsque les reconstructions sont initialisées par échantillonnage multinomial du gradient de la densité. Les quatre figures y sont virtuellement identiques.

Le tableau 3.4 présente la qualité photométrique moyenne sur toutes les vues de validation de toutes les scènes du jeu de données en fonction du degré d'harmonique et de l'algorithme d'initialisation utilisés. Lors de l'utilisation d'une initialisation aléatoire naïve, le tableau ne montre pas de corrélation claire entre le degré des harmoniques et la qualité photométrique des reconstructions, puisque les degrés zéro et trois ont autant de fois le pire score. Lors de l'utilisation de l'échantillonnage multinomial robuste, la qualité affiche une corrélation positive avec le degré, puisqu'augmenter l'un fait augmenter l'autre selon toutes les métriques. Or, la différence entre la meilleure reconstruction et la pire se mesure au troisième chiffre le plus significatif sur six.

Ces résultats sont cohérents : autant la figure 3.5 que le tableau 3.4 présentent que les harmoniques de degré supérieur ont un impact limité sur la qualité des modèles reconstruits. Nous en concluons que l'utilisation des harmoniques de degrés supérieurs offre un avantage négligeable par rapport à leur coût de stockage plus élevé. Par ailleurs, le tableau 3.5 révèle que l'utilisation d'harmoniques de degrés supérieurs a un double effet négatif sur les besoins en stockage. Le premier, attendu,

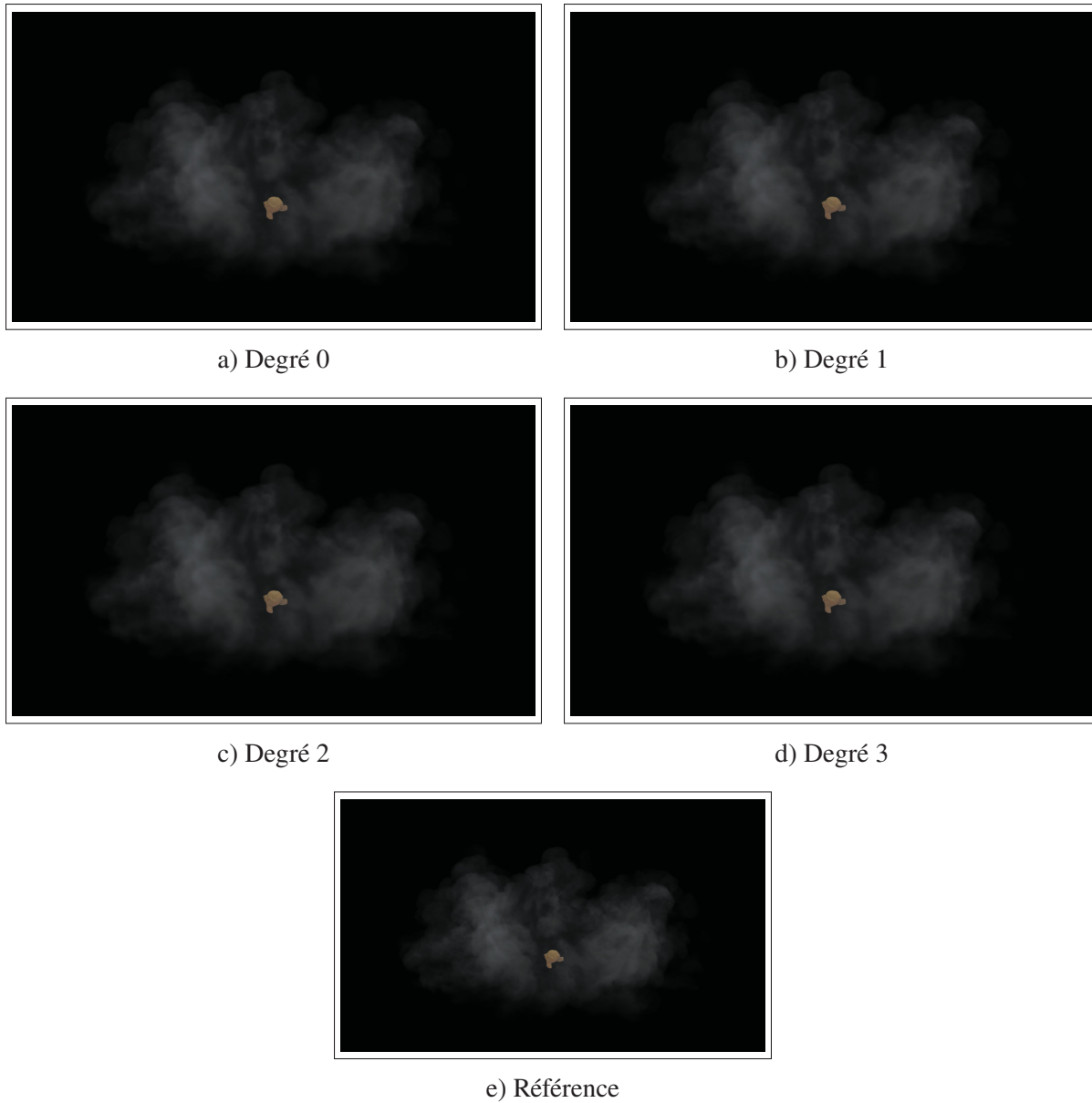


Figure 3.5 Reconstructions de *nuage de poussière*, initialisées par échantillonnage du gradient de la densité, selon le degré d'harmoniques sphériques

est qu'augmenter les harmoniques fait bien croître les besoins moyens en stockage. Le second, contraire à l'intuition selon laquelle l'utilisation d'une primitive plus puissante réduise le besoin en primitives pour encoder le même phénomène, on observe une corrélation positive du degré des harmoniques et du nombre de gaussiennes.

Tableau 3.4: Qualité photométrique moyenne sur toutes les vues de validation des scènes *théière*, *foyer* et *nuage de poussière* selon le degré d'harmoniques sphériques et l'algorithme d'initialisation

Algorithme d'initialisation	Degré d'harmoniques	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
100 k points gris à position aléatoire	0	0,279 446	5 454,63	19,104 5	0,462 091
	1	0,270 793	5 201,15	19,426 2	0,463 890
	2	0,277 310	5 370,32	19,241 8	0,462 906
	3	0,280 013	5 489,66	19,175 8	0,462 567
Échantillonnage multinomial du gradient de la densité	0	0,010 419 8	2,009 91	48,240 5	0,993 917
	1	0,010 350 1	1,932 71	48,875 0	0,994 093
	2	0,010 193 3	2,064 76	49,378 0	0,994 163
	3	0,010 091 2	1,902 61	49,462 9	0,994 219

Tableau 3.5: Coûts de stockage moyen des scènes *théière*, *foyer* et *nuage de poussière* selon le degré d'harmoniques sphériques et l'algorithme d'initialisation

Algorithme d'initialisation	Degré d'harmoniques	Nombre de coefficients par canal	Nombre moyen de gaussiennes ↓	Taille des harmoniques sur disque (megaoctets) ↓
100 k points gris à position aléatoire	0	1	51 463,7	0,617 564
	1	4	51 568,3	2,470 25
	2	9	51 450,3	5,558 07
	3	16	51 212,2	9,881 03
Échantillonnage multinomial du gradient de la densité	0	1	19 889,3	0,238 671
	1	4	20 319,5	0,954 686
	2	9	20 716,2	2,148 04
	3	16	21 374,7	3,818 74

Considérant le faible avantage numérique que procurent l'utilisation d'harmoniques sphériques de degrés élevés, leur coût de stockage croissant rapidement et leur corrélation positive avec le nombre de gaussiennes, nous croyons que l'utilisation de degrés supérieurs est, tout compte fait, antagoniste à l'objectif poursuivi. Nous recommandons, pour notre cas d'utilisation et dans des scènes volumiques isotropes comme les nôtres d'éviter la complexité liée à l'utilisation et la transformation rigide d'harmoniques sphériques, tel qu'hypothétisé en section 2.4.

3.5 Régularisations

Nous étudions dans cette section l'effet sur les reconstructions des régularisations présentées à la section 2.5. Pour mesurer leurs performances respectives, nous avons reconstruit toutes nos scènes, sans objet de support et initialisées par échantillonnage multinomial du gradient de la densité, tout en activant l'une ou l'autre des régularisations proposées et en faisant varier leur force. L'entraînement utilise l'optimisation sur bifond noir et blanc, la densification AbsGrad et des harmoniques sphériques de degré zéro.

Les figures 3.6 et 3.7 présentent des exemples de reconstructions organisées selon la régularisation utilisée et la force de ladite régularisation. La figure 3.6f montre qu'une forte régularisation de position peut faire échouer complètement la reconstruction, tandis que la figure 3.7f montre qu'elle ne permet pas d'amoindrir les artefacts flottants au haut de la reconstruction. Quant à la régularisation d'anisotropie, la figure 3.7f montre une silhouette aux contours plus lisses et une perte de netteté, ce qui indique que la régularisation réussit à rendre les gaussiennes plus rondes. Or, elle montre aussi une augmentation des artefacts piquants à la base du nuage, ce qui indique que la régularisation a pour effet secondaire d'accentuer l'anisotropie des gaussiennes les plus anisotropes.

Le tableau 3.6 abonde dans le même sens que les figures, puisqu'il rapporte que les deux régularisations produisent des reconstructions moins fidèles. Les résultats sont déjà négativement influencés à petite force et empirent d'autant à mesure qu'elle augmente. Tout compte fait, nous

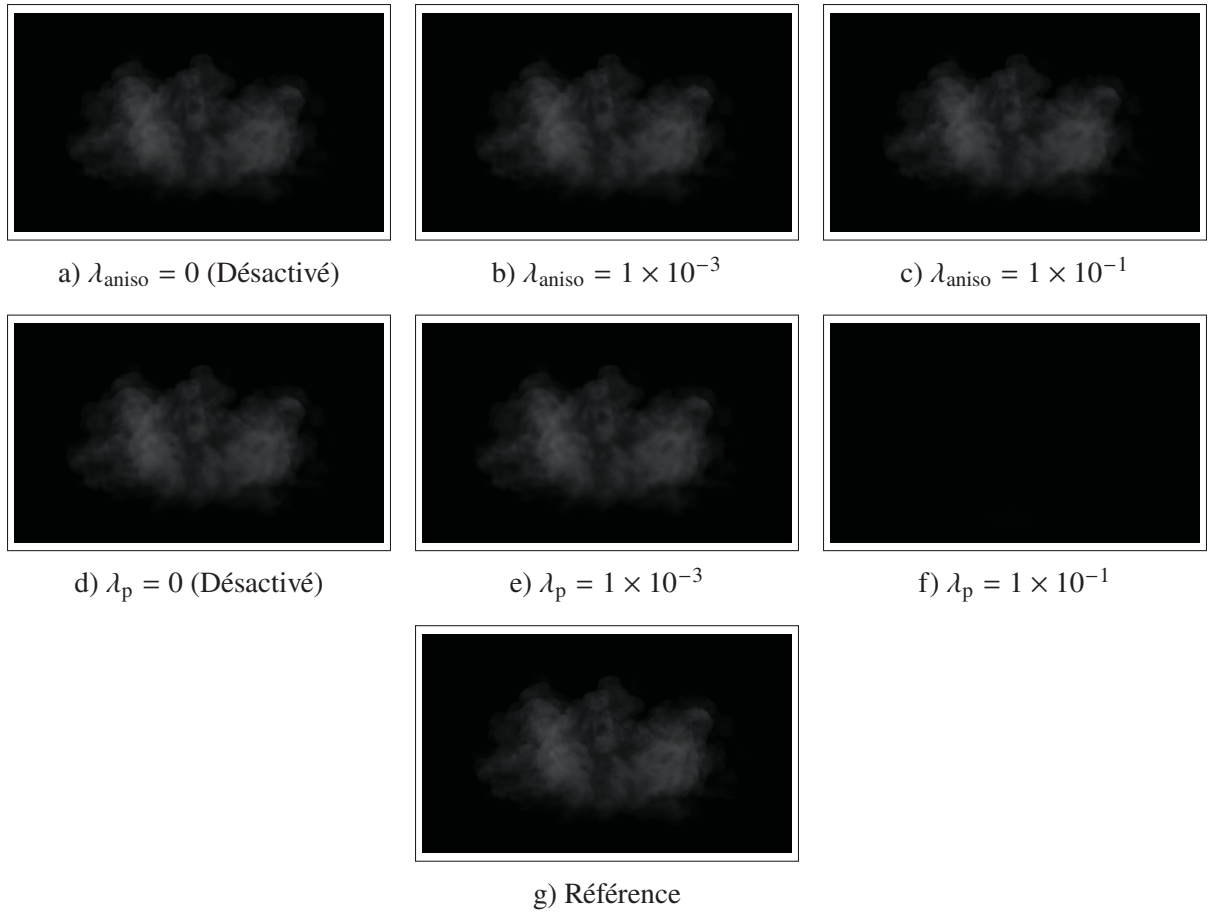


Figure 3.6 Reconstructions de *nuage de poussière* selon la régularisation et sa force

croyons que ces résultats indiquent que les régularisations de position et d'anisotropie sont inutiles ou nuisibles dans leur implémentation actuelle.

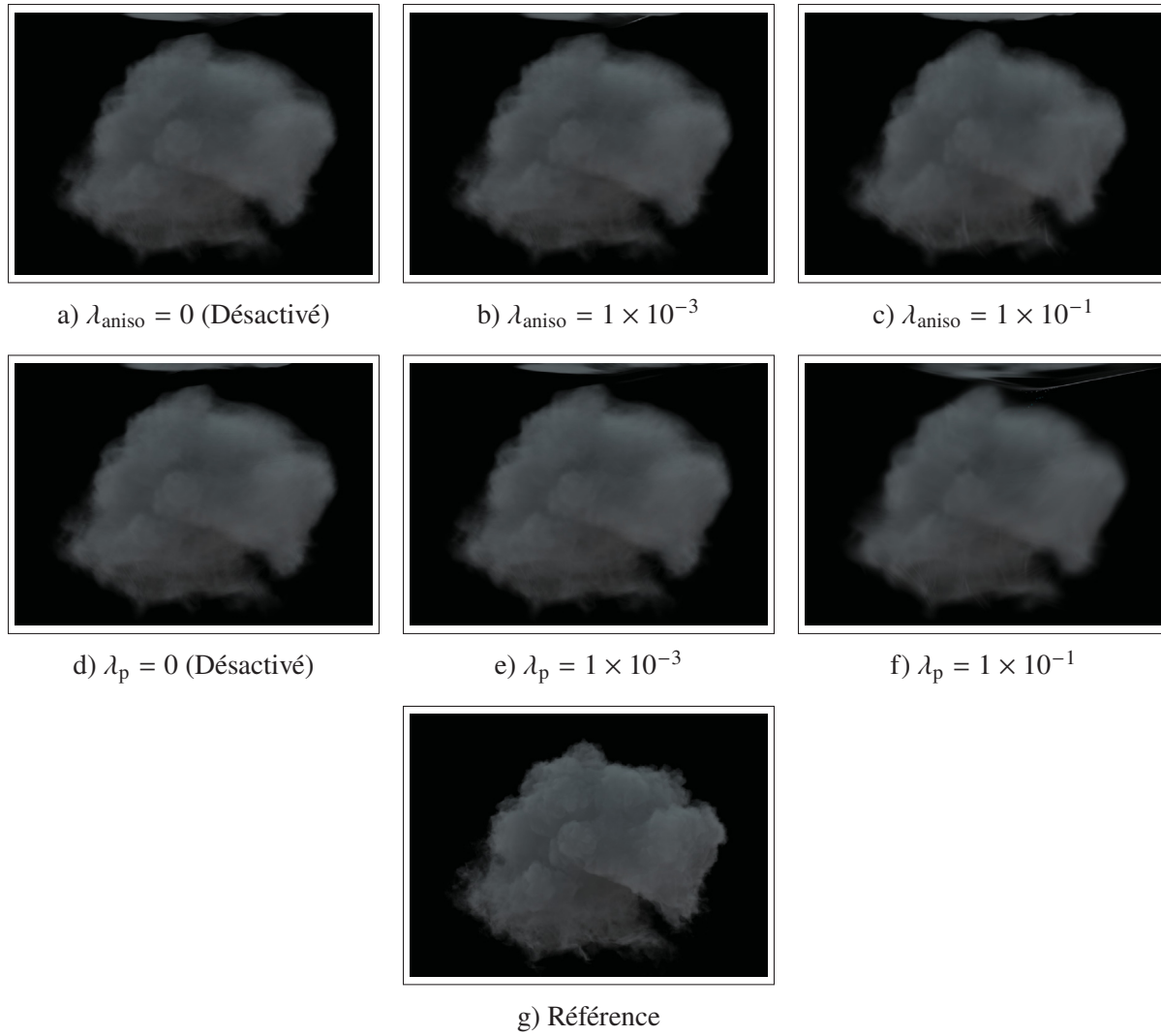


Figure 3.7 Reconstructions de *nuage de poussière* selon la régularisation et sa force

Tableau 3.6: Qualité photométrique moyenne sur toutes les vues de validation de toutes les scènes selon la régularisation et la force de régularisation

Régularisation	Force	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Anisotropie (λ_{aniso})	0	0,065 976 5	112,979	42,591 7	0,962 653
	1×10^{-3}	0,071 666 1	114,714	42,310 2	0,962 424
	1×10^{-1}	0,081 345 1	123,379	41,382 4	0,961 675
Position (λ_p)	0	0,065 934 9	113,313	42,549 9	0,962 574
	1×10^{-3}	0,073 112 5	114,868	39,450 7	0,961 570
	1×10^{-1}	0,155 601	1 035,86	26,032 9	0,898 788

3.6 Noyaux d'étalage

Cette section compare les reconstructions permises par notre implémentation du noyau quadratique à celles permises par le noyau gaussien. Nous avons pour objectif de confirmer ou d'infirmer l'applicabilité des conclusions de Chen *et al.* (2024) et de Condor *et al.* (2025) quant au choix du noyau en situation de rendu différenciable de volume par étalage, puisque les leurs sont respectivement restreintes au rendu de surfaces et au rendu par lancer de rayons. Pour mesurer la différence entre les deux noyaux, nous avons assemblé un jeu de données contenant toutes nos scènes, avec et sans objet de support, initialisées par l'échantillonnage multinomial du gradient de la densité. Nous avons reconstruit toutes lesdites scènes en utilisant l'optimisation bifond noir et blanc, la densification AbsGrad et des harmoniques sphériques de degré zéro. Nous avons toutefois reconstruit chaque scène deux fois, soit une fois avec chaque noyau.

La figure 3.8 présente les reconstructions de *thèière* lorsqu'on ne fait varier que le noyau d'étalage utilisé pour encoder la scène, tant en reconstruction qu'au rendu. Visuellement, les figures 3.8a et 3.8b sont virtuellement identiques. La figure 3.8d confirme cette observation, puisqu'on y voit que la différence absolue entre les rendus permis par les deux noyaux est extrêmement petite et concentrée sur la silhouette de l'objet de support. La région occupée par la vapeur présente une erreur pratiquement nulle.

Le tableau 3.7 présente le noyau à la meilleure qualité photométrique selon la scène et la présence d'un objet de support. Le tableau IV-1 présente les données brutes ayant permis la création du tableau comparatif 3.7. Ledit tableau met en lumière que la gaussienne performe mieux dans la plupart des cas. Le noyau quadratique a parfois mieux performé dans les scènes ne présentant pas d'objet de support, soit les deux scènes *nuage*. Cette situation, contraire aux conclusions de Condor *et al.* (2025) quant au fait que la transition nette du noyau quadratique est nuisible à la reconstruction des volumes, suggère que la frontière nette du noyau quadratique n'a pas autant d'impact sur la qualité photométrique lors de l'étalage d'une scène de volume. Tout compte fait, comme la gaussienne performe mieux dans la plupart des cas, elle constitue un meilleur choix

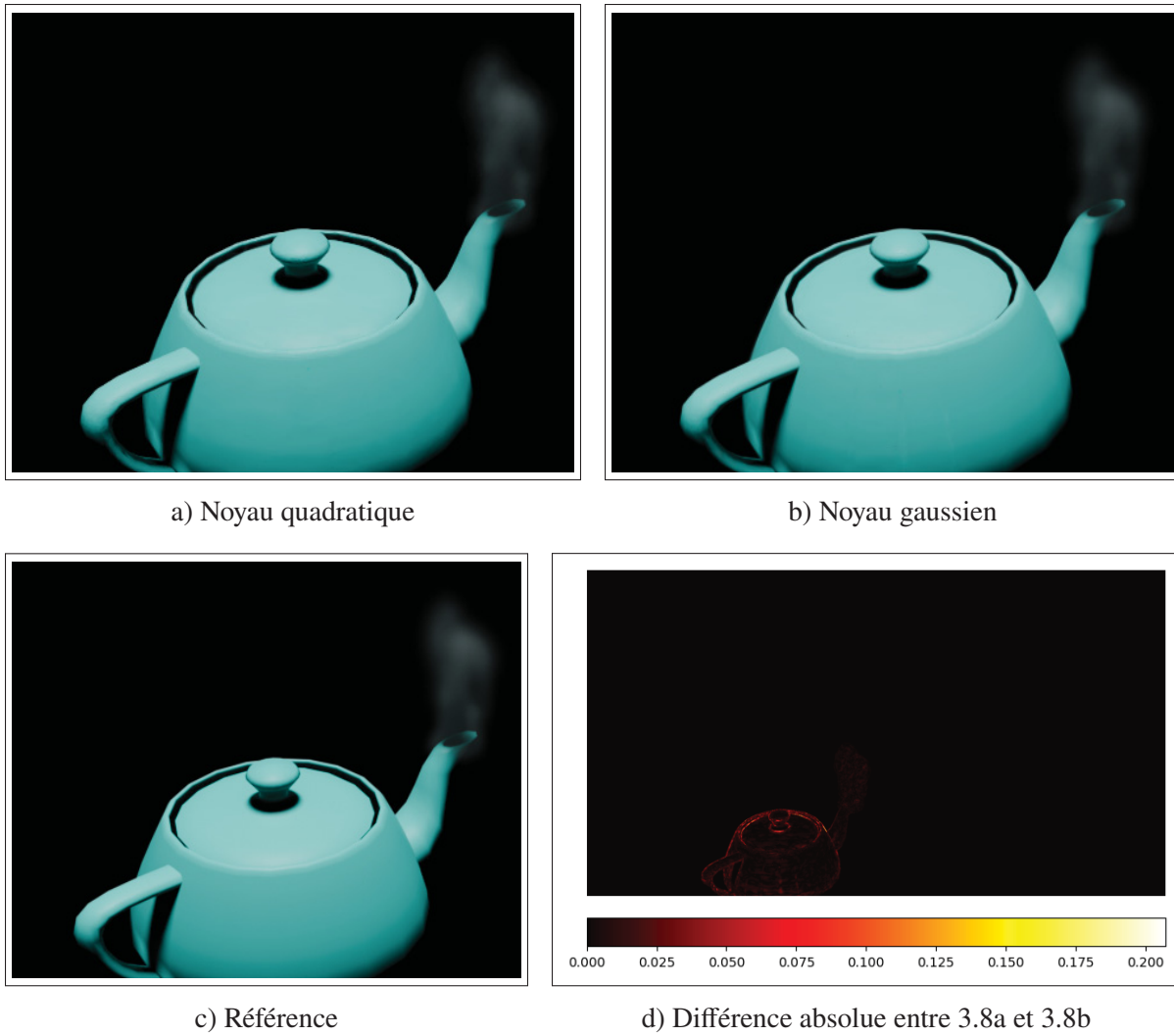


Figure 3.8 Reconstructions de *théière* selon le noyau d'étalage

par défaut, en particulier pour assurer que l'étude et la comparaison des autres méthodes est juste.

Tableau 3.7: Noyau présentant la meilleure qualité photométrique moyenne sur toutes les vues de validation selon la scène reconstruite et la présence d’un objet de support

Scène	Objet de support	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Nuage de poussière	Non	Quadratique	Quadratique	Quadratique	Gaussien
	Oui	Quadratique	Gaussien	Gaussien	Gaussien
Foyer	Non	Gaussien	Gaussien	Gaussien	Gaussien
	Oui	Gaussien	Gaussien	Gaussien	Gaussien
Théière	Non	Gaussien	Gaussien	Gaussien	Gaussien
	Oui	Gaussien	Gaussien	Gaussien	Gaussien
Nuage atmosphérique	Non	Quadratique	Gaussien	Gaussien	Gaussien

Cases bleues : Victoire du noyau gaussien

Cases oranges : Victoire du noyau quadratique

3.7 Niveau de détail maximal restructurable

À travers toutes nos expériences, nous remarquons que l’optimisation ne réussit pas à recréer précisément les transitions nettes caractéristiques des volutes de nos scènes de nuage, malgré les techniques déployés pour suréchantillonner lesdites volutes par rapport au reste du volume. Or, cette situation entre en contradiction avec les résultats de Kerbl *et al.* (2023), selon lesquels la gaussienne a une plus grande capacité à reconstruire la géométrie de haute fréquence que ses méthodes contemporaines. Qui plus est, Celarek *et al.* (2025) rapportent que l’étalage gaussien rastérisé peut reconstruire les scènes volumiques avec au moins autant de fidélité que les méthodes basées sur le lancer de rayon à condition que suffisamment de gaussiennes sont employées. Nous présentons ici une expérience visant à circonscrire la cause de la perte de netteté de nos modèles gaussiens par rapport à la référence. Pour mesurer la capacité de reconstruction maximale des détails fins et flous atteignable par 3DGS, nous userons des méthodes permettant les meilleures reconstructions, tel qu’étudié auparavant dans le présent chapitre, mais nous surspécialiserons aussi le modèle à être rendu d’un seul côté. En laissant le modèle se surspécialiser, nous pourrions mesurer la finesse de l’algorithme d’optimisation lui-même, sans autre interférence.

Nous assemblerons un jeu de données comportant toutes nos scènes, avec et sans objet de support, initialisées par échantillonnage multinomial du gradient de la densité, optimisées sur bifond noir et blanc, densifiées par AbsGrad et à la couleur encodées par harmoniques sphériques de degré zéro. Nous rendrons chacune des scènes de ce jeu de données avec la disposition de caméra tétramurale, utilisée jusqu'à présent dans toutes les expériences, mais aussi avec la disposition monomurale, toutes deux ayant été présentées en section 2.1. Comme cette dernière disposition possède moins d'angles de vues, le modèle a plus d'opportunités de spécialisation. Tel que présenté dans cette même section 2.1, nous usons du premier mur de caméras des quatre, faisant passer le nombre de caméras de deux-cent à cinquante, dans le but de maintenir la densité d'information visuelle, et usons similairement du premier quart des caméras de validation pour pouvoir comparer vue par vue la précision des reconstructions.

Le tableau 3.8 présente une comparaison photométrique de la qualité moyenne des reconstructions en fonction de la disposition de caméra et de l'algorithme d'initialisation. Les dispositions tétramurale et monomurale y sont présentées. Le tableau montre une fidélité accrue des modèles dont les vues de références sont disposées monomuralement. Nous rappelons que la fidélité est évaluée uniquement sur les vues de validation communes aux deux dispositions de caméras. Cette fidélité accrue par la disposition monomurale nous confirme que l'algorithme peut extraire plus de détail des reconstructions lorsqu'il n'a pas à se soucier d'être fidèle lorsque rendu sur 360 degrés autour de l'axe vertical.

Par contre, la différence numérique de performance présentée au tableau 3.8 n'est que légèrement perceptible dans les images présentées à la figure 3.9. Cette comparaison qualitative révèle que les reconstructions monomurales affichent plus fidèlement les turbulences locales visibles dans la référence, autant dans les vues d'entraînement que de validation. Par contre, autant les vues de validation que celles d'entraînement restent loin de la netteté de leur référence respective. La figure 3.10 montre que l'erreur est très diffuse, dépassant rarement les quatre centièmes. Or, Celarek *et al.* (2025) rapportent que la reconstruction de volumes par étalage gaussien affiche plus d'erreur en silhouette du volume lorsqu'il y a trop peu de gaussiennes pour

Tableau 3.8: Qualité photométrique moyenne sur toutes les vues de validation de toutes les scènes selon la disposition de caméras

Disposition de caméras	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Monomurale	0,004 615 21	0,370 553	55,729 3	0,998 332
Tetramurale	0,021 766 4	4,925 39	45,926 5	0,985 489

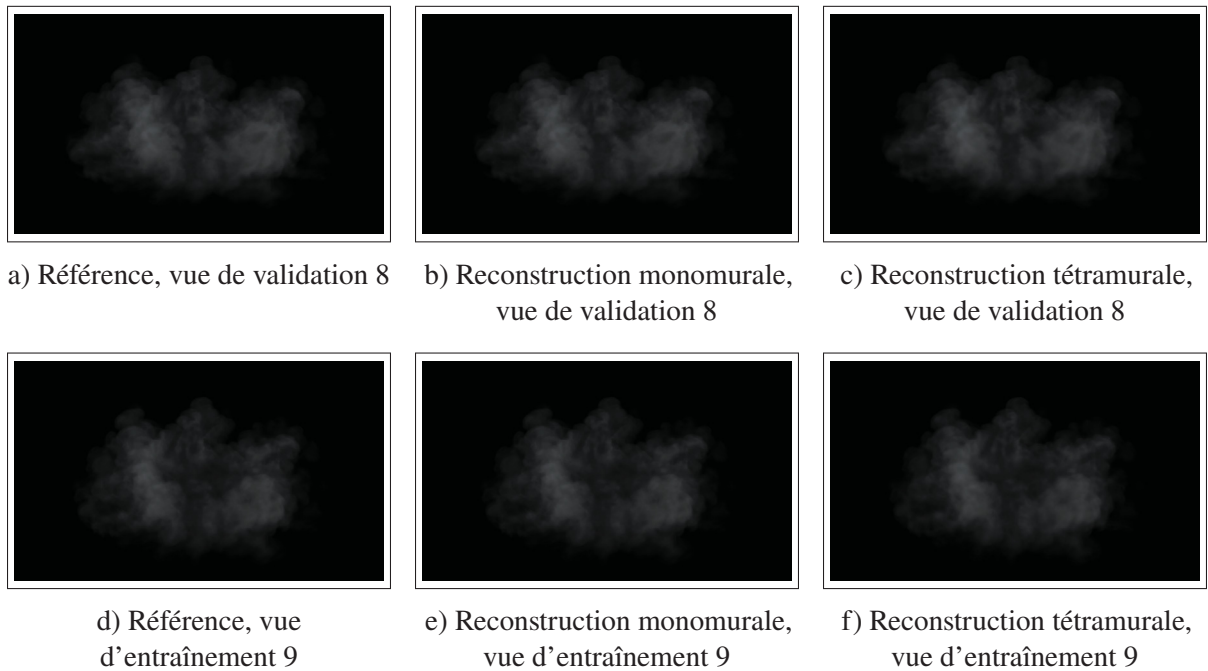


Figure 3.9 Reconstructions de *nuage de poussière* selon la disposition de caméra et la vue

reconstruire ledit volume. Conséquemment, il est improbable que nos erreurs soient causées par une sous-densification du nuage.

Or, la figure 3.10 montre aussi une erreur maximale d'au plus quatre centième. Lorsque convertie dans la plage colorimétrique standard codée sur entiers de huit bits par canal de couleur tel qu'utilisée par le format PNG, format dans lequel 3DGS exporte ses rendus, cette différence maximale correspond à dix unités sur 255. Suite à cette observation, nous avons investigué la répartition de l'erreur par canal de couleur entre la reconstruction et la référence selon la disposition de caméras. La figure 3.11 présente cette répartition de l'erreur par canal ; à la

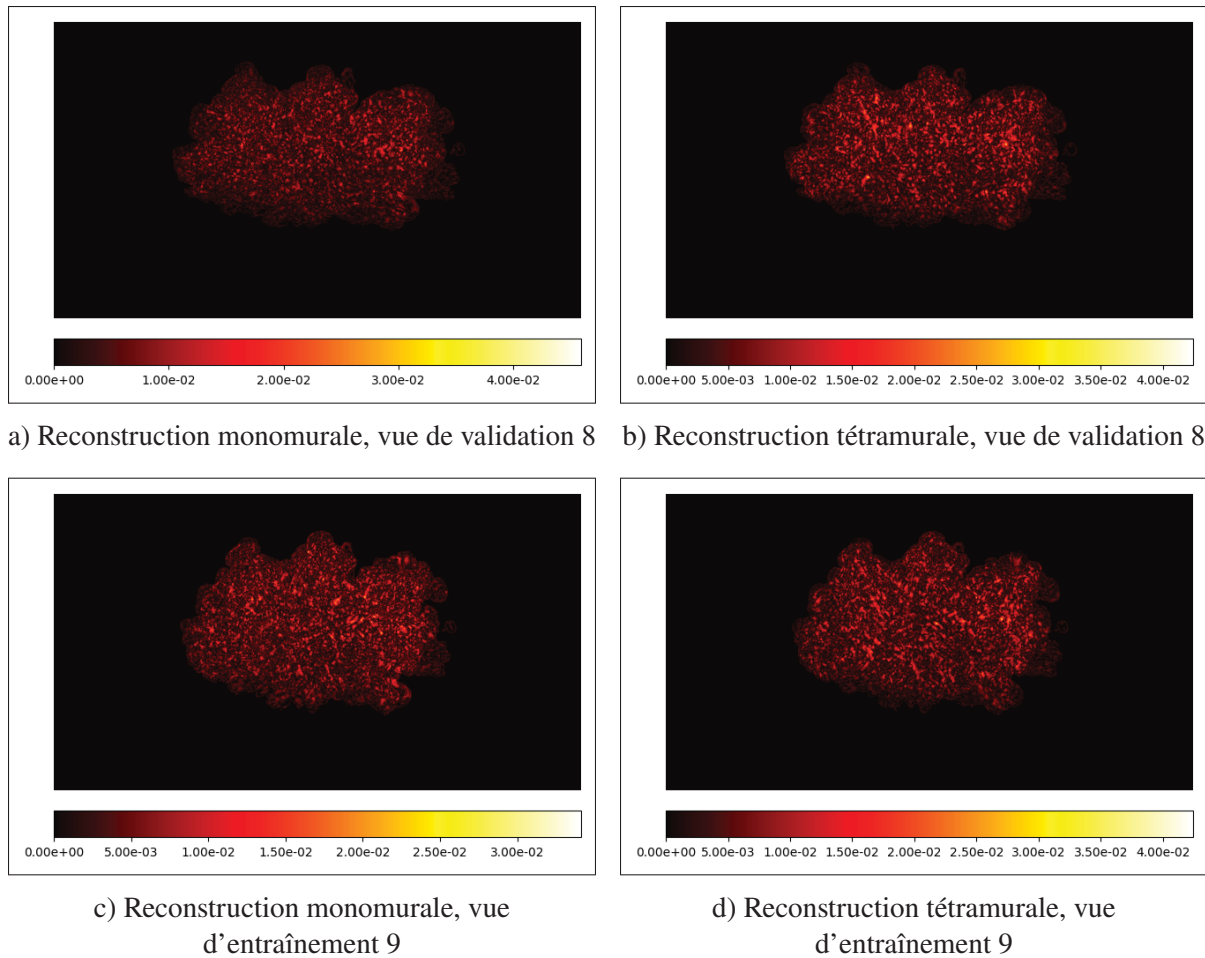
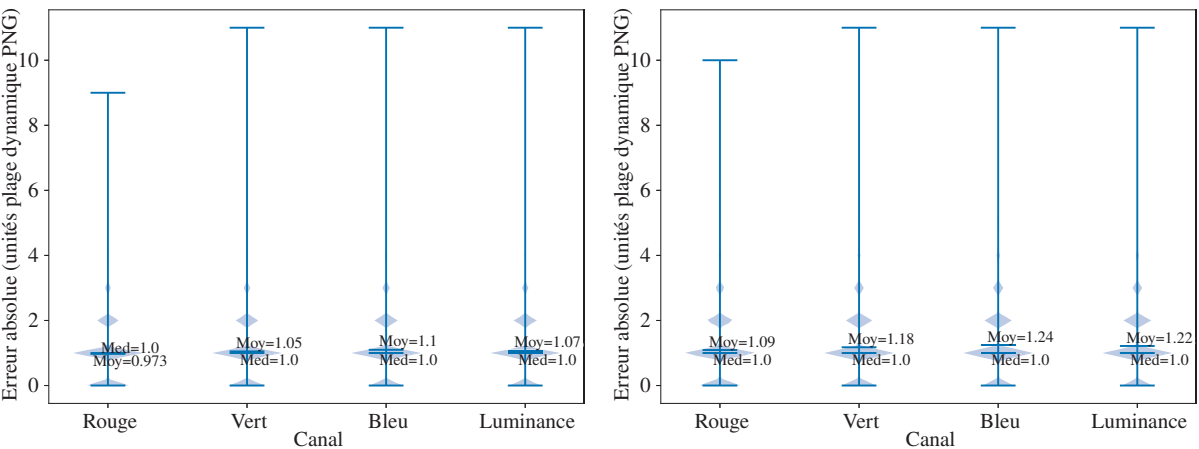


Figure 3.10 Différence de luminance des rendus de *nuage de poussière* par rapport à la référence selon la disposition de caméra et la vue

différence de la figure 3.10, l'erreur y est rapportée dans la plage dynamique du format PNG, c'est-à-dire en entiers à huit bits par canal de couleur. Nous dérivons de ces trois canaux la luminance (aussi appelé luma). Nous avons aussi retiré de ces graphiques les pixels parfaitement reconstruits (ceux dont la différence est nulle dans les trois canaux de couleurs), puisqu'ils composent le fond des images et écrasent les autres en poids parmi la distribution et cachent les tendances. La figure 3.11 montre que parmi les pixels présentant une erreur, les deux dispositions de caméras produisent majoritairement des pixels ayant une différence d'au plus trois unités, quoique la disposition monomurale montre des moyennes plus basses dans tous les canaux. En plus de la figure 3.11, les tableaux 3.10 et 3.11 montrent que la pluralité des erreurs sont de

l'ordre d'une seule unité dans tous les canaux. Ainsi, bien que l'erreur soit perceptible, elle est numériquement très faible, et se cache peut-être dans l'imprécision des nombres à virgule flottante, ainsi invisible de l'algorithme d'optimisation.



a) Reconstruction monomurale, vue de validation 8 b) Reconstruction tétramurale, vue de validation 8

Figure 3.11 Répartition de la différence par pixel et par canal des rendus de *nuage de poussière* et de la référence, en unités de plage dynamique PNG, selon la disposition de caméra et le canal de couleur

Tableau 3.10: Occurences d’erreur par canal en reconstruction monomurale de la scène *nuage de poussière* en vue de validation 8

Erreur	Rouge	Vert	Bleu	Luma
0	25634	22496	20202	19919
1	44371	45138	45836	46444
2	15692	16771	17507	18039
3	5337	6098	6512	6236
4	1584	1912	2213	1896
5	404	550	653	475
6	107	144	178	110
7	12	30	35	21
8	0	2	4	1
9	4	1	1	1
10	0	3	2	3
11	0	0	2	0

Tableau 3.11: Occurences d’erreur par canal en reconstruction tétramurale de la scène *nuage de poussière* en vue de validation 8

Erreur	Rouge	Vert	Bleu	Luma
0	25537	22484	20471	20509
1	41749	42675	43357	44043
2	12578	13808	14443	14588
3	3375	3951	4475	3958
4	737	983	1105	862
5	135	191	226	141
6	23	34	48	25
7	3	10	10	10
8	1	1	2	1
9	1	1	1	2
10	1	1	0	0
11	0	1	2	1

3.8 Composition de scène

Le présent test vise à mesurer l'utilisabilité des gaussiennes comme primitive de modélisation constructive, à l'instar des polygones. Or, la réutilisation de gaussiennes dans des environnements jamais vus à l'entraînement requiert que les gaussiennes soient rigoureusement disposées de manière physiquement cohérente, tel que révélé en section 3.1 et 3.2, où les gaussiennes flottantes compensatoires sont le principal obstacle au succès du transfert d'environnement. La composition de scènes soulève des problèmes potentiels additionnels, tels l'interpénétration et la suropacité des objets transparents. Nous pensons que les méthodes d'initialisation robustes et l'entraînement multifond proposées dans le présent mémoire offrent une solution à ces problèmes, ce qui permet de composer des scènes gaussiennes de manière visuellement convaincante.

Pour tester la contribution des méthodes proposées à la réussite de cette tâche, nous avons créé des versions composées de nos scènes *foyer*, *théière* et *nuage de poussière* dans *Blender* où le sujet est copié et transformé trois fois, de sorte que les caméras observent quatre sujets disposés dans leur scène. De plus, nous avons amélioré les capacité d'évaluation de *gsplat*, de sorte à lui permettre de charger plusieurs scènes reconstruites à la fois et de leur appliquer des transformations géométriques individuelles. Ainsi permis par ces modifications, le test de composition consiste à :

1. choisir une scène entraînée sur une référence monosujet ;
2. charger la scène choisie quatre fois dans *gsplat* ;
3. appliquer à chaque copie de la scène la translation, rotation et la mise à l'échelle relatives au sujet central extraites de la modélisation tétrasujet (par exemple, la théière 1 à une transformation nulle par rapport à elle-même, la théière 2 est translatée de cinq unités dans l'axe Y, la théière 3 subit une rotation de 180 degrés autour de l'axe Z et double de taille) ;
4. rendre les scènes tétrasujet ;
5. calculer leurs métriques par rapport aux références tétrasujet.

En tout, la présente expérience de composition de scène sera formée des données de test issues de l'exécution de la procédure sur les trois scènes susmentionnées avec objet de support, sur

Tableau 3.12: Qualité photométrique moyenne sur toutes les vues de validation de toutes les scènes en situation de composition de scène moyenne selon l’initialisation et les fonds d’entraînement

		LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Initialisation	Fond(s) d’entraînement				
Échantillonnage de champ de densité VDB par loi multinomiale	Noir	62,77 m	257,33	26,42	952,83 m
	Noir et blanc	65,10 m	250,57	26,59	952,53 m
100 k points à position et couleur aléatoires, visibilité minimale de 20	Noir	273,28 m	3,31 k	15,76	757,03 m
	Noir et blanc	67,77 m	239,19	26,79	954,10 m

monofond noir et bifond noir et blanc, et initialisées soit avec la méthode aléatoire à visibilité minimale de vingt, soit avec la méthode d’échantillonnage VDB multinomial. Le tableau 3.12 présente les résultats de cette expérience. Le tableau montre des résultats similaires à ceux présentés aux sections 3.1 et 3.2, où les fonds d’entraînement et l’initialisation influent tout deux de manière fortement positive sur la qualité des rendus en situation de composition de scènes.

L’examen des images abonde dans le même sens que les métriques, à savoir que l’entraînement bifond et l’initialisation robuste permettent de former des scènes intelligibles, sans artefacts flottants et où l’accumulation de la transparence des gaussiennes issues de scènes différentes reste plausible. Par contre, notre approche souffre d’une lacune en sa totale incapacité à comprendre le comportement de la lumière, ce qui se manifeste dans la scène *théière* quadruplée : le rendu de référence du bec et de la vapeur de la grosse théière produisent de l’ombre aux becs et à la vapeur des deux autres théières à proximité ; puisque cette situation ne peut pas être modélisé par notre approche, ces zones sont trop claires par rapport à la référence. La figure 3.12 montre les reconstructions composées de *théière*, disposées selon les fonds d’entraînement et les initialisations, tandis que 3.13 montre l’erreur carrée de ces mêmes reconstructions composées par rapport à la référence. Les figures 3.13b, 3.13c et 3.13d montrent la lacune observée ici, l’erreur y étant concentrée dans les zones où la lumière est composée incorrectement et où la reconstruction a échoué à converger. Par contre, dans les scènes où les modèles et leurs ombres



a) Monofond noir, initialisation à position et couleur aléatoires et visibilité minimale de 20



b) Monofond noir, échantillonnage multinomial du gradient de la densité



c) Fonds d'entraînement noir et blanc, initialisation à position et couleur aléatoires et visibilité minimale de 20



d) Bifond noir et blanc, échantillonnage multinomial du gradient de la densité



e) Référence

Figure 3.12 Reconstructions composées de *théière* selon la combinaison des méthodes d'entraînement

ne se chevauchent pas, le bénéfice net de notre approche ressort : alors que *Blender* produit un rendu par lancer de rayon de toutes nos scènes composées en près de onze secondes, *gsplat*

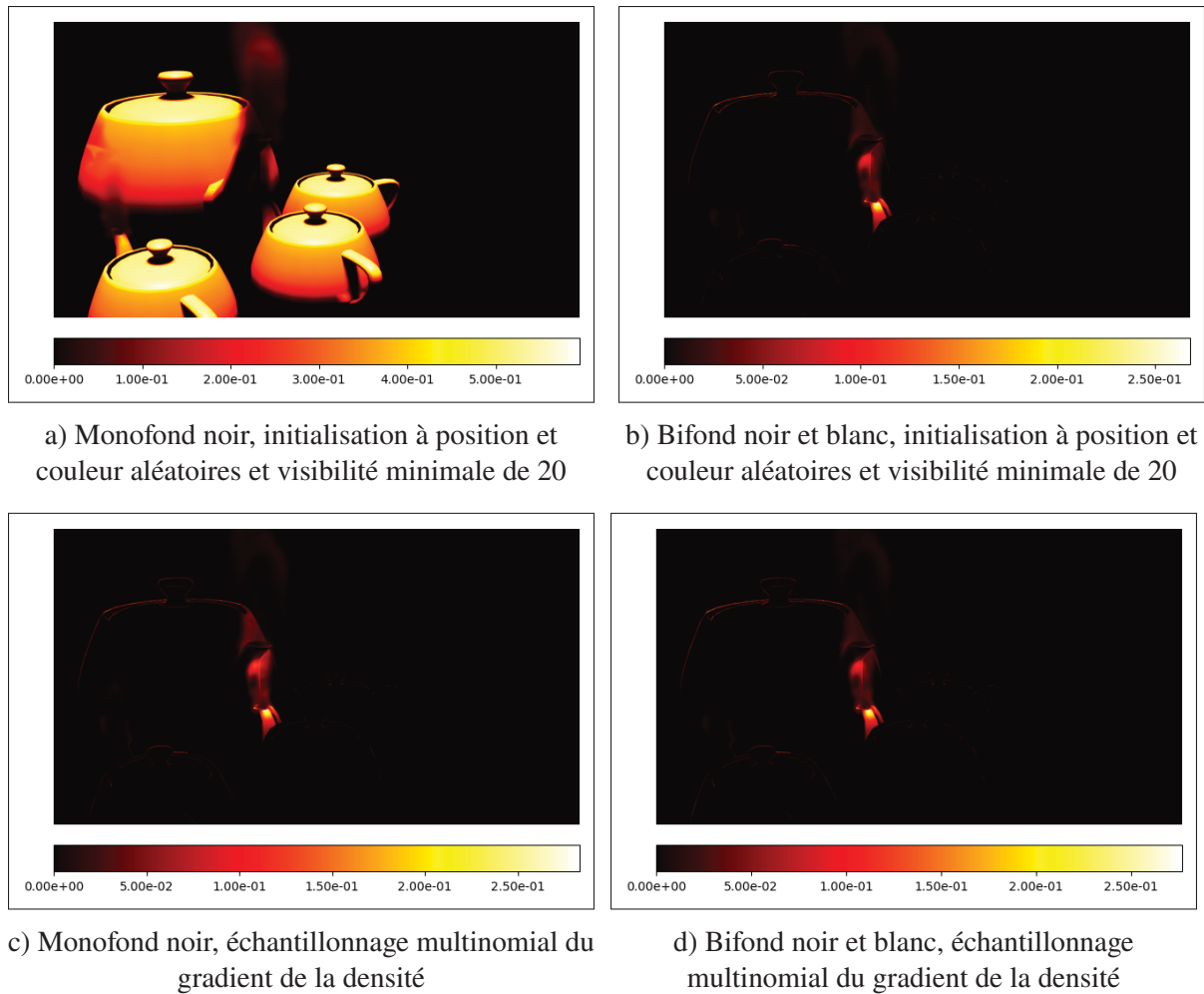


Figure 3.13 Erreur carrée des reconstructions composées de *théière* par rapport à la référence selon la combinaison des méthodes (voir figure 3.12)

produit une trame à temps pour chaque synchronisation verticale de l'écran, soit à raison d'un rendu à chaque soixantième de seconde.

Nous présentons à la figure 3.14 les modèles ayant le mieux généralisé, c'est-à-dire ceux reconstruits par l'entraînement bifond noir et blanc et initialisés par l'échantillonnage multinomial, en composition sur des cartes d'environnement et dans des environnements mélangeant des cubes modélisés polygonalement. Les rendus furent produits avec le logiciel *Blender* nommé *3DGS Render* (KIRI Innovations, 2026). La carte d'environnement et les cubes polygonaux

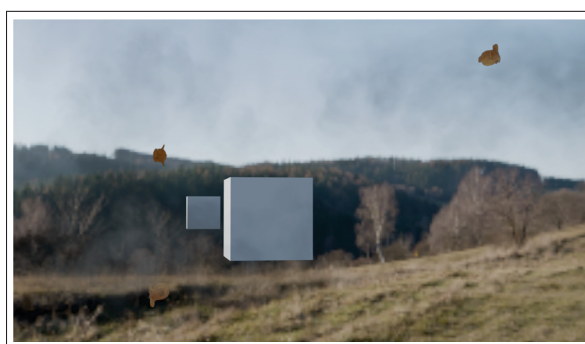
a) Scène *foyer*b) Scène *théière*c) Scène *nuage de poussière*

Figure 3.14 Rendus quadruples composés avec polygones et carte d'environnement

permettent de valider que les volumes et leur objet de support sont d'opacité plausible : les modèles sont opaques et les volumes sont transparents.

CONCLUSION ET RECOMMANDATIONS

Les travaux présentés dans le présent mémoire ont abouti à la création d'une série de méthodes visant à faciliter l'adoption de l'étalement gaussien dans des flux de création d'effets spéciaux volumiques transparents, telles la fumée et la vapeur. Ces méthodes s'articulent d'abord autour de l'intégration des outils d'étalement gaussien aux outils d'édition 3D par l'export d'environnement de simulation, de disposition de caméras et par l'échantillonnage de volumes, puis autour d'ajustements à l'optimisation 3DGS aidant spécifiquement à la reconstruction de scènes volumiques transparentes. Notre méthode d'export d'environnement permet de préserver le référentiel spatial d'une scène jusqu'à la fin de l'entraînement ; notre méthode d'initialisation aléatoire additionnée d'élagage basée sur la visibilité des points permet de capturer des scènes de vapeur avec très peu d'information géométrique, tandis que nos méthodes d'initialisation par échantillonnage du gradient de la densité et par conversion hiérarchique de cellules VDB permettent une initialisation robuste des scènes complexes ; notre méthode d'entraînement multifond permet d'augmenter la convergence des reconstructions en réduisant l'ambiguïté entre les sujets volumiques flous et leur environnement. De plus, la conversion d'une grille de simulation de volume à un nuage de gaussiennes ne compromet pas son éditabilité, sa réutilisabilité et sa composabilité, comme montré par nos résultats.

Nos résultats montrent que l'entraînement multifond est excessivement efficace par rapport à son coût en termes de gestion de données, de mémoire graphique et de temps de calcul, sans oublier qu'il permet de reconstruire avec succès les scènes difficiles initialisées naïvement sans ajout d'information géométrique sur le volume. Nous pensons que l'entraînement multifond devrait être activé par défaut, en particulier si les données d'entrées sont naïves ou si la reconstruction est destinée à être composée avec d'autres sujets dans de nouveaux environnements. Considérant les avantages de l'entraînement multifond, des travaux futurs devraient se pencher sur la possibilité de mettre à profit cette méthode pour la reconstruction de scènes réelles capturées : en utilisant

un environnement de capture à caméras calibrées et un fond vert, il serait possible d’incruster le sujet volumique sur de nouveaux fonds avant de procéder à l’optimisation 3DGS.

Nous arrivons à certaines des mêmes conclusions que Sharma & Schmalstieg (2025) : l’utilisation de volumes VDB comme à priori pour la formation d’un modèle gaussien améliore énormément la justesse géométrique des modèles d’initialisation. Or, nos méthodes, contrairement à celles de Sharma & Schmalstieg (2025), emploient une étape d’optimisation, et permettent de voir que la justesse géométrique de l’initialisation se propage et se maintient jusqu’au terme de l’optimisation. Cette robustesse est aussi maintenue lorsque le sujet de la scène est observé par des caméras très éparses, tel que relevé dans notre expérience de reconstruction par observations monomurales. Par contre, nos résultats sont contraires à ceux de Sharma & Schmalstieg (2025) en matière de niveaux de détails, puisque les conversions hiérarchiques à niveaux de détails plus élevés n’améliorent pas la qualité des reconstructions, lorsqu’utilisées avec nos méthodes.

Par ailleurs, nos résultats sont aussi contraires à ceux de Condor *et al.* (2025), selon qui le noyau quadratique performe moins bien que le noyau gaussien lors du rendu de volumes par lancer de rayon. Nos résultats indiquent plutôt que les deux noyaux performent de façon très similaire dans un pipeline de reconstruction par étalage gaussien. Il n’est pas exclu que des travaux postérieurs montrent un avantage à utiliser le noyau quadratique pour la rastérisation de volumes dans des conditions et contraintes différentes des nôtres.

De plus, contrairement aux conclusions de Condor *et al.* (2025), nous n’avons pas observé de différence significative de vitesse de convergence en fonction de la primitive chargée par le moteur 3DGS. Plus précisément, notre méthode d’échantillonnage unitaire du gradient de la densité, qui émet des points, performe dans le même ordre de grandeur que toutes les variantes de la méthode de conversion hiérarchique, qui émettent des gaussiennes de tailles variables et légèrement anisotropes. Conséquemment, nous pensons que le positionnement des points dans le volume est le facteur dominant de la qualité de la reconstruction, et que les conclusions

de Condor *et al.* (2025) ne s’appliquent pas en cas de reconstruction de volume par étalage gaussien.

Néanmoins, certains de nos efforts de recherche sont sans fruit : les régularisations d’élongation et de position ne permettent pas d’améliorer les modèles de fumée reconstruits par étalage gaussien. Or, l’inefficacité de notre régularisation d’élongation contredit les résultats des travaux antérieurs de Xie *et al.* (2024), pour qui la régularisation d’anisotropie est une composante essentielle de leur méthode d’animation. Nous ne pouvons qu’hypothétiser que les défauts des gaussiennes anisotropes ne se manifestent pas dans le cas statique étudié dans le présent mémoire. Quant à la régularisation de position, elle pourrait être implémentée plus robustement par des travaux futurs pour remplir son objectif de réduction des artefacts flottants. En effet, il serait possible d’utiliser la robustesse de l’initialisation pour calculer un terme de distance de Chamfer entre la position des gaussiennes et les points d’initialisation que l’optimisation chercherait à minimiser.

De plus, notre méthode ne permet pas de reconstruire précisément les petites volutes de vapeur locales, ce qui crée des reconstructions gaussiennes plus floues pour l’œil humain que la référence, et ce, même lorsque nous permettons au modèle de se surentraîner à rendre le sujet sur 90° autour de l’axe vertical. Nos résultats de la section 3.7 présentent pourtant qu’une majorité des pixels des images sont parfaitement reconstruits, et que les autres pixels ne présentent qu’au plus quatre pour cent d’erreur. Nous pensons que l’accumulation de couleur des pixels est limitée par la plage dynamique de huit bits utilisée pour charger les références et produire les rendus. Des travaux futurs pourraient explorer l’ajout de la capacité de charger des images à haute plage dynamique, comme celles au format EXR, à 3DGS. Parallèlement à cet ajout, l’ensemble de la boucle d’entraînement de 3DGS, incluant les noyaux CUDA de rendu et de dérivation manuelle, devrait être modifié pour qu’il ne compresse en aucun temps les couleurs. Une autre voie d’amélioration parallèle pourrait être le remplacement de l’encodage de la couleur

rouge-vert-bleu actuel par un encodage luminance-chrominance, qui prendrait mieux en compte la sensibilité accrue du système visuel humain vis-à-vis des variations d'intensité lumineuse que de celles de couleur.

Finalement, dans un contexte de production d'effets spéciaux, nous recommandons d'utiliser à tout le moins une combinaison de nos méthodes d'export de référentiel et d'entraînement multifond. Cette première facilite énormément le flux de création itératif en permettant l'intelligibilité géométrique de la méthode de bout en bout, tandis que la seconde améliore considérablement la qualité des reconstructions sans coût additionnel de temps de simulation ou de préparation des données. Additionnellement, parmi nos méthodes d'initialisation, nous recommandons l'utilisation de l'échantillonnage de champ de densité VDB par loi multinomiale ou la conversion hiérarchique VDB à bas détail. Ces méthodes se sont démarquées pour la qualité de leurs résultats et leur optimisation jusqu'à terme, même sur les scènes difficiles. Par contre, ces deux méthodes limitent la taille des grilles manipulables par leur implémentation : la première est limitée à des grilles de 2^{24} voxels actifs ; la seconde décompresse la grille VDB pour former sa représentation gaussienne, qui peut alors être trop grosse pour être chargée en mémoire graphique. Dans une telle situation, nous recommandons de sous-échantillonner la grille VDB avant de réessayer de la convertir par l'une ou l'autre des méthodes. L'utilisation de l'initialisation aléatoire, idéalement avec élagage de vue, pourrait être utilisée en dernier recours si l'export de la grille est impossible, mais si et seulement si l'entraînement multifond est utilisé en même temps.

ANNEXE I

CONVERSION DU RÉFÉRENTIEL *BLENDER* AU RÉFÉRENTIEL *COLMAP* PAR *PHOTOGRAMMETRY IMPORTER*

Tel qu'évoqué en section 2.2, nous utilisons le plugiciel *Photogrammetry Importer* (Bullinger *et al.*, 2021) s'intégrant à *Blender* pour exporter nos scènes, incluant caméras et objets polygonaux, au format *COLMAP*. Pour ce faire, le plugiciel crée le dossier de travail *COLMAP* et ses trois fichiers constitutants : `cameras.txt`, `images.txt` et `points3D.txt`.

Chaque caméra devient une entrée dans les deux premiers fichiers : `cameras.txt` contiendra les paramètres intrinsèques des caméras et `images.txt` contiendra les paramètres extrinsèques des caméras. Nos images sont rendues selon un modèle de caméra sténopé idéale, seul modèle supporté par toutes les implémentations de 3DGS et par *Photogrammetry Importer*. Dans ce modèle, les seuls paramètres intrinsèques sont les dimensions et le centre optique de l'image, triviaux à récupérer de *Blender*. Pour les paramètres extrinsèques, c'est la rotation et la position de la caméra qui doivent être ajoutés au fichier. La rotation de la caméra doit être convertie du référentiel-monde au référentiel-image en effectuant une rotation de -90° autour de l'axe X, puis être convertie en quaternion avant d'être stockée. La position est, quant à elle, stockée telle quelle. Le fichier `points3D.txt` est peuplé avec les sommets de toutes les géométries sélectionnées au moment de produire la reconstruction *COLMAP*.

Par contre, cette implémentation présente des lacunes. D'une part, le format de données *COLMAP* textuel utilisé par *Photogrammetry Importer* présente un défaut dans l'encodage des nombres à virgule flottante, ce qui peut corrompre les extrinsèques de certaines caméras exportées. Pour contourner ce problème, nous élaguons manuellement les caméras corrompues en retirant les lignes pertinentes des fichiers, identifiées en réimportant dans *Blender* la reconstruction tout juste exportée.

ANNEXE II

ÉCHANTILLONNEURS DE GRADIENT DE CHAMP DE DENSITÉ VDB

La section 2.3.2 a présenté le squelette de l'algorithme d'échantillonnage de VDB par le gradient de sa densité. Ladite section présente trois variantes de l'algorithme d'échantillonnage, qui sont implémentées comme des fonctions invocables polymorphiquement. Les trois variantes furent développées pour tester des aspects différents de l'échantillonnage. Les variantes seront expliquées ci-dessous et seront aussi détaillées à l'algorithme II-1.

L'échantillonnage multinomial permet autant de sous- que de suréchantillonner la grille VDB. Par contre, son implémentation est limitée à des grilles de 2^{24} cellules. Un tirage multinomial est une expérience aléatoire avec remplacement où le résultat peut être l'une des valeurs passées en paramètre, avec une certaine densité de probabilité. Dans notre cas, les valeurs possibles sont les indices de la grille et la densité de probabilité relative est la norme du gradient de la densité en la cellule. Nous utilisons, dans nos tests du chapitre 3, cent-mille échantillons.

L'interpolation exponentielle suréchantillonne agressivement la grille. Cette variante trouve la densité minimale et maximale de la grille, puis alloue pour chaque cellule de la grille un nombre d'échantillons en fonction de la distance séparant la densité de la cellule des extremums. La distance est portée à une puissance fractionnelle pour amplifier le nombre d'échantillons dès que la densité s'éloigne du minimum. Notre configuration utilise un et huit comme nombre d'échantillons minimal et maximal, et un tiers comme puissance.

L'échantillonnage unitaire alloue un seul échantillon à tous les voxels actifs. Cette variante permet de retirer une variable de l'expérience, c'est-à-dire la technique d'échantillonnage, pour n'étudier que l'influence du placement de points robuste.

Algorithme-A II-1 Échantillonneur de gradient de densité

Entrées : Type d'échantillonneur s ;

Paramètres de l'algorithme choisi P_s ;

Gradient de la densité en chaque cellule $\mathbf{g}$

Sorties : Indices des cellules échantillonnées $\mathbf{m} \in \{x \in \mathbb{N} | 1 \leq x \leq \#\mathbf{g}\}_m$

```

1  suivant  $s$  faire
2      cas où multinomial faire
3           $n \leftarrow P_s$ ;                                /* nombre d'échantillons */
4           $\mathbf{i} \leftarrow [1 \ 2 \ \dots \ \#\mathbf{g}]$ ;          /* indices des voxels */
5          retourner multinomial( $\mathbf{i}, \mathbf{g}, n$ ); /*  $n$  tirages indépendants parmi  $\mathbf{i}$ 
           avec les poids relatifs  $\mathbf{g}$  */
6      fin cas
7      cas où interpolation exponentielle faire
8           $u, v, \gamma \leftarrow P_s$ ; /* #minimum et #maximum d'échantillons par voxel,
           puissance */
9           $\alpha \leftarrow \left( \frac{\mathbf{g} - \min(\mathbf{g})}{\max(\mathbf{g}) - \min(\mathbf{g})} \right)^\gamma$ ; /* distance exponentielle de chaque  $\mathbf{g}_i$  par
           rapport aux extremums de toute la grille */
10          $\mathbf{n} \leftarrow \lfloor (1 - \alpha)u + \alpha v \rfloor$ ; /* nombre d'échantillons en chaque cellule
           */
11         retourner  $[ [1]_{n_1} \ [2]_{n_2} \ \dots \ [\#\mathbf{g}]_{n_{\#\mathbf{g}}} ]$ ; /* répète chaque index  $\mathbf{n}_i$ 
           fois */
12     fin cas
13     cas où unitaire faire
14         retourner  $[1 \ 2 \ \dots \ \#\mathbf{g}]$ ; /* chaque cellule une fois */
15     fin cas
16 fin d'alternative

```

ANNEXE III

CONVERSION HIÉRARCHIQUE DE GRILLE DE DENSITÉ VDB

La sous-section 2.3.3 a présenté dans les grandes lignes la méthode de conversion VDB adaptée des travaux de Sharma & Schmalstieg (2025). Or, les niveaux de détails ont été décrits, mais n'ont pas été présentés dans le détail. Les algorithmes III-1, III-2 et III-3 présente en détail l'itération sur les voxels actifs de chaque nœud-feuille pour calculer la taille et le nombre de gaussienne à émettre pour chacun des nœuds-feuilles

Algorithme-A III-1 Conversion hiérarchique à basse résolution de grille en boîtes

```
1 Fonction grouperBoîtesEnglobantes( $D \in \mathbb{R}^{x \times y \times z}$ ) début
2    $B_{min}, B_{max} \leftarrow \mathbb{R}^{3 \times 0}; \mathbf{d} \in \mathbb{R}^0;$ 
3   pour  $F \in cellulesFeuilles(D)$  faire
4      $V \leftarrow coordVoxelsActifs(F);$  /* les nœuds actifs ont au moins un
       voxel actif, donc aucune boîte n'a de volume nul */
5      $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(V, B_{min}, B_{max}, \mathbf{d});$ 
6   fin pour
7   retourner  $B_{min}, B_{max}, \mathbf{d};$ 
8 fin
```

Algorithme-A III-2 Conversion hiérarchique à résolution moyenne de grille en boîtes

```

1 Fonction grouperBlocsDenses4Ou2( $D \in \mathbb{R}^{x \times y \times z}$ ) début
2    $B_{min}, B_{max} \in \mathbb{R}^{3 \times 0}; \mathbf{d} \in \mathbb{R}^0;$ 
3   pour  $F \in cellulesFeuillesDenses(D)$  faire
4     pour  $S \in sousBlocs_{4 \times 4 \times 4}(F)$  faire
5        $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(S, B_{min}, B_{max}, \mathbf{d});$ 
6     fin pour
7   fin pour
8   pour  $F \in cellulesFeuillesÉpares(D)$  faire
9     pour  $S \in sousBlocsDenses_{2 \times 2 \times 2}(L)$  faire
10       $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(S, B_{min}, B_{max}, \mathbf{d});$ 
11    fin pour
12    pour  $\mathbf{v} \in voxelsActifs(sousBlocsÉpares_{2 \times 2 \times 2}(L))$  faire
13       $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(\mathbf{v}, B_{min}, B_{max}, \mathbf{d});$ 
14    fin pour
15  fin pour
16  retourner  $B_{min}, B_{max}, \mathbf{d};$ 
17 fin

```

Algorithme-A III-3 Conversion hiérarchique à haute résolution de grille en boîtes

```

1 Fonction grouperBlocsDenses2( $D \in \mathbb{R}^{x \times y \times z}$ ) début
2    $B_{min}, B_{max} \in \mathbb{R}^{3 \times 0}; \mathbf{d} \in \mathbb{R}^0;$ 
3   pour  $F \in cellulesFeuillesDenses(D)$  faire
4     pour  $S \in sousBlocs_{2 \times 2 \times 2}(F)$  faire
5        $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(S, B_{min}, B_{max}, \mathbf{d});$ 
6     fin pour
7   fin pour
8   pour  $F \in cellulesFeuillesÉpares(D)$  faire
9     pour  $\mathbf{v} \in voxelsActifs(F)$  faire
10       $B_{min}, B_{max}, \mathbf{d} \leftarrow ajouteBoîte(\mathbf{v}, B_{min}, B_{max}, \mathbf{d});$ 
11    fin pour
12  fin pour
13  retourner  $B_{min}, B_{max}, \mathbf{d};$ 
14 fin

```

ANNEXE IV

COMPARAISON NUMÉRIQUE DES NOYAUX D'ÉTALAGE

La section 3.6 a présenté un comparatif du noyau gaussien et de notre implémentation du noyau quadratique. Or, le tableau 3.7 a montré que le noyau gaussien performe plus souvent mieux que le noyau quadratique sur nos scènes, avec et sans objet de support. Le tableau IV-1 présente les résultats numérique ayant permis de produire le tableau 3.7 et de tirer ces conclusions.

Tableau-A IV-1: Qualité photométrique selon la scène, la présence d'objet de support et le noyau d'étalage

Scène	Présence Noyau d'objet de support		LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
Nuage de poussière	Non	Quadr.	0,027 794 2	6,872 58	40,085 8	0,976 115
		Gauss.	0,032 172 4	7,434 76	39,861 9	0,978 790
	Oui	Quadr.	0,020 078 7	2,899 53	43,916 7	0,988 034
		Gauss.	0,025 356 5	2,525 82	44,349 9	0,989 485
Foyer	Non	Quadr.	0,002 432 57	0,910 472	48,726 9	0,998 311
		Gauss.	0,002 285 17	0,762 589	49,421 1	0,998 623
	Oui	Quadr.	0,002 602 24	1,285 10	47,119 3	0,997 718
		Gauss.	0,002 514 71	1,069 70	47,862 9	0,997 994
Théière	Non	Quadr.	0,000 783 925	0,128 705	57,176 1	0,999 214
		Gauss.	0,000 397 833	0,082 745 7	59,334 3	0,999 628
	Oui	Quadr.	0,001 066 30	1,312 17	47,357 1	0,998 803
		Gauss.	0,000 805 277	1,053 89	48,196 7	0,999 016
Nuage atmosphérique	Non	Quadr.	0,222 012	476,684	21,397 7	0,871 571
		Gauss.	0,229 973	443,087	21,701 0	0,873 092

↓,↑: Sens d'optimisation de la métrique

Quadr.: Quadratique

Gauss.: Gaussien

LISTE DE RÉFÉRENCES

- Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhersch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P. & Chintala, S. (2024, apr). PyTorch 2 : Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. doi : 10.1145/3620665.3640366.
- Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P. & Hedman, P. (2022). Mip-NeRF 360 : Unbounded Anti-Aliased Neural Radiance Fields. *CVPR*.
- Blender Online Community. (2026). Blender - a 3D modelling and rendering package. Stichting Blender Foundation, Amsterdam : Blender Foundation. Repéré à <http://www.blender.org>.
- Bullinger, S., Bodensteiner, C. & Arens, M. (2021). A Photogrammetry-based Framework to Facilitate Image-based Modeling and Automatic Camera Tracking. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - GRAPP*, pp. 106-112. doi : 10.5220/0010319801060112.
- Celarek, A., Kopanas, G., Drettakis, G., Wimmer, M. & Kerbl, B. (2025). Does 3D Gaussian Splatting Need Accurate Volumetric Rendering? *Computer Graphics Forum*, 44(2), e70032. doi : <https://doi.org/10.1111/cgf.70032>.
- Chen, H., Chen, R., Qu, Q., Wang, Z., Liu, T., Chen, X. & Chung, Y. Y. (2024). Beyond Gaussians : Fast and High-Fidelity 3D Splatting with Linear Kernels. Repéré à <https://arxiv.org/abs/2411.12440>.
- Condor, J., Speierer, S., Bode, L., Bozic, A., Green, S., Didyk, P. & Jarabo, A. (2025). Don't Splat your Gaussians : Volumetric Ray-Traced Primitives for Modeling and Rendering Scattering and Emissive Media. *ACM Transactions on Graphics*, 44(1), 1–17. doi : 10.1145/3711853.
- Hedman, P., Philip, J., Price, T., Frahm, J.-M., Drettakis, G. & Brostow, G. (2018). Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6). doi : 10.1145/3272127.3275084.

- Hull, M., Yang, H., Mehta, P., Phute, M., Cho, A., Wang, H., Lau, M., Lee, W., Lunardi, W., Andreoni, M. & Chau, D. H. (2025). *3D Gaussian Splat Vulnerabilities*. CVPR Workshop on Neural Fields Beyond Conventional Cameras. Repéré à <https://arxiv.org/pdf/2506.00280>.
- Ivanic, J. & Ruedenberg, K. (1996). Rotation Matrices for Real Spherical Harmonics. Direct Determination by Recursion. *The Journal of Physical Chemistry*, 100(15), 6342–6347. doi : 10.1021/jp953350u.
- Jakob, W., Regg, C. & Jarosz, W. (2011). Progressive Expectation-Maximization for Hierarchical Volumetric Photon Mapping. *Computer Graphics Forum*, 30(4), 1287-1297. doi : <https://doi.org/10.1111/j.1467-8659.2011.01988.x>.
- JangaFX. (2020). Real-time VFX Simulations. Repéré le 2026-04-06 à <https://jangafx.com/software/embergen/download/free-vdb-animations>.
- Keetha, N., Karhade, J., Jatavallabhula, K. M., Yang, G., Scherer, S., Ramanan, D. & Luiten, J. (2024). SplatAM : Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Kerbl, B., Kopanas, G., Leimkuehler, T. & Drettakis, G. (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 1–14. doi : 10.1145/3592433.
- Kheradmand, S., Rebain, D., Sharma, G., Sun, W., Tseng, Y.-C., Isack, H., Kar, A., Tagliasacchi, A. & Yi, K. M. (2024). 3D Gaussian Splatting as Markov Chain Monte Carlo. *Advances in Neural Information Processing Systems (NeurIPS)*.
- KIRI Innovations. [original-date : 2024-09-27T06 :50 :41Z]. (2026, mar). Kiri-Innovation/3dgs-render-blender-addon. KIRI Innovations. Repéré le 2026-03-10 à <https://github.com/Kiri-Innovation/3dgs-render-blender-addon>.
- Knapitsch, A., Park, J., Zhou, Q.-Y. & Koltun, V. (2017). Tanks and Temples : Benchmarking Large-Scale Scene Reconstruction. *ACM Transactions on Graphics*, 36(4).
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. & Ng, R. (2021). NeRF : representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106. doi : 10.1145/3503250.
- Museth, K. (2013). VDB : High-resolution sparse volumes with dynamic topology. *ACM Transaction on Graphics*, 32(3). doi : 10.1145/2487228.2487235.

- Pinchon, D. & Hoggan, P. E. (2007). Rotation matrices for real spherical harmonics : general rotations of atomic orbitals in space-fixed axes. *Journal of Physics A : Mathematical and Theoretical*, 40(7), 1597–1610. doi : 10.1088/1751-8113/40/7/011.
- Radl, L., Steiner, M., Parger, M., Weinrauch, A., Kerbl, B. & Steinberger, M. (2024). StopThePop : Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics*, 4(43).
- Schönberger, J. L. & Frahm, J.-M. (2016). Structure-from-Motion Revisited. *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Schönberger, J. L., Price, T., Sattler, T., Frahm, J.-M. & Pollefeys, M. (2016a). A Vote-and-Verify Strategy for Fast Spatial Verification in Image Retrieval. *Asian Conference on Computer Vision (ACCV)*.
- Schönberger, J. L., Zheng, E., Pollefeys, M. & Frahm, J.-M. (2016b). Pixelwise View Selection for Unstructured Multi-View Stereo. *European Conference on Computer Vision (ECCV)*.
- Sharma, I. & Schmalstieg, D. [Version Number : 2]. (2025). 3D Gaussian Particle Approximation of VDB Datasets : A Study for Scientific Visualization. arXiv. Repéré le 2025-09-20 à <https://arxiv.org/abs/2504.04857>.
- Walt Disney Animation Studios. (2017). Clouds Data Set. Repéré le 2025-11-05 à <https://disneyanimation.com/resources/clouds/>.
- Wang, K., Yang, C., Wang, Y., Li, S., Wang, Y., Dou, Q., Yang, X. & Shen, W. (2024). EndoGSLAM : Real-Time Dense Reconstruction and Tracking in Endoscopic Surgeries using Gaussian Splatting. *arXiv preprint arXiv :2403.15124*.
- Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. (2004). Image quality assessment : from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612. doi : 10.1109/TIP.2003.819861.
- Williams, F., Huang, J., Swartz, J., Klar, G., Thakkar, V., Cong, M., Ren, X., Li, R., Fuji-Tsang, C., Fidler, S., Sifakis, E. & Museth, K. (2024). fVDB : A Deep-Learning Framework for Sparse, Large Scale, and High Performance Spatial Intelligence. *ACM Transactions on Graphics*, 43(4). doi : 10.1145/3658226.
- Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y. & Jiang, C. (2024). PhysGaussian : Physics-Integrated 3D Gaussians for Generative Dynamics. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389-4398. doi : 10.1109/CVPR52733.2024.00420.

- Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M. & Kanazawa, A. (2025). gsplat : An open-source library for Gaussian splatting. *Journal of Machine Learning Research*, 26(34), 1–17.
- Ye, Z., Li, W., Liu, S., Qiao, P. & Dou, Y. (2024). AbsGS : Recovering Fine Details in 3D Gaussian Splatting. *Proceedings of the 32nd ACM International Conference on Multimedia*, (MM '24), 1053–1061. doi : 10.1145/3664647.3681361.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. (2018, June). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zwicker, M., Pfister, H., van Baar, J. & Gross, M. (2001). Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, (SIGGRAPH '01), 371–378. doi : 10.1145/383259.383300.