



HAL
open science

Modèles de vision et synthèse d'images

Jean-Philippe Farrugia

► **To cite this version:**

Jean-Philippe Farrugia. Modèles de vision et synthèse d'images. Synthèse d'image et réalité virtuelle [cs.GR]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Université Jean Monnet - Saint-Etienne, 2002. Français. NNT : 2002EMSE0021 . tel-00800330

HAL Id: tel-00800330

<https://theses.hal.science/tel-00800330>

Submitted on 13 Mar 2013

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.

*

ÉCOLE NATIONALE SUPÉRIEURE
DES MINES DE SAINT-ÉTIENNE

UNIVERSITÉ JEAN MONNET
DE SAINT-ÉTIENNE

N° d'ordre : 297 ID

THÈSE

Présentée par Jean Philippe FARRUGIA

pour obtenir le titre de

Docteur

DE L'UNIVERSITÉ JEAN MONNET DE SAINT-ÉTIENNE ET DE
L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE SAINT-ÉTIENNE

Spécialité informatique – synthèse d'images

<p>Modèles de Vision et Synthèse d'Images</p>
--

Soutenue à Saint-Étienne, le 29 novembre 2002

Composition du jury :

M. Sumanta N. Pattanaik
M. Pascal Guitton
Mme. Françoise Viénot
M. Kadi Bouatouch
M. Bernard Péroche
M. Alain Trémeau

Rapporteurs

Examineurs

It's funny how the colors of the real world only seem really real when you viddy them on the screen.

Alexander De Large, alias Malcom McDowell, dans "Orange Mécanique" de Stanley Kubrick.

Remerciements

Voilà, la rédaction est enfin finie. Nous sommes le 6 février 2003, et je viens d'apporter les dernières corrections à mon manuscrit de thèse, résultat de trois ans de travail à l'École des Mines de Saint-Etienne. Lorsque j'y repense, avec le peu de recul que j'ai à l'heure actuelle, je ne peux réprimer un certain sentiment de nostalgie envers les différents événements qui ont eu lieu pendant ces trois ans. Le moment me paraît donc approprié pour remercier les personnes qui m'ont aidé, supporté et encouragé pendant ces trois années de labeur.

Je tiens en tout premier lieu à remercier mon directeur de thèse, Bernard Péroche, pour sa disponibilité et son soutien constant tout au long de ces travaux. Même si son départ pour Lyon a un peu compliqué les choses, il a toujours su rester à mon écoute.

Je voudrais également remercier Pascal Guitton et Sumanta N. Pattanaik d'avoir accepté de rapporter mon travail de thèse. Leurs conseils et leur expérience dans leurs domaines respectifs m'ont été très profitable pour mener à bien les nombreuses corrections de ce manuscrit.

Merci également à Alain Trémeau et Kadi Bouatouch pour leur remarques sur mon travail lors de la soutenance. J'essaie de faire de mon mieux pour tenir compte de l'opinion de chacun.

Un merci tout particulier à Françoise Viénot, qui a su corriger avec diligence les erreurs qui émaillaient la partie de ce manuscrit consacrée à la vision humaine, et qui a très gentiment accepté de faire partie de mon jury.

Xavier Serpaggi et Stéphane Albin m'ont supporté pendant plus de deux ans comme collègue de bureau, ce qui en fait presque des héros ! J'espère que ma naïveté et mes gamineries ne m'ont pas rendu trop pénible. . . Je souhaite bon courage à Stéphane, désormais seul et dernier thésard en synthèse d'images à Saint Etienne et j'espère pouvoir continuer à collaborer avec lui dans un avenir proche. Quand à Xavier, sa rigueur, son sens de l'organisation et ses compétences informatiques ne font aucun doute sur ses capacités à être un excellent ingénieur de recherche. Merci à eux deux pour leur support et leurs nombreux conseils pratiques.

Marc Roelens, dernier permanent encore actif du défunt laboratoire LISSE, maintient pratiquement seul l'activité synthèse d'images à l'EMSE, ce qui n'est pas un mince exploit au vu des résultats obtenus. Il a également toujours su trouver un peu de temps à consacrer à la recherche dans cette activité, et ce en dépit des très nombreuses fonctions qu'il cumule (ou a cumulé). En faisait vivre le laboratoire malgré sa "non existence" officielle, il a ainsi grandement contribué à l'aboutissement de ce travail. Qu'il en soit vivement remercié.

Je remercie également l'ensemble des doctorant et du personnel passé ou présent de l'espace Fauriel, tout particulièrement Marie Line Barneoud, Mireille Batton-Hubert, Jean Michel Moreau, Dominique Michelucci, Antoine Pauze, Annie Corbel, Philippe Jaillon, Jean François Tchébannoff, Gregory Six, Bertrand Sereno (cheers buddy, keep moving on !), Rodolphe Crespín, Jean Claude Iehl, Franck Gaultier, Faiza Abbacci, Camille Prime, Didier Vila, Laurence Gardes, Gildas Avoine, et tous ceux que j'ai oublié mais qui se reconnaîtront. . . Mention spéciale à Fernando Aguiar, pour nos nombreuses et interminables discussions, ainsi que pour les innombrables services qu'il m'a rendu avec sa voiture !

N'oublions pas mes nouveaux collègues de l'iutB de Villeurbanne, Bernard Bourriquen, Caroline Journay, René Crenn et Inaam Charaf, qui ont fait l'effort de venir à ma soutenance, et dont le soutien me fait chaud au coeur. Un grand merci également à François Maguin pour avoir courageusement assuré à ma place huit heures de travaux dirigés le jour de ma soutenance.

Le début de ma thèse à vu l'arrivée inattendue d'un tigre nommé Coralie, ma puce à moi, qui a probablement changé ma vie pour toujours. Sans elle, mon parcours n'aurait pas été le meme, et je veux qu'elle sache a quel point je suis heureux d'avoir trouvé quelqu'un qui partage mes passions, mes sentiments, mes peurs et mes joies. Je t'aime ma puce.

Je tiens également à remercier toute ma famille du coté de Dijon : mes parents et grands parents, mes frères Christophe, Jérôme et Nicolas, ma belle soeur Namiko, et ma nièce, la petite Elena. J'ajoute que la venue de mon père Denis et de mon grand père Carmel à ma soutenance a été une très grande joie et une immense fierté pour moi. Je demande à tous de croire sincèrement qu'il ne se passe pas un jour sans que je pense à eux.

Enfin, je tiens à remercier sincèrement les amis qui me sont restés fidèles malgré les années qui passent : Guillaume et Mireille Saget, qui m'ont fait l'honneur de me choisir comme parrain pour leur fille Erine, Thomas Dresler, ex Ensbanien, ex 3eme Cuir, actuel fan de James Bond et accessoirement officier comptable, Philippe "nismoGTR" Mendes, le roi du cokpit virtuel, et son amie Aya Matsuda, et Jérôme Da rugna, l'indexeur de textures cinéphile, pour leur soutien distant mais régulier. Quand à Olivier Tchou et Julien Vergne, le fait qu'ils aient fait le déplacement depuis (respectivement) Gueugnon et Velizy pour assister à ma soutenance me touche énormément. Soyez assurés que je ne vous oublie pas.

Résumé

Le photo-réalisme en synthèse d'images nécessite l'intervention de nombreux paramètres, ainsi que des calculs complexes. Le but peut être considéré comme atteint lorsque un observateur ne peut plus faire la différence entre une simulation et une photographie de la scène simulée. Pour atteindre cet objectif, il est nécessaire d'effectuer des calculs complexes, aussi bien au niveau de la géométrie de la scène, de la simulation des matériaux, ou encore de l'interaction lumière matière.

Toutefois, un élément est souvent oublié dans cette quête de réalisme : l'environnement d'observation. Cette simulation est en effet observée sur un dispositif de visualisation, par un observateur humain, placé dans un certain environnement. Or, chacune de ces trois composantes dispose de caractéristiques bien spécifiques qu'il peut être intéressant d'exploiter en synthèse d'images.

C'est ce que nous nous proposons de faire à travers une métrique perceptuelle entre images développée pour correspondre à nos besoins. Cette métrique permet de comparer deux images en utilisant un modèle de vision qui simule le traitement effectué par le système visuel humain. Elle sera ensuite intégrée dans un algorithme de rendu progressif, afin de remplacer les critères empiriques de ce dernier par des critères perceptuels. Les temps de calculs obtenus sont pratiquement deux fois inférieurs à ceux d'un calcul complet.

Abstract

Photorealism in image synthesis usually needs a huge number of parameters and complex computations. It is successful when an external observer is unable to see any difference between a photographic and a simulated picture. To achieve this goal, complex geometry, material surface and light-matter interaction simulations are needed.

But there is an element that is often neglected in this simulation process : the visualization environment. The simulated scene is viewed on a specific display, in a specific environment, by a human observer. Each one of these three components has specific characteristics which could be exploited in image synthesis.

Thus, we propose a perceptual image metric in order to perceptually compare two images by using a model which simulates the human vision process. We further integrate this metric into a progressive rendering algorithm, replacing the usual empiric parameters with perceptual ones. Computing times are then usually divided by two or three.

Table des matières

1	Introduction	1
1.1	Objectifs	1
1.2	Organisation	1
2	Notions de base	3
2.1	Synthèse d'images : principes et méthodes	3
2.2	La Modélisation géométrique	3
2.2.1	La représentation implicite	3
2.2.2	La représentation par arbre de construction (ou "Constructive Solid Geometry" (CSG))	4
2.2.3	La représentation par les bords	5
2.3	La création de la scène	5
2.3.1	L'apparence des objets	5
2.3.2	La construction de la scène	6
2.4	Le rendu	7
2.4.1	La radiosité	7
2.4.2	Le lancer de rayons	8
2.4.3	Un aspect important : l'obtention de l'image finale	8
2.5	Perception visuelle et synthèse d'images	10
2.5.1	Perception visuelle et modélisation géométrique	10
2.5.2	Perception visuelle et rendu	10
3	La perception visuelle	11
3.1	L'anatomie du système visuel	11
3.1.1	Le globe oculaire	11
3.1.2	La rétine	12
3.1.3	Les voies nerveuses et le cortex visuel	14
3.2	Caractéristiques	14
3.2.1	L'adaptation visuelle	15
3.2.2	La vision des contrastes	18
3.2.3	La vision des couleurs	21
3.3	Réflexions	27
4	Aspects Couleur	29
4.1	Notions essentielles de radiométrie et de photométrie	29
4.1.1	Radiométrie	29
4.1.2	Photométrie	29

4.2	Bases de colorimétrie	30
4.2.1	Hypothèses de base de la colorimétrie	30
4.2.2	Les fonctions colorimétriques	31
4.2.3	L'observateur standard CIE et l'espace colorimétrique CIE 1931 RGB	32
4.2.4	L'espace CIE 1931 XYZ	32
4.2.5	L'espace colorimétrique SML	33
4.2.6	L'(es) espace(s) colorimétrique(s) AC1C2	34
4.2.7	Colorimétrie des différences	34
4.2.8	L'espace CIELAB	35
4.2.9	Le modèle CIECAM97s	36
4.3	Reproduction de tons et Fidélité	37
4.3.1	Calibrage du dispositif de reproduction	37
4.3.2	Les opérateurs de reproduction de tons (ORT)	37
4.3.3	Fidélité et photo-réalisme	40
5	Métriques et Modèles	43
5.1	Principales métriques existantes	43
5.1.1	Les tests de Rushmeier <i>et al</i>	44
5.1.2	Les métrique de Gilles Rougeron et Stéphane Albin	45
5.1.3	La métrique de Neumann <i>et al</i>	46
5.1.4	La métrique de Gaddipatti <i>et al</i>	46
5.2	Les modèles de vision	47
5.2.1	Structure générale d'un modèle de vision	47
5.2.2	Le "Visual Differences Predictor" de Daly	48
5.2.3	Le "Visual Discrimination Model (VDM)" de Lubin	51
5.2.4	Le "Multiscale Model of Adaptation and Spatial Vision (MMASV)" de Pat- tanaik <i>et al</i>	54
5.3	Nos travaux	55
5.3.1	La structure de notre métrique "simple"	58
5.3.2	Résultats	61
5.3.3	Réflexions et conclusion sur la métrique	63
5.4	Un calcul adaptatif de la distance	66
5.4.1	Rappels sur la méthode adaptative de Stéphane Albin	66
5.4.2	Notre méthode adaptative	67
5.4.3	Résultats	68
5.4.4	Réflexions sur la métrique adaptative	72
5.5	Comparaisons	74
5.6	Conclusion	75
6	Rendu progressif et perceptuel	79
6.1	Le rendu	79
6.1.1	Définition	79
6.1.2	Le lancer de rayons	80
6.1.3	L'éclairage global	81
6.1.4	Les méthodes de types Monte Carlo	82
6.1.5	Le suivi de chemin, ou "Path tracing"	83
6.1.6	Les cartes de photons	84

6.1.7	Les vecteurs lumineux	84
6.1.8	Bilan Réflexions d'ordre général sur les algorithmes de calcul de l'éclairage global	85
6.2	Rendu progressif	86
6.2.1	Principes	86
6.2.2	La méthode de Painter et Sloan	87
6.2.3	Les méthodes à triangulation	88
6.2.4	La carte de cohérence directionnelle de Guo	91
6.3	Rendu perceptuel	93
6.3.1	Principes	93
6.3.2	Les travaux de Myszkowski <i>et al</i>	93
6.3.3	Les travaux de Bolin et Meyer	97
6.3.4	La méthode de Ramasubramanian <i>et al</i>	99
6.4	Notre approche	103
6.4.1	Essai préliminaire : pilotage de l'algorithme de Guo par notre modèle de vision	104
6.4.2	Notre méthode perceptuelle progressive	104
6.4.3	Résultats	107
6.4.4	Discussions sur les résultats et les paramètres	113
6.5	Conclusions	113
7	Perspectives et conclusion	121
7.1	Aspects spectraux	121
7.1.1	Pourquoi du multi-spectral ?	121
7.1.2	Modèles de vision et multi spectral	122
7.2	Plus de	124
7.2.1	La profondeur de champ	124
7.2.2	Simulation physique de l'appareil de visualisation	124
7.2.3	Les aspects temporels	125
7.2.4	Une meilleure gestion de l'information couleur	125
7.3	Adaptative et progressive ?	126
7.4	Bilan global	126
	Bibliographie	134
A	YART et les VEI	135
	L'environnement YART et les vecteurs lumineux	135
A.1	L'environnement utilisé	135
A.1.1	Description de scènes CSG avec CASTOR	135
A.1.2	Le logiciel YART	135
A.2	Les vecteurs lumineux	137
A.2.1	Principe	137
A.2.2	Le calcul des VEI	137
A.2.3	L'interpolation des VEI	138
A.2.4	Les améliorations de Xavier Serpaggi	139

Table des figures

2.1	Exemple de modélisation CSG	4
2.2	Foie modélisé en facettes	5
2.3	Exemples de FDRB	6
2.4	Image calculée avec l’algorithme de radiosit� (issue de http://hpc.doc.ic.ac.uk/~a.chung/parallelRadiosity.html)	7
2.5	Sch�ma de principe du lancer de rayons	8
2.6	Image calcul�e avec la m�thode du lancer de rayons	9
2.7	D�cimation de facettes, issu de [37]	9
3.1	Sch�ma global du syst�me visuel	11
3.2	Le globe oculaire	12
3.3	Coupe transversale de la r�tine	13
3.4	Les voies nerveuses du syst�me visuel	14
3.5	Limite de d�tection de diff�rence de luminance en fonction de la luminance du fond	15
3.6	Sensibilit� spectrale dans le domaine photopique	16
3.7	Sensibilit� spectrale des diff�rents capteurs en fonction de la luminance du fond	17
3.8	Seuil de luminance perceptible en fonction du temps pour chaque type de r�cepteur	17
3.9	R�ponse des cellules ganglionnaires pour diff�rents types de stimuli	18
3.10	Fonction de sensibilit� au contraste chromatique et achromatique	19
3.11	�volution de la fonction de sensibilit� au contraste selon la luminance de fond	19
3.12	Fonction de sensibilit� au contraste bidimensionnelle	20
3.13	Fonctions de sensibilit� fr�quentielle des champs r�cepteurs	20
3.14	Adaptation en contraste	21
3.15	Le ph�nom�ne de masquage	22
3.16	Exemple de masquage	22
3.17	Sensibilit� spectrale des cones	23
3.18	Adaptation chromatique	24
3.19	Perception de la transparence.	25
3.20	Premier exemple de contraste couleur	26
3.21	Deuxi�me exemple de contraste couleur	26
4.1	Fonctions colorim�triques de l’espace RGB	32
4.2	Fonctions colorim�triques pour l’espace XYZ	33
4.3	Diagramme de chromaticit� xy	34
4.4	Ellipses de Mac Adam	35
4.5	Sch�ma de principe de la reproduction de tons	38

4.6	Différents opérateurs de reproduction de tons : Tumblin-Rushmeier, Ward, Ward-Rushmeier-Piatko	38
4.7	Résultats de l'opérateur de reproduction de tons de Pattanaik <i>et al</i> (ligne du bas) . . .	40
4.8	comparaison scène réelle - image de synthèse (d'après [86])	41
5.1	A gauche : portrait intact (image 1), au milieu : portrait décalé (image 2), a droite : portrait bruité (image 3)	44
5.2	Schéma de fonctionnement du modèle de Daly	49
5.3	Test de validation du modèle de Daly (issu de [49])	50
5.4	Probabilité de perception de différences en fonction de la distance d'observation (issu de [49])	50
5.5	Probabilité de perception de différences en fonction du temps de calcul(issu de [49]) .	51
5.6	Fonctionnement du Visual Discrimination Model	52
5.7	Test de Meyer et Bolin ([7])	54
5.8	Le "Multiscale Model of Adaptation and Spatial Vision" ([54])	56
5.9	Résultats du MMASV pour le test des contrastes simultanés	57
5.10	Résultats du MMASV pour le test d'assimilation	57
5.11	Résultats du MMASV pour l'adaptation chromatique	58
5.12	Notre métrique perceptuelle	59
5.13	Résultats du test "Bach"	62
5.14	Résultats du test "masquage"	64
5.15	Résultats du test "contrastes simultanés"	65
5.16	Résultats du test "diffusion - assimilation"	65
5.17	Cartes des distances pour le test "Bach"	69
5.18	Cartes des distances pour le test "Masquage"	70
5.19	Images comparées lors du test "Salle à manger"	71
5.20	Cartes des distances pour le test "Salle à manger"	71
5.21	Résultats obtenus pour la scène de la salle à manger en faisant varier les paramètres de la distance adaptative	73
5.22	Tests utilisés pour la comparaison des métriques	75
5.23	Résultats des tests de comparaisons des métriques	76
6.1	Schéma de principe du lancer de rayons	80
6.2	Illustration graphique de la méthode de Monte Carlo	82
6.3	Données utilisées par la méthode de prévisualisation de Pighin (de haut en bas, et de gauche a droite) : position des bords géométriques, maillage de Delaunay contraint associé, carte des textures sans calcul d'éclaircement, image résultat.	89
6.4	Données utilisées par la méthode de Simmons (de haut en bas, et de gauche à droite) : maillage de Delaunay, carte de priorité, échantillons calculés, image reconstruite. . .	90
6.5	Classification des blocs pour la méthode de Guo	92
6.6	Résultats de la méthode de Guo. A gauche : calcul complet. A droite : DCM	93
6.7	Résultats du test "comparaison avec l'image finale" (de haut en bas, de gauche à droite) : image issue de la première itération, image finale issue du rendu "suivi de chemins", différence pixel à pixel, sortie du VDP. (issu de [49])	94
6.8	Résultats du test "VDP utilisé comme condition d'arrêt", issu de [49]	95
6.9	Résultats de la méthode fréquentielle de Bolin et Meyer (issu de [6])	98

6.10	Résultats de la méthode perceptuelle de Bolin et Meyer (<i>issu de [7]</i>). A gauche : rendu perceptuel. A droite : rendu Monte Carlo classique. Ligne inférieure : Schéma d'échantillonnage.	100
6.11	Résultats de la méthode perceptuelle de Ramasubramanian <i>et al</i> (<i>issu de [57]</i>).	103
6.12	Comparaison algorithme de Guo avec et sans critères perceptuels	104
6.13	Schéma de fonctionnement de notre méthode progressive perceptuelle	105
6.14	Résultats obtenus sur la scène "crypte"	108
6.15	Comparaison entre l'image approximée et l'image calculée entièrement sur la scène "crypte", seuil fixé à 5 JNDs	109
6.16	Comparaison entre l'image approximée et l'image calculée entièrement sur la scène "pièce avec chaise", seuil fixé à 9 JNDs	110
6.17	Comparaison entre l'image approximée et l'image calculée entièrement sur la scène "labo photo", seuil fixé à 1 JND	111
6.18	La scène "Salle à manger"	112
6.19	Résultats obtenus sur la scène "salle à manger" en faisant varier le seuil S_{sup}	115
6.20	Cornell Box obtenue avec notre méthode. Le taux d'échantillonnage est de 45 %, les seuils sont fixés à $S_{inf} = 0$ et $S_{sup} = 2,5$	116
6.21	Résultats obtenus sur les scènes "Labo photo" et "Cornell Box" en faisant varier le seuil S_{sup}	117
6.22	Résultats obtenus sur les scènes "Pièce avec chaise" et "Crypte" en faisant varier le seuil S_{sup}	118
6.23	Résultats obtenus sur les scènes "Labo photo" et "Cornell Box" en faisant varier le seuil S_{inf}	119
6.24	Résultats obtenus sur les scènes "Pièce avec chaise" et "Crypte" en faisant varier le seuil S_{inf}	120
7.1	Simulation spectrale de la dispersion de la lumière dans un diamant. En vignette : photographie réelle d'un diamant (<i>issu de [69]</i>)	122
7.2	Simulation spectrale (à gauche) et RGB (à droite) d'une bouteille de vin. La différence est assez flagrante sur le vin blanc, qui apparait vert en RGB ! (<i>issu de [69]</i>)	123
7.3	Simulations spectrale et RGB de l'éclairement d'une cornell box par une lampe à vapeur de sodium (images fournies par Jean Claude Iehl)	123
7.4	La même photographie avec deux profondeurs de champs différentes	124
7.5	Exemple de "motion blur"	125
A.1	Exemple de programme CASTOR C. Celui-ci décrit une cornell box.	136
A.2	Calcul des VEI	137
A.3	Les trois cas ne permettant pas l'utilisation des VEI du voisinage : mauvaise position relative, direction d'observation trop différentes, mauvaise orientation	139

Chapitre 1

Introduction

De tout temps, l'homme a toujours cherché à transcrire graphiquement le monde qui l'entoure. Des premières peintures des hommes primitifs aux techniques les plus modernes, son but est resté le même : reproduire fidèlement ce qu'il voit avec ses yeux.

L'avènement de l'informatique a permis l'émergence d'une nouvelle technique : la synthèse d'image. Elle consiste à obtenir une image par des moyens informatique en simulant la visualisation d'une scène à travers un dispositif optique. Des calculs complexes sont donc mis en oeuvre pour simuler les matériaux, l'éclairage et l'interaction de la lumière avec les objets. Les progrès dans ce domaine sont rapides, et il est aujourd'hui possible d'obtenir des images d'un réalisme confondant. Toutefois, les temps d'obtention sont encore élevés pour aboutir à un tel résultat. Parmi toutes les voies explorées visant à réduire le temps de calcul, il en est une dont l'exploitation est relativement récente : la prise en compte de la vision humaine.

1.1 Objectifs

Dans ce travail de thèse, notre objectif est de montrer comment il est possible de réduire les temps de calcul en utilisant les connaissances actuelles sur les capacités du système visuel humain. Nous adapterons à nos besoins un modèle informatique de la vision humaine, puis vérifierons la validité de ce modèle en l'utilisant dans une métrique perceptuelle permettant de comparer deux images. Cette métrique sera ensuite exploitée pour piloter un algorithme de rendu progressif en remplacement des critères empiriques généralement utilisés.

1.2 Organisation

La suite de ce mémoire de thèse sera organisé de la manière suivante :

- le chapitre 2 rappèlera quelques notions de base de synthèse d'image, et montrera à quels niveaux la connaissance de la vision humaine peut intervenir en synthèse d'images.
- le chapitre 3 présentera les caractéristiques du système visuel qui nous paraissent importantes.
- nous verrons dans le chapitre 4 les différents aspects de la couleur en synthèse d'image, incluant la colorimétrie, la reproduction de ton et les problèmes de fidélité.
- le chapitre 5 introduira les différentes métriques et modèles de vision existant à l'heure actuelle. Nous présenterons ensuite notre métrique perceptuelle entre images et nous comparerons ses performances avec celles d'autres algorithmes existants.

- le chapitre 6 sera consacré au rendu progressif et aux différents moyens d'intégrer un critère perceptuel dans ce type de rendu. Nous y présenterons également un algorithme de rendu progressif perceptuel, utilisant notre métrique comme critère de subdivision, et nous exposerons quelques résultats.
- Nous concluerons dans le chapitre 7 avec un bilan et les perspectives et extensions possibles de nos travaux.

Chapitre 2

Notions de base de synthèse d'images

Ce chapitre d'introduction a pour but de présenter un tour d'horizon rapide des concepts de base de la synthèse d'images. A travers ces concepts, nous introduirons la notion de perception visuelle, qui est susceptible d'intervenir à tous les stades de la création d'une image. Ceci permettra d'aboutir naturellement aux problèmes qui sont au coeur de nos travaux.

2.1 Synthèse d'images : principes et méthodes

Le but de la synthèse d'image est la conversion de la description symbolique et mathématique d'une scène en données numériques directement visualisables sur un moniteur ou tout autre dispositif de reproduction graphique. Plus pratiquement, ce processus peut être apparenté au maquettisme : après avoir conçu les plans d'une maquette, il est possible de la construire, de la peindre, puis de l'éclairer et la prendre en photo. Selon ce schéma, la création d'une image de synthèse peut se décomposer en trois phases principales : la modélisation géométrique des objets, la création de la scène, et le rendu.

2.2 La Modélisation géométrique

La création d'une image commence donc par le passage d'une idée des objets de la scène à une modélisation mathématique de ces objets. Le problème est donc le suivant : comment représenter, dans la mémoire d'un ordinateur, les propriétés géométriques d'un objet (nous verrons ses propriétés physiques dans la section suivante) ? Dans cette section, nous présenterons trois méthodes de modélisation, choisies pour être particulièrement courantes et adaptées à nos travaux, mais il en existe évidemment beaucoup d'autres.

2.2.1 La représentation implicite

L'idée la plus naturelle, analogue à ce qui existe dans d'autres domaines scientifiques, consiste à modéliser un objet par la définition, l'équation ou le système mathématique le décrivant. Par exemple, une sphère peut être décrite par l'inéquation suivante :

$$x^2 + y^2 + z^2 < R$$

R étant ici le rayon de ladite sphère. Cette représentation est très compacte et élégante mais présente néanmoins des défauts assez incontournables. Premièrement, si il est facile de représenter des objets géométriques simples comme un cube ou une sphère, il est beaucoup moins aisé de déterminer

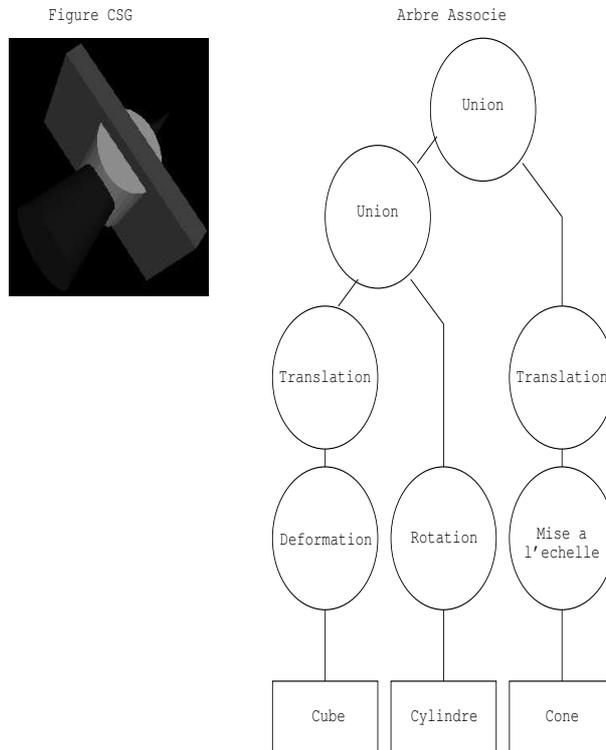


FIG. 2.1 – Exemple de modélisation CSG

l'équation d'objets complexes comme une main humaine ou une clé à molette. Deuxièmement, il faut considérer le fait que nos objets sont modélisés dans le but d'être visualisés sur écran par la suite. Malheureusement, rendre directement visualisable une fonction mathématique peut être coûteux : par exemple, en utilisant le lancer de rayons, il faut, pour chaque pixel, savoir calculer avec précision l'intersection d'une droite avec notre objet (donc savoir trouver les racines de l'équation correspondante), ce qui n'est pas forcément trivial.

2.2.2 La représentation par arbre de construction (ou “Constructive Solid Geometry” (CSG))

Cette technique repose sur le fait qu'il est possible de représenter des objets complexes comme une composition d'objets simples appelés primitives : par exemple, une table n'est rien de moins que l'assemblage de cinq cubes préalablement déformés. Nous avons vu que la représentation implicite permettait de modéliser des objets basiques. Nous pouvons alors créer des objets plus complexes en utilisant des opérateurs booléens comme la réunion ou l'intersection sur les primitives. En pratique, un objet CSG se présente sous la forme d'un arbre dont les feuilles sont les primitives utilisées et dont les noeuds internes sont les opérateurs. La figure 2.1 montre un exemple de modélisation CSG avec l'arbre associé. Cette technique s'avère performante en ce sens qu'elle permet de modéliser des objets complexes avec une taille des données très réduite. De par sa structure hiérarchique, elle permet aussi une modification locale de l'objet et est par conséquent très utilisée dans les logiciels de conception assistée par ordinateur. Notons que le lancer de rayon est particulièrement adapté à la visualisation d'objets CSG car il est possible de composer les intersections rayon/primitive pour obtenir les intersections rayon/objet. La CSG n'a toutefois pas que des avantages : il subsiste un

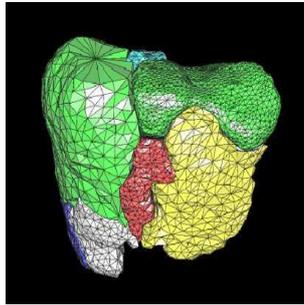


FIG. 2.2 – Foie modélisé en facettes

grand nombre d'objets, notamment les objets "naturels"(arbres, montagnes, êtres vivants), qui restent assez difficilement modélisables en CSG car leur géométrie ne résulte pas vraiment de l'assemblage macroscopique de primitives simples.

2.2.3 La représentation par les bords

Il est aussi possible de représenter un objet en matérialisant la limite entre son intérieur et son extérieur. Cette frontière est délimitée par une série d'éléments géométriques plans (généralement des triangles) reliés entre eux par leurs bords. Cette méthode offre de nombreux avantages. Tout d'abord, le modèle obtenu est facilement et rapidement visualisable car la plupart des cartes graphiques actuelles prennent en charge l'affichage de triangles tridimensionnels. Cette structure est aussi aisément déformable, aussi bien localement que globalement. Enfin, n'importe quel objet continu est représentable de cette manière car ce procédé s'apparente à un échantillonnage, et il n'existe pas d'autre limite que la mémoire disponible. Mais c'est là le problème majeur : pour obtenir une représentation correcte d'un objet un tant soit peu complexe, un grand nombre de triangles est nécessaire. La figure 2.2 représente un foie humain modélisé avec des facettes ; il apparaît évident sur cette figure que les facettes sont très nombreuses dans le but d'obtenir une précision suffisante. Les données obtenues sont donc beaucoup plus volumineuses qu'avec les représentations précédentes. Il est aussi possible que des problèmes d'ordre topologique surviennent (distinction extérieur/intérieur pas toujours évidente, triangles qui se chevauchent après une déformation ...).

2.3 La création de la scène

La description géométrique des objets ne suffit pas à elle seule à constituer une image. D'une part, ces objets n'ont pas tous la même apparence : certains sont sans doute mats, d'autres brillants, d'autres encore ont une surface bosselée ou nervurée... D'autre part, certains éléments de la scène comme l'œil, la direction de visée, le plan de projection ou les sources lumineuses ne sont pas des objets à proprement parler. Il nous faut donc maintenant spécifier tous ces paramètres pour obtenir une scène complète.

2.3.1 L'apparence des objets

Il s'agit ici d'attribuer aux objets leurs propriétés surfaciques (voire volumiques) dont on peut distinguer trois classes : les propriétés optiques, la couleur et la texture. Les premières sont généralement

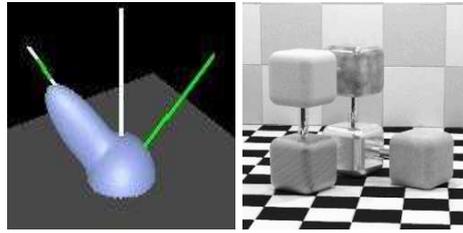


FIG. 2.3 – Exemples de FDRB

décrites par des paramètres comme une fonction de distribution de réflectance/transmittance bidirectionnelle. Cette fonction à cinq variables donne l'énergie réfléchie/transmise dans une direction d_2 en fonction de l'énergie reçue selon une direction d_1 . Elle sera relativement uniforme pour un matériau diffus mais présentera un pic pour un matériau brillant. La figure 2.3 donne à gauche un exemple de FDRB et à droite une scène comportant des cubes dont la réflectance est définie par des FDRB de (de haut en bas et de gauche à droite) ciment, peinture, bois, aluminium et béton.

L'attribution d'une couleur à un objet est un problème plus complexe qu'il n'y paraît au premier abord. Le moyen le plus usité est de choisir 3 coefficients R,G et B, tous compris entre 0 et 1. Malheureusement, il est établi que la perception des couleurs n'est pas cohérente avec ce système (nous reviendrons sur ce point dans le chapitre 4, consacré à la colorimétrie). Il est aussi possible d'attribuer des propriétés spectrales au matériau, mais la phase de rendu est alors beaucoup plus complexe. Et que faire si notre objet émet de la lumière ? Nous reviendrons sur ce problème ultérieurement dans ce mémoire.

Les textures ont fait, et font toujours, l'objet de recherches intensives dans le domaine de la synthèse d'images. Ici encore, il est possible de distinguer plusieurs méthodes d'obtention de textures. La méthode la plus classique consiste à plaquer une image 2D sur un objet 3D, avec tous les problèmes de déformation que cela entraîne. Les textures 3D sont quand à elles générées par un algorithme spécifique pour chaque motif (bois, marbre). Il apparaît évident que le choix de la méthode se fera en fonction de la complexité de la texture à représenter. Certaines textures permettent aussi de déformer localement la surface de l'objet à texturer en altérant les normales, de manière obtenir des craquelures ou des petites bosses. La combinaison des trois techniques précédentes permet de renforcer considérablement l'aspect convaincant des objets. Le domaine de recherche associé aux textures est très vaste et comporte de nombreuses autres variantes et applications : environment mapping, texels ...

2.3.2 La construction de la scène

Une fois l'apparence des objets finalisée, il nous reste à fixer les paramètres non géométriques de notre scène. Il s'agit de la position et du type des sources de lumière, de la position de la caméra (ou de l'œil), et des plans de découpage. Pour chaque source de lumière, il faudra préciser son type (uniforme ou directionnelle, ponctuelle, surfacique ou volumique ...), sa position dans l'espace et sa couleur (voire sa distribution spectrale de puissance et son solide photométrique). La caméra se voit attribuer six paramètres géométriques correspondant aux six degrés de liberté d'un objet dans l'espace, ainsi qu'un angle d'ouverture. Enfin, les plans de découpage servent à délimiter la tranche d'espace dans laquelle sera considéré l'affichage de la scène. Une distance focale peut également être ajoutée si la simulation de la profondeur de champ est nécessaire.



FIG. 2.4 – Image calculée avec l’algorithme de radiosité (issue de <http://hpc.doc.ic.ac.uk/.../a.chung/parallelRadiosity.html>)

2.4 Le rendu

C’est la phase de simulation proprement dite. Maintenant que notre scène est complètement déterminée, il nous faut convertir cette description symbolique en une image affichable sur le médium choisi. Ce problème est aujourd’hui résolu en temps réel dans un certain nombre de cas particuliers : si les matériaux sont simples (par exemple non réfléchissants), que toutes les sources de lumière sont ponctuelles et que l’on suppose que les objets n’interagissent qu’avec eux-mêmes, alors la majorité des cartes graphiques du commerce sont capables de prendre en charge le calcul de l’image directement via leur chipset spécialisé. Toutefois, pour un aspect plus réaliste, il est nécessaire de prendre en compte l’illumination globale, c’est à dire la totalité des interactions des objets de la scène entre eux. L’équation du rendu de Kajiya ([31]) donne une formalisation mathématique de ce problème. Les moyens de parvenir à cette fin sont nombreux, mais quelquefois coûteux à mettre en oeuvre. Nous évoquerons brièvement ici la radiosité et les méthodes stochastiques comme le lancer de rayons.

2.4.1 La radiosité

Le calcul de radiosité a été introduit par Goral *et al* ([22]) en 1984. Il s’inspire de travaux issus de la thermique. La scène est préalablement découpée en surfaces élémentaires appelées patches. Le calcul de l’éclairage global est ensuite assimilé à un calcul de transfert d’énergie entre les différents patches. Il en existe plusieurs variantes (radiosité progressive ou hiérarchique principalement). Nous ne rentrerons pas dans les détails techniques ici, mais il faut savoir que pour calculer les échanges d’énergie, la principale difficulté de cette méthode consiste à déterminer la visibilité d’un patch par rapport à un autre via les facteurs de forme. Cette méthode présente l’avantage considérable de fournir un résultat totalement indépendant du point de vue. Par contre, elle est beaucoup plus difficile à appliquer si la scène contient des objets qui ne sont pas purement diffus. Elle est également mal à l’aise avec les géométries très complexes, très gourmande en mémoire, et les temps de calculs peuvent atteindre plusieurs heures. Elle ne sera que très peu évoquée dans ce mémoire, car le logiciel utilisé au laboratoire LISSE est basé sur le lancer de rayons. La figure 2.4 montre une image calculée avec cette méthode.

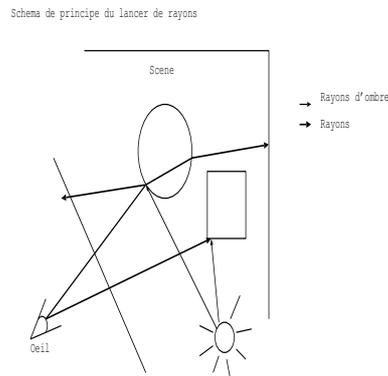


FIG. 2.5 – Schéma de principe du lancer de rayons

2.4.2 Le lancer de rayons

Introduit par Appel([4]) en 1968, mais repris par Whitted([81]) en 1979, le lancer de rayons s'appuie sur l'optique géométrique afin de simuler le trajet de l'énergie lumineuse dans la scène. La figure 2.5 donne une illustration graphique du fonctionnement de la méthode : un plan de projection, placé devant l'œil ou la caméra, représente l'image visualisée. Pour chaque pixel de ce plan, nous lançons un rayon (par rayon, nous entendons un point d'origine et une direction), puis nous calculons l'intersection éventuelle de ce rayon avec les différents objets de la scène. Si une intersection est trouvée, il faut alors calculer la luminance du pixel considéré. Pour un rendu rapide, la seule interaction avec la surface de l'objet intersecté suffit. Nous ne détaillerons pas ici cette interaction, signalons simplement que cela fait intervenir les paramètres fixés dans le paragraphe 2.3.1. Le rayon peut éventuellement poursuivre son chemin si le matériau de l'objet est transparent, par exemple. Pour une image plus réaliste, il faudra par contre prendre en compte l'éclairage indirect. En lancer de rayons, le moyen le plus efficace pour résoudre l'équation de rendu consiste à adopter une méthode de type Monte-Carlo ; ainsi, pour chaque intersection trouvée, un certain nombre de rayons sont relancés à partir de ce point dans des directions aléatoires. La luminance finale résultera de la somme de la contribution de tous ces rayons secondaires. Le grand avantage du lancer de rayon par rapport à la radiosité est qu'il peut gérer n'importe quel type de matériau, qu'il soit transparent, brillant, diffus ou autre. Il n'est pas limité par la géométrie de la scène. Mais, comme pour la radiosité, les temps de calculs sont très longs pour peu que l'utilisateur désire une image photoréaliste. Un grand nombre d'optimisations existent, mais elles seront évoquées ultérieurement dans ce mémoire. La figure 2.6 donne un exemple d'image calculée avec cette méthode.

2.4.3 Un aspect important : l'obtention de l'image finale

Le processus de rendu fournit une carte de luminances réelles. Il nous faut maintenant transformer cette carte afin de pouvoir la visualiser. Cette étape est beaucoup moins triviale qu'on ne pourrait le croire. En effet, la dynamique de luminance de cette carte dépasse de très loin celle d'un moniteur classique par exemple ; de plus, si l'image a été calculée avec des attributs spectraux pour les sources de lumière et les matériaux, il ne sera peut être pas évident de restituer correctement les "bonnes" couleurs. Comment, dans ces conditions, retranscrire "fidèlement" l'image calculée ? Ce problème est lié à une grande partie des travaux présentés dans ce mémoire, il sera donc largement repris par la suite.

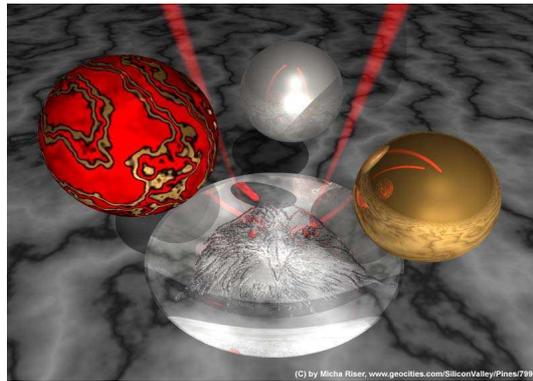


FIG. 2.6 – Image calculée avec la méthode du lancer de rayons

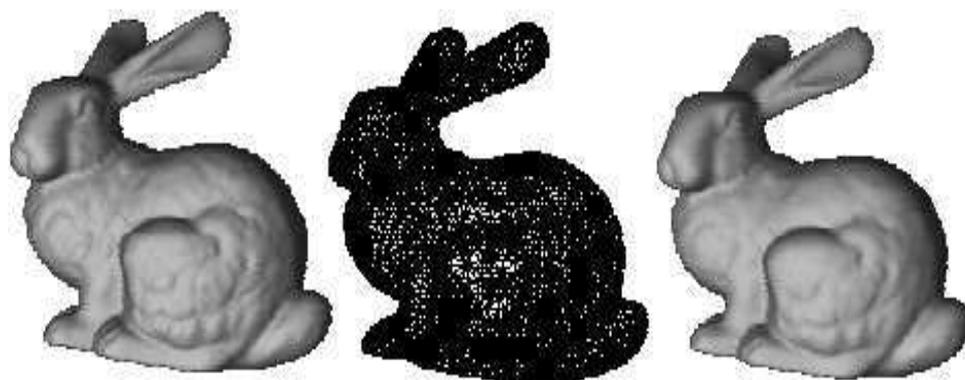


FIG. 2.7 – Décimation de facettes, issu de [37]

2.5 Perception visuelle et synthèse d'images

Comme nous venons de le voir, l'obtention d'une image de synthèse est le produit de processus variés couvrant tous les aspects géométriques (modélisation, paragraphe 2.2), visuels (apparence des objets, paragraphe 2.3.1) et lumineux (rendu, paragraphe 2.4). Il subsiste toutefois un aspect qui est presque systématiquement oublié : la perception visuelle humaine. En effet, ces images sont produites dans le but d'être visionnées par un observateur humain, et nous pouvons alors nous poser la question du niveau auquel peut intervenir ce paramètre dans la génération d'une image. Nous allons apporter quelques éléments de réponse dans les sections suivantes.

2.5.1 Perception visuelle et modélisation géométrique

La perception visuelle peut affecter le processus de modélisation d'un objet. Par exemple, dans le cas de la représentation par les bords, le fait que les facettes qui constituent l'objet soient planes, et génèrent donc des discontinuités d'ordre un, va nuire à son aspect "lisse". Nous verrons ultérieurement que cet effet peut être, entre autres, compensé ou accentué par la texture appliquée sur l'objet. Un modèle de perception visuelle pourrait aussi servir à déterminer le nombre minimal de facettes pour un objet de façon à ce qu'aucun artefact ne soit perceptible. Il pourrait alors piloter un algorithme de décimation de facettes. Ainsi, il est possible d'envisager un algorithme qui, en fonction du contexte perceptuel d'un objet, ajuste en temps réel le nombre de facettes nécessaires à sa visualisation (voir [36], [37] ou [83]). Cette méthode intéresse tout particulièrement le domaine de la réalité virtuelle, où l'affichage doit obligatoirement se faire en temps réel.

La figure 2.7, issue de [37], met en évidence la décimation perceptuelle d'un modèle à facettes représentant un lapin. La fonction utilisée considère que l'on regarde la page à une distance de 29 centimètres en se focalisant sur le haut de la feuille. Dans ces conditions d'observation, il ne devrait pas y avoir de différences perceptibles entre le modèle complet et le modèle décimé.

2.5.2 Perception visuelle et rendu

En plus de simplifier les modèles présents dans la scène, la prise en compte du système visuel humain peut aussi diminuer les temps de calcul de l'image ; il semble par exemple inutile de calculer l'illumination exhaustivement si la perception de la scène n'est pas complète. C'est sur ce point que porte l'essentiel de ce travail de thèse. L'objectif est donc la mise au point d'un modèle informatique de la vision humaine adapté à la synthèse d'images. Il sera destiné à prévoir la perceptibilité des différents aspects et détails de l'image. De cette manière, il nous sera possible de guider un algorithme de rendu pour obtenir progressivement une image de qualité croissante.

Chapitre 3

La perception visuelle

Avant de pouvoir intégrer la vision humaine dans nos algorithmes de rendu, il faut déjà connaître le fonctionnement de celle-ci. Dans ce chapitre, nous nous proposons d'étudier le système visuel. Dans un premier temps, nous aborderons la physiologie de l'œil, principalement la rétine, puis nous présenterons les caractéristiques de la vision établies à travers un certain nombre de tests sur des sujets humains. C'est à travers l'étude de ces caractéristiques qu'il nous sera possible de construire un modèle de vision.

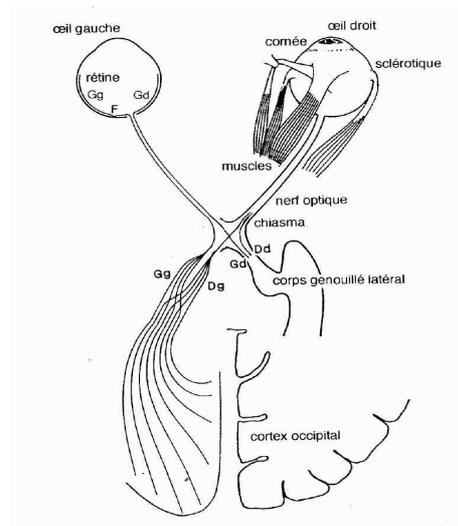


FIG. 3.1 – Schéma global du système visuel

3.1 L'anatomie du système visuel

La figure 3.1 présente un schéma général de l'anatomie du système visuel. Dans cette partie, nous détaillerons le globe oculaire, la rétine et les voies nerveuses.

3.1.1 Le globe oculaire

Sans rentrer dans les détails, signalons qu'il est constitué, de l'extérieur vers l'intérieur :

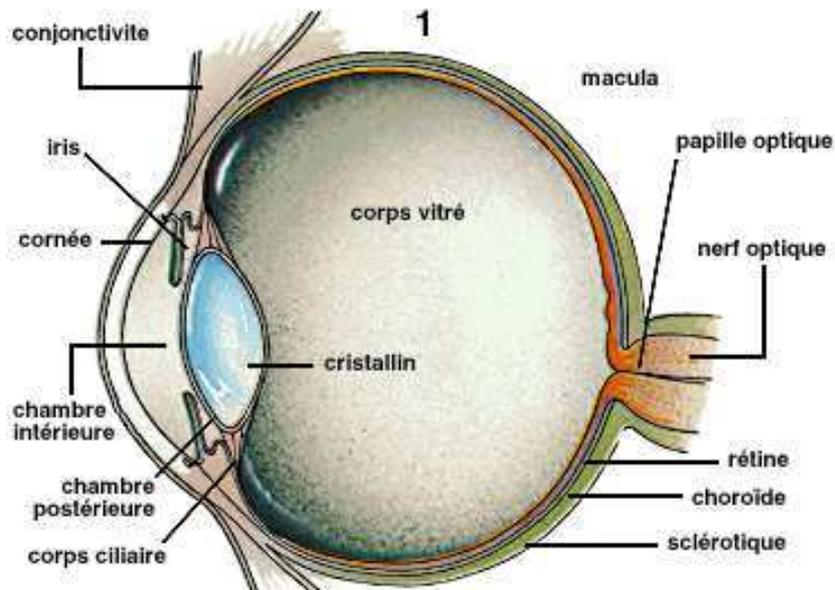


FIG. 3.2 – Le globe oculaire

- d'une enveloppe extérieure, la sclérotique, dotée à l'avant d'une partie transparente nommée la cornée.
- d'une enveloppe intérieure, la choroïde sur laquelle est placée la rétine.
- L'intérieur est globalement séparé en deux cavités :
 - une cavité antérieure remplie d'un liquide salin (l'humeur aqueuse) à l'intérieur de laquelle se trouve le cristallin et l'iris.
 - une cavité postérieure contenant un fluide albumineux, l'humeur vitrée.

Le globe oculaire mesure environ 2,5 centimètres de diamètre. Enfin, les deux yeux sont généralement séparés d'une distance de 6 centimètres environ. La figure 3.2 illustre cette description sommaire (source : <http://www-rocq.inria.fr/Marc.Thiriet/Glosr/Bio/Sens/oeil.html>)

Le processus de vision au niveau du globe oculaire s'appuie sur quatre composants essentiels. Premièrement, trois paires de muscles permettent de choisir une direction de visée. Deuxièmement, l'iris, s'ouvre ou se referme pour réguler la quantité de lumière qui va pénétrer dans l'œil. Troisièmement, le cristallin, en se déformant, joue le rôle d'une lentille biologique et permet la mise au point (ou accommodation) sur les objets et la formation d'une image inversée sur la rétine. Enfin, cette rétine, localisée au fond du globe oculaire, transforme les signaux lumineux en influx nerveux et les achemine au cortex visuel via des cellules spécialisées.

3.1.2 La rétine

La rétine est constituée d'environ 150 millions de cellules nerveuses pouvant être décomposées en deux couches. La couche plexiforme externe comprend les photorécepteurs et les cellules horizontales, la couche plexiforme interne comprends les cellules bipolaires, amacrines et ganglionnaires. La figure 3.3 présente une coupe transversale de la rétine sur laquelle ces couches sont visibles. Notons qu'une des particularités de la rétine des vertébrés est que, contrairement à ce que l'intuition nous dicte, les cellules photoréceptrices sont situées au fond de la rétine ; la lumière doit donc traverser les cellules nerveuses avant d'être captée par les photorécepteurs.

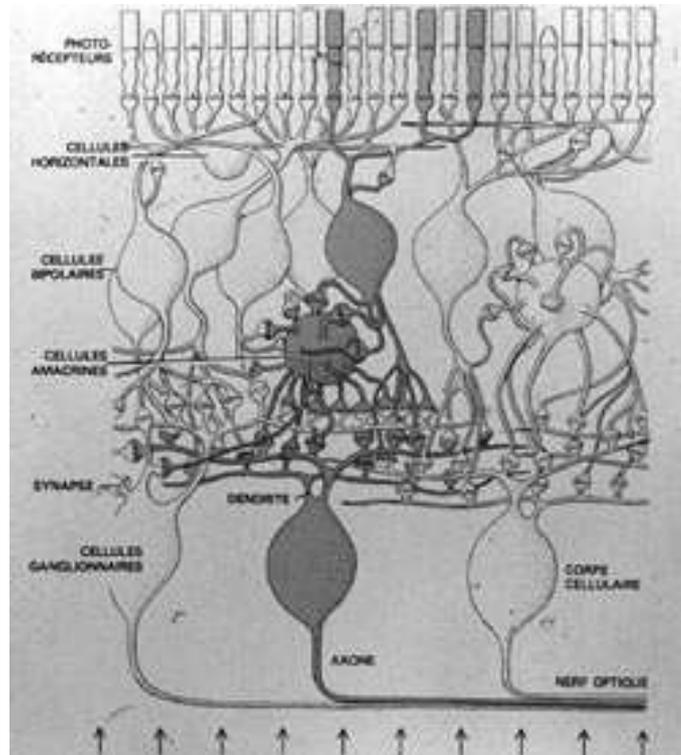


FIG. 3.3 – Coupe transversale de la rétine

Les photorécepteurs se composent de deux classes, les bâtonnets et les cônes, lesquels se redécomposent en trois types S(hort), M(edium) et L(ong). Les cônes permettent la vision des couleurs, chaque type de cône est sensible à une plage de longueur d'onde bien déterminée. Ils sont essentiellement situés dans la zone centrale de la rétine, la fovéa, correspondant à environ 2 degrés de vision, et ne sont actifs qu'à partir d'un certain niveau de luminance correspondant approximativement à une vision diurne (ou photopique). Les bâtonnets ne sont pas sensibles aux couleurs. Il ne sont que très peu présents dans la fovéa mais ils abondent dans une zone périphérique s'étendant à 20 degrés de vision. Les bâtonnets sont responsables de la vision nocturne (ou scotopique). Notons que la fovéa comporte une zone centrale, la fovéola, dans laquelle il n'y a pas de cône S ni de bâtonnet.

Que se passe-t-il lorsque une information lumineuse arrive sur la rétine ? Lorsque ce stimulus atteint un bâtonnet, il provoque un changement d'état électrique. En effet, les bâtonnets contiennent de la rodopsine, molécule qui a la faculté de changer de configuration en présence de lumière, modifiant le potentiel de la cellule qui va ainsi propager un influx nerveux. Cette dernière retourne ensuite progressivement à son état initial. Le mécanisme photoélectrique des cônes est très similaire, les substances chimiques mises en jeu spécifiant le comportement de chaque type de cône.

Chaque cellule photoréceptrice produit donc un influx nerveux. Sachant que leur densité peut atteindre $150000/mm^2$ au centre de la fovéola (zone centrale de la fovéa), un mécanisme de regroupement / compression de ces signaux est nécessaire. C'est le rôle des cellules horizontales, bipolaires et endocrines. Comme le montre la figure 3.3, les photorecepteurs délivrent leurs signaux aux cellules horizontales, responsables d'un couplage transversal des signaux, et aux cellules bipolaires. Ces dernières assurent la transmission aux cellules ganglionnaires sous la médiation des cellules amacriques qui diffusent ou accentuent l'information. Le fait que l'on passe de 105 millions de cellules

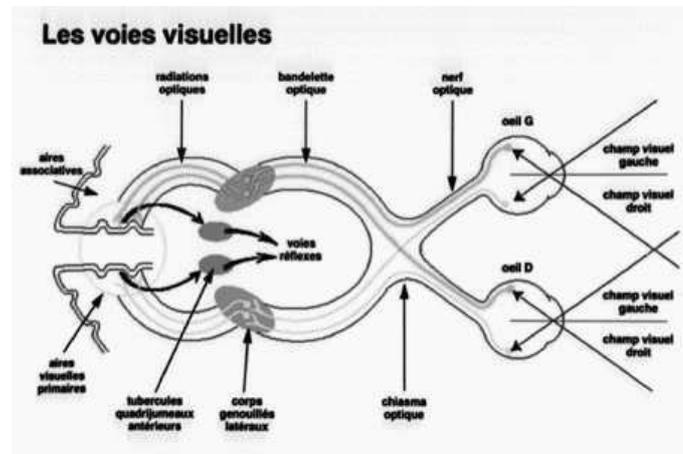


FIG. 3.4 – Les voies nerveuses du système visuel

réceptrices à environ 1,2 millions de cellules bipolaires donne une idée de la compression du signal lumineux lors de sa transformation en influx nerveux. Cette compression n'est toutefois pas uniforme. Dans la zone fovéale, où une grande précision est requise, elle est même inexistante : chaque cône est directement connecté à des cellules bipolaires ON et OFF qui lui sont propres. Les bâtonnets, par contre, sont peu présents dans la zone fovéale. Par conséquent, quinze bâtonnets sont connectés à une seule cellule bipolaire. Les cellules ganglionnaires, dont les axones forment le nerf optique, vont ensuite acheminer le signal à travers les voies nerveuses.

3.1.3 Les voies nerveuses et le cortex visuel

La figure 3.4 montre que les nerfs optiques issus des deux yeux se rejoignent et s'entrecroisent au niveau du chiasma, pour ensuite atteindre les corps genouillés latéraux. Ces derniers jouent un rôle de relais amplificateur et dirigent ensuite les connexions nerveuses vers des aires spécialisées sur le cortex visuel de l'hémisphère cérébral correspondant. La répartition des fibres sur le cortex permet de reproduire le champ visuel sur la couche corticale. La stimulation d'une petite zone corticale entraîne l'apparition d'un point lumineux dans le champ visuel.

Le cortex visuel se divise en trois zones, une zone d'intégration visuelle primaire (l'aire 17), et deux zones associatives (aires 18 et 19). Le fonctionnement du cortex visuel dépasse de très loin le cadre de ce manuscrit de thèse, mais nous pouvons tout de même signaler que l'aire 17 intègre et transmet les informations aux aires 18 et 19, lesquelles se décomposent en bandes chargées individuellement de la reconnaissance des formes, des couleurs, de la stéréoscopie et de la détection des mouvements.

3.2 Les caractéristiques du système visuel

Cette section a pour but de présenter les capacités du système visuel : en effet, après avoir brièvement détaillé son anatomie, nous allons maintenant voir comment ses composants s'organisent pour nous permettre de percevoir correctement notre environnement dans des conditions très diverses.

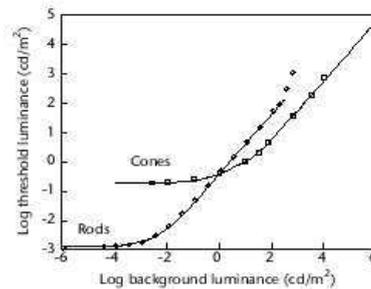


FIG. 3.5 – Limite de détection de différence de luminance en fonction de la luminance du fond

3.2.1 L'adaptation visuelle

Description du phénomène

Tout le monde a fait au moins une fois l'expérience de passer d'un jardin ensoleillé le midi en plein été à une cave sombre : après une brève période où l'on ne distingue pas grand chose, le système visuel s'adapte et on peut voir distinctement l'intérieur de notre cave. C'est là une des remarquables caractéristiques du système visuel : son incroyable faculté d'adaptation. Il faut en effet savoir que la lumière d'une journée ensoleillée peut être jusqu'à dix millions de fois plus intense que celle d'un clair de lune. Pourtant, l'œil se sent à l'aise dans ces deux situations extrêmes ; ses capacités lui permettent en effet de gérer jusqu'à 14 ordres de grandeurs en luminance. Toutefois, ses performances ne sont pas les mêmes selon le domaine dans lequel s'effectue la visualisation :

De -6 à $-2 \log Cd/m^2$ (vision de nuit ou scotopique), seuls les bâtonnets sont actifs. Ils sont très sensibles aux petites différences de luminances mais les couleurs ne sont pas perçues (d'où la célèbre maxime "la nuit, tous les chats sont gris"). Du fait de la localisation excentrée des bâtonnets dans la rétine, l'acuité visuelle est faible et les petits détails sont peu visibles.

De 2 à $8 \log Cd/m^2$ (vision photopique), les cônes permettent la vision des couleurs. Ils sont très présents dans la zone fovéale, l'acuité visuelle est donc élevée. Ils sont par contre beaucoup moins sensibles que les bâtonnets, les faibles différences de luminances sont donc difficilement perceptibles. C'est une des raisons pour laquelle les étoiles sont invisibles en plein jour.

De -2 à $2 \log Cd/m^2$ (vision mésopique). Ce domaine est assez mal connu. Les quatre types de récepteurs fonctionnent simultanément. Il existe très peu, à l'heure actuelle, de courbes de sensibilité mésopique. Contrairement à ce que l'on pourrait croire, il est relativement courant de se retrouver dans une situation de vision mésopique : au cinéma, en conduisant de nuit,...

Fonctionnement physiologique

Plusieurs composants entrent en jeu dans le mécanisme complexe de l'adaptation. Premièrement, l'iris joue un rôle régulateur en limitant l'entrée de lumière dans le globe oculaire. Deuxièmement, nous avons vu que les photorécepteurs fonctionnaient grâce à une réaction photochimique ; or si la régénération devient plus lente que la réaction elle-même (conséquence directe d'une forte luminance dans la scène), le système visuel devient moins sensible à la lumière. Ces deux actions ne couvrent pourtant qu'un ou deux ordres de grandeur et sont très insuffisantes pour justifier à elles seules la capacité d'adaptation.

En réalité, la majeure partie de ce travail s'effectue au niveau des cellules des couches plexiformes externes et internes. Le rôle des cellules horizontales, ganglionnaires et bipolaires est ici de modérer

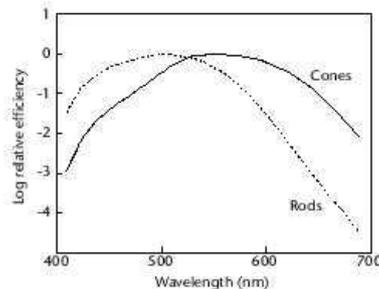


FIG. 3.6 – Sensibilité spectrale dans le domaine photopique

ou d'amplifier l'intensité du stimuli arrivant sur ces photorecepteurs. En effet, lorsqu'un récepteur est proche de la saturation, c'est à dire que la réaction photochimique est proche de son intensité maximale, il ne peut plus rendre compte d'une augmentation de la luminance, aussi petite soit elle. De même, si le signal de départ est plus faible que le seuil de perception du capteur, il ne sera pas traité non plus. Les cellules ganglionnaires agissent alors comme un multiplicateur sur l'entrée des récepteurs : le stimuli lumineux est modulé par une constante qui dépend de la luminance moyenne perçue, compensant ainsi les limites de détection des récepteurs.

Notons enfin que l'adaptation n'est pas un phénomène instantané : elle nécessite de plusieurs secondes à plusieurs minutes de transitions entre deux états radicalement différents. La transition lumineux \rightarrow sombre est plus lente que la transition sombre \rightarrow lumineux.

Quelques courbes psychométriques

Pour prendre en compte l'adaptation visuelle dans un modèle de vision, il nous faut des mesures psychovisuelles de ce phénomène. Les courbes suivantes résument les performances du système visuel en matière d'adaptation et sont basées sur des expériences de seuils de perception. Toutes ces figures sont issues de [18].

La première courbe (figure 3.5) traduit la plus petite différence de luminance perceptible en fonction de la luminance du fond. Notons que l'on considère ici que le fond est uniforme et achromatique. On dénote plusieurs points intéressants sur cette courbe :

- Les bâtonnets sont beaucoup plus sensibles que les cônes dans le domaine scotopique.
- Les cônes commencent à percevoir les différences aux alentours de $-2 \log Cd/m^2$, ce qui correspond bien au début du domaine mésopique.
- Les bâtonnets ne sont plus actifs dans le domaine photopique.
- Les deux courbes présentent une zone linéaire en log (loi de Weber), ce qui simplifie grandement l'obtention d'une fonction psychométrique.

La deuxième courbe (figure 3.6) montre la sensibilité spectrale des cônes et bâtonnets dans le domaine photopique. La troisième courbe (figure 3.7) est un prolongement de la figure 3.6 en fonction de la luminance de fond et donc du domaine de visualisation considéré. Il est visible, entre autres, que lors de la transition scotopique / mésopique, ce sont les fréquences proches du rouge qui apparaissent en premier.

La quatrième courbe représente l'évolution du seuil de luminance perceptible en fonction du temps dans le cas d'une transition état photopique vers état scotopique. Il est visible sur la figure 3.8 que cette adaptation est loin d'être instantanée. Nous pouvons également remarquer que les bâtonnets sont moins rapides à s'adapter puisqu'il faut attendre 7 minutes environ pour que leur sensibilité surpasse

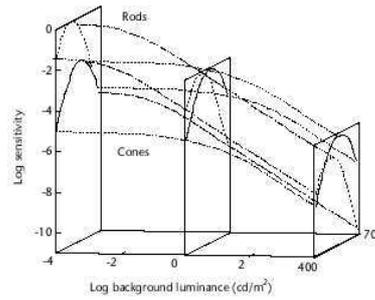


FIG. 3.7 – Sensibilité spectrale des différents capteurs en fonction de la luminance du fond

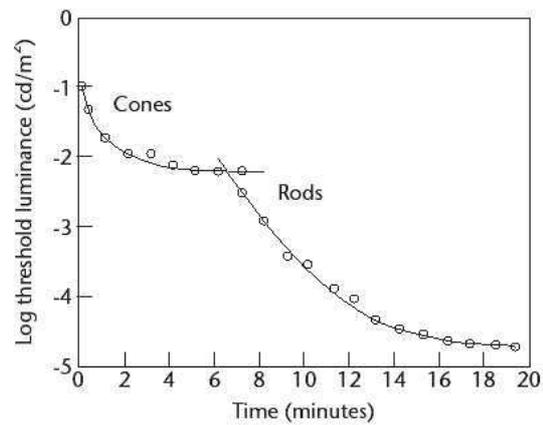


FIG. 3.8 – Seuil de luminance perceptible en fonction du temps pour chaque type de récepteur

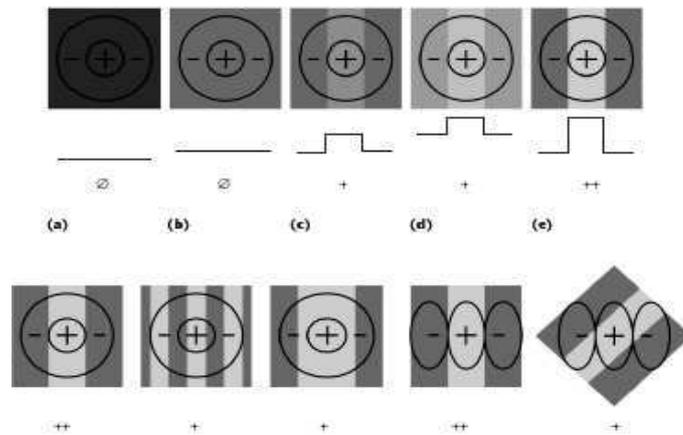


FIG. 3.9 – Réponse des cellules ganglionnaires pour différents types de stimuli

celle des cônes.

3.2.2 La vision des contrastes

Par vision des contrastes, nous entendons les capacités du système visuel à percevoir et reconnaître des motifs lumineux. Ces capacités varient en fonction du domaine de vision comme nous l'avons vu dans la section précédente. En effet, en vision scotopique, seuls les bâtonnets sont actifs et ils sont totalement absents de la fovéa ; la vision est donc floue. Ce n'est néanmoins pas le seul critère important dans ce domaine.

Le rôle de la rétine et des voies nerveuses

Il a été mis en évidence qu'à chaque cellule ganglionnaire dans la rétine correspond un champ récepteur dans l'espace visuel. Ce champ peut être chromatique ou achromatique et se décompose en une partie centrale et une partie périphérique. Ces champs fonctionnent de manière antagoniste, c'est à dire que la stimulation des cellules correspondantes ne se fait que dans le cas d'une augmentation de la différence d'intensité entre le centre et la périphérie du champ récepteur ; la réponse maximale de la cellule est alors obtenue lorsque le stimulus coïncide avec cette géométrie. Les champs récepteurs n'ayant pas tous la même taille (certains sont beaucoup plus petits que d'autres), chaque cellule ganglionnaire possède donc son pic de sensibilité ajusté sur une fréquence spatiale bien précise. Enfin, les champs récepteurs étant orientés, les cellules répondent plus favorablement si le stimulus est orienté dans la même direction. La figure 3.9, issue de [18], illustre ces propos en représentant le champ récepteur et l'intensité de la réponse de la cellule pour différents stimuli.

Mesures psychométriques de la vision des contrastes

Ces mesures sont comparables à celles effectuées pour l'adaptation ; là encore, il s'agit d'évaluer des seuils perceptuels. La figure 3.10, extraite de [18] montre les fonctions de sensibilité au contraste (FSC) chromatique (symbole carré) et achromatique (symbole rond). Elles représentent le contraste minimal discernable en fonction de la fréquence spatiale. Ces fonctions sont obtenues en demandant à un observateur humain de repérer la limite de perception d'un motif sinusoïdal mono-dimensionnel projeté sur un écran en face de lui en augmentant progressivement l'intensité de ce motif, et ce pour

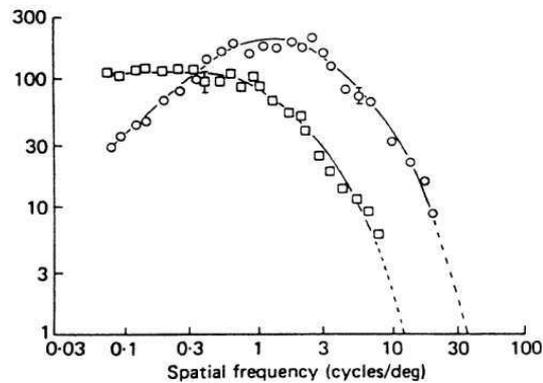


FIG. 3.10 – Fonction de sensibilité au contraste chromatique et achromatique

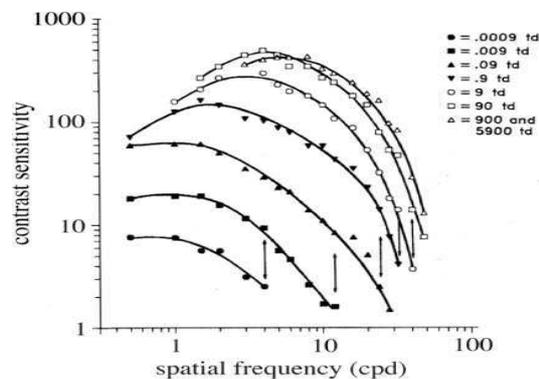


FIG. 3.11 – Évolution de la fonction de sensibilité au contraste selon la luminance de fond

une gamme de fréquences spatiales allant jusqu'à quelques dizaines de cycles par degré de vision. Selon le type de la fonction mesurée, le motif projeté est monochromatique ou isoluminant. Remarquons d'emblée sur la courbe 3.10 que la FSC achromatique est passe bande alors que la FSC chromatique est passe bas.

La fonction de sensibilité au contraste évolue en fonction du niveau d'adaptation en luminance de l'œil ; ce phénomène est une conséquence logique de l'évolution des capacités de résolution spatiale avec le niveau de luminance moyen de la scène évoquée dans la section 3.2.1. La figure 3.11 donne plusieurs FSC à des niveaux d'illumination différents.

La FSC peut être obtenue en deux dimensions, ce qui est intéressant quand on travaille dans le domaine de l'image. Représentée sur la figure 3.12, issue de [11], elle est assez peu différente de la FSC 1D, mais on notera tout de même une légère dépression au centre de la fonction, ce qui tend à montrer que le système visuel humain est moins réceptif aux contrastes diagonaux.

D'après ces mesures, il serait permis de supposer que le système visuel se comporte comme un filtre fréquentiel défini par la FSC. La reconnaissance d'un objet pourrait être alors accomplie d'après la sensibilité aux différentes composantes fréquentielles de cet objet. Malheureusement, l'expérience ne corrobore pas ces hypothèses. Il semble au contraire que toutes ces composantes fréquentielles soient détectées indépendamment. Enroth-Cugell et Robson ont postulé dans [16] l'existence d'une série de détecteurs, chacun étant sensible à une plage de fréquences distincte. Cette hypothèse rejoint l'existence des champs récepteurs de taille différente de la section 3.2.2. Ainsi, comme le montre la figure 3.13, la CSF ne serait ni plus ni moins que l'enveloppe des courbes de sensibilités fréquentielles

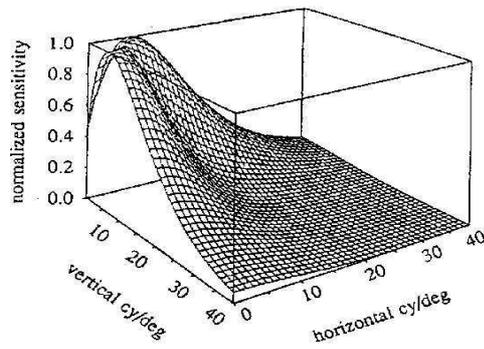


FIG. 3.12 – Fonction de sensibilité au contraste bidimensionnelle

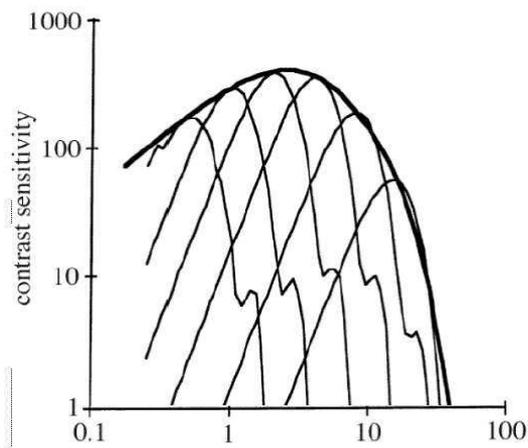


FIG. 3.13 – Fonctions de sensibilité fréquentielle des champs récepteurs

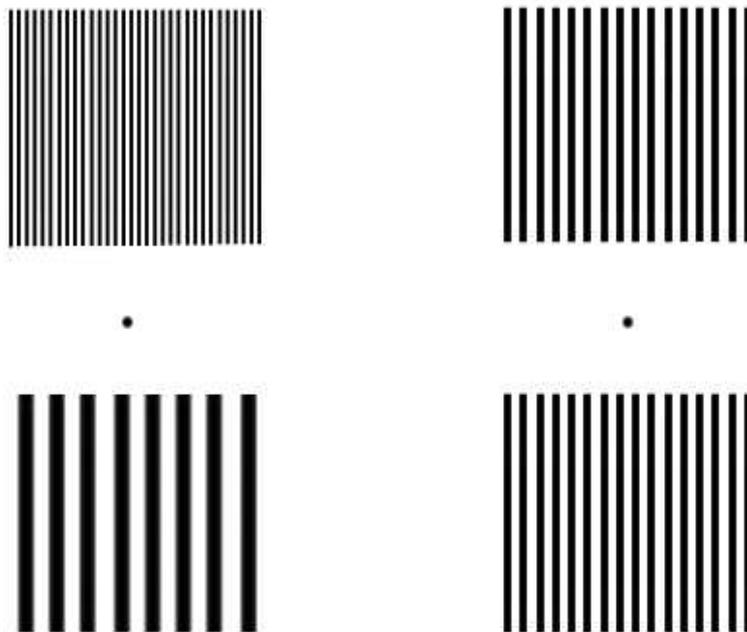


FIG. 3.14 – Adaptation en contraste

de ces récepteurs. Il est intéressant de constater que toutes ces courbes peuvent être formulées comme une différence de gaussiennes ou une fonction de Gabor.

Signalons pour finir que, d’une manière similaire à l’adaptation en luminance, l’œil humain peut également s’adapter à un contraste. Le lecteur intéressé peut se prêter à l’expérience proposée sur la figure 3.14. La partie droite de la figure montre deux motifs dont la fréquence spatiale est identique. Pour mettre en évidence l’adaptation, il faut cacher ces motifs, puis fixer la partie gauche de la figure au niveau du point central pendant environ une minute. Une fois l’adaptation effectuée, les deux motifs de la partie droite ne paraissent plus avoir la même fréquence spatiale. Les courbes présentées plus haut sont obtenues en supposant qu’il n’y a pas d’adaptation sur un contraste particulier.

Le masquage

Le masquage est l’interaction mutuelle des différents contrastes d’une image sur leurs perceptions respectives par le système visuel. Concrètement, la présence sur une image d’un ou plusieurs contrastes dits “masquants” peut perturber la visibilité d’un autre contraste, dit “masqué”. La courbe 3.15 donne le seuil de perceptibilité du contraste masqué en fonction du contraste masquant. La figure 3.16 montre un exemple de masquage. L’image de gauche comprend un motif avec de très hautes fréquences spatiales sur un fond blanc uniforme. L’image de droite est identique à ceci près qu’un bruit aléatoire a été ajouté. Le point intéressant ici est que le bruit, nettement perceptible sur le fond uniforme, l’est beaucoup moins sur le motif car masqué par les hautes fréquences de ce dernier.

3.2.3 La vision des couleurs

A l’heure actuelle, le mécanisme de la perception des couleurs est encore méconnu. Il est d’autant plus difficile à cerner que ses origines sont diverses, le phénomène est en effet à la fois physique,

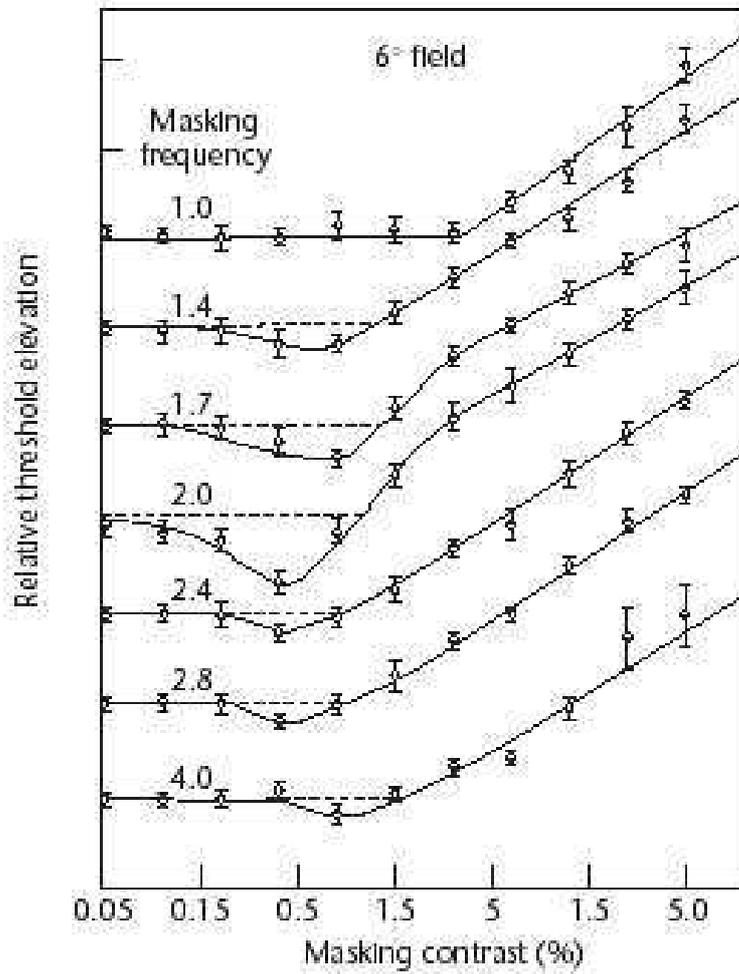


FIG. 3.15 – Le phénomène de masquage

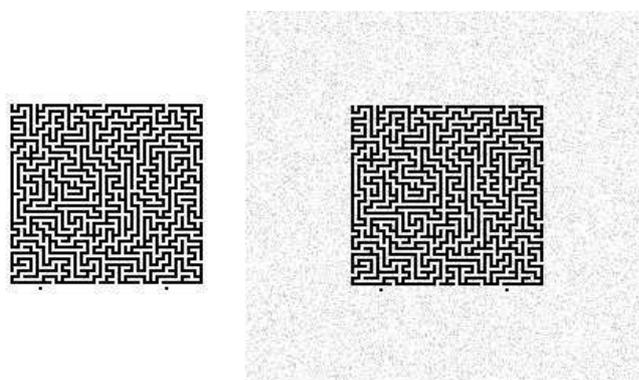


FIG. 3.16 – Exemple de masquage

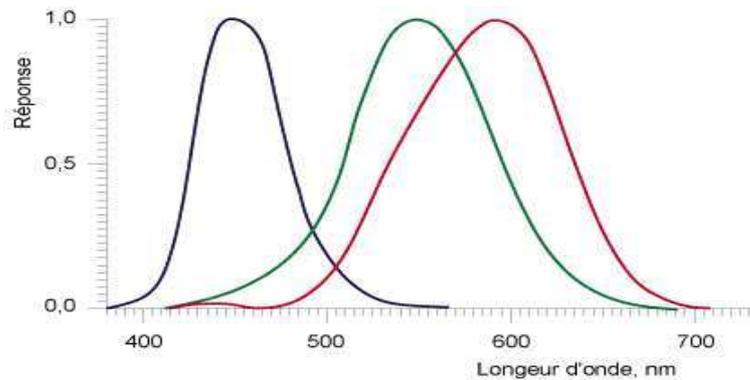


FIG. 3.17 – Sensibilité spectrale des cônes

physiologique et psychologique :

- Physique, car il dépend de la distribution spectrale de puissance de la lumière perçue (nous aborderons les termes radiométriques dans le chapitre suivant).
- Physiologique, car il dépend aussi du bon fonctionnement du système visuel de l'observateur : par exemple, il peut lui manquer un certain type de cône (daltonisme) : sa vision des couleurs en sera alors altérée.
- Psychologique enfin, car la perception des couleurs est avant tout une expérience subjective. Dans certains cas, elle peut même avoir lieu en l'absence totale de stimuli lumineux (lors de rêves, par exemple).

On estime qu'un observateur moyen est capable de distinguer une centaine de milliers de couleurs différentes. Dans cette section, nous ne nous préoccupons que de l'aspect psychovisuel de la couleur ; la mesure de la couleur et la construction des espaces colorimétriques feront partie de l'objet du chapitre 4.

Physiologie de la vision des couleurs

Comme nous l'avons vu, les cellules de la rétine responsables de la vision des couleurs sont les cônes. Lorsqu'un photon arrive sur la rétine, sa probabilité d'être absorbé par le cône sur lequel il "tombe" est dépendante de la fonction de sensibilité spectrale de ce dernier. La figure 3.17 montre ces fonctions pour les trois types de cônes. Un point important : si le photon est absorbé par la cellule, alors la réponse produite dans la cellule est la même quelle que soit la longueur d'onde du photon (principe d'univariance). Il est donc tout à fait possible que deux spectres différents produisent la même sensation colorée. Ce phénomène est connu sous le nom de métamérisme.

Nous avons aussi vu que la rétine n'est pas comparable à un capteur CCD : son plan ne représente absolument pas directement l'image telle qu'elle est perçue, les cellules des couches plexiformes effectuant un important multiplexage de ces signaux. En ce qui concerne la perception de la couleur, le travail de ces couches consiste à transformer le signal à trois composantes spectrales transmis par les cônes en trois signaux antagonistes : une opposition rouge - vert, une opposition bleu - jaune et une opposition achromatique. Ce sont ces mêmes oppositions que l'on retrouve dans les champs récepteurs. Les interactions entre les différents canaux sont très nombreuses et trop complexes pour être exposées dans ce mémoire, mais le lecteur intéressé peut se reporter à [66]. Signalons pour finir que, étant donné que le globe oculaire est un dispositif optique, l'image qu'il forme est soumise à une aberration chromatique. Ce phénomène est dû au fait que l'indice d'un milieu dépend de la

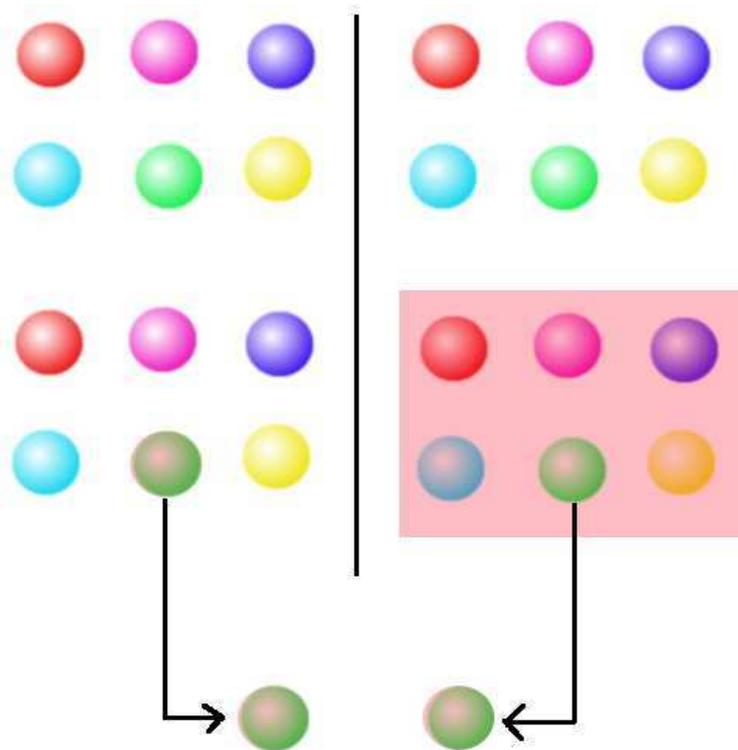


FIG. 3.18 – Adaptation chromatique

longueur d'onde du rayonnement qui le traverse. Ainsi, pour une lumière multispectrale, toutes les composantes ne sont pas déviées du même angle. En ce qui concerne un système visuel humain correctement corrigé, les rayonnements rouges convergent en avant de la rétine, et les rayonnements bleus en arrière. L'écart de mise au point nécessaire étant significatif, il y a gêne visuelle quand deux zones contigües de l'image appartiennent à des domaines spectraux éloignés : par exemple, il est plus difficile de lire un texte bleu sur un fond rouge que sur un fond vert.

L'adaptation chromatique, ou constance couleur

Un phénomène intéressant de la vision des couleurs est la constance couleur. La figure 3.18 donne un aperçu de cette caractéristique. La sphère verte de gauche semble rougeâtre par rapport à celle de droite, mais ce sont pourtant exactement les mêmes. Nous avons ici une mise en évidence du fait que le système visuel repère les couleurs relativement par rapport au contexte, et non de manière absolue. De la même façon, un illuminant coloré sur l'ensemble d'une scène ne perturbe pas la reconnaissance des couleurs : si nous plaçons un filtre rouge devant une lampe à incandescence éclairant une scène, les objets bleus de la scène nous apparaîtront toujours comme bleus, alors que leur couleur mesurée indépendamment du contexte serait plus proche de l'indigo. Si nous considérons le fait que la couleur est perçue et analysée en termes d'antagonismes, donc de différences relatives, cette constatation n'est pas fondamentalement surprenante.

Le mécanisme de la transparence est relativement similaire. Si, sur une scène, nous délimitons une zone à l'intérieur de laquelle les coordonnées rgb subissent une translation uniforme, alors l'illusion est donnée d'une surface transparente posée sur la scène. La figure 3.19 montre une illustration de ce phénomène. La figure de gauche représente un patchwork de couleurs. Sur la figure centrale, nous

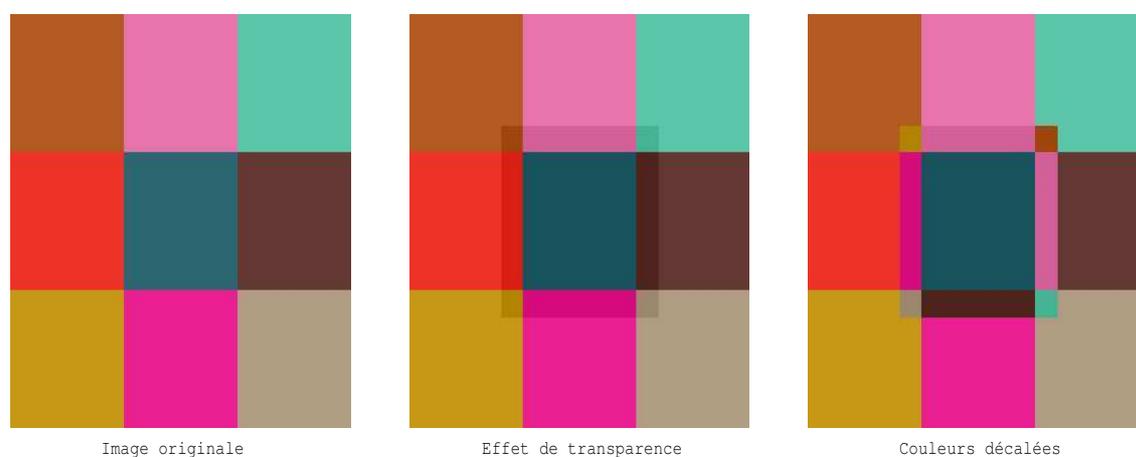


FIG. 3.19 – Perception de la transparence.

avons créé une zone centrale sur ce patchwork dans laquelle le vecteur des coordonnées colorimétriques a subi une translation uniforme. Sur la figure de droite, nous avons simplement interverti, au sein de cette zone, les couleurs correspondantes de la figure centrale. Nous constatons une sensation de transparence très nette sur la figure centrale, ce qui n'est pas du tout le cas de la figure de droite qui comporte pourtant exactement les mêmes couleurs. Cette expérience simple prouve que la perception de la transparence est aussi une affaire d'agencement géométrique.

Les contrastes couleur

La notion même de contraste couleur est assez mal définie à l'heure actuelle. Au sens strict du terme, cela signifierait la capacité à distinguer deux motifs isoluminants. On peut se demander si cette opération ne pourrait pas être étendue à la différenciation de deux couleurs, indépendamment de la géométrie. Ceci relève de la colorimétrie, une bonne partie du chapitre 4 y est consacrée.

Toutefois, nous pouvons soulever ici le problème de la perception des couleurs en fonction du contexte géométrique. Un premier exemple est donné sur la figure 3.20. Deux motifs sont présentés sur cette figure : les deux carrés centraux sont parfaitement identiques mais le fond est différent, ce qui induit une différence de perception au niveau des couleurs des carrés centraux. Ce phénomène est connu sous le nom de contrastes simultanés. Le phénomène disparaît si l'on présente séparément aux deux yeux le carré central et le fond. Nous pouvons en conclure que les traitements qui en sont responsables ont lieu avant le chiasma, qui est la section du système visuel où les informations des deux yeux se combinent.

Un deuxième exemple est présenté sur la figure 3.21. La gauche de la figure met en évidence un phénomène curieux : sur cette grille, nous avons partiellement coloré une partie des lignes de manière à représenter un disque. L'intérieur de la grille à cet endroit est perçu comme étant rose, alors qu'il est parfaitement blanc. Ce n'est plus le cas sur la partie droite : même si le cercle est toujours visible, il ressemble plutôt à un amas de taches bien rouges. Le fort contraste entre la ligne rouge et le fond blanc n'est donc pas le seul responsable de l'assimilation, la régularité de la grille joue aussi pour beaucoup dans ce phénomène. Il semble que l'effet d'assimilation est significativement réduit dès que l'on perturbe cette structure régulière, comme sur la partie droite de la figure.

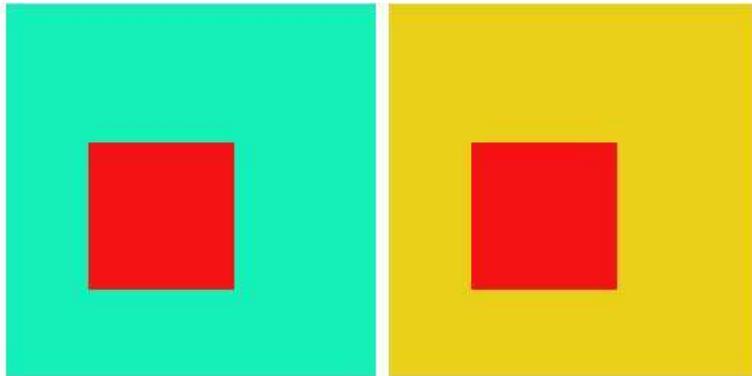


FIG. 3.20 – Premier exemple de contraste couleur

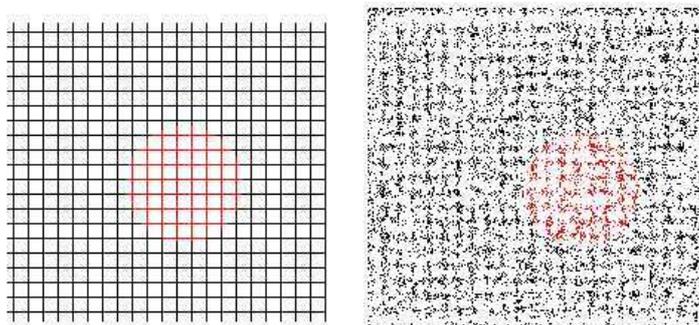


FIG. 3.21 – Deuxième exemple de contraste couleur

3.3 Réflexions

Le principal problème que nous allons nous poser maintenant est le suivant : comment les études psychovisuelles présentées dans les sections précédentes peuvent-elles être exploitées en informatique graphique ? Pour ce qui est des aspects achromatiques, reprendre les courbes existantes paraît être une bonne idée. Mais il existe un problème de taille : la dynamique en luminance d'un écran n'a aucune commune mesure avec celle d'une scène réelle. Pour ce qui est des aspects couleur, il n'existe pas énormément d'études pouvant se traduire par une courbe psychométrique. Il y a plusieurs raisons à cet état de fait. Premièrement, la couleur est une notion partiellement subjective, chacun en a sa propre perception. Deuxièmement, il ne s'agit pas d'une grandeur aussi facilement mesurable qu'une luminance. Comment dans ces conditions obtenir un modèle crédible de la vision ? Dans le chapitre suivant, nous allons voir quelles sont les solutions actuellement disponibles pour palier ces inconvénients.

Chapitre 4

Couleur, Reproduction de tons et Fidélité

Nous avons vu dans le chapitre précédent les principales caractéristiques du système visuel humain. Il nous faut maintenant appliquer ces connaissances au domaine de l'informatique graphique. Ceci nous impose un certain nombre de travaux préliminaires : comment définir physiquement les grandeurs perçues intuitivement comme l'éclairement ou l'intensité d'une source lumineuse ? Comment faire ensuite pour transcrire ces grandeurs sur un dispositif informatique de manière la plus fidèle possible ? La couleur est aussi sujette à de nombreuses interrogations. Comment disposer d'un système de codage fiable de la couleur, sachant qu'il doit offrir un juste compromis entre la fidélité de la reproduction et la taille des données nécessaires ? Peut-on reproduire toutes les couleurs discernables par le système visuel ? Ce chapitre se propose de poser le problème général de la couleur et de la fidélité en synthèse d'images.

4.1 Notions essentielles de radiométrie et de photométrie

Quelques notions de radiométrie et de photométrie sont nécessaires pour la compréhension des notions abordées dans ce manuscrit. Les définitions suivantes seront très brèves. Pour un exposé plus complet sur ces domaines, le lecteur intéressé peut consulter [65].

4.1.1 Radiométrie

En toute rigueur, la colorimétrie étudie les couleurs de champs lumineux sans forme et non localisés. Il nous faudra donc en premier lieu mesurer physiquement la puissance lumineuse. C'est le but de la radiométrie. Les unités radiométriques essentielles sont les suivantes :

- **Luminance** : flux émis par une source lumineuse rapporté à l'angle solide d'émission et à l'aire apparente de la surface émettrice, exprimée en $W.m^{-2}.sr^{-1}$
- **Intensité** : flux émis par une source lumineuse rapporté à l'angle solide d'émission, exprimée en $W.sr^{-1}$
- **Exitance** : flux émis par une source lumineuse rapporté à l'aire apparente de la surface émettrice, exprimée en $W.m^{-2}$
- **Éclairement** : flux reçu par une surface rapporté à l'aire éclairée, exprimé en $W.m^{-2}$

4.1.2 Photométrie

La photométrie est le pendant perceptuel de la radiométrie. Elle permet de mesurer les sensations visuelles liées à un stimulus lumineux. Le lien avec la radiométrie est réalisé grâce aux courbes de

sensibilité $V(\lambda)$ et $V'(\lambda)$ du système visuel. La luminance “photométrique” (appelée en réalité luminance visuelle), dans des conditions de visualisation photopique, s’obtient de la manière suivante :

$$L_v(\lambda) = K_m \cdot V(\lambda) \cdot L_e(\lambda)$$

K_m est l’efficacité lumineuse de la source, L_e est la luminance énergétique. Ainsi, si deux sources monochromatiques produisent la même sensation visuelle, leurs luminances visuelles respectives sont alors identiques mais leurs luminances énergétiques peuvent très bien être différentes.

La photométrie fait intervenir deux unités fondamentales :

– **Lumen** : unité permettant de mesurer le flux lumineux. Un lumen (lm) est équivalent à une puissance de valeur $1/683$ Watt pour une radiation monochromatique de longueur d’onde $555,016$ nanomètres.

– **Candéla** : unité du système international pour l’intensité lumineuse. 1 candéla (cd) vaut $1 \text{ lm} \cdot \text{sr}^{-1}$. Par analogie avec la radiométrie, on peut définir les grandeurs photométriques suivantes :

– **Luminance visuelle** : Flux lumineux émis par une source lumineuse rapporté à l’angle solide d’émission et à l’aire apparente de la surface émettrice, exprimée en $\text{lm} \cdot \text{m}^{-2} \cdot \text{sr}^{-1}$ ou $\text{cd} \cdot \text{m}^{-2}$.

– **Intensité visuelle** : Flux lumineux émis par une source lumineuse rapporté à l’angle solide d’émission, exprimée en candéla (cd).

– **Éclairement visuel** : Flux lumineux reçu par une surface rapporté à l’aire éclairée, exprimé en $\text{lm} \cdot \text{m}^{-2}$ ou lux (lx).

Quelques luminances visuelles usuelles en $\text{cd} \cdot \text{m}^{-2}$ (d’après [65]) :

- Ciel sans lune : 10^{-4}
- Moniteur d’ordinateur : environ 80
- Objet plan mat, soleil au zénith, temps couvert : 10000
- Disque solaire : $1,5 \cdot 10^9$

4.2 Bases de colorimétrie

La colorimétrie est la science de la mesure des couleurs. Par mesure des couleurs, nous entendons le fait de pouvoir quantifier et coder n’importe quelle couleur perceptible. Dans le langage courant, cela se limite la plupart du temps à une teinte de base (rouge, vert, bleu, violet, jaune . . .), assortie d’un adjectif quantitatif (par exemple : foncé, clair, saturé . . .) et/ou d’un nom commun se référant à un objet (jaune “citron” ou rouge “tomate”). Pour de nombreux domaines dont la couleur est une composante importante, comme les arts plastiques par exemple, le langage est donc souvent l’outil de description usuel. Toutefois, ce système “artisanal” montre bien vite ses limites : il n’est pas toujours possible de différencier deux nuances très proches (il existe plus d’un “vert foncé”) sans intervention humaine. Des lors, la spécification précise d’une couleur devient vite cauchemardesque (exemple : “vert avocat mais plus foncé, avec une légère tendance sur le rouge, le tout plutôt dans des tons pastels”) et sera de toute façon interprétée subjectivement par la personne à qui la description est destinée. Il est donc nécessaire de mettre sur pied un système rigoureux de spécification de la couleur.

4.2.1 Hypothèses de base de la colorimétrie

L’objectif de la colorimétrie est la mesure de la couleur. Dans le chapitre consacré au système visuel, nous avons vu que la perception de la couleur est due en premier lieu à trois types de cellules nommées les cônes. La représentation de la couleur par trois composantes paraît donc la plus évidente. C’est l’hypothèse de trichromaticité. Young[85] émit cette hypothèse le premier en 1802 :

Prenant la suite de Young, Maxwell[43] démontra qu'une sensation de couleur peut être représentée par la réponse conjointe des trois capteurs. Selon cette hypothèse, n'importe quelle couleur peut être reproduite par combinaison linéaire de trois couleurs primaires indépendante entre elles (c'est à dire qu'un primaire ne peut être obtenue par combinaison linéaire des deux autres). Avec trois primaires P_1 , P_2 et P_3 , il est donc possible de reconstruire un stimulus S par additivité :

$$S = x_1.P_1 + x_2.P_2 + x_3.P_3$$

Rappelons qu'ici que l'égalisation est purement visuelle : elle est due au phénomène de métamérisme, déjà évoqué dans le chapitre précédent. Il est évident qu'ajouter 3 composantes primaires ne permet pas de reconstituer tous les spectres possibles. Il faut également faire remarquer que les quantités x_i peuvent dans certains cas être négatives. En pratique, cela revient à ajouter une quantité x_i de la primaire P_i au stimulus que l'on veut égaliser.

Les résultats des expériences de Young et Maxwell montrent également que les égalisations couleur obéissent aux lois de proportionnalité et d'additivité. Ce sont les lois de Grassmann :

- Si un stimulus Q_1 est visuellement équivalent à un stimulus Q_2 , et si α est un facteur de proportionnalité, alors $\alpha.Q_1$ est visuellement équivalent à $\alpha.Q_2$.
- Si en plus on a deux stimuli Q'_1 et Q'_2 visuellement équivalents, alors $Q_1 + Q'_1$ est visuellement équivalent à $Q_2 + Q'_2$.

En choisissant trois primaires, il est donc possible de former un espace vectoriel des couleurs, que nous appellerons par la suite espace colorimétrique. Chaque couleur y est représentée par un vecteur à trois composantes. Étant donné que le choix des primaires est libre, on peut par exemple prendre le pic de sensibilité de chaque type de cône.

4.2.2 Les fonctions colorimétriques

La fonction colorimétrique $p_i(\lambda)$ d'une primaire P_i donne la quantité x_i nécessaire pour recréer le stimulus $S(\lambda)$ associé à une lumière monochromatique de longueur d'onde λ . Pour une radiation monochromatique S de longueur d'onde λ_0 , nous avons donc :

$$S = p_1(\lambda_0).P_1 + p_2(\lambda_0).P_2 + p_3(\lambda_0).P_3$$

Les lois de Grassmann, notamment l'additivité, permettent d'ajouter entre eux des stimuli de longueurs d'ondes monochromatiques. Par extension au continu, il est alors possible de reconstituer le stimulus lié à n'importe quelle distribution spectrale $DSP(\lambda)$. On aura alors

$$S = x_1.P_1 + x_2.P_2 + x_3.P_3$$

avec :

$$\begin{aligned} x_1 &= \int_{i=0}^n DSP(\lambda).p_1(\lambda_i) d\lambda \\ x_2 &= \int_{i=0}^n DSP(\lambda).p_2(\lambda_i) d\lambda \\ x_3 &= \int_{i=0}^n DSP(\lambda).p_3(\lambda_i) d\lambda \end{aligned}$$

Nous verrons dans la section 4.2.3 comment déterminer expérimentalement les fonctions colorimétriques.

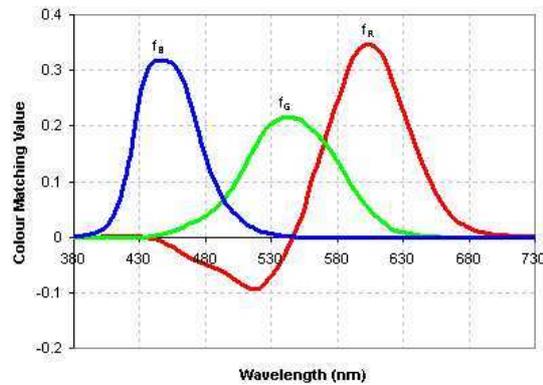


FIG. 4.1 – Fonctions colorimétriques de l'espace RGB

4.2.3 L'observateur standard CIE et l'espace colorimétrique CIE 1931 RGB

La perception des couleurs étant en partie subjective, il est évident qu'il faudra, d'une manière ou d'une autre, prendre en compte l'observateur. La CIE (Commission Internationale de l'Éclairage) a dans ce but déterminé en 1931 un observateur dit standard, dont les caractéristiques visuelles sont normalisées, qui sera utilisé pour la construction de toute la théorie de la couleur. Les caractéristiques de cet observateur standard sont basées sur les expériences d'égalisations menées par Guild[23] et Wright[82]. Il était alors demandé aux observateurs d'égaliser deux champs de deux degrés de vision chacun, sur un fond achromatique. Le premier champ était constitué d'une radiation monochromatique, alors que le deuxième était le résultat de la composition de trois radiations monochromatiques dont l'observateur pouvait contrôler les fonctions de mélange. Ces expériences ont abouti à l'établissement de fonctions colorimétriques standard, données sur la figure 4.1. Les primaires choisies sont $R = 650nm$, $V = 530nm$, $B = 460nm$. L'espace colorimétrique associé est l'espace CIE 1931 RGB. On a donc $S = r.R + v.V + b.B$, r, v, b représentant les coordonnées trichromatiques. Ces dernières sont déterminées avec la méthode évoquée dans la section précédente. Notons qu'une deuxième version de l'observateur standard a vu le jour en 1964, prenant en compte des champs visuels plus étendus que les deux degrés initiaux.

4.2.4 L'espace CIE 1931 XYZ

Il est visible sur la courbe 4.1 que la fonction colorimétrique $r(\lambda)$ prend des valeurs négatives. C'est en partie ce qui a conduit la CIE à choisir d'autres primaires pour aboutir au système CIE 1931 XYZ. Les fonctions colorimétriques de cet espace ont été choisies de manière à satisfaire plusieurs critères. Nous venons de citer le premier (toutes les valeurs doivent être positives). Le deuxième stipule que la composante Y doit représenter la luminance. Enfin, chaque fonction est calée et normalisée sur les mesures effectuées pour l'observateur standard. Le passage des coordonnées trichromatiques RGB aux coordonnées XYZ se fait avec la matrice suivante :

$$\begin{vmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,81240 & 0,01063 \\ 0,00 & 0,01 & 0,99 \end{vmatrix}$$

On a donc $S = x.X + y.Y + z.Z$, avec x, y, z représentant les coordonnées trichromatiques et X, Y, Z représentant les primaires.

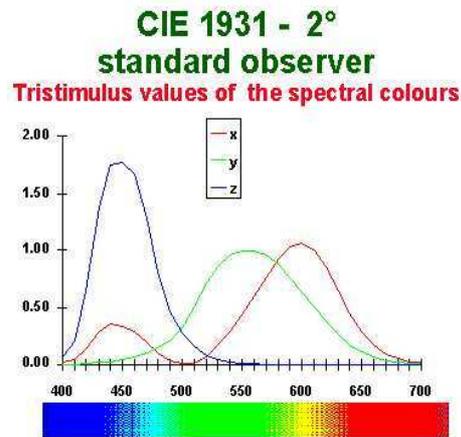


FIG. 4.2 – Fonctions colorimétriques pour l'espace XYZ

Les fonctions colorimétriques associées aux trois primaires n'ont ici que des valeurs positives. Il est intéressant de remarquer que les primaires utilisées pour l'espace XYZ ne sont pas physiquement réalisables, ce sont des primaires théoriques. La figure 4.2 montre les fonctions trichromatiques $x(\lambda), y(\lambda), z(\lambda)$.

Il est quelquefois plus pratique de manipuler deux dimensions au lieu de trois. Pour le repérage des couleurs, il est possible de projeter l'espace colorimétrique XYZ dans un plan. En effet, en faisant subir aux coordonnées trichromatiques la transformation $x' = \frac{x}{x+y+z}$, on a alors $x' + y' + z' = 1$. x' et y' suffisent donc pour déterminer z' . En représentant les couleurs dans le plan défini par x' en abscisse et y' en ordonnée, nous obtenons un diagramme de chromaticité (figure 4.3). Pour information, le triangle représente l'ensemble des couleurs reproductibles par un dispositif informatique.

4.2.5 L'espace colorimétrique SML

Il peut être intéressant, lors de travaux sur l'étude de la vision des couleurs, de disposer d'un espace dont les fonctions colorimétriques correspondent à une certaine réalité physiologique, et ne soit donc plus déterminées à partir d'expérimentations dans lesquelles le contexte joue une importance fondamentale. C'est dans cette optique que la CIE a proposé l'espace colorimétrique SML (pour "Short Medium Long"). Les fonctions colorimétriques de cet espace sont les fonctions de sensibilité spectrale des cônes, normalisées de manière à ce que la somme des fonctions L et M redonnent la fonction d'efficacité lumineuse $V(\lambda)$ (il a en effet été mis en évidence que seuls les cônes M et L contribuent à la luminance). La normalisation de S est arbitraire.

L'obtention des coordonnées dans l'espace SML dépend donc des fonctions de sensibilité $S(\lambda)$, $M(\lambda)$ et $L(\lambda)$ choisies. La CIE recommande les fonctions colorimétriques de Stiles et Burch (1959). Les fonctions de Smith et Pokorny (1975) sont également très appréciées. Ces dernières s'obtiennent depuis les coordonnées XYZ avec la matrice suivante :

$$\begin{vmatrix} 0,15514 & 0,54312 & -0,03286 \\ -0,15514 & 0,45684 & +0,03286 \\ 0,00000 & 0,00000 & 0,01608 \end{vmatrix}$$

Le fait d'utiliser cet espace couleur a de nombreuses avantages :

- Il permet une simulation des phénomènes rétiens comme l'aberration chromatique.

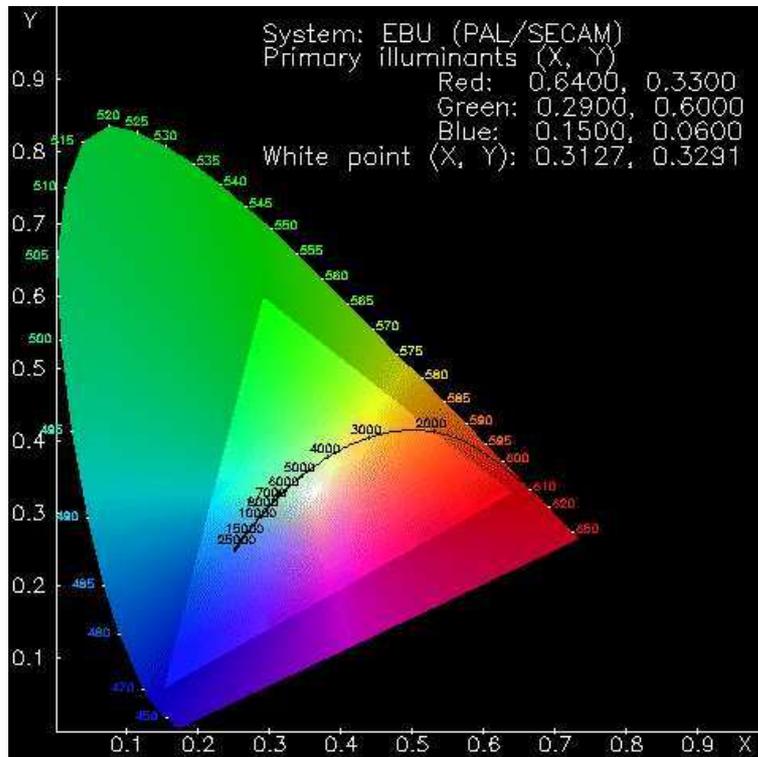


FIG. 4.3 – Diagramme de chromaticité xy

- Il permet de prédire la visibilité des couleurs dans le cas d’observateurs dichromates (ou “daltoniens” dans le langage courant) dépourvus d’un des trois types de récepteurs.
- Il est mieux adapté à des problèmes dont la résolution était empirique, comme par exemple le métamérisme.

Plus de détails concernant SML peuvent être trouvés dans [75].

4.2.6 L’(es) espace(s) colorimétrique(s) AC1C2

On dénomme généralement par AC1C2 un espace colorimétrique qui code la couleur sur trois canaux antagonistes : un canal achromatique A, un canal d’opposition rouge - vert C1 et un canal d’opposition bleu - jaune C2. Il est généralement dérivé d’un espace SML. Il existe plusieurs moyens d’atteindre cet objectif. Par exemple, dans [46], Meyer utilise une transformation de Kahunen - Love pour obtenir un espace dont les trois canaux sont le plus indépendants possible. Les espaces AC1C2 présentent des avantages certains pour simuler les traitements des couches plexiformes internes de la rétine, ils sont donc très présents dans les modèles de vision (voir le chapitre 5).

4.2.7 Colorimétrie des différences

Un des avantages à organiser les couleurs en espace vectoriel est le fait de pouvoir établir, sur cet espace, une métrique. Les espaces colorimétriques cités dans les sections précédentes présentent un inconvénient de taille à ce niveau là : ils ne sont pas perceptuellement uniformes. En effet, les différences entre deux couleurs ne sont pas perçues avec la même intensité quelle que soit la couleur, et ce même si la comparaison est effectuée indépendamment du contexte.

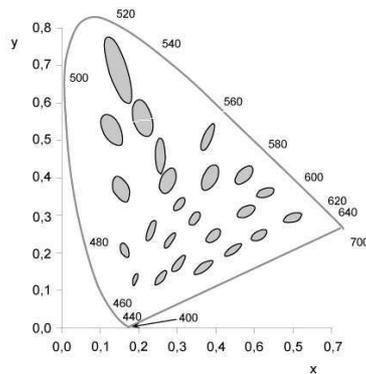


FIG. 4.4 – Ellipses de Mac Adam

MacAdam a, dans [40], effectué une série d'études sur les seuils de différenciabilité des couleurs. Le but était d'obtenir un diagramme de chromaticité "unifié", dans lequel la distance euclidienne entre deux couleurs est la plus proportionnelle possible à l'écart visuel. Un des résultats de cette étude est représenté sur la figure 4.4 : chaque ellipse représente une zone de "non différenciabilité" à l'intérieur même du diagramme CIE 1931 ; il est ici clairement visible que toutes les ellipses n'ont pas la même taille.

Plusieurs méthodes sont envisageables pour palier cet inconvénient :

- un changement de repère local peut transformer l'ellipse en cercle. Cette solution n'est pas complètement satisfaisante, dans la mesure où elle reste locale.
- une transformation projective globale. Les tentatives sont nombreuses dans ce domaine, comme par exemple le diagramme de Judd ([30])
- une déformation de l'espace consistant à reconstituer une surface gauche à partir du diagramme de chromaticité en le subdivisant en carreaux perceptuellement uniformes, donc non nécessairement tous de la même taille, puis en mettant à l'échelle ces carreaux. Ces travaux sont ceux de Mac Adam dans [40].

Toutes les solutions précédentes sont des transformations du diagramme de chromaticité, ce qui sous-entend que les ellipses de Mac Adam sont indépendantes de la luminance. Ce n'est pas tout à fait vrai dans la pratique : il paraît naturel que ces ellipses devraient s'agrandir quand la luminance baisse.

Signalons que d'autres approches concernant la vision des couleurs, et notamment la colorimétrie des différences, peuvent être trouvées dans [71].

4.2.8 L'espace CIELAB

Il existe de nombreux travaux visant à mettre au point une transformation colorimétrique permettant de passer de XYZ à un espace le plus uniforme possible. La plus connue reste celle de l'espace CIELAB. Les coordonnées L^* , a^* , b^* s'obtiennent grâce à la transformation suivante :

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{frac{1}{3}} - 16$$

$$a^* = 500\left(f_x\left(\frac{X}{X_n}\right) - f_y\left(\frac{Y}{Y_n}\right)\right)$$

$$b^* = 500\left(f_y\left(\frac{Y}{Y_n}\right) - f_z\left(\frac{Z}{Z_n}\right)\right)$$

Les fonctions f_x et f_y sont définies de la manière suivante :

$$f_x\left(\frac{X}{X_n}\right) = \begin{cases} \left(\frac{X}{X_n}\right)^{1/3} \text{ si } \frac{X}{X_n} > 0,008856 \\ 7,787 \cdot \left(\frac{X}{X_n}\right) + \frac{16}{116} \text{ sinon} \end{cases}$$

$$f_x\left(\frac{Y}{Y_n}\right) = \begin{cases} \left(\frac{Y}{Y_n}\right)^{1/3} \text{ si } \frac{Y}{Y_n} > 0,008856 \\ 7,787 \cdot \left(\frac{Y}{Y_n}\right) + \frac{16}{116} \text{ sinon} \end{cases}$$

$$f_x\left(\frac{Z}{Z_n}\right) = \begin{cases} \left(\frac{Z}{Z_n}\right)^{1/3} \text{ si } \frac{Z}{Z_n} > 0,008856 \\ 7,787 \cdot \left(\frac{Z}{Z_n}\right) + \frac{16}{116} \text{ sinon} \end{cases}$$

Dans cet espace colorimétrique, la métrique euclidienne peut être utilisée de manière relativement fiable :

$$\Delta E = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$$

Le résultat de cette opération est une valeur calibrée en Just Noticeable Differences (JNDs). Cette unité de mesure de la perception des différences fonctionne de la manière suivante : si, sur une comparaison entre motifs, la valeur de la JND en un point est de 1, alors cela signifie que la probabilité qu'un observateur moyen perçoive une différence en ce point est de 0,75. Si la valeur de la JND est 2, alors la probabilité est de $0,75 + 0,75 \cdot (1 - 0,75)$. D'une manière générale, la formule $JND_{N+1} = JND_N + 0,75(1 - JND_N)$, avec $JND_1 = 0,75$, permet de calculer la probabilité correspondante à chaque valeur de JND.

Les conditions de validité de CIELAB sont celles des tests de différenciation, donc assez restrictives : champ de deux degrés de vision, uniforme, et fond achromatique. C'est évidemment bien loin des conditions de visualisation d'une scène réelle. Beaucoup de recherches actuelles portent donc sur l'établissement d'une méthode ou d'un modèle capable de s'affranchir de ces limitations.

4.2.9 Le modèle CIECAM97s

Le modèle d'apparence couleur CIECAM97s [38] s'appuie sur une description formelle mais précise de la scène visualisée. On y compte 5 champs visuels de taille croissante :

- l'échantillon lui-même, représentant la zone visualisée ;
- le voisinage immédiat de l'échantillon ;
- l'arrière plan, ou fond ;
- la périphérie, équivalente à la vision périphérique
- le champ d'adaptation, qui englobe tous les autres et qui va jusqu'aux limites du champ visuel.

Un nombre conséquent de paramètres permet d'affiner l'apparence colorée.

- Influence de l'échantillon ;
- Effet de contraste de clarté ;
- Impact de la périphérie ;
- Induction chromatique ;
- Luminance d'adaptation.

Ces paramètres permettent d'en calculer d'autres, comme le degré d'adaptation ou l'induction lumineuse ou chromatique par l'arrière plan. Le modèle lui-même se décompose en trois phases : l'adaptation chromatique, la dynamique de réponse des cônes, et la détermination des attributs de l'apparence colorée. Le modèle n'étant pas utilisé par nos travaux, nous n'entrerons pas dans les détails pratiques, mais le lecteur intéressé peut se reporter à [38].

4.3 Reproduction de tons et Fidélité

Nous avons quelques outils pour mesurer et quantifier la couleur. Il nous reste maintenant à franchir une étape supplémentaire : la reproduction de cette information couleur sur un écran de manière correcte. Dans un cadre un peu plus général, nous pouvons nous interroger sur la question de la fidélité en synthèse d'images ; il y a plusieurs manières d'être fidèle :

- Fidélité vis à vis de la carte de luminance : le but est ici de convertir de manière satisfaisante la carte obtenue à la fin du rendu en une image affichable sur un écran. Cette opération, nommée reproduction de tons (ou tone mapping), est cruciale car elle conditionne pour beaucoup la qualité du rendu définitif. Elle est considérée comme réussie si l'observateur de l'image sur écran a la sensation que ce qu'il voit est fidèle à ce qu'il verrait si il était physiquement présent dans la scène simulée.
- Fidélité vis à vis de la "réalité", ou plus modestement, vis à vis d'une photographie. Concrètement, nous souhaiterions qu'un observateur "moyen" ne puisse pas faire de différence entre une scène en image de synthèse et une photographie de la même scène. Ce problème porte le nom de photo-réalisme.

Dans cette section, nous nous attacherons à décrire ces problèmes et les solutions qui se présentent à nous. Les deux aspects de la fidélité évoqués plus haut ont un point commun : leur réussite dépend du calibrage du moniteur. Nous allons donc brièvement présenter ce problème en premier, avant de passer en revue les deux autres.

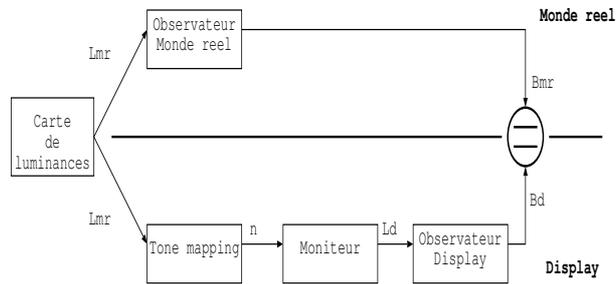
4.3.1 Calibrage du dispositif de reproduction

Avant de pouvoir effectuer notre opération de conversion luminances réelles - luminances affichables, il nous faut déjà nous assurer que le dispositif de reproduction utilisé est correctement calibré pour l'usage que l'on veut en faire. Cela suppose éventuellement une caractérisation de ses fonctions de réponse aux coordonnées trichromatiques pour chacun des canaux. Dans le meilleur des cas, cette réponse est linéaire dans sa plus grande partie. Pour la déterminer, divers instruments de mesure peuvent être utilisés, comme un luminance-mètre ou un spectrophotomètre. Dans ces domaines, le lecteur intéressé peut se reporter aux travaux de Cowan([10]) ou Brainard([8]). Signalons qu'en recherche sur la vision, un moniteur est toujours calibré dans un but bien précis : maximisation des contrastes discernables, des détails visibles, optimisation du rendu des couleurs ... Il est en effet impossible d'obtenir tous ces effets à la fois. Les thèses de Dumont-Bècle([13]) et Deniel([12]) en sont des exemples représentatifs.

4.3.2 Les opérateurs de reproduction de tons (ORT)

La figure 4.5, issue de [72] illustre le principe de base d'un ORT. Nous avons deux observateurs, le premier se trouve en face de la scène en luminances réelles, le deuxième regarde un écran sur lequel est affichée cette même scène, convertie au préalable par l'ORT et le circuit vidéo du moniteur. Le problème est simple : il nous faut égaliser les sensations visuelles des deux observateurs. Cette sensation visuelle, dénommée "brillance" par les anglo-saxons, se mesure en bril.

La figure 4.6 montre trois images issues de la même carte de luminances. Seuls les opérateurs de reproduction de tons changent. Nous allons présenter les principaux travaux, dans le domaine de la synthèse d'images, concernant les ORT.



D'après Tumblin-Rushmeier (92)

FIG. 4.5 – Schéma de principe de la reproduction de tons



FIG. 4.6 – Différents opérateurs de reproduction de tons : Tumblin-Rushmeier, Ward, Ward-Rushmeier-Piatko

L'ORT de Tumblin - Rushmeier

Selon les travaux de Tumblin et Rushmeier[72], l'obtention d'un ORT est un processus logique : en se basant sur la figure 4.5, un ORT sera obtenu simplement en prenant un modèle d'observateur du monde réel (RWO), en lui ajoutant un modèle inversé d'observateur de l'écran (DO) et un modèle inversé de moniteur. Les fonctions utilisées par Tumblin et Rushmeier sont les suivantes :

- RWO : $B_{rw} = 10^{\beta_{rw}} \cdot L_{rw}^{\alpha_{rw}}$
- DO : $B_d = 10^{\beta_d} \cdot L_d^{\alpha_d}$
- Moniteur : $n = [(Ld/Ld_{max}) - 1/C_{max}]^{1/\gamma}$

Les fonctions utilisées sont toutefois relativement complexes, et sont basées sur des fonctions psychométriques mesurant des capacités achromatiques. En toute rigueur, cet ORT ne peut donc pas être utilisé avec des images en couleurs.

L'ORT de Ward

Dans [77], Ward veut simplifier le processus en établissant une relation linéaire entre les luminances réelles et celles qui sont émises par l'écran, un facteur unique servant donc à la conversion de toute l'image. L'idée de Ward est que ce facteur devrait être égal au rapport entre le seuil de luminance perceptible sur l'écran et celui perceptible sur la carte de la luminance ; ces seuils ont fait l'objet d'études, présentées dans le chapitre précédent, pour lesquelles des formulations mathéma-

tiques existent. Ward utilise la formule élaborée par la CIE en 1981 ([70].)

$$\delta L(L_a) = 0.0594.(1.219 + L_a^0.4)^2 .5$$

Ce modèle a le mérite de donner de bons résultats avec une solution qui, même si elle est basée sur des critères perceptuels, reste facilement applicable. Toutefois, pour obtenir ces seuils, il est nécessaire de connaître les caractéristiques du moniteur (en particulier sa luminance maximale) ainsi que la luminance d'adaptation. Cette notion, dont il n'existe pas de définition précise à l'heure actuelle, est censée représenter l'état d'adaptation de l'observateur et peut donc recouvrir un bon nombre d'aspects psychovisuels. Elle est généralement assimilée à la luminance du fond. Les paramètres de la méthode de Ward sont donc difficiles à définir.

L'ORT de Ward - Rushmeier - Piatko

La conception de cet opérateur vient d'une constatation sur les précédents travaux : certains conservent la visibilité globale, d'autres préservent la sensation visuelle, notamment tout ce qui est lié à l'éblouissement ou aux halos de lumière, mais aucun n'intègre les deux aspects. Pour aboutir à cet objectif, Ward *et al* proposent dans [34] d'effectuer une égalisation d'histogramme basée sur des fonctions psychovisuelles. La méthode est la suivante :

1. Sur une première copie de l'image, on applique :
 - une fonction d'acuité visuelle
 - une fonction de sensibilité à la couleur
 - une fonction de calcul d'éblouissement
2. Une deuxième copie de l'image est filtrée de manière à obtenir qu'un pixel de cette image représente un degré de vision.
3. Une fonction de sensibilité au contraste est appliquée sur cette image.
4. Le logarithme de la luminance est calculé en chaque pixel de manière à approximer la brillance.
5. Une égalisation d'histogramme est effectuée sur cette image.
6. Cette égalisation est reportée sur la première image pour obtenir l'image affichable.

L'ORT de Pattanaik *et al*

Dans [55], les auteurs utilisent un modèle de vision complet. Nous reviendrons sur le modèle ultérieurement dans ce mémoire, il ne sera donc pas détaillé dans cette section. Le principe est le suivant : un ORT sera obtenu avec un processus en deux phases ; L'image en luminances réelles est convertie en contrastes perçus avec un modèle de vision, puis ces contrastes sont traités à travers le modèle de vision inversé, réajusté en changeant les paramètres de manière à prendre en compte le changement scène réelle / écran. Cet opérateur est le seul, à l'heure actuelle, à prendre en compte les trois domaines de visualisation : scotopique, mésopique et photopique. Il est également très complet sur tous les aspects couleur de l'image à afficher. La figure 4.7 montre quelques exemples d'images utilisant cet algorithme.

Les ORT fonctionnant en temps réel

Certaines applications, comme la réalité virtuelle, nécessitent de faire l'opération de reproduction de tons en temps réel, c'est à dire en moins d' $1/24^{eme}$ de seconde. Par exemple, dans le cas de "walkthroughs"(c'est à dire une application dans laquelle l'utilisateur peut diriger interactivement



FIG. 4.7 – Résultats de l’opérateur de reproduction de tons de Pattanaik *et al* (ligne du bas)

la caméra), la prise en compte des phénomènes d’adaptation visuelle quand le point de vue passe d’une pièce sombre à une pièce fortement éclairée peut apporter un degré de réalisme supplémentaire. Nous avons retenu trois travaux ayant cet objectif, mais c’est un sujet qui est appelé à se développer énormément dans un avenir proche.

Scheel *et al* ont dans [62] mis au point une méthode de reproduction de tons utilisant au maximum les capacités graphiques de la machine. Les fonctions mathématiques de l’opérateur, basées sur [77], sont stockées et utilisées dans des mémoires normalement réservées aux textures. Le calcul de la luminance d’adaptation se fait en échantillonnant l’espace image avec un processus aléatoire pondéré par une fonction de distribution privilégiant une zone d’environ deux degrés de vision autour du point de focalisation.

Durand et Dorsey présentent dans [14] une mise en œuvre en temps interactif de l’algorithme de simulation de l’adaptation visuelle de Ferwerda *et al* ([19]), à laquelle ils ajoutent quelques fonctionnalités comme les halos lumineux provoqués par une source intense. Le gain de vitesse par rapport à l’algorithme original se fait surtout en tabulant toutes les fonctions mathématiques et en utilisant les convolutions câblées disponibles dans OpenGL.

Artusi et Wilkie ([5]) proposent quant à eux une méthode de linéarisation des processus de reproduction de tons de manière à les approximer par une “look up table” immédiatement accessible, donc très rapide. Le gros avantage de leur méthode est qu’elle peut s’appliquer à n’importe quel ORT.

4.3.3 Fidélité et photo-réalisme

Le deuxième aspect de la fidélité en synthèse d’image occupe une place de choix dans les objectifs principaux de cette science : le photo-réalisme. Évitions d’emblée la confusion avec le physico-réalisme, qui impose une simulation la plus complète possible de tous les phénomènes physiques présents dans la scène virtuelle, et ce même si ils ne sont pas perceptibles. Pour l’instant, notre objectif sera considéré comme atteint si un observateur moyen est incapable de faire la différence entre une image de synthèse et une photographie de la même scène. Il est ici important de préciser le contexte : les deux images sont deux cartes de pixels, affichées sur le même écran et visualisées dans les mêmes conditions physiques, aussi bien au niveau de l’observateur qu’au niveau de l’environnement. La figure 4.3.3 représente un exemple de ce qu’il est possible d’obtenir à l’heure actuelle en matière de photo-réalisme. L’image de droite est une photographie, celle de gauche une image de synthèse.

Le photo-réalisme nécessite d’intervenir à deux niveaux. Premièrement, une base physique pour le calcul des images est nécessaire : en effet, dès que la simulation se complexifie un peu, les mo-



FIG. 4.8 – comparaison scène réelle - image de synthèse (d’après [86])

dèles empiriques ne pourront plus donner des résultats qui “semblent” réalistes. Les techniques de rendu réaliste seront exposées dans le chapitre 6. Deuxièmement, le photo-réalisme fait intervenir un observateur humain pour son processus de validation. Dans [60], Rushmeier *et al* adressent pour la première fois ce problème non négligeable (cet article sera exposé plus en détail dans le chapitre suivant). Par la suite, McNamara a effectué de nombreux travaux sur la perception des différences et la fidélité en synthèse d’images ([45], [44]). Tous ces travaux suggèrent que l’observateur humain pourrait être remplacé par un modèle de vision, qui est une simulation informatique du processus de vision humaine. Une métrique de validation pourrait alors être bâtie sur ce modèle. Nous allons présenter les modèles de vision et les méthodes de validation du photo-réalisme dans le chapitre 5.

Chapitre 5

Métriques et Modèles de vision pour la synthèse d'images

Dans le chapitre précédent, nous avons vu comment mesurer et quantifier la lumière et la couleur. Nous avons également passé en revue les capacités et caractéristiques du système visuel. Nous allons maintenant exploiter ces études pour les appliquer au rendu d'images réalistes. Nous voulons dans un premier temps mesurer la similarité entre deux images, et valider le fait qu'une simulation soit "suffisamment proche" d'une scène réelle. Nous avons pour cela conçu une métrique permettant de mesurer la distance entre deux images : prenant en entrée deux images, cet algorithme renvoie une carte des différences ou une valeur de distance. Cette carte se doit d'être cohérente avec la perception visuelle : détecter des différences invisibles à l'œil nu ne serait d'aucune utilité dans le cadre de ce travail. Comme le montre la figure 5.1, la métrique euclidienne ne satisfait pas du tout ce critère : nous avons sur cette figure trois portraits de Jean-Sebastien Bach. Le premier n'est pas altéré (image 1), le deuxième a subi un léger décalage sur la droite (image 2), le troisième a été bruité (image 3). En appliquant une métrique euclidienne, pixel à pixel, dans l'espace colorimétrique CIELAB, la différence moyenne vaut 6,60 JNDs entre les images 1 et 3, et 10,43 JNDs entre les images 1 et 2. Elle est donc plus importante dans la configuration où le système visuel perçoit pourtant des différences bien moindres. Nous pouvons constater que la comparaison pixel à pixel d'une image ne donne pas un résultat cohérent avec la vision humaine. Le fait d'utiliser CIELAB n'arrange pas grand chose, car l'utilisation de ce dernier se fait dans des conditions de visualisation bien spécifiques qui ne sont absolument pas respectées ici. La nécessité d'une métrique perceptuelle, qui permettrait de déterminer les différences perçues entre deux images, est donc bien réelle. Dans ce chapitre, nous allons présenter les métriques entre images déjà proposées, ainsi que notre propre contribution.

5.1 Principales métriques existantes

Les travaux dans ce domaine abondent en traitement et analyse d'images ; dans les domaines de la compression d'image, de la mesure de qualité, ces outils sont en effet très répandus. Ahumada a fait un très bon article de synthèse sur ce sujet : dans ([2]), il fournit un tableau comparatif des travaux existants (dont la taille impressionnante donne une idée des investigations passées et présentes) et en extrait le schéma global d'une métrique (il se limite toutefois aux métriques qui traitent les images statiques et monochromatiques). Selon ces travaux, une métrique est un processus qui, étant données deux images I_1 et I_2 , fournit dans un premiers temps deux sorties perceptuelles $P(I_1)$ et $P(I_2)$ issues d'un modèle visuel, qui n'est pas nécessairement un modèle de vision, mais peut être simplement une

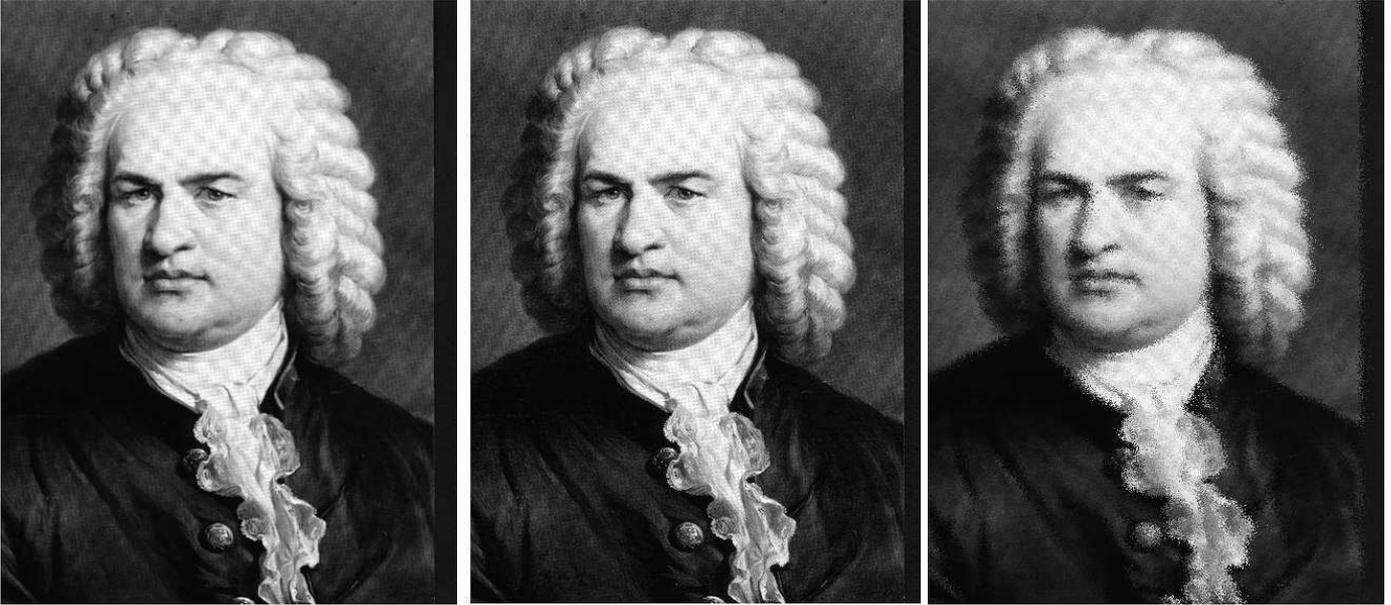


FIG. 5.1 – A gauche : portrait intact (image 1), au milieu : portrait décalé (image 2), a droite : portrait bruité (image 3)

fonction psychométrique. Dans un deuxième temps, une règle d'intégration $Q(P(I_1), P(I_2))$ donne la distance entre I_1 et I_2 . Pour que la métrique soit valable, il est nécessaire que le résultat de Q soit proportionnel à la probabilité qu'un observateur perçoive une différence entre les deux images. Idéalement, la règle d'intégration doit donner un résultat calibré en Just Noticeable Differences (JNDs). Rappelons qu'une JND vaut 1 en un point (i, j) de l'image quand un observateur "moyen" a une probabilité de 0,75 de percevoir une différence en ce point (i, j) . Nous allons exposer dans cette section les métriques utilisées en synthèse d'images, en présentant si besoin est leur modèle visuel et leur règle d'intégration.

5.1.1 Les tests de Rushmeier *et al*

Dans [60], Rushmeier *et al* ont les premiers l'idée d'appliquer des modèles issus de la compression d'images afin de les appliquer à la comparaison d'images réelles et synthétiques. Les auteurs ont donc acquis des images en luminances d'une scène réelle (une salle de conférence spécialement aménagée), puis ils ont relevé toutes les cotes de cette pièce pour en créer une simulation synthétique. Ils ont ensuite comparé trois métriques issues de l'analyse d'images et ont pour cela formalisé le cadre de l'étude en imposant cinq propriétés à vérifier sur les métriques testées. Soit $M(A;B)$ la métrique comparant les images A et B ($M(A;B)$ est une valeur numérique); il est alors nécessaire d'avoir :

- $M(A;A) = 0$
- $M(A;B) = M(B;A)$
- Si A et B sont similaires et si A et C sont sensiblement différentes, alors $\frac{M(A;B)}{M(A;C)} \gg 1$
- Si A, B, C sont trois images similaires, alors $\frac{M(A;B)}{M(A;C)} \simeq 1$
- Si les différences entre les images A et B sont similaires aux différences entre les images C et D , alors $\frac{M(A;B)}{M(C;D)} \simeq 1$

Toutes les métriques testées vérifient les deux premières propriétés. Pour avoir des images “sensiblement différentes”, les auteurs ont utilisé des images aléatoires, c’est à dire dont la luminance des pixels est attribuée aléatoirement mais dont la moyenne et l’écart type sont les mêmes que celles de l’image réelle. Pour avoir des images similaires, les auteurs ont utilisé des images synthétiques simulant la scène réelle, mais sans aucun calcul d’éclairement global ; l’image est calculée avec un terme ambiant constant. Pour la dernière propriété, ils utilisent deux configurations différentes de la pièce, et comparent les résultats avec les deux simulations respectives. Les trois métriques utilisées furent celles de Mannos et Sakrison ([42]), Gervais *et al* ([21]) et Daly([11]) (nous reviendrons sur cette dernière ultérieurement dans ce chapitre). Les premières conclusions montrent que, quelle que soit la métrique choisie, elle s’avère dans tous les cas meilleure qu’une métrique euclidienne. Ce résultat est dû en grande partie à l’extrême sensibilité aux variations géométriques de cette dernière. L’article présente ensuite une comparaison détaillée des trois autres métriques, que nous ne détaillerons pas ici. Ces travaux ont pour principal intérêt d’avoir montré l’utilité d’un modèle de vision dans un processus de rendu photoréaliste.

5.1.2 Les métrique de Gilles Rougeron et Stéphane Albin

Au sein du laboratoire LISSE, Gilles Rougeron([59]) a fait office de pionnier en matière de métrique perceptuelle. L’idée était d’utiliser l’espace *LLab*, mis au point par Luo([39]) pour comparer deux images de synthèse. Cet espace colorimétrique est censé être plus performant que $L^*a^*b^*$, car il prend en compte le contexte de l’échantillon visualisé. Cette méthode ne peut s’appliquer que dans certaines conditions : premièrement, le point de vue doit être identique pour les deux images. Deuxièmement, les images ne doivent pas comporter de textures. Enfin, les scènes ne contiennent qu’une seule source lumineuse. Le modèle visuel de Gilles Rougeron est basé sur l’espace colorimétrique *LLab*, dont l’utilisation requiert le calcul d’un certain nombre de paramètres (dont la luminance du fond) et conduit à la détermination de trois coordonnées de luminance (L), de clarté (C) et d’angle de teinte (H). La distance entre deux couleurs est alors une distance euclidienne exprimée avec ces coordonnées.

Pour calculer les paramètres nécessaires à l’utilisation de *LLab*, Gilles Rougeron utilise deux masques centrés sur le pixel focalisé : une zone de deux degrés, correspondant à la zone fovéale, et une zone de vingt degrés autour du pixel, correspondant à la vision périphérique. Les pixels situés dans la première zone forment le focus, les autres forment le fond. Ces zones sont raffinées par segmentation : seuls les pixels appartenant au même objet que le pixel central seront considérés comme faisant partie du focus (la connaissance exacte de la géométrie de la scène permet une segmentation très simple). La luminance du fond sera alors calculée comme une moyenne pondérée des pixels de la zone de fond, ce qui permet la détermination des coordonnées des pixels de la zone de focus.

Après ce modèle visuel, la règle d’intégration est la suivante : l’erreur pixel à pixel est calculée pour chaque pixel de la zone de focus, et la moyenne est affectée au pixel central. Les résultats de cette méthode sont assez encourageants, mais l’utilisation de l’espace *LLab* rend les temps de calcul prohibitifs : le calcul d’une carte de distance peut nécessiter plusieurs dizaines de minutes !

Stéphane Albin a repris cette métrique lors de son stage de DEA ([3]). L’objectif premier étant de diminuer le temps nécessaire, il a modifié la méthode de Gilles Rougeron en deux points :

- Il a remplacé l’espace *LLab* par $L^*a^*b^*$, nettement plus facile à mettre en oeuvre.
- Il a effectué un calcul partiel de la distance : celle-ci n’est plus évaluée sur tous les pixels, mais seulement sur un faible pourcentage représentatif de l’image. De la même manière que pour un sondage, ses tests démontrent qu’une population de 2000 individus (qui correspondent en fait à 2000 pixels dont il calcule effectivement la distance) suffit pour obtenir une erreur inférieure à

2%, pour peu que la population totale soit suffisamment grande.

Avec ces améliorations, il obtient des temps de calculs réellement très courts : sur un Pentium4 2Ghz doté d'un gigaoctet de mémoire vive, ils sont difficilement mesurables. De plus, les limitations de la méthode de Gilles Rougeron ne s'appliquent plus ici. Nous reviendrons sur cette méthode dans la section consacrée à notre distance adaptative. Ces travaux présentent un intérêt certain, car les métriques ont une structure facile à mettre en oeuvre, mais les espaces colorimétriques sont utilisés en dehors du cadre de leurs utilisation normale : en effet, LLab n'est valable que si le fond du motif considéré est achromatique, ce qui n'est généralement pas le cas dans une image de synthèse. Quant à $L^*a^*b^*$, il n'est conçu que pour mesurer des distances entre deux couleurs. Ces méthodes ne prennent pas en compte non plus tous les effets spécifiques à la vision des contrastes.

5.1.3 La métrique de Neumann *et al*

L'idée de base de la Méthode de Neumann *et al*, présentée dans [51], est d'effectuer sur l'image une transformation basée sur une fonction de sensibilité au contraste. Le modèle visuel est composé d'un filtrage bien spécifique que nous allons détailler.

Le filtrage ne sera calculé que sur un nombre restreint de pixels choisis avec un algorithme pseudo-aléatoire. Sur chacun de ces pixels, un rectangle sera placé, et une moyenne de la couleur (exprimée en coordonnées $L^*a^*b^*$) des pixels à l'intérieur de ce rectangle sera calculée. Le problème maintenant est de choisir la taille du rectangle. Cette taille sera elle aussi déterminée aléatoirement, mais ce processus aléatoire sera guidé par une fonction de sensibilité au contraste (FSC) de la manière suivante :

- Une formulation mathématique de la FSC est déterminée. Les auteurs ont choisi celle de Mannos et Sakrison ([42]).
- La fonction de répartition de la FSC est calculée, puis normalisée et inversée.
- Grâce a cette fonction, nous pouvons associer un nombre aléatoire entre 0 et 1 à une fréquence spatiale en cycles par degrés de vision. Il nous faut maintenant convertir cette fréquence spatiale en cycles par pixels.
- Un facteur r représentant le nombre de pixels contenus dans un degré de vision est calculé. r peut être calculé en fonction de la taille de l'écran, de sa résolution et de la distance de visualisation avec la formule suivante : $r = \frac{2.R.d.\tan(1/2)}{W}$, d étant la distance de visualisation, R la résolution de l'écran et W la taille de la diagonale de l'écran.
- En divisant la fréquence, en cycles par degrés, par r , nous obtenons une fréquence en cycles par pixels, et en inversant cette quantité, nous obtenons la taille du rectangle à placer autour du pixel.

Pour ce qui est de la règle d'intégration, les auteurs font la moyenne de toutes les valeurs obtenues dans tous les rectangles. L'avantage de cette métrique est d'être très rapide : en effet, les résultats varient très peu dès que l'on dépasse le seuil des 10 000 rectangles calculés, il n'est donc pas très utile d'aller au delà. En contrepartie, elle ne rend qu'un nombre pour la distance entre les deux images, ce qui n'est pas toujours très parlant. D'autre part, elle utilise aussi l'espace colorimétrique $L^*a^*b^*$ en dehors du cadre normal de son utilisation.

5.1.4 La métrique de Gaddipatti *et al*

Dans [20], Gaddipatti *et al* présentent une métrique reposant sur une décomposition en ondelettes de Daubechies combinée avec une fonction de sensibilité au contraste bidimensionnelle. Le principe du modèle visuel est donc de décomposer l'image avec un procédé de type ondelettes de Daubechies, puis de pondérer chaque niveau de la décomposition en ondelettes par le coefficient de la FSC

correspondant à la représentation fréquentielle de ce niveau. Le résultat de cette opération est une “sensibilité” par niveau et par pixel, $S_{m,i,j}$ (niveau m , pixel i, j), calculée de la manière suivante :

$$S_{m,i,j} = C_m \cdot W(A, m, i, j)$$

A est l’image à traiter et W est la décomposition en ondelettes. Le coefficient C_m est donné par la formule suivante :

$$C_m = \int_{FB_m} FSC \cdot \omega \, d\omega$$

avec $FB_m = (\frac{\pi}{2^m}, \frac{\pi}{2^{m-1}})$

La règle d’intégration est tout simplement une moyenne des moindres carrés des sensibilités de tous les pixels pour tous les niveaux de la décomposition :

$$M_p(A, B) = \frac{\sum_{m,i,j} |S_A(m, i, j) - S_B(m, i, j)|^2}{N_h \cdot N_v}$$

N_h et N_v sont les dimensions horizontale et verticale de l’image, en pixels.

Les auteurs démontrent ensuite que la métrique euclidienne n’est pas une métrique fiable, puis testent leur méthode. Pour cela, ils altèrent une image successivement avec une même transformation, puis comparent les évolutions des différences entre deux images successives. La transformation utilisée dans ce cas est un flou gaussien, appliqué sur une image médicale, une “lenna” et une image de synthèse. Les résultats semblent confirmer que la méthode à ondelettes suit bien plus fidèlement l’impression visuelle que la métrique euclidienne soumise aux mêmes tests.

Cet article est intéressant dans la mesure où il se focalise sur la vision des contrastes. Toutefois, l’aspect couleur n’est considéré qu’à travers le fait que l’espace colorimétrique $L^*a^*b^*$ est utilisé, encore une fois, pour l’encodage des images à traiter. Si l’on se réfère à la documentation existante sur la vision des couleurs, c’est loin d’être suffisant. De plus, le résultat fourni par cette méthode n’est pas calibrée en JNDs, les résultats sont donc difficilement quantifiables. L’idéal serait un modèle visuel capable de gérer à la fois la vision spatiale et la vision des couleurs, afin d’avoir une prise en compte globale de la perception visuelle. Pour cela, il nous faut un modèle de vision.

5.2 Les modèles de vision

Pour être perceptuellement fiables, les métriques sont généralement basées sur des modèles de vision, c’est à dire que leur modèle visuel n’est plus une simple fonction psychométrique mais un processus évolué : dans le cadre de ce mémoire, un modèle de vision est alors un algorithme qui prend en entrée des signaux lumineux et qui renvoie l’interprétation qu’en fait le système visuel au niveau du cortex. C’est donc une simulation des traitements effectués dans la rétine et les voies nerveuses. Le but est alors de concevoir une formalisation mathématique des études exposées dans les chapitres précédents. Les tentatives en ce sens sont nombreuses, mais elle sont assez rarement spécifiques au domaine de la synthèse d’images.

5.2.1 Structure générale d’un modèle de vision

En suivant le cheminement de l’information lumineuse dans le système visuel, nous avons successivement :

- *Les photorécepteurs* (cônes et bâtonnets) effectuent un processus de conversion d’énergie lumineuse en signaux électriques. Cette étape peut par exemple s’apparenter à un changement d’espace colorimétrique.

- *Les cellules des couches plexiformes* : nous avons vu que les cellules ganglionnaires sont toutes sensibles à une plage de fréquences différente. Le traitement qu'elles effectuent peut être assimilé à une décomposition fréquentielle de l'information transmise par les cônes. Quant aux cellules horizontales, elle ont pour rôle d'accentuer ou de réduire les antagonismes dans les signaux perçus ; ce traitement peut être modélisé par une fonction de gain.
- *Les voies nerveuses* réunissent les signaux venant des deux yeux et les transmettent au cortex visuel. A la sortie de la rétine, ces signaux sont antagonistes. Ce changement peut, là encore, s'apparenter à un changement d'espace colorimétrique.
- *Les traitements corticaux* organisent et interprètent l'information. Selon la puissance désirée pour le modèle, il est possible ici d'intégrer toutes les fonctionnalités qui n'ont pas leur place dans les étapes précédentes.

5.2.2 Le "Visual Differences Predictor" de Daly

Le modèle de Daly([11]) est un modèle achromatique destiné à prédire la visibilité de différences entre deux images. C'est donc une métrique perceptuelle basée sur un modèle de vision. Ce dernier est intéressant à plus d'un titre. Premièrement, sa structure est relativement simple. Deuxièmement, malgré le fait qu'il ne soit pas particulièrement récent, il reste relativement performant par rapport à des algorithmes plus actuels. Enfin, c'est un des premiers à avoir été utilisé dans le domaine de la synthèse d'images.

La structure du modèle de Daly

La figure 5.2 illustre le "pipeline" de fonctionnement du modèle de Daly. La structure se décompose en trois parties :

- Une simulation de l'adaptation en luminance des cellules rétiniennes, nommée "Amplitude Non-linearity", modélisée par une fonction de type "Threshold versus intensity", similaire à celles du chapitre 3.
- Une prise en compte de la vision spatiale modélisée par une fonction de sensibilité au contraste bidimensionnelle.
- Une simulation des traitements corticaux, nommée "Detection mechanism". Cette étape comprend elle même 2 sous-étapes :
 - Une décomposition spatiale orientée, basée sur la "cortex transform" de Watson([80])
 - Une fonction de masquage, qui simule le phénomène perceptuel du même nom.

Le modèle est ensuite utilisé pour obtenir une carte des différences perçues entre deux images. La règle d'intégration est la suivante :

- Les contrastes locaux sont calculés en soustrayant les résultats du modèle visuel sur les deux images à comparer.
- Une fonction psychométrique, chargée d'augmenter la probabilité de détection quand le contraste est élevé, est appliqué sur ces contrastes. Cette fonction peut s'apparenter à la fonction de transfert du système visuel. Cette fonction psychométrique fournit une probabilité pour un contraste donné.
- Les probabilités sont sommées (au sens probabiliste) pour chaque canal et chaque niveau de la décomposition. Le résultat est une carte des probabilités de détection des différences.

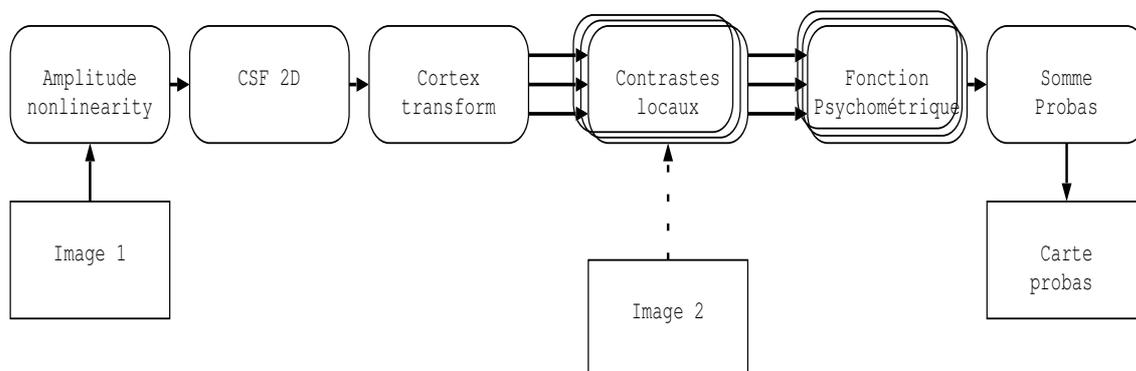


FIG. 5.2 – Schéma de fonctionnement du modèle de Daly

Résultats et performances

Le modèle de Daly a été utilisé par Myszkowski dans [49] pour évaluer ses performances dans le cadre d'une utilisation en synthèse d'images. Nous allons ici présenter les résultats qu'il a obtenus. L'auteur divise ses travaux en trois phases : une phase de validation pour vérifier si le modèle peut effectivement détecter avec pertinence des différences entre deux images de synthèse, une autre pour tester l'importance relative de chaque composante du modèle, et une phase de mise en application dans laquelle le modèle va piloter un algorithme de rendu progressif. Dans ce chapitre, nous ne présenterons que les résultats de la première phase, afin d'avoir une idée des performances du modèle. La troisième phase sera évoquée dans le chapitre 6, consacré aux algorithmes de rendu progressifs et perceptuels.

La phase de validation est un ensemble de tests simples. Pour le premier, il s'agit de vérifier si la méthode de Daly est capable de refléter la perception humaine dans une tâche qui consiste à détecter des différences sur des ombres portées masquées par une texture, de fréquence variable, appliquée sur toute l'image. La figure 5.3 illustre ce test et est structurée de la manière suivante :

- La ligne supérieure représente les images à comparer, la ligne inférieure les cartes de distance.
- La partie gauche de la figure contient les images avec la texture masquante, ici un quadrillage de l'image.
- La partie droite de la figure contient les images sans texture masquante.

D'après Myszkowski, les résultats sont concluants : le modèle de Daly reflète parfaitement le fait que les différences sur les ombres sont beaucoup plus facilement perceptibles sans texture masquante. Le deuxième test évalue la perception des différences sur les ombres portées en fonction de la distance de visualisation de l'image. Le résultat est porté sur la figure 5.4. Enfin, le troisième test compare des images issues d'étapes successives du rendu progressif. La figure 5.5 illustre bien que les résultats donnés par la métrique euclidienne ne sont pas satisfaisants : la probabilité de percevoir une différence est, en effet, approximativement constante dans le temps, ce qui ne correspond pas à ce que l'œil perçoit : dans un processus de rendu progressif, il y a généralement beaucoup plus de différences entre les premières itérations qu'entre les dernières.

Réflexions sur le modèle de Daly

Les résultats de la section précédente montrent l'efficacité du modèle de Daly. Il s'avère performant là où cela semble nécessaire en synthèse d'images. Toutefois, son utilisation pose un certain nombre de problèmes. Premièrement, ce modèle ne prend pas en compte la vision des couleurs. Une

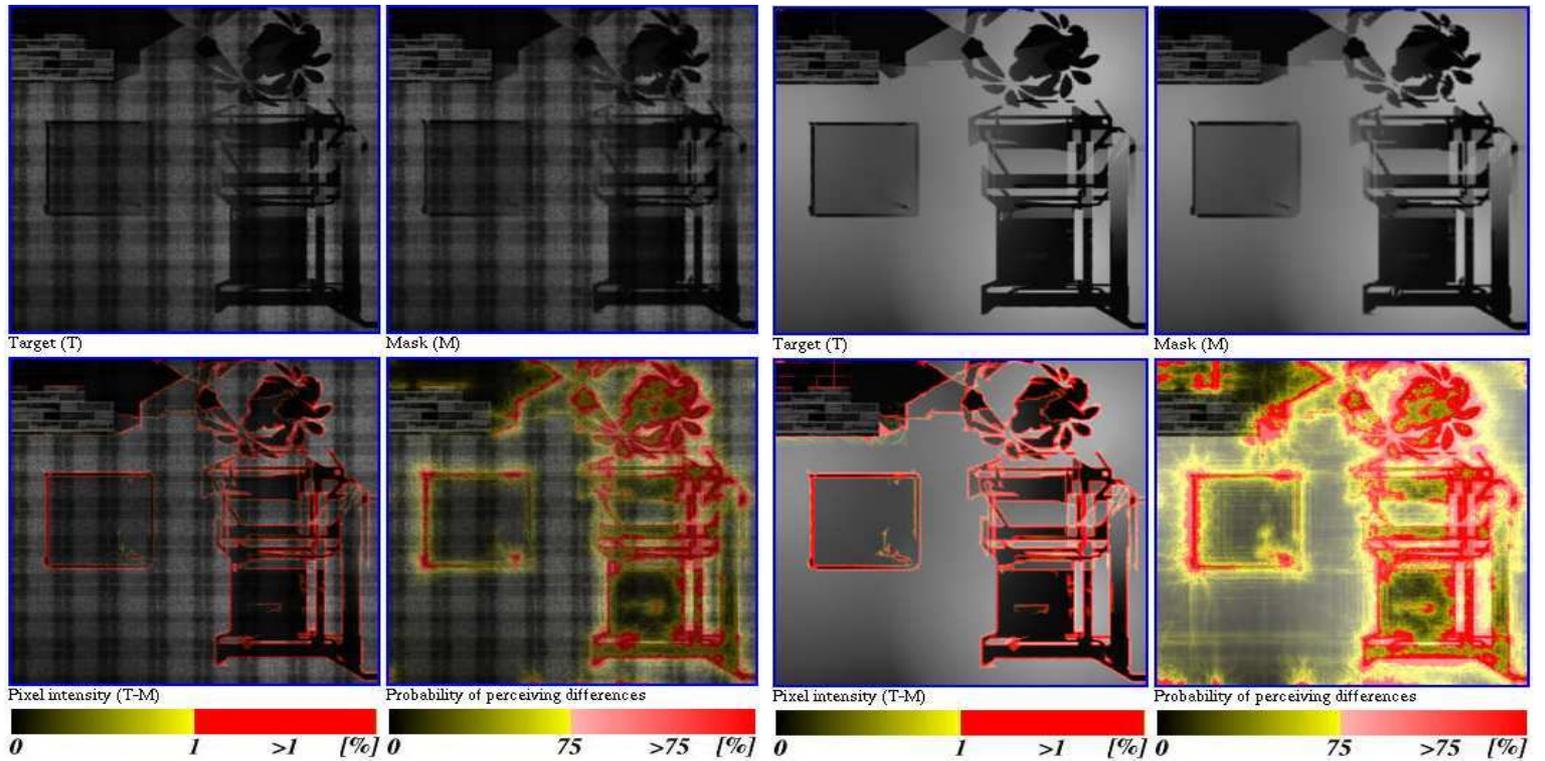


FIG. 5.3 – Test de validation du modèle de Daly (issu de [49])

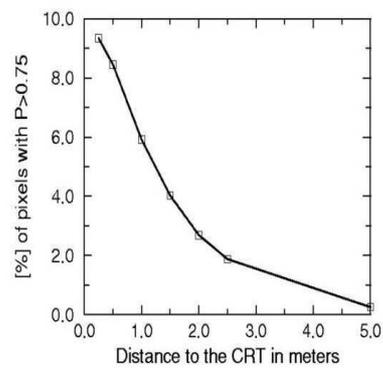


FIG. 5.4 – Probabilité de perception de différences en fonction de la distance d'observation (issu de [49])

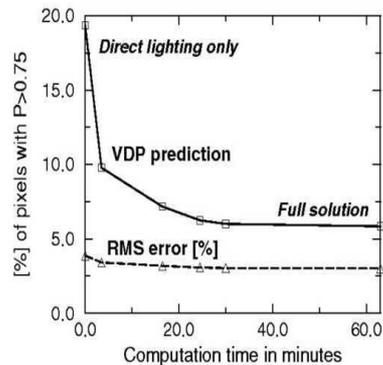


FIG. 5.5 – Probabilité de perception de différences en fonction du temps de calcul(issu de [49])

extension en ce sens pose le problème de l'interaction mutuelle des canaux. Deuxièmement, la multiplicité des canaux lors de la décomposition fréquentielle orientée provoque une consommation de mémoire et des temps de calculs qui sont loin d'être négligeables, et c'est un point particulièrement crucial si l'algorithme doit être intégré dans un processus de rendu. Enfin, la structure elle-même du modèle a été critiquée car elle ne respecte pas le cheminement de l'information dans le système visuel (par exemple, la fonction de sensibilité au contraste est placée avant la décomposition fréquentielle), Daly ayant choisi de placer bout à bout des fonctions psychométriques plutôt que de simuler le fonctionnement de la vision humaine. Ce n'est pas une approche absurde en soi, mais les fonctions utilisées étant issues d'études totalement distinctes et décorellées, l'unité du système résultant n'est pas garantie.

5.2.3 Le "Visual Discrimination Model (VDM)" de Lubin

Structure du VDM

Le modèle de Lubin, le "Visual Discrimination Model" ([35]) va s'attacher à simuler l'acquisition et les traitements de l'information lumineuse, de son arrivée dans le globe oculaire jusqu'aux traitements corticaux de bas niveau. Contrairement à l'approche de Daly, Lubin opte pour un processus se déroulant dans l'espace image. La figure 5.6 montre le schéma de fonctionnement du VDM. Son modèle de vision comporte quatre phases principales :

- Une simulation de l'optique particulière de l'œil et de la rétine : pour cela, l'image est filtrée avec une "Point Spread Function" pour tenir compte des capacités de résolution spatiale des photorécepteurs, puis échantillonnée à 120 pixels par degré.
- Un calcul des contrastes locaux : l'image est décomposée avec une pyramide laplacienne ([9]) à sept niveaux. Chaque pixel de chaque niveau est ensuite divisé par la valeur correspondant au même pixel du niveau supérieur (ce dernier étant au préalable ramené à la bonne taille). Le résultat est approximativement égal au contraste de Weber : $\frac{L_{max} - L_{min}}{L_{moyen}}$
- Une prise en compte de l'orientation : chaque niveau de la pyramide est convolué avec quatre paires de filtres orientés (une dérivée seconde de gaussienne directionnelle et sa transformée de Hilbert). L'énergie de contraste est ensuite obtenue en sommant les résultats de l'application de ces paires de filtres sur chaque niveau.
- Une phase "transducer" : chaque énergie de contraste est normalisée par le carré du seuil de détection de contraste pour la fréquence correspondant au niveau courant de la pyramide. Ces énergies normalisées sont ensuite multipliées par une fonction sigmoïde, nommée transducer,

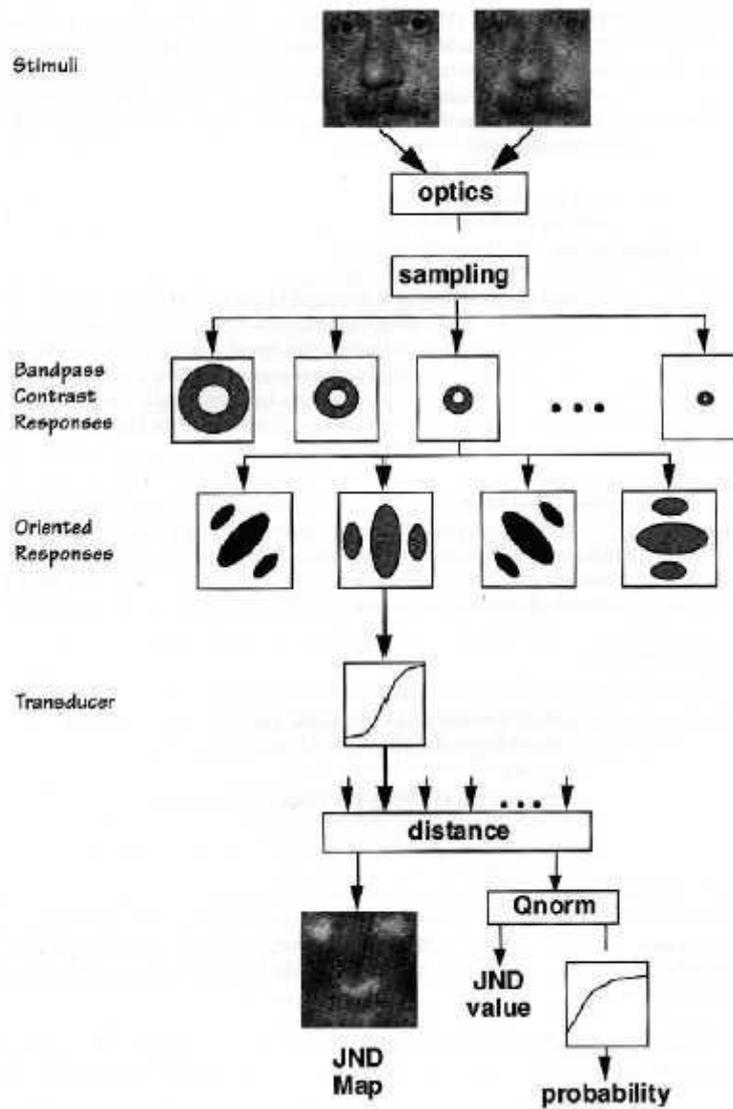


FIG. 5.6 – Fonctionnement du Visual Discrimination Model

similaire à la fonction psychométrique de Daly. Il s'ensuit une simulation de l'excentricité de la fovéa avec une convolution par un noyau circulaire dont le diamètre varie en fonction de la distance au centre de la fovéa.

Lubin considère que la sortie de son modèle est un vecteur à 28 composantes (7 niveaux pour la pyramide, et quatre orientations différentes). Pour la règle d'intégration, il suffit donc d'appliquer le modèle de vision sur les deux images à comparer et de calculer la distance entre les deux vecteurs correspondants. Lubin applique toutefois une variation subtile de la métrique euclidienne : il utilise une puissance 2,4 au lieu de 2.

Résultats et performances

Plutôt que de présenter les résultats du modèle de Lubin, nous allons ici parler de son adaptation au domaine de la synthèse d'images. Dans [7], Meyer et Bolin en ont mis au point une version modifiée et simplifiée qui s'exécute beaucoup plus rapidement que le modèle initial, mais dont les performances sont similaires.

- Ils ont remplacé la décomposition pyramidale par une décomposition en ondelettes de Haar, beaucoup moins volumineuse.
- Ils ont modifié le modèle pour prendre en compte les images en couleurs. Plusieurs modifications ont été effectuées dans ce sens :
 - L'acquisition du signal par les cônes est simulée en utilisant l'espace colorimétrique SML.
 - Un filtrage est effectué sur le canal S pour simuler l'aberration chromatique.
 - L'espace colorimétrique AC1C2 permet de décorer les trois canaux avant les traitements corticaux.

A quelques paramètres près, le reste du modèle est identique. Les auteurs ont également intégré cette métrique dans un algorithme de rendu progressif, nous examinerons leur approche dans le chapitre suivant.

La figure 5.7 montre un test effectué avec cette métrique. Une distortion sinusoïdale est appliquée sur l'ensemble de l'image de la chapelle, mais elle n'est pas visible partout avec la même intensité : sur la partie de l'image comportant de hautes fréquences spatiales, la distortion est presque imperceptible. La carte de visibilité, à droite sur la figure, retranscrit bien cette impression visuelle : les parties noires correspondent aux zones où les différences sont les moins perceptibles.

Réflexions sur le Modèle de Lubin

D'après son auteur, le modèle de Lubin nécessite beaucoup moins de recalibration que la méthode de Daly lorsque l'image change. C'est un avantage certain pour intégrer ce modèle au sein d'un algorithme de rendu, qui lui soumettra intensivement des images pouvant être assez différentes. La structure du modèle semble également plus logique, d'un point de vue physiologique. Enfin, les tests effectués par Bolin et Meyer montrent qu'il semble adapté à la synthèse d'images. Toutefois, la décomposition sur 28 canaux occupe une mémoire plus importante que celle occupée par le modèle de Daly. De plus, le modèle reste essentiellement achromatique, les modifications apportées dans [7] se limitant à une démultiplication indépendante du modèle sur chaque canal. Nous allons voir dans la section suivante un modèle d'apparence couleur plus complet.

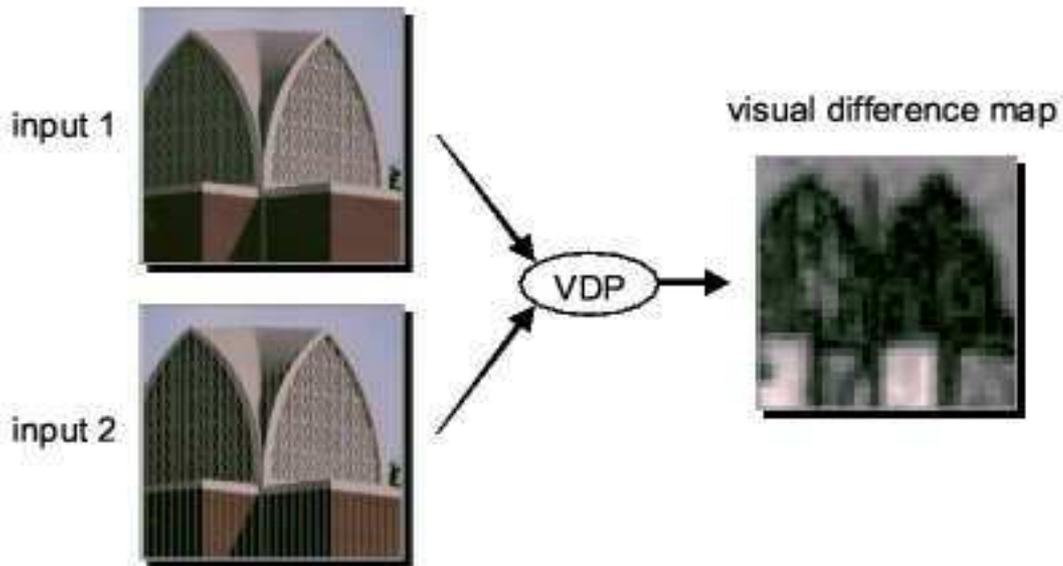


FIG. 5.7 – Test de Meyer et Bolin ([7])

5.2.4 Le "Multiscale Model of Adaptation and Spatial Vision (MMASV)" de Pattanaik *et al*

Ce modèle, présenté par Pattanaik *et al* dans [54], reproduit toutes les étapes du cheminement du stimulus lumineux dans le système visuel humain. Le but ici est d'obtenir des effets que les précédents modèles ne prenaient pas en compte, comme les domaines de vision mésopique et scotopique, l'adaptation chromatique, les changements dans la perception des contrastes en fonction de l'état d'adaptation en luminance, . . . Pour cela, le modèle de Pattanaik *et al* travaille sur quatre canaux (trois canaux chromatiques, plus les bâtonnets) et utilise des fonctions psychométriques évoluées.

Structure du MMASV

La structure du MMASV, visible dans sa globalité sur la figure 5.8, est la suivante :

- Traitements préalables de l'image : Simulation de la dispersion et de la diffraction de la lumière dans le globe oculaire, puis obtention des signaux des cônes et des bâtonnets (intégration par les fonctions de sensibilité spectrales, ou transformation colorimétrique si les données spectrales ne sont pas disponibles),
- Décomposition spatiale et adaptation : le MMASV décompose l'image avec une pyramide de type "différence de gaussiennes", similaire à celle du VDM de Lubin. La différence vient de l'intervention d'une fonction de gain, censée reproduire le travail des cellules ganglionnaires des couches plexiformes de la rétine. Cette fonction de gain augmente ou diminue la normalisation du contraste en fonction de la luminance du fond, représentée dans le modèle par la valeur affectée au pixel correspondant du niveau supérieur de la pyramide.
- Signaux antagonistes : l'espace colorimétrique AC1C2 est utilisé pour transformer l'information couleur SML en signaux antagoniste : A (achromatique), C1 (opposition rouge - vert) et C2 (opposition bleu - jaune).
- Fonctions "transducer" : la sensibilité au contraste n'est pas la même pour tous les états d'adap-

tation en luminance de l'oeil (voir figure 3.11). Ces fonctions modifient donc les contrastes de chaque niveau de la pyramide de manière à prendre en compte ce phénomène. Elles dépendent du canal et de la bande de fréquence (niveau de la pyramide) considérés.

- Unification des signaux des cônes et bâtonnets : Pattanaik *et al* considèrent que les signaux issus des bâtonnets ne contribuent qu'à l'information achromatique ; ils ne sont donc ajoutés qu'au canal A.

Résultats et performances

Le MMASV a été utilisé avec succès dans le cadre d'un algorithme de reproduction de tons dans [55] (voir la section 4.3.2 pour les résultats dans ce domaine). Dans [54], les auteurs ont utilisé la sortie du modèle pour la formater en termes de description de l'apparence couleur : Brillance, Clarté, Coloration, Teinte, Saturation, Chroma. Pour mettre en évidence les performances de leur modèle, les auteurs examinent les brillances obtenues sur trois types de tests.

Sur la figure 5.9, la ligne supérieure représente un test de contrastes simultanés : un carré dont la luminance ne varie pas est placé successivement sur trois fonds de luminances différentes. La ligne inférieure illustre la brillance perçue telle qu'elle est calculée à la sortie du modèle. Sur ces cartes de brillances, il est nettement visible que le carré n'est pas perçu de manière identique sur les trois fonds.

La figure 5.10 montre un test d'assimilation similaire à celui qui est présenté dans le chapitre 3. Une grille uniforme monochrome est coloriée uniquement sur les lignes de manière à dessiner un motif. Même si l'intérieur des carreaux est parfaitement blanc, il est perçu comme étant légèrement rosé. La carte de brillance rendue par le modèle reflète parfaitement ce phénomène d'assimilation.

Enfin, la figure 5.11 prouve que le modèle gère l'adaptation chromatique, surtout utile quand le modèle est utilisé dans un opérateur de reproduction de tons. La même scène est éclairée par trois illuminants distincts : blanc, rouge et bleu. Sur la ligne supérieure, aucune prise en compte de l'adaptation chromatique n'est effectuée. Sur la ligne du milieu, le MMASV est utilisé. Enfin, la ligne inférieure permet de constater le changement d'illuminant sur le cadre gris, dont la présence amoindrit d'ailleurs l'effet d'adaptation chromatique.

Réflexions sur le modèle de Pattanaik *et al*

Les résultats exposés dans la section précédente semblent confirmer que le modèle de Pattanaik *et al* est plus performant que ses prédécesseurs. Il permet la prédiction de nombreux phénomènes pas ou peu considérés jusqu'ici. Mais ces performances ont un prix : le modèle est très exigeant, aussi bien au niveau du temps machine que de la mémoire vive utilisée. Il n'est donc pas utilisable tel quel dans un algorithme de rendu sans effectuer quelques modifications. Nous reviendrons sur ce modèle lors de la section consacrée à notre métrique adaptative.

5.3 Nos travaux sur les métriques perceptuelles

Dans cette section, nous allons présenter notre contribution personnelle. La motivation principale des travaux présentés ici était double : nous voulions une métrique perceptuelle entre images, dotée d'un modèle de vision, à la fois rapide et performante :

- Performante, en terme de nombre de phénomènes pris en compte. La métrique idéale devrait pouvoir gérer correctement les effets de masquage et d'adaptation en luminance. Les phénomènes dus à la vision des couleurs devrait aussi être considérés, même si leur importance est moindre.

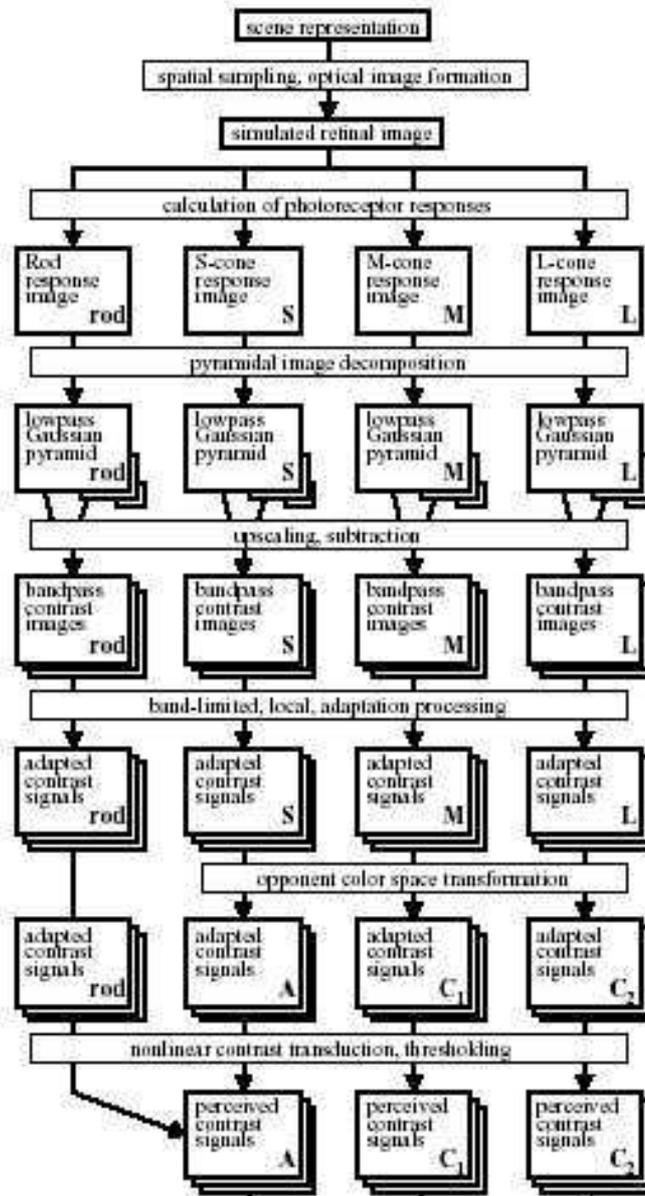


FIG. 5.8 – Le “Multiscale Model of Adaptation and Spatial Vision” ([54])

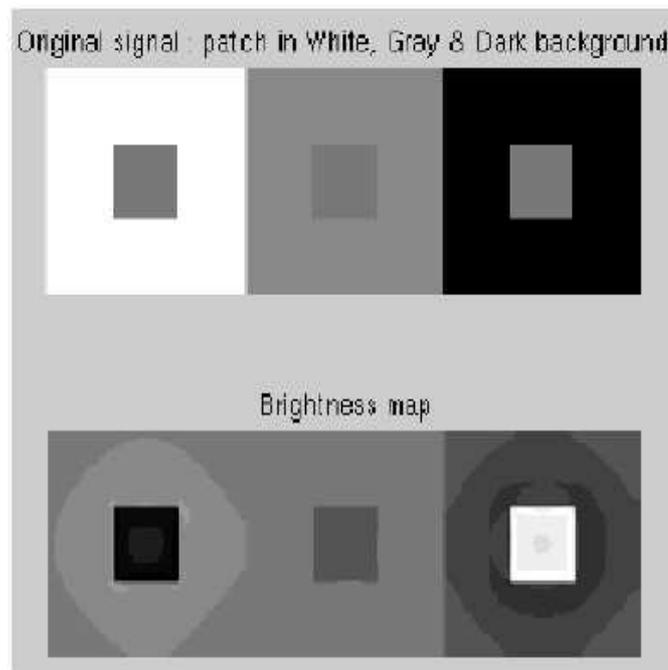


FIG. 5.9 – Résultats du MMASV pour le test des contrastes simultanés

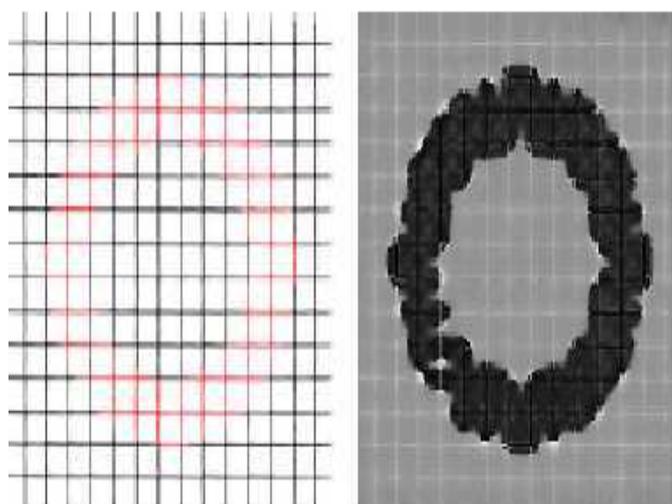


FIG. 5.10 – Résultats du MMASV pour le test d'assimilation

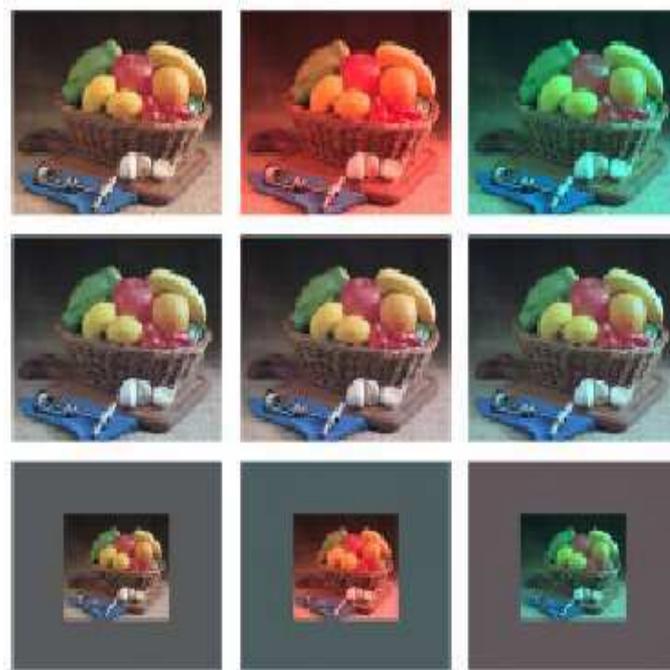


FIG. 5.11 – Résultats du MMASV pour l'adaptation chromatique

- Rapide, car elle doit à terme s'intégrer dans un algorithme de rendu réaliste. Elle ne doit donc pas pénaliser le processus de rendu, déjà très exigeant en terme de calculs.

Nos travaux concernant une métrique visuelle peuvent se scinder en deux parties. Une première version de la métrique est basée sur un modèle simple de Ahumada et Beard ([1]) qui ne comporte pas de transformation multi-échelles. Nous avons ensuite créé une deuxième métrique, basée sur le MMASV de Pattanaik *et al*, qui s'appuie sur un calcul adaptatif de la distance.

5.3.1 La structure de notre métrique "simple"

Pour atteindre l'objectif de performance, il nous faut nous inspirer d'un modèle de vision existant et éprouvé. La finalité du modèle étant d'être intégré dans une métrique perceptuelle, nous avons privilégié les modèles destinés à la détection. Nous avons choisi le modèle de Ahumada et Beard ([1]), que nous avons modifié pour qu'il puisse traiter les images en couleur. Les travaux présentés dans cette section sont la continuation directe de [17].

Pour obtenir une méthode rapide, il nous faut isoler et remplacer ce qui fait la lenteur des métriques précédentes : le fait qu'elles soient multi-échelles. Considérons le MMASV de Pattanaik *et al* : la pyramide gaussienne a sept niveaux, qu'il faut multiplier par quatre canaux, ce qui donne 28 cartes de données, toutes à traiter indépendamment jusqu'à la fin du modèle. Même si les fonctions qui suivent la décomposition pyramidale ne sont pas très complexes, elles ne sont pas linéaires et la consommation CPU, faible pour une seule carte, devient non négligeable sur 28 cartes. Notre premier objectif sera donc de remplacer la transformation multi-échelles par une transformation mono-échelle, peut être plus complexe, mais qui économisera du temps de calcul pour la suite du traitement.

La structure globale de notre modèle de vision simple est visible sur la figure 5.12. Nous allons maintenant détailler et justifier les différentes étapes de ce modèle. Signalons que tous les paramètres de ce modèle, notamment les coefficients et les filtres, sont issus de [1] et sont déterminés par un

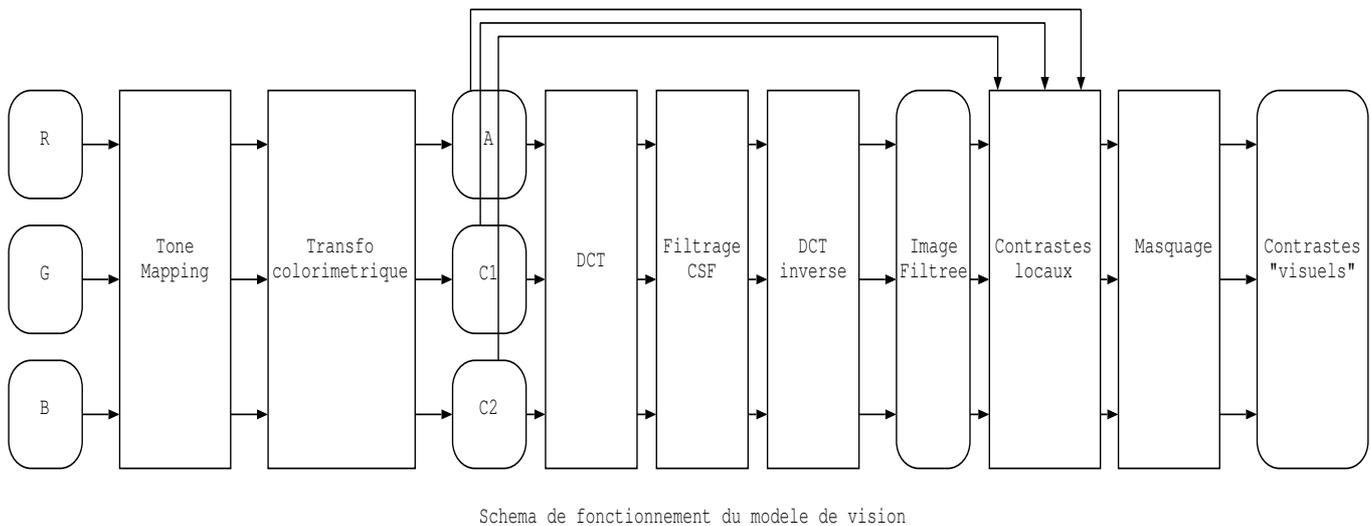


FIG. 5.12 – Notre métrique perceptuelle

ajustement des moindres carrés sur les mesures obtenues pour des expériences de détection diverses.

Étapes préliminaires

Les données fournies en entrée par l'utilisateur sont des images couleur sur trois canaux R, V et B. Chaque canal est une carte de flottants positifs exprimant des luminances en $Cd.m^{-2}$. Les valeurs en chaque pixel peuvent donc être vues comme une représentation spectrale minimaliste sur 3 longueurs d'ondes R,V,B.

Étape 1 : Reproduction de tons

Les données sont généralement issues d'un algorithme de rendu. Les luminances peuvent donc être très importantes, et ne seront pas visualisées directement par l'utilisateur. C'est pourquoi il nous a paru judicieux de placer en tout premier lieu un algorithme de reproduction de tons dans notre modèle. La tâche qui incombe à cet ORT est de déterminer les valeurs de luminances effectivement perçues par l'utilisateur ; en d'autres termes, ce sont les luminances émises par le moniteur. Cela implique, idéalement, de connaître les caractéristiques du moniteur, et donc d'effectuer un calibrage initial. Pour ne pas alourdir la méthode plus que nécessaire, nous avons choisi d'utiliser l'opérateur de Ward ([77]), particulièrement simple puisqu'il ne calcule qu'un seul facteur pour la conversion de toute l'image. L'image est ensuite convertie pour obtenir des coordonnées colorimétriques XYZ.

Étape 2 : Transformation colorimétrique

Notre modèle est un modèle couleur : il faut donc nous préoccuper de cet aspect. Nous avons choisi d'utiliser l'espace colorimétrique AC1C2, proposé par Meyer ([46]) de manière à découpler au maximum les différents canaux pour les traiter indépendamment. Cette transformation colorimétrique nous fournit trois canaux : un canal achromatique, un canal chromatique d'opposition rouge - vert et un canal chromatique d'opposition jaune - bleu. La transformation s'obtient à partir des coordonnées XYZ avec la matrice suivante :

$$\begin{vmatrix} -0,0177 & 1,0090 & 0,0073 \\ -1,5370 & 1,08210 & 0,3209 \\ 0,1946 & -0,2045 & 0,5264 \end{vmatrix}$$

Étape 3 : Filtrage par la fonction de sensibilité au contraste

Pour atteindre notre objectif de rapidité, nous voulions trouver un moyen de remplacer la décomposition multi-échelles inhérente à tous les modèles précédents. Pour ce faire, nous avons choisi une approche fréquentielle : nous utilisons une fonction de sensibilité au contraste dans un espace fréquentiel. Nous nous inspirons ici de la méthode de Neumann *et al* ([51]). Pour remplacer la décomposition multi-échelles, les auteurs utilisent un filtrage par des rectangles dont la taille est déterminée par un processus aléatoire pondéré par une fonction de sensibilité au contraste (voir la section 5.1.3); ce faisant, les auteurs espèrent filtrer l'image en privilégiant les fréquences auxquelles le système visuel répond avec le plus d'intensité. Plutôt que d'utiliser des rectangles, nous avons choisi de pondérer directement les fréquences. Nous avons choisi d'utiliser les fonctions de sensibilité au contraste de Daly [11] (pour le canal achromatique A) et Mullen [47] (pour les canaux chromatiques C1 et C2), dont nous convertissons les fréquences en cycles par pixel, de manière similaire à [51]. Nous appliquons sur notre image I une transformation en cosinus discrète, similaire à la méthode utilisée pour le format JPEG. Cette transformation peut être effectuée sur l'image entière ou par blocs, cette dernière option engendrant une précision moindre de la réponse de la métrique. Dans l'espace des fréquences, nous multiplions chaque fréquence par le coefficient correspondant dans la fonction de sensibilité au contraste. Une fois ce filtrage effectué, nous opérons une transformée en cosinus discret inverse, pour obtenir notre image filtrée F .

Étape 4 : Contraste locaux et énergie de contraste

Cette étape est similaire au calcul des contrastes locaux dans [1] : nous soustrayons l'image filtrée F de l'image initiale I , puis nous divisons le résultat par l'image filtrée F . Pour chaque pixel p , nous effectuons donc :

$$Co(p) = (I(p) - F(p))/F(p)$$

Nous calculons ensuite l'énergie de contraste avec la formule suivante :

$$E(p) = Co(p)^2 \times F_e(p)$$

F_e est un filtre gaussien avec les caractéristiques suivantes :

- Taille du filtre : 25 minutes d'arc
- Diffusion de la gaussienne : 1 minute d'arc

Étape 5 : Masquage

Le masquage des contrastes est pris en compte par la fonction "transducer" suivante, toujours extraite de [1] :

$$V(p) = \frac{Co(p)}{(1 + (7.0.E(p))^2)^{0.5}} \quad (5.1)$$

Cette fonction augmente ou diminue l'intensité du contraste local en fonction de l'énergie de contraste.

Règle d'intégration et obtention de la carte de distance

Nous reprenons ici la métrique de Minkowski utilisée par Ahumada, qui est similaire à celle utilisée par Lubin dans son VDM : les sorties du modèle de vision sont considérées comme des vecteurs ; pour calculer la carte de distance entre deux images, il faut appliquer le modèle indépendamment sur ces deux images afin d'obtenir deux vecteurs, puis calculer la distance entre ces deux vecteurs à l'aide d'une métrique de type Minkowski. Cette métrique est donnée par l'équation suivante :

$$d' = 10.5((V_1(x,y) - V_2(x,y))^4)^{0.25} \quad (5.2)$$

5.3.2 Résultats

Dans cette section, nous allons présenter quelques résultats obtenus avec notre métrique. Les images de synthèse sont réalisées avec le logiciel YART, développé au sein du laboratoire LISSE par Marc Roelens ([58]). Les images de tests "non synthétiques" ont été créées avec le logiciel GIMP. Le tableau 5.3.2 récapitule les temps de calcul pour la métrique perceptuelle et le nombre moyen de JNDs pour chaque test. Les résultats seront présentés sous la forme d'images des différences linéaire : les JNDs sont affichés linéairement en niveaux de gris de 0 à 10 ; toute différence supérieure à 10 JNDs est traduite par un pixel blanc. Nous présenterons, pour chaque test, deux cartes de différences : la première sera obtenue avec une distance euclidienne dans l'espace CIELAB, la deuxième est la carte des différences issue de notre métrique. Tous ces résultats ont été obtenus sur un ordinateur de type PC, doté d'une mémoire vive de 1Go et d'un processeur Intel Pentium 4 cadencé à 2 Ghz.

	CIELAB moyenne	Distance moyenne	Temps
Bach décalé	10,01 JNDs	1,54 JND	4 s
Bach bruité	6,60 JND	1,20 JND	4 s
Masquage "sol HF"	0,25 JND	0,045 JND	5,36 s
Masquage "sol BF"	0,31 JND	0,19 JND	5,35 s
Contrastes simultanés	92,77 JNDs	4,64 JNDs	1,16 s
Diffusion - assimilation	3,03 JNDs	0,24 JND	1,11 s

Test "portrait de J.S Bach"

Nous allons ici vérifier si notre métrique prédit correctement la visibilité des différences sur les portraits de Bach présentés sur la figure 5.1. Ce test est constitué de deux portraits altérés et d'un portrait de référence. Le premier portrait altéré a été légèrement décalé sur la droite par rapport à la référence. Le deuxième a été bruité. Nous allons comparer successivement ces deux portraits modifiés à la référence. Les résultats, visibles sur la figure 5.13, montrent clairement que les cartes de différences obtenues pixel à pixel avec l'espace CIELAB (cf. la colonne de gauche) ne traduisent pas la sensation visuelle éprouvée. Le tableau 5.3.2 nous montre également que la différence moyenne obtenue avec notre méthode est de 1,54 JND, alors que la valeur obtenue avec la métrique euclidienne est de 10,41 pour les portraits décalés. Pour le portrait bruité, la moyenne sur l'ensemble de l'image est de 1,2 JNDs avec la métrique perceptuelle, et de 6,60 avec la métrique euclidienne. Si il peut paraître étonnant que les valeurs de la métrique perceptuelle soient du même ordre pour les deux portraits, il est néanmoins raisonnable de dire qu'au bout d'un certain temps, l'utilisateur devrait se rendre compte que les portraits sont décalés, et cette différence deviendrait alors parfaitement visible. Elle n'a toutefois aucune raison d'être supérieure à la différence obtenue avec le portrait bruité.

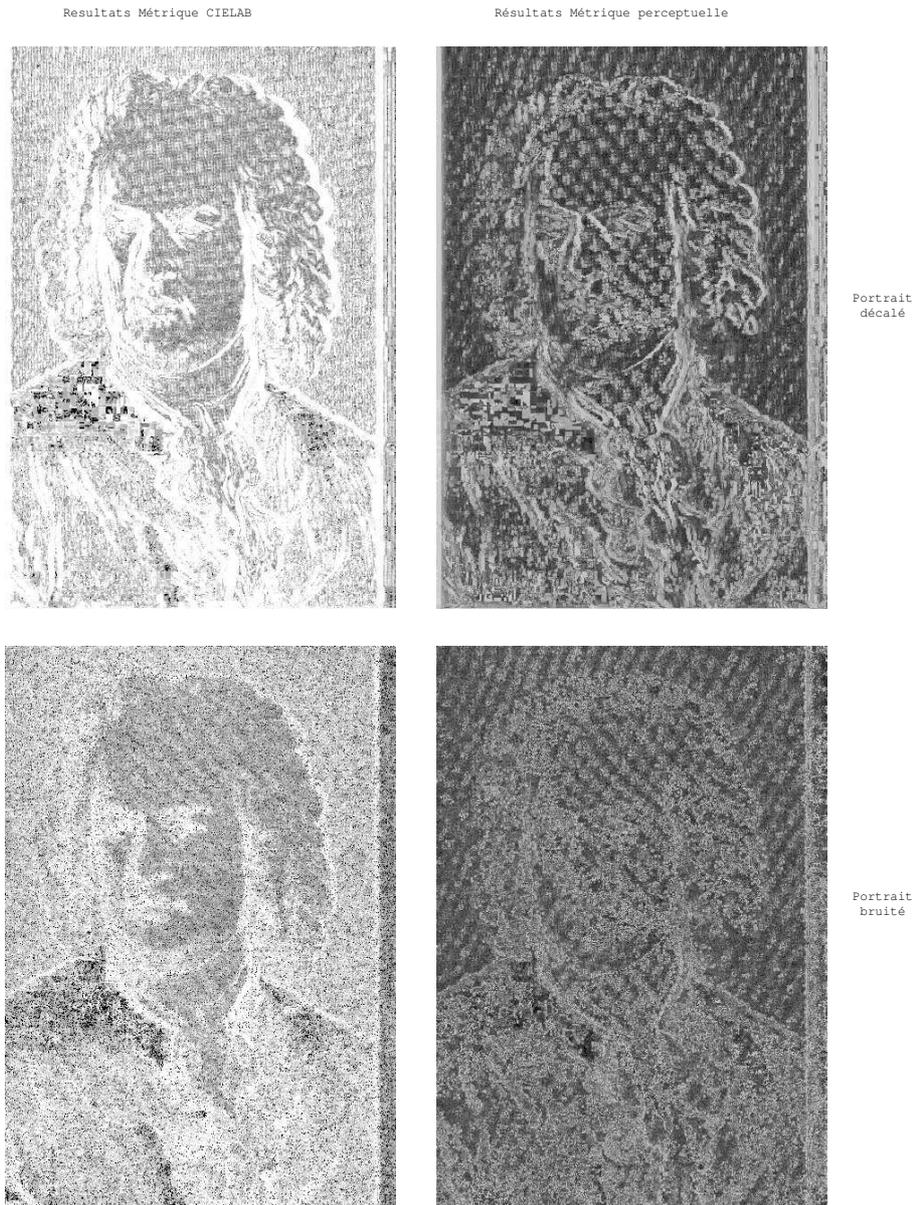


FIG. 5.13 – Résultats du test “Bach”

Test “masquage”

Nous allons maintenant vérifier si notre métrique peut prendre en compte le masquage. Pour cela, nous avons conçu un test à base d'image de synthèse comportant des sources de lumière non ponctuelles. Le principe est le suivant : nous avons deux scènes similaires représentant une série de sphères reposant sur un sol plat ; la seule différence entre les deux scènes étant la texture sur le sol. Pour chacune de ces deux scènes, nous créons deux images en faisant varier le nombre de rayons d'ombre. La source de lumière étant non ponctuelle, les ombres apparaîtront donc très bruitées sur le rendu où ce paramètre est faible. Nous avons choisi huit rayons par pixels dans la première version de la scène, et soixante quatre dans la seconde. L'effet de masquage devrait faire en sorte que le bruit des ombres sur le sol sera nettement moins visible si la texture de ce dernier est très dense. Les résultats sont visibles sur la figure 5.14. La métrique CIELAB ne fait aucune différence entre les bruits quel que soit le cas de figure, ce qui n'est pas le cas avec la métrique perceptuelle.

Test “contrastes simultanés”

Nous allons tester maintenant notre métrique en reprenant le test des contrastes simultanés présenté sur la figure 3.20. Nous avons deux carrés de couleur rouge placés sur des fonds de couleurs différentes, et nous allons comparer ces deux images. Le but ici est de vérifier si la métrique peut prédire la différence de perception induite par la couleur de fond. La figure 5.15 illustre les résultats. La métrique euclidienne basée sur CIELAB ne donne pas ici de bons résultats : en effet, le calcul se faisant au niveau du pixel, le contexte n'est par conséquent pas pris en compte. Sur la carte obtenue avec la métrique perceptuelle, le carré central apparaît légèrement gris, ce qui confirme qu'il y a une différence de perception entre les deux images.

Test “diffusion - assimilation”

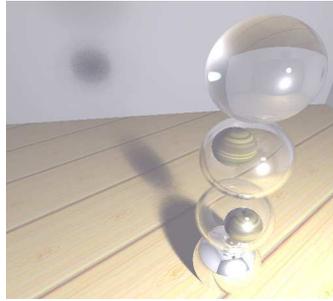
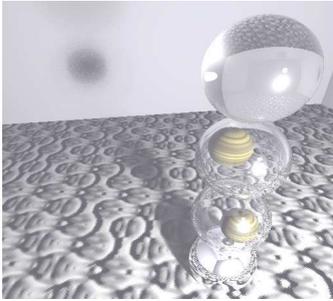
Nous reprenons ici un test similaire à celui de la figure 3.21 : une grille régulière dont on colore les lignes de manière à dessiner un motif. L'intérieur des carreaux est perçu comme étant coloré, alors qu'il est totalement blanc. La figure 5.16 illustre les résultats de ce test, qui démontrent que la carte des différences issue de la distance perceptuelle est plus proche de la sensation visuelle éprouvée que la carte issue d'une métrique euclidienne dans CIELAB.

5.3.3 Réflexions et conclusion sur la métrique

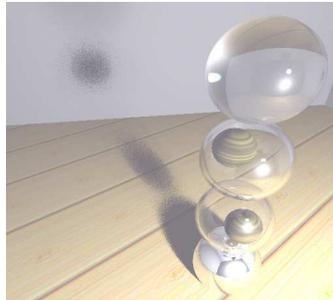
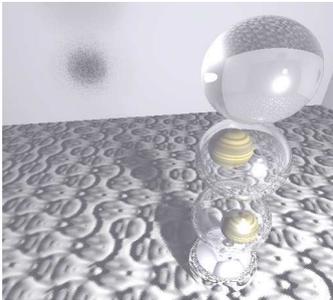
Les résultats sur la métrique présentés dans cette section sont encourageants. Ils traduisent plutôt bien l'impression visuelle attendue, et ce avec un modèle de vision simplifié qui rend les temps de calculs très courts. Des aspects importants comme la couleur, la vision des contrastes sont pris en compte, et seule l'orientation n'est pas traitée. Enfin, les valeurs rendues par le modèle restent dans le domaine du vraisemblable.

Nous nous sommes toutefois posé la question de la pertinence du modèle. Prévu initialement pour prédire la détectabilité de détails sur deux images similaires, nous avons dû le modifier grandement pour l'adapter à nos besoins. Il est évident qu'un modèle plus adapté et plus complet pourrait donner de meilleurs résultats, mais nous avons vu que les temps de calculs d'un modèle classique peuvent vite devenir rédhibitoires. Pour tenter de réduire ce coût, nous avons mis au point une méthode adaptative de calcul de la carte de distance.

Images à comparer

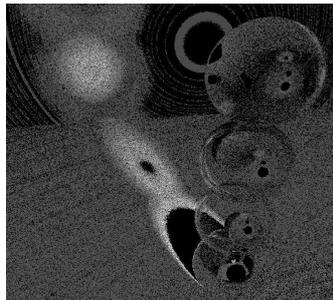
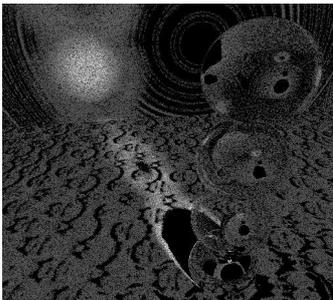


64 rayons par pixel

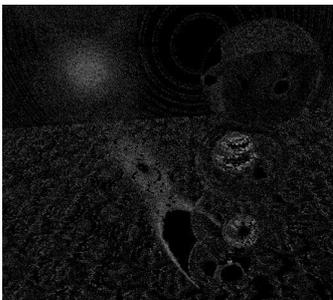


8 rayons par pixel

Cartes des différences



Résultats
métrique CIELAB



Résultats
métrique perceptuelle

FIG. 5.14 – Résultats du test “masquage”

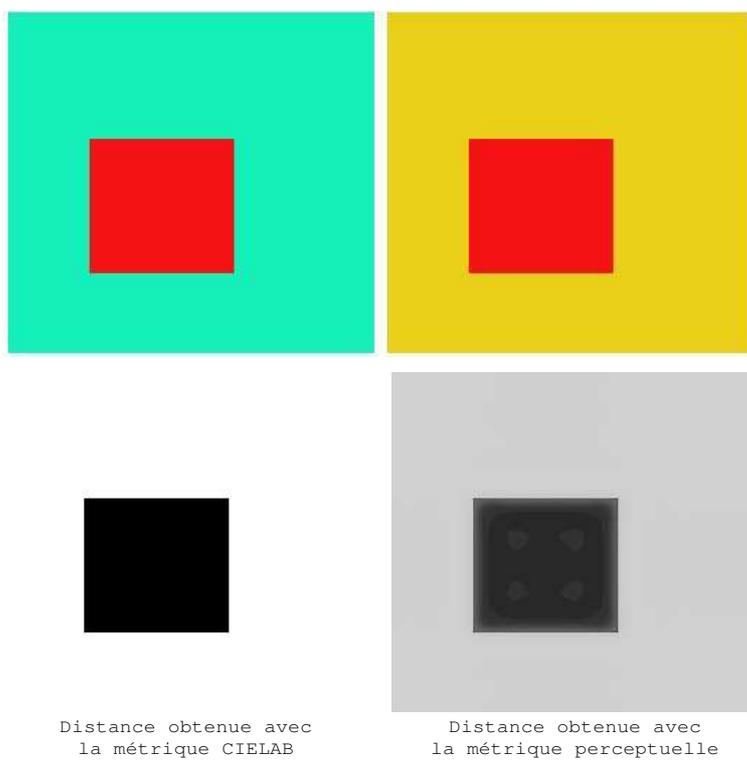


FIG. 5.15 – Résultats du test “contrastes simultanés”

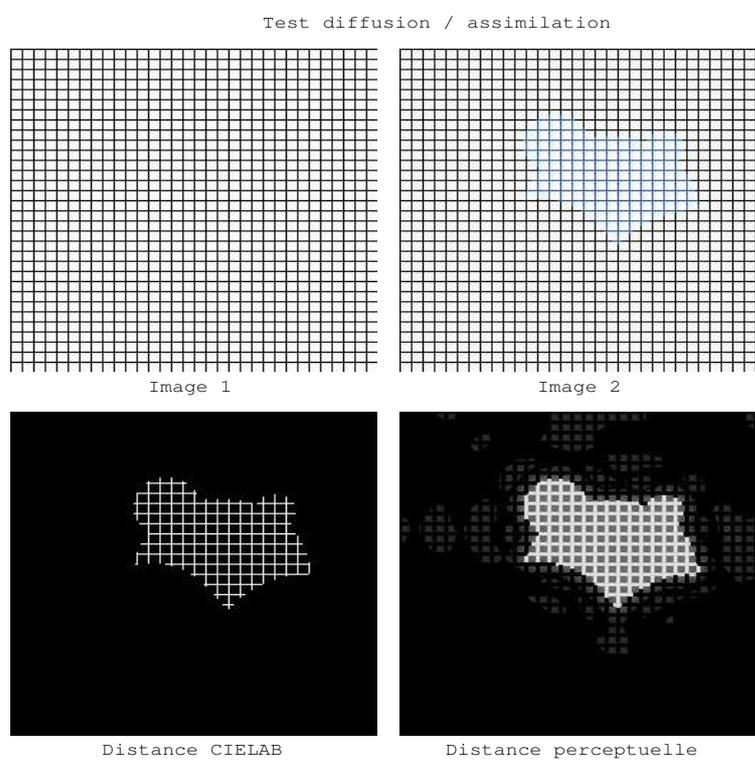


FIG. 5.16 – Résultats du test “diffusion - assimilation”

5.4 Un calcul adaptatif de la distance

Reprenant les travaux effectués par Stéphane Albin [3], nous avons conçu un algorithme qui donne une carte de distance sans faire un calcul de distance pour chaque pixel. Pour cela, nous effectuons ce que Stéphane Albin appelle un “sondage” de l’image, basé sur une subdivision du plan image en quad-tree. Chaque cellule de ce quad-tree contient un ou plusieurs échantillons sur lesquels est appliquée la distance. Nous utilisons un modèle de vision complexe (en l’occurrence [54]) sur les pixels choisis par le “sondage”. En fonction de critères variables, il est alors possible de raffiner le résultat en partitionnant l’image et en recommençant les étapes précédentes sur chaque cellule.

5.4.1 Rappels sur la méthode adaptative de Stéphane Albin

Le “sondage”

De la même manière que pour un sondage classique, nous allons effectuer un calcul sur un “panel représentatif” permettant d’avoir une bonne idée de la valeur réelle recherchée. Le problème est donc de choisir, sur la totalité de l’image, un ensemble représentatif de pixels.

Pour ce faire, nous utilisons une méthode de Monte-Carlo : soit M la moyenne de la distance entre deux images calculée sur un nombre N de pixels :

$$M = \frac{1}{N} \int_0^N distance(p) dp$$

La question est ici de déterminer la valeur de N de manière à ce que cette moyenne calculée soit suffisamment proche de la valeur moyenne réelle. Stéphane Albin a effectué des tests et a montré que 2000 échantillons suffisaient dans la plupart des cas pour obtenir une erreur inférieure à 2%, et ce quelle que soit la taille de l’image pour peu que les dimensions de cette dernière soient assez grandes. L’inconvénient de la méthode de Monte-Carlo est le fait qu’elle ne rend qu’une valeur moyenne là où nous avons besoin d’une carte de valeurs. Il est toutefois possible d’obtenir une telle carte en subdivisant l’image de manière adaptative.

Pour se faire, nous avons procédé de la manière suivante. L’image est partitionnée suivant un arbre quaternaire (“quad tree”), la racine de l’arbre représentant l’image entière (l’état initial de l’algorithme ne comprend que cette racine). Pour chaque cellule du quad tree, un nombre d’échantillons est choisi. Ce nombre est fonction de la taille (produit de la largeur et de la hauteur) de la cellule :

- $N = 2000$ échantillons si la taille est supérieure à 10000.
- $N = 500$ échantillons si la taille est comprise entre 1000 et 10000.
- $N = taille/3$ échantillons si la taille est inférieure à 1000.

Les échantillons sont ensuite positionnés au hasard dans la cellule. Il peut ici être judicieux d’utiliser une suite pseudo-aléatoire uniforme pour la répartition des échantillons dans la cellule. Par la suite, les cellules doivent être subdivisées et rééchantillonnées selon des critères que nous allons expliciter.

Les critères de subdivision

Le critère de subdivision d’une cellule doit être choisi de manière pertinente car il conditionne pour beaucoup la vitesse d’obtention et la qualité de la carte de distances. Dans [3], Stéphane Albin a testé deux critères de subdivision, tous deux plus ou moins basés sur la variance des résultats obtenus pour les échantillons de la cellule.

Le premier critère se propose de calculer une “homogénéité” pour la cellule via une classification des échantillons déterminée selon leurs valeurs. Si 95% des échantillons d’une cellule appartiennent

à la même classe, alors cette cellule sera considérée comme étant homogène. La définition des classes est laissée à l'appréciation de chacun, ce qui peut causer quelques problèmes.

Le deuxième critère est issu des travaux de Jean Luc Maillot (voir [41]). Il s'agit ici de déterminer les zones où la distance est à peu près constante. Cette opération est effectuée en calculant le cardinal de l'ensemble constitué des éléments dont l'écart à la moyenne est inférieur à un certain ε . Si cette valeur, rapportée au nombre d'échantillons de la cellule, est inférieure à un pourcentage fixé, alors la cellule sera subdivisée :

$$Si \frac{Card\{x \in [x - \varepsilon; x + \varepsilon]\}}{|X|} < Pctage\ fixe \Rightarrow subdivision$$

Le choix de la valeur de ε est, là encore, dépendant de l'application et du résultat désiré.

5.4.2 Notre méthode adaptative

Le calcul de la distance

Obtenir un algorithme adaptatif impose le fait d'être capable de calculer une distance pour un pixel donné, et seulement celui-ci. Dans sa méthode, Stéphane Albin utilise une simple distance CIELAB entre les pixels. Pour notre part, nous avons décidé d'utiliser le modèle de Pattanaik *et al* ; en effet, ce dernier serait exigeant si il était appliqué entièrement sur tous les pixels, mais son utilisation adaptative peut s'avérer payante. Il nous faut donc vérifier que ce modèle peut être appliqué en un point précis ; c'est presque le cas, seule la décomposition pyramidale ne peut être effectuée ponctuellement : les masques utilisés sont de taille 5x5, et la pyramide comporte sept niveaux, ce qui signifie que calculer cette décomposition pour un pixel fait intervenir la quasi totalité de l'image. Cette décomposition pyramidale sera donc effectuée en amont, avant le début du processus adaptatif.

Le modèle de Pattanaik présenté dans [54] était utilisé pour prédire différents aspects de l'apparence couleur. Afin d'utiliser ce modèle dans une métrique, nous avons adopté une méthode similaire à celle de Lubin dans [35] : la sortie du modèle est considérée comme un vecteur à 21 composantes (trois canaux couleurs, décomposés sur sept niveaux). Une métrique de Minkowski est ensuite appliquée aux sorties du modèle pour obtenir une carte de distances entre les deux images. Il y a certainement mieux à faire dans ce cas : nous pourrions par exemple utiliser un modèle d'apparence couleur pour convertir la sortie du modèle de Pattanaik *et al* en caractéristiques visuelles plus parlantes, puis comparer ces caractéristiques grâce à une heuristique. Nous avons préféré, pour l'instant, conserver les mêmes méthodes d'obtention que pour les autres modèles.

Notre algorithme

Nous allons dans cette section récapituler et mettre en forme notre algorithme adaptatif de calcul de distance. Notre méthode est quasiment identique à celle de Stéphane Albin, à ceci près qu'un véritable modèle de vision est utilisé en lieu et place de la distance CIELAB. Notre algorithme est le suivant :

- Décomposition pyramidale des deux images à comparer.
- Initialisation de l'arbre quaternaire : la racine est créée, elle correspond à l'image entière.
- Pour chaque cellule :
 - Positionner les échantillons dans la cellule comme indiqué dans la section 5.4.1.
 - Calculer une distance avec le modèle présenté dans [54] pour chaque échantillon.
 - Affecter la moyenne des résultats à la cellule entière.
 - Calculer le critère de subdivision (homogénéité).

	Tps méthode complète	Tps méthode adaptative	Pctage x	Seuil N
Bach bruité	10,02s	4,54s	70%	8,0 JNDs
Masquage	10,8s	4,67s	90%	1,0 JNDs
Salle à manger	5,39s	3,09s	85%	1,0 JNDs

TAB. 5.1 – Récapitulatif des résultats de la distance adaptative

– Si le critère est supérieur à un certain seuil, alors la cellule est subdivisée.

Le critère de subdivision est l’homogénéité de la cellule : le pourcentage d’échantillons dont la distance calculée est inférieure à N JNDs doit être, soit inférieur à $x\%$, soit supérieur à $(100 - x)\%$. Si ce n’est pas le cas, la cellule est subdivisée. Les paramètres N et x sont à ajuster en fonction du cas étudié, c’est un problème sur lequel nous reviendrons dans nos conclusions sur la métrique adaptative. L’algorithme s’arrête lorsqu’il n’y a plus aucune cellule à subdiviser. L’image de distance est alors obtenue de la même manière que pour la distance non adaptative (interpolation linéaire de la carte de distances, avec un seuil à 5 JNDs).

5.4.3 Résultats

Nous allons dans cette section présenter quelques résultats obtenus avec notre distance adaptative. Nous allons également nous attarder sur le rôle que jouent les paramètres dans l’obtention d’un résultat de qualité. Pour chaque test, nous présenterons la carte de distance obtenue avec la méthode adaptative, celle obtenue avec le modèle de Pattanaik complet, ainsi que les cellules de l’arbre quaternaire et les échantillons effectivement calculés (précisons que ces échantillons sont en noir sur les figures). La machine sur laquelle ont été effectués ces tests est un PC équipé d’un processeur Pentium 4 cadencé à 2Ghz et doté d’un gigaoctet de mémoire vive. Le tableau 5.4.3 récapitule, pour l’ensemble des tests, les temps de calcul des images de distance avec les méthodes adaptative et complète, ainsi que les paramètres utilisés pour la distance adaptative.

Résultats du test “Bach”

Nous avons ici comparé le portrait de Bach non modifié avec son homologue bruité. L’image de distance est visible sur la figure 5.17. Le résultat est très similaire à celui obtenu avec la distance non adaptative, pour un temps de calcul deux fois moindre. Quelques explications sur les paramètres choisis : le pourcentage pour l’homogénéité est de 70%, ce qui peut paraître peu élevé au premier abord¹. Cela peut toutefois s’expliquer par le fait que l’image de distance est bruitée, il est donc très difficile d’obtenir des cellules de grande taille avec 95% des pixels dans la même tranche de valeur. D’autre part, il ne faut pas non plus choisir une valeur “seuil de JNDs” trop faible, car la distance moyenne sur cette carte de JNDs étant de 6,28, une valeur de l’ordre de 4 ou 5 aurait pour effet de subdiviser systématiquement presque tous les blocs. Nous avons choisi d’utiliser la valeur 8.0.

Résultats du test “Masquage”

Ce test reprend certaines images du test “Masquage” utilisé pour la distance non adaptative : nous comparons ici les images calculées avec 8 et 64 rayons par pixels, pour la scène avec une texture à fréquence spatiale élevée. La encore, l’image de distance obtenue avec la méthode adaptative présente

¹Une explication plus détaillée sur le choix des paramètres est donnée dans la section 5.4.4

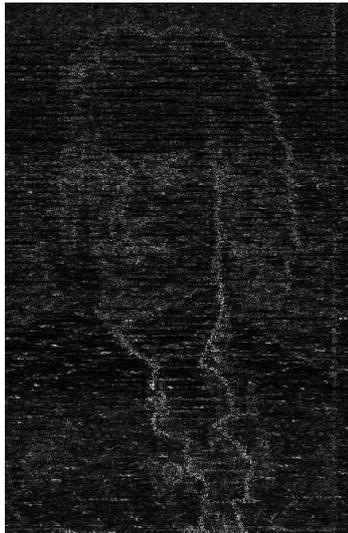
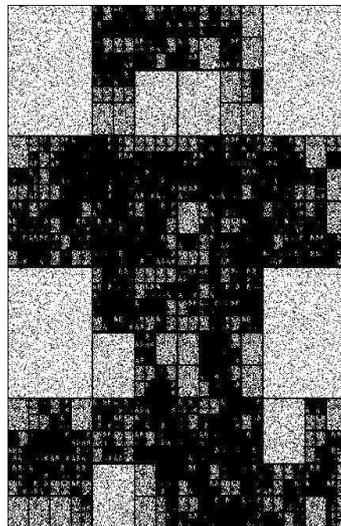


Image de distance
Méthode complète



Image de distance
Méthode adaptative



Subdivision de l'image
et
échantillons calculés

FIG. 5.17 – Cartes des distances pour le test “Bach”

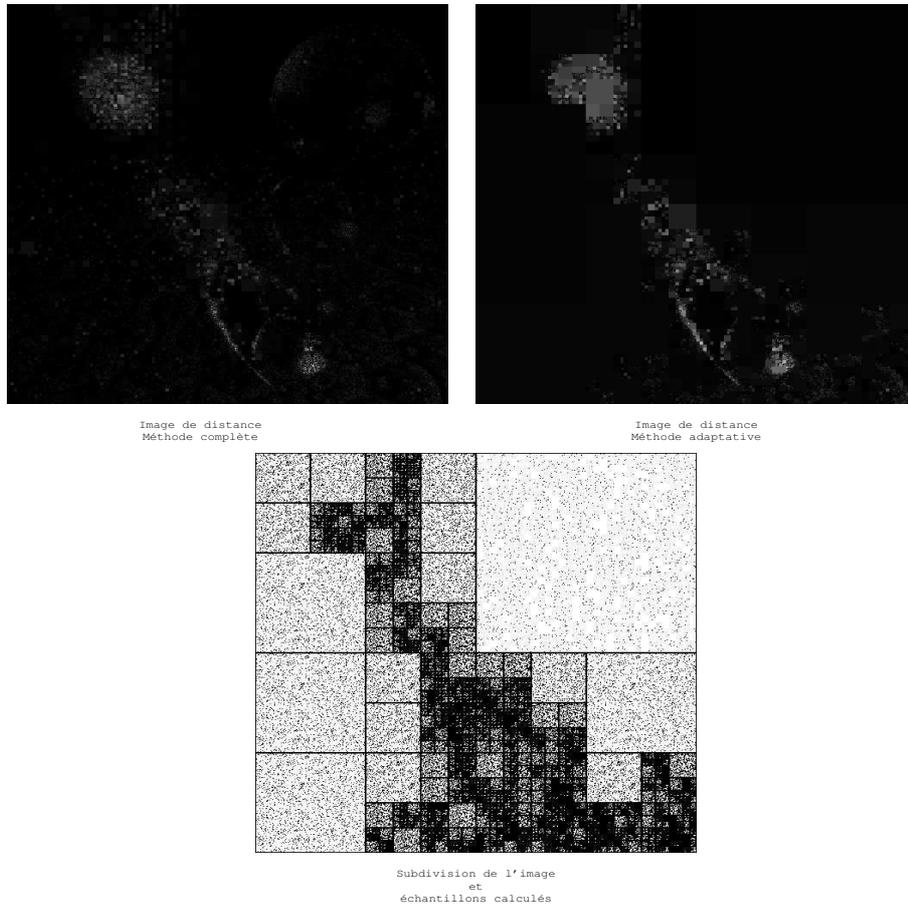


FIG. 5.18 – Cartes des distances pour le test “Masquage”

les mêmes caractéristiques que celle obtenue avec la méthode complète. Les paramètres sont de 90% et 1 JND.

Résultats du tests “Salle à manger”

Nous comparons ici deux images de synthèse représentant une salle à manger. Les deux images (à gauche sur la figure 5.19) sont calculées avec la méthode des vecteurs lumineux introduite par Jacques Zaninetti ([87]). Sur la première image, le calcul est effectué pour chaque pixel, même si l’éclairage indirect est issu d’une approximation vectorielle. Par contre, la deuxième image est obtenue avec un algorithme progressif présenté par Guo dans [24]. Cet algorithme se propose d’approximer l’image finale en ne calculant effectivement qu’une faible proportion des pixels de l’image ; il est basé sur la cohérence directionnelle de l’information dans l’espace image. Nous reviendrons dans le chapitre 6 sur la méthode des vecteurs lumineux et l’algorithme de Guo.

Les paramètres sont fixés à 85% et 1.0 JND. Dans cette configuration, la figure 5.20 démontre que les cartes complète et adaptative sont très similaires. De plus, la carte des échantillons montre qu’ils semblent placés dans les zones non uniformes. Toutefois, les performances en terme de temps de calcul sont moindres que sur les autres images. Ceci s’explique sans doute par le fait que pour obtenir un nombre suffisant de détails significatifs sur l’image, un grand nombre d’échantillons soit nécessaire.



FIG. 5.19 – Images comparées lors du test “Salle à manger”

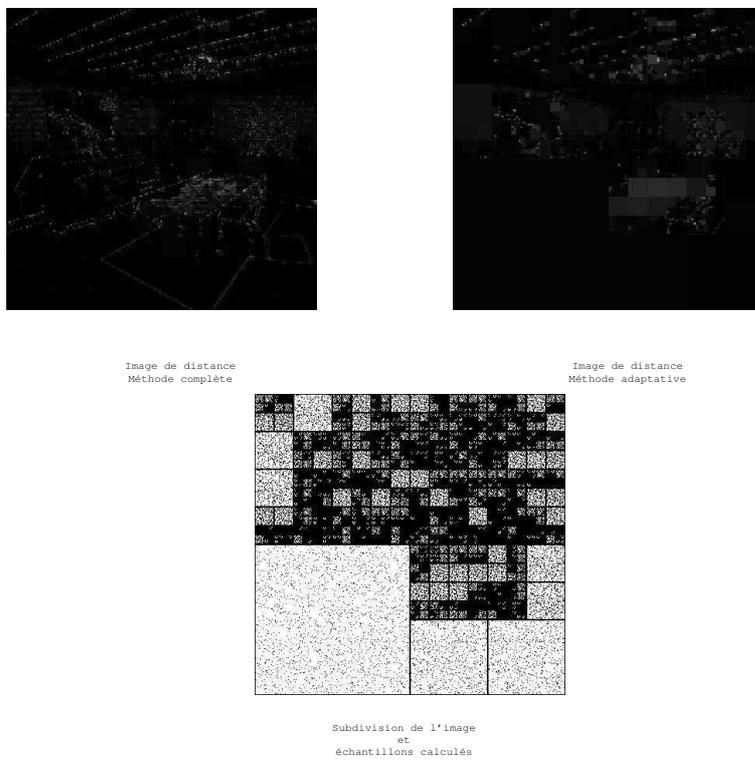


FIG. 5.20 – Cartes des distances pour le test “Salle à manger”

Pour ce test, nous avons choisi d'effectuer un essai sur la variation des paramètres afin d'observer leur influence sur le résultat. La figure 5.21 montre l'évolution de la carte de distance obtenue pour des paramètres variant de 65% à 95% pour le pourcentage d'homogénéité et de 1,0 à 3,0 pour le seuil en JNDs. Les résultats obtenus nous conduisent à discuter sur le choix des paramètres dans la section suivante.

5.4.4 Réflexions sur la métrique adaptative

Le choix des paramètres

La métrique adaptative, basée sur le modèle de Pattanaik *et al* donne des résultats satisfaisants, et ce dans des temps presque systématiquement deux fois inférieurs à ceux du modèle complet. Cela impose toutefois de choisir les "bons" paramètres. Malheureusement, il n'y a pas à proprement parler de méthode infaillible pour obtenir un résultat optimal. Toutefois, en examinant le rôle de ces paramètres, nous pouvons déterminer des valeurs qui devraient aboutir à une image de distances correcte.

Le premier paramètre est le pourcentage conditionnant l'homogénéité ou la non homogénéité de la cellule. Il est aisé d'imaginer que lorsque la carte de distance est très bruitée, aucune cellule de taille supérieure à quelques dizaines de pixels ne sera très homogène. En conséquence, si le pourcentage choisi est trop élevé, toutes les cellules, ou presque, seront subdivisées, ce qui entraîne une chute de performance. Nous recommandons donc :

- de choisir une valeur de l'ordre de 90% si les deux images ne sont pas (ou peu) bruitées.
- de choisir une valeur plus faible si ce n'est pas le cas.

Le deuxième paramètre est le seuil en JNDs qui permet de calculer le pourcentage de pixels de la même "classe" associé à une cellule. Si cette valeur est trop importante lors de la comparaison de deux images assez similaires, la subdivision va s'arrêter très tôt et le résultat ne sera pas significatif. L'idéal serait de choisir, pour ce paramètre, une valeur proche de la moyenne des JNDs de la carte de distances ; cette valeur nous est évidemment inconnue, mais nous pouvons là encore anticiper : il est possible d'évaluer grossièrement "à l'oeil" la différence entre les deux images en tenant compte du fait qu'une JND de valeur 1 signifie que la différence est peu visible, alors qu'une JND de valeur 5 traduit une différence très nette. Nous recommandons donc :

- de choisir une valeur comprise entre 1 et 4 si les deux images sont peu ou assez peu différentes.
- de choisir une valeur comprise entre 5 et 10 si les deux images sont très différentes.

Surcoût lié au calcul adaptatif

D'autre part, nous avons cherché à évaluer le surcoût provoqué par le processus adaptatif ainsi que les structures de données qui lui sont associées. Il est en effet légitime de se demander dans quelle mesure le fait d'introduire un arbre quaternaire et une subdivision adaptative alourdit le temps de calcul de la distance. Si ce processus s'avère trop coûteux, la solution risque de ne pas être viable. Nous nous sommes donc, dans un premier temps, assuré que la solution adaptative converge bien vers la solution non adaptative si l'on prend le soin de régler les paramètres à $x = 100\%$ et $N = 0$ JND. Nous avons ensuite comparé les temps mis pour calculer la carte de distance entière avec la méthode adaptative et la méthode non adaptative, et ce sur sept tests identiques mais de taille d'image décroissante. Le tableau 5.2 récapitule ces résultats et il semblerait que le surcoût soit indépendant de la taille. Le temps moyen pour la distance adaptative est environ 1,3 fois plus long que le temps mis par la distance non adaptative.

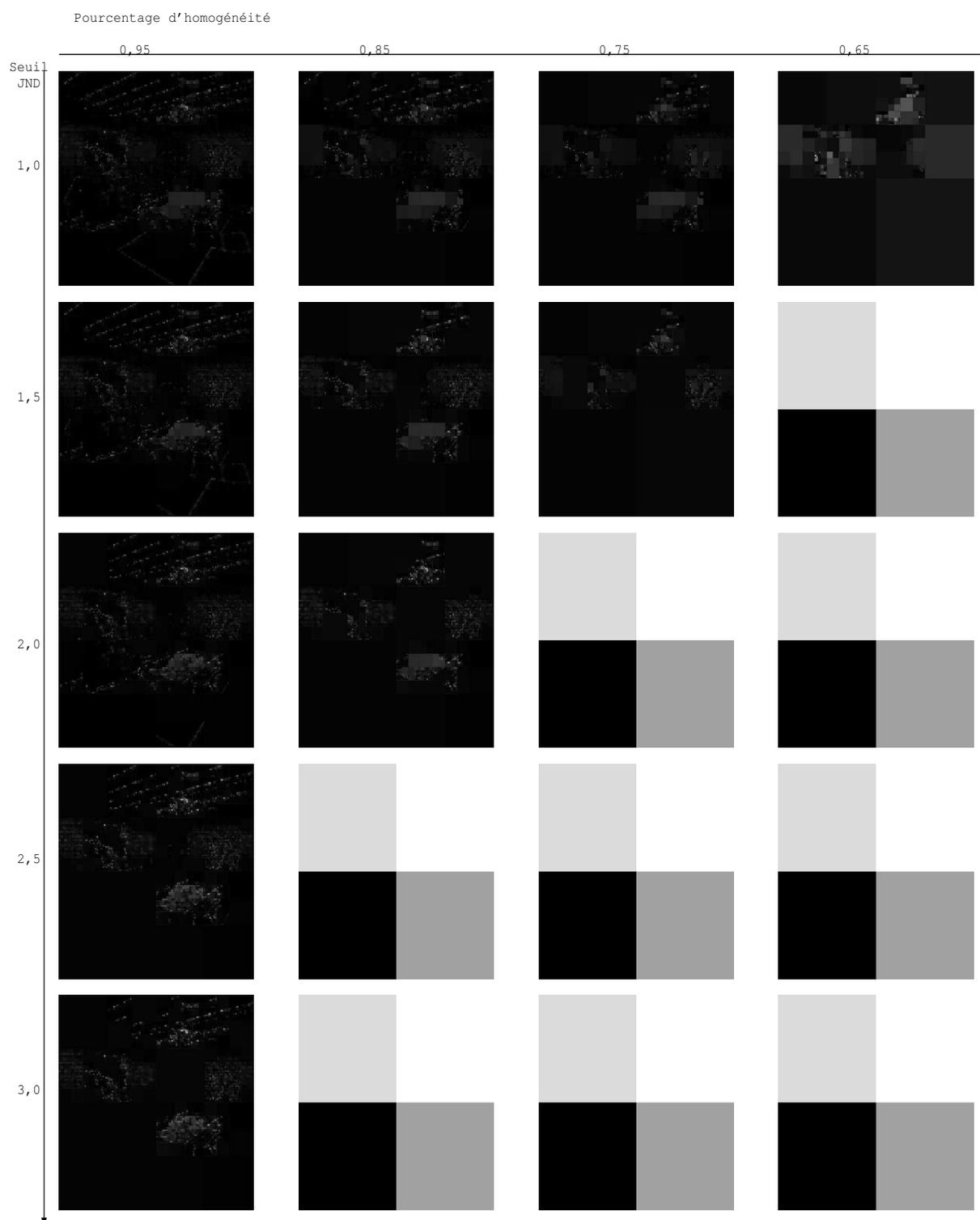


FIG. 5.21 – Résultats obtenus pour la scène de la salle à manger en faisant varier les paramètres de la distance adaptative

	Temps méthode adaptative	Temps méthode complète	Rapport
Test 1	20,45s	14,71s	1,39
Test 2	14,36s	12,54s	1,14
Test 3	11,09s	9,63s	1,51
Test 4	9,79s	7,71s	1,26
Test 5	7,94s	6,3s	1,26
Test 6	6,68s	5,04s	1,32
Test 7	5,13s	3,58s	1,43

TAB. 5.2 – Tableau récapitulatif des temps de calcul pour l'évaluation du surcoût de la méthode adaptative dans le cas d'une image de synthèse

5.5 Tentative de comparaison entre différentes métriques entre images

Bien que les métriques entre images commencent à se répandre dans le domaine de la synthèse d'images, les recherches visant à démontrer la viabilité de ces métriques sont encore rares. Comment, en effet, juger autrement que par une appréciation subjective, le fait qu'une carte de distances corresponde bien aux différences perçues ? Ce problème en appelle également un autre : est-il possible de comparer deux métriques censées simuler le comportement de la vision humaine ? A priori, si leur données de sortie sont exprimées en JNDs, il suffirait de comparer ces sorties. Il nous est toutefois difficile de comparer nos méthodes avec celle existantes, car il nous faudrait retrouver les mêmes tests, effectués dans les mêmes conditions. De plus, le détail de l'implémentation d'un modèle de vision n'est pas toujours complètement détaillé dans la littérature courante. Dans [49], l'auteur fournit une adresse internet où il est possible de récupérer les tests qu'il a effectués avec le modèle de Daly, ainsi que les résultats de ces tests. Dans cette section, nous allons donc comparer trois métriques : notre métrique non adaptative, basée sur le modèle de Ahumada et Beard ([1]), notre métrique adaptative, basée sur le modèle de Pattanaik([54]), et le VDP de Daly ([11]), en utilisant les tests fournis par Myszkowski. Précisons d'emblée que nous n'avons pas reproduit le VDP, nous utiliserons ici les résultats obtenus par Myszkowski.

Ces tests, déjà évoqués dans la section 5.2.2, étudient la perception des différences sur des ombres portées éventuellement masquées par une texture ajoutée sur l'image. Nous allons, dans cette section, étudier les réponses de trois métriques différentes sur trois cas :

- Pas de texture de masquage.
- Texture de masquage de type damier et de taille 4×4 .
- Texture de masquage de type damier et de taille 64×64 .

La figure 5.22 représente ces trois tests, et la figure 5.23 montre les résultats issus de chaque métrique : le VDP de Daly, notre métrique non adaptative issue de Ahumada et Beard, et notre métrique adaptative. Ces cartes ont toutes été obtenues de la même manière que celle de Myszkowski dans [49]. Chaque carte représente la probabilité de détection d'une différence par un code de couleur : ce code varie du noir au jaune de 0 à 0,75, puis du rouge clair au rouge foncé de 0,75 à 1,0. Il est très visible sur ces cartes que la comparaison de méthodes différentes est très difficile, notamment à cause de la subjectivité dans l'interprétation des résultats. Les 3 cartes de rendu paraissent en effet "plausibles". Il semble toutefois évident que notre métrique non adaptative manque un peu de précision par rapport aux deux autres, principalement dans les zones où la probabilité de détection d'une différence est faible (inférieure à 0,75).

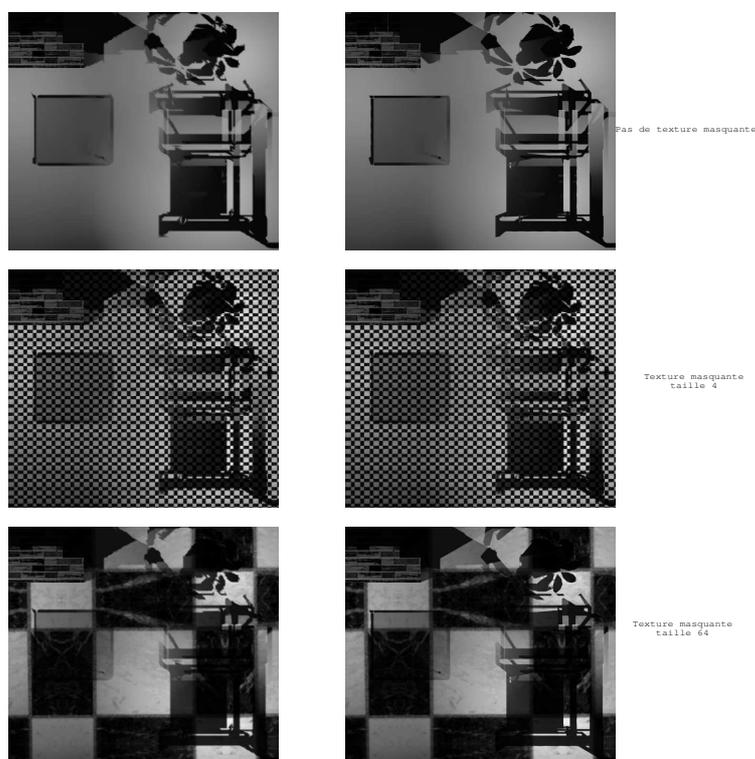


FIG. 5.22 – Tests utilisés pour la comparaison des métriques

5.6 Conclusion et perspectives pour les métriques entre image

Malgré toutes les difficultés que cela peut présenter (difficulté de validation des résultats, choix du modèle entre autre), nous avons montré qu'il est possible d'obtenir une métrique basée sur un modèle de vision qui ne soit pas trop exigeante pour la machine tout en donnant des résultats corrects. Cela permet, entre autre, de commencer à comparer des images de synthèse évoluées avec des photographies de manière plus rigoureuse qu'un simple jugement visuel. Il reste toutefois encore beaucoup de voies à explorer.

Nous pourrions par exemple utiliser des données spectrales avec une véritable simulation de la rétine. Chaque cellule photoréceptrice aurait alors une sensibilité spectrale propre, et l'intégration du signal spectral se ferait donc de manière plus fine qu'actuellement, où il repose essentiellement sur la définition de l'observateur standard.

Les traitements corticaux sont également à peine effleurés dans les modèles actuels. Il existe actuellement des travaux sur la formation et l'interprétation de la couleur au niveau neuronal (voir [76]) dont il serait intéressant de s'inspirer. Ces études pourraient mener à la modélisation du stimulus telle qu'il est représenté au niveau cortical, et permettrait une nouvelle fois de ne plus se reposer uniquement sur des mesures psychovisuelles. D'une manière plus générale, mais aussi plus ambitieuse, une véritable modélisation du système visuel pourrait être effectuée non pas en effectuant des expériences pour déterminer comment réagit un sujet standard à un stimulus précis (qui, la plupart du temps, est assez éloigné d'une situation réelle), mais en simulant un à un le fonctionnement physiologique de tous les composants du système visuel. Cela peut paraître ambitieux, mais ce n'est pas aussi irréalisable que l'on pourrait le croire au premier abord. La rétine, par exemple, commence à être assez bien connue (voir [26], par exemple), et des composants en silicium simulant son fonctionnement existent

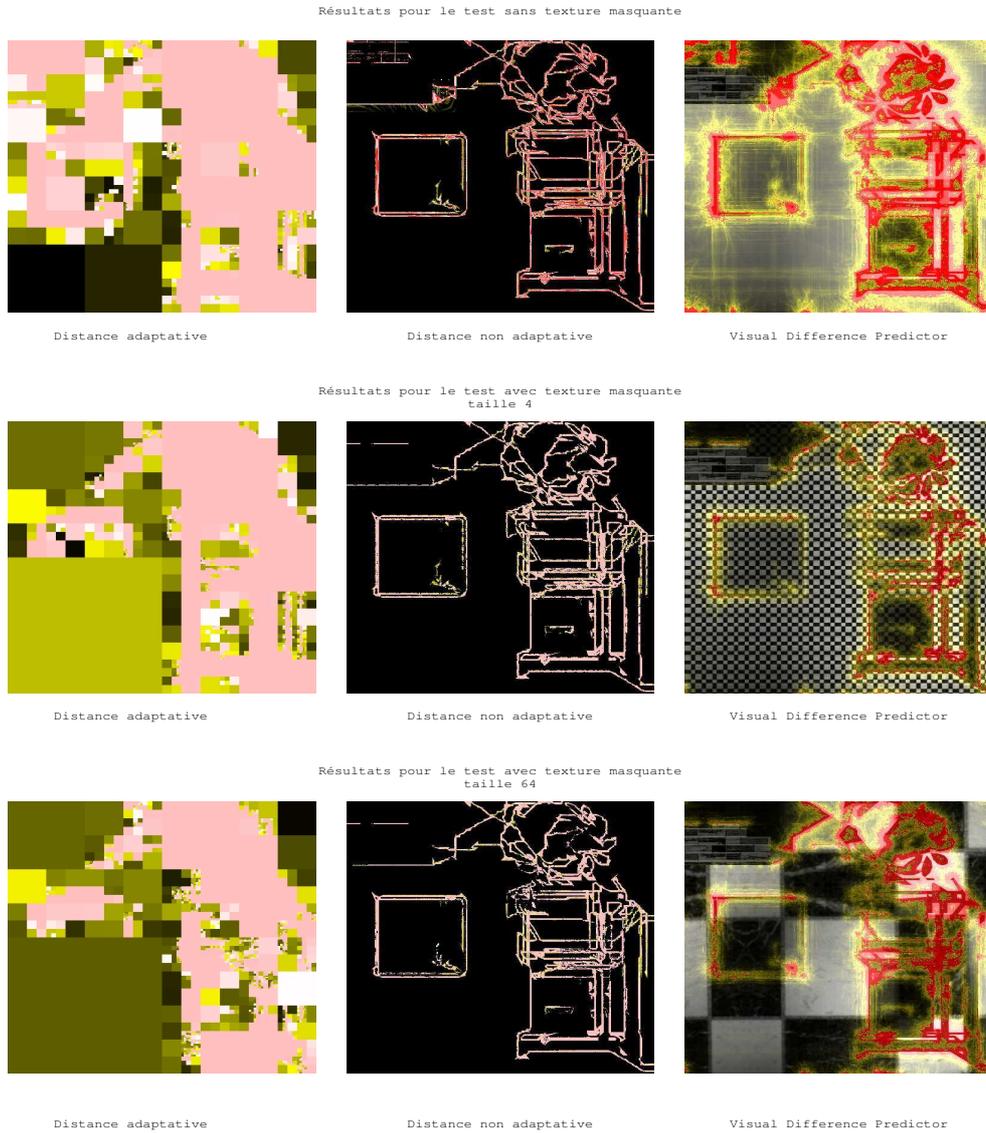


FIG. 5.23 – Résultats des tests de comparaisons des métriques

déjà.

Enfin, les images perçues par le système visuel ne sont pas nécessairement statiques : il pourrait être utile de comparer deux vidéos pour établir une métrique de qualité. Les travaux dans ce sens sont abondants (voir [79], [73]), mais encore assez peu utilisés. Ils reposent sur des études psychovisuelles de la perception du mouvement, qui pourraient s'avérer utiles en synthèse d'image dans le cadre de calcul d'animations complexes. Par exemple, si un objet de la scène bouge très vite devant la caméra, il ne sera pas nécessaire de modéliser ses moindres détails car ces derniers ne seront pas perceptibles. Signalons également qu'au delà de la simple perception cinétique, un mouvement rapide peut aussi engendrer une sensation de couleur : des motifs monochromes judicieusement placés sur un disque en rotation à haute vitesse (disque de Benham) provoquent l'apparition d'une multitude de couleurs sur ce disque. Aucun modèle, à l'heure actuelle, n'est en mesure de simuler ce phénomène.

Une dernière question se pose enfin : comment utiliser une métrique entre images dans le domaine de la synthèse d'images ? Il ne s'agirait plus dans ce cas de comparer deux images, mais de guider un algorithme de rendu de manière à ne calculer que ce qui sera perceptible, et ce afin d'économiser du temps de calcul. Ce problème va faire l'objet du chapitre qui va suivre.

Chapitre 6

Rendu progressif et perceptuel

Notre objectif est maintenant d'utiliser nos connaissances en perception visuelle pour les appliquer en synthèse d'image. De nos jours, et malgré la puissance grandissante des machines, le calcul d'une image de synthèse photo-réaliste reste un processus extrêmement coûteux : il n'est pas rare de devoir patienter plusieurs heures. En effet, la sophistication des algorithmes de rendu croît en fait aussi rapidement, si ce n'est plus, que les performances des ordinateurs ! En 1980, date à laquelle la première image calculée en tracé de rayon a été présentée, la communauté scientifique s'extasia devant le réalisme des divers reflets et transparences d'une boule de verre posée sur un damier. Il est aujourd'hui possible d'obtenir une telle image pratiquement en temps réel, et elle semblerait bien peu réaliste aux yeux d'un spécialiste. En effet, les exigences sont désormais beaucoup plus élevées : une image de synthèse évoluée comportera un calcul de l'éclairage global, des caustiques, des milieux participants, des effets visuels (éblouissement ou halo), des particules, des éléments non nécessairement solides ... La liste est encore longue, et il n'est pas encore ici question d'animation ! Devant la demande, toujours grandissante, d'obtention d'images photoréalistes dans des temps très courts, voire interactifs, il est impératif de trouver des méthodes capables de réduire le temps de calcul. Dans ce chapitre, après une présentation plus détaillée du processus de rendu, nous allons en premier lieu présenter les algorithmes de rendu progressif, dont le but est de reconstituer l'image à partir d'une poignée d'échantillons judicieusement choisis. Nous allons ensuite évoquer les méthodes de rendu perceptuel, qui tentent de prendre en compte le système visuel humain pour calculer "au plus juste". Enfin, nous terminerons en présentant notre contribution, dont la finalité est la réunion des deux approches précédentes.

6.1 Le rendu d'une image de synthèse

6.1.1 Définition

On appelle rendu la phase du processus de création d'une image de synthèse qui consiste à convertir la description géométrique et physique complète de la scène en une image bidimensionnelle, visible sur un dispositif de reproduction (écran, imprimante ou autre). Ce rendu peut être plus ou moins long selon la complexité de la scène, aussi bien au niveau géométrique (objets très détaillés de forme complexe, par exemple) que physique (sources lumineuses non classiques, définition des matériaux évoluée ...). Il existe de nombreuses méthodes permettant le rendu, mais nous n'en détaillerons qu'une dans ce mémoire : il s'agit du lancer de rayons, utilisé par le logiciel YART du laboratoire LISSE, écrit par Marc Roelens([58]).

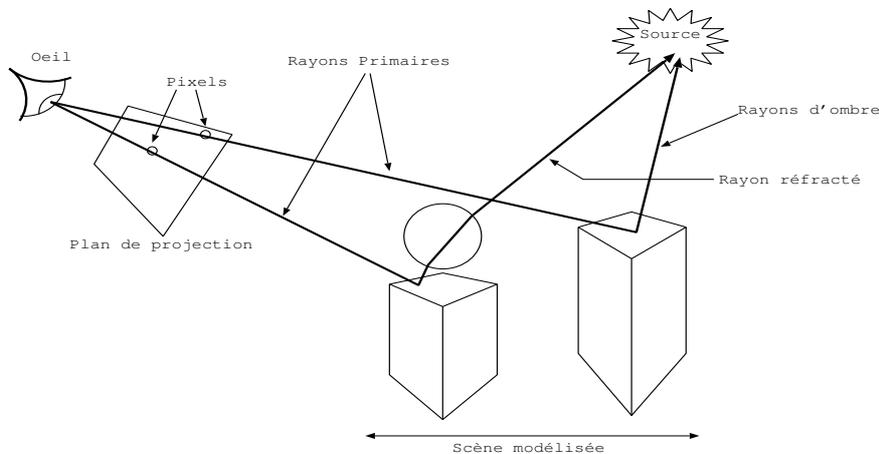


FIG. 6.1 – Schéma de principe du lancer de rayons

6.1.2 Le lancer de rayons

La méthode du lancer de rayons s’inspire de l’optique géométrique. En ramenant l’énergie lumineuse à la notion de rayon rectiligne émanant d’une source, le problème est de représenter le cheminement de la lumière à travers la scène. Concrètement, cela voudrait dire lancer une infinité de rayons depuis les sources lumineuses de la scène, interpréter correctement les interactions de ces rayons avec les différents obstacles rencontrés, pour finir dans l’œil de l’observateur où la lumière est collectée et transformée en image. Cette méthode toutefois pose un sérieux problème : si la scène est très grande et/ou très complexe, la probabilité qu’un rayon partant de la source arrive dans l’œil est infime. Il faudrait donc lancer un très grand nombre de rayons depuis la source pour avoir la chance de voir se former une image.

Dans un souci d’efficacité, Appel a proposé dans [4] une méthode dite “inverse” : partant du principe de retour inverse de la lumière, il suggère de suivre les rayons non pas des sources vers l’œil, mais de l’œil vers les sources de lumière. Destinée à l’origine à l’élimination des faces cachées, mais impraticable, par manque de moyens techniques, à l’époque de sa publication, cette méthode fut reprise par Whitted dans [81] pour obtenir une nouvelle méthode de rendu simple mais performante, dont les caractéristiques permettent, entre autres, d’obtenir des images de scènes faisant intervenir des matériaux non opaques, ce qui était impossible avec les autres méthodes connues jusqu’alors. Nous allons maintenant examiner en détail la méthode et les éléments intervenant dans l’algorithme de Whitted.

La figure 6.1 illustre graphiquement cette méthode de rendu. Un rayon dénommé primaire part depuis l’œil, son origine. Sa direction est donnée par la droite reliant l’œil au pixel du plan de projection dont on veut calculer la couleur (pour avoir une image entière, il nous faudra donc au moins un rayon par pixel). Ce rayon, lors de son arrivée dans la scène, va éventuellement rencontrer des objets. Lors de la première intersection avec un objet de la scène, il faut vérifier la visibilité des sources de lumière depuis ce point d’intersection en lançant un rayon d’ombre vers les sources. En connaissant cette visibilité, le modèle utilisé pour le(s) matériau(x) de l’objet rencontré et les caractéristiques des sources de lumière, il est alors possible de calculer et d’affecter la couleur du pixel désiré. Nous avons, dans ce cas, fait une hypothèse simpliste : nous avons supposé que l’objet rencontré est diffus, c’est à dire qu’il renvoie la lumière de manière uniforme. Si ce n’est pas le cas, comme pour un miroir par exemple, le rayon est alors susceptible de poursuivre son chemin. Un

second rayon, réfracté ou réfléchi, est émis depuis le point d'intersection, et sa direction est déterminée en appliquant les lois de l'optique géométrique (Snell - Descartes). Le processus se poursuit jusqu'à ce que le rayon rencontre une surface diffuse ou le vide.

Cette algorithmes, très simple, fait donc l'hypothèse qu'une surface diffuse ne réémet pas de lumière (sauf pour l'oeil), ce qui est faux dans la pratique : un objet rouge, diffus, placé à coté d'une surface blanche provoquera une tache rougeâtre sur cette surface, faible certes mais parfaitement visible. Le lancer de rayons tel qu'il est présenté ci dessus oublie donc un certain nombre de chemins lumineux importants pour obtenir un aspect réaliste : les réflexions diffuses, mais aussi les caustiques (concentration d'énergie lumineuse en un point donné) en sont le parfait exemple. L'ensemble des phénomènes non pris en compte par ce lancer de rayons primaire sont regroupés sous la bannière de l'"éclairage global".

6.1.3 L'éclairage global

Pour qu'une image de synthèse semble un tant soit peu réaliste, il faut tenir compte de l'ensemble des interactions lumineuses dans la scène. Le terme "éclairage global" représente ces interactions. Calculer l'éclairage global, c'est donc parvenir à modéliser et calculer les échanges d'énergie lumineuse entre tous les éléments constitutifs de la scène. En particulier, les matériaux diffus agissent réellement sur l'éclairage de la scène, et ne se contentent pas d'absorber la lumière.

Plutôt que de chercher une solution spécifiquement adaptée au lancer de rayons, Kajjiya a donné un formalisme mathématique de ces considérations dans [31]. Le résultat de ces recherches est l'équation de rendu, référence actuelle du rendu réaliste, qui se présente sous la forme suivante :

$$L_r(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} \rho(x, \vec{\omega}_i \rightarrow \vec{\omega}_r) L_i(x, \vec{\omega}_i) \cos \theta_i d\omega_i \quad (6.1)$$

Quelle que soit la méthode de rendu utilisée, qu'il s'agisse du lancer de rayons, de la radiosité ou autre, calculer l'éclairage global correspond à résoudre cette équation en tout point de la scène. Elle peut être interprétée de la manière suivante : La luminance réfléchie L_r en un point x de la scène, dans une direction $\vec{\omega}_r$, est égale à la luminance propre L_e émise par ce point dans cette direction, à laquelle s'ajoute la somme des luminances L_i dans toutes les directions $\vec{\omega}_i$ du domaine $\vec{\Omega}$, réfléchies dans la direction $\vec{\omega}_r$ par le matériau dont la fonction de distribution de réflectance bidirectionnelle est ρ (merci à Xavier Serpaggi, auteur de cette traduction littérale de l'équation de rendu). Cette équation ne peut pas être résolue exactement, sauf pour des scènes très simples sans grand intérêt. La principale difficulté vient de sa récursivité : pour connaître les luminances incidentes L_i , situées dans l'intégrale, il faut résoudre l'équation de rendu en un point différent de x , repoussant ainsi le problème sans le solutionner. La majorité des algorithmes de rendu s'efforcent donc de calculer une solution approchée de cette équation.

Jacques Zaninetti a proposé, dans [86], une représentation plus lisible de l'équation de rendu, en séparant les différents types de chemins lumineux. Son utilisation en est alors grandement facilitée. Voici cette formulation :

$$L_r(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + L_{spec}(x, \vec{\omega}_r) + L_{dir}(x, \vec{\omega}_r) + L_{caust}(x, \vec{\omega}_r) + L_{ind}(x, \vec{\omega}_r) \quad (6.2)$$

Quelques explications s'imposent :

- $L_e(x, \vec{\omega}_r)$ est la luminance émise du point x dans la direction $\vec{\omega}_r$.
- L_{spec} est la luminance spéculaire réfléchie due à la luminance incidente.
- L_{dir} est la luminance réfléchie due à l'éclairage direct, c'est à dire provenant directement de la source d'une source de lumière.

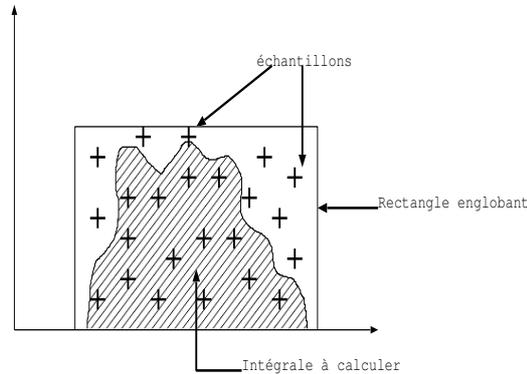


FIG. 6.2 – Illustration graphique de la méthode de Monte Carlo

- L_{caust} est la luminance réfléchie issue de rayons uniquement réfléchis ou transmis, c'est à dire n'ayant jamais rencontré de surface diffuse. Ces chemins sont souvent responsables de concentration d'énergie lumineuse, nommées caustiques.
- L_{ind} est la luminance réfléchie due à la luminance indirecte, englobant tous les cas non représentés ci dessus.

Pour résoudre l'équation de rendu, il peut être utile de calculer ces contributions séparément (voir [86] pour plus de détails).

6.1.4 Les méthodes de types Monte Carlo

Pour obtenir une solution à l'équation de rendu, il nous faut donc chercher des méthodes numériques adaptées. Parmi les plus populaires, on trouve les méthodes de type "Monte Carlo". Nous allons ici en rappeler brièvement le principe, d'un point de vue mathématique indépendant de la synthèse d'images, et présenter quelques solutions en lançant des rayons qui s'inspirent de ces méthodes.

Bien que leurs champs d'application soient très divers, les méthodes de Monte Carlo vont principalement nous servir pour effectuer des intégrations numériques, dans le but de résoudre l'équation de rendu. Notons toutefois que ces méthodes peuvent également être utilisées pour des équations aux dérivées partielles, des systèmes d'équations linéaires, ou encore la résolution de problèmes d'optimisation. Les méthodes de Monte Carlo s'appuient sur la répétition d'une expérience en vue d'évaluer une quantité. Dans le cas du calcul d'une intégrale, nous pourrions brièvement l'illustrer de la manière suivante : supposons que l'on veuille calculer l'intégrale représentée sur la figure 6.2. La fonction correspondante est ici bien trop complexe pour être intégrée par un calcul analytique. Supposons maintenant que cette intégrale puisse être bornée graphiquement par un rectangle. Si nous échantillons ce domaine rectangulaire, nous pouvons obtenir une approximation de la valeur de l'intégrale en admettant qu'elle devrait être proche de la surface du rectangle multipliée par la proportion d'échantillons situés à l'intérieur de la surface de l'intégrale. La précision augmente naturellement avec le nombre d'échantillons.

Nous allons maintenant donner une définition plus formelle, très inspirée de [15]. Supposons que l'on veuille évaluer l'intégrale suivante :

$$I = \int_0^1 g(x) dx \quad (6.3)$$

La méthode de Monte Carlo consiste à écrire l'intégrale 6.3 sous la forme

$$I = E[g(U)]$$

où U est une variable aléatoire suivant une loi uniforme sur $[0, 1]$. On utilise ensuite la loi des grands nombres :

$$\frac{1}{n} \sum_{i=1}^n g(U_i) \rightarrow E[g(U)]$$

En d'autres termes, si $u_1, u_2, u_3, \dots, u_n$ sont des valeurs de U , on aura

$$\frac{1}{n} (g(u_1) + g(u_2) + g(u_3) + \dots + g(u_n)) \rightarrow I$$

Il a été démontré que la vitesse de convergence est de l'ordre de $1/\sqrt{n}$, ce qui est relativement médiocre : pour obtenir une précision 4 fois meilleure, il faudra calculer 16 fois plus d'échantillons. Il existe de nombreuses méthodes pour accélérer la convergence, dont la stratification où l'échantillonnage d'importance. En deux mots, la stratification divise le domaine d'intégration en plusieurs zones plus petites, sur lesquelles l'erreur commise devra être plus faible, alors que l'échantillonnage d'importance consiste à augmenter la densité des échantillons dans les zones où le besoin s'en fait sentir (zones "importantes", d'où l'appellation). Nous ne détaillerons pas ces améliorations, mais le lecteur intéressé peut se reporter à [25].

Il existe de nombreux algorithmes de lancer de rayons plus ou moins basés sur les méthodes de Monte Carlo. Dans les sections suivantes, nous allons en présenter quelques unes, parmi les plus utilisées en rendu réaliste.

6.1.5 Le suivi de chemin, ou "Path tracing"

Dans [31], Kajiya a, en plus de l'équation de rendu, proposé une méthode pour en obtenir une solution approximative. Le principe est de ne plus se limiter aux rayons primaires, mais de poursuivre le lancer de rayons après chaque intersection, la valeur finale de l'éclairement étant la contribution totale du chemin suivi. La grande difficulté est évidemment de décider où et comment poursuivre le calcul. Il est aisé d'imaginer l'explosion combinatoire provoquée par cette méthode, pour peu que la scène comporte quelques objets transparents ou réfléchissants, et les temps de calculs qui en découlent directement. D'autre part, la diversité des chemins possibles fait que d'un pixel sur l'autre, les luminances peuvent varier très fortement¹. Les images obtenues peuvent donc être très bruitées, ce qui nuit grandement à l'aspect réaliste de l'image. Les améliorations de cette méthode sont nombreuses, nous pouvons citer (entre autres) :

- Le suivi de chemins bidirectionnel ([33]), qui lance des rayons depuis l'œil mais également depuis les sources. L'objectif est alors de reconnecter, via des rayons d'ombre, les deux chemins issus des sources de lumière et de l'observateur, pour obtenir une contribution globale plus rapidement qu'avec un seul chemin.
- Le "Next Event Estimator" ([53]), qui effectue un calcul d'éclairement direct en chaque point du chemin suivi de manière à cibler les contributions importantes.
- "Metropolis Light Transport", qui utilise non plus un suivi mais une mutation des chemins qui modifient leur contribution à l'image finale. De plus amples détails peuvent être trouvés dans [74].

¹Par exemple, un chemin peut avoir comme terminaison une source de lumière, donnant ainsi une luminance élevée, alors que le chemin issu d'un de ses voisins aboutira sur un matériau non réfléchissant, créant ainsi un fort contraste entre les deux pixels

En 1988, Ward a proposé dans [78] un suivi de chemins d'un genre un peu particulier pour calculer les inter réflexions diffuses : lors d'une intersection d'un rayon primaire avec un objet, les rayons secondaires ne sont relancés qu'en direction des objets les plus proches, de manière à obtenir une approximation locale de l'éclairage indirect. Cette approximation locale est stockée et pourra être réutilisée dans les calculs d'éclairage des autres points. Signalons enfin que les méthodes de Monte Carlo sont extrêmement sensibles à l'uniformité du tirage aléatoire utilisé. C'est pourquoi il existe des méthodes dites "quasi" Monte Carlo (comme par exemple [32]), dont la fonction n'est plus aléatoire mais quasi aléatoire, et dont la distribution est connue et maîtrisée, permettant ainsi une convergence plus rapide.

6.1.6 Les cartes de photons

Jensen a présenté dans [29] une technique permettant de résoudre le problème de l'éclairage global. Cette méthode est, tout comme le suivi de chemins bidirectionnel, une méthode en deux passes, mais la différence est dans les techniques d'estimation utilisée et les structures de stockages. Dans [27], Jean Claude Iehl présente de nombreuses extensions de cette méthode pour obtenir un lancer de rayons spectral doublé d'un contrôle perceptuel de la qualité du rendu.

Jensen introduit dans ses travaux la notion de photon : ce terme, à ne pas confondre avec son analogue de la physique des particules, représente une quantité élémentaire d'énergie émise par une source de lumière. Cette quantité est ponctuelle, mais peut présenter une distribution d'énergie associée. Le processus de rendu se décompose en deux phases :

Première phase : Propagation

Lors de la phase de propagation, des photons sont émis depuis la source de lumière dans toutes les directions et sont déposés sur les surfaces de la scène. Un pré-traitement permet de déterminer la visibilité des surfaces non diffuses depuis les sources de lumière. Les photons sont alors répartis en deux classes, selon qu'ils seront émis vers ces surfaces ou non (on parlera par la suite de photon normal ou "caustique"). Les photons caustiques sont émis en plus grande quantité, car des variations d'énergie importantes sont susceptibles de se produire là où ils seront déposés. Tous les photons sont stockés dans des "k-d trees", structure de données permettant une recherche efficace (atout nécessaire pour la deuxième phase). Ces structures sont locales pour les photons caustiques, globale pour les autres, et sont appelées "photon maps", ou cartes de photons.

Deuxième phase : Collecte

Cette deuxième phase ressemble à un suivi de chemins, à ceci près que la luminance en chaque pixel est évaluée grâce à une technique combinant estimation de densité ([67]) et Monte Carlo. L'équation de rendu est séparée selon les contributions à la luminance : la contribution des caustiques est obtenue grâce à une estimation de densité sur les photons caustiques autour du point d'intersection, alors que la contribution indirecte est calculée en relançant des rayons depuis le point d'intersection, puis en appliquant l'estimation de densité sur les photons non caustiques depuis ces nouveaux points d'intersection.

6.1.7 Les vecteurs lumineux

Jacques Zaninetti a, dans [86], présenté une méthode d'interpolation vectorielle de l'éclairage indirect. Cette technique, inspirée de [78], est celle qui est utilisée dans le logiciel YART au laboratoire

LISSE (présenté dans l'annexe A). Toutes les images de synthèse présentées dans ce mémoire ont été calculées en utilisant ce logiciel.

Le principe de la méthode des vecteurs lumineux est le suivant. Le processus de rendu se partage, comme pour la carte de photons, en deux phases : une phase de calculs des VEI (vecteurs d'éclairement indirect), et une phase d'interpolation.

Calcul des VEI

Dans cette phase, un certain nombre de points, choisis au hasard dans l'espace image, vont subir un calcul de l'éclairement indirect émis par les points correspondants dans la scène. Une méthode de Monte Carlo est utilisée pour obtenir ce résultat : un hémisphère, placé autour du point x (de la scène) considéré, est partitionné en cellules à angle solide constant. Dans chaque cellule, on relance un rayon, dont on calculera la contribution à l'éclairement indirect en x . Notons que ce procédé est récursif : pour calculer la contribution de ce rayon, il faudra sans doute calculer l'éclairement en un autre point de la scène avec la même méthode. Cette récursion est, dans la méthode de Zaninetti, stoppée par un terme ambiant constant au bout d'un nombre limité de réflexions.

En utilisant les contributions partielles de chaque cellule et la FDRB du matériau au point x , il est ensuite possible de déduire l'éclairement P et la direction d d'une source de lumière virtuelle destinée à remplacer la contribution indirecte. Notons que la carte de VEI obtenue au final, appelée germe, est réutilisable pour un calcul ultérieur.

Interpolation

Le calcul de l'image proprement dite se fait en lançant un rayon primaire, puis en recherchant des VEI dans le voisinage de l'intersection de ce rayon avec la scène. Les VEI éventuellement trouvés seront gardés ou éliminés du calcul en fonction de plusieurs critères, dont :

- La différence d'altitude entre le point d'intersection et le VEI.
- La direction d'observation pour laquelle le VEI a été calculé.
- La différence relative d'orientation des surfaces sur lesquelles sont placés respectivement le point d'intersection et le VEI.

Une décision est ensuite prise après ce tri. Si il reste suffisamment de VEI dans le voisinage du point d'intersection, alors l'éclairement indirect en ce point sera calculé par interpolation vectorielle de ces VEI voisins pondérée par un noyau gaussien. Si ce n'est pas le cas, un nouveau VEI est calculé au point d'intersection, il sera stocké et pourra être réutilisé par la suite.

6.1.8 Bilan Réflexions d'ordre général sur les algorithmes de calcul de l'éclairement global

Nous allons ici émettre un certain nombre de réflexions sur les méthodes présentées jusqu'ici pour résoudre l'équation de rendu. La méthode de lancer de rayons passe la plupart de son temps à calculer des intersections entre un rayon et la scène. Il est donc possible d'avoir une bonne idée du coût d'une méthode à base de lancer de rayons en estimant le nombre d'intersections qu'elle va devoir calculer.

De ce point de vue, il est assez évident de constater que la méthode de suivi de chemins classique risque d'être réellement très coûteuse : en considérant une image 512×512 , et en relançant 64 rayons par pixel avec une profondeur de récursion arbitrairement fixée à 5, le nombre d'intersections à calculer s'élève à $2,8 \cdot 10^{14}$. Nous pouvons bien sûr diminuer le nombre de rayons relancés ou la profondeur de récursion, mais dans un cas comme dans l'autre cela diminue aussi la qualité du résultat : un nombre trop faible de rayons relancés se traduit par l'apparition d'un bruit sur l'image, alors

que diminuer la profondeur de récursion fausse le calcul de l'éclairage global en le remplaçant par un terme ambiant. Les images calculées en suivi de chemins font souvent office de référence, car l'exhaustivité des calculs qu'elles nécessitent garantissent que la solution finale obtenue sera une simulation fidèle de la scène synthétisée. Mais si la scène est complexe, les temps de calculs peuvent facilement dépasser plusieurs dizaines d'heures.

La méthode de la carte de photons est beaucoup plus efficace : en se concentrant sur les zones "problématiques", c'est à dire considérées comme étant susceptibles de contenir des variations brusques et importantes de l'énergie lumineuse, elle limite très fortement le nombre de calculs inutiles. Elle permet de simuler de manière élégante les caustiques et (un peu moins élégamment) l'éclairage indirect. Toutefois, elle ne peut se passer d'une phase de pré-calcul destinée à choisir les directions privilégiées. Si cette phase se déroule mal, alors l'algorithme est mal initialisé et le calcul beaucoup moins efficace.

Les vecteurs lumineux permettent également une accélération importante des calculs d'éclairage global. Cette méthode est particulièrement efficace sur le calcul du terme indirect de l'équation de rendu. Toutefois, elle est soumise au même problème que la carte de photons : l'efficacité de l'algorithme est très sensible au germe. Or, ce germe est généré totalement aléatoirement alors qu'il pourrait se concentrer sur des zones où l'éclairage global est susceptible de créer des problèmes. Xavier Serpaggi a, dans [63] et [64], proposé un certain nombre d'améliorations à la méthode des vecteurs lumineux, dont une porte précisément sur l'optimisation du germe. D'autre part, des taches, dues à une interpolation trop importante ou malvenue, peuvent apparaître si les paramètres concernant la classification des VEI sont mal choisis. Ils sont, en effet, plus ou moins dépendants de la complexité de la scène. Nous pouvons ajouter un troisième inconvénient : malgré le fait que la carte de photon et les vecteurs lumineux soient plus rapides que le suivi de chemins, les temps de calcul restent relativement longs, et ce pour une raison très simple : quoi qu'il se passe, un calcul d'éclairage, plus ou moins complexe, est systématiquement effectué pour tous les pixels.

Nous avons donc trois problèmes à résoudre. Pour ce qui est du dernier, nous pouvons très bien imaginer une méthode qui n'effectuerait ce calcul que pour certains pixels, et pratiquerait une interpolation pour les autres, quitte à poursuivre le calcul (éventuellement jusqu'au calcul complet de l'image) si le résultat n'est pas satisfaisant. Ces méthodes sont dites "progressives", car elles permettent d'obtenir *progressivement* une image de qualité croissante, mais qui dans l'idéal devrait très vite ressembler au résultat final. Nous allons présenter quelques unes de ces techniques progressives dans la section 6.2.

Reste donc deux problèmes. Premièrement, la rapidité d'obtention du résultat des solutions présentées dépend de l'initialisation et des paramètres. Deuxièmement, des artefacts graphiques sont susceptibles d'apparaître sur l'image finale, nuisant à l'aspect photoréaliste désiré. Pour essayer de résoudre ces problèmes, nous allons tenter de prendre en compte les propriétés perceptuelles du système visuel humain. Il existe un certain nombre d'algorithmes de rendu perceptuels, nous en présenterons quelques uns dans la section 6.3

6.2 Les algorithmes de rendu progressif

6.2.1 Principes

Comme cela a été vu dans la section précédente, obtenir une image de synthèse réaliste avec prise en compte de l'éclairage global est un processus extrêmement long et lourd à mettre en place. Pour pouvoir réduire ces coûts, nous pouvons essayer d'agir sur deux facteurs : la mémoire et le temps CPU utilisés. Nous avons choisi de réduire le deuxième, pour plusieurs raisons. Premièrement, la mémoire

est de moins en moins chère : un gigaoctet de mémoire vive est aujourd'hui relativement abordable. Ensuite, il y a à l'heure actuelle une forte demande en images réalistes en temps interactif, voire en temps réel. Or, le problème de l'éclairage global ne sera pas résolu en temps réel si les algorithmes restent en l'état actuel. Il faut donc mettre au point une technique d'approximation s'approchant du résultat final le plus vite possible.

Nous appellerons ici "algorithme de rendu progressif" une technique qui n'échantillonne qu'une partie du plan image et qui interpole les valeurs des pixels restant, donnant ainsi très rapidement une image approximative mais ayant toutes les caractéristiques de l'image finale. Le schéma de ces méthodes est classique et très similaire à celui de notre distance adaptative : le plan image est subdivisé en cellules ; pour chacune de ces cellules, un critère de subdivision est calculé. L'algorithme de rendu parcourt ensuite l'ensemble des cellules. Si le critère ne satisfait pas une certaine condition, la cellule sera subdivisée. L'algorithme s'arrête lors qu'il ne reste plus de cellules à subdiviser.

Au fur et à mesure que le calcul avance dans le temps, l'image devient de plus en plus proche de son état final, mais l'utilisateur est libre d'arrêter le calcul quand il le désire. Dans cette section, nous allons présenter quelques techniques de rendu progressif, avec leurs avantages et leurs inconvénients. Nous nous limiterons aux méthodes qui nous ont inspiré pour notre travail de thèse.

6.2.2 La méthode de Painter et Sloan

Le but de la méthode de Painter et Sloan est d'obtenir efficacement des images avec anti crénelage. Rappelons que ce désagréable phénomène, bien connu en traitement du signal, est dû à la trop faible résolution des écrans et à un échantillonnage trop faible de la fonction d'éclairage. Il se manifeste par exemple sous la forme d'un effet d'escalier sur les bords des objets. Pour palier cet inconvénient, les pixels sont généralement sur-échantillonnés en lançant plusieurs rayons et en perturbant légèrement la direction de visée ("jittering") ; or, il se trouve qu'en 1989, obtenir une image calculée en anti-crénelage n'était pas un processus rapide. En fait, le calcul était effectué brutalement pour tous les pixels de l'image, et ce quelque soient leurs contextes respectifs. Les temps de calcul résultants étaient évidemment très longs.

Painter et Sloan ont donc présenté dans [52] une solution à ce problème qui sélectionne, au fur et à mesure du calcul, les pixels "méritant" un calcul d'anti-crénelage complet. Le plan image est subdivisé selon un "kd tree", les feuilles contiennent la valeur et la position des échantillons, alors que les noeuds internes stockent la valeur moyenne couverte, la variance interne, et le nombre d'échantillons de ce noeud. Pour obtenir de nouveaux échantillons, un parcours de l'arbre est effectué en partant de la racine. Une décision est prise à chaque noeud, pour décider quelle partie de l'image devra être raffinée. Les critères de décision dépendent du niveau auquel on se trouve dans l'arbre, mais ils utilisent tous la variance interne et la taille du noeud :

- Pour les niveaux "au dessus" du pixel, le critère est principalement la taille du noeud.
- Pour les niveaux de l'ordre du pixel, le critère est un produit entre la variance et la taille.
- Lorsque les niveaux inférieurs au pixel sont atteints, le but est alors d'augmenter la précision pour la valeur du pixel associé. Seule la variance est prise en compte.

La condition d'arrêt est basée sur un intervalle de confiance, calculé à partir de la variance et du nombre d'échantillons de chaque noeud. L'image est reconstruite à partir des échantillons en utilisant une interpolation "constante par morceaux", puis en filtrant le résultat.

Cet algorithme a le mérite d'être l'un des premiers à présenter un rendu progressif en lancer de rayons. La méthode est rigoureuse, et les résultats sont encourageants par rapport à un calcul complet. Toutefois, il subsiste plusieurs problèmes. Premièrement, l'interpolation utilisée pourrait être plus évoluée. Ensuite, les critères sont purement statistiques et ne cherchent qu'à évaluer le plus ra-

pidement possible l'équation de rendu, pratiquement sans tenir compte de paramètres géométriques ou d'éventuelles cohérences dans l'image. Enfin, la condition d'arrêt étant assez exigeante, le calcul reste relativement long pour obtenir une image de qualité. Dans la suite de cette section consacrée au rendu progressif, nous allons examiner d'autres méthodes plus récentes, faisant appel à des critères plus "visuels".

6.2.3 Les méthodes à triangulation

Nous allons, dans cette section, présenter deux méthodes faisant appel à une subdivision constituée d'une triangulation de Delaunay. La première est due à Pighin *et al* ([56]). La deuxième est l'oeuvre de Simmons et Séquin ([68]).

La méthode de Pighin *et al*

Le point de départ de la méthode de Pighin *et al* est de reprendre ce qui faisait défaut à la méthode de Painter et Sloan, à savoir une interpolation plus efficace lors de la reconstruction de l'image à partir des échantillons, tout en utilisant au maximum les capacités des cartes 3D. Globalement, les auteurs vont donc utiliser une triangulation de Delaunay en lieu et place du kd tree de Painter et Sloan, avec en vue la possibilité d'afficher la structure résultante très rapidement : en effet, la très grande majorité des cartes graphiques actuelles sont capables d'afficher rapidement des triangles de Gouraud en très grand nombre (de l'ordre d'une centaine de millions en une seule passe). Pour traiter les textures, les auteurs vont également mettre à contribution les capacités du matériel graphique, en stockant une "detail map" contenant les textures sans éclairage, qui sera additionnée au rendu lors de l'affichage. La triangulation sera ensuite raffinée à chaque itération en fonction de critères plus simples que ceux de Painter et Sloan. Nous allons maintenant détailler l'algorithme de Pighin *et al*. Sur la figure 6.3, nous pouvons visualiser les données utilisées par cette méthode, ainsi que le résultat obtenu au bout d'une itération.

Étape 0 : précalculs Il est nécessaire d'effectuer un certain nombre de calculs préliminaires avant de commencer la phase de rendu. Pour contraindre la triangulation de Delaunay, il nous faut la position de tous les bords visibles, qu'ils soient issus de bords d'objet, d'ombre ou encore de réflexions dans un miroir plan. Ensuite, il faut constituer une carte, nommée "detail map" par les auteurs, qui contiendra les textures projetées sur les objets, mais sans calcul d'éclairage. Pour les textures "bit-map", cette opération peut être effectuée très rapidement en utilisant l'accélération matérielle de la carte graphique.

Étape 1 : création de l'ensemble d'échantillons initial Toute méthode progressive commence par le choix des échantillons de départ. Les auteurs ont ici choisi une grille régulière d'échantillons, dont ils obtiennent ensuite une triangulation de Delaunay. Toutefois, pour limiter les erreurs d'interpolation, la triangulation est contrainte sur les bords des objets.

Étape 2 : interpolation L'image est obtenue en affichant les triangles ainsi créés et en superposant la "detail map" afin d'ajouter les textures. Toutes ces opérations utilisent l'accélération matérielle de la carte graphique.



FIG. 6.3 – Données utilisées par la méthode de prévisualisation de Pighin (de haut en bas, et de gauche a droite) : position des bords géométriques, maillage de Delaunay contraint associé, carte des textures sans calcul d'éclairage, image résultat.

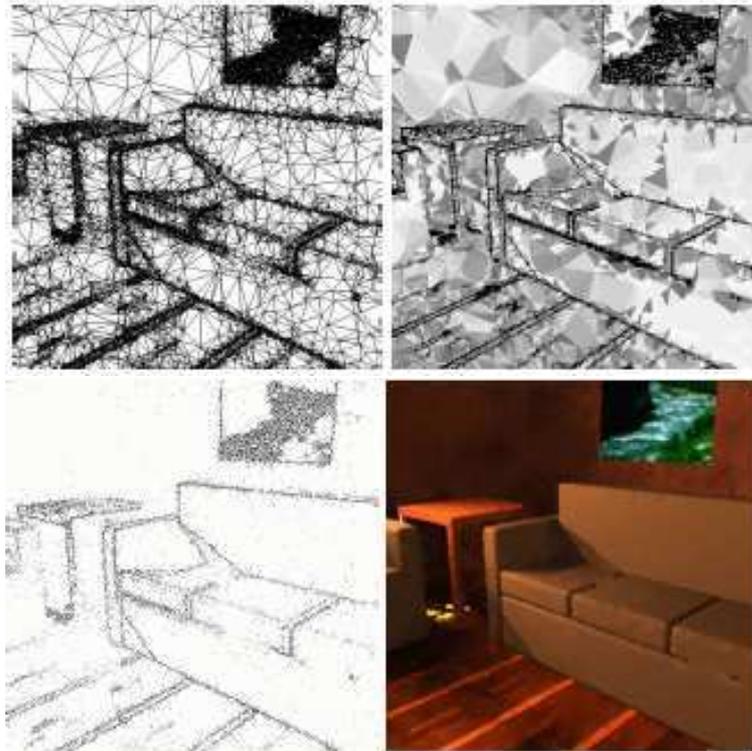


FIG. 6.4 – Données utilisées par la méthode de Simmons (de haut en bas, et de gauche à droite) : maillage de Delaunay, carte de priorité, échantillons calculés, image reconstruite.

Étape 3 : raffinage Une fois cette triangulation mise en place, un poids est affecté à chaque arc. Il est égal à la somme de la taille de l’arc avec la différence de luminance entre ses deux extrémités. La structure est parcourue, et un nouvel échantillon est ajouté au milieu de l’arc ayant le poids le plus fort. Le processus recommence ensuite à partir de l’étape 2, jusqu’à ce que l’utilisateur arrête le processus ou que tous les pixels du plan image soient échantillonnés.

Méthode de Simmons et Séquin

Dans [68], Simmons et Séquin ont proposé une méthode similaire, également basée sur une triangulation de Delaunay. Nous n’en présenterons ici que les caractéristiques originales par rapport à [56]. Pour le détail complet de la méthode, voir [68]. La figure 6.4 représente les données calculées et utilisées par cette technique.

Contrairement à Pighin *et al*, qui triangulaient directement le plan image, Simmons et Séquin utilisent une triangulation “2D et demi”, selon leurs propres termes. Cette dernière s’applique en effet sur une sphère unité centrée sur le point de vue. Cette projection permet d’éviter de tout recalculer dans le cas où le point de vue bougerait légèrement. L’entité élémentaire de la structure de subdivision n’est plus l’arc mais le triangle. Le poids de chaque triangle est calculé en appliquant une somme pondérée des différentes valeurs R,G,B de chacun de ses sommets et de la différence de profondeur des pixels constituant la cellule. La triangulation complète, une fois projetée sur le plan image, constitue une carte de priorité. Chaque nouvel échantillon est issu d’une suite pseudo-aléatoire 2D altérée suivant la carte de priorité. Il est ajouté à l’intérieur d’un triangle et en crée ainsi trois nouveaux. La condition de Delaunay est vérifiée sur ces nouveaux éléments, et la structure est corrigée si besoin est.

Bilan

Ces deux méthodes ont en commun l'utilisation intensive de l'accélération matérielle dont bénéficient les cartes vidéo modernes. Elles sont donc particulièrement bien adaptées à la pré visualisation de scènes. La méthode de Simmons et Séquin propose même la prise en compte du changement de point de vue, avec une réactualisation en temps réel de la structure de visualisation. Les limites sont toutefois assez rapidement visibles. Premièrement, les scènes géométriquement complexes seront difficiles à afficher : les scènes naturelles, par exemple, présentent trop peu de linéarités dans leur géométrie pour pouvoir être approximées rapidement par des triangles. Ensuite, comme précisé plus haut, la scène se doit d'être modélisée avec des facettes, alors qu'il existe d'autres modèles (CSG, splines, blobs, ...) permettant parfois une modélisation plus précise ou plus élégante². Enfin, la cohérence spatiale de l'image n'est exploitée que sur les bords visibles, alors qu'une image de synthèse classique comporte beaucoup d'autres zones susceptibles d'être interpolées facilement, à commencer par toutes celles où la luminance des pixels varie peu (ombres douces, éclairage indirect, ...). Dans la section suivante, nous allons présenter un algorithme prenant en compte ces faibles variations.

6.2.4 La carte de cohérence directionnelle de Guo

En 1998, Guo proposa dans [24] de réduire les temps de calculs d'une image en lançant des rayons en tirant parti de la cohérence directionnelle de l'image. En effet, une image présente la plupart du temps des zones où la variation de l'éclairage est uniforme, donc facilement interpolable. Par contre, cette zone n'est pas nécessairement purement horizontale ou verticale. Il est donc inutile d'essayer de l'approximer en "rasterisant" cette partie du plan image. Guo propose alors de calculer une direction de moindre discordance ("least discrepency direction"), direction dans laquelle la luminance des pixels est censée ne pas être sujette à d'importantes variations. Nous allons maintenant détailler cette méthode.

Guo partage le plan image suivant un arbre quaternaire, mais contrairement aux méthodes habituelles, l'arbre est, dès son initialisation, subdivisé de telle manière que l'image est recouverte de blocs 8×8 . Les luminances des quatre coins de chaque bloc sont calculés, puis interpolés linéairement, donnant ainsi une première image. Cette image va ensuite être raffinée par classification des blocs selon une estimation de leur complexité :

- Si le bloc est à peu près uniforme, il sera classé comme étant "smooth".
- Si le bloc présente une discontinuité, il sera classé comme étant "edge".

Le problème est maintenant d'évaluer la complexité de ces blocs en ayant uniquement la valeur de la luminance aux quatre coins. Guo utilise ici un critère basique : si le contraste de Michelson, donné par $\frac{\max - \min}{\max + \min}$, de ces quatre coins dépasse un certain seuil fixé, le bloc est considéré comme complexe. Étant donné que nous traitons une scène synthétique, la position et le nombre des objets sont connus. Guo considère aussi que les blocs traversés par les frontières de deux objets ou plus ne peuvent pas être classés "smooth".

Il faut maintenant déterminer la direction de moindre discordance pour tous les blocs "edge". Premièrement, tous les pixels de son contour sont calculés, en plus des quatre coins. Guo effectue ensuite une recherche de discontinuité sur ce contour. Les blocs "edge" sont alors reclassés en deux sous catégories : "simple" ou "complex", selon le type de discontinuités rencontrées sur le contour. La figure 6.5 illustre les différents types de blocs avec les types de discontinuités possibles. Pour ce faire, trois tests sont mis en place : un test d'ordre zéro, un test d'ordre un, et un test "objet".

²Notons toutefois que la plupart des scènes modélisées avec ces autres modèles peuvent se convertir en facettes relativement facilement.

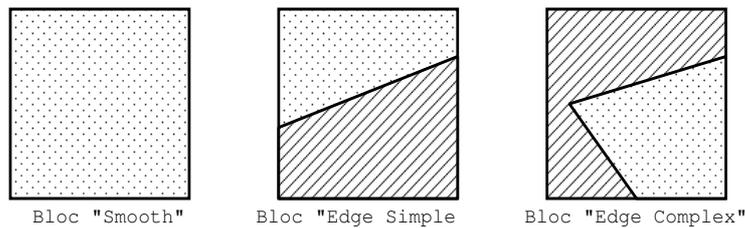


FIG. 6.5 – Classification des blocs pour la méthode de Guo

- Test d’ordre zéro : peut on trouver sur le contour deux, et seulement deux, pixels qui partagent le contour en deux domaines de luminance à peu près uniformes ? Si ce n’est pas le cas, le test échoue.
- Test d’ordre un : la technique consiste ici à évaluer la direction de la discontinuité au voisinage des deux points de rupture. Si la discontinuité est rectiligne, alors l’écart entre les deux directions de discontinuité ne devrait pas dépasser un certain seuil. Si ce n’est pas le cas, ce test échoue.
- Test objet : Si les pixels du contour du bloc appartiennent à au moins deux objets différents, alors le test échoue.

Un bloc passant avec succès les trois tests est classé “edge simple”. Si ce n’est pas le cas, il est classé “edge complex”. Pour chaque bloc “edge”, la direction de moindre discordance est calculée en calculant sur le contour C , la quantité suivante :

$$d(n) = \frac{1}{s} \int_C (f(x+t(x)n) - f(x))^2 ds \quad (6.4)$$

La direction t qui minimise cette intégrale est la direction de moindre discordance. Afin de simplifier la détermination de cette direction, Guo ne considère que 8 directions réparties entre 0 et π . La structure obtenue, l’arbre quaternaire avec les blocs classés en trois catégories et leur direction de moindre discordance associée, est appelée la carte de cohérence directionnelle (“Directional Coherence Map”, ou “DCM”).

L’interpolation s’effectue ensuite en fonction du type de chaque bloc :

- Les blocs “smooth” sont interpolés linéairement, de manière classique, suivant la luminance de leurs coins.
- Les blocs “edge” sont également interpolés de manière linéaire, mais sur tout leur contour suivant la direction de moindre discordance.

La méthode de la carte de cohérence directionnelle donne d’excellents résultats. Guo prétend que sur certaines scènes, six pour cent des pixels réellement calculés suffisent pour obtenir une image indiscernable de l’image obtenue avec le calcul complet de tous les pixels. Nous avons essayé cet algorithme, et la figure 6.6 donne un exemple de ce que nous obtenons avec notre propre implémentation. La partie gauche de la figure montre l’image obtenue avec un calcul complet utilisant les vecteurs lumineux. La partie droite de la figure montre le résultat obtenu avec la méthode de Guo, en ne calculant que 30 % des pixels. Les images sont effectivement très semblables, mais nous remarquons toutefois un effet de “blocs” assez prononcé sur le fond de l’alcôve, au centre de l’image. Les paramètres utilisés dans notre test sont les mêmes que dans [24]. Les temps de calcul sont d’environ 90 secondes pour l’image complète, et 54 secondes pour l’image calculée avec la DCM. Les calculs sont, une fois encore, effectués sur un Pentium 4 2Ghz, équipé de 1Go de mémoire vive.

Signalons enfin que la méthode de Guo possède deux autres défauts. Premièrement, les textures ne sont pas gérées, ce qui peut s’avérer gênant : imaginons que notre alcôve soit creusée dans un marbre

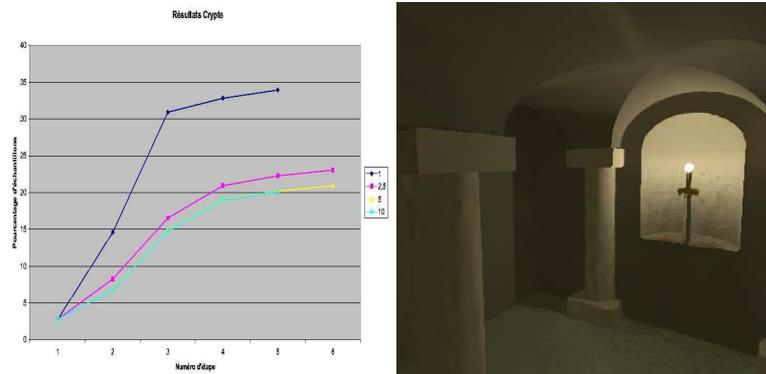


FIG. 6.6 – Résultats de la méthode de Guo. A gauche : calcul complet. A droite : DCM

veiné, avec une texture très fine. Dans ce cas, il sera impossible d’approximer les blocs de cette région malgré le fait que l’éclairage dans cette zone soit relativement simple. Ensuite, les objets très fins risquent d’apparaître très tard dans le rendu progressif, car tant que les blocs sont de grande taille, ils sont susceptibles d’être manqués. Scheel *et al* ont proposé dans [61] des améliorations de la DCM en vue d’éliminer ces défauts. Pour les petits objets, un calcul exact des pixels intérieurs au bloc est effectué si jamais ce dernier contient (ou est traversé par) un objet très fin ou très petit. Ce test est réalisé avant l’interpolation en testant si le pixel qui s’appête à être interpolé appartient au même objet que les pixels du contour du bloc. Les textures seront simplement stockées dans une “detail map”, similaire à celle de Pighin *et al*. Le calcul de l’éclairage est ensuite effectué indépendamment des textures. Notons que cette dernière technique est plus facile à appliquer si la scène ne contient que des surfaces diffuses.

6.3 Les algorithmes de rendu perceptuel

6.3.1 Principes

Le rendu perceptuel se propose d’incorporer, dans un algorithme de calcul de l’éclairage global, la prise en compte de la perception de la scène simulée par le système visuel humain, et ce dans un but bien précis : économiser du temps de calcul en délaissant les caractéristiques non visibles, au profit des zones de la scène où la perception des détails est plus élevée. Ainsi, le calcul de l’image se fait “au plus juste”. Malgré le fait que les travaux dans ce domaine en synthèse d’image sont relativement récents, ils sont nombreux. Nous présenterons ici ceux que nous avons jugés comme étant les plus importants, notamment ceux qui nous ont inspiré pour ce travail de thèse.

6.3.2 Les travaux de Myszkowski *et al*

Myszkowski a proposé de très nombreux travaux sur le rendu perceptuel. Les premiers d’entre eux, concernant la validation du VDP de Daly, ont déjà été évoqués dans le chapitre précédent. Nous allons ici présenter la partie dédiée au rendu piloté par le VDP, ainsi que les recherches ultérieures menées par le même auteur dans ce domaine.

Dans l’article [49], l’auteur va, après avoir validé le VDP de Daly pour une utilisation en synthèse d’images, évaluer les applications possibles d’un tel modèle au sein d’un algorithme de rendu. Trois applications sont testées :

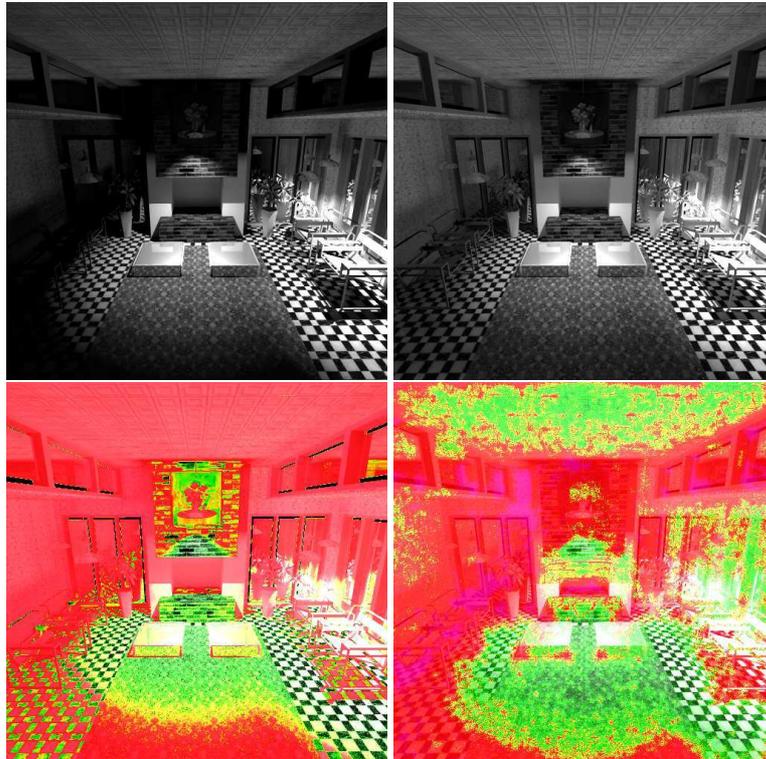


FIG. 6.7 – Résultats du test “comparaison avec l’image finale” (de haut en bas, de gauche à droite) : image issue de la première itération, image finale issue du rendu “suivi de chemins”, différence pixel à pixel, sortie du VDP. (issu de [49])

- Utilisation du VDP pour comparer une image intermédiaire avec l’image finale.
- Utilisation du VDP comme condition d’arrêt.
- Utilisation du VDP comme critère de subdivision et de simplification du maillage.

Dans les deux premiers cas, deux algorithmes de rendu ont été utilisés : un lancer de rayons à base de suivi de chemins et un algorithme de radiosité hiérarchique. Pour le troisième, il semble que seul l’algorithme de radiosité ait été utilisé.

Utilisation du VDP pour comparer une image intermédiaire avec l’image finale

Pour ce test, Myszkowski a choisi de comparer les itérations successives du rendu avec le résultat final, obtenu quand l’algorithme s’arrête. La figure 6.7 montre l’image obtenue après la première itération, l’image finale, la carte des différences pixel à pixel, et le résultat du VDP, le tout pour l’algorithme de suivi de chemin³. Il apparaît ici clairement que le VDP atténue les différences dans les zones où la fréquence spatiale est élevée, comme sur le plafond ou le sol par exemple.

Utilisation du VDP comme condition d’arrêt

Idéalement, une condition d’arrêt devrait arrêter le rendu quand l’image obtenue devient indiscernable d’une hypothétique image “complète”, sur laquelle l’équation de rendu aurait été entièrement

³Les résultats obtenus semblent toutefois peu dépendants de la méthode de rendu choisie.

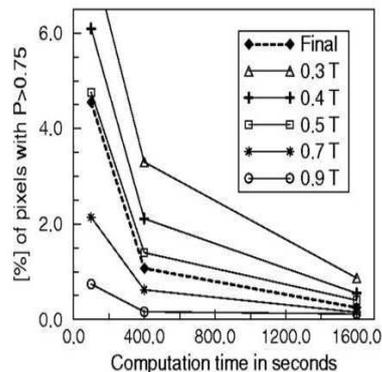


FIG. 6.8 – Résultats du test “VDP utilisé comme condition d’arrêt”, *issu de [49]*

et exactement résolue. Étant donné qu’il est évidemment impossible d’obtenir une telle image, la question est de savoir quelles images comparer pour utiliser le VDP comme condition d’arrêt. Myszkowski suggère l’hypothèse suivante : si deux images successives du rendu sont indiscernables, c’est à dire qu’il n’y a aucune différence visible entre ces deux images, alors le rendu peut s’arrêter⁴. Cette hypothèse semble censée : en effet, si deux images successives produites par l’algorithme de rendu sont très peu différentes, on peut légitimement supposer qu’il en sera de même pour les itérations suivantes. Il n’est donc pas nécessaire de poursuivre le rendu, puisque tous les détails qui seront ajoutés par la suite seront, en principe, indiscernables.

Myszkowski compare donc deux images issues de l’algorithme de rendu à deux instants différents, et examine la diminution de la détection de probabilité entre ces deux images. Le problème est maintenant de déterminer l’intervalle de temps idéal. L’auteur a essayé de déterminer ce temps expérimentalement : considérant l’image I_T obtenue au bout du temps T , il a examiné les différences entre I_T et $I_{\alpha T}$ (α étant un réel compris entre 0 et 1) ainsi que les différences entre I_T et I_C , cette dernière étant l’image dite “complète”. Les résultats sont reportés sur la figure 6.8. Sur cette figure, $\alpha = 0,5$ semble être la valeur qui donne la comparaison la plus proche de celle que l’on aurait avec l’image complète. Toutefois, l’auteur souligne qu’en pratique, ce facteur α a peu d’importance : en effet, d’une valeur de α à l’autre, les zones ayant les plus fortes différences sont situées aux mêmes endroits, ce qui signifie que ce seront les mêmes zones qui seront raffinées dans l’hypothèse où le processus de rendu serait guidé par le VDP.

Utilisation du VDP comme critère de subdivision et de simplification du maillage en radiosit 

En radiosit , le maillage de la sc ne est le support d’une approximation locale de la fonction d’ clairage. Si ce maillage est trop grossier, des artefacts tr s visibles apparaissent sur l’image ; si il est trop fin, les temps de calcul sont inutilement  lev s. Myszkowski a tent  d’utiliser le VDP pour valider une m thode de simplification de maillage en radiosit ⁵. Cette m thode comporte 3 passes :

- Choix des  l ments   subdiviser par une m thode plus ou moins empirique⁶.
- Validation par le VDP de ces  l ments.

⁴Nous reprendrons cette hypoth se dans notre propre m thode.

⁵Une m thode de simplification subdivise ou r unit les  l ments d’un maillage en fonction des exigences du processus de rendu. D’ordinaire, les crit res sont simples (diff rence de luminance, par exemple)

⁶Cette m thode comprend en une reproduction de ton pour obtenir la “couleur” de l’ l ment en coordonn es RGB affichables, suivie d’un seuillage de ces coordonn es.

- Réunification des éléments inutilement subdivisés.

La deuxième phase fonctionne de manière similaire au test précédent : pour vérifier la validité des choix effectués lors de la première phase, l'image résultant du nouveau maillage est calculée et comparée avec l'itération précédente. Plus de détails peuvent être trouvés dans [49].

Cette étude a clairement montré qu'un modèle de vision peut être d'une grande utilité en synthèse d'images. En le faisant intervenir à plusieurs niveaux dans le processus de rendu, Myszkowski a mis en évidence les bénéfices potentiellement exploitables en termes de temps de calcul (pour les deux premiers tests) et de mémoire utilisée (pour le troisième). Mais il ne s'est pas arrêté à cette estimation des possibilités du VDP et a, par la suite, mis en oeuvre des algorithmes de rendu basés sur un modèle de vision adapté. Nous allons ici présenter brièvement quelques uns de ces travaux consacrés au rendu perceptuel d'animations.

Dans [50], Myszkowski *et al* combinent deux techniques de rendu pour alléger les temps de calcul de séquences de type "walkthrough"⁷ : ils utilisent simultanément un lancer de rayon de type "carte de photons", et un rendu basé sur images (IBR)⁸. Évidemment, ces deux méthodes n'ont pas le même prix, et l'idée ici est d'utiliser le lancer de rayons uniquement là où c'est nécessaire. Pour effectuer ce travail, Myszkowski *et al* ont mis au point une métrique spatio-temporelle perceptuelle. Le rôle de cette métrique sera de déterminer les zones où un lancer de rayons est souhaitable, en mesurant les différences de perception entre une partie de l'image calculée avec la méthode IBR et la même zone calculée en lancer de rayons. La méthode impose un certain nombre de limites : en plus du fait que les objets de la scène sont statiques, le chemin et les positions de la caméra doivent être connus à l'avance. Le modèle de vision utilisé est une version dérivée d'un modèle statique modifié pour prendre en compte le paramètre temporel.

Concrètement, le chemin complet suivi par la caméra est découpé uniformément en segments S , la première étape k_1 et la dernière étape k_{2n} de ce segment sont calculées entièrement en lancer de rayon. Deux versions de l'étape k_n sont générées par IBR, respectivement à partir de k_1 et k_{2n} . Ces deux images sont ensuite comparées avec la métrique spatio-temporelle. En fonction des résultats obtenus, les objets dont les pixels projetés ont une différence trop élevée sont recalculés sur k_n en utilisant le lancer de rayon, de même que les pixels "vides", n'ayant pu être reprojétés avec l'IBR. Si la différence entre les deux images est toujours trop importante après ces traitements, le segment S est subdivisé en deux parties et l'algorithme recommence récursivement. Si par contre la différence est faible, les deux images dérivées sont "mêlées" pour obtenir l'image correspondant au segment S , et toutes les images de ce segment sont générées par la même méthode. Toutefois, la métrique n'est effectivement calculée que pour k_n , les autres sont supposées suffisamment proche des extrémités du segment pour donner une approximation correcte.

Dans [48], les mêmes auteurs s'inspirent cette fois d'une méthode de calcul basée sur l'estimation de densité, qu'ils étendent à l'animation. Ils y intègrent, de manière similaire à la méthode exposée ci-dessus, une métrique spatio-temporelle qui guidera le choix du nombre de photons à utiliser pour chaque image de la séquence. Le schéma global de la méthode est dans l'ensemble très similaire à l'article [50] : la séquence est partagée en segments. La fonction d'illumination est échantillonnée sur chaque image du segment, puis l'estimation de densité permet de décider, en fonction des résultats, si ce segment nécessite d'être allongé ou raccourci. La métrique perceptuelle est ensuite utilisée pour adapter le nombre de photons pour chaque frame, de manière à ce que le bruit généré ne soit pas perceptible. Enfin, la fonction d'éclaircissement est reconstruite en utilisant ces données.

⁷"parcours" en français : seule la caméra bouge, les objets de la scène sont fixes

⁸le terme anglo-saxon est "Image Based Rendering"

La contribution de Myszkowski au rendu perceptuel est donc très importante. Toutefois, une rapide comparaison entre les techniques d'évaluation du VDP et les algorithmes de rendu met en évidence la principale difficulté rencontrée lorsque l'on veut faire du rendu perceptuel : si les bénéfices que l'on peut en retirer sont clairs, l'intégration d'un modèle de vision au sein d'un algorithme de rendu est loin d'être évidente et amène de nombreuses questions. Que comparer ? Quand ? Avec quoi ? En quels termes doit on évaluer la différence ? Myszkowski a, par exemple, systématiquement choisi des images successives, ce qui n'est pas nécessairement l'idéal. Nous allons maintenant présenter d'autres travaux dont l'approche du problème est différente, ouvrant ainsi de nouvelles pistes de réflexion.

6.3.3 Les travaux de Bolin et Meyer

Bolin et Meyer ont travaillé sur les aspects perceptuels en lancer de rayon. Ils ont présenté deux algorithmes que nous allons détailler dans cette section.

En 1995, l'idée principale des auteurs de [6] était de tirer parti de trois caractéristiques du système visuel : la sensibilité au contraste, la non linéarité de perception des différences de contraste, le masquage. Ces caractéristiques étant essentiellement fréquentielles, les auteurs ont choisi d'effectuer leur calculs directement dans le domaine fréquentiel plutôt que dans le domaine spatial. Ils ont donc conçu un modèle visuel intégrant trois composantes simulant ces caractéristiques. Ce modèle est constitué des étapes suivantes :

- Une transformation colorimétrique pour obtenir, à partir des données XYZ de départ, des coordonnées SML représentant les signaux reçus par les cônes.
- Une loi exponentielle pour simuler la perception visuelle des contrastes.
- Une transformation colorimétrique pour obtenir des coordonnées AC1C2.
- Un filtrage pour simuler les fonctions de sensibilité au contraste relatives au stimuli chromatiques et achromatiques.

L'aspect fréquentiel de la méthode est inspiré de l'algorithme de compression JPEG. L'image est partagée en une suite de blocs 8×8 . Chaque bloc est placé dans une liste d'importance, triée du plus au moins important. Le schéma d'échantillonnage est le même pour chaque bloc. Les termes fréquentiels manquants sont obtenus par la méthode des moindres carrés, ce qui requiert une inversion de matrice pour chaque terme calculé. Afin de simuler simplement le masquage, la précision de la méthode de Monte Carlo utilisée pour obtenir les échantillons décroît si la fréquence augmente, car le bruit est moins visible dans ces conditions. Pour choisir le bloc qui doit être traité, le modèle est appliqué sur les blocs, et une valeur de priorité est calculée en fonction d'une variance des résultats au sein de chaque bloc. Cette valeur est censée représenter la visibilité des artefacts pour chaque bloc, et conditionne sa place dans la liste de priorité. Enfin, les blocs sont filtrés pour limiter le crénelage.

Cette méthode, très originale, montre son efficacité sur les résultats de la figure 6.9. Sur la partie supérieure, nous voyons deux séries de deux carrés placés sur un fond différent. La différence de luminance entre le petit carré et le grand est identique dans les deux cas, mais la différence perçue n'est pas la même. La partie inférieure de la figure permet de vérifier le schéma d'échantillonnage de cette scène. Il est clairement visible que quand la différence de luminance perçue est faible, l'échantillonnage est également moins important. Toutefois, il est difficile de mettre en évidence les performances de cette méthode sur une image un peu plus évoluée. D'autre part, elle ne tient compte que des phénomènes liés à la vision des contrastes. Toutes les autres caractéristiques de la vision ne sont pas exploitées.

En 1998, les mêmes auteurs vont proposer dans [7] une autre méthode basée sur un modèle complet, et non plus uniquement fréquentiel. Ce modèle est dérivé de celui de Lubin, et nous avons présenté son fonctionnement et ses performances dans la section 5.2.3. Nous allons dans cette section

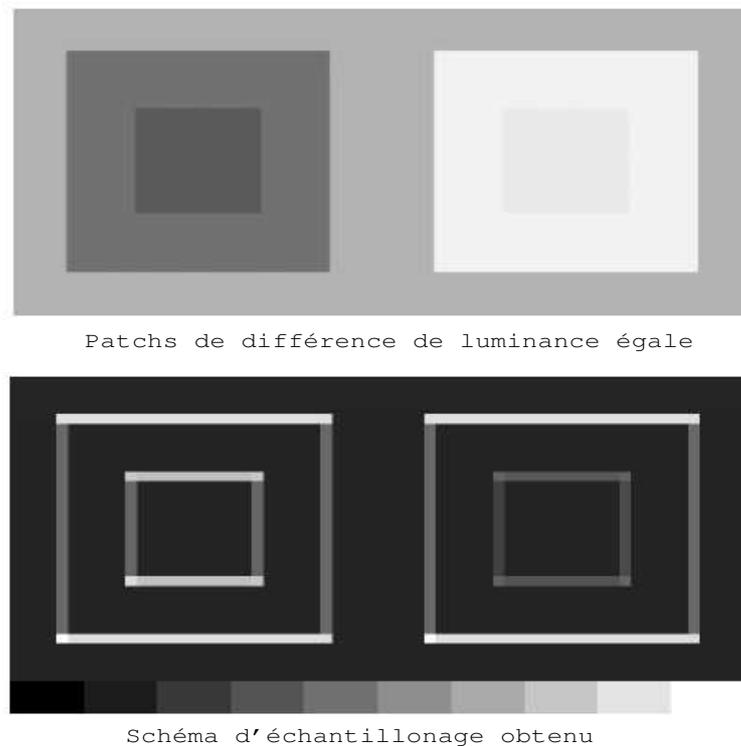


FIG. 6.9 – Résultats de la méthode fréquentielle de Bolin et Meyer (*issu de [6]*)

présenter l’algorithme de rendu adaptatif basé sur ce modèle. Il a pour objectif de guider le placement des échantillons dans la scène de manière à minimiser l’erreur perceptuelle.

Le schéma de fonctionnement de cette méthode est le suivant :

1. La scène est échantillonnée.
2. Une approximation de la représentation corticale de la scène est calculée à partir de ces échantillons et stockée dans un arbre quaternaire.
3. Une erreur locale sur cette approximation est calculée de manière similaire à Painter et Sloan dans [52].
4. Deux images sont construites statistiquement à partir de cette approximation.
5. Ces images sont comparées via le modèle de vision et les erreurs résultantes sont stockées dans l’arbre.
6. L’erreur est propagée de manière à affecter une erreur moyenne à chaque noeud de l’arbre.
7. Le prochain échantillon est choisi en suivant dans l’arbre le chemin d’erreur maximale.
8. Le noeud correspondant à cet échantillon est traité, et la structure mise à jour en conséquence.

Nous ne détaillerons pas toutes les étapes, mais l’approximation de la représentation corticale ainsi que le calcul d’erreur méritent quelques explications supplémentaires.

La représentation corticale

Comme nous l’avons vu dans la section 5.2.3, Bolin et Meyer ont remplacé la “cortex transform” utilisée par Lubin par une transformation à base d’ondelettes de Haar. Cette dernière est une décomposition fréquentielle de l’image sur trois niveaux. Il est donc tout à fait possible de la représenter par

un quad tree en stockant les valeurs moyenne des pixels dans les feuilles et les termes de détails dans les noeuds supérieurs. Lorsqu'un échantillon est calculé, sa valeur est ajoutée dans la feuille correspondant à la position de l'échantillon pour contribuer à la valeur moyenne. Les termes de détails et passe bas concernés sont ensuite mis à jour dans les noeuds supérieurs en remontant la nouvelle valeur moyenne. Si tous les fils d'un noeud contiennent au moins un échantillon, il est possible de calculer ces termes de manière classique. Si ce n'est pas le cas, les termes de détails sont indéterminés, et les termes passe bas sont calculées à partir des valeurs disponibles chez les fils.

Le mécanisme de calcul et de propagation d'erreur

Deux erreurs sont calculées dans cette méthode : une erreur statistique, en calculant en chaque noeud la variance des termes détails et passe bas, et une erreur perceptuelle calculée avec la métrique entre images. Mais pour utiliser une métrique entre images, il nous faut deux images. Bolin et Meyer ont choisi, contrairement à Myszkowski, de dériver deux images à partir des termes de détails disponible au moment de la comparaison. Ils effectuent un calcul statistique destiné à obtenir deux cartes telle que la première minimise l'énergie de contraste, alors que la deuxième la maximise. Afin d'obtenir un tel résultat, ils utilisent la variance des termes de détails : la première image sera constituée des termes dont l'écart à la moyenne est inférieur à 25%, la deuxième contiendra ceux dont l'écart à la moyenne est supérieur à 75%. Une erreur locale est ensuite calculée pour ce noeud avec la métrique, à laquelle viendra s'ajouter l'erreur maximale parmi les fils de ce noeud.

Résultats et réflexions

Les résultats de cet algorithme semblent convaincants. Les performances du modèle, mises en évidence dans la section 5.2.3, sont confirmées dans le rendu progressif, comme il est possible de le constater sur la figure 6.10. Nous pouvons voir, sur la gauche de la figure, que le rendu perceptuel utilise beaucoup moins d'échantillons que la méthode de type Monte Carlo employée à droite. Cependant, ces images ne font intervenir que l'éclairage direct, et ne présentent donc aucune difficulté particulière. Il nous est donc impossible de vérifier si la méthode est capable de gérer les phénomènes complexes comme l'éclairage global ou les caustiques.

6.3.4 La méthode de Ramasubramanian *et al*

Dans [57], Ramasubramanian *et al* ont proposé une méthode originale, à plusieurs niveaux, pour utiliser un modèle de vision dans une méthode de calcul de l'éclairage global. Le modèle est différentiel, c'est à dire qu'il fournit pour une image le seuil différentiel en luminance, pour chaque pixel. Par seuil différentiel d'un pixel, nous entendons la quantité de luminance qu'il est possible d'ajouter ou de retrancher à ce pixel sans que le système visuel ne perçoive de différence. La carte de seuils obtenue pour chaque pixel permet ensuite de comparer perceptuellement deux images en comparant directement leurs luminances.

L'établissement d'une telle métrique se fait à partir des mesures déjà présentées dans les sections 3.2.1 et 3.2.2. Le modèle de vision associé se décompose en deux parties, respectivement dédiées à la détection des luminances et des contrastes. La partie contraste joue un rôle de pondération par rapport à la partie luminance et est constituée de deux entités consacrées à la sensibilité au contraste et au masquage.

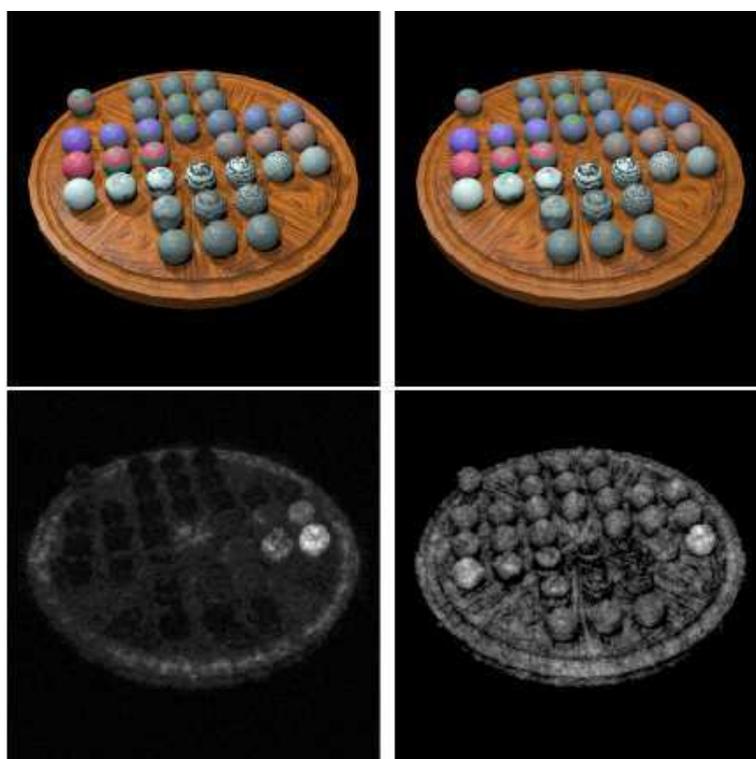


FIG. 6.10 – Résultats de la méthode perceptuelle de Bolin et Meyer (*issu de [7]*). A gauche : rendu perceptuel. A droite : rendu Monte Carlo classique. Ligne inférieure : Schéma d'échantillonnage.

La détection des luminances

Il s'agit dans ce cas de choisir une fonction donnant la luminance minimale détectable en fonction de la luminance du fond (voir la section 3.2.1). Les auteurs ont choisi la formulation utilisée par Ward dans [77] :

$$\delta L(L_a) = 0.0594.(1.219 + L_a^{0.4})^{2.5} \quad (6.5)$$

La partie contraste

Conformément à ce qui a été évoqué précédemment, les données concernant la détection des luminances sont obtenues dans des conditions très particulières : fond achromatique et motifs uniformes, ce qui est évidemment très loin des conditions de visualisation d'une image habituelle. La raison est fort simple : la détection des luminances est affectée par la vision des contrastes. Deux aspects sont présents dans ce cas : la sensibilité au contraste et le masquage.

Il paraît naturel que de faibles variations de luminances soient moins facilement détectables si l'image a une structure spatiale complexe. En effet, la fonction de sensibilité au contraste (FSC)⁹ (3.2.2) montre clairement que la détection varie en fonction du contenu fréquentiel de l'image visualisée. Afin de tenir compte de cet aspect, les auteurs utilisent ici la FSC de Barten pour pondérer les résultats obtenus dans la partie précédente :

$$S_{CSF}(f, L) = a.f.exp(-b.f)\sqrt{1 + 0,006.exp(b.f)} \quad (6.6)$$

Dans cette équation,

- S_{CSF} est la sensibilité au contraste.
- $a = 440(1 + 0,7/L)^{0,2}$
- $b = 0,3(1 + 100/L)^{0,15}$
- L est la luminance moyenne du dispositif de visualisation
- f est la fréquence spatiale considérée

Le facteur de pondération est obtenu de la manière suivante :

$$F_{CSF}(f) = \frac{\delta L_{CSF}(f)}{\delta L(L_a)} \quad (6.7)$$

où

- $\delta L_{CSF} = L/S_{CSF}$
- L est la luminance moyenne du dispositif de visualisation

Dans la section 3.2.2, nous avons indiqué que la vision des contrastes est le résultat de la contribution de plusieurs mécanismes passe-bandes centrés sur des fréquences différentes. Pour refléter cet état de fait, Ramasubramanian *et al* utilisent une approche multi-échelles pour calculer le facteur F_{CSF} . Ils utilisent pour cela une décomposition de type pyramide gaussienne (de manière identique à [55]), appliquent l'équation précédente à chaque niveau de la pyramide, puis somment les contributions.

Le masquage, évoqué dans la section 3.2.2, peut lui aussi influencer la détection des différences de luminance. Le facteur utilisé par les auteurs pour la prise en compte du masquage est issu d'une fonction donnée par Daly :

$$F_{masking}(C_n) = (1 + (0,0513(392,498.C_n)^{0,7})^4)^{1/4} \quad (6.8)$$

⁹Rappelons que cette fonction fournit la sensibilité au contraste (égale à l'inverse du plus petit contraste détectable) en fonction de la fréquence spatiale

ou C_n est le contraste masquant normalisé.

Une fois tous ces facteurs calculés, le seuil de luminance minimal discernable est obtenu de la manière suivante :

$$\delta L_r(x, y) = \delta L(L_a) \times F_{spatial}(x, y) \quad (6.9)$$

$F_{spatial}(x, y)$ est obtenu en sommant les produit $F_{masking} \times F_{CSF}$ pour chaque bande de fréquence de la pyramide gaussienne.

L'algorithme de rendu perceptuel

Ramasubramanian *et al* se proposent d'utiliser leur modèle pour guider un processus de rendu en lancer de rayons de type suivi de chemins. Le principe est de comparer deux images successives issues du rendu en calculant directement leur différence en luminance, et de stopper le calcul sur les pixels dont la différence de luminance est inférieure au seuil de perception fourni par le modèle. Ce dernier a toutefois un défaut de taille : la décomposition pyramidale et les traitements qui s'ensuivent sont très exigeants en termes de temps CPU, et risquent d'alourdir les temps de calculs au lieu de les diminuer.

Pour contourner ce phénomène, les auteurs pré-calculent la partie contraste du modèle avant de commencer le calcul de l'éclairage indirect : pour ce faire, ils utilisent le fait que l'éclairage indirect ne provoque pas (ou très peu) l'apparition de hautes fréquences dans l'image. Il est donc possible d'obtenir une très bonne approximation du facteur $F_{spatial}$ en appliquant le modèle sur une image sans éclairage indirect, obtenue avec un terme ambiant constant.

Globalement, la méthode de rendu se présente de la manière suivante :

1. Une image initiale est générée est utilisant un terme ambiant constant. Le facteur $F_{spatial}$ est pré-calculé.
2. Une image prenant en compte l'éclairage global est générée avec une précision faible
3. L'image de l'étape courante est comparée avec celle de l'étape précédente en calculant directement la différence de luminance entre ces deux images. Le seuil différentiel pour chaque pixel est calculé pour cette carte en multipliant le facteur $F_{spatial}$ par la luminance du pixel.
4. L'algorithme s'arrête si la différence de luminance est inférieure au seuil différentiel pour tous les pixels.
5. Dans le cas contraire, des échantillons sont relancés en fonction de la valeur de cette différence par rapport au seuil différentiel (la proportion d'échantillons relancés est proportionnelle à l'écart entre les deux). Si cette différence est nulle, il n'y aura plus de nouvel échantillon à cet endroit.
6. L'algorithme retourne à l'étape 4.

Résultats et réflexions

Quelques résultats de cette méthode sont visibles sur la figure 6.11. Il est possible de voir la solution de référence, issue d'un calcul complet en suivi de chemins, la solution perceptuelle obtenue avec la même scène, la carte des seuils différentiels (plus le pixel est blanc, plus le seuil est élevé) ainsi que la carte de densité des échantillons utilisés. Sur cette dernière, nous pouvons notamment remarquer qu'il y a très peu d'échantillons sur le tapis, les défauts étant peu visibles dans ces zones à cause de la texture.

Au vu de ces résultats, le modèle semble performant. Grâce au précalcul du terme lié au contraste spatial, le coût d'évaluation du modèle est très faible. Les échantillons sont judicieusement placés et

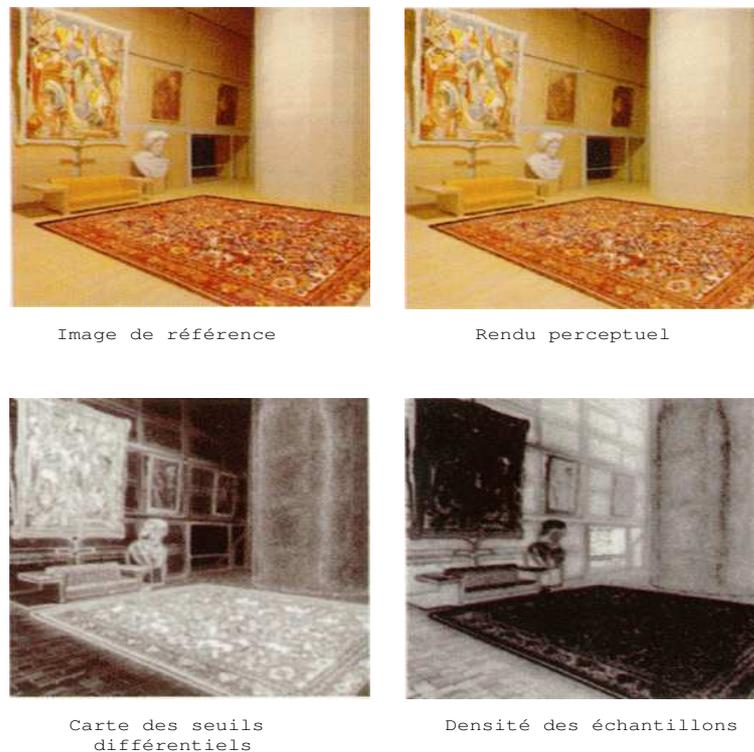


FIG. 6.11 – Résultats de la méthode perceptuelle de Ramasubramanian *et al* (issu de [57]).

l'image calculée avec l'algorithme perceptuel est visuellement identique à la solution de référence, et ce avec un pourcentage d'échantillons très faible (5% à 10 % environ) par rapport à ceux nécessaires pour la méthode classique.

Toutefois, il faut imposer de sérieuses restrictions pour obtenir un calcul rapide du facteur fréquentiel du modèle. Par exemple, les caustiques lumineuses ne sont la plupart du temps pas dues à l'éclairage indirect, et doivent donc être générées pendant cette phase préliminaire, ce qui alourdira considérablement les temps de calcul. Cette méthode interdit également les milieux participants, pour les mêmes raisons. D'autre part, il est très difficile à l'heure actuelle de définir des seuils différentiels en couleur. Pour cette raison, le modèle de vision ne prend en compte que la composante achromatique des pixels. Ceci implique qu'une importante différence entre deux couleurs isoluminantes ne sera pas détectée et donc ignorée dans le facteur fréquentiel.

6.4 Notre approche : Un rendu perceptuel progressif

Nous allons maintenant présenter nos travaux en matière de rendu perceptuel. Notre but est de réunir les deux approches, perceptuelles et progressives, présentées dans les sections précédentes.

Tous les algorithmes progressifs sont basés sur des critères empiriques. Dans la méthode de Guo par exemple, les critères de subdivision des cellules sont des mesures de contraste de Michelson. Mais, perceptuellement parlant, une telle mesure n'a pas beaucoup de sens si elle est isolée de son contexte. Il pourrait donc s'avérer intéressant d'utiliser, en lieu et place de ces critères, un modèle de vision.

D'autre part, les algorithmes perceptuels de la section précédente ont un défaut commun : la mé-



FIG. 6.12 – Comparaison algorithme de Guo avec et sans critères perceptuels

thode de rendu utilisée est brutale. La convergence vers l’image se fait donc relativement lentement, même si le modèle de vision permet un meilleur échantillonnage de la scène. Il serait peut être judicieux d’utiliser les méthodes progressives de manière à obtenir, à partir des échantillons déterminés perceptuellement, une image intermédiaire de meilleure qualité.

Nous proposons dans cette section deux tentatives réunissant donc les deux aspects. Dans un premier temps, nous reprendrons l’algorithme de Guo en remplaçant les critères de subdivision par une mesure perceptuelle. Nous présenterons dans un deuxième temps une méthode progressive de notre cru dérivée de celles de Guo([24]) et Pighin([56]). Pour classifier les cellules, nous utilisons notre métrique entre images de la section 5.3.1.

6.4.1 Essai préliminaire : pilotage de l’algorithme de Guo par notre modèle de vision

Nous avons voulu tester notre métrique pour voir si elle était capable de remplacer avantageusement des critères empiriques dans un algorithme de rendu progressif. Dans la méthode de Guo, le principal inconvénient est que tous les blocs sont systématiquement subdivisés à chaque itération. Bien que cela permette de garantir la convergence vers l’image finale calculée entièrement, nous pensons qu’il serait possible de laisser invariants un certain nombre de blocs qui seraient devenus suffisamment proches de leur état final, ce qui permettrait d’économiser une grande quantité de calculs. Nous avons donc choisi de modifier l’algorithme de Guo de la manière suivante :

- pour remplacer les critères de subdivision, nous utilisons notre métrique sur deux images successives issues du rendu.
- Nous affectons à chaque bloc de la subdivision une valeur de JND correspondant à la moyenne des JNDs des pixels internes au bloc.
 - Si cette valeur est supérieure à un seuil S_{sup} donné, le bloc sera classé “edge”.
 - Si cette valeur est inférieure à un seuil S_{inf} donné, le bloc sera classé “final”. Cette classification, qui n’existe pas dans l’algorithme de Guo, signifie que le bloc ne sera plus subdivisé par la suite.
 - Si cette valeur est comprise entre S_{inf} et S_{sup} , le bloc sera classé comme “smooth”.

Les résultats obtenus sont encourageants : la figure 6.12 montre l’image obtenue avec l’algorithme de Guo traditionnel et sa version “perceptuelle” ainsi que la proportion d’échantillons utilisée pour reconstruire l’image. Il apparaît évident qu’à qualité équivalente, cette dernière est plus faible pour l’algorithme perceptuel. Nous pouvons donc supposer qu’un modèle de vision sera plus efficace pour guider un algorithme de rendu qu’un simple critère empirique.

6.4.2 Notre méthode perceptuelle progressive

L’essai précédent montre que le guidage par un modèle de vision est généralement plus fiable qu’en utilisant des critères empiriques. Toutefois, l’algorithme de Guo, même si il semble être la meilleure

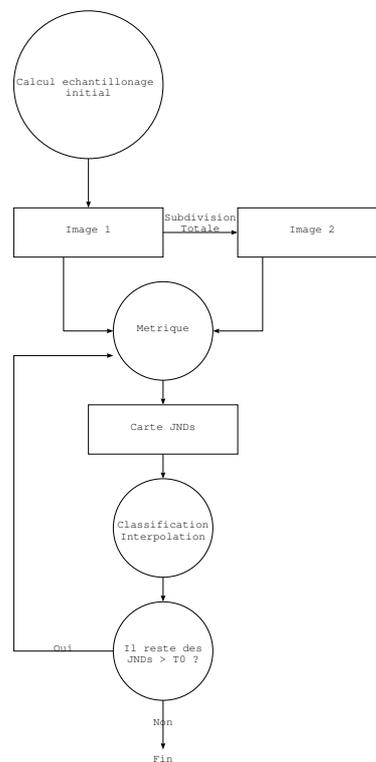


FIG. 6.13 – Schéma de fonctionnement de notre méthode progressive perceptuelle

méthode actuelle, présente un inconvénient particulièrement gênant dans notre approche : la grille d'échantillonnage est régulière et ne tient absolument pas compte de la géométrie de la scène. Ceci est partiellement corrigé en utilisant l'interpolation directionnelle mais dans ce cas l'ensemble initial d'échantillons pourrait être mieux placé pour accélérer la convergence. Pour palier ces défauts, nous voulions concevoir une méthode progressive originale s'inspirant à la fois de la carte de cohérence directionnelle et de méthodes utilisant une triangulation du plan basée sur la géométrie de la scène, comme Tapestry par exemple.

Le schéma de fonctionnement de notre méthode (figure 6.13) est très similaire à celui d'un algorithme progressif classique auquel il faudrait ajouter l'évaluation d'un modèle de vision simplifié. Notre plan image est subdivisé avec une triangulation de Delaunay. Chaque triangle est classé "Final", "Smooth", ou "Edge" (cf section 6.4.1). Selon sa classe, le triangle pourra éventuellement être subdivisé et interpolé de manière différente. Pour déterminer la classe de chaque triangle, nous appliquons notre métrique entre image sur deux images successives issues du rendu. Chaque triangle se voit alors affecter une JND correspondant à la moyenne des JNDs des pixels situés dans son intérieur. La valeur de JND d'un triangle détermine ensuite sa classe. Nous allons maintenant détailler le déroulement de notre algorithme étape par étape.

Étape 1 : initialisation

La première chose à faire est d'initialiser cet algorithme : en effet, notre métrique ne sait que comparer deux images. Ces deux premières images ont une importance cruciale : si l'algorithme est mal initialisé, il sera beaucoup moins efficace.

Nous générons deux images initiales de la manière suivante :

- Nous repérons les contours de l’image, puis nous effectuons un échantillonnage aléatoire du plan image en augmentant la probabilité d’apparition des échantillons lors de la proximité d’un contour. Un échantillon aura donc plus de chances d’apparaître si il est plus proche d’un contour. Par contre, la probabilité qu’il apparaisse sur le contour est nulle¹⁰.
- Nous interpolons ensuite les couleurs des sommets des triangles via un lissage de Gouraud, et nous affichons directement le résultat grâce au hardware graphique de la machine de travail, ce qui fournit notre première image.
- La deuxième est obtenue en subdivisant tous les triangles de la première image. Cette subdivision consiste à placer un nouvel échantillon au centre de chaque triangle, puis de re-trianguler, toujours selon les critères de Delaunay, le résultat.

Étape 2 : comparaison des deux images et classification des triangles

Notre méthode est basée sur la comparaison de deux images successives. Nous pouvons raisonnablement estimer que s’il existe, sur deux images consécutives issues du rendu, des zones où les différences ne sont plus perceptibles, alors ces zones ne nécessitent pas de subdivision¹¹.

Pour chaque itération, nous utilisons donc notre métrique simple (non adaptative) pour comparer l’image obtenue avec celle de l’itération précédente. Nous obtenons donc une valeur de JND pour chaque pixel. Pour affecter une JND à chaque triangle, nous calculons la moyenne des JNDs des pixels situés à l’intérieur de ce triangle. La classification découle ensuite directement de cette valeur moyenne T_{moy} . Nous utilisons deux seuils S_{inf} et S_{sup} pour caractériser les trois classes :

- si $T_{moy} < S_{inf}$, alors le triangle est considéré comme “final”
- si $S_{inf} < T_{moy} < S_{sup}$, alors le triangle est considéré comme “smooth”
- si $T_{moy} > S_{sup}$, alors le triangle est considéré comme “edge”

Au final, nous obtenons une carte de triangles classifiés, qu’il nous faut maintenant afficher correctement.

Étape 3 : Interpolation des triangles et subdivision

La valeur de la JND d’un triangle détermine sa subdivision éventuelle et son interpolation.

- Si le triangle est classé “final”, aucune interpolation n’est effectuée et ce triangle ne sera plus subdivisée par la suite.
- Si le triangle est classé “smooth”, une simple interpolation de type “Gouraud” est réalisée à partir des sommets.
- Si le triangle est classé “edge”, une interpolation plus complexe intervient.

L’interpolation des triangles “edge” est inspirée de la méthode utilisée par Guo dans [24] (décrite dans la section 6.2.4), ayant pour but d’exploiter la cohérence directionnelle. Nous appliquons exactement les mêmes principes, en remplaçant les blocs par des triangles. Le contour des triangles “edge” est donc entièrement calculé¹², et une direction de moindre discordance en est extraite. L’interpolation se fait alors linéairement, selon cette direction de moindre discordance. Toutefois, nous avons ici éliminé la distinction “edge simple / edge complexe” présente dans l’algorithme initial. Ensuite, les triangles qui ne sont pas classés “final” seront subdivisés, et l’algorithme reprend à l’étape 2. Il

¹⁰Nous cherchons ici à capturer les discontinuités de l’image, une cellule alignée sur une discontinuité n’est pas favorable à une interpolation de type “Guo”.

¹¹Ce point peut prêter à discussion, nous y reviendrons ultérieurement

¹²... en tenant compte du fait que les interpolations dues à l’algorithme de rendu lui même (en l’occurrence, les vecteurs lumineux), peuvent avoir lieu.

s'arrête lorsqu'il ne reste plus aucun triangle à subdiviser. La figure 6.13 présente un récapitulatif du fonctionnement de l'algorithme.

6.4.3 Résultats

Nous allons dans cette section présenter les résultats de notre méthode progressive perceptuelle. Toutes les images ont été calculées en utilisant un ordinateur de type Pentium 4, doté d'un Gigaoctet de mémoire vive et d'une carte accélératrice nVidia Geforce 3. Toutes les images sont calculées dans une résolution de 512x512 pixels. Le logiciel utilisé est une version modifiée par nos soins de YART, logiciel créé par Marc Roelens ([58]) et muni du calcul des vecteurs lumineux de Jacques Zaninetti([87]), tous deux présentés en annexe.

Dans un premier temps, nous testerons notre algorithme avec quelques scènes créées par Jacques Zaninetti et Xavier Serpaggi. Nous présenterons pour chaque test le résultat obtenu avec le rendu à base de vecteurs lumineux, le résultat calculé avec notre méthode, ainsi la carte des échantillons utilisés pour l'approximation. Nous indiquerons également le taux d'échantillonnage et le temps de calcul. Pour ces premiers tests, le seuil inférieur sera systématiquement fixé à zéro ; nous considérons en effet que le triangle doit arrêter sa subdivision uniquement lorsque la JND qui lui est affectée est nulle.

Dans un deuxième temps, nous présenterons un essai de variation sur les deux paramètres afin d'étudier leur influence sur l'échantillonnage de la scène.

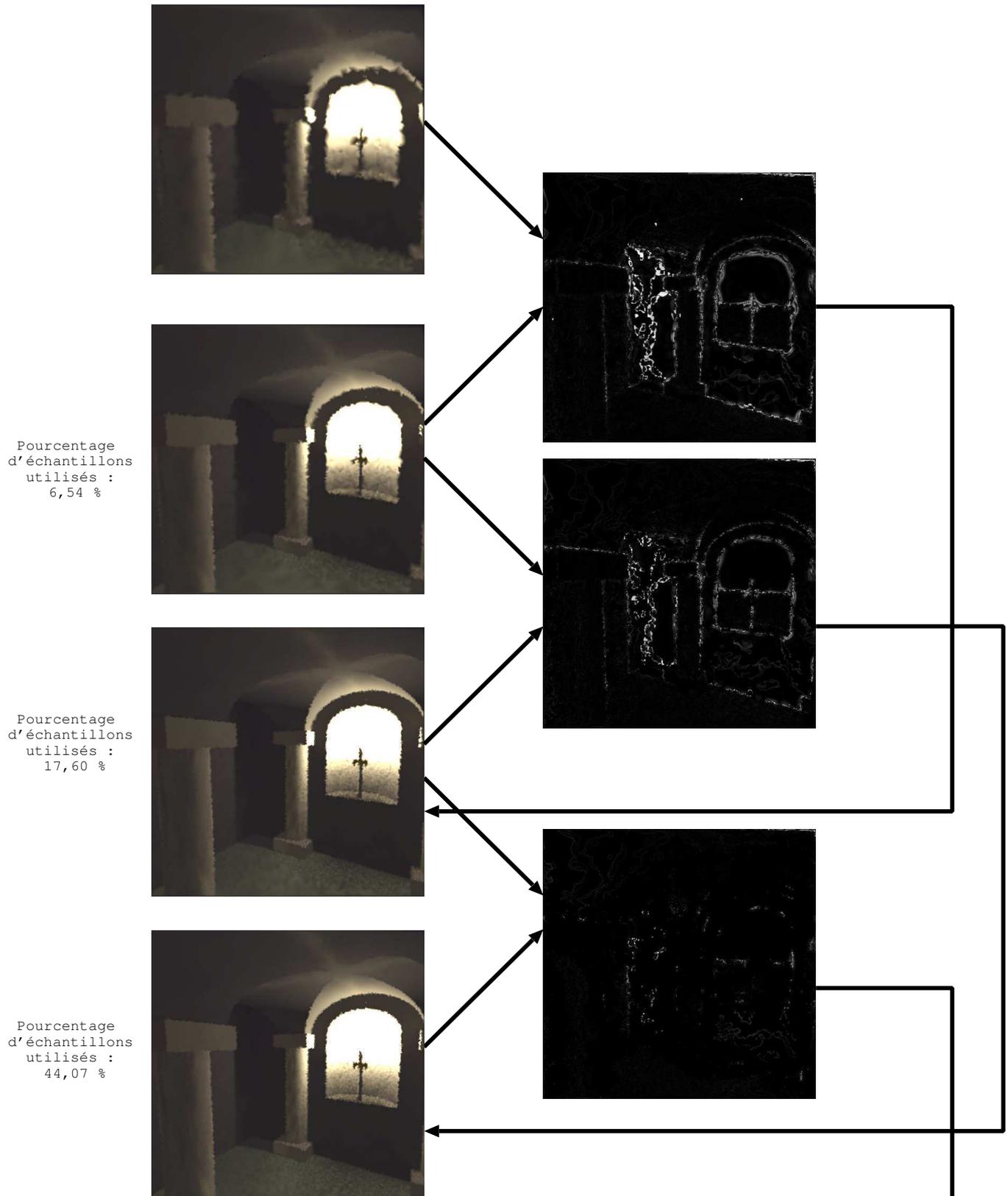
Scène "Crypte"

Cette scène représente une crypte sombre comprenant une alcôve éclairée. Sur la figure 6.14, nous présentons la succession des images intermédiaires pour cette scène. Comme indiqué lors de la description de la méthode, une image de distance est évaluée à partir de deux images successives de l'algorithme de rendu. La carte de distance obtenue conditionne la subdivision des cellules pour l'image suivante. Signalons que toutes les itérations sont présentées ; le temps de calcul nécessaire pour obtenir la dernière est de 68 secondes, avec des seuils inférieur et supérieur respectivement fixés à 0 et 5 JNDs. Il est notamment assez visible que l'échantillonnage se fait dans les zones qui en ont le plus besoin, en particulier la zone centrale de l'image, où les détails sont particulièrement perceptibles.

La figure 6.15 permet de comparer l'image obtenue en fixant le seuil supérieur à 5 JNDs et l'image obtenue avec la méthode des vecteurs lumineux. Cette dernière s'obtient en 90 secondes.

Scène "Pièce avec chaise"

Cette scène a la particularité d'être entièrement illuminée par un éclairage indirect : en effet, la source lumineuse présente dans le lampadaire n'éclaire que le plafond, qui renvoie ensuite la lumière dans la scène. Si nous calculions cette scène avec un algorithme ne prenant pas en compte l'éclairage indirect, seul le plafond serait visible. Le pourcentage d'échantillonnage est de 54 % et le seuil supérieur est fixé à 9 JNDs (cf figure 6.16). Toutefois, nous avons constaté un défaut majeur de notre méthode sur cette scène : malgré ce pourcentage, l'image traditionnelle s'obtient plus rapidement que l'image approximée ; ceci est dû au fait que la scène se calcule très rapidement, le surcoût de la métrique et du processus d'interpolation n'est pas négligeable dans ce cas. En règle générale, notre méthode fonctionne d'autant mieux que le temps de calcul de l'image est long.



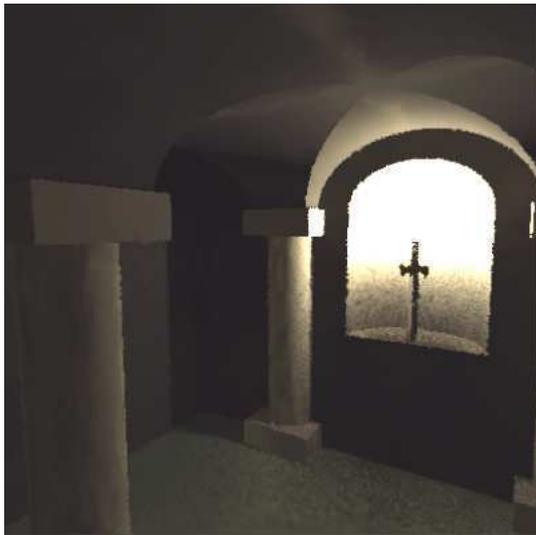


Image obtenue avec
notre méthode

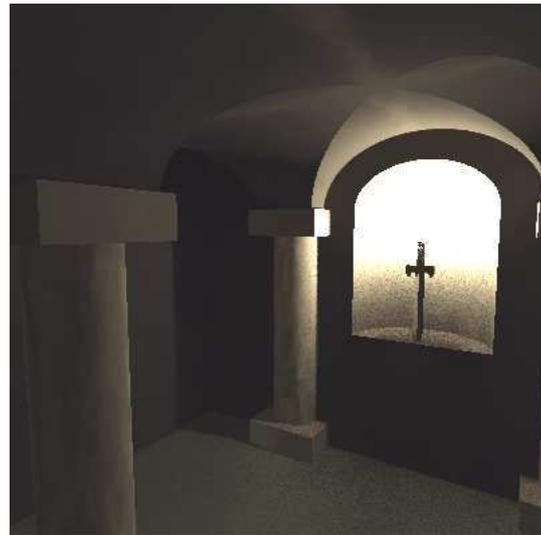
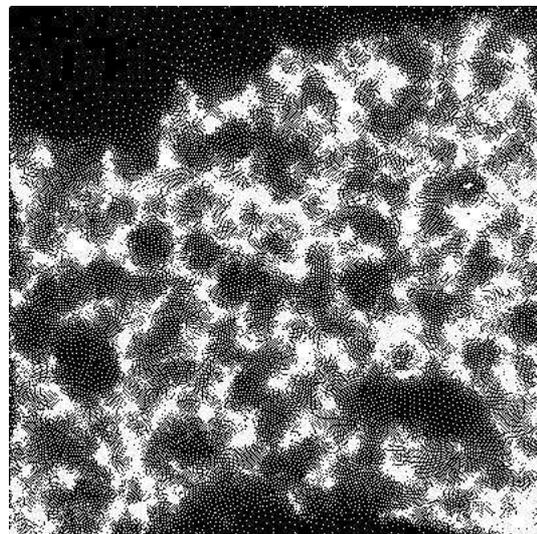


Image obtenue avec
les vecteurs lumineux



Carte d'échantillonnage

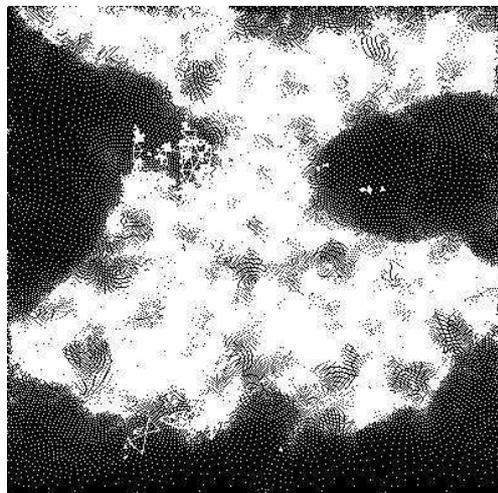
FIG. 6.15 – Comparaison entre l'image approximée et l'image calculée entièrement sur la scène "crypte", seuil fixé à 5 JNDs



Image approximée avec
notre méthode



Image calculée avec
les vecteurs lumineux



Carte d'échantillonnage

FIG. 6.16 – Comparaison entre l'image approximée et l'image calculée entièrement sur la scène "pièce avec chaise", seuil fixé à 9 JNDs

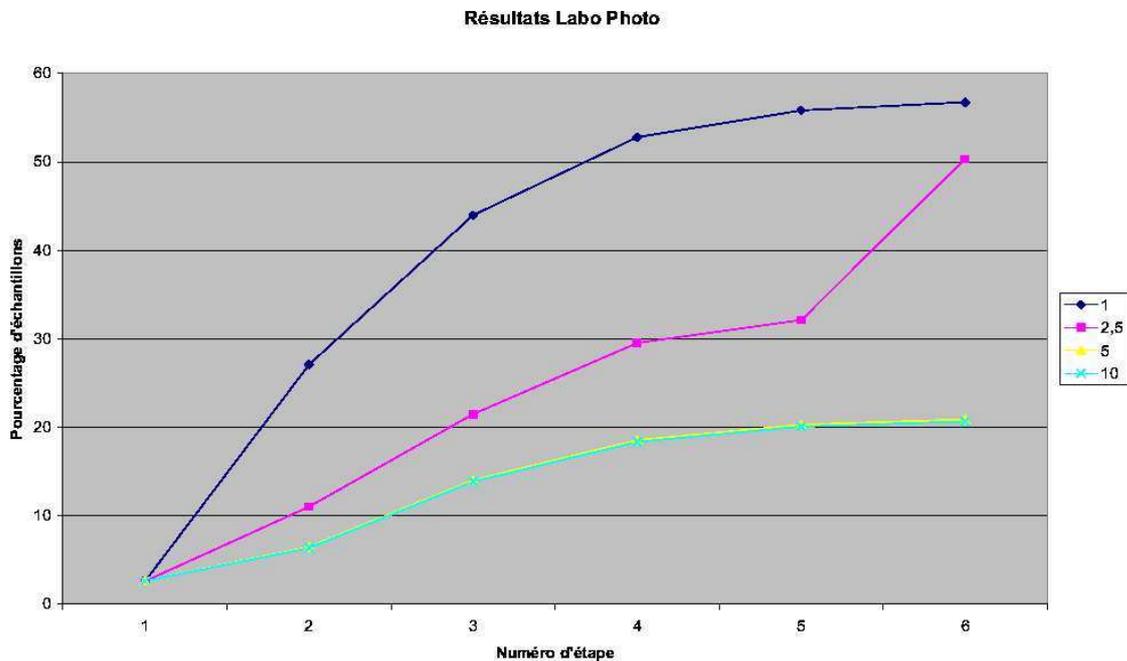


FIG. 6.17 – Comparaison entre l’image approximée et l’image calculée entièrement sur la scène “labo photo”, seuil fixé à 1 JND

Scène “Labo Photo”

Tout comme la scène précédente, cette scène est entièrement illuminée indirectement. Elle représente un laboratoire photographique à moitié plongé dans l’obscurité, dont seule la pièce voisine contient une source de lumière. Dans le laboratoire, la lumière provient donc uniquement d’une porte entrebaillée. Sur la figure 6.17, nous pouvons comparer l’image issue de notre méthode progressive avec celle obtenue avec la méthode des vecteurs lumineux. Le seuil supérieur est fixé à 1 JND. L’image calculée avec les vecteurs lumineux nécessite 5 minutes de calcul, alors que notre méthode obtient une image visuellement identique en 4 minutes 19 secondes. La carte d’échantillonnage montre bien que l’algorithme se concentre sur l’illumination et délaisse les zones uniformes.

Scène “Salle à manger”

Cette scène va faire l’objet d’un test particulier. La scène représente une salle à manger garnie de meubles, avec un miroir et un tableau sur le mur (voir figure 6.18).

Pour cette scène, nous allons faire varier le seuil supérieur S_{sup} entre 1 et 10, fixer le seuil inférieur S_{inf} à zéro et observer les résultats obtenus. Les résultats sont présentés sur la figure 6.19. Nous avons représenté sur cette figure les images successives obtenues lors du rendu avec notre méthode de la salle à manger, pour les seuils S_{sup} fixés à 10, 5, 2,5 et 1,0.

Nous pouvons constater sur cette figure que les images obtenues avec les seuils 10 et 5 identiques ; ces seuils sont donc probablement trop élevés et les triangles sont tous subdivisés de la même manière. En ce qui concerne les seuils 2,5 et 1, le meilleur compromis entre qualité du résultat et échantillonnage semble être le seuil de 2,5. Nous allons détailler le choix des paramètres dans la section suivante.



FIG. 6.18 – La scène “Salle à manger”

6.4.4 Discussions sur les résultats et les paramètres

L'exemple précédent montre que le choix des paramètres a une influence sur l'évolution du calcul. Nous voulons maintenant, comme nous l'avons fait pour la distance adaptative, proposer un choix de paramètres idéaux permettant d'obtenir le meilleur résultat le plus rapidement possible. Pour ce faire, nous allons faire varier les différents seuils sur les scènes précédentes, et examiner les résultats.

Tests avec variation du seuil supérieur

Les figures 6.21 et 6.22 montrent l'évolution du pourcentage d'échantillons sur les cinq premières images issues de l'algorithme de rendu, en faisant varier le seuil supérieur entre 0 et 10. Le seuil inférieur est, quand à lui, fixé à zéro. Le premier test concerne une cornell box (figure 6.20), scène géométriquement peu complexe. Les courbes mettent en évidence le fait que changer le seuil supérieur diminue simplement la qualité générale pour cette scène. Pour les autres scènes par contre, nous pouvons remarquer que la courbe $S_{sup} = 1.0$ se démarque très nettement des autres. Il est donc probable que le seuil "idéal" se trouve entre 1 et 2,5, ce qui correspond à la définition des JNDs. Les tests effectués en laboratoire semblent montrer que 2,5 est une bonne valeur.

Tests avec variation du seuil inférieur

Les figures 6.23 et 6.24 montrent l'évolution du pourcentage d'échantillons sur les six premières images issues de l'algorithme de rendu, en faisant varier le seuil inférieur entre 0 et 0,5. Exception faite de la scène du laboratoire photo, la nette différence entre la courbe $S_{inf} = 0$ et les autres montre très nettement que la plupart des triangles semblent avoir une JND propre comprise entre 0 et 0,5 ; cela ne peut se produire que si les images intermédiaires sont très proches, ce qui est généralement le cas. Il est donc déconseillé de fixer S_{inf} à une valeur supérieure à 0,5, sous peine d'obtenir un résultat très approximatif.

6.5 Conclusions

Nous avons vu dans ce chapitre un éventail assez large de méthodes destinées à produire le plus rapidement possible une image intermédiaire aussi proche que possible de l'image finale. Notre contribution, réunir les approches perceptuelle et progressive, peut significativement réduire les temps de calcul. Dans les tests présentés, il est pratiquement systématiquement deux fois inférieur au temps nécessaire pour obtenir l'image entière. Mais nous pensons que l'avantage majeur de cette approche est qu'elle propose d'utiliser des critères moins difficiles à appréhender que pour une méthode progressive classique : en effet, un seuil perceptuel est significatif quelque soit le type d'image généré par l'algorithme : une JND correspond à une erreur qu'un observateur moyen a une probabilité de détection de 0,75.

Toutefois, il subsiste encore un certain nombre de problèmes. Le premier est représenté par le fait que nous comparons deux images successives issues de l'algorithme. Si jamais la variation entre ces deux images est suffisamment faible pour être inférieure au seuil inférieur, le calcul sera stoppé beaucoup trop tôt. L'idéal serait bien entendu de comparer l'image intermédiaire et l'image finale, mais c'est évidemment impossible. Nous présenterons une ébauche d'alternative à cette méthode dans le chapitre suivant, réservé aux perspectives de notre travail.

Le deuxième problème de notre méthode est qu'elle ne gère pas les textures. Ceci est dû au fait que le logiciel YART, utilisé dans ce mémoire, ne traite que des scènes modélisés en Constructive Solid

Geometry. Il est donc impossible de récupérer dans un prétraitement les textures projetées sur le plan image pour ensuite les intégrer dans le rendu de manière élégante. Toutefois, ce problème disparaîtra immédiatement si un autre logiciel de rendu est utilisé.

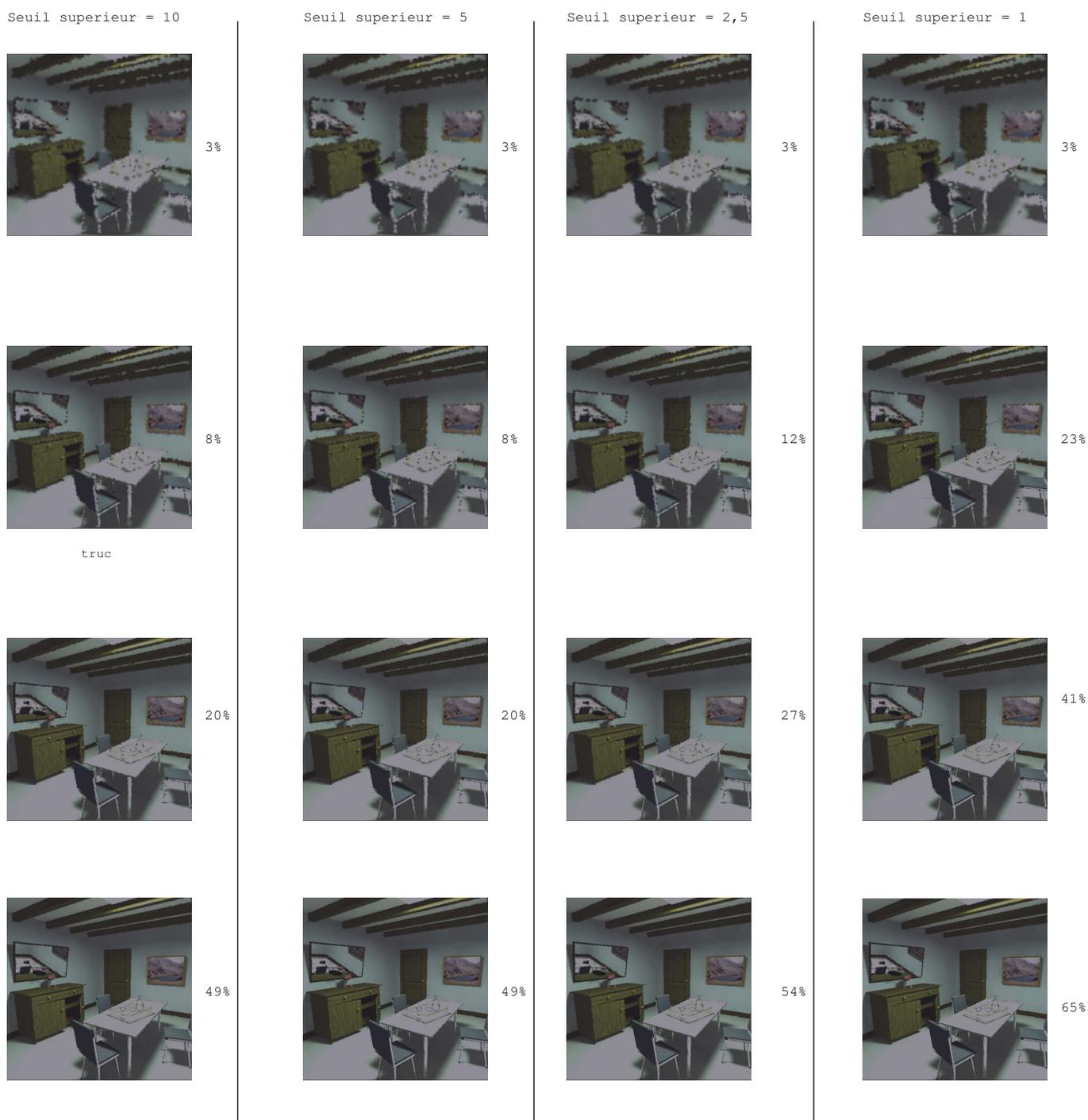


FIG. 6.19 – Résultats obtenus sur la scène “salle à manger” en faisant varier le seuil S_{sup}

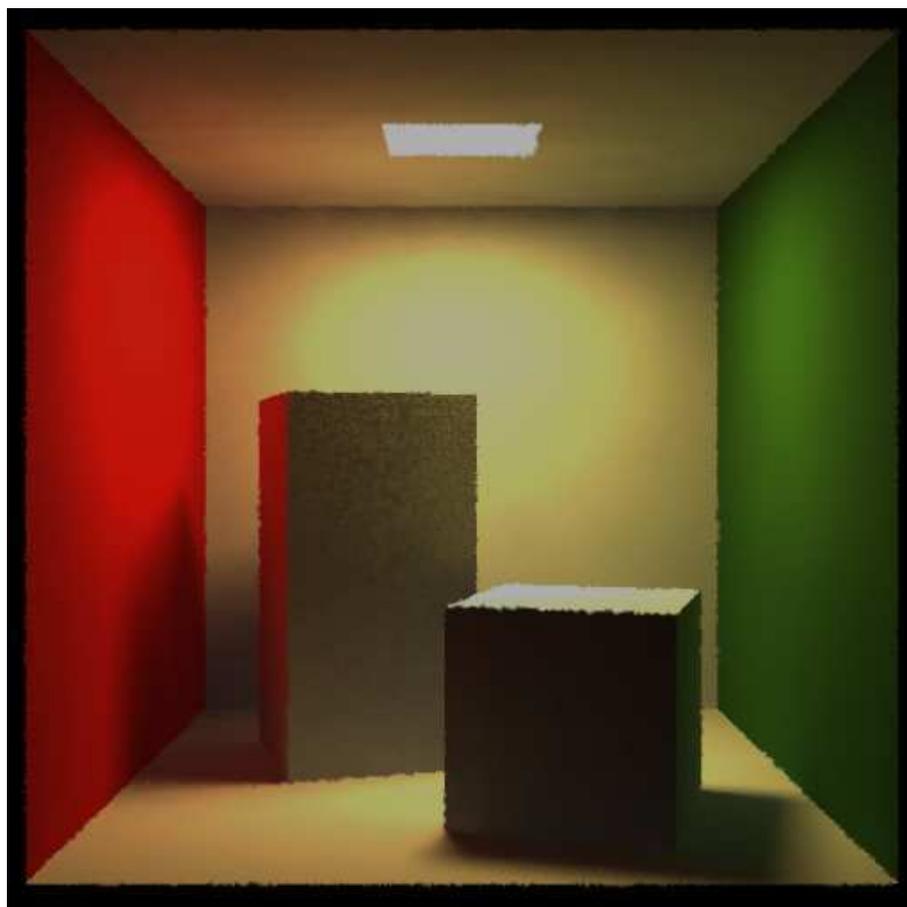


FIG. 6.20 – Cornell Box obtenue avec notre méthode. Le taux d'échantillonnage est de 45 %, les seuils sont fixés à $S_{inf} = 0$ et $S_{sup} = 2,5$

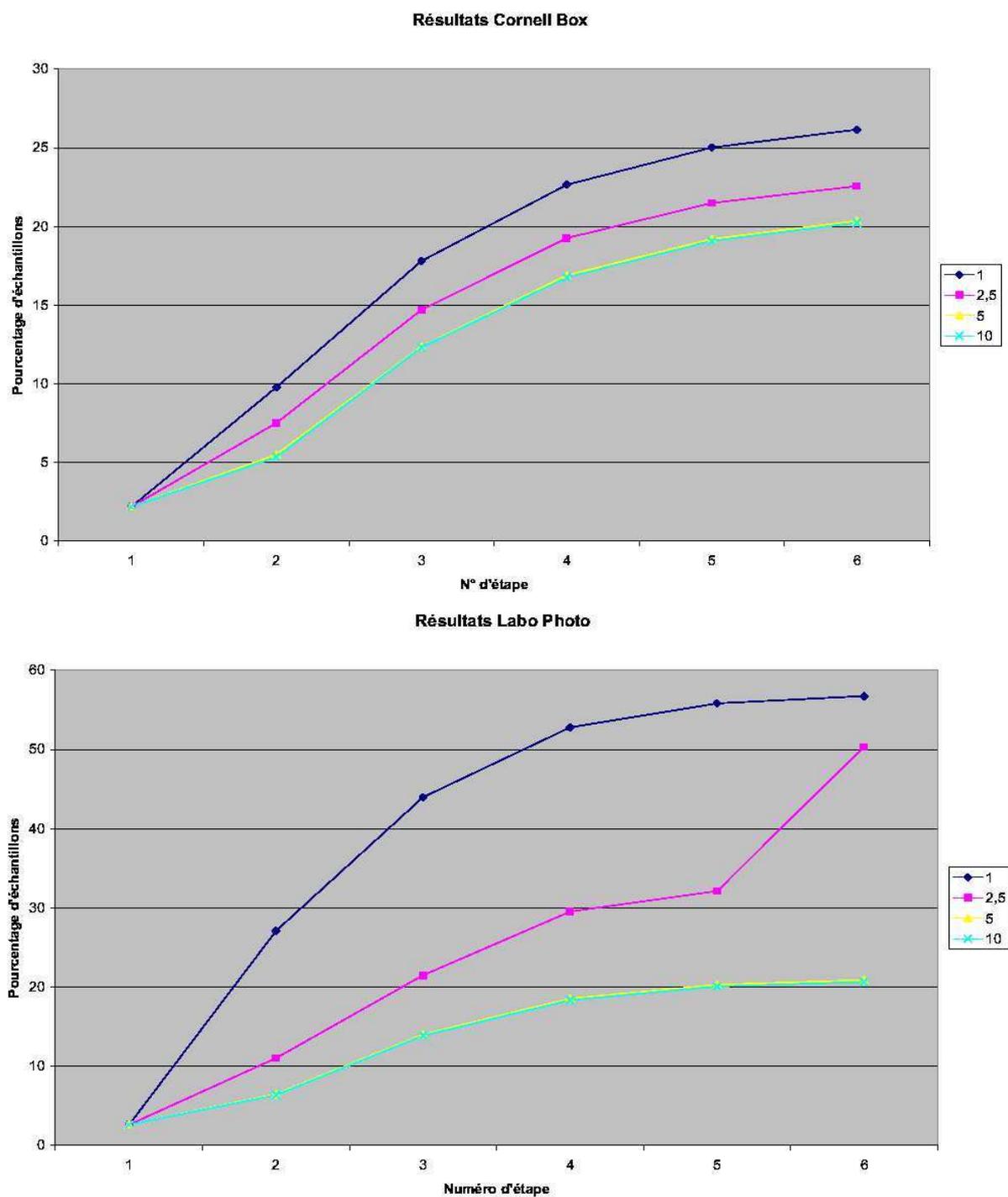


FIG. 6.21 – Résultats obtenus sur les scènes “Labo photo” et “Cornell Box” en faisant varier le seuil S_{sup}

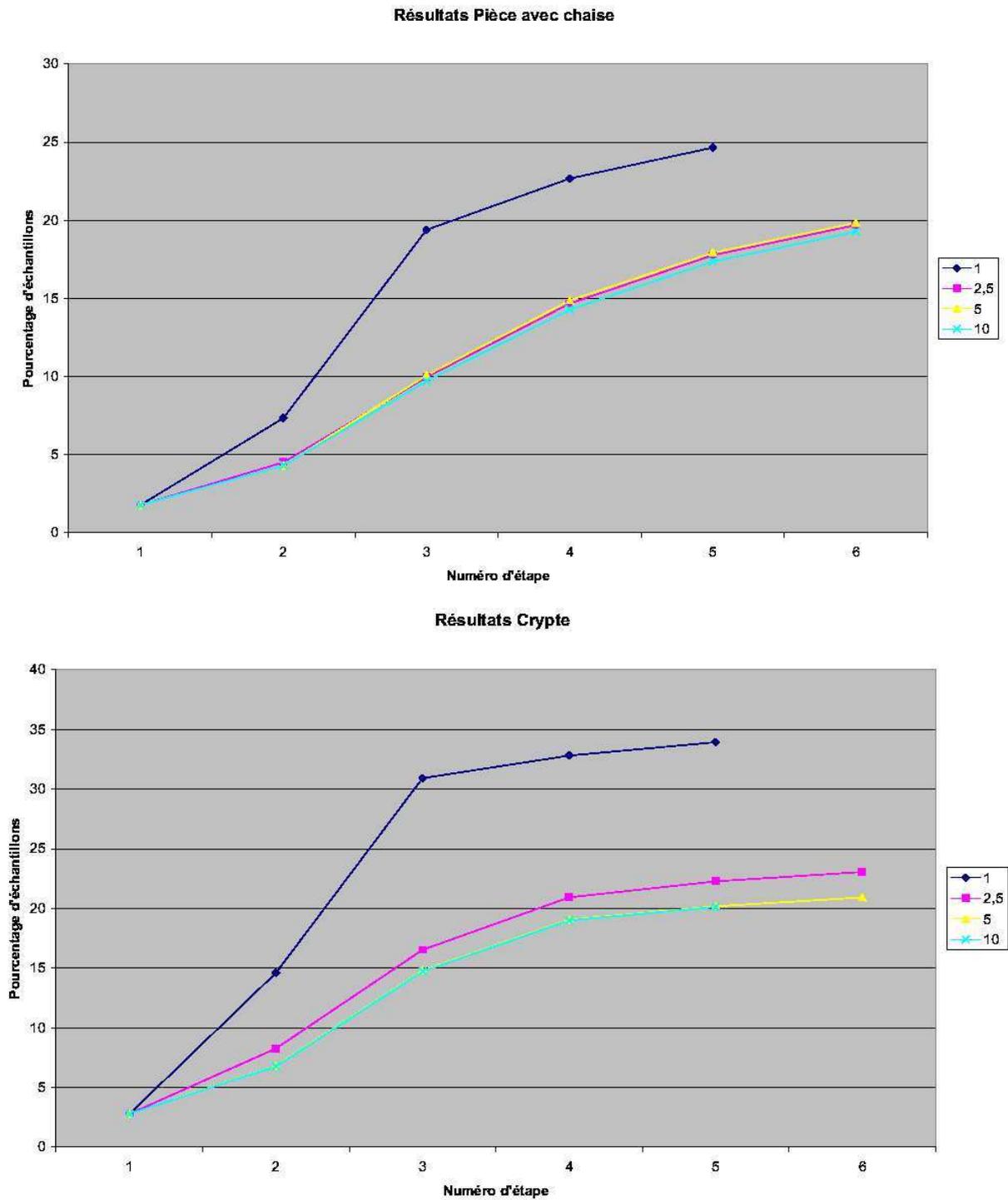


FIG. 6.22 – Résultats obtenus sur les scènes “Pièce avec chaise” et “Crypte” en faisant varier le seuil S_{sup}

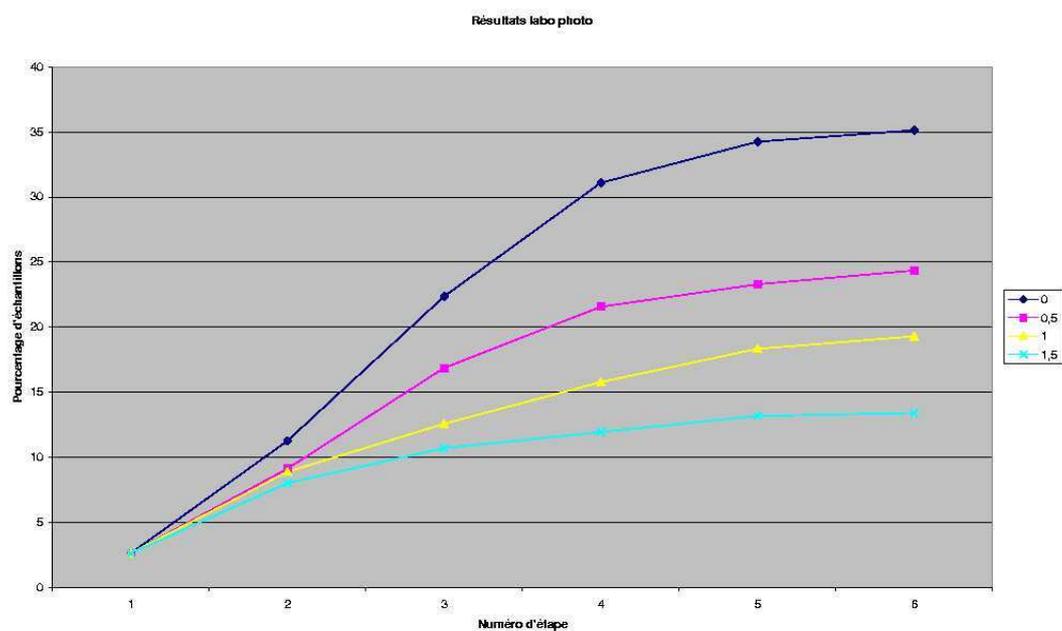
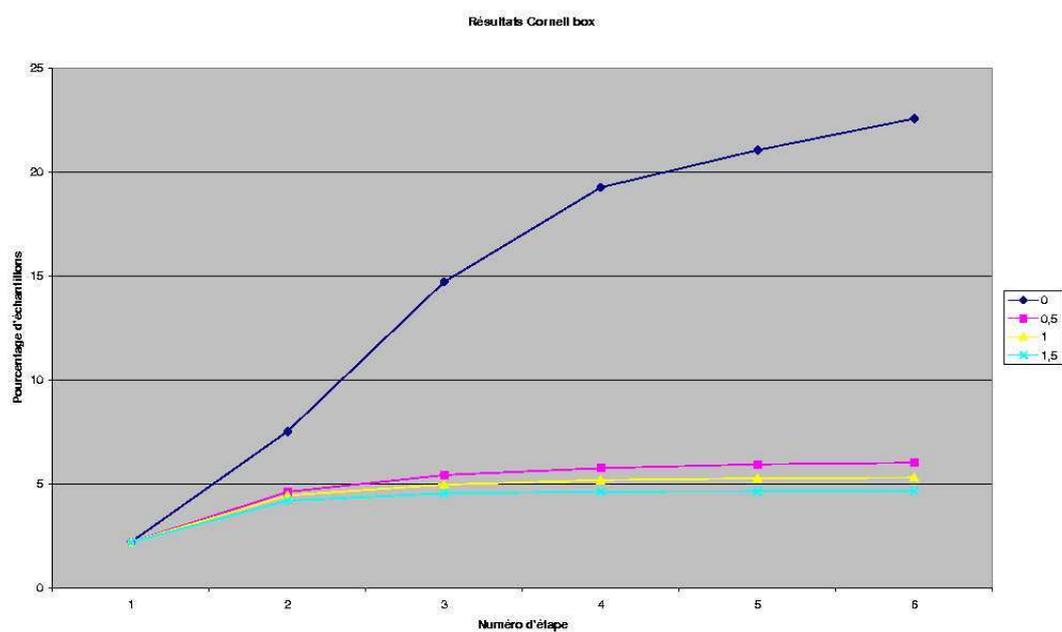


FIG. 6.23 – Résultats obtenus sur les scènes “Labo photo” et “Cornell Box” en faisant varier le seuil S_{inf}

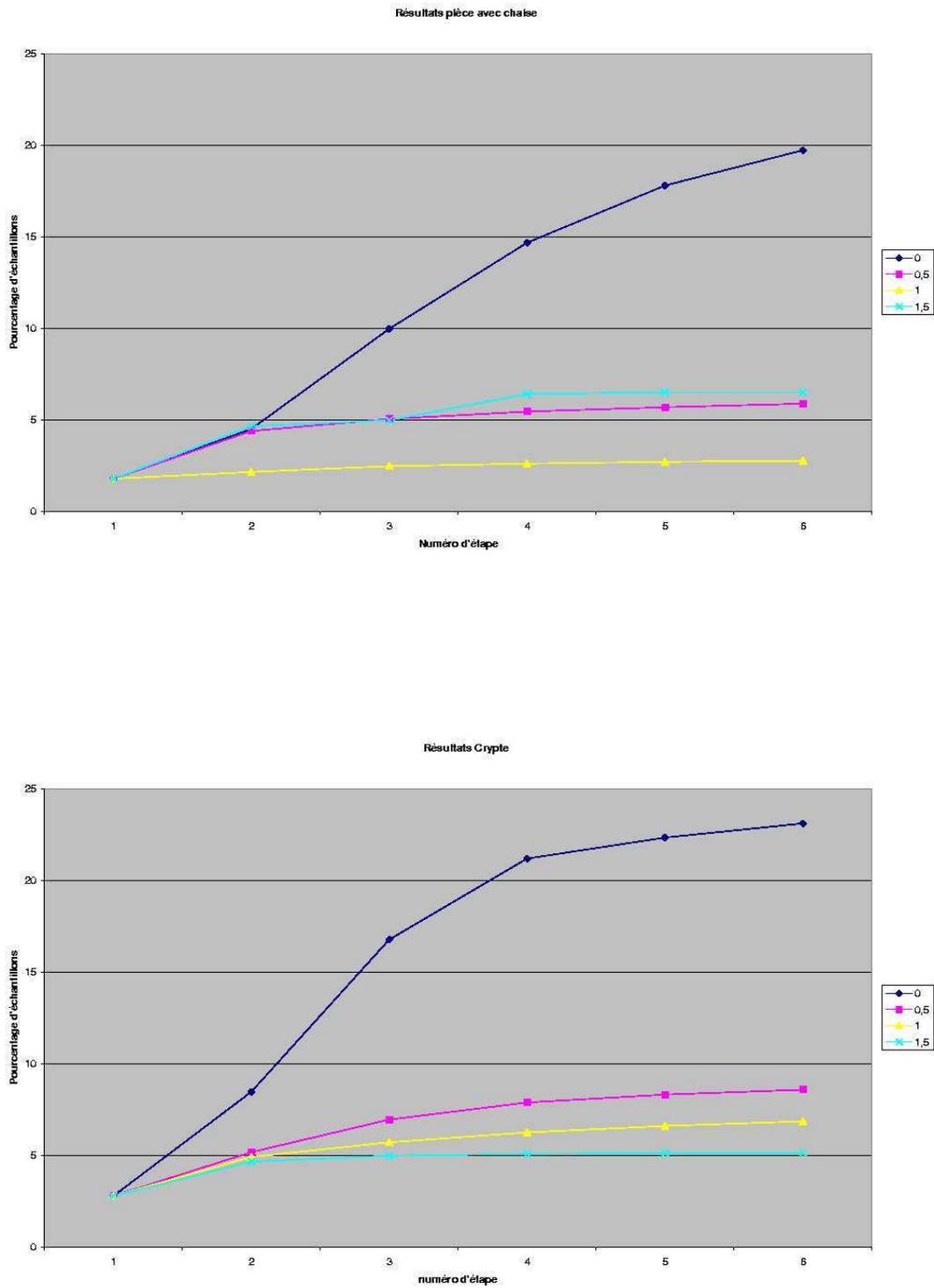


FIG. 6.24 – Résultats obtenus sur les scènes “Pièce avec chaise” et “Crypte” en faisant varier le seuil S_{inf}

Chapitre 7

Perspectives et conclusion

Lors de ce travail de thèse, nous avons eu un bon aperçu des recherches actuelles en termes de synthèse d'image prenant en compte des aspects perceptuels. La première conclusion qui vient à l'esprit est que, devant la sophistication des algorithmes actuels, il est aujourd'hui impensable de mettre ces aspects de côté. Toutefois, la majorité des approches suivent le même chemin. Dans cette section, nous allons essayer de donner quelques alternatives pour les travaux à venir, puis nous conclurons.

7.1 Aspects spectraux

7.1.1 Pourquoi du multi-spectral ?

Travailler en multi-spectral en synthèse d'image consiste à définir, stocker et prendre en compte lors des calculs le comportement spectral des différents éléments de la scène. Ainsi, l'imagerie multi-spectrale prend de plus en plus d'ampleur, et de l'avis général des spécialistes en la matière, cette approche est vouée, à plus ou moins long terme, à remplacer l'approche RGB.

Les calculs spectraux permettent en effet d'obtenir un niveau de réalisme inatteignable avec un système trichromatique. Lors du rendu d'un matériau dont la distribution spectrale est complexe, un algorithme RGB va se contenter d'échantillonner cette fonction sur seulement trois longueurs d'ondes, donnant du coup une approximation très médiocre de l'aspect visuel de ce matériau, alors que l'on peut s'attendre à une restitution plutôt fidèle avec des calculs spectraux. Ces derniers permettent également la simulation des matériaux fluorescents, ainsi que des phénomènes optiques bien spécifiques comme la diffraction ou la dispersion. La figure 7.1 illustre le résultat d'une simulation de la dispersion de la lumière dans un diamant.

La figure 7.2 illustre l'erreur que l'on peut commettre en omettant l'aspect spectral des matériaux, alors que la figure 7.1 montre un exemple de simulation de la dispersion. Enfin, dans le domaine de la simulation réaliste d'éclairage, les sources de lumières ont une importance cruciale et ne peuvent être correctement simulées avec un triplet RGB. L'image 7.3, représentant une cornell box éclairée avec une lampe à vapeur de sodium, montre les deux images calculées en RGB et en spectral. Là encore, du fait de l'échantillonnage peu judicieux du spectre d'émission de la lampe, l'image RGB semble bien peu réaliste.

Malheureusement, cette qualité de rendu a un prix : les données définissant la scène sont beaucoup plus volumineuses, une distribution spectrale devant être stockée au lieu d'un simple triplet rgb. D'autre part, les calculs ne se contentent plus de trois canaux chromatique mais de toute une distribution spectrale. Selon la manière dont cette distribution est échantillonnée, les temps de calculs peuvent



FIG. 7.1 – Simulation spectrale de la dispersion de la lumière dans un diamant. En vignette : photographie réelle d'un diamant (*issu de [69]*)

très vite devenir prohibitifs. La recherche actuelle vise donc à maintenir cette qualité de rendu tout en réduisant la taille des données et les temps de calcul.

7.1.2 Modèles de vision et multi spectral

Bien que les approches spectrales en synthèse d'images commencent à être relativement nombreuses, très peu font usage de critère perceptuels pour leur mise en œuvre. C'est pourtant une approche naturelle, car une image spectrale peut être conçue comme une donnée brute telle qu'elle arrive dans l'oeil, avant même d'être captée par les photorécepteurs. Nous allons dans cette section passer en revue ce que pourrait apporter un modèle informatique de la vision humaine dans le cadre de calculs spectraux.

Pour obtenir une description spectrale de notre scène, il nous faut en premier lieu savoir discrétiser efficacement une distribution spectrale. La question est de savoir quel schéma d'échantillonnage choisir. Un échantillonnage uniforme du spectre sera précis si le pas est suffisamment fin, mais consommera énormément de mémoire. Même si c'est un problème de moins en moins important de nos jours, la taille des données obtenue reste beaucoup trop importante : en considérant une image spectrale 512×512 et une discrétisation du spectre sur 64 pas uniformes, on atteint facilement le Gigaoctet. La solution est évidemment une décomposition adaptative du spectre. Gilles Rougeron a présenté dans [59] une méthode adaptative basée sur une analyse en composantes principales. Jean Claude Iehl a, quant à lui, proposé une nouvelle classe d'algorithmes de rendu spectral s'adaptant à la complexité de la simulation à réaliser, tout en contrôlant perceptuellement l'erreur commise. Sans rentrer dans les détails, signalons que sa méthode se base sur des expressions symboliques de rendu, évaluées partiellement, qu'il utilise pour le calcul de l'éclairage local et global. Le contrôle percep-



FIG. 7.2 – Simulation spectrale (à gauche) et RGB (à droite) d'une bouteille de vin. La différence est assez flagrante sur le vin blanc, qui apparaît vert en RGB ! (issu de [69])

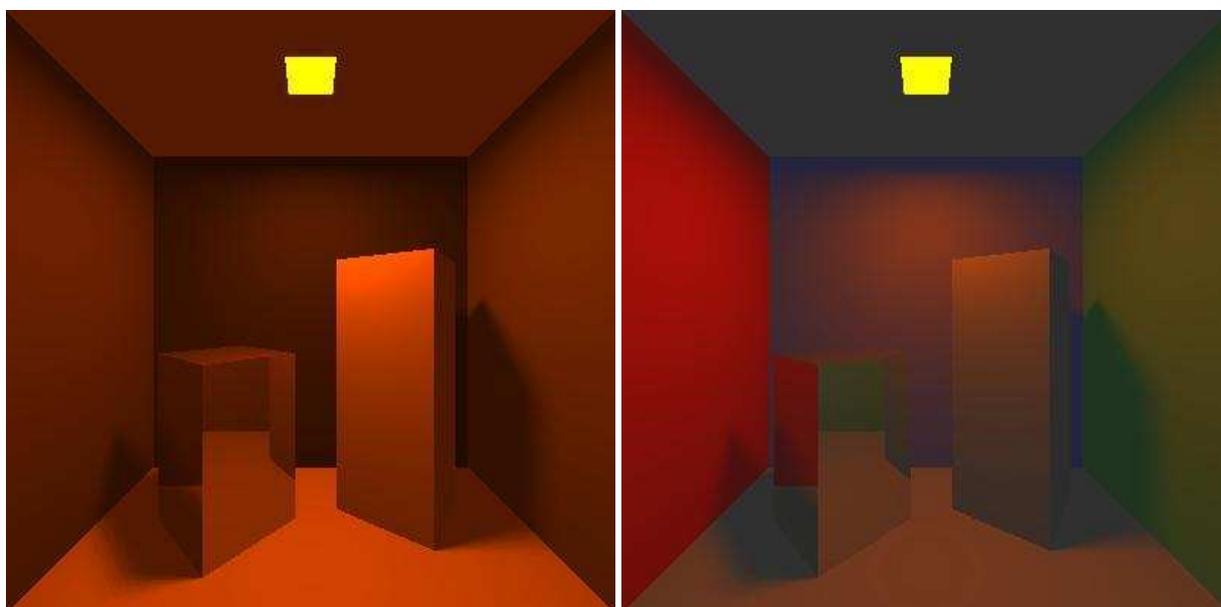


FIG. 7.3 – Simulations spectrale et RGB de l'éclairage d'une cornell box par une lampe à vapeur de sodium (images fournies par Jean Claude Iehl)

tuel est obtenu avec une segmentation adaptative de l'espace XYZ. Pour plus de détails, se reporter à [27] et [28].

Toutefois, de manière analogue aux algorithmes de rendu, nous pensons qu'il serait intéressant de concevoir une méthode d'échantillonnage adaptatif du spectre basée sur un modèle de vision : cette méthode permettrait d'ajuster la décomposition pour obtenir le meilleur compromis entre la qualité du rendu du matériau et la taille des données.

Cette approche pose toutefois un certain nombre de problèmes. Premièrement, utiliser cette forme compressée de la distribution force le processus de rendu à des manipulations non standards, qui alourdissent un peu le traitement. Deuxièmement, cette décomposition adaptative perceptuelle suppose un prétraitement de la scène : en effet, pour parvenir à l'optimisation attendue il faudrait, dans le cas d'un matériau par exemple, être capable de prédire l'aspect de ce matériau dans le contexte où il est placé. Pour cela, il faudrait a priori commencer le rendu, ce qui est impossible vu que la discrétisation du spectre n'est pas achevée. . . Une possibilité pourrait être dans un premier temps d'effectuer une subdivision grossière à l'aide d'un critère statistique simple, de la poursuivre normalement du-

rant le calcul, et de la stopper dès que la différence avec l'étape précédente n'est plus perceptible, une approche somme toute très semblable à celles qui ont été présentées dans ce mémoire.

7.2 Plus de caractéristiques visuelles dans le modèle de vision

Les aspects spectraux ne sont pas les seuls à être "oubliés" dans notre modèle. Il existe de nombreuses autres caractéristiques que nous pourrions exploiter pour accélérer le rendu. Nous allons en détailler quelques unes dans cette section.

7.2.1 La profondeur de champ

La profondeur de champ est l'intervalle de distance dans lequel la vision de la scène est nette. Cela sous-entend la prise en compte du point de focalisation dans l'algorithme de rendu. Par exemple, lorsque l'oeil focalise sur un objet qui est très proche, l'arrière plan qui se trouve derrière cet objet apparaît flou. Il existe des méthodes qui permettent de simuler la profondeur de champs en synthèse d'image. Ceci est intéressant pour notre approche, dans le sens où ce qui est flou n'a pas besoin d'être détaillé outre mesure. La figure 7.4 donne un exemple de cette caractéristique.



FIG. 7.4 – La même photographie avec deux profondeurs de champs différentes

7.2.2 Simulation physique de l'appareil de visualisation

D'une manière plus générale, la recherche de la simulation de l'appareil de visualisation dans son entier fera inmanquablement surgir des limites visuelles exploitables. En effet, le modèle de caméra utilisé en synthèse d'images actuellement est en fait un modèle parfait : il correspond au fameux "trou d'épingle" et est dépourvu de tout défaut physique : il est rempli de vide, la lumière s'y propage en ligne droite, et il n'y a pas de dispersion.

Si nous remplaçons maintenant cet appareil par la simulation d'un objectif photographique, les choses seront tout à fait différentes : il va falloir gérer la focalisation, l'abberation chromatique, voire

même les effets dus aux lentilles non minces. Nous pourrions aussi remplacer le trou d'épingle par la simulation d'un oeil humain et de tout son dispositif optique : cornée, iris, humeur acqueuse ... le tout jusqu'à la rétine.

7.2.3 Les aspects temporels

Nous avons déjà évoqué les aspects temporels de la vision humaine. Les intégrer à une méthode perceptuelle est évidemment intéressant pour le rendu d'animation, mais également pour le rendu d'image fixes : les effets de "motion blur"¹ entre autres, perturbent la visibilité des détails sur les objets d'une scène (voir figure 7.5). Cette caractéristique peut également être exploitée dans un algorithme de rendu, avec entre autre la détermination d'un foyer d'attention dans la scène. Yee a travaillé sur ces aspects, il a présenté les résultats de ses travaux dans [84]



FIG. 7.5 – Exemple de "motion blur"

7.2.4 Une meilleure gestion de l'information couleur

Nous l'avons signalé dans la conclusion de la section 5, la rétine est aujourd'hui un composant assez bien connu du système visuel. Pourquoi alors, dans la continuité de la démarche évoquée en 7.2.2, ne pas simuler l'oeil humain dans son intégralité ? Sans doute parce qu'à l'heure actuelle, une telle simulation est un processus très lourd. Toutefois, les progrès dans ce domaine sont rapides, et la puissance des moyens informatique est en constante augmentation, ce qui devrait rendre la chose possible dans un délai relativement court. Les bénéfices que l'on pourrait en retirer sont immédiats :

¹flou lors de la visualisation d'un objet, généralement du à une vitesse de déplacement élevée

tous les phénomènes psychovisuels liés au dispositif optique seraient gérés d'un seul coup. Cela impose bien évidemment de disposer de données spectrales dès le départ.

Il serait possible de rétorquer à cette idée le fait que les courbes psychovisuelles sont elles aussi bien connues, et permettent de définir la réponse du système visuel sans se préoccuper d'une simulation complexe. Nous répondrons que ce n'est pas vrai pour les caractéristiques couleur : en adoptant l'approche du paragraphe précédent, nous pourrions disposer d'une représentation "physiologique" de la perception de la couleur au niveau de la rétine. Les travaux de Wandell vont en ce sens (voir [76]).

7.3 Vers une nouvelle méthode adaptative progressive

Nous avons vu que notre algorithme de rendu a deux gros défauts. Le premier est que nous comparons deux images intermédiaires successives. Ainsi, si la convergence n'est pas monotone, l'algorithme s'arrêtera beaucoup trop tôt. Ce défaut n'est pas spécifique à notre méthode, mais général à tous les algorithmes progressifs. Le deuxième défaut est que le surcoût du modèle de vision est trop élevé. Nous voudrions nous affranchir de ces inconvénients.

Pour cela, nous avons songé à utiliser notre distance adaptative. Outre le fait qu'elle dispose d'un modèle de vision plus performant que notre modèle simplifié, elle permet aussi une évaluation progressive de la distance. Nous avons vu que l'idéal serait de pouvoir disposer en permanence de la distance entre l'image en cours et l'image finale. Notre idée est donc de mettre sur pied une méthode qui calculerait l'image approximée tout en évaluant progressivement la distance entre cette image et l'image finale. Nous pensons que notre métrique adaptative peut se révéler utile dans ce cas. Nous présentons ici une ébauche d'algorithme.

Notre idée se base sur la métrique adaptative présentée dans la section 5.4, tout en s'inspirant de la méthode de Ramasubramanian *et al*([57]). Nous voulons éviter de calculer la différence entre deux images intermédiaires issues du rendu : nous voulons évaluer simultanément la distance entre image et l'image intermédiaire, c'est à dire que nous allons nous servir des mêmes échantillons pour reconstruire la carte de distance et l'image approximée elle même. Notre méthode envisage d'effectuer un calcul du terme ambiant "par zones".

Il reste bien entendu de grosses incertitudes sur cette méthode : comment évaluer le terme ambiant ? Comment le reporter dans l'image finale ? Dans quelles limites peut on considérer qu'il n'est pas nécessaire de modifier la décomposition pyramidale ? Toutes ces questions restent en suspens et devront être étudiées par la suite.

7.4 Bilan global et réflexions diverses

A l'heure actuelle, le pilotage d'un algorithme de rendu par un modèle de vision est une réalité. Les travaux présentés dans ce mémoire démontrent bien les bénéfices potentiels qu'il est possible d'en retirer, mais de nombreux problèmes de nature purement informatique subsistent. Par exemple, l'évaluation du modèle reste couteuse. Pour limiter le surcoût, la seule issue est d'utiliser un modèle "léger", donc limité dans ces possibilités. Dans la section précédente, nous avons présenté une alternative qui pourrait partiellement résoudre ce problème, mais de nombreuses autres pistes existent.

Le calibrage des dispositifs mis en place, ainsi que la retranscription et l'interprétation des données rendues par le modèle est aussi un problème crucial. Pour une métrique par exemple, l'affichage des résultats est délicat : retranscrire les JNDs linéairement suffit il à obtenir une image de distance significative ? Rien n'est moins sur. Les disparités dans les résultats de nos comparaisons entre les

métriques illustrent parfaitement cet état de fait. Pour ce qui est des algorithmes de rendu guidés par un modèle de vision, l'efficacité de la méthode sera fortement conditionnée par le calibrage du modèle, ainsi que le choix des paramètres qui peuvent, dans le pire des cas, changer d'une scène à l'autre.

Enfin, et d'une manière plus générale, il est intéressant de noter que la majorité des travaux de ce type en synthèse d'images sont des adaptations de méthodes existantes dans les sciences de la couleur et de la vision. Les modèles sont la plupart du temps repris tels quels, pratiquement sans adaptation aucune. Fait encore plus curieux, seuls deux ou trois modèles de vision sont vraiment reconnus dans la communauté synthèse d'image, alors que ces modèles sont peu connus dans la communauté colorimétrie et vision humaine. Dès lors, ces deux communautés ont encore vraisemblablement encore beaucoup à se dire . . .

Bibliographie

- [1] Albert Ahumada and Bettina L. Beard. A simple vision model for inhomogeneous image quality assessment. *SID Digest of Technical Papers*, 29, 1998.
- [2] Albert J. Ahumada. Computational image quality metrics : A review. *Society for Information Display International Symposium Digest of Technical Papers*, 24 :305–308, 1993.
- [3] S. Albin. Distance entre images. Technical report, Ecole Nationale Supérieure des Mines de Saint Etienne, Saint Etienne, France, September 1998.
- [4] Arthur Appel. Some techniques for shading machine renderings of solids. In *AFIPS 1968 Spring Joint Computer Conf.*, volume 32, pages 37–45, 1968.
- [5] A. Artusi and A. Wilkie. A new real time tone mapping model. Technical Report TR-186-2-02-02, Institute of Computer Graphics and Algorithms, Vienna University of Technology, A-1040 Karlsplatz 13/186/2, 2002.
- [6] Mark R. Bolin and Gary W. Meyer. A frequency based ray tracer. In *Computer Graphics (SIGGRAPH 95 Conference Proceedings)*, volume 29, pages 409–418, 1995.
- [7] Mark R. Bolin and Gary W. Meyer. A perceptually based adaptive sampling algorithm. In *Computer Graphics (SIGGRAPH 98 Conference Proceedings)*, volume 32, pages 299–310, July 1998.
- [8] D.H. Brainard. Calibration of a computer controlled color monitor. *Color Research and Application*, 14 :23–34, 1989.
- [9] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4 :532–540, 1983.
- [10] W. B. Cowan. An inexpensive scheme for calibration of a colour monitor in terms of CIE standard coordinates. *Computer Graphics*, 17(3) :315–321, July 1983.
- [11] Scott Daly. The visual differences predictor : an algorithm for the assessment of image fidelity. In A. B. Watson, editor, *Digital Images and Human Vision*, pages 179–206, Cambridge, 1993. MIT Press.
- [12] Jean Marc Deniel. *Modélisation des luminaires et des BRDF : réalisation, mesure et compression*. PhD thesis, IFSIC/IRISA, Université Rennes 1, Rennes, April 2002.
- [13] Patricia Dumont-Bècle. *Simulation de l'aspect des revêtements peinture sur véhicules virtuels*. PhD thesis, Université de Marne La Vallée, Marne La Vallée, October 2002.
- [14] Fredo Durand and Julie Dorsey. Interactive tone mapping. In B. Péroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 219–230, New York, NY, 2000. Springer Wien.

- [15] L. Elie and B. Lapeyre. Introduction aux méthodes de monte carlo. Technical report, Ecole nationale des Ponts et Chaussées, CERMICS, Ecole Nationale des Ponts et Chaussées, C.E.R.M.I.C.S., 6 et 8 avenue Blaise Pascal, Cité Descartes, Champs-sur-Marne, F-77455 MARNE LA VALLEE Cedex 2, September 2001. Disponible sur <http://cermics.enpc.fr/bl/PS/SIMULATION-X/poly-monte-carlo-x.pdf>.
- [16] C. Enroth-Cugell and J.G. Robson. The contrast sensitivity of retinal ganglion cells of the cat. *J. Physiology*, 187 :517–552, 1966.
- [17] Jean-Philippe Farrugia and Bernard Péroche. A perceptual digital image metric. In *Color in Graphics and Image Processing*, Saint Etienne, 2000.
- [18] James A. Ferwerda. Elements of early vision for computer graphics. *IEEE Computer Graphics and Applications*, 21(5) :22–33, September/October 2001.
- [19] James A. Ferwerda, Sumant Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 249–258. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.
- [20] A. Gaddipatti, R. Machiraju, and R. Yagel. Steering image generation with wavelet based perceptual metric. *Computer Graphics Forum*, 16(3) :241–252, August 1997. Proceedings of Eurographics '97. ISSN 1067-7055.
- [21] M.J Gervais, L.O. Harvey, and J.O. Roberts. Identification confusions among letters of the alphabet. *Journal of experimental psychology : Human perception and performance*, 10 :655–666, 1984.
- [22] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. In H. Christiansen, editor, *SIGGRAPH '84 Conference Proceedings (Minneapolis, MN, July 23-27, 1984)*, pages 213–222. ACM, July 1984.
- [23] J. Guild. The colorimetric properties of the spectrum. *Philosophical Transactions of the Royal Society*, 230 :149–187, 1931.
- [24] Baining Guo. Progressive radiance evaluation using directional coherence maps. In *Computer Graphics (SIGGRAPH 98 Conference Proceedings)*, volume 32, pages 255–266. Addison Wesley, July 1998.
- [25] J.M. Hammersley and D.C. Handscomb. *Monte Carlo methods*. Chapman and Hall, 1964.
- [26] Jeanny Héroult. Traitement et codage de la couleur dans le système visuel. In *Ecole d'été - Images Couleur*, volume 1, September 1999.
- [27] J.C. Iehl. *Etudes sur le lancer de rayons spectral*. PhD thesis, École des Mines de Saint Etienne, November 2000.
- [28] Jean Claude Iehl and Bernard Péroche. An adaptive spectral rendering with a perceptual control. *Computer Graphics Forum*, 19(3) :291–299, August 2000.
- [29] Henrik Wann Jensen. Rendering caustics on non-Lambertian surfaces. *Computer Graphics Forum*, 16(1) :57–64, March 1997.
- [30] D.B Judd and G. Wyszecki. *Color in business, science and industry*. Wiley and Sons, 1975.
- [31] J. T. Kajiya. The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.
- [32] Alexander Keller. *Quasi-Monte Carlo Methods for Photorealistic Image Synthesis*. PhD thesis, University of Kaiserslautern, June 1997. ISBN 3-8265-3330-5.

- [33] Eric P. Lafortune and Yves D. Willems. Bi-directional Path Tracing. In H. P. Santo, editor, *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.
- [34] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4) :291–306, October 1997.
- [35] J. Lubin. A visual discrimination model for imaging system design and evaluation. *Vision Models for Target Detection and Recognition*, pages 245–283, 1995.
- [36] David Luebke and Carl Erikson. View-dependent simplification of arbitrary polygonal environments. *Computer Graphics*, 31(Annual Conference Series) :199–208, August 1997.
- [37] David Luebke and Benjamin Hallen. Perceptually-Driven simplification for interactive rendering. In *Rendering Techniques 2001 : 11th Eurographics Workshop on Rendering*, pages 223–234, July 2001.
- [38] M.R. Luo and R.W.G Hunt. The structure of the cie 1997 colour appearance model (ciecam97s). *Color Research and Application*, 23 :138–146, 1998.
- [39] R. Luo. The llab model for colour appearance and colour difference evaluation. In *Color Imaging : Device-Independent Color, Color Hard Copy, and Graphic Arts*, volume 2658, pages 261–269. SPIE, 1996.
- [40] D.L. MacAdam. On the geometry of color space. *Journal of the Franklin Institute*, 238 :195–210, 1944.
- [41] J.L. Maillot. *Pseudo-réalisme et progressivité pour le tracé de rayons*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 1996.
- [42] J.L. Mannos and D.J. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, 20 :525–536, 1974.
- [43] J.C. Maxwell. On the theory of compound colours. *Philosophical Transactions*, 50 :57–84, 1860.
- [44] Ann McNamara. *Comparing Real and Synthetic Scenes Using Human Judgements of Lightness*. PhD thesis, Department of Computer Science, University of Bristol, Bristol, UK, October 2000.
- [45] Ann McNamara. Visual perception in realistic image synthesis. In D. Duke and R. Scopigno, editors, *Computer Graphics Forum*, volume 20(4), pages 211–224. Blackwell Publishing, 2001.
- [46] Gary W. Meyer. Wavelength selection for synthetic image generation. *Computer Graphics and Image Processing*, II :39–44, 1986.
- [47] Kathy T. Mullen. The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *J. Physiol. (Lond)*, 359 :381–400, 1985.
- [48] Karol Myszkowski, Takehiro Tawara, Hiroyuki Akamine, and Hans-Peter Seidel. Perception-guided global illumination solution for animation. In *Computer Graphics (SIGGRAPH 2001 Conference Proceedings)*, volume 35, pages 221–230, August 2001.
- [49] Karol Myszkowski. The visible differences predictor : Applications to global illumination problems. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 233–236, New York, NY, 1998. Springer Wien.
- [50] Karol Myszkowski, Przemyslaw Rokita, and Takehiro Tawara. Perceptually-informed accelerated rendering of high quality walkthrough sequences. In Dani Lischinski and Greg Ward Larson, editors, *Rendering Techniques '99, Eurographics*, pages 5–18. Springer-Verlag Wien New York, 1999.

- [51] László Neumann, Kresimir Matkovic, and Werner Purgathofer. Perception based color image difference. *Computer Graphics Forum*, 17(3) :233–241, September 1998.
- [52] James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 281–288, July 1989.
- [53] S. N. Pattanaik and S. P. Mudur. The potential equation and importance in illumination computations. *Computer Graphics Forum*, 12(2) :131–136, June 1993.
- [54] Sumanta N. Pattanaik, Mark D. Fairchild, James A. Ferwerda, and Donald P. Greenberg. Multiscale model of adaptation, spatial vision and color appearance. In *Proceedings of the Sixth Color Imaging Conference, Society for Imaging Science and Technology*, November 1998.
- [55] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *Computer Graphics (SIGGRAPH 98 Conference Proceedings)*, volume 32, pages 287–298, July 1998.
- [56] Frédéric P. Pighin, Dani Lischinski, and David Salesin. Progressive previewing of ray-traced images using image plane discontinuity meshing. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 115–126, New York City, NY, June 1997. Eurographics, Springer Wien.
- [57] Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *Computer Graphics (SIGGRAPH 99 Conference Proceedings)*, volume 33, pages 73–82, 1999.
- [58] M. Roelens. *Un environnement pour le tracé de rayons utilisant une modélisation par arbre de construction*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, February 1993.
- [59] G. Rougeron. *Problèmes liés à la couleur en synthèse d'images*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 1998.
- [60] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust. Comparing real and synthetic images : some ideas about metrics. In *6th Eurographics Workshop on Rendering, Dublin*, pages 213–222, June 1995.
- [61] A. Scheel, M. Stamminger, J. Putz, and H.-P. Seidel. Enhancements to directional coherence maps. In *Ninth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 2001)*, Plzen, Czech Republic, February 2001. University of West Bohemia. Available from <http://wscg.zcu.cz/wscg2001>.
- [62] A. Scheel, M. Stamminger, and H.-P. Seidel. Tone reproduction for interactive walkthroughs. *Computer Graphics Forum*, 19(3) :301–312, August 2000.
- [63] X. Serpaggi. *Variations sur le calcul des vecteurs d'éclairage indirect*. PhD thesis, École des Mines de Saint Etienne, December 2001.
- [64] X. Serpaggi and B. Peroche. An adaptive method for indirect illumination using light vectors. In *Computer Graphics Forum (Proceedings of Eurographics 2001)*, volume 20, pages C–278–C–287, September 2001.
- [65] Robert Seve. *Physique de la couleur, de l'apparence colorée à la technique colorimétrique*. Masson, Collection Physique Fondamentale, 1996.
- [66] S.K. Shevell and K. Knoblauch. Red/green and yellow/blue equilibrium hues measured in lms space : From linearity to non-monotonicity. *Investigative Ophthalmology and Visual Science*, 39, 1998.

- [67] B.W. Silverman. *Density Estimation for statistics and data analysis*. Chapman & Hall, 1986.
- [68] Maryann Simmons and Carlo H. Séquin. Tapestry : A dynamic mesh-based display representation for interactive rendering. In Bernard Péroche and Holly Rushmeier, editors, *Rendering Techniques 2000 : 11th Eurographics Workshop on Rendering*, pages 329–340, July 2000.
- [69] Yinlong Sun, David Fracchia, Mark S. Drew, and Thomas W. Calvert. A spectrally based framework for realistic image synthesis. *The Visual Computer*, 17(7) :429–444, 2001.
- [70] CIE technical comitee 3.1. An analytic model for describing the influence of lighting parameters upon visual performance. Technical Report CIE 19.21, Commission internationale de l'éclairage, 1981.
- [71] Alain Tremeau. *Contribution des modèles de la perception visuelle à l'analyse d'images couleur*. PhD thesis, IUP Vision, Université Jean Monnet, Saint Etienne, October 1993.
- [72] Jack Tumblin and Holly E. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6) :42–48, November 1993. also appeared as Tech. Report GIT-GVU-91-13, Graphics, Visualization & Usability Center, Coll. of Computing, Georgia Institute of Tech.
- [73] Christian van den Branden Lambrecht. Issues in vision modeling for perceptual video quality assessment. *Signal Processing*, 78(2) :231–252, 1999.
- [74] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 65–76, August 1997.
- [75] F. Viénot. La spécification lms. perspectives pour la colorimétrie et pour la compréhension de la couleur. In *Actes de l'École de printemps Lumière - Matière - Perception à apt*, pages 15–28, March 2000.
- [76] B.A. Wandell. Computational neuroimaging of human visual cortex. *Annual Review of Neuroscience*, 10(22) :145–173, 1999.
- [77] Greg Ward. A contrast-based scalefactor for luminance display. In Paul Heckbert, editor, *Graphics Gems IV*, pages 415–421. Academic Press, Boston, 1994.
- [78] Gregory J. Ward and Francis M. Rubinstein. A new technique for computer simulation of illuminated spaces. *Journal of the Illuminating Engineering Society*, 17(1) :80–91, Winter 1988.
- [79] A.B. Watson. Toward a perceptual video quality metric. *Human Vision and Electronic Imaging III*, (3299) :139–147, 1998.
- [80] Andrew B. Watson. The cortex transform : Rapid computation of simulated neural images. *Computer Vision, Graphics, and Image Processing*, 39(3) :311–327, September 1987.
- [81] T. Whitted. An improved illumination model for shaded display. In *Computer Graphics (Special SIGGRAPH '79 Issue)*, volume 13, pages 1–14, August 1979.
- [82] W.D. Wright. A re-determination of the trichromatic coefficients of the spectral colours. *Transactions of the Optical Society*, 30 :141–164, 1929.
- [83] Julie C. Xia and Amitabh Varshney. Dynamic view-dependent simplification for polygonal models. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 327–334, Los Alamitos, October 27–November 1 1996. IEEE.
- [84] Hector Yee, Sumanta Pattanaik, and Donald P. Greenberg. Spatiotemporal sensitivty and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics*, 20(1), January 2001.

-
- [85] T. Young. On the theory of light and colours. *Philosophical Transactions of the Royal Society of London*, 92 :20–71, 1802.
- [86] J. Zaninetti. *Modélisation vectorielle de l'éclairage global en lancer de rayons*. PhD thesis, École des Mines de Saint Etienne, October 1998.
- [87] Jacques Zaninetti, Xavier Serpaggi, and Bernard Péroche. A vector approach for global illumination in ray tracing. In David Duke, Sabine Coquillart, and Toby Howard, editors, *Computer Graphics Forum*, volume 17(3), pages 149–158, 1998.

Annexe A

L'environnement utilisé et les vecteurs lumineux

Cette annexe a pour but de présenter le logiciel utilisé pour appliquer notre modèle de vision, ainsi que les techniques de calcul d'éclairage global qui y sont utilisées.

A.1 L'environnement utilisé

A.1.1 Description de scènes CSG avec CASTOR

Le logiciel de synthèse d'images utilisé pour nos travaux utilise une modélisation par arbre de construction ("constructive solid geometry", ou CSG). Rappelons brièvement que cette méthode forme des objets complexe à partir de primitives simples combinées entre elles avec des opérations booléennes. Pour décrire notre scène CSG, nous utilisons un langage nommé CASTOR, développé au sein du laboratoire LISSE. La synthèse de CASTOR étant un peu ardue pour les scènes complexes, il existe une extension nommée CASTOR C permettant de modéliser la scène dans un langage un peu plus "naturel". Nous montrons un exemple de fichier CASTOR C sur la figure A.1.

A.1.2 Le logiciel YART

Le logiciel Yart a été écrit par Marc Roelens [58]. Bien qu'étant relativement polyvalent¹, son application principale reste le calcul d'image de synthèse réaliste. Yart est un logiciel conçu de manière évolutive, c'est à dire que les membres successifs du laboratoire ont amené leur contribution personnelle au fil des années, le rendant donc particulièrement riche. Mais à l'heure actuelle, son architecture ne lui permet plus d'exploiter efficacement les machines actuelles : il ne gère pas plusieurs processeurs, n'exploite pas l'accélération matérielle des cartes graphiques et est incapable de lire un autre format de scène que CASTOR. Les textures sont également assez peu pratique à utiliser. Tous ces inconvénients expliquent la relative pauvreté des images de ce manuscrit. Toutefois, ce logiciel est actuellement en refonte complète au laboratoire LISSE.

¹Yart a été utilisé pour des calculs de propagation de pollution de l'air.

Imprimé par Jean-Philippe FARRUGIA

```

08 jan 02 17:14          cornell.c          Page 1/2
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include ".\nac.h"
#include ".\OBJETS/ObjetsCastor.h"

/*****
int model (int argc, char **argv)
{
  T_Matiere CouSol =
  { M_SINGLE_ISO, 1.0, 1.0, 1.0, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouPlafond =
  { M_SINGLE_ISO, 1.0, 1.0, 1.0, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouMurGauche =
  { M_SINGLE_ISO, 1.0, 0.2, 0.2, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouMurFond =
  { M_SINGLE_ISO, 0.6, 0.6, 1.0, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouMurDroite =
  { M_SINGLE_ISO, 0.6, 1.0, 0.6, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouMurArriere =
  { M_SINGLE_ISO, 1.0, 1.0, 1.0, 1,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouCube1 =
  { M_SINGLE_ISO, 1.00, 1.00, 1.00,
    0.0, 0.0, 0.0, 0.0 };
  T_Matiere CouCube2 =
  { M_SINGLE_ISO, 0.01, 138./255., 161./255.,
    0.0, 0.0, 0.0, 1.00 };
  T_Matiere Cou_abajour =
  { M_SINGLE_ISO, 0.95, 0.95, 1.0,
    0.0, 0.0, 0.0, 1.0 };
  T_Matiere Cou_VeryNoir =
  { M_SINGLE_ISO, 0.01, 0.01, 0.01,
    0.0, 0.0, 0.0, 1.0 };
  T_Matiere C_verre =
  { M_DEVIANT, 1.0, 1.0, 0.1, 1.3,
    0.0, 0.0, 0.0, 0.0 };

  char ch[500];
  int TrouSource, source, source2, Ma_Scene, Piece, Cube1, Cube2, abajour, temp,
  troucube, tourcube;
  double x_piece, y_piece, z_piece;
  double x_source, y_source, taille_source;

  #if 0
  Init_Rendu_Indirect(
    "cornell_fichier.ve", /* Nom du fichier de stockage des VEI ou "" */
    30, /* taille du germe de VEI = N*Init-2 */
    8, 16, /* Nombre de VEI pour interpoler */
    32, 32, /* Nombre de decoupage de l'hemisphere */
    3, 0.005, /* profondeur et contribution minimale pour les reflexions */
    1, 0.1, /* profondeur et contribution minimale pour Monte Carlo */
    0.8, /* variation max d'eclaircissement due au gradient */
    0.8, /* cos de l'angle maxi acceptable entre 2 VEI */
    0.8, /* variance maxi acceptable entre les VEI valides */
    1, 0, 0, /* IL FAUT CHANGER IL FAUT S'ACROCHER, CALCULER TOUT */
    0.0, 0.0, 0.0 /* ambiant arbitraire pour fin de recursivite (beurk) */
  );
  #else
  Init_Rendu_Schlick(1, 0.2, 3, 0.001);
  #endif

  eye( 2.9, 1.49, 1.69 );
  aim( 0.0, 1.39, 1.4 );
  ang(45,45);
  depth(0.00001, 10000.0);
  roll(0.0);

  /*****
  x_piece=3; y_piece=3; z_piece=3;
  x_source=0.9;
  taille_source=0.4;
  y_source= (y_piece-taille_source)/2.0;

  Piece = PieceFermee( x_piece, y_piece, z_piece, 0.2,
    &CouSol, &CouPlafond,
    &CouMurGauche, &CouMurFond, &CouMurDroite, &CouMurArriere
  );

  abajour = aff(0.02, 0.02, 0.3, Cylindre(&Cou_abajour));
  temp = aff(0.15, 0.15, 0.1,
    differ(
      Sphere(&Cou_abajour),
      trans(0., 0., -0.01, Sphere(&Cou_abajour)),
      fin
    )
  );
  abajour = reunion(abajour, trans(0., 0., -0.099, temp), fin);
  abajour = trans(0., 0., -0.3, abajour);

  #if 1
  source = SourceSpherique(
    1.5, 1.5, 2.65, /* position */
    0.05, /* rayon */
    1, /* nombre de rayons d'ombre */
    10.0, 13.0, 14.0 /* couleur */
  );
  #else
  source = SourceSurfaciqueVisible(
    1.25, 1.75, 2.99, /* Origine */
    0.5, 0, /* Vecteur de base 1 */
    0, -0.5, 0, /* Vecteur de base 2 */
    0.01, /* epaisseur visible */
    1*0.8414, 1*1.0894, 1*1.0334, /* luminance RVB */
    0.02, /* Angle solide maxi */
    2, /* Recursion maxi */
    0,
    &CouSol /* couleur des faces non lumineuses */
  );
  #endif

  /*****
  Cube1 =
  trans( 0.7, 0.4, 0.0,
  rotat( 0,0,1, 45,
  aff( 0.7, 0.7, 1.2,
  Cube( &CouCube1 ) ));

  Cube2 =
  Cube( &CouCube2 );
  
```

lundi 14 octobre 2002

```

08 jan 02 17:14          cornell.c          Page 2/2
/* troucube =
trans(0.5, 0.5, 0.5,
rotat(1,0,0, 50,
trans(0., 0., -1.,
affin(0.2, 0.2, 2.,
Cylindre(&CouCube2)))));
tourcube =
trans(0.5, 0.5, 0.5,
rotat(1,0,0, 50,
trans(0., 0., -1.,
affin(0.2, 0.2, 2.,
Cylindre(&C_verre)))));
Cube2 = differ(Cube2, troucube, fin);
Cube2 = reunion(Cube2, tourcube, fin);
Cube2 = inter(Cube2, trans(0., -1., 0., homot(2., Cube(&CouCube2))), fin);

/*
Cube2 =
trans( 0.6, 1.8, 0.01,
rotat( 0,0,1, 55,
affin( 0.7, 0.7, 1.9,
Cube2));

/** Cube1 =
trans( 0.8, 1.2, 0.8,
homot( 0.7,
Sphere( &C_verre ) ); **/

Ma_Scene = reunion( Piece, Cube1, Cube2, /*trans(1.5, 1.5, 3, abajour),*/
fin);

sprintf( ch, "(taille_octree %g %g %g %g)",
xmin(Ma_Scene), xmax(Ma_Scene),
ymin(Ma_Scene), ymax(Ma_Scene),
zmin(Ma_Scene), zmax(Ma_Scene) );

telquel(ch);
pas_de_lumiere_par_defaut();
visu(Ma_Scene);

return 0;
}
  
```

1/1

FIG. A.1 – Exemple de programme CASTOR C. Celui-ci décrit une cornell box.

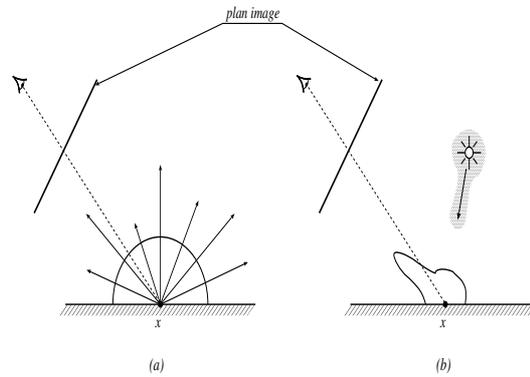


FIG. A.2 – Calcul des VEI

A.2 Les vecteurs lumineux

A.2.1 Principe

Dans ce manuscrit, il a été clairement mis en évidence que le calcul de l'éclairement global était un processus long. Nous avons aussi montré que les méthodes de type Monte Carlo paraissent incontournables pour obtenir une simulation correcte. Pourtant, si nous devons effectuer ce calcul pour tous les points de la scène, le temps d'obtention de l'image serait bien trop important. Nous voudrions calculer cette quantité uniquement en certains points, et d'interpoler le reste en se servant de ces échantillons. Les vecteurs d'éclairement indirect (VEI) ont été conçus par Jacques Zaninetti ([86],[87]) dans cette optique. Ils ont été mis en oeuvre au sein du logiciel YART.

Le calcul des VEI s'effectue donc en deux passes :

- La première crée un certain nombre de VEI sur l'image pour initialiser l'interpolation.
- La deuxième phase est la phase de rendu. Le processus essaie d'interpoler les VEI pour calculer l'éclairement indirect. Si ce n'est pas possible, alors un nouveau VEI est créé à cet endroit. Ce VEI pourra être utilisé à son tour pour une interpolation.

Nous allons maintenant détailler ces différentes phases. Les explications et figures suivantes sont extraites de la thèse de Xavier Serpaggi([63]).

A.2.2 Le calcul des VEI

Le principe est d'échantillonner l'éclairement indirect incident. Pour ce faire, nous plaçons, en un point visible de la scène déterminé par un rayon primaire, un hémisphère partitionné en angle solide constant. Pour chaque cellule ainsi obtenue, nous lançons un rayon. Ces derniers permettent alors d'obtenir une moyenne de l'éclairement indirect parvenant en ce point. Pour ne pas conserver des données trop volumineuses, cette approximation est remplacée par une source de lumière virtuelle, dotée d'une direction D et d'un éclairement P . La figure A.2 représente cette méthode de calcul des VEI.

Détaillons un peu : nous voulons résoudre l'équation de rendu [31]. Pour rappel, cette équation est la suivante :

$$L_r(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} \rho(x, \vec{\omega}_i \rightarrow \vec{\omega}_r) L_i(x, \vec{\omega}_i) \cos \theta_i d\omega_i \quad (\text{A.1})$$

Comme indiqué dans la section 6.1.3, nous pouvons décomposer cette équation en termes plus parlants :

$$L_r(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + L_{spec}(x, \vec{\omega}_r) + L_{dir}(x, \vec{\omega}_r) + L_{caust}(x, \vec{\omega}_r) + L_{ind}(x, \vec{\omega}_r) \quad (A.2)$$

ou $L_e(x, \vec{\omega}_r)$ est la composante directe de l'éclairement, $L_{spec}(x, \vec{\omega}_r)$ correspond au pic spéculaire et $L_{ind}(x, \vec{\omega}_r)$ à la composante indirecte, celle qui nous intéresse.

Une approximation discrete de la partie correspondant à l'éclairement indirect peut s'écrire :

$$L_{ind}(x, \vec{\omega}_r) \approx \frac{2\pi}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \rho_d(x, \vec{\omega}_{j,k} \rightarrow \vec{\omega}_r) L_i(x, \theta_j, \phi_k) \cos\theta_j \quad (A.3)$$

où M et N correspondent au nombre de découpages, à angle solide constant, d'un hémisphère centré en x , selon les angles azimutal et zénithal respectivement. Dans l'équation précédente, ρ_d représente la partie non purement spéculaire de la BRDF. Chaque rayon lancé dans la cellule donne donc des informations sur l'énergie incidente en x selon la direction (θ_i, ϕ_k) ayant subi au moins une réflexion sur une surface diffuse. Le nombre maximal de réflexions nécessaires n'est pas limité. Lorsqu'une surface diffuse est rencontrée, il est tout à fait possible de poursuivre le processus, qui ne prendra fin que lorsqu'une source lumineuse sera rencontrée, ou que l'énergie récoltée devient trop faible suite aux absorptions successives. Pour éviter de pousser la récursion aussi loin, l'utilisateur fixe deux grandeurs : une profondeur maximale de récursion au delà de laquelle l'éclairement indirect est remplacé par un terme ambiant constant, et une quantité d'énergie minimale à récolter par rayon, en dessous de laquelle le calcul est là encore remplacé par un terme ambiant.

Les informations énergétique récoltées pour chaque cellule permettent de déduire une luminance moyenne P_0 de l'éclairement émis par le point x . Avec la BRDF locale au point x , il sera alors possible de calculer l'éclairement P et la direction \vec{D} d'une source de lumière virtuelle. Les deux grandeurs P et \vec{D} forment le VEI :

$$L_r(x, \vec{\omega}_r) = P f_r(x, \vec{D} \rightarrow \vec{\omega}_r) \quad (A.4)$$

Ainsi, la source de composantes P et \vec{D} donne bien le même éclairement que celui obtenu par un échantillonnage exhaustif.

A.2.3 L'interpolation des VEI

Nous voulons maintenant calculer l'éclairement indirect à l'aide des VEI. En chaque point visible de la scène (ceci se passe donc dans l'espace objet) une recherche est faite dans un voisinage proche dont la taille est fixée arbitrairement. Les éventuels VEI trouvés dans cette zone sont triés en fonction de différents critères (voir figure A.3 :

- La position relative du point x et du VEI en considérant les objets sur lesquels ils se trouvent. Le VEI ne doit pas être trop *au-dessus* ou *au-dessous* du point x .
- La direction d'observation pour laquelle le VEI a été calculé et la direction d'observation permettant de voir le point x . Ces deux directions doivent être *proches*.
- L'orientation relative des surfaces des objets en x et au point où le VEI a été calculé. Il ne doit pas y avoir un angle trop important entre ces deux surfaces.

A ces critères purement géométriques, se greffent d'autres moyens de sélection basés sur des critères statistiques. La variance de l'éclairement représenté par chaque VEI est prise en compte, ainsi que le gradient d'éclairement dans le voisinage du point considéré. Ces deux dernières notions permettent de détecter des singularités dans la fonction même d'éclairement indirect. Une variance

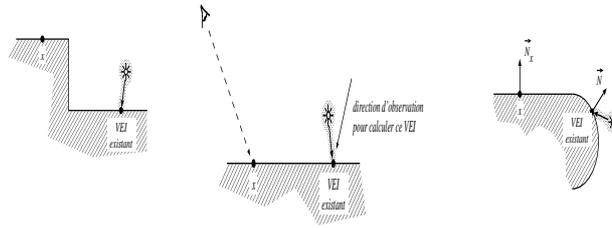


FIG. A.3 – Les trois cas ne permettant pas l'utilisation des VEI du voisinage : mauvaise position relative, direction d'observation trop différentes, mauvaise orientation

inter-VEI élevée met en évidence une zone de la scène où l'éclairage est fortement perturbé alors qu'un fort gradient permet de se rendre compte que la façon dont l'éclairage varie est modifiée. Dans les deux cas, il semble nécessaire de raffiner localement la représentation que nous avons de cette fonction d'éclairage indirect en calculant un nouveau VEI. Ces critères, associés aux précédents, permettent de déterminer la validité d'un VEI pour l'interpolation. Les VEI non valides ne seront bien évidemment pas pris en compte pour le calcul.

Une fois ce tri effectué, deux cas peuvent se présenter. Dans le premier, le plus favorable, le nombre de VEI satisfaisant aux critères est suffisant. La valeur du terme $L_{ind}(x, \vec{\omega}_r)$ est donc obtenue par interpolation des valeurs P et \vec{D} de ces derniers. Pour parfaire l'interpolation, chaque VEI est pondéré en fonction de la distance le séparant du point x : plus la distance est faible plus grande sera l'influence du VEI sur le résultat final. Dans le second cas, le nombre de VEI restants est insuffisant (en pratique, il est inférieur à un seuil fixé par l'utilisateur) et il est donc impossible de déterminer une valeur de $L_{ind}(x, \vec{\omega}_r)$ par interpolation. Il est donc nécessaire de procéder à un calcul complet de l'éclairage indirect en x . Celui-ci est fait exactement comme décrit à la section précédente et le nouveau VEI, évalué exactement en ce point, est à son tour stocké pour être réutiliser le cas échéant.

A.2.4 Les améliorations de Xavier Serpaggi

Xavier Serpaggi a, dans son travail de thèse, apporté un certain nombre d'améliorations au niveau du calcul des VEI et de l'utilisation des VEI. Nous allons brièvement présenter ces améliorations, qui seront utilisés pour nos travaux futurs. Nous invitons les lecteurs désirant plus de détails à se reporter à [63] et [64].

Un meilleur calcul des VEI

Nous avons vu qu'afin de représenter l'éclairage en un point, les VEI sont calculés avec une méthode de type Monte Carlo, dont le support des échantillons est un hémisphère centré sur le point considéré. Or, ce calcul est totalement indépendant de l'endroit où il est effectué dans la scène. Il serait sans doute plus logique d'opter pour une méthode adaptative : l'échantillonnage devrait être fin quand la situation est complexe, et plutôt grossier dans le cas contraire. Pour ce faire, deux approches ont été proposées par Xavier Serpaggi.

La première privilégie le parcours en profondeur, en augmentant la récursion dans le calcul du VEI. Mais en relançant une multitude de rayons, cette technique est très coûteuse, et devient vite impraticable. Afin d'obtenir une meilleure efficacité, Xavier Serpaggi propose de récolter l'information le long d'un seul chemin, à la manière du suivi de chemins de Kajiya.

La seconde prend un autre point de vue : quelquesoit le point de la scène considéré, la subdivision de l'hémisphère est toujours régulière. En adoptant un parcours de l'arborescence des rayons en

largeur, nous pouvons augmenter ou diminuer le nombre de rayons par hémisphère, et tenter ainsi de l'adapter au contexte local. Cela peut être effectué via un test statistique séquentiel de vraisemblance. Ce test permet de maîtriser l'erreur effectuée sur le choix entre deux hypothèses formulées sur l'homogénéité en termes d'éclairement d'une cellule de l'hémisphère. La subdivision ne sera donc pas uniforme et va dépendre de la répartition de la luminance qui lui est incidente.

Une meilleure interpolation

L'interpolation est une phase extrêmement importante puisque c'est à ce moment-là que l'on va reconstruire l'éclairement indirect qui sera visualisé sur l'image. Un des problèmes soulevés par J. Zaninetti est le caractère empirique des paramètres utilisés pour effectuer cette interpolation. Plusieurs sont nécessaires, ils ne sont pas faciles à interpréter et leur réglage demande beaucoup de temps et d'expérience. De plus, ils sont figés pour la totalité de l'image. Xavier Serpaggi a mis au point une méthode permettant de tirer parti des VEI obtenus pendant le calcul du germe pour se faire une idée de la fonction à reconstruire et pour pouvoir y adapter les paramètres de l'interpolation.