



**HAL**  
open science

# Débruitage d'images de rendues par méthodes de Monte Carlo spectrales

Mathieu Noizet, Robin Roupael, Stéphanie Prévost, Hervé Deleau, Luiz Angelo Steffenel, Laurent Lucas

► **To cite this version:**

Mathieu Noizet, Robin Roupael, Stéphanie Prévost, Hervé Deleau, Luiz Angelo Steffenel, et al.. Débruitage d'images de rendues par méthodes de Monte Carlo spectrales. Journées Françaises d'Informatique Graphique (JFIG 2022), Nov 2022, Bordeaux, France. hal-04676433

**HAL Id: hal-04676433**

**<https://hal.science/hal-04676433v1>**

Submitted on 3 Oct 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Débruitage d'images de rendues par méthodes de Monte Carlo spectrales

M. Noizet<sup>1</sup>, R. Rouphael<sup>1,2</sup>, S. Prévost<sup>1</sup>, H. Deleau<sup>1</sup>, L-A. Steffemel<sup>1</sup> and L.Lucas<sup>†1</sup>

<sup>1</sup>Université de Reims Champagne-Ardenne, LICIS, France

<sup>2</sup>United Visual Researchers, France



**Figure 1:** Image de référence (262k spp) et sa version bruitée (1k spp). Résultats obtenus par le DEMC et nos contributions : TEMC-1 et 2.

## Résumé

Le débruitage - ou denoising - d'images rendues par méthodes de Monte Carlo est un thème de recherche actif qui fait l'objet depuis des décennies de nombreuses études. Nous présentons dans cet article une nouvelle approche par apprentissage profond spécifiquement adaptée au rendu spectral. Dans ce cadre et à l'inverse des approches standard de denoising, qui trop souvent génèrent sur ce type d'images des aberrations chromatiques, nous proposons une méthode permettant une meilleure adaptation au bruit en séparant le traitement de la chrominance et de la luminance. Organisée autour de deux architectures, TEMC-1 assure tout d'abord le traitement distinct de ces deux composantes en se basant sur deux denoisers appropriés. TEMC-2 ensuite fusionne les deux denoisers précédemment évoqués afin d'améliorer les échanges d'informations et les temps d'apprentissage. Nos premiers résultats démontrent que notre approche sur la séparation des canaux offre une robustesse supérieure aux approches conventionnelles tout en délivrant une qualité de reconstruction proche de l'image référence.

## CCS Concepts

•Computing methodologies → Ray tracing; Neural networks; Image processing;

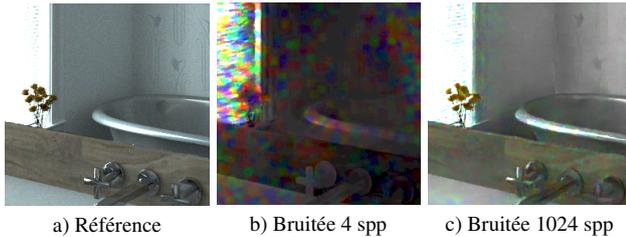
## 1. Introduction

Si la simulation optique prédictive ambitionne de résoudre l'équation de transport de la lumière [Kaj86], sa résolution reste encore aujourd'hui un problème complexe, difficilement solvable de manière analytique lorsqu'il s'agit notamment de prendre en charge des phénomènes physiques comme la polarisation, la photoélasticité, les interférences par couches minces et tous autres

phénomènes de luminescence ou de dispersion de la lumière. Ainsi, pour être en capacité de simuler de manière précise et prédictive le transport de la lumière, l'usage d'algorithmes de rendu spectral de type Monte Carlo est nécessaire. Cette classe d'algorithmes stochastiques d'illumination globale permet d'obtenir des images d'une grande fidélité à l'inverse des approches plus conventionnelles basées sur le modèle RGB.

Elle soulève cependant de nombreux défis, qui visent à réduire significativement les temps de calcul nécessaires à la génération d'une image, ce qui revient à réduire conjointement l'espace des positions et celui des valeurs. Par positions, il faut comprendre ici la question de l'échantillonnage qui peut être adapté de manière

<sup>†</sup> Cette recherche est soutenue par l'ANR-PRCE (LUCE ANR-21-CE33-0010). Les auteurs tiennent à remercier les plateformes Centre Image et ROMEO de l'URCA pour leur contribution matérielle en ressources de calcul à haute performance.



**Figure 2:** Aberrations chromatiques issues du denoiser d'OptiX

20 préférentielle [VHH\*19], et par valeurs, la question de la quantifi-  
21 cation des différentes longueurs d'ondes.

22 L'approche que nous proposons dans cet article, relève du  
23 débruitage. Si ce champ de recherche fait aujourd'hui l'objet de  
24 très nombreux travaux, disponibles les cas échéants sur « étagères »  
25 [NVi17, Int21], ceux traitant d'images rendues par méthodes de  
26 Monte Carlo spectrales restent beaucoup plus confidentiels. Une  
27 des raisons à cela tient dans la nature du bruit qui génère  
28 d'importantes aberrations chromatiques (cf. figure 2) lorsque celui-  
29 ci n'est pas correctement géré.

30 Bien que majoritairement utilisées avec l'espace RGB, les méth-  
31 odes de denoising basées *Deep Learning* ont supplanté les méth-  
32 odes traditionnelles. Elles restent cependant très imparfaites sur  
33 des entrées spectrales essentiellement pour des raisons de com-  
34 plexité de mise en œuvre. Aussi, afin de nous permettre de valider  
35 nos hypothèses, nos données d'entrée calculées par rendu spec-  
36 tral (16 bandes spectrales) ont été ré-encodées dans un espace col-  
37 orimétrique de plus faible dimension, l'espace sRGB en définition  
38 HDR Dolby Vision. Cette adaptation nous permet de nous focaliser  
39 sur le traitement i) des aberrations chromatiques issues du bruit et  
40 ii) de l'équilibre entre chrominance et luminance qui, dans le cas  
41 des images obtenues par rendu spectral, est très largement sous-  
42 estimé.

43 **Contribution.** Fort de ce constat, nous posons l'hypothèse que  
44 la séparation des canaux codant luminance et chrominance per-  
45 met de les considérer avec la même importance pour obtenir un  
46 rendu spectral sans bruit. Pour la vérifier, nous proposons une  
47 nouvelle approche appelée TEMC adaptée du DEMC [YWH\*19]  
48 pour le bruit chromatique. Nous la déclinons sous forme de  
49 deux architectures. La première, TEMC-1, se compose de deux  
50 réseaux distincts travaillant en parallèle sans interaction. La se-  
51 conde, TEMC-2, regroupe les deux précédents réseaux en une struc-  
52 ture unique pour leur permettre i) d'interagir et ii) d'accélérer les  
53 temps d'apprentissage.

54 Après un rapide état de l'art en section 2 sur les méthodes de  
55 denoising appliquées au rendu de Monte Carlo, nous présenterons  
56 en section 3 nos deux architectures et les comparerons au DEMC  
57 en section 4 avant de conclure et de fournir quelques perspectives.

## 58 2. Travaux antérieurs

59 La réduction du bruit de Monte Carlo dans des images RGB donne  
60 régulièrement lieu à publications. En 2015, Zwicker et al. [ZJL\*15]  
61 proposent une taxonomie pour répertorier les différentes approches

62 en deux catégories : les méthodes *a priori* et *a posteriori*. Si les  
63 premières se concentrent sur l'analyse des échantillons pour max-  
64 imiser la pertinence des prochains échantillons, les secondes se  
65 concentrent sur des méthodes de reconstruction d'images sans bruit  
66 à partir d'une entrée bruitée. Dans cet article, nous nous consacrons  
67 uniquement sur les méthodes *a posteriori*.

68 Dans les méthodes *a posteriori*, deux architectures majeures  
69 ressortent : les architectures dites de *direct prediction* représen-  
70 tées par le *Recurent Denoiser Auto-Encoder* (RDAE), que nous ap-  
71 pellerons auto-encodeur, proposé par Chaitanya et al. [CKS\*17],  
72 et les architectures par *kernel prediction* représentées par le *Kernel*  
73 *Prediction Convolutional Network* (KPCN) proposé par Bako et  
74 al. [BVM\*17]. La catégorie des *direct prediction* est conçue pour  
75 être utilisable dans un contexte de traitement d'images interactives  
76 très bruitées avec peu d'échantillons ( $\approx 1-4$ ), au prix de la présence  
77 d'artéfacts dans l'image débruitée (flou, perte de détails...). Con-  
78 trairement à celle-ci, les méthodes par *kernel prediction* évitent  
79 ces différents artéfacts mais nécessitent plus d'échantillons ( $\approx 64$ ).  
80 Dans notre contexte de rendu spectral, nous nous baserons sur  
81 l'adaptation des denoisers de type *direct prediction*, plus adaptés  
82 à notre niveau de bruit.

83 Yang et al. [YWH\*19] ont montré que la concaténation de cer-  
84 taines données (albédo avec l'image bruitée) pour l'entrée du de-  
85 noiser apportait des artéfacts. Ils ont donc proposé le réseau, ap-  
86 pelé *Dual Encoder Monte Carlo Denoiser* (DEMC) basé sur deux  
87 encodeurs et un décodeur, afin de dissocier l'extraction réalisée sur  
88 l'image bruitée et les données auxiliaires (albedo, normales, pro-  
89 fondeur...). Suivant le même principe visant à associer un traite-  
90 ment par donnée, Zhilin Cai et al. [CZM\*21] proposent de corrélér  
91 différents réseaux spécifiques à un bruit simple. L'intérêt est qu'en  
92 cumulant ces réseaux, le denoiser est capable de s'adapter à dif-  
93 férents types de bruit. Parallèlement à ces travaux, afin que le de-  
94 noiser se rapproche au maximum de la vérité terrain, l'utilisation de  
95 *Generative Adversial Network* (GAN) a également été mis en avant  
96 [XZW\*19]. Leur objectif est généralement d'intégrer un facteur  
97 dans la fonction d'erreur, évoluant suivant le résultat du générateur,  
98 i.e. le denoiser.

99 Cependant, même avec la séparation des entrées et l'utilisation  
100 de GAN, le débruitage d'un rendu de Monte Carlo reste complexe.  
101 Back et al. [BHHM20] et Arthur Firmino et al. [FFJ22] proposent  
102 d'utiliser un second estimateur. Placé après le denoiser, ce sec-  
103 ond réseau prend en entrée l'image bruitée et le résultat débruité  
104 afin de reconstruire de nouveau l'image finale pour y injecter de  
105 l'information non approximée, réduisant ainsi l'impact du denoiser  
106 à mesure que le nombre d'échantillons augmente. Ces solutions  
107 nous permettent de garantir que le résultat du rendu tendra tou-  
108 jours vers la vérité terrain même avec un denoiser. De plus, Back et  
109 al. [BHHM22] proposent une solution post-denoiser permettant au  
110 denoiser d'apprendre seul et de s'adapter à l'ensemble des bases de  
111 données, évitant ainsi de devoir créer une base de données suffisam-  
112 ment riche pour représenter l'ensemble des cas réels d'utilisations.  
113 Bien que ces dernières solutions soient très intéressantes, nous  
114 les avons écartées afin de rester dans un contexte contrôlé et de  
115 s'assurer, dans un premier temps, de la capacité de notre denoiser à  
116 corriger correctement les images issues d'un rendu de Monte Carlo  
117 spectral.

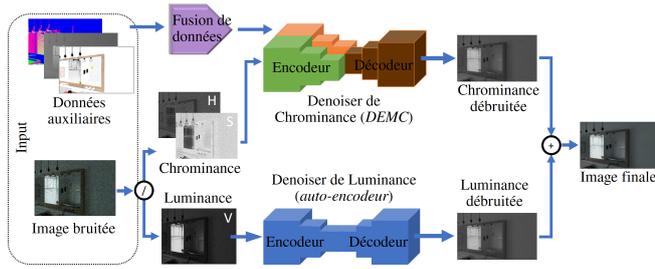


Figure 3: Architecture du pipeline TEMC-1 basée sur un DEMC et un auto-encodeur.

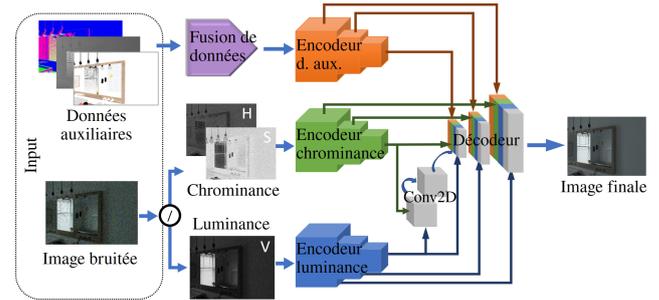


Figure 4: Architecture du pipeline TEMC-2.

118 Pour le rendu spectral, l'hypothèse que le denoiser se concentre  
 119 plus sur la chrominance que la luminance a été énoncée dans  
 120 [RNP\*22]. La solution proposée pour diminuer le problème était  
 121 d'utiliser deux denoisers distincts prenant exactement la même entrée.  
 122 Le premier denoiser est entraîné sur une base de données  
 123 d'images spectrales pour apprendre à débruiter la chrominance  
 124 alors que le second est entraîné sur une base de données RGB  
 125 pour apprendre à débruiter la luminance. La séparation de ces  
 126 canaux n'est réalisée qu'après leur denoising respectif afin de reconstruire  
 127 l'image finale. Cette méthode n'est pas optimale car les  
 128 deux denoisers ne sont pas entraînés spécifiquement pour corriger  
 129 leurs canaux réciproques et cette architecture empêche tout partage  
 130 d'information, entraînant ainsi, une baisse d'informations essentielles  
 131 pour la reconstruction. Nous pensons qu'une séparation des  
 132 canaux en amont permettrait de résoudre ces problèmes.

### 133 3. Approche proposée

134 Afin de prendre en compte notre hypothèse sur la séparation de  
 135 la chrominance et de la luminance, nous proposons une approche  
 136 appelée Triple Encodeur Monte Carlo (TEMC) basée sur trois encodeurs,  
 137 un par type de donnée. Il prend en entrée les données auxiliaires  
 138 (albedo, normal, profondeur...) et l'image spectrale bruitée convertie  
 139 dans l'espace sRGB en définition HDR Dolby Vision et ré-exprimée en HSV  
 140 afin de séparer chrominance (canal H, S) et luminance (canal V).  
 141 Toutefois, les données auxiliaires ne sont pas transmises directement  
 142 au réseau, elles sont converties en une carte, compressant l'ensemble  
 143 des informations extraites par le sous-réseau proposé et utilisé par  
 144 Yang et al. [YWH\*19] dans leur DEMC. Pour implémenter cette  
 145 approche, nous proposons deux architectures.  
 146

147 La première architecture, notée TEMC-1 et illustrée par la figure  
 148 3, est constituée de deux denoisers. Le premier est un DEMC constitué  
 149 de deux encodeurs et d'un décodeur qui traitent la chrominance et  
 150 les données auxiliaires compressées. Le deuxième denoiser est un  
 151 auto-encodeur [RNP\*22] (un encodeur et un décodeur) dédié au  
 152 traitement de la luminance. Si cette architecture permet de bien  
 153 séparer les différentes données, la reconstruction de l'image finale  
 154 passe par 2 décodeurs distincts, chacun dédié à un seul type de  
 155 données sans communication entre l'un et l'autre. Bien que cette  
 156 séparation améliore la qualité du résultat final, le manque de  
 157 communication engendre un temps d'apprentissage plus long.

158 Aussi, nous explorons également une seconde architecture, notée

159 TEMC-2 et illustrée en figure 4. Celle-ci permet l'interaction entre  
 160 les deux denoisers afin de reconstruire directement l'image finale  
 161 et s'efforce de réduire le temps d'apprentissage tout en améliorant  
 162 la qualité de la reconstruction. Elle reprend les 3 encodeurs et  
 163 fusionne les deux décodeurs distincts. Désormais, les résultats des  
 164 encodeurs de chrominance et de luminance (respectivement les blocs  
 165 vert et bleu dans la figure 4) sont concaténés et préparés par des  
 166 convolutions stabilisant l'apprentissage, avant d'être donnés en  
 167 entrée du décodeur. Chaque encodeur envoie les données extraites  
 168 à chacune des couches de ce dernier pour établir la communication.

### 169 4. Implémentation

170 **Auto-encodeur.** L'auto-encodeur utilisé se base principalement  
 171 sur l'architecture de Chaitanya et al. [CKS\*17]. Afin de fluidifier  
 172 le réseau, comme indiqué dans la littérature, nous avons délaissé  
 173 leur partie *recurrent correction*. Le réseau se construit par  
 174 l'enchaînement d'une convolution 2D de 32 filtres puis de 5 blocs  
 175 de *downsampling*, de 5 blocs d'*upsampling* et d'une convolution  
 176 2D avec 32 filtres où :

- 177 • un bloc de *downsampling* est constitué d'une convolution 2D et  
 178 d'un *max-pooling*. Les différentes convolutions utilisent réciproquement  
 179 32, 48, 56, 80 et 104 filtres.
- 180 • un bloc d'*upsampling* est constitué d'un *upsampling* suivi, de  
 181 deux convolutions 2D. Le nombre de filtre utilisé est réciproquement  
 182 : 152, 152, 112, 112, 88, 88, 64, 64, 64 et 32.
- 183 • pour l'ensemble des convolutions, nous avons utilisé un *kernel size*  
 184 de  $3 \times 3$ , un *stride* de  $2 \times 2$  et une activation de type *ReLU*,  
 185 sauf pour la dernière convolution qui a une activation de type  
 186 *Sigmoid*.

187 **TEMC-1.** Il est constitué d'un auto-encodeur comme détaillé  
 188 précédemment et d'un DEMC décrit dans l'article [YWH\*19] où  
 189 aucune modification des paramètres (nombre de couche, nombre de  
 190 filtre, *stride*) n'a été réalisée.

191 **TEMC-2.** Il est constitué de 3 instances d'encodeur et d'un  
 192 décodeur. Les blocs encodeur et décodeur proviennent du DEMC, sur  
 193 lesquels aucune modification n'a été réalisée. La communication  
 194 entre ces blocs, comme illustrée en figure 4, a été mise en place  
 195 avec l'utilisation de *skip connection*.

196 **Fonction d'erreur.** Afin d'expérimenter nos deux approches,  
 197 nous avons retenu la fonction d'erreur proposée par Zhao et al.

198 [ZGFK17]. Cette dernière pondère la *Mean Absolute Error* (MAE),  
 199 permettant ainsi de se rapprocher de la vérité terrain avec une com-  
 200 paraison pixel à pixel, et la MS-SSIM permettant d'analyser la  
 201 structure de l'image débruitée.

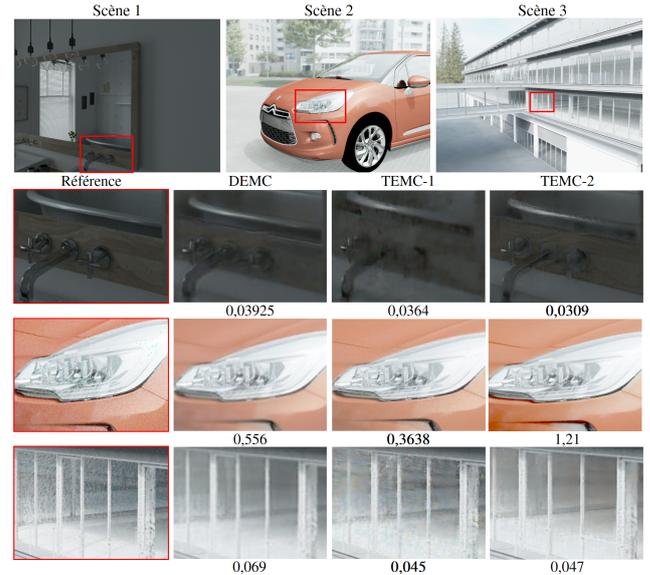
202 **Apprentissage.** Notre base d'images d'entraînement et de test a été  
 203 générée par le *path tracer* spectral [PJH16] d'UVR, *Predict Engine*  
 204 [Uni22]. Elle est constituée de 700 paires d'images, comprenant la  
 205 référence et l'entrée bruitée, d'une résolution de  $1024 \times 768$  pix-  
 206 els, où 210 d'entre elles sont utilisées pour le test et 490 sont utili-  
 207 sées pour l'entraînement. Afin de générer ces paires d'images,  
 208 nous avons utilisées 47 points de vue, pris dans 4 scènes avec  
 209 des effets de lumières distincts (éclairage direct/indirect, réflexion,  
 210 spéculaire...). De plus, pour réaliser une augmentation de données,  
 211 nous avons, d'une part, généré plusieurs fois chaque point de vue  
 212 avec des graines aléatoires assurant ainsi un bruit différent sur cha-  
 213 cune, et d'autre part utilisé des sélections aléatoires de sous-images  
 214 de  $128 \times 128$  générés à chaque époque. Les images de référence  
 215 sont calculées sur la base de 262k spp alors que les images bruitées  
 216 ont un nombre d'échantillon par pixel variant de 512 à 2048 spp,  
 217 nous permettant ainsi d'inclure des intensités de bruit différentes.

218 Les denoisers sont implémentés à l'aide de Tensorflow et de  
 219 Keras. Nous avons réalisé 5000 époques d'entraînement en utili-  
 220 sant l'optimisateur d'Adam, avec une valeur de *learning rate* fixée  
 221 à  $10^{-4}$  et un *batch size* de 32. Les entraînements ont été réalisés sur  
 222 4 GPUs Nvidia Tesla P100 (16 Go VRAM) avec des temps moyens  
 223 d'apprentissage de 6h pour DEMC, de 8h pour TEMC-2 et de plus  
 224 de 24h pour TEMC-1.

## 225 5. Résultats et discussion

226 La figure 5 illustre les résultats obtenus par le DEMC et nos  
 227 deux architectures TEMC-1 et 2. En comparant nos résultats avec  
 228 DEMC, nous pouvons constater que TEMC-1 et TEMC-2 obti-  
 229 ennent globalement une amélioration visuelle et quantitative sur  
 230 l'ensemble des scènes utilisées. Par exemple, en observant les  
 231 zooms des scènes 1 et 2 dans la figure 5, nous remarquons que  
 232 le TEMC gagne notamment sur l'aspect chromé du robinet et sur le  
 233 reflet spéculaire du bloc optique de la voiture. De manière générale,  
 234 quantitativement en utilisant la métrique RelMSE, nous notons un  
 235 gain d'environ 0.2 avec l'approche TEMC par rapport au DEMC.  
 236 En nous basant sur ces constatations, l'hypothèse posée sur la sépa-  
 237 ration des canaux de luminance et de chrominance est confirmée.

238 En comparant maintenant les deux architectures proposées,  
 239 TEMC-1 et TEMC-2, nous constatons la présence d'artefacts vi-  
 240 suels sur les images du TEMC-1, contrairement au TEMC-2 qui  
 241 offre des résultats perceptiblement plus élégants. Cependant et  
 242 mise à la part la scène 1, les résultats quantitatifs du TEMC-1,  
 243 malgré la présence de ses artefacts, surpassent ceux du TEMC-  
 244 2. Cette contradiction aux niveaux des résultats visuels et quan-  
 245 titatifs, nous amène à douter de la pertinence de l'utilisation de  
 246 la métrique RelMSE dans un contexte de rendu spectral. Tout-  
 247 fois, le gain dans les temps d'apprentissage entre ces deux ar-  
 248 chitectures, (pour rappel >24h pour le TEMC-1 et  $\approx 6$ h pour le  
 249 TEMC-2), confirme que la mise en place de communications inter-  
 250 blocs (encodeur/décodeur) est un réel atout et nous conforte dans  
 251 le développement du TEMC-2.



252 **Figure 5:** Résultats sur trois scènes des denoisers DEMC  
 253 [YWH\*19] et TEMC-1 et 2, appliqués sur des images bruitées  
 254 (1024 spp), avec leur taux d'erreur (métrique RelMSE).  
 255

## 252 6. Conclusion et perspectives

253 Nous avons abordé dans cet article la problématique du débruitage  
 254 d'images de rendues par méthodes de Monte Carlo spectrales.  
 255 Nous avons présenté l'hypothèse qu'un traitement séparé de la  
 256 luminance et de la chrominance favoriserait le restauration de  
 257 ce type d'images. Afin de la valider, nous avons proposé une  
 258 nouvelle approche, appelée TEMC, déclinée en deux architec-  
 259 tures : TEMC-1 et 2, où toutes les deux ont un encodeur par  
 260 type de données dès l'entrée du réseau. Leur différence se situe  
 261 au niveau de l'inter-communication entre encodeur/décodeur et  
 262 chrominance/luminance, qui est absente dans le TEMC-1 contrairement  
 263 au TEMC-2. La comparaison de leur temps d'apprentissage  
 264 respectif (+24h,  $\approx 8$ h) montre que cet ajout les réduit d'une manière  
 265 significative. La comparaison de nos résultats quantitatifs et visuels  
 266 avec ceux du DEMC confirme notre hypothèse sur la séparation.  
 267 Cependant, cette même comparaison entre les TEMC-1 et 2 nous  
 268 montre la présence d'artefact dans le premier tout en gardant une  
 269 mesure d'erreurs la plus faible. La perception visuelle du TEMC-2  
 270 étant supérieure avec un temps d'apprentissage le plus petit, nous  
 271 privilégierons dans nos futurs travaux son développement.

272 Fort de ce constat, nous pensons que la métrique RelMSE n'est  
 273 pas la plus adaptée dans un contexte de rendu spectral, une étude  
 274 est prévue pour améliorer la fonction d'erreur. En outre, l'ensemble  
 275 des méthodes actuelles se limite à une description du rendu spec-  
 276 tral aux 3 canaux de l'espace sRGB. Or le domaine spectral offre  
 277 une dimensionnalité bien supérieure, comme avec les données ra-  
 278 diométriques et photométriques définis par bandes spectrales. Cette  
 279 perte d'information est non négligeable, car leur intégration per-  
 280 mettrait au réseau d'apprendre les phénomènes physiques de la lu-  
 281 mière. Toutefois, leur prise en compte reste, à ce jour, une problé-  
 282 matique ouverte.

283 **Références**

- 284 [BHHM20] BACK J., HUA B.-S., HACHISUKA T., MOON B.: Deep  
285 combiner for independent and correlated pixel estimates. *ACM Trans-*  
286 *actions on Graphics* 39, 6 (Dec. 2020), 1–12. URL: [https://dl.](https://dl.acm.org/doi/10.1145/3414685.3417847)  
287 [acm.org/doi/10.1145/3414685.3417847](https://dl.acm.org/doi/10.1145/3414685.3417847), doi:10.1145/  
288 [3414685.3417847](https://doi.org/10.1145/3414685.3417847). 2
- 289 [BHHM22] BACK J., HUA B.-S., HACHISUKA T., MOON B.: Self-  
290 supervised post-correction for monte carlo denoising. In *ACM*  
291 *SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA,  
292 2022), SIGGRAPH '22, Association for Computing Machinery.  
293 URL: <https://doi.org/10.1145/3528233.3530730>, doi:  
294 [10.1145/3528233.3530730](https://doi.org/10.1145/3528233.3530730). 2
- 295 [BVM\*17] BAKO S., VOGELS T., MCWILLIAMS B., MEYER M.,  
296 NOVÁK J., HARVILL A., SEN P., DEROSE T., ROUSSELLE F.: Kernel-  
297 predicting convolutional networks for denoising Monte Carlo renderings.  
298 *ACM Transactions on Graphics* 36, 4 (July 2017), 1–14. URL: [https:](https://dl.acm.org/doi/10.1145/3072959.3073708)  
299 [/dl.acm.org/doi/10.1145/3072959.3073708](https://dl.acm.org/doi/10.1145/3072959.3073708), doi:10.  
300 [1145/3072959.3073708](https://doi.org/10.1145/3072959.3073708). 2
- 301 [CKS\*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C.,  
302 SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Inter-  
303 active reconstruction of Monte Carlo image sequences using a recur-  
304 rent denoising autoencoder. *ACM Transactions on Graphics* 36, 4  
305 (July 2017), 98:1–98:12. URL: [https://doi.org/10.1145/](https://doi.org/10.1145/3072959.3073601)  
306 [3072959.3073601](https://doi.org/10.1145/3072959.3073601), doi:10.1145/3072959.3073601. 2, 3
- 307 [CZM\*21] CAI Z., ZHANG Y., MANZI M., OZTIRELI C., GROSS M.,  
308 AYDIN T. O.: Robust Image Denoising using Kernel Predicting Net-  
309 works. 4. 2
- 310 [FFJ22] FIRMINO A., FRISVAD J. R., JENSEN H. W.: Progressive  
311 Denoising of Monte Carlo Rendered Images. Accepted: 2022-04-  
312 22T06:26:14Z Publisher: The Eurographics Association and John Wi-  
313 ley & Sons Ltd. URL: [https://diglib.eg.org/443/xmliu/](https://diglib.eg.org/443/xmliu/handle/10.1111/cgf14454)  
314 [handle/10.1111/cgf14454](https://diglib.eg.org/443/xmliu/handle/10.1111/cgf14454), doi:10.1111/cgf.14454. 2
- 315 [Int21] INTEL: Intel open image denoise. [https://www.](https://www.openimagedenoise.org)  
316 [openimagedenoise.org](https://www.openimagedenoise.org), 2021. 2
- 317 [Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of the*  
318 *13th Annual Conference on Computer Graphics and Interactive Tech-*  
319 *niques* (New York, NY, USA, 1986), SIGGRAPH '86, Association for  
320 Computing Machinery, p. 143–150. URL: [https://doi.org/10.](https://doi.org/10.1145/1145/15922.15902)  
321 [1145/15922.15902](https://doi.org/10.1145/15922.15902), doi:10.1145/15922.15902. 1
- 322 [NVi17] NVIDIA: Optix denoiser. [https://developer.nvidia.](https://developer.nvidia.com/rtx/ray-tracing/optix)  
323 [com/rtx/ray-tracing/optix](https://developer.nvidia.com/rtx/ray-tracing/optix), 2017. 2
- 324 [PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based*  
325 *Rendering: From Theory to Implementation*, 3rd ed. Morgan Kaufmann  
326 Publishers Inc., San Francisco, CA, USA, 2016. 4
- 327 [RNP\*22] ROUPHAEL R., NOIZET M., PRÉVOST S., DELEAU H.,  
328 STEFFENEL L.-A., LUCAS L.: Neural Denoising for Spectral Monte  
329 Carlo Rendering. In *Eurographics 2022 - Posters* (2022), Sauvage  
330 B., Hasic-Telalovic J., (Eds.), The Eurographics Association. doi:  
331 [10.2312/egp.20221011](https://doi.org/10.2312/egp.20221011). 3
- 332 [Uni22] UNITED VISUAL RESEARCHERS (UVR): Predict engine.  
333 <https://www.united-vr.com/>, 2022. 4
- 334 [VHH\*19] VORBA J., HANIKA J., HERHOLZ S., MÜLLER T.,  
335 KRIVÁNEK J., KELLER A.: Path guiding in production. In *ACM SIG-*  
336 *GRAPH 2019 Courses* (New York, NY, USA, 2019), SIGGRAPH '19,  
337 ACM, pp. 18:1–18:77. URL: [http://doi.acm.org/10.1145/](http://doi.acm.org/10.1145/3305366.3328091)  
338 [3305366.3328091](http://doi.acm.org/10.1145/3305366.3328091), doi:10.1145/3305366.3328091. 2
- 339 [XZW\*19] XU B., ZHANG J., WANG R., XU K., YANG Y.-L., LI C.,  
340 TANG R.: Adversarial Monte Carlo denoising with conditioned auxiliary  
341 feature modulation. *ACM Transactions on Graphics* 38, 6 (Nov. 2019),  
342 1–12. URL: [https://dl.acm.org/doi/10.1145/3355089.](https://dl.acm.org/doi/10.1145/3355089.3356547)  
343 [3356547](https://dl.acm.org/doi/10.1145/3355089.3356547), doi:10.1145/3355089.3356547. 2
- 344 [YWH\*19] YANG X., WANG D., HU W., ZHAO L.-J., YIN B.-C.,  
345 ZHANG Q., WEI X.-P., FU H.: DEMC: A Deep Dual-Encoder Net-  
346 work for Denoising Monte Carlo Rendering. *Journal of Computer Sci-*  
347 *ence and Technology* 34, 5 (Sept. 2019), 1123–1135. URL: <https://doi.org/10.1007/s11390-019-1964-2>, doi:10.1007/  
348 [s11390-019-1964-2](https://doi.org/10.1007/s11390-019-1964-2). 1, 2, 3, 4
- [ZGFK17] ZHAO H., GALLO O., FROSIO I., KAUTZ J.: Loss Functions  
for Image Restoration With Neural Networks. *IEEE Transactions on*  
*Computational Imaging* 3, 1 (Mar. 2017), 47–57. Conference Name:  
IEEE Transactions on Computational Imaging. doi:10.1109/TCI.  
2016.2644865. 4
- [ZJL\*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RA-  
MAMOORTHY R., ROUSSELLE F., SEN P., SOLER C., YOON S.-E.:  
Recent Advances in Adaptive Sampling and Reconstruction for Monte  
Carlo Rendering. *Computer Graphics Forum* 34, 2 (May 2015), 667–  
681. URL: [https://onlinelibrary.wiley.com/doi/10.](https://onlinelibrary.wiley.com/doi/10.1111/cgf.12592)  
1111/cgf.12592, doi:10.1111/cgf.12592. 2