Etudes de méthodes et outils pour la cohérence visuelle en réalité mixte appliquée au patrimoine

Emmanuel Durand

2013-ENAM-0043

Ecole doctorale n° 432 : Science des Métiers de l'Ingénieur

Doctorat ParisTech
THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers
Spécialité : Informatique - Traitement du signal

présentée et soutenue publiquement par

Emmanuel DURAND

le *19 Novembre 2013*

Etudes de méthodes et outils pour la cohérence visuelle en réalité mixte appliquée au patrimoine

Directeur de la thèse : **Frédéric MÉRIENNE**
Co-encadrement de la thèse : **Christian PÈRE, Patrick CALLET et Thomas MULLER**

Jury

M. Daniel MENEVEAUX, Pr., Université de Poitiers	Rapporteur
M. Malik MALLEM, Pr., Université d'Evry	Rapporteur
M. Jean-Pierre JESSEL, Pr., IRIT	Président
M. Frédéric MÉRIENNE, Pr., Le2i, Arts et Métiers ParisTech	Examinateur
M. Christian PÈRE, Mcf., Le2i, Arts et Métiers ParisTech	Examinateur
M. Patrick CALLET, Pr., Ecole Centrale Paris et Mines ParisTech	Examinateur
M. Julien ROGER, Reponsable industriel, on-situ	Examinateur
M. Thomas MULLER, Responsable industriel	Examinateur

THÈSE

Arts et Métiers ParisTech - Centre de Cluny
Le2i – Institut Chalon sur Saône

Remerciements

Je tiens à remercier l'équipe de la société on-situ, en particulier Julien Roger et Jean-Michel Sanchez pour m'avoir accueilli et permis de mener à bien mon travail tout en m'offrant la possibilité de participer à de nombreux projets annexes au sein de l'équipe. Mes remerciements vont également vers Christian Père et Frédéric Mérienne, pour leurs conseils et leur confiance, ainsi qu'à Patrick Callet pour son enthousiasme à partager sa passion. Ma profonde gratitude est pour Thomas Muller qui m'a suivi sans faillir même après son départ de on-situ, et qui a été une source fructueuse de remarques et de critiques quant à mon travail.

Je remercie aussi les membres du jury Daniel Meneveaux et Malik Mallem pour avoir accepté de rapporter ce travail, ainsi que Jean-Pierre Jessel pour avoir accepté de participer au jury de soutenance.

Egalement remerciés, toute l'équipe de on-situ ainsi que celle de l'Institut Chalon sur Saône pour la bonne humeur et l'ambiance chaleureuse que toutes ces personnes véhiculent, rendant le travail dans cet environnement agréable.

Enfin, je voudrais exprimer toute ma gratitude à ma famille pour leur soutien inconditionnel et leur confiance, de même qu'à ma compagne pour ces mêmes raisons ainsi que pour sa patience et sa compréhension.

Table des matières

Chapitre 1

Introduction et contexte

1.1 Réalité mixte : définition et utilisations

La réalité mixte est une extension de la réalité virtuelle, consistant à mélanger des éléments issus de celle-ci avec des éléments réels tirés de l'environnement de l'utilisateur, le tout en temps réel ou éventuellement en temps contraint pourvu qu'il ait le sentiment que cette composition forme une scène finale virtuellement cohérente où les éléments virtuels et réels semblent interagir.

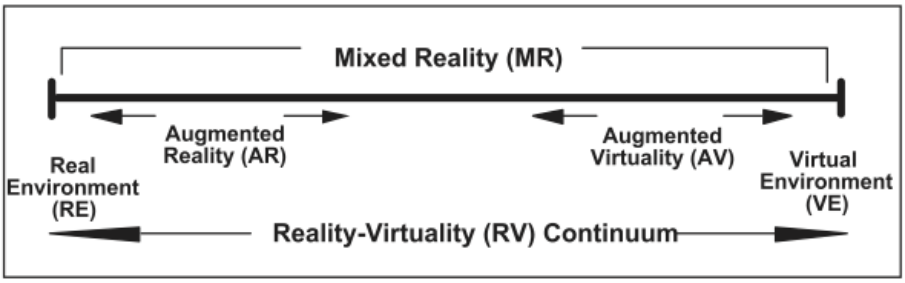


Figure 1.1 – Continuum Réel - Virtuel

Les applications de la réalité mixte peuvent différer de par le rapport entre la quantité d'informations provenant de l'environnement réel et de l'environnement virtuel. Cela est représenté par Milgram [72] par l'échelle du continuum réel-virtuel (figure 1.1) : lorsque le réel est prépondérant, on parle plutôt de réalité augmentée ; et lorsque le virtuel est prépondérant, on parle de virtualité augmentée.
La réalité mixte ne se limite pas au domaine de la vision mais couvre l'ensemble de nos sens. Nous nous limiterons cependant dans la suite de ce travail à l'aspect visuel. Aujourd'hui, la réalité mixte est le plus couramment implémentée sous la forme de la réalité augmentée, dans des domaines très divers :

- Divertissement : jeux vidéo, parcs d'attraction [4, 6, 8]
- Aide à la formation [6]
- Muséographie : diffusion d'informations contextuelles, intégration d'édifices disparus dans une vue réelle [5, 111, 110, 7, 25]
- Aide au pilotage / à la conduite : *head up display* des pilotes aériens, à l'étude pour une intégration dans les automobiles [3]
- Aide à la maintenance : diffusion de plans en surimpression du matériel réel [70].

Pour s'assurer de la bonne corrélation entre les éléments virtuels et réels (par exemple le positionnement correct d'un élément par rapport à un autre), la réalité mixte impose de connaître certaines caractéristiques de l'environnement réel, selon l'application. Il peut s'agir de l'orientation et de la position de l'utilisateur dans cet environnement, ou d'une connaissance plus complète de la géométrie des objets réels, toujours dans le but de pouvoir placer les éléments virtuels de manière cohérente par rapport aux éléments réels. On peut décomposer la chaîne de réalité mixte de la manière suivante (voir figure 1.2) :

- Captation de la scène réelle : par l'intermédiaire de caméras par exemple
- Analyse des données capturées : c'est à dire l'extraction d'informations, comme la détection d'objets ou de personnes, leur suivi, etc.
- Détermination des paramètres de la simulation (comportement)
- Simulation de la scène
- Application de la simulation.

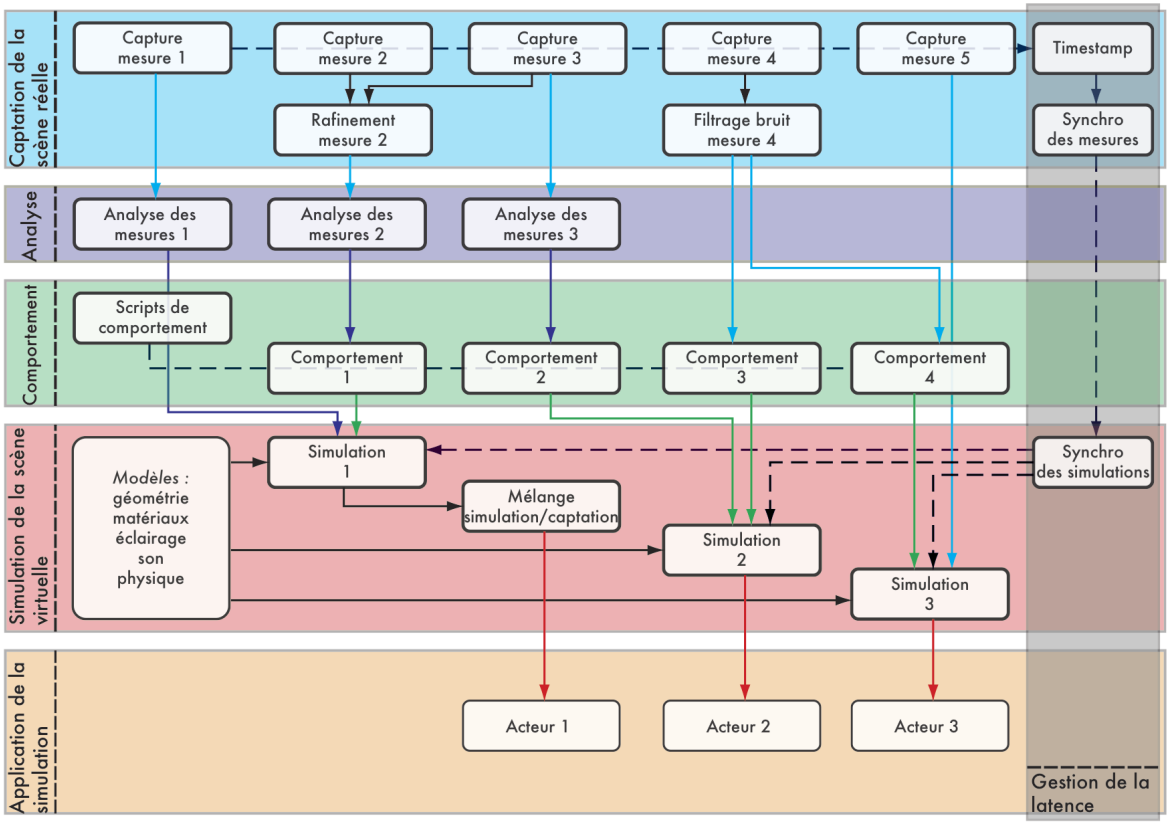


Figure 1.2 – Chaîne de RM générique

1.2 ray-on, un dispositif de réalité mixte

Présentation générale

Ce travail de thèse s'est déroulé dans le cadre du développement d'un dispositif de réalité mixte, ray-on, imaginé spécifiquement pour la valorisation du patrimoine existant ou disparu. Pour répondre aux besoins de la société on-situ, celui-ci devait en plus de pouvoir être utilisé en extérieur à toute époque de l'année, offrir la possibilité d'orienter son point de vue de

manière à avoir une vue d'ensemble du site. Devant l'absence de solution clé en main, ray-on a été développé et fabriqué sous la direction de on-situ.

La nécessité d'un tel dispositif s'est posée pour permettre la diffusion à l'échelle 1/1 d'une maquette numérique, créée par on-situ, réprésentant l'environnement du dispositif à une époque différente. La première et principale application de ce système, sur le site de l'abbaye de Cluny en Bourgogne, est décrite dans la section suivante.

Figure 1.3 – ray-on / on-situ

Le dispositif ray-on se présente sous la forme d'un écran haute luminosité fixé sur un pied par une liaison permettant deux rotations, ce qui offre finalement un champ de vision de 360° sur l'axe vertical, et 120° sur l'axe horizontal (figure 1.3). La grande surface de l'écran d'une diagonale de 32" couvre une part importante du champ de vision de l'utilisateur qui peut se concentrer sur le contenu plutôt que sur la manipulation du dispositif, ce qui aurait été le cas avec un écran plus réduit tel que ce que peut proposer une tablette.

Le point de vue de la scène diffusée par l'écran reproduit le champ de vision moyen d'un utilisateur manipulant le dispositif. De cette manière, il est possible d'obtenir une cohérence entre les éléments affichés et les éléments réels environnants : ils sont dans le prolongement les uns des autres, les éléments virtuels semblant du point de vue de l'utilisateur être à la même échelle que les éléments réels. On peut parler ici de cohérence géométrique.

Pour renforcer ce sentiment de continuité, la maquette virtuelle dispose de matériaux modélisés à partir d'éléments réels, les vestiges de l'édifice dans le cas de l'église abbatiale de Cluny. De plus la maquette est modélisée avec précision, certain éléments tels que les chapiteaux ayant été scannés avec une précision pouvant atteindre le millimètre. L'aspect visuel des objets virtuels est semblable à leur pendant réel : on peut parler de cohérence d'aspect.

La réalité mixte est implémentée actuellement de deux manières dans ray-on. D'une part, il est possible d'intégrer, dans un flux vidéo, la maquette numérique d'un édifice à son emplacement juste comme c'est le cas pour un des dispositifs installés à Cluny. D'autre part, et cela a été l'un des sujets de ce travail de thèse, l'environnement lumineux réel est reproduit sur cette maquette virtuelle. Il s'agit ici de virtualité augmentée puisqu'un élément de l'environnement réel est intégré, de manière cohérente, dans la scène virtuelle : on peut parler alors de cohérence lumineuse.

Figure 1.4 – De gauche à droite : cohérence géométrique / d'aspect / lumineuse

Pour reprendre la chaîne de réalité mixte présentée précédemment, celle implémentée pour ray-on est décrite par la figure 1.5. A noter que, actuellement, la luminosité de l'écran n'est pas contrôlée de manière logicielle et ce contrôle n'est destiné qu'à permettre la lisibilité de l'écran en plein soleil.

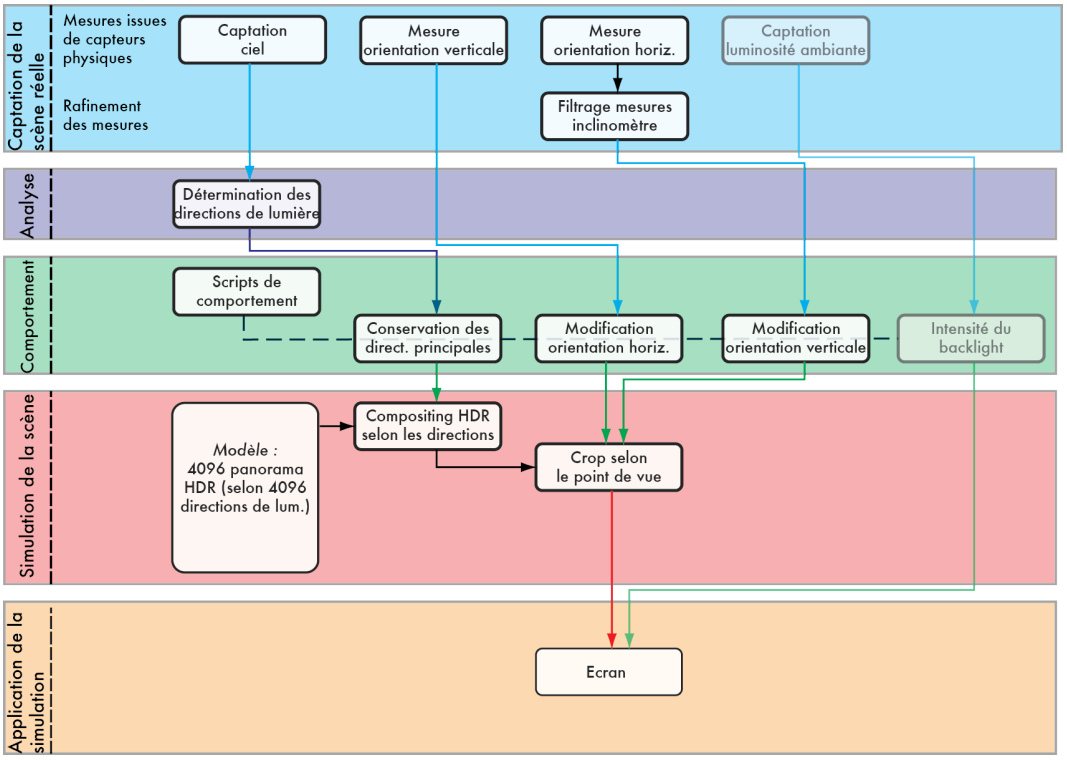


Figure 1.5 – Chaîne de RM de ray-on actuelle

Architecture

L'incarnation la plus simple du système ray-on complet, capable de présenter à l'utilisateur une vue alternative de son environnement dont l'éclairage est une reproduction de l'éclairage

réel, est composée de trois éléments distincts, aussi bien par leur fonctionnalité que par le localisation puisqu'ils peuvent être éloignés de plusieurs dizaines de mètres :

- Le dispositif à proprement parler, tel que perçu par les utilisateurs,
- la régie où se trouve l'organe de calcul du panorama,
- la caméra d'environnement chargée de capturer l'éclairage réel.

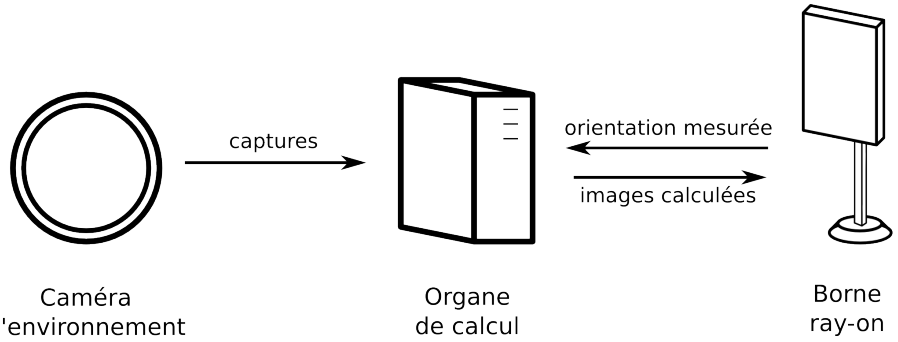


Figure 1.6 – Schéma de l'architecture du système ray-on

La complexité de la scène virtuelle, qui est composée de plusieurs millions de triangles et de nombreuses textures très détaillées, ainsi que la précision de la reproduction de l'éclairage désirant et impliquant donc plusieurs centaines de sources lumineuses, a imposé le déport de l'organe de calcul pour ne pas être limité par la puissance à notre disposition.

Caméra d'environnement

La caméra est de type industriel, dotée d'un capteur d'une résolution peu importante en regard des standards actuels (environ de 640x480) mais qui a l'avantage d'avoir une dynamique importante et d'avoir un taux de rafraîchissement élevé, ce qui se révèle intéressant dans le cas où l'on désire capturer rapidement des images à haute dynamique (plus de détails à ce sujet sont donnés dans la suite de ce document). De plus, la précision offerte par une résolution plus importante ne serait d'aucune utilité puisque les sources lumineuses que l'on cherche à reproduire occupent plus d'un pixel sur la capture.

D'autre part, la caméra est équipée d'un objectif de type *fisheye*, permettant la capture de l'environnement lumineux sur un hémisphère complet. Deux raisons à ce choix : la première est purement technique, puisque les méthodes de captation au delà d'un hémisphère sont très onéreuses et n'offrent a priori pas la compacité ni la fiabilité du couple caméra industrielle / objectif *fisheye* choisi. La seconde est que la part la plus importante de la luminance provient de la partie supérieure d'une capture de l'éclairage prise en extérieure, c'est à dire du ciel. Dans un contexte tel que le nôtre, on peut donc omettre la partie inférieure.

Régie

La régie comporte comme dit plus haut l'organe de calcul du panorama, sous la forme d'un ou plusieurs ordinateurs qui ont pour tâches de créer une image haute dynamique à partir des captures basse dynamique issues de la caméra, d'en extraire une représentation compressée de l'éclairage réel, et d'utiliser cette représentation pour reproduire ce dernier dans la scène virtuelle. Enfin, le panorama ainsi calculé est envoyé au dispositif selon les informations d'orientation que celui-ci mesure à chaque instant.

En outre, et sans entrer dans les détails, la régie comporte les organes de contrôle et de régulation du dispositif. Celui-ci est en effet susceptible d'être installé à l'extérieur et doit donc

supporter des différentiels de température très importants au cours de l'année. Les organes de contrôle sont donc chargés de maintenir une température acceptable pour le matériel, et de prévenir les pannes dûes à des conditions anormales.

Dispositif ray-on

Le dispositif manipulé par les utilisateurs est composé de deux éléments principaux : l'écran, qui diffuse le contenu envoyé par la régie ; et l'ensemble des capteurs d'orientation, qui vont donner à l'organe de calcul situé en régie les informations nécessaires pour reproduire le point de vue de l'utilisateur dans la scène réelle.

L'orientation dans le plan horizontal est mesurée par un codeur absolu, précis et au rafraîchissement élevé. Dans le plan vertical, l'orientation est mesurée par un accéléromètre. Bien que placé aussi près du centre de rotation que mécaniquement possible, la mesure issue de ce capteur n'est pas fiable au cours d'un mouvement pour déterminer l'orientation, aussi est-elle filtrée ce qui réduit sensiblement la réactivité du système sans pour autant être préjudiciable à l'expérience de l'utilisateur.

1.3 Le site pilote de l'abbaye de Cluny

Le site de l'abbaye de Cluny a été tout particulièrement adapté en tant que site pilote pour le développement de ray-on. L'église abbatiale de Cluny est en effet difficile à appréhender pour les visiteurs : utilisée comme carrière de pierre après la révolution française, il reste aujourd'hui moins de 10% de ce qui fut plusieurs siècles durant la plus grande église de la chrétienté. Cela limite la compréhension du lieu aux visiteurs, et encourage la mise en place de techniques adaptées pour valoriser le lieu.

Jusqu'à présent, quatre dispositifs ray-on ont été installés sur le site, tous au cours de cette thèse. Ils présentent autant de points de vue différents sur l'édifice, deux étant en intérieur (ou le seraient si l'édifice était encore complet) et deux proposant un point de vue extérieur, disposés comme présenté sur la figure 1.7.

Les deux dispositifs en intérieur visibles sur les figures 1.3 et 1.4, situés au niveau du grand et du petit transept, rendent compte du volume intérieur de l'église, en diffusant à l'échelle 1/1 la maquette numérique. Le contenu présenté est totalement virtuel, les parties existantes ayant été reproduites par photomodélisation. Le reste a été modélisé à partir des travaux d'archéologues et d'historiens, s'appuyant par exemple sur des édifices issus du courant de pensée clunisien pour imaginer ce à quoi devait ressembler la grande église de Cluny.

Les dispositifs en extérieur sont sensiblement différents dans leur scénario. Celui du parc (figure 1.8) fournit un point de vue distant sur l'église, permettant de l'appréhender dans son ensemble par rapport au cloître et aux deux clochers subsistants. Cependant, lorsque l'utilisateur le manipule, la vue change et la focale augmente pour fournir une vue rapprochée, alors que les bâtiments du cloître disparaissent : l'utilisateur peut ainsi apprécier les détails de l'extérieur de l'église.

Enfin, le dispositif de la tour des fromages de Cluny propose une implémentation plus "standard" de la réalité augmentée. Installé en hauteur (figure 1.9), elle diffuse une vue de l'église intégrée dans la ville de Cluny actuelle. Le travail sur la reproduction de la lumière, présenté dans ce document, permet à la reproduction virtuelle de s'intégrer à la source vidéo (figure 1.10), en temps réel.

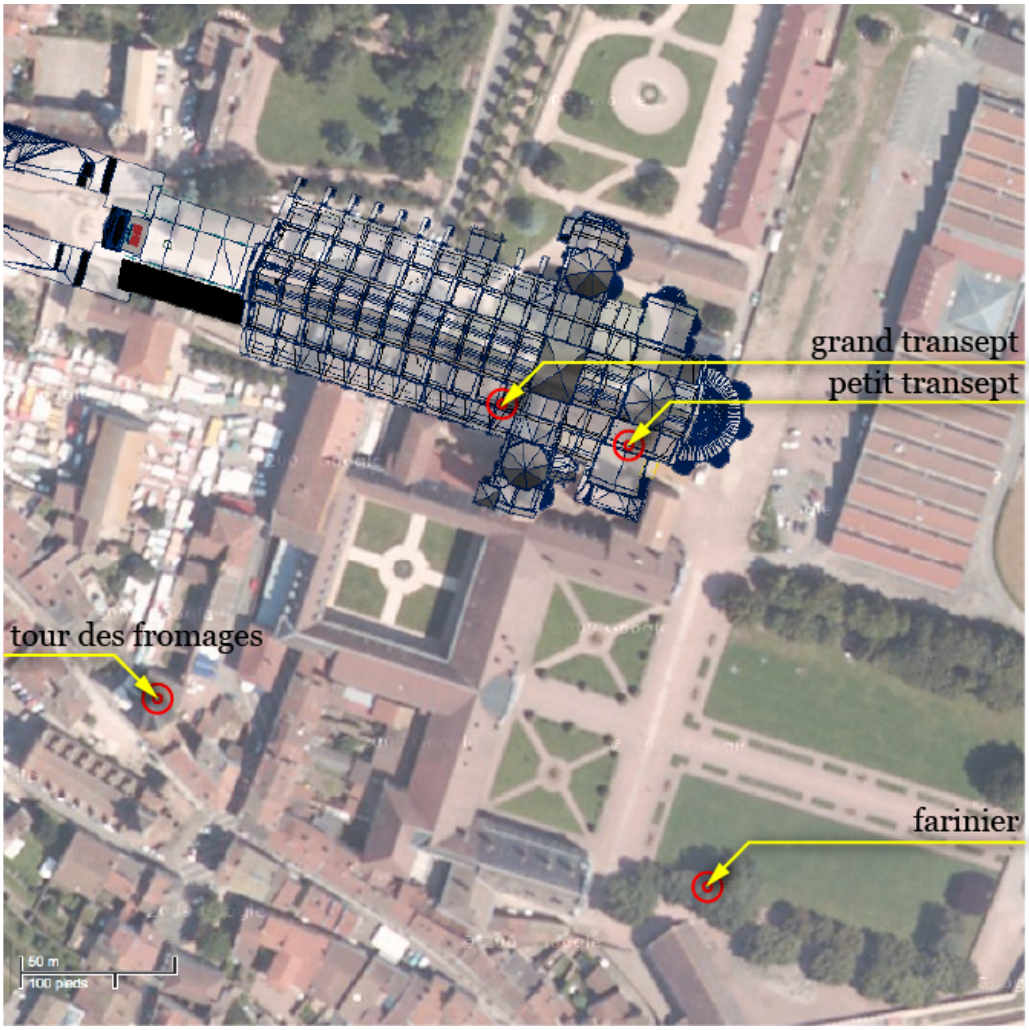


Figure 1.7 – Position des dispositifs ray-on

Figure 1.8 – ray-on / parc de l'abbaye / on-situ

Figure 1.9 – ray-on / tour des fromages / on-situ

Figure 1.10 – Intégration de l'église à la capture vidéo / on-situ

1.4 Participation au projet ray-on

Outre le travail de recherche développé dans la suite de ce document, une part importante de cette thèse en entreprise a été allouée au développement de ray-on durant toutes les phases, de la conception à la création du contenu et au développement logiciel, ainsi que dans certains choix graphiques.
La conception mécanique ayant été prise en charge par un cabinet d'étude externe, ce travail a consisté au choix des composants électroniques incluant les éléments permettant le déport de la partie informatique dans une régie à distance, ainsi que l'ensemble des capteurs permettant le bon fonctionnement du dispositif. De la même manière, la conception du système de refroidissement qui permet au dispositif de fonctionner sur une large plage de températures, en intérieur comme en extérieur, a été un sujet abordé pendant la conception du dispositif.
Par la suite, l'installation des dispositifs sur le site de Cluny s'étant inscrite dans le cadre plus vaste d'une rénovation des vestiges de l'église abbatiale ainsi que du farinier, il a été nécessaire d'encadrer la partie du chantier qui concernait ray-on (passage des câbles et des tuyaux pour le refroidissement, mise en place du socle). L'installation à proprement parler des dispositifs (montage sur le socle, mise en place des régies) a été faite par nos soins, de même que leur entretien tout au long de l'année.
La conception n'a pas été arrêtée après la fin du développement du premier dispositif ray-on installé dans le grand transept de l'église. Celui-ci peut en effet être considéré comme un prototype, et en tant que tel présente des défauts de conception. Les dispositifs suivants ayant été installés une année plus tard, nous avons pu prendre en compte ces défauts et avons modifié la conception pour améliorer en particulier la fiabilité. Les trois dispositifs les plus récents nécessitent ainsi beaucoup moins d'entretien que le premier.

Le travail relatif au développement du dispositif a également concerné la partie logicielle de ray-on. En dehors des développements liés aux travaux de recherche décrits par la suite, ce travail a été relatif à l'intégration logicielle des capteurs de ray-on (ainsi que le filtrage de leurs mesures), la création du shader pour l'accumulation des différentes couches du rendu avant affichage, et le rendu des bases de données d'images pour parvenir au rendu (voir chapitre 3 pour plus de détails sur la raison d'être de ces bases de données).

Chapitre 2

Problématique

2.1 Verrous liés à la réalité mixte

La principale caractéristique de la réalité mixte est de produire un mélange cohérent entre réel et virtuel. Cette cohérence désigne, entre autres possibilités de la RM : le positionnement réaliste des objets réels et virtuels entre eux ; les interactions lumineuses, c'est-à-dire l'éclairage d'objets réels par des sources virtuelles et inversement ; les interactions physiques, sous la forme de modifications du virtuel par les objets réels. La mise en place de ces points nécessite d'avoir accès à certaines caractéristiques de la scène réelle, comme sa géométrie, les matériaux la composant, le suivi d'objets précis, la détection et modélisation des sources lumineuses, etc. Les mêmes informations sont nécessaires du côté de la scène virtuelle mais, par définition, sont aisément accessibles.

La question de la détermination de la géométrie de la scène, si elle n'est pas spécifiée en amont de manière figée [107, 99], est abordée la plupart du temps par l'intermédiaire de caméras [84]. Pour obtenir une représentation 3D, les systèmes multi-caméras (stéréo ou plus) ainsi que les z-cameras sont les méthodes les plus utilisées. Cependant, elles présentent des limitations : la reconstruction multi-caméras est approximative et/ou très éloignée du temps réel, tandis que les z-cameras sous leur forme actuelle sont sensibles aux sources infrarouges ambiantes comme c'est le cas en extérieur et ont une portée réduite (de l'ordre de la dizaine de mètres).

Pour passer outre le problème de la reconstruction 3D, certaines implémentations se contentent d'utiliser des masques sur l'image réelle pour y ajouter les éléments virtuels. Cette approche convient pour des objets fixes mais n'est pas adaptée dans le cas, par exemple, de l'intégration d'objets réels mobiles dans une scène virtuelle. L'utilisation de marqueurs permet une implémentation plus souple, dans le sens où le point de vue et les objets peuvent ne pas être fixes. Elle est, en revanche, limitée aux implémentations de réalité augmentée (ajout d'objets virtuels dans une scène réelle), et n'est pas adaptée à tous les sites, en particulier les lieux culturels où la multiplication de marqueurs peut dénaturer l'environnement.

La segmentation présente également dans ses implémentations des limitations. Les méthodes de segmentation automatiques se basent pour la plupart sur une modélisation de l'arrière plan entretenue au fil du temps afin de supporter les changements importants de luminosité ou les modifications de l'arrière plan. La qualité du résultat reste cependant très variable. Des méthodes semi-automatiques, nécessitant l'intervention de l'utilisateur, donnent de très bons résultats mais sont intrinsèquement inadaptées à une utilisation temps réel.

Outre les verrous liés à la connaissance de la géométrie de la scène réelle, la reproduction de

l'éclairage réel sur les objets virtuels, ainsi que la réillumination d'objets réels par des sources virtuelles sont encore peu mises en oeuvre dans le cadre de la réalité mixte. De nombreuses approche proposent d'utiliser la notion de réalité diminuée pour améliorer l'intégration du réel au virtuel, en apposant par exemple un filtre sur les images réelles [44, 33, 19]. Nous souhaitons dans ce travail mettre l'accent sur le photoréalisme du résultat final.

2.2 Niveaux d'intégration réel - virtuel

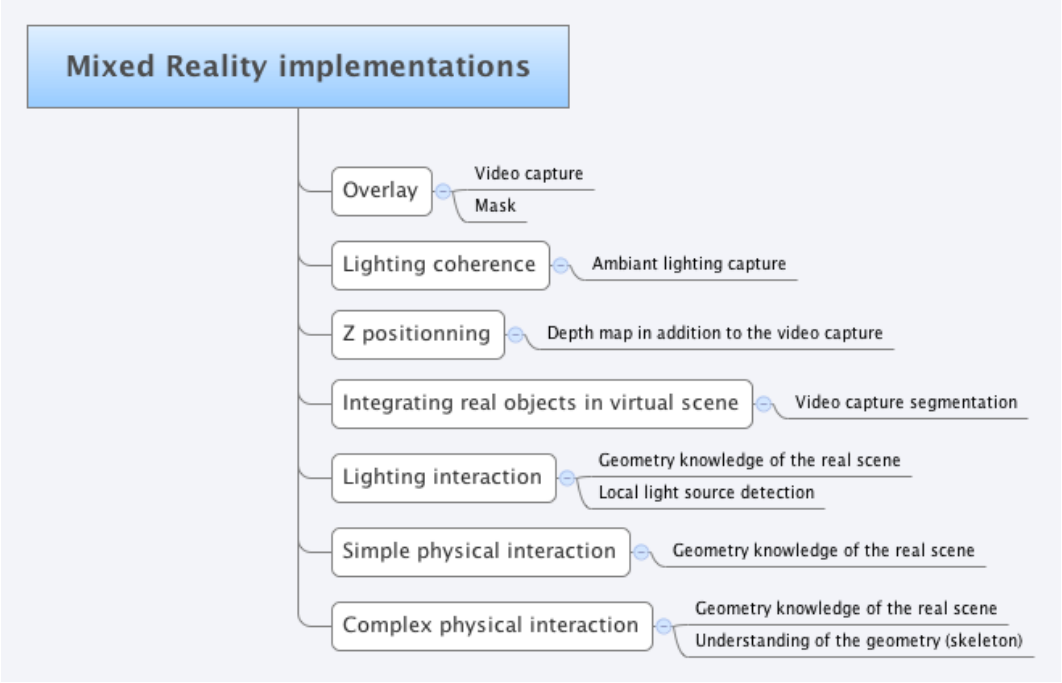


Figure 2.1 – Besoins de différentes implémentations de RM

On peut noter que la connaissance complète de la géométrie de la scène n'est pas nécessaire pour tous les niveaux d'intégration, et que l'on peut se contenter d'informations parcellaires dans certains cas :

- Détermination des contours des objets (segmentation)
- Détermination de la profondeur selon l'axe orthogonal au plan focal de la caméra (notée Z)
- Détermination des volumes de la scène.

De la même manière, la connaissance de l'environnement lumineux se décompose en deux types de sources :

- Sources globales : l'éclairage ambiant d'une pièce, ou l'éclairage naturel en extérieur
- Sources locales : sources lumineuses effectivement présentes dans l'environnement proche de l'utilisateur (une lampe par exemple)

Le *compositing* du réel et du virtuel peut être amélioré de deux manières différentes :

- En produisant du contenu virtuel photoréaliste [90, 79, 78]
- En filtrant les sources virtuelle et réelle pour qu'elles aient une apparence semblable, non-photoréaliste [19, 33, 90]

Dans notre optique de fenêtre donnant sur une version alternative du réel (uchronie), nous nous orientons vers la première solution.

2.3 Questions de recherche

Suite à la section précédente, on voit qu'un point d'évolution important pour améliorer les implémentations actuelles de la RM est une meilleure connaissance de la géométrie et de l'environnement lumineux de la scène réelle. Dans un premier temps, nous limiterons l'extraction d'informations géométriques à celles théoriquement accessibles à partir d'un unique point de vue, celles-ci étant suffisantes pour donner accès à de nombreuses nouvelles applications de RM :

- Segmentation en 2D d'un objet dans l'image selon le point de vue
- Evaluation de la distance au point de vue d'un objet

De la même manière, nous nous contenterons d'envisager la reproduction lumineuse globale de la scène réelle dans la scène virtuelle et mettrons de côté la détection de sources lumineuses locales.

Ce travail de thèse se propose donc d'aborder les deux questions de recherche suivantes :

- *Comment assurer, dans le cadre d'une application temps réel et en particulier de réalité mixte, que l'intégration d'éléments virtuels dans une scène réelle soit juste du point de vue de l'éclairage et de la couleur ?*
- *Comment segmenter un flux vidéo à l'éclairage changeant en temps réel, en maintenant un niveau de qualité constant quelque soit l'éclairage ?*

Chapitre 3

Cohérence lumineuse

La problématique de la cohérence lumineuse peut se décomposer en trois points qui sont : capture de l'environnement lumineux, reproduction de celui-ci sur les objets virtuels, et calcul des interactions lumineuses entre les objets réels et virtuels. L'intégration d'objets virtuels dans la capture d'une scène réelle profite de la cohérence lumineuse, que ce soit par la vraisemblance des niveaux de luminance, la cohérence de la colorimétrie entre les sources d'image réelles et virtuelles, ou la reproduction d'ombrages (liste non exhaustive).

Dans le cadre de la production audiovisuelle, ou plus largement pour tout ce qui ne touche pas au temps réel et donc à la réalité mixte, cette problématique est adressée par un travail au cas par cas. Celui-ci consiste principalement à ajuster les niveaux de luminance et les balances des blancs, ainsi qu'à diverses opérations de post-production concernant par exemple les ombres. Dans le cadre d'une application temps réel, ce travail n'est pas possible et nous avons au mieux l'application d'une balance des blancs et un réglage des niveaux automatiques, au pire aucun des deux. Dans tous les cas le résultat n'est pas optimal.

3.0.1 Problématique scientifique

Dans ce travail, nous cherchons à obtenir une cohérence lumineuse améliorée, de manière totalement automatique. Nous nous concentrerons sur la question des niveaux de luminance et la colorimétrie, en mettant de coté les interactions lumineuses qui forment, à elles seules, un sujet complet. Notre objectif est de pouvoir intégrer un objet virtuel dans une scène réelle, de telle manière que ses composantes trichromatiques soient identiques à celles objet réel qui aurait servi de modèle.

3.0.2 Question de recherche

La question de recherche qui est posée ici est la suivante : *Comment assurer, dans le cadre d'une application temps réel et en particulier de réalité mixte, que l'intégration d'éléments virtuels dans une scène réelle soit juste du point de vue de l'éclairage et de la couleur ?*

Pour traiter cette question, nous nous placerons dans des conditions d'éclairage quelconque mais néanmoins suffisante pour que les images capturées ne soit pas dégradées par un bruit important ou une dynamique réduite.

3.1 Modélisation et état de l'art

3.1.1 Capture de l'environnement lumineux

La détection des sources lumineuses consiste à produire une modélisation de l'environnement lumineux réel. La première phase consiste à capturer une image de la scène réelle pour en tirer les informations dont nous avons besoin. La méthode la plus immédiate est d'utiliser une caméra équipée d'un objectif fish-eye (offrant un champ de vision de $2 * \pi$ stéradians) [28, 51].
Une autre technique est celle utilisée par Debevec et al. [28] qui utilisent une sphère chromée qui, après calibrage géométrique, permet d'obtenir une modélisation de l'éclairage sur l'ensemble de l'environnement, c'est-à-dire sur 360°. L'avantage de cette technique est de pouvoir utiliser une unique caméra pour la capture de l'éclairage et la capture vidéo qui sera utilisée pour le compositing. En revanche ladite caméra sera nécessairement visible sur la capture de l'environnement lumineux, ainsi qu'un angle mort à l'opposé de la position de la caméra par rapport à la sphère chromée.
Ces deux premières techniques mènent le plus souvent à des méthodes d'éclairage de type éclairage à partir d'images (*image based lighting* ou IBL).

Aittala et al. [12] proposent d'aborder le problème par une approche dite de "rendu inverse" (*inverse rendering*), consistant à simuler l'éclairage d'une balle diffusante (une balle de ping pong dans ce cas) présente dans la scène réelle jusqu'à retrouver l'éclairage qui lui est appliqué.
D'autres approches existent qui, bien qu'étant moins précises, nécessitent un matériel moindre. Par exemple, Liu et al. [68] propose de modéliser l'environnement lumineux extérieur comme étant l'apport d'une part du soleil, d'autre part du reste du ciel. L'apport de chaque part est déterminé par une série de captures prises pour une position du soleil équivalente. Lalonde et al. [61] proposent quant à eux de déterminer les directions de lumière plus précisément, toujours à partir d'une seule image de la scène (et non de l'environnement lumineux).

L'ensemble de ces techniques présentent l'inconvénient de ne représenter l'environnement lumineux qu'en un point, sauf pour l'approche de capture de l'éclairage en stéréo proposée par Corsini et al. [22]. Ainsi, selon la scène et la distance à laquelle on se trouve de la position de capture de l'éclairage, les résultats se retrouvent faussés, et ce d'autant plus que l'environnement contient de sources lumineuses locales (ou non situées à l'infini par rapport aux dimensions de la scène). L'approche stéréo permet de connaître plus précisément la position de ces sources lumineuses locales et donc de mieux simuler l'éclairage en tout lieu de la scène.

A noter que selon la dynamique de la scène considérée, il peut s'avérer nécessaire de capturer une image à haute dynamique (HDRI pour *High Dynamic Range Image*) [28] de l'environnement dans le cas de l'utilisation d'un fish-eye ou d'une sphère chromée. Un appareil photo numérique récent peut capturer une image avec une dynamique supérieure à 10 stops (le rapport entre l'intensité la plus forte et la plus faible vaut au moins 2^{10}), une caméra industrielle de bonne qualité environ 6 stops, ce qui peut s'avérer insuffisant en particulier pour les scènes en extérieur où on peut atteindre facilement une dynamique de 20 stops. Nous parlons ici de la dynamique réelle du capteur, et non de la dynamique maximale offerte par le format dans lequel sont transmises les images : la plupart des appareils actuels sont dotés de convertisseurs analogique/numérique travaillant sur 12 ou 14 bits (offrant une dynamique théorique maximale de 12 ou 14 stops), mais la dynamique réelle dépend des caractéristiques du capteur et en particulier de la profondeur du puits de potentiel de ses photosites.

3.1.2 Combinaison des LDR

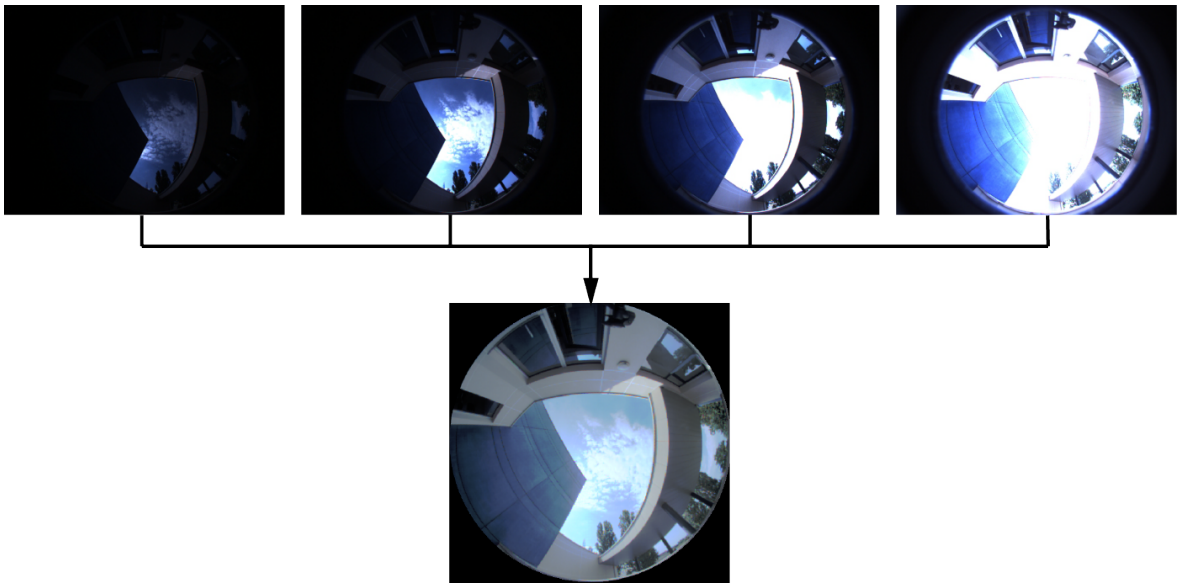


Figure 3.1 – Haut : images sources / Bas : image HDR résultante (après *tone mapping*)

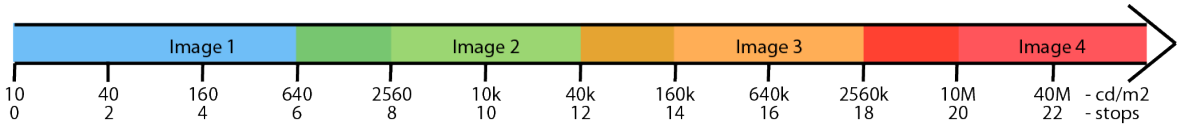


Figure 3.2 – Positionnement et recouvrement des images LDR sur l'échelle de puissance lumineuse

Extraction de l'information de luminance Une image HDR est créée à partir d'une série d'images d'une même scène prises à différents niveaux d'exposition (figure 3.1). Les images sont prises de manière à ce que deux images consécutives aient une zone de recouvrement au niveau de la portion de dynamique de la scène mesurée, comme montrée sur la figure 3.2. Pour calculer la luminance d'un pixel, nous avons besoin de connaître l'indice de lumination (noté EV) qui a été sélectionné au moment de la prise de vue :

$$EV = log_2(\frac{f^2}{t}\frac{100}{S})$$

avec :
- f : ouverture de l'objectif
- t : temps d'exposition, en secondes
- S : sensibilité ISO du capteur

En considérant que la caméra a été calibrée pour avoir une sortie linéaire en fonction de l'intensité lumineuse (ce qui peut revenir à annuler la plupart des traitements internes faits par la caméra sur l'image issue du capteur, les capteurs CCD et CMOS ayant une réponse linéaire), la luminance relative de chaque pixel peut être calculée de la manière suivante :

$$L_{pix} = \frac{Z}{Z_0} * \frac{K * f^2}{S * t}$$

$$L_{pix} = \frac{Z}{Z_0} * \frac{K}{100} * 2^{EV}$$

avec :

- L_{pix} : luminance du pixel sur le canal considéré (selon l'espace colorimétrique utilisé) dans l'image HDR
- Z : valeur du pixel sur l'image LDR (*Low Dynamic Range*) considérée
- Z_0 : valeur "de base" sur l'image LDR, définie par le type de sensibilité ISO considéré. La norme la plus utilisée est la ISO_{SOS} (pour *Standard Output Sensitivity*), où $Z_0 = 0.71 * Z_{max}$.

Equation 3.1 – Luminance d'un pixel selon les paramètres de la prise de vue

Le paramètre K est laissé à la discrétion du fabricant de l'appareil, et peut varier d'après la norme (ISO 2720) entre 10.6 et 13.4 [54], même si certains constructeurs prennent des libertés (Kodak par exemple utilise une valeur de 15.4 pour certains de ses capteurs). La connaissance de cette constante n'a pas d'influence sur la qualité des HDRI créées, seulement sur la justesse physique des luminances obtenues. Nous choisirons pour l'instant une valeur intermédiaire $K = 12$. De même pour le paramètre Z_0 qui pourra dans un premier temps être choisi arbitrairement ($Z_0 > 0$).

$$L_{pix} = \frac{Z}{Z_0} * \frac{12}{100} * 2^{EV}$$

$$L_{pix} = 2^{log_2 Z - log_2 Z_0 + EV + log_2(12) - log_2 100}$$

$$L_{pix} = 2^{log_2 Z - log_2 Z_0 + EV - 3}$$

Dans une certaine mesure, il est possible de calibrer les HDRI de telle manière qu'il soit possible d'en tirer des informations de luminance physiquement cohérentes, en choisissant les paramètres K et Z_0 cohérents avec les choix du fabricant, et en corrigeant les résultats par comparaison avec un luminance-mètre. Inanici et al. [48] montre ainsi que l'on peut obtenir une précision de l'ordre de 10% dans l'estimation de la luminance sur une très large plage de dynamique.

Il reste alors à combiner les informations issues des différentes images LDR. Une méthode couramment utilisée consiste à pondérer la contribution de chaque image pour un pixel donné en fonction de son écart à la valeur médiane (128 par exemple pour une image en RGB 8 bits par composante). On considère en effet que les capteurs ont la meilleure précision aux alentours de cette valeur. Nous verrons cependant par la suite que la courbe de réponse d'une caméra n'est pas nécessairement linéaire, il est alors nécessaire de convertir les valeurs brutes selon une échelle évoluant linéairement avec la quantité de lumière atteignant le capteur.

Pour conserver notre approche consistant à prendre la valeur centrale comme référence, nous pouvons définir Z_{stop} comme étant l'écart, en stops, entre la quantité de lumière correspondant à la valeur de référence (c'est à dire 128) et la valeur considérée. Z_{stop} s'écrit donc comme suit :

$$Z_{stop} = log_2 \frac{Z}{Z_0}$$

On comprendra que le calibrage de la réponse d'une caméra consiste à associer à chaque valeur de chacun de ses canaux un écart en stops à la valeur de référence considérée. La

luminance du pixel s'exprime finalement ainsi, à une constante prêt :

$$L^* = 2^{Z_{stop}+EV}$$

Il est alors possible de convertir les images d'une représentation discrète, à partir de nombres entiers, à une représentation par des nombre réels ayant une correspondance physique puisque quantifiant la quantité de lumière arrivant au pixel. Reste maintenant à fusionner les informations issues des différentes images pour en tirer une HDRI.

Fusion des images : Avant tout, rappelons l'intérêt de la fusion d'images LDR en une HDRI. Une photo prise de manière automatique sera exposée de manière à ce que les valeurs de luminance aux alentours de la moyenne de la scène soient décrites correctement, ce qui ne sera pas le cas pour les zones sombres (sous-exposées) ou très lumineuses (sur-exposées). En combinant des images du même point de vue à différentes expositions, il devient possible d'extraire des détails de ces zones sous et sur exposées.
L'étape précédente nous a permis de représenter toutes ces images sur une même échelle, qui est celle de la luminance relative L^*. La luminance d'un même pixel est donc calculée autant de fois qu'il y a d'images, donnant en théorie un résultat identique à chaque fois, ce qui n'est pas le cas en pratique. La fusion de toutes ces sources d'information part du principe que les capteurs sont plus précis aux alentours de la valeur médiane (soit 128), pour l'être de moins en moins à mesure que l'on s'approche des bornes 0 et 255. Une approche courante consiste à pondérer les informations de chaque image en fonction de l'écart de la valeur du pixel à la valeur médiane, en appliquant une courbe gaussienne par exemple (voir figure 3.3).

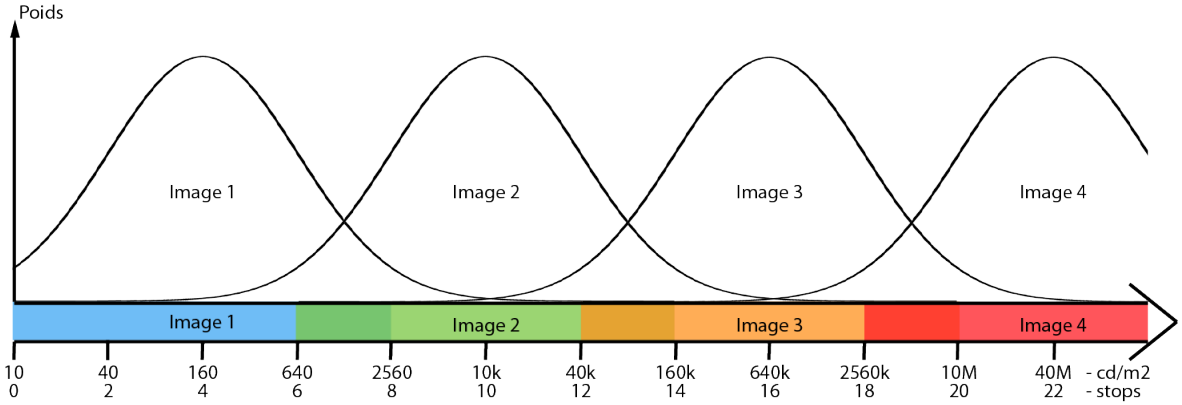


Figure 3.3 – Création d'une HDRI - ordonnée : poids attribué à chaque image

Le résultat de cette fusion est une image HDR profitant de la multiplicité des sources d'information pour avoir un bruit très réduit, en dehors des limites inférieures et supérieures. La dynamique de cette nouvelle image est la suivante :

$$d_{stop} = d_{cam} + (n_{images} - 1).\delta_{images}$$

avec :
- d_{stop} : dynamique de l'image HDR
- d_{cam} : dynamique d'une capture simple issue de la caméra
- n_{images} : nombre d'images LDR utilisées pour former l'image HDR
- δ_{images} : différence d'exposition, en stops, entre deux images LDR successives

Figure 3.4 – Coloration selon l'image prépondérante (voir 3.1 et 3.3)

3.1.3 Reproduction de l'environnement lumineux

La reproduction de l'éclairage réel dans la scène virtuelle peut être abordée de plusieurs manières, selon la qualité, la précision et le niveau d'interactivité que l'on veut obtenir. Les méthodes présentées ici sont organisées de la moins juste visuellement à la plus juste.

La méthode la plus approximative, mais également celle qui offre le plus haut niveau d'interactivité, s'appuie sur une modélisation de l'environnement lumineux décomposé en une lumière d'environnement (le ciel par exemple), et une lumière directionnelle (le soleil). C'est la stratégie utilisée par Liu et al. [68] et Lalonde et al. [61]. Après avoir déterminé les paramètres de ces deux sources lumineuses, il est aisé de les utiliser dans le cadre d'un rendu en temps réel. Cette approche réduit grandement la complexité de l'éclairage, ce qui en réduit la vraisemblance en particulier dans le cadre d'éclairage artificiel (scène d'intérieur, de nuit, etc.). De plus les diverses interactions lumineuses restent à la charge du moteur de rendu. Une approche semblable consiste à modéliser l'éclairage comme une combinaison d'éclairages "unitaires" [12], les limitations étant les mêmes.

3.1.3.1 Transfert de radiance pré-calculé (*Precomputed radiance transfer*)

Afin de reproduire plus précisément l'éclairement dans une scène virtuelle, les méthodes de type *Precomputed Radiance Transfer* (ou PRT) sont très utilisées, en particulier dans le milieu du jeu vidéo ce qui explique leur développement rapide. L'objet des PRT est de fournir une représentation compressée des interactions lumineuses entre les objets en tenant compte de leur matériau (représenté par une fonction de distribution de la réflectance bidirectionnelle, ou BRDF), en particulier dans le cas d'interactions complexes, cette représentation permettant ensuite de simplifier grandement l'équation de rendu, qui peut alors être résolue de manière

approchée en temps réel. Les premiers à envisager cette approche ont été Sloan et al. [92]. Le principe des méthodes de type PRT est de ramener l'équation de rendu à une combinaison linéaire de réalisations de l'éclairage selon des environnements lumineux prédéterminés.

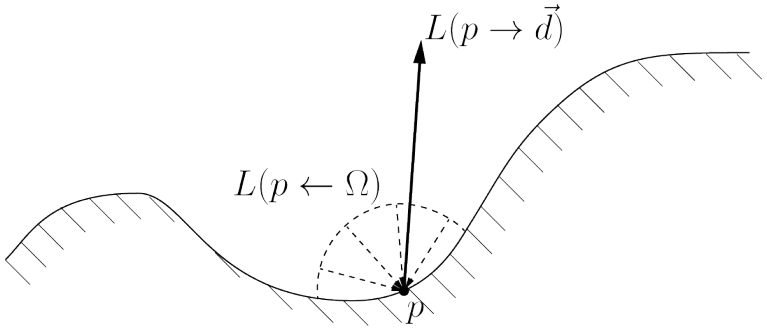


Figure 3.5 – Réalisation de la luminance dans la direction $\vec{d}$ pour l'ensemble des rayons incidents au point p

$$L(p \to \vec{d}) = L_e(p \to \vec{d}) + \int_\Omega f_r(p, \vec{s} \to \vec{d}) L(p \leftarrow \vec{s}) H_{N_p}(-\vec{s}) d\vec{s}$$

avec :

- Ω : on intègre sur la sphère
- $L(p \to \vec{d})$: luminance quittant le point p dans la direction $\vec{d}$.
- $L_e(p \to \vec{d})$: luminance émise du point p dans la direction $\vec{d}$ par l'objet. Cette valeur est nulle pour la plupart des matériaux, et on la considérera comme telle dans la suite.
- $f_r(p, \vec{s} \to \vec{d})$: fonction de distribution de la réflectance bidirectionnelle (*Bidirectional Reflectance Distribution Function.* Voir fonction de distribution de la réflectance bidirectionnelle. (BRDF)) au point p avec une direction incidente $\vec{s}$ et une direction de réflexion $\vec{d}$
- $L(p \leftarrow \vec{s})$: luminance incidente au point p selon la direction $\vec{s}$
- $H_{N_p}(-\vec{s})$: coefficient d'atténuation, correspondant au cosinus entre la normale à l'objet et le rayon incident (ramené à 0 si le résultat est négatif)

Equation 3.2 – Equation de rendu

En partant de l'équation de rendu 3.2, on peut obtenir la relation suivante par l'intermédiaire d'un développement de Neumann :

$$L(p \to \vec{d}) = L_0(p \to \vec{d}) + L_1(p \to \vec{d}) + ... + L_n(p \to \vec{d}) + ...$$

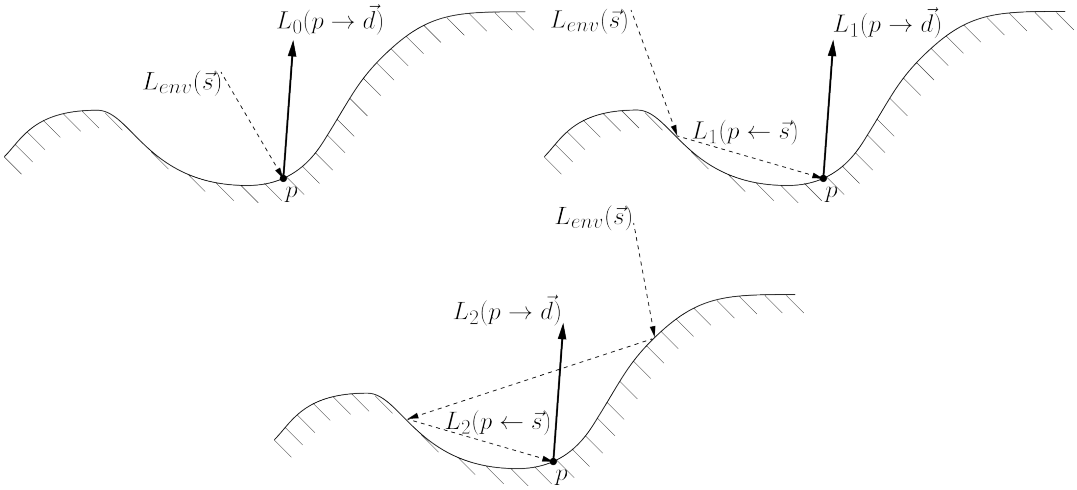


FIGURE 3.6 – Trois niveaux de rebonds intervenant dans la luminance $L(p \to \vec{d})$

Chaque membre de cette somme correspond à la contribution des rayons ayant fait n rebonds avant de parvenir au point p. L_0 correspond donc à l'illumination directe de l'objet par l'environnement lumineux :

$$L_0(p \to \vec{d}) = \int_\Omega f_r(p, \vec{s} \to \vec{d}) L_{env}(\vec{s}) V(p \to \vec{s}) H_{N_p}(-\vec{s}) d\vec{s}$$

avec :
- $L_{env}(\vec{s})$: luminance de l'environnement dans la direction $\vec{s}$
- $V(p \to \vec{s})$: visiblité (valant 0 ou 1) du point p dans la direction s.

La contribution d'un nouveau rebond correspond à l'illumination par le rebond précédent :

$$L_n(p \to \vec{d}) = \int_\Omega f_r(p, \vec{s} \to \vec{d}) L_{n-1}(p \leftarrow \vec{s})(1 - V(p \to \vec{s})) H_{N_p}(-\vec{s}) d\vec{s}$$

On a ici le complémentaire de la visibilité, c'est à dire $(1 - V(p \to \vec{s}))$, puisque pour que la direction $\vec{s}$ puisse être issue d'un rebond, il faut nécessairement que le point p ne soit pas visible de l'environnement selon cette direction.
Pour les approches suivantes, l'hypothèse est faite que les matériaux sont parfaitement diffusants. Un raisonnement semblable est possible pour des matériaux ayant des BRDF plus complexes. Considérons donc un matériau ayant un facteur de réflexion (ou réflectance) ρ_d, l'éclairage direct devient :

$$L_0(p) = \frac{\rho_d}{\pi} \int_\Omega L_{env}(\vec{s}) V(p \to \vec{s}) H_{N_p}(-\vec{s}) d\vec{s}$$

Plus généralement :

$$L_n(p) = \frac{\rho_d}{\pi} \int_\Omega L_{n-1}(p \leftarrow \vec{s})(1 - V(p \to \vec{s})) H_{N_p}(-\vec{s}) d\vec{s}$$

A noter que le matériau étant considéré comme diffusant, la réflectance du matériau ne dépend plus de la direction d'observation.
Le principe des méthodes PRT consiste alors à échantillonner l'environnement lumineux de manière à se ramener à un nombre fini de réalisations de celui-ci. On note alors :

$$L_{env}(\vec{s}) = \sum_i l_i y_i(\vec{s})$$

L'ensemble y_i définissant une base de l'espace des cartes de luminance, choisie de telle manière que la carte L_{env} puisse être représentée par une somme pondérée des éléments de cette base. L'illustration la plus simple d'une telle base est celle pour laquelle chaque y_i représente la contribution d'un pixel de la carte de luminance. La somme pondérée par l'ensemble des l_i, représentant alors la couleur de chacun des pixels, permet de reconstituer la carte de luminance complète.

Après simplification, on obtient les expressions suivantes :

$$L_0(p) = \sum_i l_i t^0_{pi} \text{ avec : } t^0_{pi} = \frac{\rho_d}{\pi} \int_\Omega y_i(\vec{s}) V(p \to \vec{s}) H_{N_p}(-\vec{s}) d\vec{s}$$

$$L_n(p) = \sum_i l_i t^n_{pi} \text{ avec : } t^n_{pi} = \frac{\rho_d}{\pi} \int_\Omega t^{n-1}_{pi} (1 - V(p \to \vec{s})) H_{N_p}(-\vec{s}) d\vec{s}$$

Finalement, on obtient la luminance au point p :

$$L(p) = \sum_i l_i (t^0_{pi} + t^1_{pi} + ... + t^n_{pi} + ...)$$

$$L(p) = \sum_i l_i t_{pi}$$

La luminance au point p est donc la combinaison linéaire des t_{pi}, qui sont les réalisations de la radiosité de l'objet au point p éclairé par une réalisation figée y_i de l'environnement lumineux, et sont donc précalculables. Le temps de rendu final se retrouve indépendant de la précision souhaitée, que l'on ne veuille avoir que l'éclairage direct, les ombres, ou les interréflexions pour obtenir l'éclairage global.

Cependant, la quantité de données à stocker et à manipuler devient rapidement très importante avec la complexité de la scène et la qualité de l'échantillonnage de l'environnement lumineux. Pour résoudre ce point, la formulation des t_{pi} comme dépendants de réalisations de base de l'environnement lumineux permet d'envisager d'exprimer les y_i dans une base autorisant une manipulation plus souple. Les deux approches les plus utilisées sont les harmoniques sphériques et les ondelettes (de Haar en particulier).

Les méthodes de type PRT donnent de très bons résultats, mais imposent cependant que la scène soit géométriquement figée : les objets ne peuvent être déplacés ni modifiés (des approches récentes permettent cependant d'envisager des déformations). Ces méthodes conservent de plus la complexité géométrique de la scène, ce qui peut s'avérer problématique dans le cas d'un modèle de taille importante. Les méthodes présentées par la suite sont indépendantes de la géométrie.

3.1.3.2 Image based rendering

Les méthodes de type *Image-based Rendering* (IBR), permettent de s'affranchir de la complexité de la scène. Parmi celles-ci, celles s'appuyant sur la fonction plénoptique ont en

outre l'avantage de permettre une modification du point de vue dans une certaine mesure (correspondant au déplacement de la tête par rapport à un écran par exemple). La fonction plénoptique, explicitée par Adelson et al. [10], décrit le flux lumineux tel qu'il est appréhendé par l'oeil humain, en fonction de sa position dans l'espace et de son orientation :

$$P = P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$$

avec :
- (θ, ϕ) : orientation de l'observateur
- (V_x, V_y, V_z) : position de l'observateur
- λ : longueur d'onde, en m
- t : temps, en s

Cette fonction à sept dimensions a été simplifiée par Gortler et al. [40] en une fonction 4D, en paramétrant les rayons arrivant sur l'oeil comme passant au travers de deux plans parallèles (u, v) et (s, t). L'ensemble des rayons pour une vue donnée s'intersectent au point focal correspondant à la position de l'oeil. De plus, la longueur d'onde et le temps peuvent être considérés comme constants si la scène ne change pas. Cette paramétrisation est nommée *Lumigraph* :

$$L = L(s, t, u, v)$$

L'intérêt de cette paramétrisation est de simplifier la représentation des données par rapport à la fonction plénoptique de base, tout en conservant (dans une moindre mesure) la possibilité de déplacer le point de vue. Cependant il manque la notion d'illumination pour faire le lien avec notre problématique. La fonction plénoptique a été complétée par Wong et al. [105] afin de contenir une description de l'éclairage ambiant :

$$I = P(\theta, \phi, \theta_L, \phi_L, V_x, V_y, V_z)$$

avec :
- I : luminance
- (θ_L, ϕ_L) : direction d'une lumière directionnelle

En combinant cette expression avec la paramétrisation en 4D de Gortler et al. [40], on obtient :

$$I = L(\theta_L, \phi_L, s, t, u, v)$$

Il s'agit de la fonction plénoptique-illumination, en 6D. La valeur d'un pixel est déterminée selon la direction de la lumière directionnelle (θ_L, ϕ_L), et la direction du point de vue (s, t, u, v) : on a finalement une paramétrisation proche de celle d'une BRDF. On peut en effet considérer que la valeur d'un pixel correspond à la radiosité de l'élément de coordonnées (s, t) du plan st selon la direction (s, t, u, v) illuminé par la lumière directionnelle (θ_L, ϕ_L). Wong et al. [105] parle alors de BRDF apparente, ou pBRDF pour *pixel BRDF* puisqu'il s'agit de la BRDF d'un plan imaginaire représentant ce qui se trouve au-delà.
La quantité de données pour représenter la pBRDF est importante mais peut être compressée, par l'utilisation d'harmoniques sphériques ou d'ondelettes par exemple. En revanche, elle est constante quelque soit la complexité de la scène et de son éclairage. Cette approche profite également de la linéarité de l'éclairage présentée plus haut, permettant d'envisager des éclairages complexes.

Du fait qu'il s'agisse d'une méthode de type IBR, la scène est bien entendue considérée comme fixe.

Enfin, le type d'approche le plus rigide, ou le plus limitant quant aux interactions et aux déplacements de l'utilisateur, fait également partie des méthodes de type IBR. Il s'agit d'une approche proposée en premier lieu par Haeberli [42], basée elle aussi sur la linéarité de l'apport des sources lumineuses d'une scène. Dans son exemple, il propose d'éclairer une scène (réelle) avec une lumière bleue, puis avec une lumière rouge. Les combinaisons linéaires des deux images issues de ces éclairages différents permettent de recréer tout un panel de nouveaux éclairages intermédiaires, voire impossibles : en appliquant des coefficients négatifs, on parvient à créer des sources de lumière "négatives", c'est à dire qui réduisent l'intensité lumineuse 3.7.

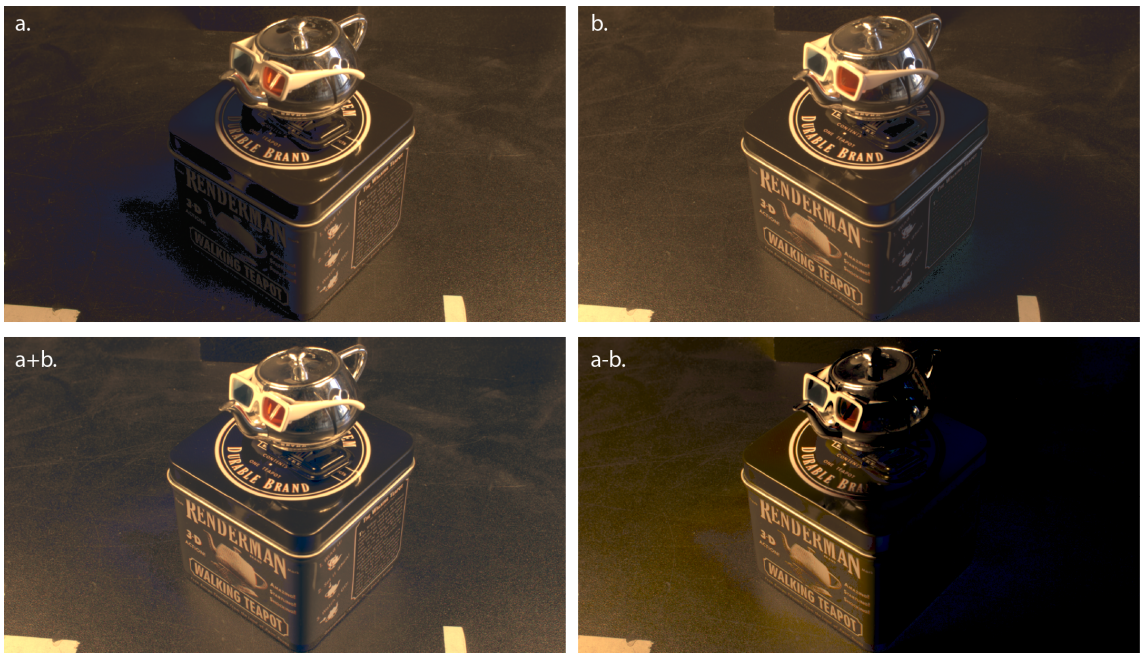


Figure 3.7 – Haut : deux photos d'une scène éclairée de deux manières différentes - Bas : somme et soustraction des deux images précédentes, la soustraction revenant à simuler une source "négative" de lumière.

En étendant ce principe à un nombre plus important d'images sources, il est envisageable de recréer n'importe quel éclairage. Nimeroff et al. [76] se proposent par exemple de simuler les conditions lumineuses tout au long d'une journée à partir d'une base de neuf images sources. Debevec et al. [27] développent cette idée et proposent de construire une base de 2048 images, puis d'utiliser une capture de l'éclairage (HDRI sphérique ou de type lat-long) pour réilluminer la scène (un visage en l'occurrence), de même que Muller et al. [74] dans le cadre de la mise en valeur architecturale.
Ces méthodes ne permettent pas de changer la position du point de vue dans la scène, mais en utilisant des panoramas autorisent malgré tout son orientation. Les temps de calculs et quantités de données sont, comme pour les méthodes précédentes basées sur la fonction plénoptique, indépendants de la complexité de la scène et de l'éclairage, et il est possible de compresser la base de données pour les réduire.

3.1.4 Modèles pour la mesure de la luminance et la colorimétrie

Pour mesurer la qualité de la reproduction de l'environnement lumineux en utilisant notre méthode, nous avons choisi de la définir de deux points de vues : celui de la photométrie, et celui de la perception par l'oeil humain. Deux espaces de représentation des images ont donc été utilisées.

Le premier espace sera utilisé pour mesurer l'écart de luminance entre une scène réelle capturée par une caméra, et sa reproduction virtuelle éclairée selon notre méthode. En partant de la méthode de capture des HDRI expliquées en section 3.1.2, un espace à une seule dimension décrivant uniquement le niveau de luminance est immédiatement utilisable. On peut souligner que dans le cas d'images HDR définies dans le format Radiance HDR, la luminance d'un pixel s'exprime de la manière suivante :

$$L_{pix} = 179(0.263L_{red} + 0.655L_{green} + 0.082L_{blue})$$

Cette expression est liée en particulier à l'illuminant utilisé comme référence par ce format, qui est l'illuminant neutre E. Ce sujet est détaillé dans ce qui suit.
Le second espace sera utilisé pour mesurer l'écart colorimétrique entre la capture de la scène réelle et la version virtuelle, tel que perçu par l'oeil humain. Il est donc nécessaire de s'assurer que les couleurs soient exprimées précisément tout au long de la chaîne colorimétrique, c'est à dire en limitant toute éventuelle perte d'information et s'assurant de comparer les couleurs dans un même espace colorimétrique.

L'oeil humain permet une vision de la couleur basée sur trois types de photorécepteurs différents que sont les cônes. Il est donc théoriquement possible de décrire une couleur, telle qu'elle est visible par l'oeil humain, à partir d'un triplet de valeurs nommé tristimulus.

Une couleur peut donc être décomposée en une somme pondérée de trois couleurs dites primaires, celles-ci formant un espace colorimétrique. Le choix des couleurs primaires est basé soit sur l'expérimentation, comme c'est le cas pour les espaces CIE XYZ et CIE RGB, mais peut également être lié à la recherche d'un espace remplissant une ou des conditions particulières. Ainsi, l'espace CIE L*a*b* a la particularité qu'un écart donné dans n'importe quelle direction de l'espace donnera la sensation d'une différence de couleur identique.

Nous allons dans un premier temps nous intéresser à l'espace CIE XYZ, nommé également CIE 1931 XYZ. Issu des travaux de W. David Wright et John Guild, il permet la représentation de l'ensemble des couleurs visibles par l'oeil humain (soit la réponse moyenne de l'oeil humain aux rayonnements électromagnétiques). Il est donc particulièrement adapté pour être la base de nombreux autres espaces colorimétriques, et les conversions d'espaces se font la plupart du temps en passant par celui-ci. Un espace en dérivant est l'espace xyY, dans lequel les composantes xy définissent la chromaticité, et la composante Y la luminance. Une représentation de cet espace est visible sur la figure 3.8. Les composantes sont obtenues par les relations suivantes :

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}$$

L'espace CIE RGB est également le résultat des travaux de Wright et Guild. Contrairement à l'espace CIE XYZ, l'objet de cet espace est de représenter les couleurs comme une combinaison de trois couleurs primaires. Une des conséquences de ce choix est que certaines couleurs ne peuvent être représentées sans avoir recours à des composantes négatives. Il est

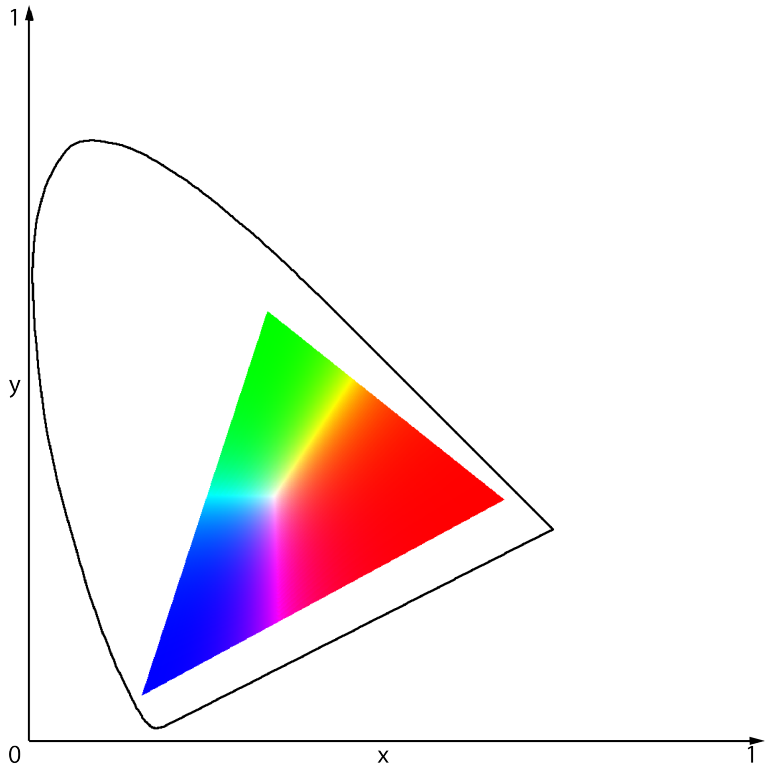


Figure 3.8 – Espace CIE xy - La zone colorée correspond à l'ensemble des couleurs représentables par l'espace sRGB, la frontière noire à l'ensemble des couleurs visibles par l'oeil humain.

malgré tout possible de passer d'un espace à l'autre, au prix d'une perte éventuelle d'information.

Des espaces plus spécifiques existent, adaptés à certaines utilisations. Ceux qui nous intéressent dans le cadre de ce travail sont l'espace sRGB et le L*a*b*. L'espace sRGB est dérivé du RGB et résulte en première approximation de l'application d'un gamma de 2.2 sur les trois composantes C_{RGB} (voir plus loin pour la relation exacte). Cet espace a été créé à destination de l'affichage sur les écran cathodiques (CRT) pour lesquels la luminosité n'est pas une fonction linéaire de la puissance du faisceau.

$$C_{sRGB} = \begin{cases} 12,92.C_{RGB} & C_{RGB} \leq 0,0031308 \\ (1+0,055) * C_{RGB}^{1/2,4} - 0,055 & C_{RGB} > 0,0031308 \end{cases}$$

L'espace CIE L*a*b*, comme dit précédemment, vise à ce que l'écart entre deux couleurs dans cet espace soit cohérent avec la différence ressentie entre celles-ci par l'oeil humain. Dans cet espace, la composante L* désigne la clarté (de noir à blanc), a* est l'axe allant de rouge à vert, et b* de jaune à bleu (figure 3.9). La conversion de XYZ vers L*a*b* se fait de la manière suivante :

$$L = 116.f(y) - 16$$

$$a = 500.(f(x) - f(y))$$

$$b = 200.(f(y) - f(z))$$

$$f_c = \begin{cases} \sqrt[3]{c_r} & c_r > 0,008856 \\ \frac{903,3.c_r+16}{116} & c_r \leq 0,008856 \end{cases}$$

$$c_r = \frac{C}{C_w}$$

C_w désigne les composantes du point blanc considéré (aussi nommé illuminant), dont les coordonnées dans l'espace XYZ sont (X_w, Y_w, Z_w).

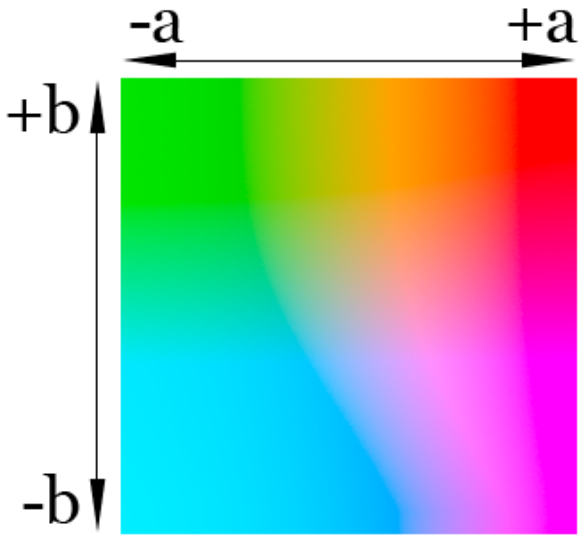


Figure 3.9 – L'espace L*a*b*, pour L*=75%

L'illuminant dépend de l'éclairage, c'est à dire de la source de lumière considérée. Tous les espaces colorimétriques ne sont pas définis avec un illuminant neutre ($x = y = \frac{1}{3}$). Les illuminants que nous aurons à utiliser dans la suite seront l'illuminant E (qui est justement neutre) et l'illuminant D65 qui correspond à la lumière du jour à midi ($x = 0,31271$, $y = 0,32902$ dans sa définition de 1931).
Certains espaces colorimétriques imposent un illuminant, comme c'est le cas pour le sRGB pour lequel les couleurs sont exprimées selon l'illuminant D65. Pour le CIE XYZ et le CIE RGB en revanche, on peut passer d'un illuminant à l'autre (on parle alors d'adaptation chromatique). Notons que les adaptations chromatiques se font rarement de manière exacte et que la qualité du résultat dépend de la méthode utilisée.

3.2 Approche proposée

3.2.1 Calibrage des caméras

Comme nous l'avons vu dans l'état de l'art sur la capture de l'environnement lumineux, plusieurs paramètres sont au libre choix du fabricant de la caméra utilisée. Dans l'optique d'avoir des HDRI physiquement correctes il est nécessaire dans un premier temps soit d'accéder à ces paramètres, soit de les déterminer par l'expérimentation. Les fabricants étant peu désireux de partager ces informations, c'est la seconde méthode qui a été privilégiée. Quelques uns d'entre eux fournissent des documents sur leurs méthodes de calibrage des capteurs (c'est le cas de Kodak), mais les informations ne sont que parcellaires et nécessitent de toute façon d'être complétées.

Figure 3.10 – Droite : balance des blancs automatique, luminance globale mesurée sur l'HRDI de $236cd/m^2$ - Gauche : balance des blancs neutre, luminance globale mesurée sur l'HRDI de $280cd/m^2$

Le calibrage vise donc à évaluer la réponse du capteur à un flux lumineux, afin de pouvoir estimer à l'inverse le flux lumineux en fonction de la réponse du capteur. La connaissance de la sensibilité du capteur permet, après la construction de l'image HDR, d'évaluer la luminance des objets de la scène.
Pour obtenir une estimation correcte de la luminance, il faut également s'affranchir de la courbe de réponse possiblement accidentée du capteur, c'est à dire s'assurer de sa linéarité : en omettant les phénomènes de saturation, on a donc pour une luminance L incidente sur le capteur une réponse $\alpha.L$ du capteur, α étant une valeur constante pour une exposition donnée.

Cette évaluation est cependant achromatique : la luminance évaluée ici est la luminance totale, sans pouvoir discerner la part de luminance issue du canal rouge, vert ou bleu. De plus, la luminance est définie dans le cadre du format Radiance HDR comme étant :

$$L_{pix} = 179(0.263L_{red} + 0.655L_{green} + 0.082L_{blue})$$

Equation 3.3 – Luminance d'un pixel dans le format Radiance HDR

On voit donc que la proportion de luminance apportée par chaque canal n'est pas la même, ce qui peut se révéler problématique dans le cas où chaque canal n'est pas calibré correctement. C'est le cas lorsque la balance des blancs d'une caméra n'est pas réglée correctement, ou est réglée de manière automatique.
Quelques précisions au sujet de la balance des blancs. Ce mécanisme permet de pondérer la quantité de rouge et de bleu (par rapport au canal vert, du moins pour les capteurs à base de filtre de Bayer), afin que les objets blancs restent blancs sur les images capturées quelque soit l'éclairage. Dans notre cas, ce mécanisme entraîne la perte de la proportion réelle de rouge, vert et bleu, et donc fausse la mesure finale. La puissance lumineuse totale mesurée sur l'image HDR n'est donc pas correcte, et il est de plus impossible d'évaluer la proportion de celle-ci due à chacun des trois canaux (figure 3.10). Il faut donc mettre en place une méthode permettant d'obtenir une balance des blancs "neutre", ne modifiant pas les proportions réelles.

Finalement, nous avons vu que le calcul de la luminance nécessite d'avoir une source de capture dont la réponse est linéaire, et une balance des blancs neutre. Le calibrage de la sensibilité ISO de la caméra étant basé sur l'évaluation de la luminance (ce point étant détaillé

dans ce qui suit), et la balance des blancs nécessitant elle-même d'avoir une réponse linéaire du capteur, l'ordre du calibrage est le suivant :

1. Calibrage de la réponse de la caméra
2. Détermination d'une balance des blancs neutre
3. Evaluation de la sensibilité ISO

Courbe de réponse de la caméra : La fusion d'images LDR pour former une image HDR impose tout d'abord de connaître la courbe de réponse de la caméra, afin de pouvoir établir une relation entre la valeur des pixels et leur luminance. Dans un premier temps, avant le calibrage de la caméra, nous parlerons de luminance relative L^*.

A l'image des dispositifs de reproduction vidéo, tels que les écrans ou les vidéoprojecteurs, les caméras et appareils photos appliquent la plupart du temps une courbe de correction dite "gamma" aux images qu'ils renvoient. Sur chaque canal, la formule suivante est appliquée :

$$c_{pix} = c_{cam}^{\gamma}$$

c_{pix} étant la valeur du canal sur l'image finale, et c_{cam} la valeur telle que capturée par la caméra. Le but de cet exposant est de mieux correspondre à la courbe de réponse de l'oeil humain qui perçoit plus de détails dans les zones sombres que dans les zones trop exposées. A titre d'exemple, le gris moyen d'une image sur laquelle un gamma de 2.2 a été appliqué, ou plus exactement ce qui semble être le gris moyen à l'oeil humain, a une valeur de $0.5^{2.2} = 0.18$. On comprend aisément qu'avec une telle transformation, une luminance deux fois supérieure de la scène ne correspond pas à une valeur deux fois supérieure dans l'image finale.
A noter que certains espaces colorimétriques intègrent par définition une transformation semblable à la correction gamma. C'est le cas de l'espace sRGB qui est celui utilisé par défaut sur la plupart des ordinateurs et dans certains formats comme le jpeg. La courbe est cependant sensiblement plus complexe qu'un simple exposant γ.

D'autre part, le calibrage permet de connaître la dynamique réelle (en stops) de chaque pixel d'une caméra. Il faut en effet dissocier la précision de la conversion analogique-numérique (A/N) de la dynamique du capteur. Augmenter la précision de la conversion A/N, en passant par exemple de 12 à 14 bits, réduit les erreurs de mesure du signal analogique en sortie du capteur mais n'augmente pas nécessairement la dynamique. Un des intérêts de l'utilisation d'un convertisseur plus précis est d'améliorer la reproduction des zones sombres du fait d'un bruit plus faible.

Pour pouvoir fusionner de manière correcte des images LDR, il est nécessaire de se ramener à des images dont la courbe de réponse associée est linéaire. Sur une telle image, une valeur de pixel deux fois supérieure implique une luminosité multipliée par deux.

Nous avons utilisé le logiciel HDRShop, issu des travaux de Paul Debevec et al. [28] [26] pour déterminer la courbe de réponse de nos appareils. Le principe de base est simple. Considérons un capteur doté d'un seul pixel, avec lequel nous prendrions une série de photos d'une même scène avec différents temps d'exposition, les autres paramètres (ouverture de l'optique, éclairage de la scène, etc.) ne changeant pas.

Ainsi pour un capteur linéaire, la valeur du pixel pour chaque exposition devrait être liée au temps d'exposition comme suit, avec $\{t_i\}$ l'ensemble des temps d'exposition et $\{c_i\}$ l'ensemble des réalisations de la valeur du pixel :

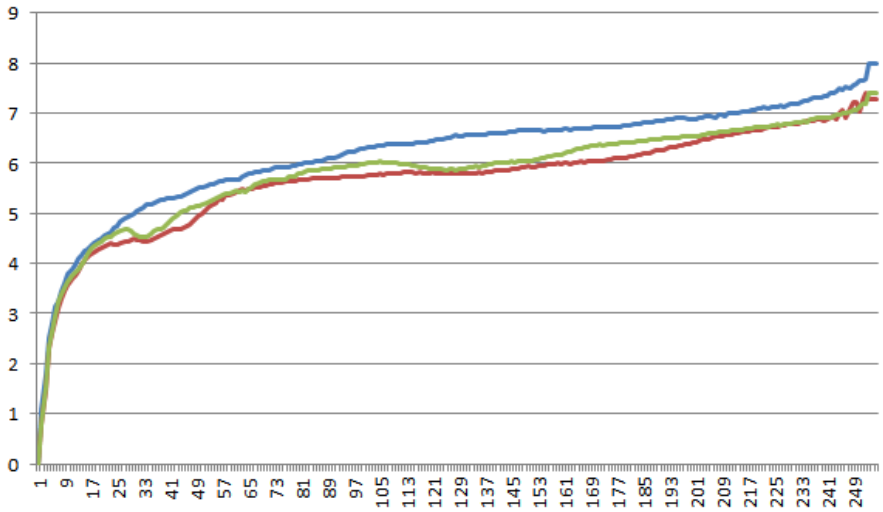


Figure 3.11 – Courbe de réponse RGB d'une caméra - abscisse : valeur du pixel ; ordonnée : exposition en stops

$$k_i = \frac{t_{i+1}}{t_i}$$

$$c_{i+1} = k_i.c_i$$

Dans le cas d'une image sur laquelle serait appliquée une courbe gamma, il s'agirait de déterminer le coefficient γ à l'aide des couples $\{t_i, c_i\}$. Plus généralement, il s'agit de déterminer une courbe approchant au mieux l'ensemble de ces réalisations.

Le résultat du calibrage par HDRShop est une table de correspondance (ou LUT pour *Lookup Table*) fournissant, pour chaque valeur d'entrée comprise entre 0 et 255 (image 8bits par canal), une valeur correspondant au nombre de stops d'écart à la valeur moyenne (127.5 dans l'image d'origine), et ce pour chaque canal. Cette LUT permet alors de convertir une image non linéaire en image linéaire.

Le résultat du calibrage des caméras donne une dynamique d'environ sept stops, ce qui est cohérent avec la documentation des capteurs. Dans la suite de cette section, nous considérerons que toutes les images sont converties pour être linéaires.

Balance des blancs : Dans le cadre des capteurs employant un filtre de Bayer (figure 3.12), le canal vert est le principal puisque ce filtre est composé d'autant de photosites verts que de rouges et de bleus réunis. Ce choix est dû à la position centrale du vert dans le spectre visuel de l'oeil humain, ce qui en fait le canal priviligié pour mesurer la luminance. La balance des blancs se fait donc par rapport à celui-ci : un coefficient est appliqué aux canaux rouge et bleu et le canal vert reste inchangé. En notant (r, g, b) les valeurs renvoyées par le capteur, et (R, G, B) les valeurs après l'application de la balance, on a donc :

$$\begin{cases} R = \alpha.r \\ G = g \\ B = \beta.b \end{cases}$$

Pour déterminer les coefficients α et β, notre démarche a consisté à mesurer la couleur d'un élément éclairé par une source connue, pour évaluer le biais par rapport à la couleur

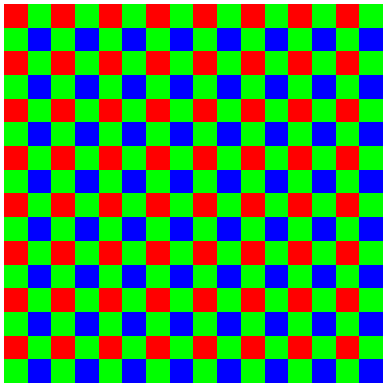


Figure 3.12 – Organisation des pixels sur un capteur doté d'un filtre de Bayer

réelle attendue et estimée à partir de la température de couleur de la source lumineuse et les propriétés du matériau de l'objet. En l'occurrence, nous avons utilisé les éléments gris de la mire colorimétrique ColorChecker de x-rite (voir figure 3.19). La mesure de la couleur sur des éléments gris permet en effet de s'assurer que les proportions de rouge, vert et bleu restent telles que lors de l'émission par la source lumineuse.
Pour reprendre les notions sur les espaces colorimétriques données précédemment, cela revient à chercher à retrouver les caractéristiques de l'illuminant (notre source connue) dans la lumière diffusée pour les éléments chromatiquement neutres. L'utilisation des bons paramètres α et β nous permettra d'avoir une mesure neutre sur ces éléments dans le cas d'un illuminant lui aussi neutre.

La température de couleur permet de déterminer le spectre d'une source lumineuse modélisée par un corps noir doté de cette même température, et en particulier les proportions de rouge, vert et bleu qu'elle émet. Ce spectre peut être évalué à l'aide de la loi de Planck 3.4 [81].

$$B_\lambda(T) = \frac{2.h.c^2}{\lambda^5}.\frac{1}{exp(\frac{h.c}{\lambda.k_B.T}) - 1}$$

avec :
- $B_\lambda(T)$ la puissance à la longueur d'onde et la température considérée, en $W.m^{-2}.Hz^{-1}.sr^{-1}$
- la longueur d'onde considérée λ
- la température considérée T
- la constante de Planck $h = 6,626.10^{-34}J.s$
- la constante de Boltzmann $k_b = 1,381.10^{-23}J.K^{-1}$
- la vitesse de la lumière dans le vide c

Equation 3.4 – Loi de Planck

Cette relation est cependant peu pratique à utiliser. Kim et al. [56] proposent de calculer le lieu des corps noirs à la température T, dans l'espace colorimétrique CIE xyY, de la façon suivante :

$$x = \begin{cases} -0,2661239.\frac{10^9}{T^3} - 0,2343580.\frac{10^6}{T^2} + 0,8776956.\frac{10^3}{T} + 0,179910 & 1667K \leq T \leq 4000K \\ -3,0258469.\frac{10^9}{T^3} + 2,1070379.\frac{10^6}{T^2} + 0,2226347.\frac{10^3}{T} + 0,0240390 & 4000K \leq T \leq 25000K \end{cases}$$

$$y = \begin{cases} -1,1063814.x^3 - 1,34811020.x^2 + 2,18555832.x - 0,20219683 & 1667K \leq T \leq 2222K \\ -0,9549476.x^3 - 1,37418593.x^2 + 2,09137015.x - 0,16748867 & 2222K \leq T \leq 4000K \\ 3,0817580.x^3 - 5,87338670.x^2 + 3,75112997.x - 0,37001483 & 4000K \leq T \leq 25000K \end{cases}$$

La composante Y dans l'espace CIE xyY correspond à la luminance et peut dans notre cas être choisie arbitrairement puisque c'est la chromaticité représentée par les composantes xy qui nous intéresse. On peut alors convertir le résultat de l'espace CIE xyY à l'espace CIE XYZ, puis à l'espace CIE RGB, en considérant un illuminant neutre (CIE E) pour ces deux derniers :

$$X = \frac{Y}{y}.x \qquad Z = \frac{Y}{y}.(1 - x - y)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 2,745 & -0,900 & -0,471 \\ -0,514 & 1,425 & 0,088 \\ 0,005 & 0,015 & 1,009 \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

En connaissant ainsi la colorimétrie de l'éclairage utilisé, nous pouvons déduire quelle mesure attendre des caméras si celles-ci sont calibrées. Contrairement à une balance des blancs "standard" où l'on cherche à annuler l'influence de l'éclairage sur la couleur des objets, nous souhaitons ici un calibrage neutre afin de mesurer la colorimétrie réelle des objets sous l'éclairage utilisé. Cela signifie que les éléments blancs de notre mire doivent être mesurés avec les mêmes rapports α et β que ceux représentant la température de couleur de l'éclairage. Notons qu'une balance des blancs neutre équivaut à exprimer les canaux RGB dans un espace de couleur doté de l'illuminant E.

Calibrage de la sensibilité ISO avec un luxmètre : La caméra étant destinée à rester installée sur le site, une caméra industrielle de type GC655C de Prosilica a été choisie. Cependant, sur ce type de matériel, très peu d'informations sont disponibles, pas même la sensibilité de base du capteur. N'ayant pas accès à la méthode de spécification de la sensibilité utilisée par Prosilica, ce calibrage est approximatif dans le sens où on ne trouvera pas la valeur constructeur réelle, mais permet de partir sur des bases cohérentes.

Le calibrage de la caméra est réalisé à l'aide d'un luxmètre. Le principe est le suivant : il s'agit de comparer les mesures délivrées par le luxmètre avec celles déterminées sur les HDRI obtenues par l'intermédiaire de la caméra. La relation liant les mesures du luxmètre (en *lux* donc) avec celles mesurées sur les HDRI (en $W.sr^{-1}.m^{-2}$, ou $cd.m^{-2}$) est issue de la modélisation des matériaux par la loi de Lambert 3.5.

$$L = I.\frac{\cos\theta}{\pi}$$

avec :
- L : luminance en $cd.m^{-2}$, ou $lm.sr^{-1}.m^{-2}$ - Valeur mesurée par le HDRI
- I : illuminance en *lux* ou $lm.m^{-2}$ - Valeur mesurée sur les luxmètre
- θ : angle d'incidence, mesuré par rapport à la normale de la surface

Equation 3.5 – Loi de Lambert

En considérant une source lumineuse placée orthogonalement à une surface parfaitement diffusante, θ peut être considéré comme étant nul. La luminance et l'illuminance sont alors reliées par un simple facteur π.
Rappelons que par définition, le format de fichier Radiance HDR permet le calcul de la luminance totale (en tenant compte de l'apport des trois composantes RGB) de la manière suivante :

$$L_{pix} = 179(0.263L_{red} + 0.655L_{green} + 0.082L_{blue})$$

Cette relation est liée à l'utilisation de l'illuminant E pour exprimer les valeurs des trois canaux. Le calibrage colorimétrique précédent se faisant avec ce même illuminant, nous pouvons utiliser la relation sur chacun de nos images LDR, et nous pouvons en déduire la sensibilité S de nos caméras en utilisant la relation précédent avec l'expression de la luminance donnée dans la section 3.1.2 :

$$L_{pix} = \frac{Z}{Z_0} * \frac{K * f^2}{S * t}$$

3.2.2 Reproduction de l'éclairage dans une scène virtuelle

3.2.2.1 Echantillonnage de l'environnement lumineux

La méthode de rendu choisie est de type IBR (*Image Based Rendering*), pour des raisons de qualité de la reproduction de l'environnement lumineux. Comme cela a été expliqué dans l'état de l'art, cette approche nécessite d'échantillonner l'environnement lumineux afin de se ramener à la somme pondérée d'un nombre fini de réalisations unitaires de l'éclairage.
L'environnement est capturé par l'intermédiaire d'une caméra dotée d'un objectif de type fish-eye, pointant à la verticale en direction du zénith, offrant un champ de vision à 180° dans toutes les directions. Nous cherchons donc à discrétiser une demi-sphère. A partir du travail de Schroeder et al. [88] et Robart et al. [85], nous avons décidé de partir d'une pyramide à base carrée raffinée par étapes successives jusqu'à obtenir la résolution désirée.

Les procédures de subdivision représentées par les deux méthodes de la figure 3.13 sont très semblables. L'objet de départ est, dans notre cas, un demi-octaèdre (c'est à dire une pyramide), dont chaque face triangulaire est subdivisée en quatre nouveaux triangles. Cette procédure est appliquée à chaque niveau de subdivision. Dans le cas de l'approche proposée par Robart [85], les triangles sont projetés à chaque itération sur la sphère dans laquelle est inscrit l'octaèdre de départ.
Ces deux décompositions ont l'avantage d'être très adaptée à une utilisation avec les ondelettes de Haar, mais ne produisent pas des faces couvrant toutes le même angle solide. L'avantage de la solution de Robart est de limiter sensiblement le rapport entre la face couvrant le plus grand angle solide et celui couvrant le plus petit, celui-ci passant d'environ 5 dans le cas de la méthode de Schroeder et al. (cette valeur dépendant du niveau de subdivision) à environ 2 (figure 3.14). Ceci permet d'avoir une subdivision qui aura moins tendance à favoriser certaines directions de lumière.

Nous avons donc choisi d'utiliser la subdivision proposée par Robart et al. L'image

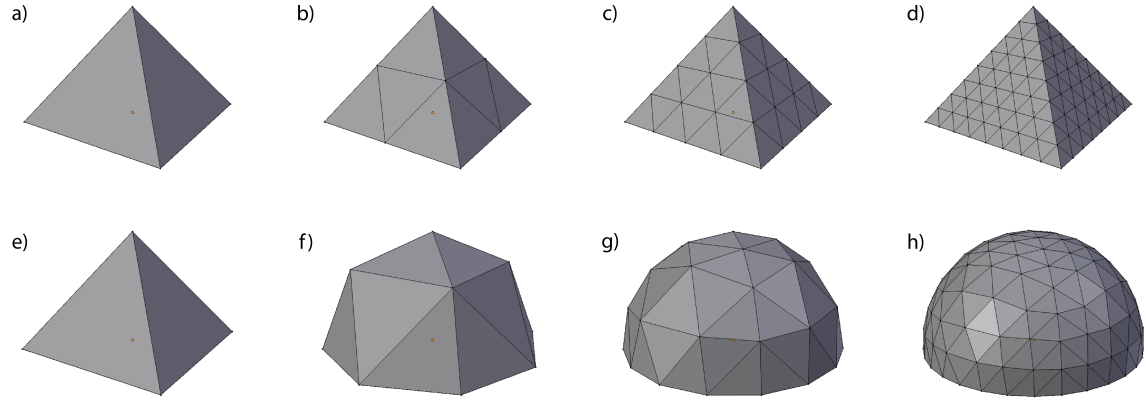


Figure 3.13 – Différentes subdivisions. a,e) Pyramide d'origine ; b,c,d) 3 niveaux de subdivision [88] ; f,g,h) 3 niveaux de subdivision avec projection sur la sphère [85].

Figure 3.14 – Coloration des triangles selon l'angle solide correspondant

capturée par la caméra fish-eye (nommée *lightprobe*) est appliquée en tant que texture sur celle-ci comme le montre la figure 3.15.

Pour chaque facette est alors calculée son exitance totale sur chacun des canaux RGB, c'est à dire le flux lumineux équivalent partant de la facette considérée. Cette exitance est issue de la somme des exitances de tous les pixels inclus dans l'angle solide correspondant à la facette considérée, à un coefficient près pour conserver la somme des exitances des facettes égale à l'exitance sur l'ensemble de la demi-sphère. Celle-ci équivaut à la somme totale des exitances des pixels de l'HDRI :

$$E_\Omega = \sum_{p \in P} E_p$$

Avec P le nombre total de pixels de la capture de l'éclairage, Ω la demi-sphère, E_Ω et E_p les exitances sur la demi-sphère et sur un pixel p, respectivement, en $cd.m^{-2}$. Il s'agit de la transposition de l'éclairement de l'environnement lumineux sur la scène, par unité de surface des objets de cette scène, en une exitance émise par les facettes de notre discrétisation. L'éclairement mesuré par chaque pixel de la *light probe* peut être exprimée comme le produit de la luminance mesurée par ce pixel par l'angle solide qu'il couvre :

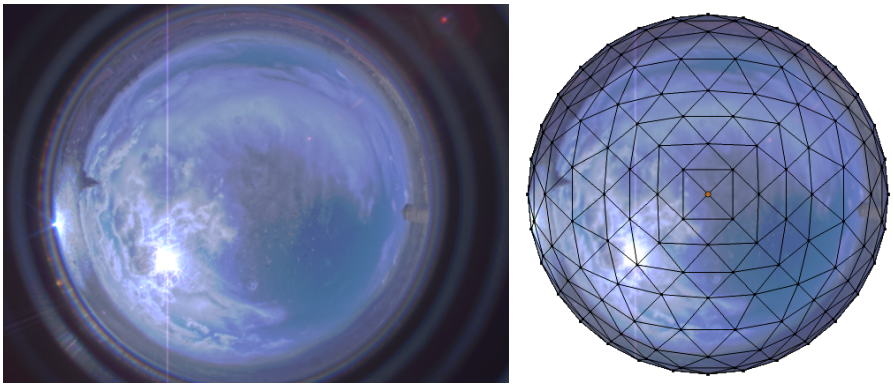


Figure 3.15 – Gauche : lightprobe source ; Droite : lightprobe projetée sur le maillage

$$E_\Omega = \sum_{p \in P} L_p S_p$$

Avec L_p la luminance du pixel et S_p son angle solide. Cet éclairement, égal à l'exitance issue des facettes, peut être exprimé en prenant soin de séparer les énergies apportées par chaque facette de la subdivision de la demi-sphère :

$$E_\Omega = \sum_{i \in \Omega} \sum_{p \in i} L_p S_p$$

i désignant l'index des facettes. En considérant que les angles solides couverts par chaque pixel au sein d'une même facette sont égaux, on a finalement en les notant s_i :

$$E_\Omega = \sum_{i \in \Omega} s_i \sum_{p \in i} L_p$$

$$E_\Omega = \sum_{i \in \Omega} \frac{S_i}{P_i} \sum_{p \in i} L_p$$

Avec S_i et P_i l'angle solide et le nombre de pixels de la facette i, respectivement. Le coefficient $\frac{S_i}{P_i}$ permet de prendre en considération les variations dans la forme des différentes facettes ainsi que le fait que tous les pixels ne couvrent pas le même angle solide.
Finalement, on obtient un champ de luminance de 4^n directions, selon le degré de précision désiré, représentant une approximation de la *light probe*.

3.2.2.2 Création de la base de données

Rappelons tout d'abord que, dans cette étude, l'environnement lumineux est considéré comme étant composé de sources situées à l'infini de la scène, impliquant que tous les rayons lumineux issus d'une même source sont parallèles. Sur la base de la première approximation apportée par l'échantillonnage proposé précédemment, nous allons maintenant supposer que tous les rayons issus d'une même facette (c'est à dire contenus dans un même angle solide représenté par ladite facette) sont eux aussi parallèles.

Nous pouvons dans ces conditions modéliser l'éclairage par une facette par une source lumineuse située à l'infini dans la direction définie par le centre de la facette. La luminosité ainsi que la couleur de cette source sont issues des données extraites pour la facette, comme expliqué dans la section précédente.
Finalement, l'environnement lumineux est représenté par un nombre fini de sources lumineuses dont les directions et les caractéristiques sont issues de la discrétisation de la demi-sphère représentant l'environnement lumineux réel capté par notre caméra fish-eye. Ce nouvel environnement lumineux tend vers l'environnement avant discrétisation à mesure que l'on affine celle-ci, et il est facile de vérifier que les énergies totales des deux environnements sont égales.

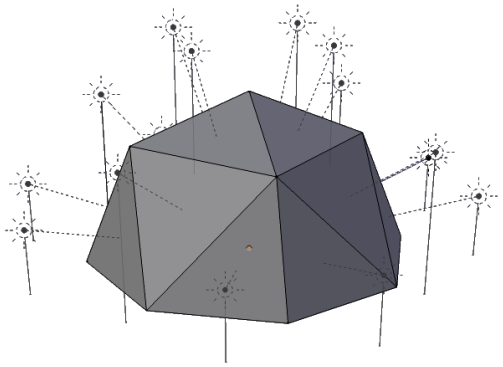


Figure 3.16 – Ensemble des directions de lumière pour une subdivision de niveau 2

La *light probe* issue de la capture du ciel permet, comme nous l'avons vu dans ce qui précède, de déterminer l'énergie apportée par chaque facette pour l'illumination de la scène. Pour utiliser cette information dans le cadre d'une méthode de type *Image Based Rendering* (IBR), nous avons besoin de précalculer quelle sera l'influence de chaque direction de lumière sur l'illumination de la scène.
Pour ce faire, nous avons choisi de calculer le rendu de notre scène pour chaque direction de lumière en attribuant une valeur unitaire à celle-ci, qui plus est sans unité. Le résultat sera finalement pour chacune de ces directions une image haute dynamique représentant un ensemble de coefficients correspondant aux pixels de l'image, qu'il faudra appliquer à l'énergie issue de la facette, pour obtenir une image de la contribution de cette facette. Les images précalculées étant un ensemble de coefficients, elles n'ont pas d'unité.
La création de ces sources lumineuses directionnelles unitaires n'est pas évidente, du fait des différences entre les moteurs de rendu au niveau des échelles de valeurs. Notre travail se base sur le format Radiance HDR, impliquant que la puissance de la source lumineuse à spécifier au moteur de rendu doit être calculée dans le cadre de ce format. Une méthode simple a été mise en oeuvre pour déterminer la puissance lumineuse des sources pour que leur influence soit celle d'une source unitaire, ceci afin de calculer l'influence d'une source unitaire dans l'éclairage de la scène. En pondérant l'image résultante par la puissance lumineuse déterminée sur la capture de l'environnement lumineux dans la direction considérée, on obtient la contribution réelle (photométrique) de cette direction.

Cette méthode est indépendante du moteur de rendu utilisé. La principale contrainte sur celui-ci est qu'il respecte la conservation de l'énergie, c'est à dire que les matériaux

ne renvoient pas plus d'énergie qu'ils n'en reçoivent. Elle a été mise en place et testée avec succès avec deux moteurs de rendu différents, Mental Ray et LuxRender. Elle se base sur une scène simple, composée de :

- un plan, doté d'un matériau blanc parfaitement diffusant (de type Lambert)
- une source lumineuse directionnelle blanche également, orientée selon la normale au plan
- une caméra, orientée de la même manière que la caméra réelle

Nous partons de l'expression de la réflectance d'un objet parfaitement diffusant :

$$L_r(x, \vec{\omega_r}) = \frac{\rho_d}{\pi} E(x)$$

x est le point de l'objet considéré, $E(x)$ l'éclairement en ce point, $\vec{\omega_r}$ la direction de l'observation, ρ_d le facteur de réflexion. Nous considérerons ici que ce dernier vaut 1. A partir de cette expression, il est évident que pour une source de lumière unitaire, la valeur de tous les pixels d'un rendu du point de vue d'une caméra dont la direction identique à celle de la source lumineuse doit être $\frac{1}{\pi} = 0.318$ sur tous les canaux RGB. Le rendu doit de plus être fait dans un espace de couleur doté de l'illuminant neutre E. Sachant cela, il est possible de déterminer le coefficient à appliquer à la source de lumière pour en faire une source unitaire.

3.2.2.3 Ré-illumination de la scène : sommation sur le champ de radiance

Il reste maintenant à faire la somme des luminances issues de la *light probe*, pondérées pour chaque pixel par les coefficients issus de la base de données pré-rendue, selon la formule vue dans l'état de l'art :

$$L(p) = \sum_i l_i t_{pi}$$

L'ensemble de la chaîne étant physiquement cohérente, le résultat de cette somme l'est également à condition que la méthode de rendue employée pour le pré-rendu respecte la contrainte de conservation de l'énergie.

3.2.3 Métriques

Les deux espaces de représentation de nos images, définis dans la section 3.1.4, nous donnent accès à deux métriques qui nous permettront de quantifier la qualité de la reproduction de l'éclairage. Le premier espace est à une seule dimension qui est la luminance, et la métrique utilisée est adaptée à la dynamique très importante que l'on peut retrouver dans une scène réelle. Elle est définie par la relation suivante, L_1 et L_2 étant les deux luminances à comparer :

$$L_{stop} = log_2 \frac{L_1}{L_2}$$

Le second espace, qui est l'espace colorimétriqué L*a*b*, a la particularité d'être plus représentatif de la perception humaine en plus de couvrir la totalité du gammut de

l'oeil humain. Cet espace est doté de la métrique δ_E, dont plusieurs définitions existent. La première définition de cette métrique est une simple distance euclidienne entre deux couleurs (L_1, a_1, b_1) et (L_2, a_2, b_2) (delta-E 1976) 3.6.

$$\Delta E = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2}$$

Equation 3.6 – Formulation de la métrique Delta E 1976

Cependant, malgré les particularités de l'espace L*a*b*, cette métrique n'est pas totalement satisfaisante et tend à noter comme éloignées au niveau perception deux couleurs qui sont très peu discernables par l'oeil humain. D'autres expressions ont donc été proposées pour répondre à ces lacunes, en particulier le delta-E 1994 qui est très utilisé. Il est calculé selon la relation 3.7.

$$\Delta E = \sqrt{(\frac{\Delta L}{K_L S_L})^2 + (\frac{\Delta C}{K_C S_C})^2 + (\frac{\Delta H}{K_H S_H})^2}$$

avec :

$$C_1 = \sqrt{a_1^2 + b_1^2}$$

$$C_2 = \sqrt{a_2^2 + b_2^2}$$

$$\Delta L = L_1 - L_2$$

$$\Delta C = C_1 - C_2$$

$$\Delta H = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2 - (\Delta C)^2}$$

$$S_L = 1$$

$$S_C = 1 + K_1 C_1$$

$$S_H = 1 + K_2 C_1$$

Equation 3.7 – Formulation de la métrique Delta E 1994

Les paramètres K_1, K_2 K_L, K_C et K_H sont dépendants du domaine d'utilisation de cette métrique. Les valeurs recommandées sont les suivantes, et nous utiliserons quant à nous celles destinées aux arts graphiques :

	Arts graphiques	Textiles
K_1	0.045	0.048
K_2	0.015	0.014
K_L	1	2
K_C	1	1
K_H	1	1

L'utilisation de la métrique delta-E nécessite quelques précautions pour que la comparaison de deux couleurs ait un sens. Les objets dont sont tirées les couleurs doivent être éclairés de manière identique (le type d'éclairage et son intensité en particulier),

et les couleurs doivent être exprimées selon le même illuminant. Les mesures n'ont de sens que si ces deux conditions sont remplies.
Dans le cadre de cette étude, le delta-E est utilisé pour évaluer les différences de chromaticité entre deux échantillons de même indice issus de deux images différentes. Son formulation standard intègre cependant la considération de la différence de luminance par le biais du ΔL. La différence de luminance étant évaluée à part, nous allons tâcher de limiter les différences de luminance entre les éléments en exposant les deux images de manière identique. Nous chercherons donc à ce que l'élément blanc d'indice 19 soit le plus saturé possible (du point de vue de sa luminance), en choisissant une exposition telle que la composante principale de cet élément blanc ait une valeur comprise entre 0,95 et 1. De cette manière, l'ensemble des autres éléments auront des couleurs dont les composantes sont comprises entre 0 et 1, les expositions des deux images étant semblables.

Pour pouvoir interpréter correctement les valeurs mesurées, nous aurons besoin d'une référence. Une caractéristique importante du delta-E est qu'une valeur inférieure à 1 signifie qu'il est a priori impossible à l'oeil humain de discerner les deux couleurs considérées. Au-delà en revanche, l'interprétation des résultats dépend de la méthodologie utilisée pour la mesure, même s'il est admis dans le domaine de l'impression qu'une valeur entre 3 et 6 est acceptable. Le domaine de l'informatique graphique (hors impression) ne semble pas avoir de référentiel commun à ce sujet. De plus, le niveau pour lequel intervient la perception d'une différence entre les couleurs est dépendant des couleurs en question, l'oeil humain étant plus doué pour discerner certaines nuances. Pour mieux appréhender les mesures, celles-ci seront soutenues par une représentation des couples de couleurs comparées.
Nous analyserons dans un premier temps les résultats du calibrage des caméras, en mettant en opposition les images de la mire ColorChecker par celles-ci et la simulation des couleurs de la mire sous un éclairage identique. Puis nous étudierons les résultats de la simulation du re-éclairage de la mire selon la méthode présentée dans ce travail, en comparant les résultats aux prises de vue réelles. Pour quantifier l'apport de la prise en compte de la colorimétrie de l'éclairage dans cette simulation, nous simulerons également les éclairages en tenant compte uniquement de la luminance et en omettant les informations chromatiques.

3.3 Expérimentations

3.3.1 Banc de mesure

Le banc de mesure est organisé autour d'une mire de calibrage colorimétrique dont les caractéristiques sont connues et bien documentées. Nous avons ainsi utilisé la mire ColorChecker de X-Rite, fournie avec un tableau répertoriant les valeurs des différentes couleurs, aussi bien dans l'espace sRGB que dans l'espace L*a*b*. Ceci nous a permis de créer une modélisation fidèle de la mire pour être utilisée dans les deux moteurs de rendu choisis. Les différents éléments de couleur sont supposés être parfaitement diffusants (bien que cela soit physiquement impossible, l'approximation reste valable), l'orientation de la prise de vue n'influe donc théoriquement pas sur la luminance mesurée.

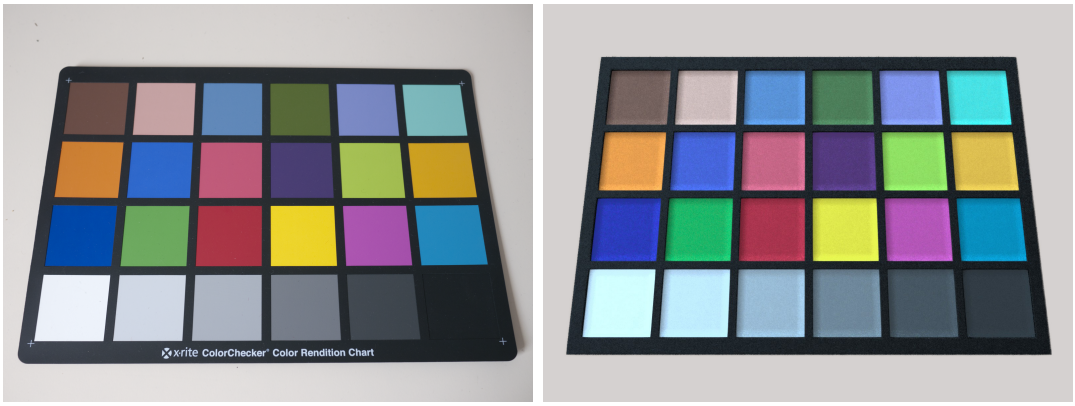


Figure 3.17 – Mire ColorChecker réelle et virtuelle

Les dispositifs de mesure du banc sont les suivants :
- un luxmètre
- une caméra pointée en direction de la mire, dont les prises de vues permettront d'en produire une image HDR physiquement correcte
- une caméra équipée d'un objectif fish-eye, placée au niveau de la mire, qui permettra la capture de l'environnement lumineux pour produire une *light probe* HDR.

Le luxmètre est placé de manière à être dans le champ de vision de la première caméra. Les deux caméras étant commandées de manière synchronisées, l'ensemble des mesures sont prises au même instant t.

Les mesures, sous la forme d'une information de luminance globale et de deux images HDR, permettent alors de créer d'une part, une simulation de la scène réelle (la mire) à l'aide de l'algorithme IBR présenté précédemment et de la base d'images précalculées, d'autre part une seconde simulation en convertissant la *light probe* d'une projection fish-eye à une projection latitude / longitude, la plupart des moteurs de rendu (et les deux choisis en particulier) n'acceptant pas les images fisheye comme texture d'environnement.

Connaissant les caractéristiques de la mire, il est possible à partir de la luminance globale de déterminer une approximation des niveaux de luminance attendus sur les deux simulations, ainsi que sur la capture HDR réelle de la mire.

Les mesures porteront sur l'ensemble des couleurs de la mire. Les éléments colorés permettront d'apprécier la justesse de la colorimétrie, tandis que les éléments neutres (et en particulier le blanc) permettront de valider en plus la simulation de la puissance lumineuse.

3.3.2 Conditions expérimentales et hypothèses

Le calcul de la luminance finale des objets est le résultat d'une chaîne comportant des approximations acceptable dans la plupart des cas. L'objet de l'expérimentation est cependant de déterminer si ces approximations n'empêchent pas d'avoir une bonne reproduction des niveaux de luminosité et de la colorimétrie des objets.

Nous émettons donc quelques hypothèses pour tenir compte des approximations

évoquées. Tout d'abord, la luminance de référence étant mesurée de la même manière au calibrage de la caméra, et comme référence pour évaluer la qualité de la reproduction, nous considérerons qu'un biais dans la mesure de la sensibilité sera constant tout au long des expérimentations.

Une autre source d'approximation tient dans la méthode de captation de l'environnement lumineux. Puisque nous utilisons un objectif fish-eye, nous capturons un dôme de 180° et non la totalité de l'environnement. Cependant la géométrie de notre objet de test est assimilable à un plan, nous ferons donc l'hypothèse que la partie non capturée n'éclaire pas la mire de référence.

Enfin, bien que cela soit très peu probable physiquement, l'ensemble des éléments de la mire sont modélisés par des matériaux parfaitement diffusants. La fonction de la mire de couleur suppose que les éléments soient suffisament diffusants pour que l'orientation de celle-ci n'influence pas le calibrage, nous pouvons donc considérer qu'il s'agit d'une hypothèse cohérente.

Deux environnements seront considérés dans le cadre de nos expérimentations, avec deux conditions d'éclairage différentes pour chacun. Le premier environnement sera en intérieur, et nommé "Bureau" par la suite. La mire ColorChecker sera disposée sur un banc de reproduction photographique qui est équipé de quatre lampes fournissant un éclairage uniforme. On prendra soin pour cette configuration de placer la caméra dotée de l'objectif fish-eye (chargée de capturer l'environnement lumineux) à la même position que la mire, du fait de la proximité des sources lumineuses. Les captures sont alors prises en deux temps, d'abord la capture de l'éclairage puis la capture de la mire une fois celle-ci remise en place. Cette configuration permettra d'évaluer la capacité de notre méthode à reproduire d'une part un éclairage diffus (lampes éteintes, éclairage naturel), d'autre part un éclairage artificiel (lampes allumées).

Le second environnement sera en extérieur, et nommé "Extérieur" par la suite. Les sources lumineuses étant suffisament distantes pour pouvoir être considérées à l'infini, la caméra fish-eye sera simplement placée à proximité de la mire. Les deux conditions d'éclairage seront en plein soleil d'une part afin d'évaluer la qualité de la reproduction dans le cas de la présence d'une source ponctuelle prépondérante par rapport au reste de l'éclairage, et à l'ombre d'autre part.

3.3.3 Mesures

Dans cette section, les éléments de la mire ColorChecker ne sont pas disposés dans l'ordre de leur indice, mais dans l'ordre de leur couleur de leur longueur d'onde dans la mesure du possible (hormis les gris), de manière à faire apparaître les éventuelles particularités liées à telle ou telle gamme de couleurs. Les différents niveaux de gris seront quant à eux placés à la fin, dans l'ordre décroissant de réflectance.
Notons d'autre part que les couleurs représentées dans ce document sont dépendantes de la qualité de la colorimétrie du support sur lequel il est lu. Les conditions idéales sont obtenues dans le cas d'un support calibré.

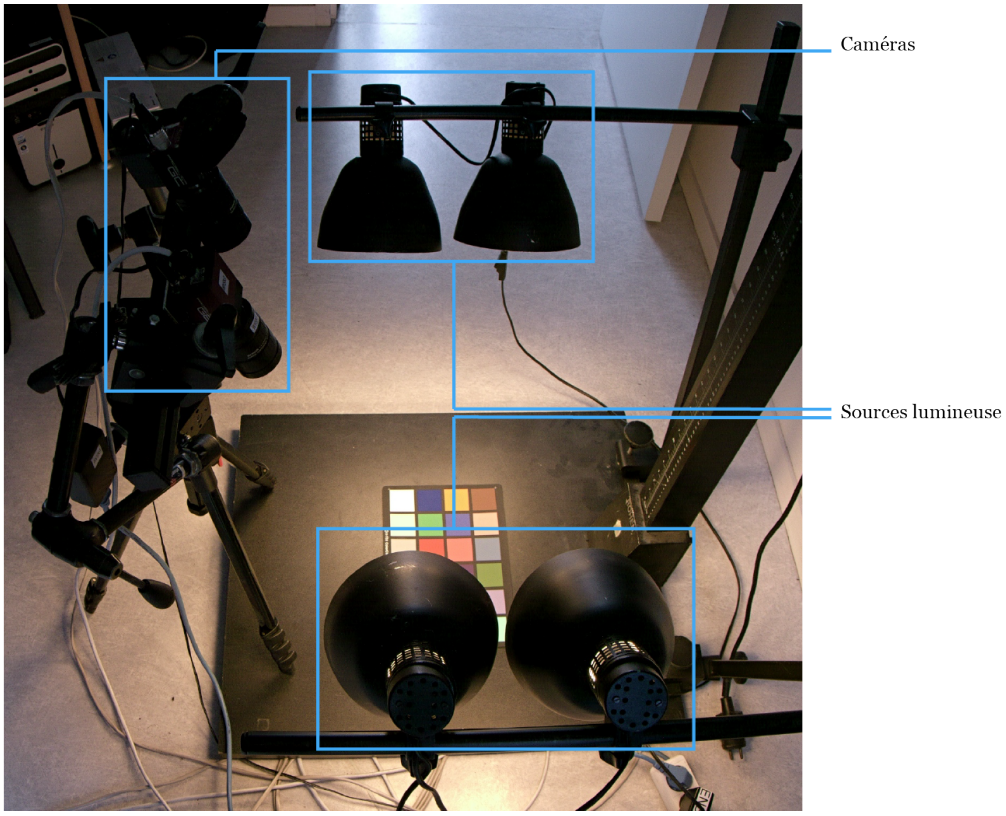


Figure 3.18 – Banc de calibrage de la balance des blancs.

3.3.3.1 Calibrage des caméras

Dans nos expérimentations, nous avons utilisé comme source de lumière celles dont est équipé notre banc de reprographie. Cette source de lumière, à base d'ampoules basse consommation, est donnée pour avoir une température de 2700°K (figure 3.18). Après application des relations précédentes, cette source de lumière a les rapports suivants entre ses canaux :

$$\begin{cases} \frac{R}{G} = \alpha.\frac{r}{g} = 1,82193588 \\ \frac{B}{G} = \beta.\frac{b}{g} = 0,35415524 \end{cases}$$

La suite de notre démarche repose sur le fait que si l'on considère que la balance des blancs de la caméra est neutre, c'est à dire ne déforme pas les rapports entre rouge, vert et bleu, la mesure de la couleur d'un élément gris éclairé par la source lumineuse précédente devrait donner les mêmes rapports entre ces trois canaux.
Par défaut, ce n'est a priori pas le cas. Notre méthode consiste à mesurer, sur les éléments gris de notre mire, les rapports $\frac{R}{G}$ et $\frac{B}{G}$ avec α et β égaux à 1. Sachant quelle doit être leur valeur réelle dans l'espace RGB doté d'un illuminant neutre, il nous est possible de déterminer les valeurs à attribuer à α et à β :

$$\alpha = \frac{G}{R}.1,82193588$$

$$\beta = \frac{G}{B}.0,35415524$$

Figure 3.19 – Mire ColorChecker : numérotation des différentes zones.

Les résultats, pour chacune des deux caméras, sont détaillés dans les tableaux de la figure 3.20. Afin d'éliminer les erreurs de mesure dues au bruit ainsi que les problèmes de saturation, l'ensemble des mesures ont été faites sur des images HDR. De plus, chaque mesure reportée dans le tableau correspond à la moyenne sur la zone de la mire considérée. Enfin, plusieurs itérations ont été faites pour affiner le résultat et s'assurer de la validité de la démarche.

Finalement, les paramètres se stabilisent après trois itérations sur les valeurs suivantes, arrondies du fait de limitations des caméras qui n'acceptent que des entiers pour paramètres de la balance des blancs :

	Caméra GC655C	Caméra GE1910C
α	1,68	2,16
β	1,37	1,21

Le calibrage de la sensibilité ISO a été fait dans les mêmes conditions d'éclairage, après avoir calibré chacune des caméras.Nous avons utilisé une mire colorimétrique de marque X-Rite, disposant d'un élément blanc diffus dont la réflectivité est donnée pour être de 95%. Le luxmètre et la mire sont disposés comme sur la figure 3.21. Le calibrage est fait par comparaison entre les valeurs relevées par le luxmètre et dont on peut déduire la luminance qui sera captée par les caméras, et les valeurs mesurées sur les images HDR.

Notre banc de mesure est composé des éléments cités précédemment (mire colorimétrique et luxmètre), des deux caméras que nous désirons calibrer, ainsi que d'une source lumineuse dont nous pouvons modifier les caractéristiques. Il s'agit d'un projecteur doté d'une lentille de Fresnel, permettant d'influer sur la concentration du flux lumineux. En outre la hauteur du projecteur par rapport à la scène peut être modifiée. En pratique les valeurs mesurées par le luxmètre vont de 5000 à 20000 lux.

Itération 1 - GC655C					
Zone de la mire	19	20	21	22	23
R	10,59	7,6557	4,5583	2,4834	1,3299
G	9,0235	6,1265	4,2398	2,2612	1,2303
B	2,3702	1,6283	1,1685	0,6294	0,3398
R/G	1,173602261	1,24960418	1,07512147	1,09826641	1,08095586
B/G	0,262669696	0,26577981	0,27560262	0,27834778	0,2761928
R%	160,4508831				
B%	130,3390038				

Itération 1 - GE1910C					
Zone de la mire	19	20	21	22	23
R	5,1382	3,5315	2,2511	1,2238	0,6432
G	5,6359	3,9917	2,4655	1,3242	0,7149
B	1,6065	1,1605	0,723	0,3867	0,2113
R/G	0,911691123	0,88471077	0,91303995	0,92418064	0,89970625
B/G	0,285047641	0,29072826	0,29324681	0,29202537	0,29556581
R%	200,9490139				
B%	121,5679875				

Figure 3.20 – Mesures et résultat pour l'obtention d'une balance des blancs neutre - R% et B% : rapports (en pourcentage) $\frac{R}{G}$ et $\frac{B}{G}$, respectivement.

La première étape du calibrage donne les résultats du premier tableau de la figure 3.22. En spécifiant comme sensibilités ISO des valeurs de départ "standard" , c'est à dire 80 ISO pour chacune des caméra (GC655C et GE1910C), on obtient les mesures de luminance rapportées dans le premier tableau de la figure 3.22. Si les valeurs sont pour l'instant fausses, on voit bien sur la figure 3.23 que l'écart en stops entre la valeur mesurée sur les HDRI et celles données par le luxmètre est constant pour chacune des caméras, indiquant que la méthode de calcul semble valable.
En modifiant les sensibilités des caméras de manière à rattraper ces écarts, c'est à dire en divisant celles d'origine par $2^{-1,24}$ et $2^{0,43}$ (respectivement pour la GE1910C et la GC665C), on trouve des sensibilités de 33 ISO et 107 ISO. L'amélioration de l'écart est visible sur la figure 3.23.

3.3.3.2 Validation du calibrage colorimétrique

Les figure 3.24 et 3.25 présentent les résultats de la comparaison entre les images capturées par les caméras calibrées (en luminance et en chromaticité) et la simulation de la même scène, l'éclairage étant toujours celui d'un banc de reproduction doté de lampes diffusant une lumière de température 2700°K. Les deux graphiques indiquent l'écart delta-E entre les couleurs capturées et la simulation, et le diagramme permet de comparer de visu les couleurs correspondantes. La figure 3.26 présente les captures d'où sont issus ces résultats

Pour commencer, nous pouvons remarquer que la moyenne du delta-E des deux caméras est sensiblement identique (4.4448 pour la GC655C contre 4.2967 pour la GE1910C). Par rapport aux standards de l'impression, cette valeur est tout à fait correcte mais cache quelques disparités.
Le calibrage a été fait en se basant sur les six éléments neutres, indicés de 19 à 24. Pour les deux caméras ces éléments ont un delta-E inférieur à la moyenne indiquant une bonne colorimétrie pour ceux-ci, ce que confirme le diagramme. Les différences observées sont à mettre en relation avec les différences notées dans les tableaux de la

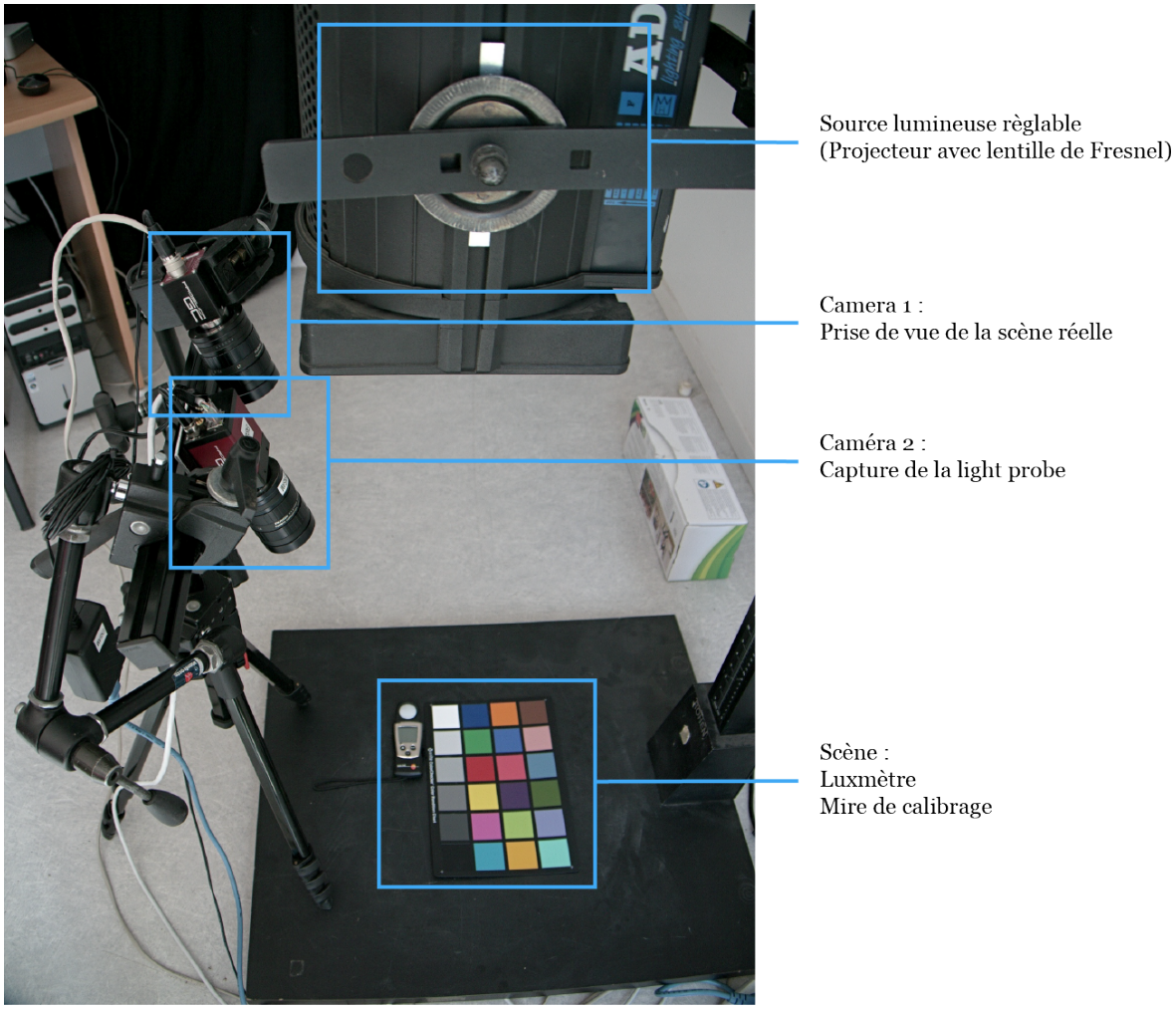


FIGURE 3.21 – Banc de mesure pour calibrage et validation de la chaîne de reproduction de la lumière

figure 3.20, où l'on voit clairement que les éléments n'imposent pas la même balance des blancs (rapports rouge/vert et bleu/vert), alors même qu'ils sont supposés être tous neutres. Les différences sont cependant minimes, et à mettre sur le compte à priori de la courbe de réponse des caméras qui ne semble pas totalement linéaire après correction.

Notons enfin que les deux caméras ont un calibrage qui est proche mais pas identique, avec un delta-E moyen entre les deux de 4.58 (voir figure 3.27). Cette différence a de grande chance d'influer sur les mesures de différence colorimétrique entre la capture par la caméra GE1910C de la mire et la simulation de celle-ci, l'éclairage utilisé pour la simulation étant issu de l'autre caméra.

3.3.3.3 Simulation de l'éclairage - luminance uniquement

La figure 3.28 présente les résultats de la comparaison de la simulation de l'éclairage (tel que capturé par la caméra GC655C équipée alors d'un objectif fish-eye) avec l'image issue de la capture HDR de la mire ColorChecker par la seconde caméra, dans les mêmes conditions d'éclairage. Ici, seule la luminance a été reproduite, sans prise en compte de

Calibration - GE1910C : 80 ISO - GC655C : 80 ISO						
N°	Mesure luxmètre (lux)	Estimation luminance Blanc 95% (cd/m2)	Mesure HDR GE1910C (cd/m2)	Différence (stops)	Mesure HDR GC655C (cd/m2)	Différence (stops)
1	5000	1511,971959	675	-1,163471977	2150	0,507905276
2	6000	1814,366351	781	-1,216071336	2530	0,479671595
3	7000	2116,760743	907	-1,222683755	2920	0,464110158
4	8000	2419,155135	1030	-1,231858952	3310	0,452327928
5	9000	2721,549527	1160	-1,230303485	3600	0,403568616
6	10000	3023,943919	1280	-1,240287574	4010	0,407170853
7	11000	3326,338311	1370	-1,279759015	4430	0,413371791
8	13000	3931,127094	1610	-1,287882319	5300	0,431049352
9	15000	4535,915878	1910	-1,247821246	5980	0,3987516
10	17000	5140,704662	2110	-1,284723131	6760	0,395057116
11	20000	6047,887837	2430	-1,31547507	7680	0,344674927
			Moyenne	-1,247303442		0,427059928
			Ecart type	0,042348463		0,04591152

Vérification - GE1910C : 33 ISO - GC655C : 107 ISO						
N°	Mesure luxmètre (lux)	Estimation luminance Blanc 95% (cd/m2)	Mesure HDR GE1910C (cd/m2)	Différence (stops)	Mesure HDR GC655C (cd/m2)	Différence (stops)
1	5000	1511,971959	1620	0,09956243	1590	0,07259538
2	10000	3023,943919	3110	0,0404832	3070	0,02180727
3	15000	4535,915878	4510	-0,0082665	4450	-0,0275885
4	20000	6047,887837	5970	-0,0187005	5840	-0,050463
			Moyenne	0,02826968		0,00408777
			Ecart type	0,05407682		0,05473101

Figure 3.22 – Deux étapes successives du calibrage des caméras - haut : valeurs avant calibrage (sensibilités estimées par comparaison avec un APN) - bas : valeurs après calibrage

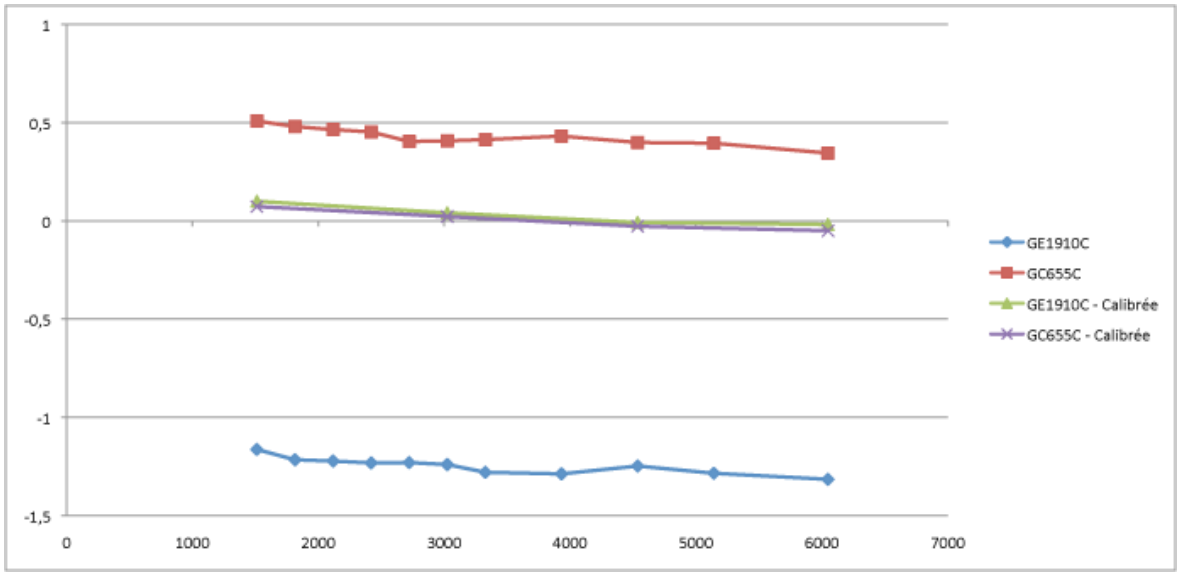


Figure 3.23 – Ecart des valeurs avant et après calibrage entre la mesure du luxmètre et les mesures sur les HDRI - Abscisse : radiosité (cd/m^2) - Ordonnées : Ecart avec le luxmètre (en stops)

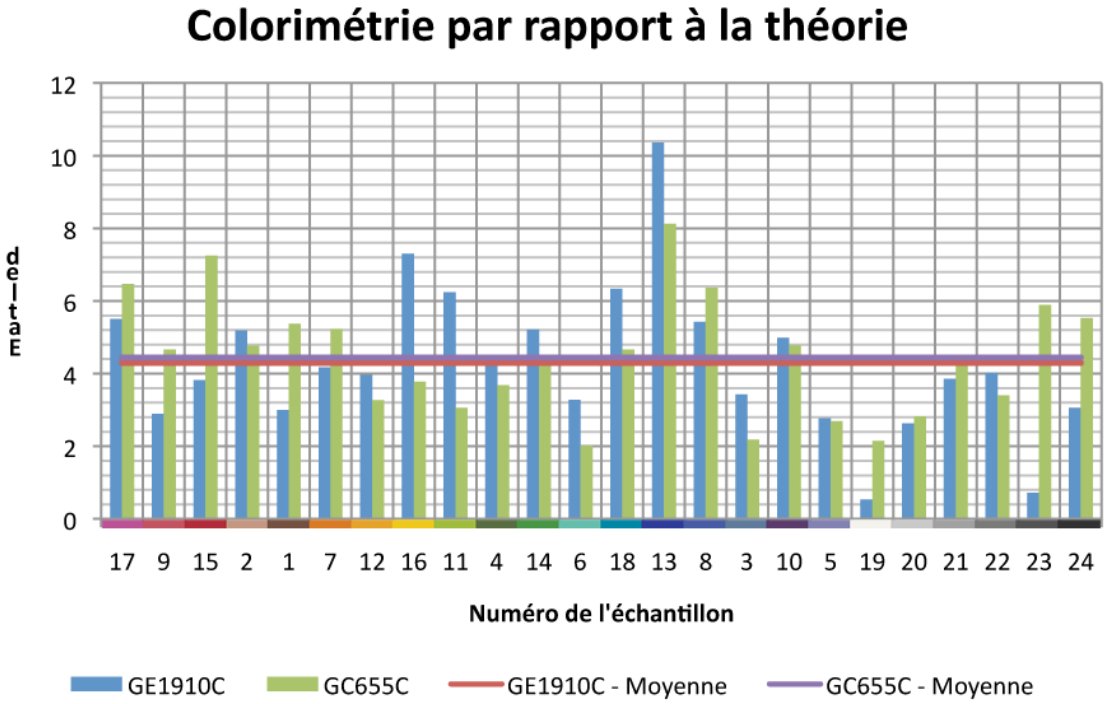


Figure 3.24 – Graphe représentant l'écart de couleur (delta-E 1994) entre les valeurs RGB théoriques et les mesures (ligne rouge : delta-E moyen)

la colorimétrie de l'éclairage. Celle-ci est évaluée dans la section suivante. Pour rappel, la sensibilité (et donc la précision de la mesure de la luminance) a été calibrée par rapport à l'élément 19 de la mire, c'est à dire le blanc (figure 3.19).

On peut séparer les configurations d'éclairage en deux groupes. Le premier groupe correspond aux éclairages n'ayant pas de direction prépondérante, c'est le cas de la configuration "Bureau - Naturel" et "Extérieur - Ombre". Le second correspond au contraire aux éclairages ayant une ou des directions d'intensité très importante par rapport aux autres, pour les configurations "Bureau - Banc" et "Extérieur - Soleil".

Commençons par les configurations d'éclairage diffus. La différence de luminance moyenne est de 0.4 stop pour la configuration "Bureau - Naturel", et 0.5 stop pour la configuration "Extérieur - Ombre". Pour comparaison, l'écart de luminance entre les éléments neutres successifs de la mire (indices 19 à 24) varie entre 1 et 1.5 stops. Cependant on peut observer des variations selon la couleur, en particulier les tons bleus pour lesquels l'erreur s'approche de 1 stop. L'erreur est également croissante pour les éléments neutres lorsque la réflectivité diminue : si elle est comprise entre 0.2 et 0.3 stop pour les quatre premiers éléments neutres, elle monte à 0.48 et 0.8 pour les deux derniers (respectivement).
Pour compléter cette observation, le graphe "Réflectance" (toujours dans la figure 3.28) indique la portion de lumière blanche réfléchie par chacun des éléments de la mire. On peut identifier une relation entre une réflectance élevée et une erreur faible, et inversement une luminosité faible et une erreur plus importante. Le calcul semble soutenir cette idée, avec une corrélation de -0.68 entre les résultats de la configuration "Bureau

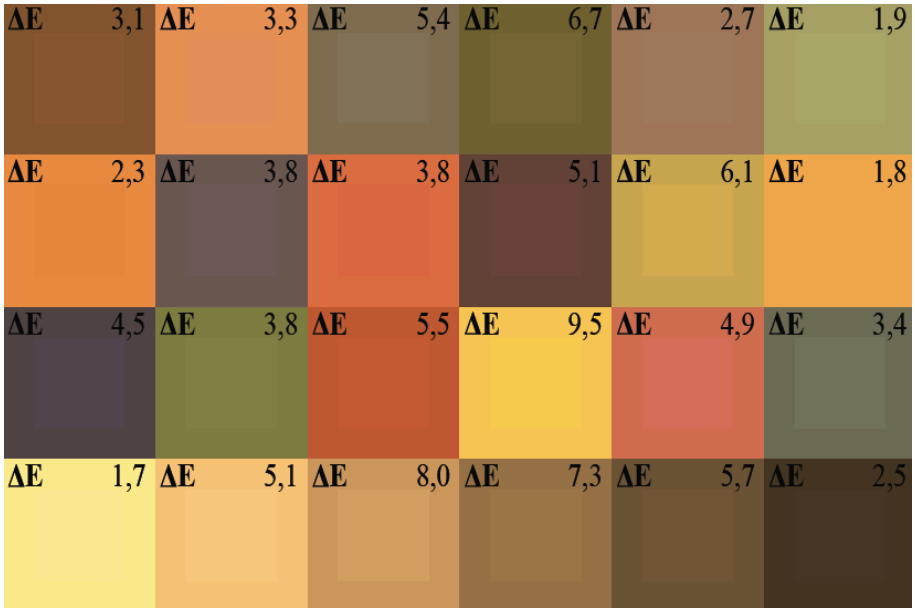


Figure 3.25 – Comparaison des colorimétries des deux caméras - extérieur de chaque case : GC655C, intérieur : GE1910C

Figure 3.26 – Captures pour la vérification du calibrage des deux caméras.

- Banc" et la luminosité des éléments, et -0,66 pour les résultats de la configuration "Extérieur - Soleil" et les mêmes luminosités (les valeurs étant négatives puisque les variables évoluent en sens inverse).

La configuration "Bureau - Banc" est la même que "Bureau - Naturel", hormis le fait que les lampes du banc de reproduction sont allumées. Celles-ci diffusent pour rappel un éclairage à 2700°K, celui-ci étant prédominant sur l'éclairage naturel de la pièce. Le graphe correspondant sur la figure 3.28 indique une erreur plus importante que pour les configurations d'éclairage naturel, avec une différence d'exposition moyenne de 1.1 stops entre la simulation et la capture HDR. Cette erreur plus importante peut s'expliquer par le fait que notre méthode considère toutes les sources lumineuses comme étant placées à l'infini par rapport à notre scène ce qui n'est pas le cas dans cette configuration.
La seconde configuration d'éclairage comprenant au moins une source prépondérante est celle nommée "Extérieur - Soleil". La configuration est identique à "Extérieur -

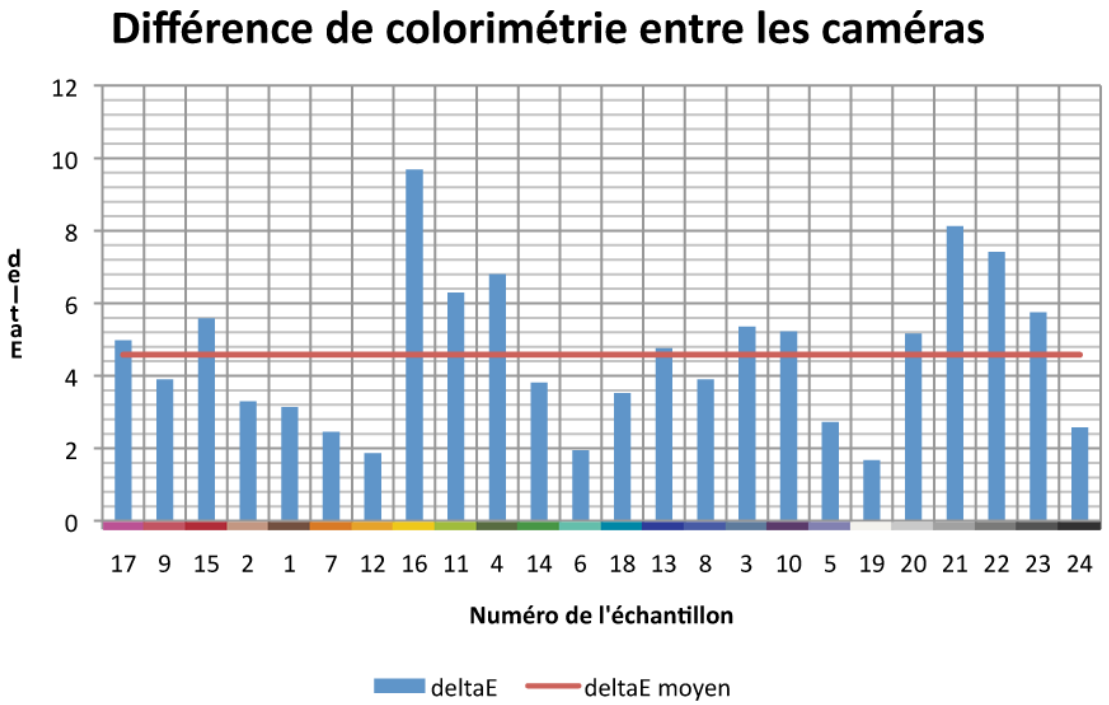


Figure 3.27 – Ecart de colorimétrie entre les deux caméras

Ombre" mais à un moment différent de la journée, où l'emplacement de la prise de vue n'était pas ombragé. Le soleil étant une source de lumière particulièrement puissante, nous avons été limité par notre caméra qui ne permet pas des temps d'exposition inférieurs à $1/100000^{eme}$ de seconde. Un filtre a donc été utilisé pour palier cette limitation, et la caméra a été recalibrée pour tenir compte de cette modification. Finalement, on peut observer que la source d'éclairage prépondérante (le soleil ici) est mieux reproduite que dans le cas où cette source est proche (cas de l'éclairage du banc de reproduction), et est du niveau de la reproduction d'un éclairage plus diffus.

3.3.3.4 Simulation de l'éclairage - luminance et colorimétrie

Dans cette section, nous nous intéresserons uniquement à la colorimétrie en omettant volontairement la reproduction de la luminance, cet aspect ayant été couvert dans la section précédente. La simulation de la colorimétrie de l'éclairage sera en revanche évaluée en comparaison avec la simulation de l'éclairage ne prenant en compte que la luminance de l'éclairage. Deux types de configurations seront étudiés ici : l'un en éclairage naturel, c'est à dire très proche d'un éclairage neutre ; l'autre en éclairage artificiel très coloré, qui est celui de notre banc de reproduction. Les captures et les simulations sont illustrées dans les figures 3.29 et 3.30, pour les configurations "Extérieur - Ombre" et "Bureau - Banc", respectivement.

Avant de passer aux mesures, nous pouvons anticiper quelques uns des résultats à partir de la connaissance des éclairages. L'éclairage naturel en pleine journée est modélisé de manière standard par l'illuminant D65, et a comme point blanc dans l'espace de chromaticité xy le point de coordonnée (0.31382, 0.33100). Ce point blanc est très

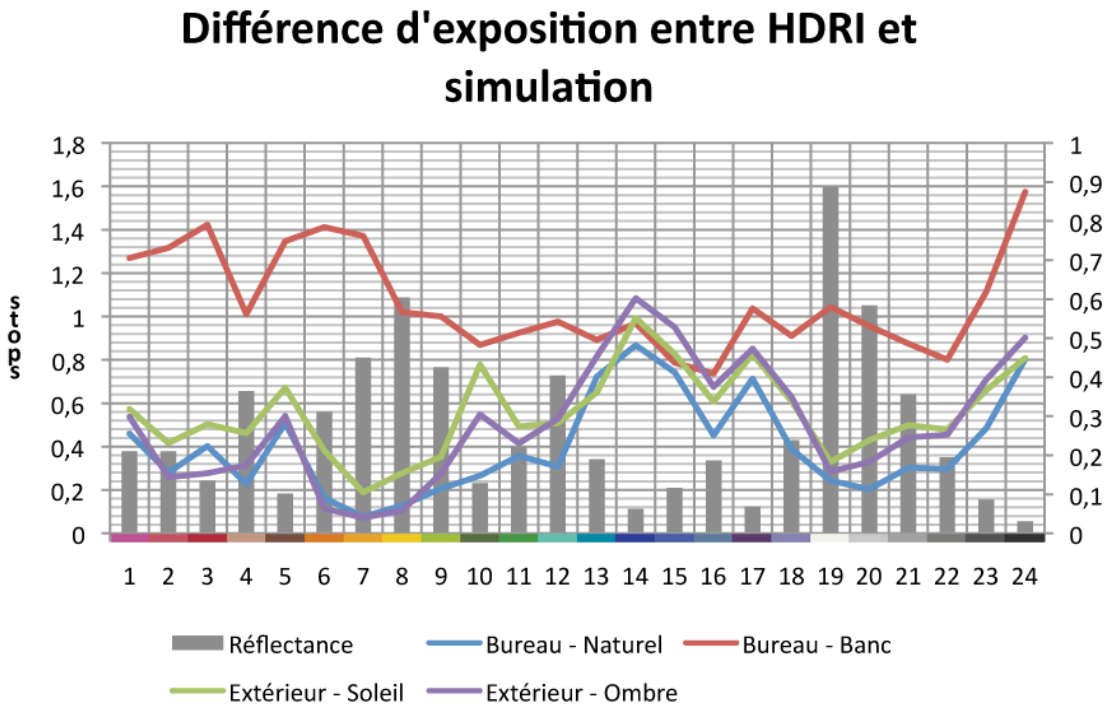


Figure 3.28 – Ecart de luminance entre la capture HDRI et la simulation

proche de l'illuminant neutre (nommé illuminant E) dont les coordonnées sont (1/3, 1/3). Or la simulation de l'éclairage ne prenant en compte que la luminance correspond à un éclairage avec un illuminant E. On peut donc s'attendre à ce que dans le cas d'un éclairage naturel, la différence de colorimétrie entre la simulation de la luminance seule et la simulation prenant en compte la colorimétrie soit minime.
En revanche, l'éclairage du banc de reproduction correspond à un illuminant de coordonnées (0.4593, 0.4106). La prise en compte de la colorimétrie de l'éclairage devrait donc dans ce cas apporter un gain significatif à la qualité de la simulation.

Les figures 3.31 et 3.32 présentent les résultats des simulations d'un éclairage naturel. Dans le cas de la configuration "Bureau - Naturel", hormis quelques légères variations les simulations avec et sans prise en charge de la colorimétrie de l'éclairage donnent des résultats très proches avec un delta-E de 8.8 pour la simulation de la luminance seule, et 9.0 pour la simulation complète. On peut remarquer cependant une différence de couleur plus importante dans les bleus que dans le reste du spectre. Cet aspect sera discuté plus loin.
La configuration "Extérieur - Ombre" présente un écart plus important entre les deux simulations. Le delta-E moyen de la simulation complète est de 8.3 quand celui de la simulation partielle est de 9.0. Un pic dans le delta-E est toujours présent au niveau des bleus.

La figure 3.33 décrit la qualité de la reproduction d'un éclairage artificiel. Comme nous l'avons vu dans la section précédente, la reproduction de la luminance est moins précise que dans les configurations mettant en scène un éclairage naturel. Cependant on peut voir que la colorimétrie est du même niveau avec un delta-E moyen de 8.5.

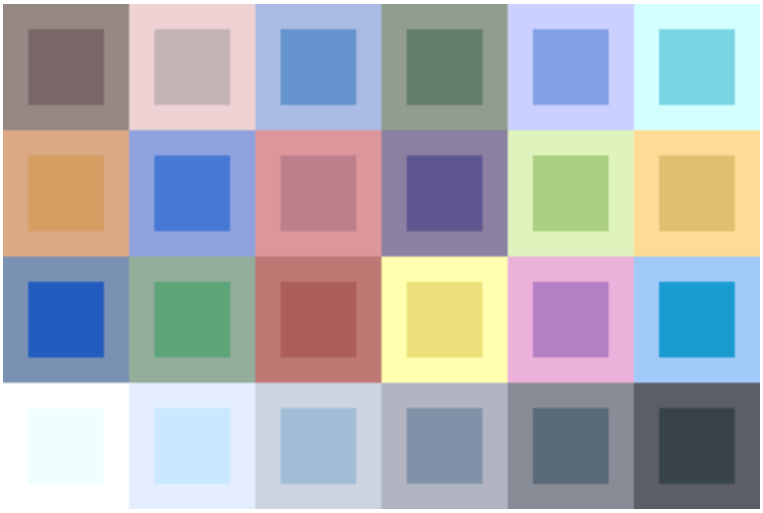


Figure 3.29 – Capture (extérieur) et simulation (intérieur) de la configuration "Extérieur - Ombre"

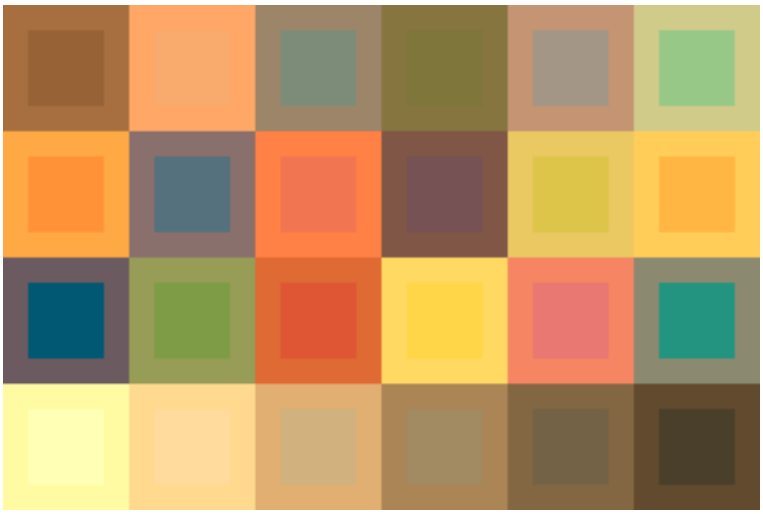


Figure 3.30 – Capture (extérieur) et simulation (intérieur) de la configuration "Bureau - Banc"

Cette configuration met en avant l'apport de la prise en compte de la colorimétrie de l'éclairage puisque la simulation complète est bien plus juste que la simulation de la luminance seule, celle-ci donnant un delta-E moyen de 21.5. Le pic au niveau des tons bleus est toujours présent, quelque soit la simulation considérée.

Pour revenir sur les pics observés dans les tons bleus, nous pouvons écarter les explications liées à une différence dans la mesure du delta-E, celle-ci étant faite de manière identique sur les captures HDR de la mire et sur la simulation. De même, le calibrage des caméras n'indiquait pas de différence plus importante entre les deux appareils au niveau des bleus par rapport au reste du spectre.
Trois pistes restent pour l'explication de ces pics. Tout d'abord, la première piste concerne la production de notre base de données d'images éclairées par des lumières directionnelles unitaires d'illuminant neutre, c'est à dire l'illuminant E. Pour le travail présenté ici, l'ensemble des images a été rendu avec le moteur de rendu non biaisé

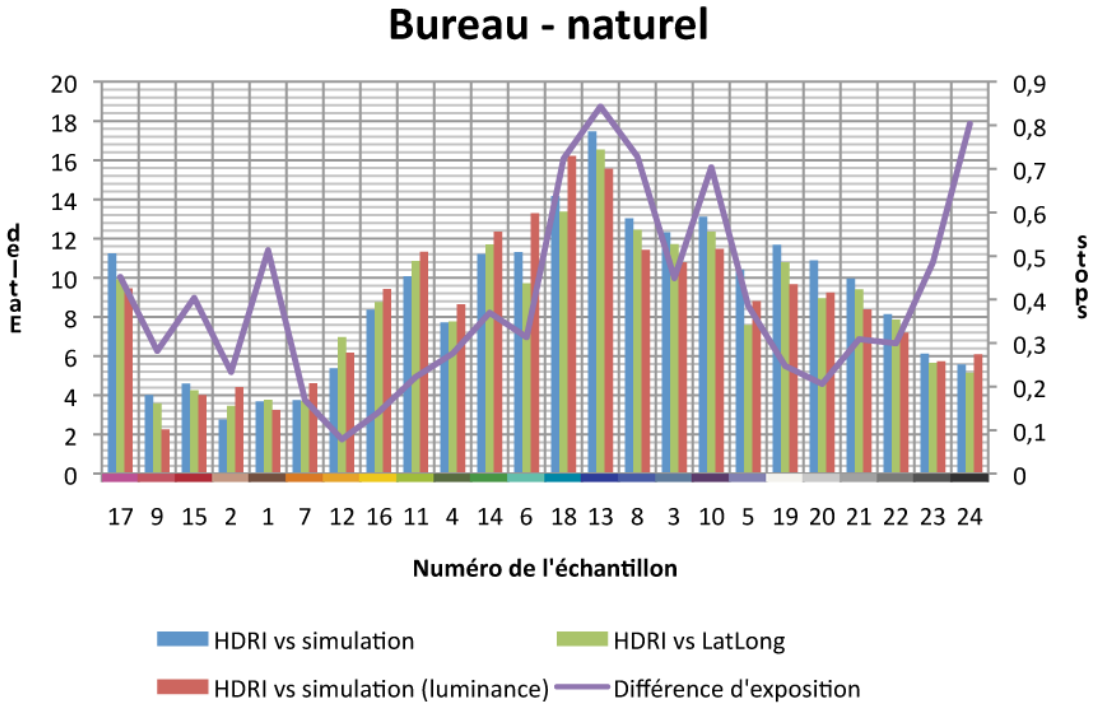


Figure 3.31 – Simulation de l'éclairage, configuration "Bureau - naturel"

LuxRender, lui même basé sur le moteur Pbrt. Une erreur colorimétrique dans le rendu de ces images se retrouverait évidemment dans la simulation finale. Mais comme on peut le voir sur la figure 3.34, il n'y a pas de pic particulier au niveau des tons bleus pour un rendu avec l'illuminant E.
Ensuite, il semble d'après les figures 3.31 3.32 3.33 que le pic puisse être lié aux erreurs de reproduction de la luminance. En effet, le delta-E prend en compte la différence de luminosité des deux couleurs comparées (et ce quelque soit la définition utilisée). Notre méthode de mesure de la différence entre deux couleurs tente de minimiser l'impact d'une éventuelle différence de luminosité en exposant les deux images sources de la même manière, mais elle ne permet pas de la mettre de coté totalement, notamment dans les cas où l'erreur de reproduction de la luminance varie trop fortement sur l'image simulée. Dans nos simulations, l'erreur peut varier sur une plage de 1 stop.

Enfin, la troisième piste se rapporte à la méthode de calibrage employée pour les deux caméras. Pour rappel, celles-ci sont calibrées avec une lumière très éloignée d'une lumière neutre, dont la distribution spectrale est représentée sur la figure 3.35. La distribution de la lumière émise par un corps noir à 2700°K est telle qu'elle contient très peu de lumière bleue, avec le risque de rendre le calibrage de cette partie du spectre approximative ce qui se voit notamment sur la figure 3.24 pour l'échantillon d'index 13. Cependant, il ne semble pas qu'il y ait eu d'erreur dans le calibrage puisque l'écart de couleur entre les deux caméras est faible selon la figure 3.27.

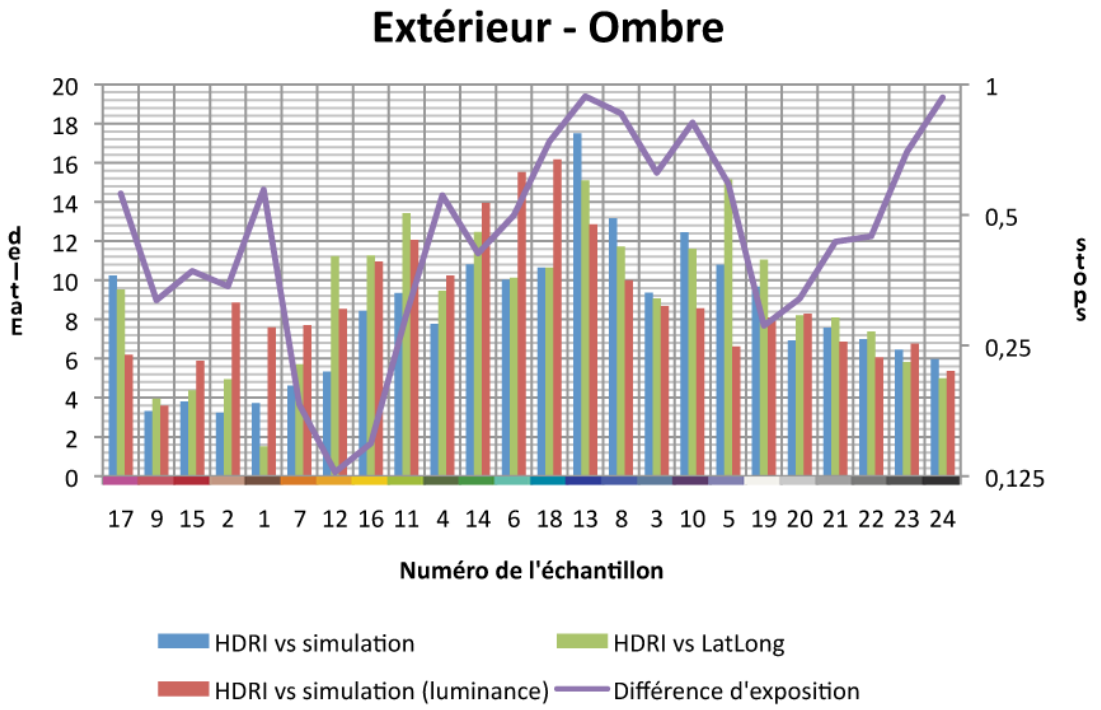


Figure 3.32 – Simulation de l'éclairage, configuration "Extérieur - Ombre"

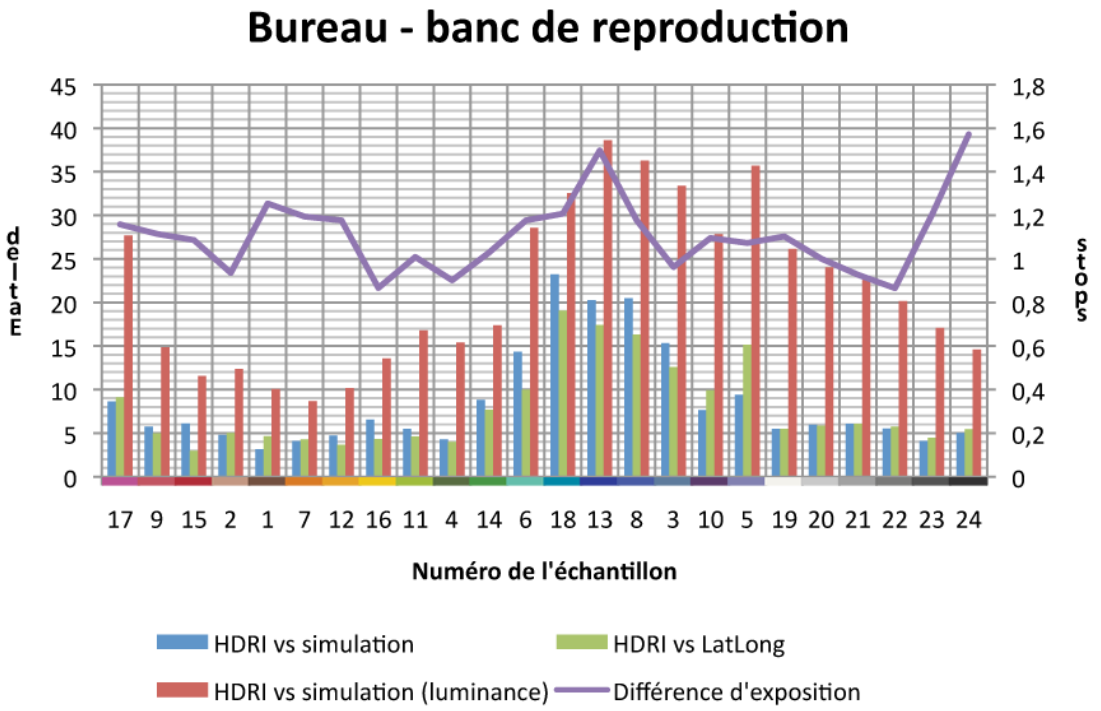


Figure 3.33 – Simulation de l'éclairage, configuration "Bureau - banc"

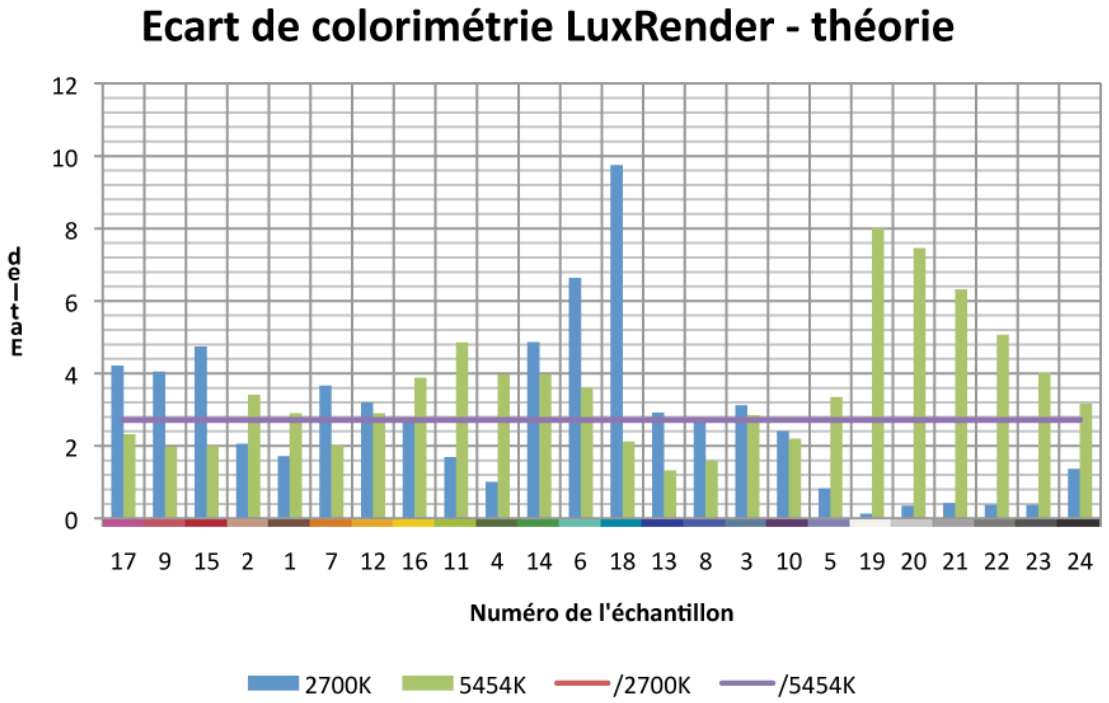


Figure 3.34 – Ecart de colorimétrie entre le rendu par LuxRender et la théorie, pour un éclairage à 2700°K (banc de reproduction) et 5454°K (Illuminant E)

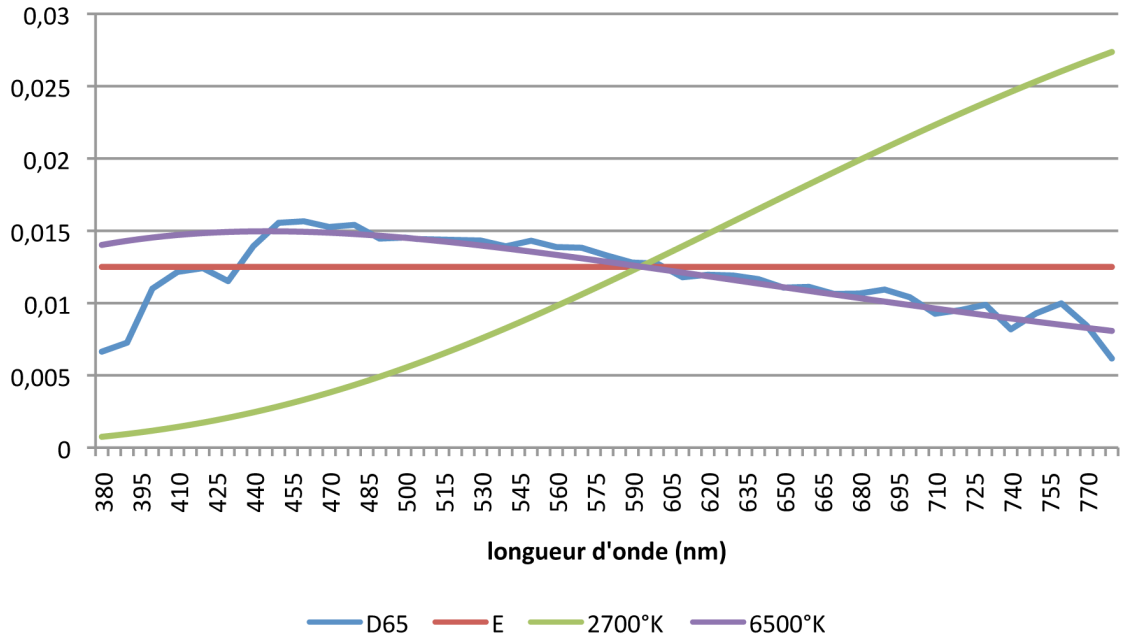


Figure 3.35 – Distribution spectrale des trois illuminants utilisés dans ce travail

3.3.4 Analyse complémentaire

Pour mieux déterminer quelles sont les sources des limitations de notre méthode de reproduction de l'éclairage, deux expérimentations supplémentaires ont été faites. La première a pour principale différence le choix d'une température de couleur différente pour la source de lumière utilisée lors du calibrage : une source à 6500°K a été choisie, produisant un illuminant proche du standard D65. Comme on peut le voir sur la figure 3.35, une telle source est beaucoup plus homogène dans sa distribution que la source utilisée plus haut.

La seconde expérimentation se base quant à elle sur l'introduction de profils colorimétriques, afin d'obtenir un calibrage plus fin. Les images sources sont les mêmes que pour les deux expérimentations précédentes mais sont corrigées selon le profil dédié à chaque caméra.

3.3.4.1 Calibrage avec une source à 6500°K

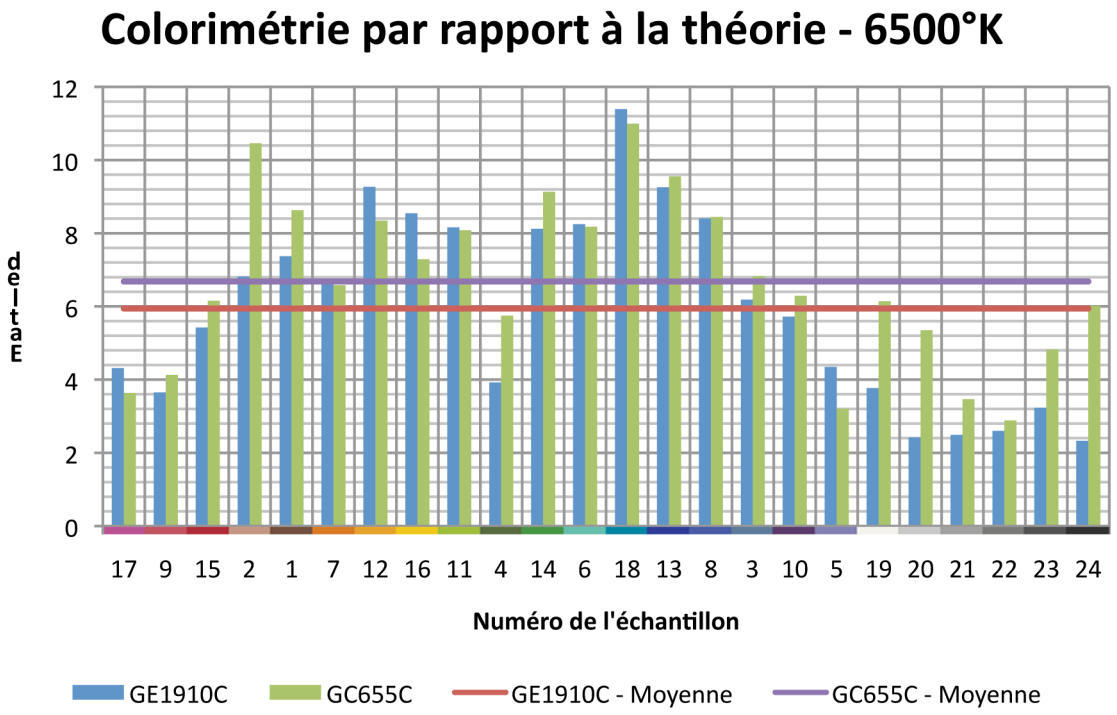


Figure 3.36 – Résultat du calibrage des caméras avec un illuminant à 6500°K

Le calibrage avec une source à 6500°K a été fait de la même manière que dans la section 3.2.1. Le résultat du calibrage est assez différent de celui obtenu avec un illuminant à 2700°K (voir 3.2.1), les coefficients étant les suivants :

	Caméra GC655C	Caméra GE1910C
α	1,36	1,78
β	2,03	1,70

La figure 3.36 présente le résultat du calibrage de nos deux caméras, c'est à dire l'écart de colorimétrie entre la théorie et la valeur mesurée. Malgré un calibrage légèrement moins bon que celui de notre précédente expérimentation (voir figure 3.37), l'écart de colorimétrie moyen donne un deltaE d'environ 6 et 6,7 ce qui reste correct. On peut dors et déjà remarquer la présence de pics pour les index 18 et dans une moindre mesure 13.

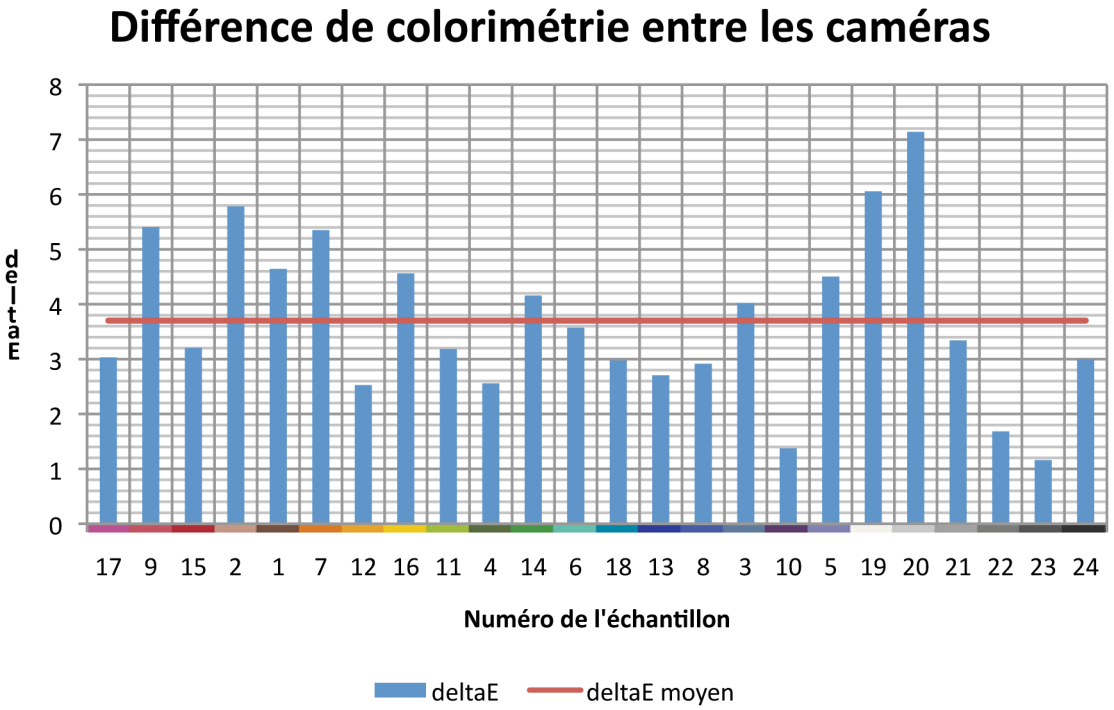


Figure 3.37 – Différence de colorimétrie entre les caméras après le calibrage avec un illuminant à 6500°K

D'autre part, remarquons que l'écart de colorimétrie entre les deux caméras est plus faible pour cette seconde expérimentation que pour la première (figure 3.37) avec un deltaE moyen de 3,7 contre 4,5. Cela confirme que la phase de calibrage a été faite correctement.

Deux nouvelles simulations de l'éclairage ont été faites, dans des conditions proches de celles déjà mises en place lors de l'expérimentation précédente. Les configurations sont nommées "Bureau - naturel" (éclairage capturé dans un bureau avec pour seule source l'éclairage naturel extérieur) et "Extérieur - nuageux". Les figures 3.39 et 3.40 présentent les résultats de ces simulations, ainsi que les résultats liés aux configurations équivalentes de l'expérimentation précédente ("Bureau - naturel" et "Extérieur - ombre", respectivement). La figure 3.38 montre une comparaison entre la simulation et la capture pour chacun de ces deux configurations.

Les résultats de ces deux simulations sont positifs. L'écart de colorimétrie est globalement meilleur avec un deltaE inférieur de 1 environ pour les deux cas par rapport aux résultats de la première expérimentation, et moins de disparité dans les résultats. Ceci est d'autant plus intéressant que le calibrage semble moins précis selon la

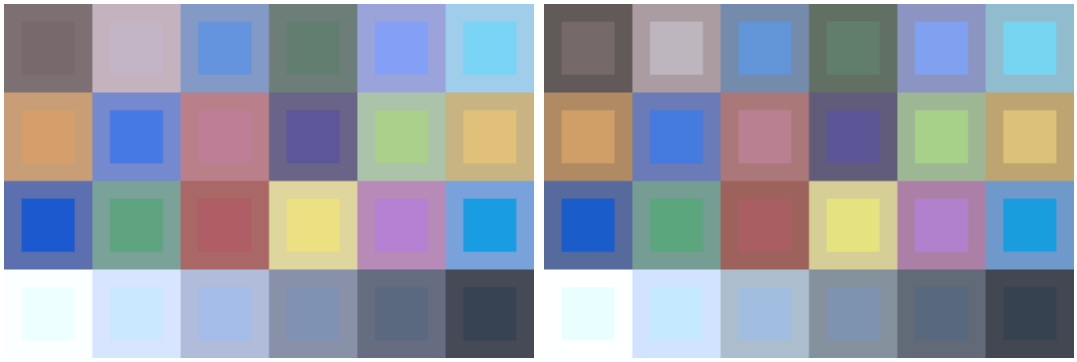


Figure 3.38 – Capture (extérieur) et simulation (intérieur) pour les configurations "Extérieur - nuageux" (gauche) et "Bureau - Naturel" (droite)

figure 3.36. Cependant, le pic au niveau de la couleur d'index 13 reste présent malgré une amélioration sensible pour ses voisins.

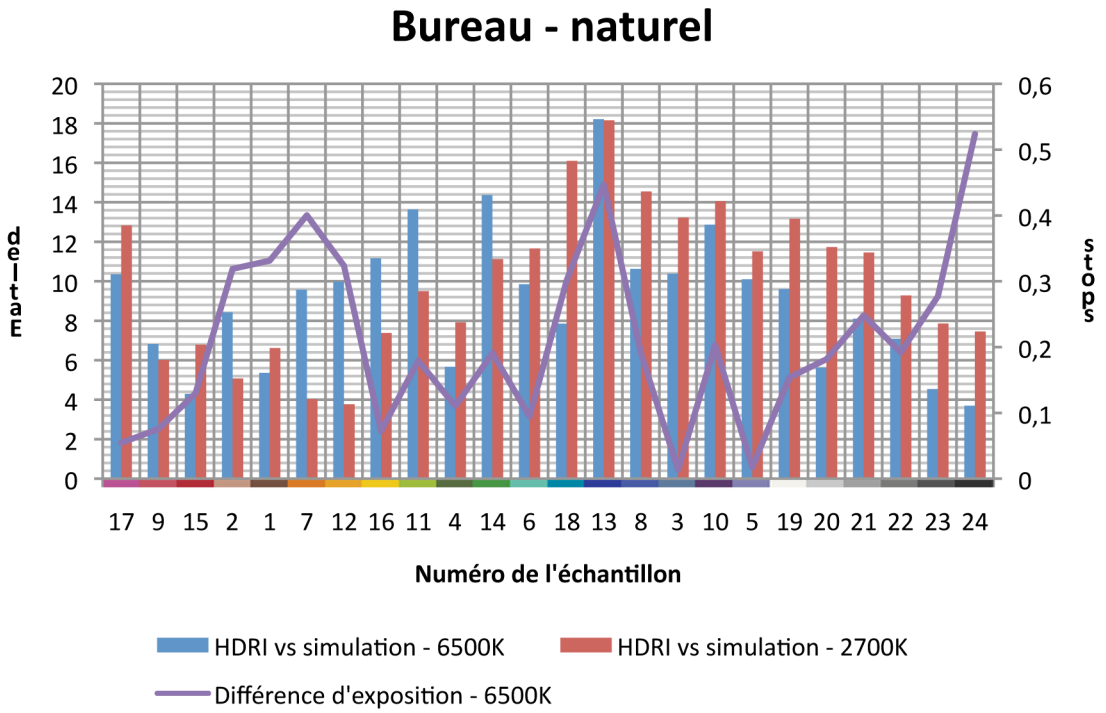


Figure 3.39 – Simulation de l'éclairage, configuration "Bureau - Naturel" après le calibrage avec un illuminant à 6500°K

Finalement, le calibrage en utilisant un illuminant plus uniforme est intéressant et utile puisqu'il améliore globalement la reproduction de l'éclairage par rapport au calibrage avec une source à 2700°K. Mais ce n'est pas une solution à notre problème d'écart colorimétrique très important au niveau de l'élément d'index 13 de notre mire de couleur.

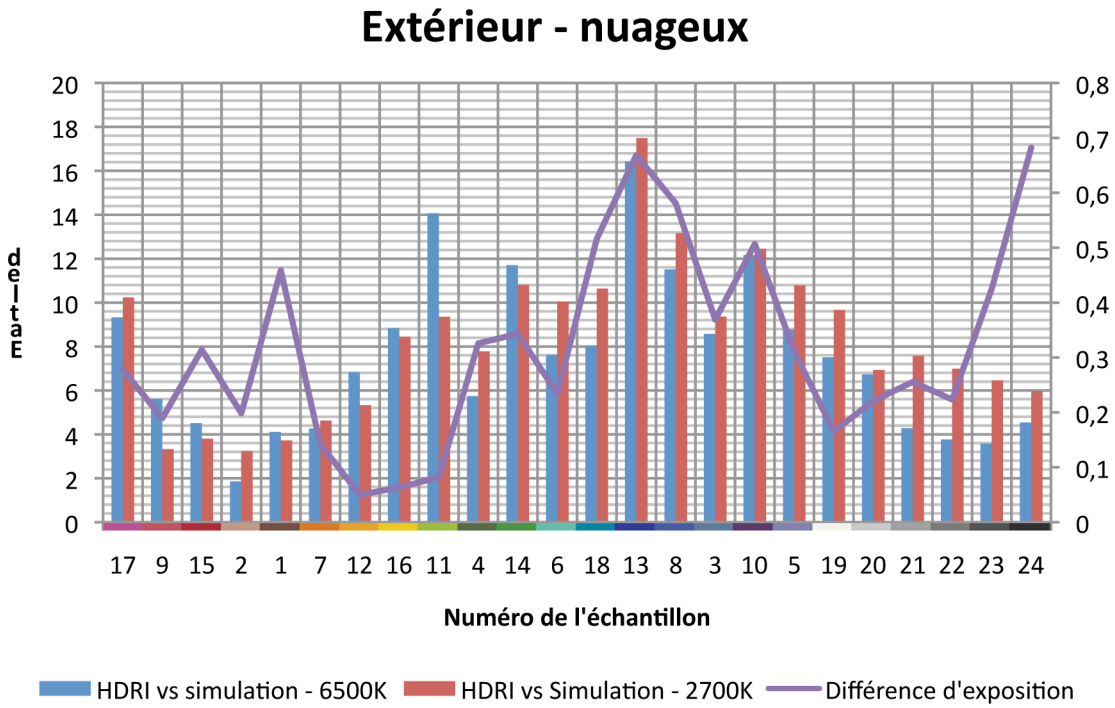


Figure 3.40 – Simulation de l'éclairage, configuration "Extérieur - Nuageux" après le calibrage avec un illuminant à 6500°K

3.3.4.2 Calibrage par l'intermédiaire de profils colorimétriques

Un profil colorimétrique (ou profil ICC, voir [1]) est ordinairement créé, pour un dispositif de captation, à partir de la capture d'une mire de couleur pré-calibrée comme c'est le cas de la mire que nous avons utilisé jusqu'ici. Son principal avantage par rapport à la méthode employée jusqu'à présent, c'est à dire une balance des blancs, est de considérer l'ensemble des éléments de la mire (neutres comme colorés) pour corriger l'espace colorimétrique de sortie du dispositif pour qu'il corresponde au mieux à l'espace défini par l'illuminant ayant servi au calibrage.

Nous avons dans notre cas utilisé les mêmes captures que lors des calibrages par la balance des blancs, et avons tiré pour chacun des caméras et dans chaque condition lumineuse un profil colorimétrique. Nous avons pour cela utilisé le système de gestion de couleur libre ArgyllCMS [37]. Ces profils ont par la suite été appliqués sur l'ensemble des images basse dynamique capturée, avant qu'elles ne soient combinées pour former les images haute dynamique sur lesquelles les mesures ont été faites. Les profils ICC ont été appliqués en utilisant la bibliothèque libre LittleCMS [69].

Le résultat du calibrage est présenté sur les figures 3.41 et 3.42. On peut noter que les colorimétries des deux caméras sont maintenant extrêmement proches, avec un deltaE valant 3.2 contre 3.7 et 4.5 pour les deux cas précédents. L'éléments 13 est toujours problématique, malgré une différence à peine perceptible à l'oeil sur un écran calibré.

L'utilisation des caméras calibrées de la sorte donne finalement les résultats du tableau 3.43. Par rapport au calibrage simple à 6500°K, le deltaE moyen entre la

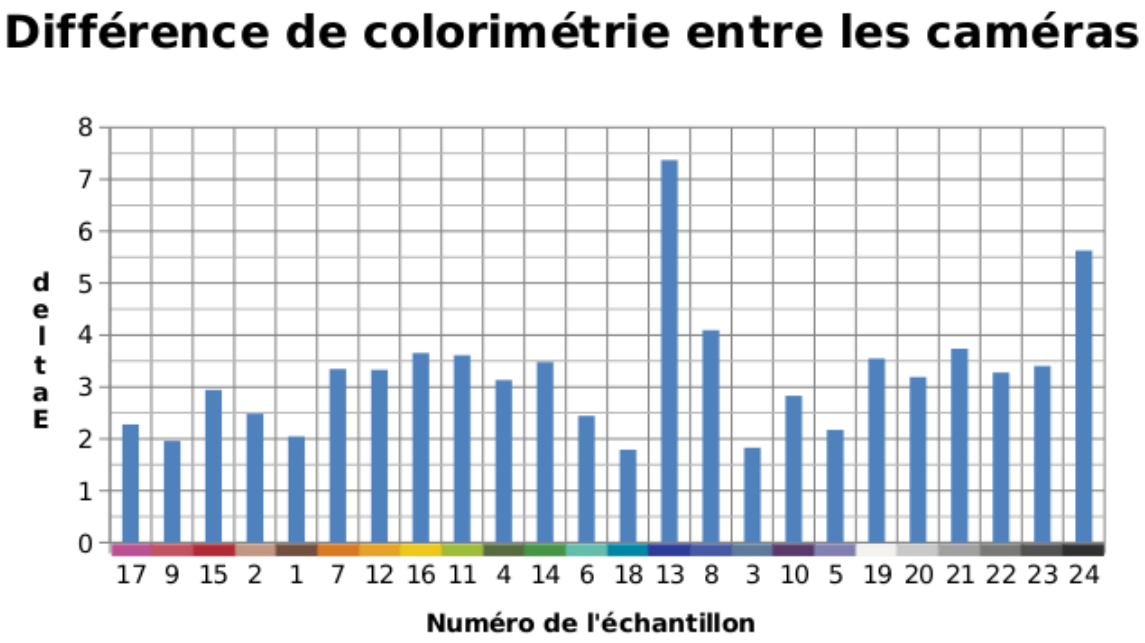


Figure 3.41 – Ecart de colorimétrie entre les deux caméras, après calibrage à 6500°K avec un profil ICC

simulation et la capture réelle est ici de 7.3 contre 9.1 pour la scène "Bureau - Naturel", et de 6.3 contre 7.7 pour la scène "Extérieur - Nuageux". L'intérêt de l'utilisation de profils colorimétriques est donc assez évident, malgré la persistance de l'erreur au niveau de l'élément 13.

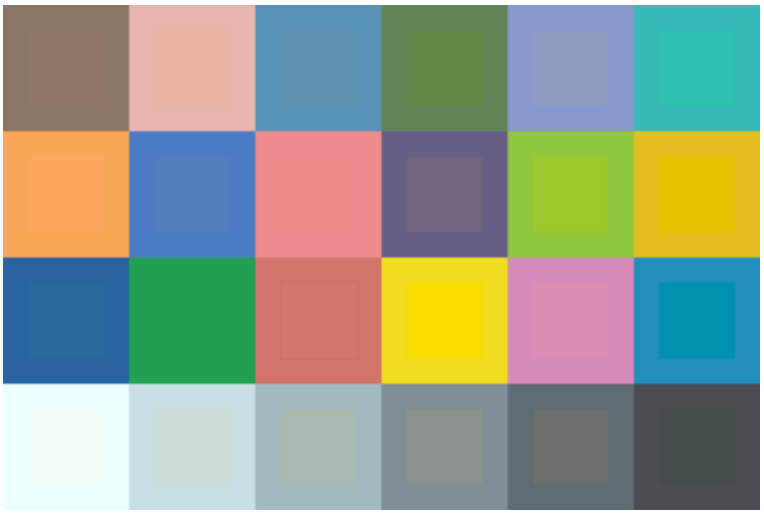


Figure 3.42 – Comparaison de la colorimétrie des deux caméras - extérieur : GC655C, intérieur : GE1910C

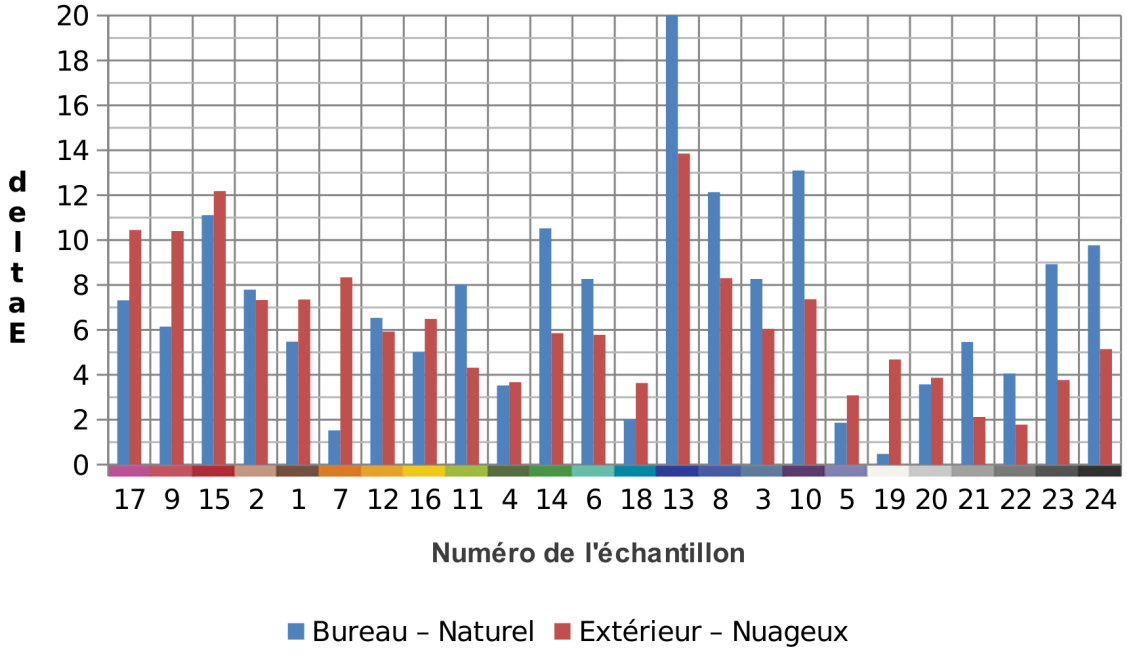


Figure 3.43 – Différence de colorimétrie entre la simulation et la capture HDRI, pour un calibrage à 6500°K avec des profils colorimétriques

3.4 Discussion

Nous proposons dans ce chapitre une chaîne de reproduction de l'éclairage réel sur des objets virtuels en mettant l'accent sur la justesse photométrique. Nos différents essais montrent que cette chaîne permet de reproduire le même ordre de grandeur d'intensité lumineuse entre un objet réel et sa simulation, et d'en approcher la colorimétrie.

Nous avons cependant identifié quelques biais qu'il faudra corriger pour que la reproduction soit plus juste. La principale piste est de changer de source lumineuse pour le calibrage. Nous nous sommes en effet basés pour notre première expérimentation sur un éclairage basse consommation pour celui-ci, qui présente l'inconvénient d'être peu puissant dans la partie bleue du spectre.

Le changement de source lumineuse, remplacé par une source plus équilibré, est très bénéfique à la qualité de la reproduction finale comme le montrent les expérimentations. L'utilisation de profils colorimétriques améliorent encore les résultats : les couleurs mesurées par les caméras sont alors extrêmement proches (voir figures 3.41. Malgré cela, la reproduction souffre toujours d'écarts de colorimétrie pouvant se révéler important bien que la moyenne soit proche du seuil admis comme délimitant le niveau de distinction de l'oeil humain, pour deux éléments éloignés.

Pour conclure cette section, la figure 3.44 montre l'intégration du lapin de Stanford [9] dans deux scènes à l'éclairage très différent selon notre méthode. La première image est issue de captures sous éclairage artificiel à 2700°K, la seconde de captures en lumière naturelle, en extérieur. L'ombre au sol a été reproduite en simulant un support dans la scène virtuelle, puis en faisant le rendu du support seul afin de pouvoir extraire l'ombre indépendemment du modèle du lapin.

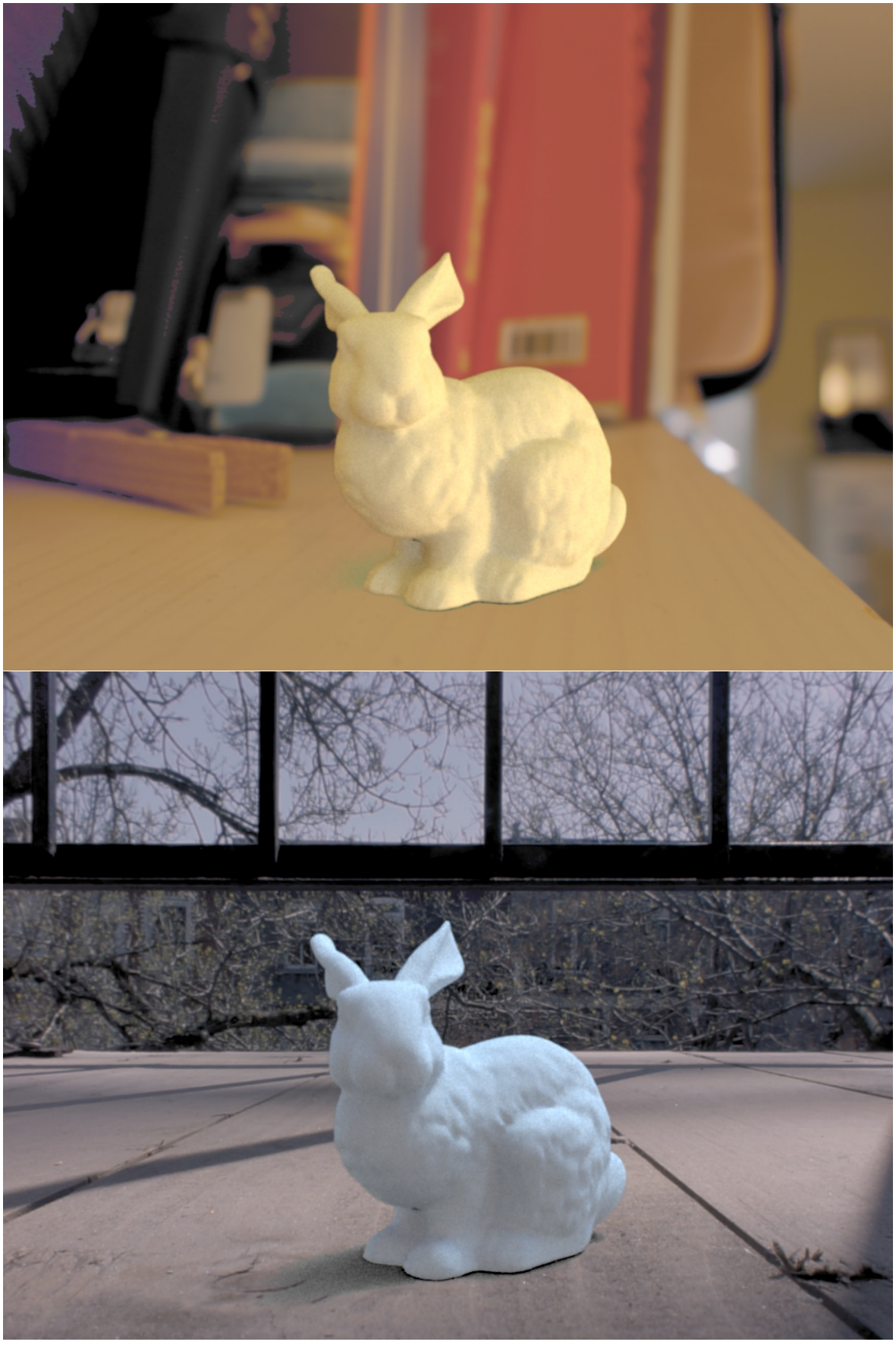


Figure 3.44 – Exemples de reproduction de l'éclairage à l'aide de notre méthode, sous deux conditions lumineuses différentes.

Chapitre 4

Segmentation 2.5D

4.1 Problématique

La segmentation d'image est une branche de la vision par ordinateur qui reste une affaire de compromis. Les méthodes connues pour appréhender ce problème sont le plus souvent le résultat d'un équilibre entre la vitesse d'exécution, la précision du résultat et la résilience vis à vis des changements de la scène d'origine.

4.1.1 Problématique scientifique

Dans le cadre de notre travail, c'est à dire en supposant l'utilisation de la segmentation d'image pour le dispositif ray-on, la principale contrainte est que la segmentation doit être fonctionnelle tout au long de la journée, quelle que soit l'environnement lumineux. En contrepartie la géométrie du lieu est supposée figée puisqu'il s'agit de monuments historiques. Nous devons donc établir une méthode de segmentation particulièrement permissive quant aux variations lumineuses et pouvant fonctionner en temps réel.

4.1.2 Question de recherche

La question de recherche étudiée dans cette section est la suivante : *Comment segmenter un flux vidéo à l'éclairage changeant en temps réel, en maintenant un niveau de qualité constant quelque soit l'éclairage ?*

4.2 Modélisation et état de l'art

4.2.1 Segmentation d'image

Par définition, la segmentation d'image consiste à détecter et à isoler les différents éléments d'une image, pour une utilisation ultérieure de ces éléments. Si certaines approches envisagent de détourer les éléments sans introduire de classification particulière entre eux (référence nécessaire ici), nous nous concentrerons ici sur les techniques discriminant l'avant plan (ou FG pour foreground) de l'arrière plan (BG, background).

Les applications de cette segmentation FG/BG sont multiples, depuis la surveillance de trafic automobile ou de parking pour les plus anciennes aux implémentations en réalité mixte.
L'approche la plus évidente de segmentation BG/FG consiste en une simple soustraction entre une image de l'arrière plan "vide" et une image de l'arrière plan agrémenté d'éléments d'avant plan. Cet algorithme basique inaugurait l'architecture qui est toujours utilisée aujourd'hui dans le cadre de la segmentation, à quelques variations près :

- Initialisation : Création du modèle caractérisant l'arrière plan.
- Captation : Mesure d'une nouvelle image de la scène, comprenant FG et BG.
- Catégorisation : Classement de chaque pixel de la nouvelle image mesurée entre FG, BG, et indéterminé.
- Filtrage : Détermination du label des pixels indéterminés en fonction de leur voisinage, correction des erreurs.

L'approche la plus simple consiste à prendre une photo de la scène considérée à un instant t, puis à soustraire cette image à toute image prise ultérieurement ce qui fournit un masque qui discrimine de manière plus ou moins fidèle l'arrière plan de l'avant plan. Cette approche comporte de nombreux défauts, qui ont mené à la création d'améliorations visant à les compenser voire à les annuler. Quelques exemples de ces limitations :

- L'image "source" prise à l'instant t n'est pas nécessairement exempte d'éléments mobiles n'appartenant pas à l'arrière plan.
- Les arrières plans mobiles (tels que les arbres) seront considérés comme des éléments d'avant plan.
- Tout changement lumineux impose d'utiliser une nouvelle image de l'arrière plan prise dans ces mêmes conditions lumineuses.
- Si un élément de l'avant plan a une couleur proche de l'arrière plan, il sera considéré comme appartenant à l'arrière plan.

Figure 4.1 – Exemple de segmentation approximative du fait d'un léger changement d'éclairage de la scène

4.2.1.1 Modélisation de l'arrière plan

Des modèles plus évolués de représentation de l'arrière plan ont été développés pour améliorer le résultat de la segmentation. Outre le changement d'espace colorimétrique, passant de l'espace sRGB communément utilisé à un espace séparant la chromacité de la luminance par exemple (L*a*b, HSV, HSL, ...) pour réduire la sensibilité aux changements d'éclairage, la grande majorité des modélisations utilisées actuellement prennent en considération la variabilité de la mesure d'un élément, en modélisant sa couleur par une gaussienne par exemple [47]. Typiquement, une gaussienne est associée

à chaque pixel de l'arrière plan afin de le décrire. Cette approche est dite "paramétrique" puisqu'il s'agira de déterminer, à partir d'une séquence d'image de l'arrière plan, les paramètres de la gaussienne c'est à dire sa moyenne et son écart type. Selon le type de données considéré, ces paramètres pourront être multi-dimensionnels (trois dimensions dans l'espace sRGB par exemple). Dans le cas présent de la modélisation par une gaussienne, la détermination de ses paramètres est simple et pourra éventuellement être améliorée par la détection des éléments fortement éloignés de la moyenne.

Dans de nombreux cas, une gaussienne unique n'est pas suffisante pour décrire les états que peut prendre un pixel. C'est le cas notamment lorsque l'arrière plan est mobile. Pour prendre en compte ces différents états a été introduite la modélisation par un mélange de gaussiennes [93, 50, 24, 109, 95, 108, 62], chaque état se retrouvant représenté par une gaussienne. Pour obtenir ce mélange de gaussiennes, l'approche initiale était basée sur l'association de l'algorithme K-means et d'un algorithme de maximisation de l'espérance (*Expectation Maximization*, ou EM). Le premier permet de séparer une population en paquets d'éléments proches, tandis que le second va améliorer cette séparation et la compléter avec l'écart-type de chaque paquet (et donc de chaque gaussienne). L'inconvénient de cette méthode est principalement que le résulat final dépend grandement du nombre de paquets, qu'il est nécessaire de spécifier à l'avance [93, 50] [108]. Une méthode capable de déterminer par elle-même le nombre de modes présents dans la population est maintenant régulièrement utilisée, il s'agit de l'algorithme dit de décalage de la moyenne (*Mean shift*) [62].

Parallèlement à ces méthodes dites paramétriques, des méthodes non-paramétriques ont été étudiées. La plus répandue est basée sur l'utilisation de noyaux gaussiens (*gaussian kernel*) [31, 21, 77]. Cette méthode peut être considérée comme une extension de la modélisation par histogramme, à savoir que chaque réalisation d'un pixel contribue de manière égale à la modélisation globale. Dans le cas d'un histogramme, si la réalisation n du pixel a pour valeur x, elle participera uniquement à la valeur x de l'histogramme. En revanche pour les noyaux gaussiens, cette réalisation de valeur x participera à la valeur x de l'histogramme ainsi qu'à ses voisins, selon une gaussienne dont l'écart-type est soit fixé quelque soit la valeur x, soit variable [73].
L'avantage par rapport à une modélisation par histogramme est de pouvoir prendre en compte le bruit de mesure de la caméra et éventuellement de limiter l'impact du fait que l'arrière plan n'aurait pas présenté certaines valeurs durant la période de calibrage. Par rapport à un mélange de gaussienne, l'utilisation de noyaux gaussiens permet une modélisation plus proche de l'observation mais moins aisée à manipuler.

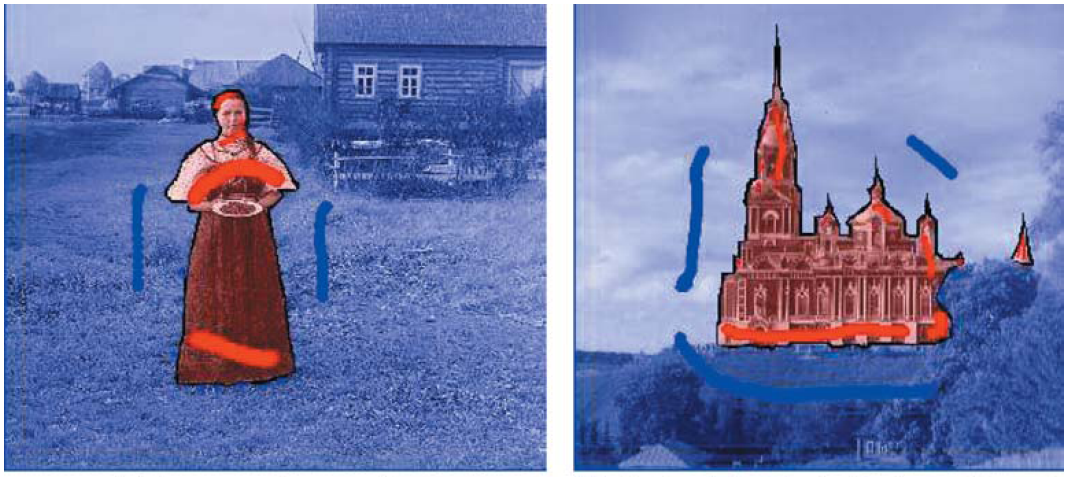
En complément de l'information de couleur, certaines méthodes proposent de considérer l'environnement des pixels par l'intermédiaire de la texture à un niveau local [80] ou les contours environnants [96].
De la même manière, la connaissance de la carte de profondeur (donnant accès à une représentation tridimensionnelle de la scène) s'est révélée être très intéressante. Si elle peut être utilisée comme un simple canal supplémentaire aux canaux représentant la couleur [104, 46, 45, 39], Leens et al. [63] proposent de faire une segmentation à part entière depuis cette carte pour ensuite la comparer à d'autres sources afin de profiter des avantages de chaque modalité. La source de cette carte de profondeur est soit une z-camera dédiée [104, 23] [13], soit un couple de caméras en stéréo [46, 45, 39].
Une variante des méthodes basées sur la reconstruction stéréo consiste à utiliser les

images issues de plusieurs caméras et d'en comparer les captures, comme décrit par Ivanov et al. [49]. Dans leur implémentation, les pixels des caméras sont associés grâce à un laser projeté sur l'arrière plan et identifié sur chacune des vues. De cette association, une correspondance dans la couleur de chaque élément est obtenue. Par la suite, une modification de cette correspondance apportera une indication qu'un objet de l'avant plan se trouve dans la scène.
Des informations peuvent également être extraites des séquences d'image. Mittal et al. [73] étudient la probabilité qu'un élément de la scène représenté par un pixel ait bougé, en prenant en considération les changements éventuels d'appartenance (à l'arrière plan ou l'avant plan) de ce pixel. Xiao et al. [106] étudient quant à eux des scènes fixent dans lesquelles la caméra se déplace, ce qui permet de se ramener à un problème de mise en correspondance de différentes vues (à la manière de la reconstruction stéréo). Bugeau et al. [18] vont eux utiliser le flux optique, selon l'algorithme de Kanade et Lucas (voir algorithme KLT, pour Kanade-Lucas-Tomasi), pour suivre certains éléments facilement repérables de l'image. De ce flux optique sera déterminée l'appartenance des éléments à l'avant ou à l'arrière plan, appartenance qui sera alors étendue aux éléments environnant.

La modélisation sur la base des informations précédentes ne permet cependant d'envisager la segmentation que dans le cadre de la configuration de l'arrière plan tel qu'il était au moment de cette modélisation. Si cette configuration change, du fait par exemple d'un changement d'éclairage ou du déplacement d'un objet de l'arrière plan, des erreurs de segmentations peuvent apparaître (selon la méthode de segmentation utilisée). Il est donc nécessaire de mettre à jour le modèle de l'arrière plan.
Vijverberg et al. [102] proposent une méthode de compensation des changements d'éclairage en l'évaluant sur l'ensemble de l'image à chaque itération de la segmentation. Ko et al. [57] modélisent quant à eux l'arrière plan comme une succession de couches qui peuvent s'occulter les unes les autres selon leurs mouvements respectifs, ce qui permet de prendre en charge les arrière plans mobiles

Enfin, on peut noter une dernière source d'information : l'intervention d'un utilisateur, qui spécifiera selon une précision variable (selon l'algorithme) ce qui appartient à l'avant plan et ce qui appartient à l'arrière plan. Rother et al. [86], Lempitsky et al. [64] et Liu et al. [67] demandent à l'utilisateur de spécifier une boîte autour de l'objet à segmenter, tandis que Duchenne et al. [30] et Vicente et al. [101] proposent de tracer des zones (remplies) sur l'arrière plan et sur l'avant plan. Les résultats de la segmentation avec ces méthodes sont globalement meilleurs (moins d'erreur de classement) que les méthodes entièrement automatiques.

Cependant, certaines approchent tentent de remplacer l'intervention de l'utilisateur en simulant son comportement. Fu et al. [35] utilisent pour ce faire une carte de saillance, qui simule l'attention portée par la vision humaine sur une image et donc les zones d'intérêt de celle-ci. D'autres méthodes proposent d'utiliser un filtre de Kalman pour mettre à jour la modélisation (Stauffer et al. [93]). Shoushtarian et al. [91] adaptent la modélisation de chaque pixel de l'arrière plan s'il n'est pas considéré comme occulté par un objet de l'avant plan ce qui permet de ne pas incorporer au modèle des objets de l'avant plan restés trop longtemps immobiles.



(a) A woman from a village (b) A church in Mozhaisk (near Moscow)

FIGURE 4.2 – Segmentation après intervention de l'utilisateur [14]

4.2.1.2 Obtention des coûts d'appartenance à partir des informations de couleur

La modélisation obtenue précédemment n'est pas utilisable telle quelle pour déterminer la segmentation. Ce modèle doit être traité pour en tirer, pour chaque pixel, des coûts d'appartenance à l'avant ou à l'arrière-plan. Une méthode pour obtenir ces coûts est celle utilisée par Rother et al. [86] qui est basée basé sur l'utilisation de mélanges de gaussiennes. Voyons comment en tirer une représentation adaptée à un algorithme de segmentation.

La forme générique des gaussiennes est la suivante :

$$\phi(x) = \frac{1}{(2*\pi)^{m/2}.|\vec{\sigma}|^{1/2}}.e^{-\frac{1}{2}(\vec{x}-\vec{\mu}).\vec{\sigma}^{-1}.(\vec{x}-\vec{\mu})^T}$$

où $\vec{x} = (x_1, ..., x_m)$ est le vecteur décrivant une observation dans l'espace considéré de dimensions m (dans lequel est représenté chaque pixel), $\vec{\mu} = (\mu_1, ..., \mu_m)$ le vecteur moyen de la gaussienne et $\vec{\sigma}$ est la matrice de covariance définie positive de dimensions $m \times m$.
La modélisation par un mélange de gaussiennes (ou *Gaussian Mixture Model*, GMM) est une approche paramétrique de l'utilisation de gaussiennes pour représenter une population : il s'agit de rechercher pour un nombre K de gaussiennes composant le mélange les paramètres $\vec{\mu}$ et $\vec{\sigma}$ de celles-ci.
Nommons χ_{BG} et χ_{FG} les graines de l'arrière et de l'avant plan :

$$\chi_{BG} = \{\vec{x_1}, \vec{x_2}, ..., \vec{x_n}\}$$
$$\chi_{FG} = \{\vec{y_1}, \vec{y_2}, ..., \vec{y_n}\}$$

Cette graine étant issue de la population totale de chaque ensemble γ_{BG} et γ_{FG}.

Après avoir obtenu la GMM (à partir d'un algorithme détaillé plus loin), la probabilité qu'un pixel représenté par le vecteur $\vec{x}$ appartienne (par exemple) au BG est calculée de la manière suivante :

$$\hat{p}(\vec{x}|\chi_{BG}, BG) = \sum_{k=1}^{K} \omega_k \phi(\vec{x}|\vec{\mu_k}, \vec{\sigma_k})$$

Avec $\hat{p}$ la probabilité supposée d'appartenance de $\vec{x}$ au BG, $\phi(\vec{\mu_k}, \vec{\sigma_k})$ les gaussiennes du mélange de K gaussiennes modélisant le BG, et ω_k le poids associé à chacune d'elles. Le mélange de gaussiennes formant cette modélisation s'inscrit dans le cadre d'un espace de représentation des pixels donné. On peut distinguer deux cas principaux, qui sont les espaces à une seule dimension tel que celui ayant pour seul canal la luminance L, et ceux ayant trois dimensions (RGB, L*a*b, etc).

La création d'une GMM à partir des observations se fait en deux temps. Tout d'abord une estimation des gaussiennes est déterminée par l'algorithme k-means. Puis cette estimation est affinée par l'algorithme espérance-maximisation (*expectation maximisation*, EM).
L'algorithme k-means consiste à regrouper les observations en K partitions, chaque observation étant associée à une partition en fonction de sa distance à la moyenne de celle-ci. L'algorithme se déroule comme décrit dans l'algorithme 4.1.

Algorithme 4.1 Algorithme k-means

Initialisation aléatoire de K moyenne m_k
X_k ensemble des observations de la partition k de moyenne m_k
tantque $assignationsModifees = True$ **faire**
  Assigner $x_i \in \chi$ à un X_k selon sa distance à m_k
  Calcul des m'_k, nouvelles moyennes des X_k
  si $m'_k \neq m_k$ **alors**
    $assignationsModifiees \leftarrow False$
  finsi
  $m_k \leftarrow m'_k$
fin tantque

On peut noter que l'assignation d'un x_i à un X_k en fonction de sa distance à la moyenne de cet ensemble revient à faire une partition de Voronoï de l'espace. Ce partitionnement est de plus non unique et le résultat dépend de l'initialisation.
L'algorithme précédent fournit une première estimation des μ_k ainsi que des ω_k (qui sont proportionnels au nombre d'observations associées à chaque ensemble X_k), il reste à déterminer les σ_k par l'algorithme EM (qui affinera également les μ_k et ω_k).
L'algorithme EM permet, en substance, d'estimer les paramètres de la GMM qui maximiseront la vraisemblance L de celle-ci pour une population γ données, tout en n'en connaissant que les observations χ. L'algorithme est décrit en 4.2.

Nous obtenons finalement une expression de la probabilité qu'un pixel appartienne à l'arrière plan ou à l'avant plan, selon le cas considéré. Le coût résultant de cette probabilité est calculé de telle manière qu'une probabilité élevée d'appartenance à un label implique un coût faible (voir [86]). L'utilisation du logarithme permet de sanctionner beaucoup plus sévèrement les choix peu probables :

$$C_{GMM}(BG) = -\ln \hat{p}(\vec{x}|\chi_{BG}, BG)$$

Algorithme 4.2 Algorithme Expectation - Maximisation

$\sigma_k \leftarrow 1$
$L^0 \leftarrow \frac{1}{n}\sum_{i=1}^{n}\log(\sum_{j=1}^{k}\omega_j^0\phi(x_i|\mu_j^0,\sigma_j^0))$ {Vraisemblance initiale}
$m \leftarrow 0$
répéter
  $m \leftarrow m+1$
  {Expectation step}
  $\gamma_{ij}^m \leftarrow \frac{\omega_j^m\phi(x_i|\mu_j^m,\sigma_j^m)}{\sum_{l=1}^{k}\omega_l^m\phi(x_i|\mu_l^m,\sigma_l^m)}, i=1...n, j=1...k$
  $n_j \leftarrow \sum_{i=1}^{n}\gamma_{ij}^m, j=1...k$
  {Maximisation step}
  $\omega_j^{m+1} \leftarrow \frac{n_j^m}{n}, j=1...k$
  $\mu_j^{m+1} \leftarrow \frac{1}{n^m}\sum_{i=1}^{n}\gamma_{ij}^m.x_i, j=1...k$
  $\sigma_j^{m+1} \leftarrow \frac{1}{n_j^m}\sum_{i=1}^{n}\sum_{i=1}^{n}\gamma_{ij}^m.(x_i-\mu_j^{m+1}).(x_i-\mu_j^{m+1})^T$
  {Vérification de la convergence}
  $L^{m+1} \leftarrow \frac{1}{n}\sum_{i=1}^{n}\log(\sum_{j=1}^{k}\omega_j^{m+1}\phi(x_i|\mu_j^{m+1},\sigma_j^{m+1}))$
jusqu'à $|L^{m+1}-L^m| > \delta$

$$C_{GMM}(FG) = -\ln\hat{p}(\vec{x}|\chi_{FG}, FG)$$

4.2.1.3 Segmentation à partir de la modélisation

Après que la modélisation de l'arrière plan a été créée et une nouvelle capture de la scène effectuée par la caméra, il reste à déterminer le label à attribuer à chacun des pixels de cette nouvelle capture (BG, FG ou indéterminé). Cette labelisation peut être faite à différentes échelles, du pixel (échelle locale) à l'image complète (échelle globale)

La labelisation à l'échelle du pixel consiste à ne pas considérer le voisinnage de chacun des pixels. Le label est déterminé à partir de sa valeur actuelle et de la modélisation de l'arrière plan. Dans [47], des seuils sont appliqués aux deux caractères discriminatoires conservés, à savoir la distorsion de la luminosité et la distorsion de la chromacité. Un raisonnement équivalent est appliqué dans le cas où la modélisation est basée sur une approche probabiliste, auquel cas les seuils s'appliqueront sur la probabilité qu'un pixel appartienne à l'avant ou à l'arrière plan [31, 93].
Afin d'obtenir des résultats plus uniformes, ces seuils sont souvent appliqués sur une fenêtre autour du pixel considéré, comme pour Pilet et al. [80] [62, 95, 67].

Pour conserver une homogénéité dans la segmentation, des algorithmes prenant en compte l'image dans sa globalité ont été mis en place. Parmi les méthodes recueillant beaucoup d'attentions depuis quelques années se trouvent ceux mettant en oeuvre les champs de Markov (*Markov Random Field*, MRF), et en particulier leur résolution par l'utilisation de *graph-cut*. Les champs de Markov permettent de décrire les problèmes relatifs à la vision par ordinateur comme une minimisation d'une énergie, celle-ci ayant pour une configuration f (c'est à dire pour une segmentation donnée) :

$$E(f) = E_{smooth}(f) + E_{data}(f)$$

$E_{smooth}(f)$ représente ici l'énergie de liaison entre les éléments de l'image, donc

entre les pixels. Une labellisation différente entre deux pixels proches conduira à une énergie de liaison importante ce qui réduit la probabilité que deux pixels voisins n'appartiennent pas à la même portion lors de la segmentation. $E_{data}(f)$ représente l'énergie des données, et qualifie la pertinence de l'attribution de chaque pixel à telle ou telle portion de la segmentation

La classe d'algorithme de type *graph-cut*, inaugurée par Ford et Fulkerson [34], est basée sur la théorie des graphes, et part du principe qu'il est plus simple d'évaluer un flux maximum dans un graphe qu'une coupe minimale (équivalente à une énergie minimale dans le cadre d'une minimisation d'énergie) [14] [96, 36, 58, 103].
Ces algorithmes peuvent être utilisés avec pratiquement n'importe quelle forme de modélisation de l'arrière plan, la seule contrainte étant que l'énergie soit de la forme de l'énergie de Gibbs donnée plus haut [60]. Les *graph-cuts* sont ainsi utilisés pour déterminer la segmentation dans le cadre d'une modélisation par mélange de gaussienne [24, 109] avec des spécificités telles que la prise en compte du contraste de l'image [96] ou de graines préalablement spécifiées [14].

Certains des algorithmes évoqués ici sont décrits avec plus de précision dans la section 4.2.3.

4.2.2 Captation de la géométrie de la scène

On désigne par carte de profondeur (ou *depth map*) une représentation sans unité de la profondeur de la scène selon l'axe Z d'une caméra (les axes X et Y étant ceux du plan du capteur de celle-ci). Après calibrage, il est possible d'en tirer des mesures voire un modèle tridimensionnel de la scène capturée.

Historiquement, les principales méthodes de création de carte de profondeur sont basées sur une reproduction de notre propre vision, c'est à dire à partir de deux sources d'images en stéréo. Si ces techniques restent les plus abordées dans la littérature scientifique, notamment du fait de la possibilité de multiplier les sources d'images et des progrès envisageables de par la facilité d'accès à des calculateurs puissants (la reconstruction stéréo étant une opération gourmande), d'autres approches sont apparues au fil des années, certaines aboutissant à des produits commercialisés.

4.2.2.1 Théorie de la reconstruction stéréo

La mise en correspondance d'image stéréo consiste à associer à chaque pixel de l'image de gauche un pixel de l'image de droite (en négligeant les occlusions éventuelles de même que l'unicité de la correspondance pour chaque pixel). Cette mise en correspondance produit une carte de disparité $d(x, y)$, dont la valeur en chaque pixel de coordonnées (x, y) tend vers zéro lorsque la distance du couple de caméras à l'objet augmente. En considérant que les deux caméras ont le même axe $\vec{y}$, la relation entre les images droite $I(x, y)$ et gauche $I'(x', y')$ est la suivante :

$$\begin{cases} x' = x + d(x, y) \\ y' = y \end{cases}$$

L'ensemble des valeurs que peut prendre la disparité entre les pixels d'une image et de l'autre forme l'espace des disparité, noté (x, y, d). Une image dans l'espace des

disparités (ou DSI, *disparity space image* [89]) est une fonction définie sur (x, y, d) dans $\mathbb{R}$ qualifiant la probabilité que le pixel (x, y) de l'image gauche soit associé au pixel (x', y') de l'image droite avec une disparité d donnée. C'est à partir de cette probabilité que le coût de correspondance, permettant le choix final du pixel associé, est calculé.

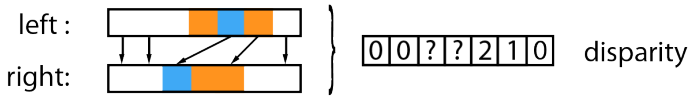


Figure 4.3 – La corrélation entre les pixels de l'image gauche et droite (lorsque c'est possible) permet de déterminer la disparité par pixel entre ces images. Les pixels n'ayant pas de correspondance sont considérés comme occultés.

La carte de disparité $d(x, y)$ est produite en choisissant à partir de cette image définie sur (x, y, d) la valeur de d la plus probable. Nous verrons que selon la méthode mise en oeuvre - locale ou globale, la valeur choisie correspondra forcément à celle pour laquelle le coût de correspondance est le plus faible (méthode locale) ou non (méthode globale). L'algorithme 4.3 décrit la structure générale des algorithmes de calcul de l'espace des disparités.

Algorithme 4.3 Boucle de création de l'espace de disparité

pour $0 \leq d \leq maxDisparity$ **faire**
  pour $0 \leq y \leq height$ **faire**
    pour $0 \leq x \leq width$ **faire**
      $p \leftarrow I(x, y)$
      $p' \leftarrow I'(x' + d, y)$
      $DSI(x, y, d) \leftarrow C(p, p')$
    fin pour
  fin pour
fin pour

4.2.2.2 Etat de l'art

Les algorithmes de reconstruction de carte de profondeur à partir d'une paire d'images stéréo sont pour le moins nombreuses. Scharstein et al. [89] a dressé en 2001 une taxonomie des algorithmes de mise en correspondance stéréo. Dans son mémoire de thèse, Scharstein décrit la structure générale de ces algorithmes [87] :

- Calcul du coût de correspondance entre un pixel de l'image de droite et un pixel de l'image de gauche
- Aggrégation des coûts
- Détermination de la disparité et optimisation du résultat
- Rafinement de la disparité (optionnelle)

Cette architecture peut varier sensiblement d'une implémentation à l'autre, en particulier deux étapes peuvent se voir effectuées en même temps. Une description plus précise des aboutissants de chaque étape est donnée dans la section correspondante.

4.2.2.3 Coût de correspondance

SAD / SSD : L'évaluation du coût de correspondance a pour finalité de créer l'image (DSI) de la fonction définie de l'espace des disparités (x, y, d) dans $\mathbb{R}$. Les premières approches, qui sont toujours parmi les plus communes, évaluent la distance dans l'espace de couleur considéré entre les valeurs des couples de pixels. Celles-ci sont dénommées SSD, pour *summed squared differences*. Une déclinaison mettant en oeuvre les différences absolues au lieu des différences élevées au carré, souvent nommée SAD pour *summed absolute differences* a été proposée par Kanade et al. [52] dans un souci de réduction des temps de calcul, et est également très utilisée.

Par exemple, dans le cas de l'espace sRGB, les coûts sont calculés comme suit :

$$C_{SAD}(p, p') = |r - r'| + |g - g'| + |b - b'| \qquad \text{(SAD)}$$

$$C_{SSD}(p, p') = \sqrt{(r - r')^2 + (g - g')^2 + (b - b')^2} \quad \text{(SSD)}$$

Avec p et p' les pixels de coordonnées (x, y) et (x', y) des images I et I'. Finalement, l'approche SAD correspond à une distance de Manhattan dans l'espace sRGB, tandis que l'approche SSD est une distance euclidienne.
L'expression du calcul de distance dépend de l'espace colorimétrique utilisé. En particulier dans les espaces colorimétriques cylindriques tels que le HSV et le HSL (qui présentent l'avantage d'offrir une meilleure représentation des distances entre couleurs que le sRGB), les calculs de distance doivent être adaptés.
Dans les espace HSV et HSL, la distance euclidienne entre deux points est calculée comme suit :

$$C_{HSV}(p, p') = sqrt(s.\cos h - s'.\cos h'^2 + (s.\sin h - s'.\sin h')^2 + v^2 \quad \text{(HSV)}$$

$$C_{HSL}(p, p') = sqrt(s.\cos h - s'.\cos h'^2 + (s.\sin h - s'.\sin h')^2 + l^2 \quad \text{(HSL)}$$

Les espaces dérivant du CIE XYZ, tel que le CIE L*a*b, sont considérés quant à eux comme des espaces orthonormaux au même titre que l'espace sRGB.

Normalized cross correlation (NCC) : La corrélation croisée et la corrélation croisée normalisée (NCC pour *normalized cross correlation*) [66] sont deux méthode de mesure de la similarité de deux signaux (dans le domaine fréquentiel). La corrélation croisée normalisée est considérée comme étant plus adaptée aux applications de mise en correspondance de signaux, mais est bien simple à mettre en oeuvre dans le domaine spatial que dans le domaine fréquentiel. La méthode mise en oeuvre dans ce mémoire est issue du travail de Lewis [66].
Dans notre cas, la corrélation croisée est une corrélation associée à la distance euclidienne entre un modèle et l'image sur laquelle on cherche à placer celui-ci. Elle s'exprime, dans le cas d'images, de la façon suivante :

$$(f * t)(u, v) = d_{f,t}(u, v) = \sqrt{\sum_{x,y} (f(x, y) - t(x - u, y - v))^2}$$

Avec f une image dans laquelle on cherche à retrouver le motif t, u et v évoluant dans les limites de f, x et y évoluant dans les limites de t. Cette expression présente l'inconvénient d'être sensible aux changements d'exposition de l'image. Aussi, on préfère utiliser la corrélation croisée normalisée, dont l'expression est :

$$\gamma(u,v) = \frac{\sum_{x,y}(f(x,y) - \overline{f})(t(x,y) - \overline{t})}{\sigma_f \sigma_t}$$

Avec σ_f et σ_t les écarts type de f et t, $\overline{f}$ et $\overline{t}$ leurs moyennes. Dans un premier temps, pour évaluer la qualité de la segmentation basée sur le calcul de la NCC, nous utiliserons cette expression malgré l'existence d'approches plus efficaces en temps de calcul. On a donc :

$$C_{NCC}(p,p') = \frac{\sum_{x,y}(p(x,y) - \overline{p})(t(x,y) - \overline{t})}{\sigma_f \sigma_t}$$

où $t(x,y)$ est une portion de p'.

Autres approches D'autre approches moins fréquentes existent dans la littérature : les approches binaires impliquant la présence ou non de caractéristiques [41] ; la maximization d'informations mutelles qui s'attache à rechercher des similarités entre deux images [55].

L'efficacité de l'implémentation de ces méthodes va dépendre de l'espace colorimétrique dans lequel les images sont représentées. Ainsi, Ivanov et al. [49] introduira la notion de luminosité d'un pixel calculée à partir des composantes RGB, et Harville et al. [46] travaillera dans l'espace YUV. Elle dépendra également de la qualité des caractéristiques (coins, arêtes ...) recherchées dans l'image.

4.2.2.4 Aggrégation des coûts

Suite à cette évaluation des coûts, un lissage est la plupart du temps appliqué pour réduire l'influence du bruit, ainsi que pour obtenir des surfaces plus homogènes. La méthode la plus répandue est un lissage sur une fenêtre définie soit en paramètre de l'algorithme, soit en fonction du contenu de l'image [53].
A noter que le filtrage de l'image DSI peut être fait soit en deux dimensions pour une valeur de disparité d donnée, soit en trois dimensions dans l'espace (x,y,d) [82].

4.2.2.5 Détermination de la disparité et optimisation

La création de la carte de disparité se fait durant cette étape. L'algorithme choisi sera chargé de déterminer, à partir de la DSI, quelle disparité associer à chaque pixel. Ce choix peut se faire à deux échelles différentes : soit au niveau local, en étudiant chacun des pixels indépendamment des autres ; soit au niveau global, en considérant l'image comme un tout.

Dans le cas des méthodes locales, l'approche considérée est de choisir pour chaque pixel la disparité ayant obtenue le meilleur score (ou le coût le moins élevé). Une vérification croisée, consistant à évaluer la disparité en choisissant l'image gauche comme

référence, puis l'image droite (donnant ainsi deux cartes de disparités), et en comparant leurs résultats, permet de rechercher les zones d'occlusion entre les deux images.
Cependant, certaines méthodes locales sont légèrement différentes et prennent en compte la disparité des pixels voisins du pixel considéré. C'est le cas de Stefano et al. [94] dont la méthode s'assure qu'à un pixel de l'image droite ne correspond qu'un seul pixel de l'image gauche.

Les méthodes d'optimisation globales sont semblables à celles utilisées pour la segmentation d'image et se basent sur les champs de Markov, avec une énergie à minimiser de la forme suivante :

$$E(d) = E_{data}(d) + \lambda . E_{smooth}(d)$$

Le terme $E_{data}(d)$ est en relation avec le coût issu de la DSI, tandis que le terme $E_{smooth}(d)$ est dépendant des différences de disparité entre pixels voisins.

$$E_{data}(d) = \sum_{x,y} DSI(x, y, d)$$

$$E_{smooth}(d) = \sum_{x,y} \sum_{u,v} \rho(|d(x, y) - d(x + u, y + v)|)$$

ρ est une fonction définie positive telle que sa valeur est nulle lorsque la disparité des deux pixels considérés est égale, et prend une valeur positive lorsque ce n'est pas le cas. Dans la plupart des implémentations, on a $u, v \epsilon [-1; 1]$. Selon l'implémentation, ρ se contentera de lisser la carte de disparité, ou conservera les discontinuités de disparité en les détectant pour limiter l'effet dans ces zones.
La classe d'algorithmes la plus couramment utilisée à ce jour pour résoudre ce problème de minimisation globale est celle des *graph-cut*, issue de la théorie des graphes [16, 15, 59, 29, 103].

Les algorithmes de minimisation de l'énergie évoqués ici étant les mêmes que ceux utilisés pour la segmentation d'image, des précisions sont disponibles à leur sujet dans la section 4.2.3.

4.2.2.6 Rafinement de la disparité

Dans la majorité des cas, la carte de disparité obtenue est constituée de valeurs définies sur $\mathbb{N}$, ce qui amène une précision dans le meilleur des cas de 0.5 pixel. Des méthodes de rafinement de la disparité existent pour améliorer la précision de la carte de disparité.
Tian et al. [98] présente la plupart des algorithmes utilisés à ces fins. A noter que ceux-ci sont destinés au domaine du recalage d'image (*image registration*), donc peuvent s'appliquer aussi bien à la mise en correspondance stéréo qu'à la segmentation d'image.

La méthode la plus évidente consiste à augmenter artificiellement la résolution des images d'entrée (*upsampling*), d'en tirer une carte de disparité en haute résolution puis de réduire la résolution de celle-ci pour retrouver la résolution d'origine. Le coût en calcul d'une telle méthode est important, une multiplication par deux de la résolution sur les deux axes augmentant par quatre le nombre de pixels à évaluer tout en ne fournissant une précision que deux fois supérieure au mieux. Des méthodes par itération

successive en augmentant progressivement la résolution existent et réduisent fortement la charge de calcul supplémentaire (algorithme *coarse to fine*).

Une autre méthode couramment utilisée est de déterminer une interpolation, le plus souvent linéaire ou quadratique [71], entre les deux valeurs de disparité les plus susceptibles de correspondre au pixel considéré. Cette approche permet d'envisager des précisions supérieures au dixième de pixel, avec toutefois le désavantage de lisser les hauts contrastes dans le cas où l'interpolation ne les prend pas en considération.

4.2.2.7 Solutions matérielles

Parallèlement aux méthodes "historiques" de détermination de la profondeur mettant en oeuvre un couple de caméras, d'autres supports matériels ont fait leur apparition, menant pour certains à des solutions commerciales tout en un fournissant une carte de disparité de la scène. Nous ne parlerons pas ici des solutions commerciales mettant en oeuvre des caméras stéréo associées à des algorithmes présentés précédemment.

Z-cameras : Les Z-cameras sont constituées, comme leurs consoeurs fournissant une représentation RGB de la scène, d'un capteur de technologie équivalent (CMOS) mais spécifiquement adapté pour être plus sensible aux infrarouges, ainsi que d'un émetteur de ces mêmes rayons infrarouges. L'émetteur diffuse un signal périodique dont le capteur mesurera la partie réfléchie par les objets de la scène. En étudiant la périodicité du signal réfléchi, et en particulier le décalage de phase par rapport au signal source, il est possible de déterminer l'éloignement des objets.

Si les résultats issus des Z-cameras actuellement disponibles sont prêts à l'emploi, ils ont pour l'instant le défaut d'avoir une résolution limitée (au mieux en QVGA, soit 320x240), ce qui est faible au regard des capacités des caméras haute définition disponibles. Ainsi, Hahne et al. [43] propose de combiner les résultats d'une Z-camera à un couple de caméras stéréo pour profiter des avantages des deux méthodes, la vitesse de rafraîchissement et le nombre limité de faux positifs d'un coté, la résolution de l'autre. Quant à Wang et al. [104], son approche permet d'extraire de manière précise l'avant plan d'une scène en combinant une Z-camera et une caméra haute définition, sur la base d'un seuil appliqué à la mesure de la Z-camera.

Citons enfin la z-caméra développée par PrimeSense et commercialisée par Microsoft, nommée Kinect (figure 4.4). Destinée au grand public, elle a la particularité d'être très abordable et de proposer une résolution élevée (pour une z-caméra) de 640x480. Son fonctionnement est différent des approches précédentes puisque ce dispositif est doté d'un projecteur laser infrarouge qui couvre la scène avec un motif particulier. La déformation de ce motif, détecté par une caméra infrarouge, permet d'estimer la carte de profondeur.

Caméras plénoptiques : Le principe des caméras plénoptiques est d'intégrer un réseau de micro-lentilles à une caméra standard, permettant à celle-ci de capturer de multiples points de vue en même temps [75, 11]. A partir de cette multitude de captures très légèrement différentes les unes des autres, il est possible de déterminer le champ de lumière de la scène. Comme expliqué par Ng et al. [75], un champ de lumière L décrit

Figure 4.4 – La z-caméra Kinect, de Microsoft

dans un espace à quatre dimensions (u, v, s, t) la trajectoire des rayons lumineux entre le diaphragme et le capteur. La connaissance de ce champ de lumière permet, entre autres, de déterminer la carte de profondeur d'une scène et de modifier le plan de mise au point après la prise de vue.

Cette approche est encore jeune et souffre d'une limitation de la résolution de sortie. La résolution finale étant divisée par le nombre de micro-lentilles, il faut choisir entre la résolution du champ de lumière et la résolution finale des images. Récemment, la jeune entreprise LYTRO créée suite aux travaux de Ng et al. [75] a dévoilé le projet de commercialiser une caméra plénoptique. Cet appareil n'est pas encore disponible à date de ce travail.

Caméras à diaphragme codé : Lors de la capture d'une image par une caméra, trois zones peuvent être extraites de celle-ci : une zone nette (à proximité du plan de mise au point), la zone floue en avant de la zone nette, et une autre en arrière. Le flou qui est appliqué dans ces deux dernières zones dépend de deux facteurs qui sont la distance au plan de netteté, et la forme du diaphragme. C'est sur ce second facteur que joue les caméras à diaphragme codé pour tirer une carte de profondeur.
Le flou correspond à une convolution, dans le domaine fréquentiel, de la réponse fréquentielle du diaphragme sur l'image. Ce flou est nommé en photographie cercle de confusion, et son rayon dépend de la distance de l'objet au plan de netteté. Les diaphragmes étant, dans la mesure des possibilités techniques, circulaires, leur réponse fréquentielle tend à supprimer toutes les hautes fréquences et donc à faire disparaître les détails d'une image. En faisant une déconvolution, il n'est alors pas possible de retrouver ces détails.
Ainsi, dans les caméras à diaphragme codé, celui-ci est choisi pour avoir une réponse fréquentielle conservant la plus large bande d'information possible. Après la prise de vue, il reste à déterminer la dimension du flou qui a été appliqué, et à effectuer une déconvolution avec ce paramètre. Le résultat est d'une part une image nette sur la totalité de la profondeur de la scène ; d'autre part une évaluation de la profondeur des objets à partir de la taille du flou déterminée précédemment.

Quelques implémentations ont été faites, en laboratoire, de caméra à diaphragme codé, notamment par Farid et al. [32], Levin et al. [65] (figure 4.5) et Veeraraghavan et

al. [100]. Ce dernier propose une méthode basée sur une matrice LCD monochrome utilisée comme diaphragme, ce qui a deux avantages : d'une part, il est possible d'adapter le motif en fonction de la distance des objets, et d'autre part en modulant le motif le champ de lumière peut être extrait de la scène sans perte de résolution contrairement aux caméras plénoptiques.

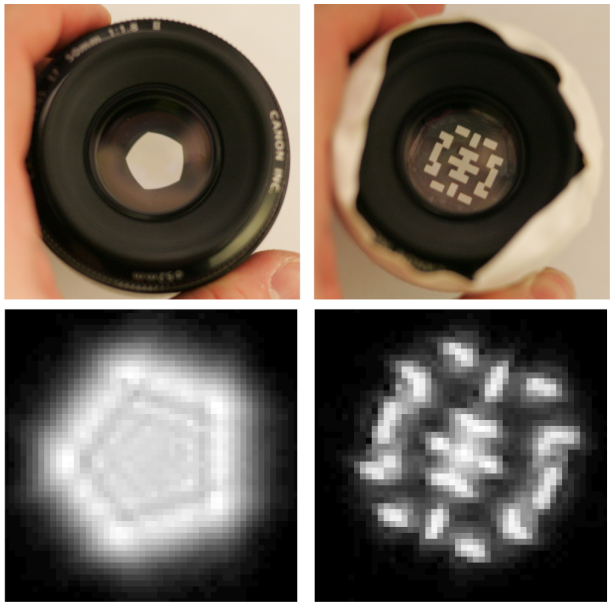


Figure 4.5 – Optique dotée d'un diaphragme codé [65]

4.2.3 Résolution des graphes formés par ceux deux problèmes

4.2.3.1 Cas de la minimisation locale :

La minimisation locale est équivalente à considérer que l'énergie liée au lissage (E_{smooth}) est nulle quelle que soit la configuration : chaque pixel se retrouve alors considéré indépendamment de ses voisins, il s'agit donc de choisir la configuration pour laquelle l'énergie d'un pixel est la moins élevée. Cette méthode est particulièrement simple à mettre en place et ne nécessite pas d'algorithme particulier.

4.2.3.2 Cas de la minimisation globale :

La minimisation globale de l'énergie consiste à prendre en compte l'image dans son ensemble pour en déduire la meilleure configuration f des pixels de telle sorte que l'énergie $E(f)$ de l'ensemble du graphe représentant l'image soit minimale. La configuration de chaque pixel n'est donc pas nécessairement celle pour laquelle l'énergie issue de ce pixel en particulier est la plus faible.
L'approche la plus courante pour résoudre ce problème de minimisation est d'exprimer l'image comme étant un champ aléatoire de Markov (*Markov Random Field*, ou MRF). Un MRF est un graphe $\mathcal{G} = \{\nu, \varepsilon\}$ du type suivant :

- $\nu = \{p_1, ..., p_n\}$ l'ensemble des noeuds du graphe, correspondant entre autres aux pixels de l'image. Chaque noeud est associé à une variable aléatoire $f_j \in F = \{f_1, ..., f_n\}$ représentant soit la disparité du pixel associé (dans le cas de la mise en correspondance stéréo), soit le label du pixel (dans le cas de la segmentation).
- $\varepsilon = \{e_1, ..., e_m\}$ les liens entre les noeuds et leur voisinnage $\mathcal{N}$.

En traitement d'image, on considère la plupart du temps le voisinage d'un pixel en 4-connectivité ou en 8-connectivité, selon que l'on y inclut les 4 ou 8 pixels voisins du pixel considéré. Ce travail couvrira pour l'instant le cas de la 4-connectivité. Les noeuds terminaux sont eux liés à tous les noeuds des pixels. Un MRF suit de plus la loi suivante :

$$p(f_i|\{f_j\}_{j\in\varepsilon i}) = p(f_i|\{f_j\}_{j\in\mathcal{N}_i})$$

Cette relation signifie que la configuration d'un pixel de l'image peut être déterminée par la connaissance des pixels de son voisinnage. On peut noter que la probabilité qu'un noeud p_i prenne la configuration f_i est la suivante, à une constante près :

$$p(f_i|\nu) = p(\nu|f_i).p(f_i)$$

Cette expression donne ainsi, en passant en logarithme :

$$\log p(f_i|\nu) = \log p(f_i) + \log p(\nu|f_i)$$

La première partie de cette égalité correspond à l'énergie du noeud p_i, la seconde est la somme de l'énergie liée au voisinnage de p_i et de l'énergie liée à la valeur à priori (sans prendre en compte le voisinnage) de f_i. L'énergie totale du graphe est alors calculée de la manière suivante :

$$E(f) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{N}_p} V_{(p,q)}.(f_p, f_q) + \sum_{p\in\mathcal{P}} g(i_p, p, f_p)$$

où f est une configuration donnée de l'ensemble des variables aléatoires f_j de l'image, p et q sont deux pixels de l'image, q appartenant au voisinnage de p, i_p la valeur de p et f_p la variable aléatoire associée à p. $g(.)$ quant à elle est la fonction qualifiant le coût de l'attribution de la valeur de f_p pour l'observation i_p.
On peut dissocier cette expression de $E(f)$ en deux membres, qui sont l'énergie liée aux données E_{data} et l'énergie de lissage E_{smooth} :

$$E_{data}(f) = \sum_{p\in\mathcal{P}} g(i_p, p, f_p)$$

$$E_{smooth}(f) = \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{N}_p} V_{(p,q)}(f_p, f_q)$$

L'énergie E_{data} est liée aux coûts (de segmentation ou de reconstruction stéréo) détaillés précédemment. En revanche l'énergie de lissage est issue des relations entre pixels voisins, c'est à dire soit de différences dans leur attribut f_j, soit du fait de leurs caractéristiques dans l'image (différences de couleurs, contraste, etc).

Expression des coûts : Quelque soit le problème traité, le lissage a pour objectif de réduire les différences de labellisation le long d'un même objet ainsi que d'améliorer la détermination du label des pixels pour lesquels celui-ci est incertain. Comme nous l'avons vu précédemment, l'énergie de lissage a pour forme :

$$E_{smooth}(f) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p} .\rho(|f_p - f_q|)$$

$\rho(.)$ étant une fonction définie positive de la forme suivante :

$$\rho(|f_p - f_q|) = V_{p,q}(f_p, f_q)$$

Les algorithmes d'optimisation utilisés dans ce travail, construits autour des *graph-cut*, imposent comme nous le verrons dans la suite que $V_{p,q}(f_p, f_q)$ soit une semi-métrique, voire une métrique (voir glossaire) [17].
Les coûts de données et de lissage mis en oeuvre dans le cadre de la reconstruction stéréo et de la segmentation sont finalement les suivants :

- **Reconstruction stéréo** : Selon la méthode employée pour le calcul du coût de donnée (SAD, SSD, NCC, ...) nous avons pour une configuration f donnée :

  $$E_{data}(f) = \sum_{p \in \mathcal{I}, p' \in \mathcal{I}'} C_{stereo}(p, p')$$

  Le coût de lissage permet quant à lui de réduire la probabilité que deux pixels voisins aient une disparité différente. Il est choisi comme suit :

  $$V_{p,q}(fp, fq) = \frac{|(fp - fq) * k_{smooth}|}{d_{max}}$$

  Avec k_{smooth} un coefficient pondérant l'influence du lissage, et d_{max} la disparité maximale considérée pour la reconstruction.

- **Segmentation** : Comme expliqué dans la section précédente, le coût de données est calculé comme étant l'opposé du logarithme de la probabilité qu'un pixel appartienne à l'arrière plan ou à l'avant plan :

  $$E_{data}(f) = \sum_{p \in \mathcal{P}} -\ln(p|(\chi_{BG}, \chi_{FG}), f_p)$$

  Il s'agit donc de la somme des probabilités que p appartienne au BG ou au FG (selon sa configuration f_p) connaissant les graines du FG et du BG.
  L'énergie de lissage quant à elle tente d'empêcher deux pixels voisins d'appartenir à des portions différentes. Pour ce faire, nous avons utilisé la métrique proposée par Boykov et al. [14] en l'étendant à des images à plus d'un canal :

  $$V_{p,q}(fp, fp) = \exp\left(-\frac{(I_p - I_q)^2}{2 * \sigma^2}\right) . \frac{1}{dist(p,q)} * k_{smooth}$$

  Avec $(I_p - I_q)$ l'expression du calcul de la distance entre p et q dans l'espace colorimétrique considéré, et $dist(p,q)$ la distance entre ces mêmes pixels dans l'image. Intuitivement, on comprend que plus deux pixels sont éloignés dans leur espace colorimétrique et dans l'image, moins l'énergie de lissage est importante.

Résolution du graphe : Comme dit précédemment, le graphe est résolu par l'utilisation d'algorithmes de la classe des *graph-cuts* [34]. Pour ce faire, aux noeuds du graphe décrit précédemment sont ajoutés deux noeuds dits terminaux, notés S et T (*source* et *sink*), représentant chacun une configuration possible de chacun des pixels, parmi l'ensemble des labels $\mathcal{L}$. Ces deux terminaux sont liés à l'ensemble des autres noeuds, mais pas entre eux.
Le graphe est alors construit comme suit :

- S et T représentent chacun une configuration possible, par exemple un niveau de disparité ou un label (BG ou FG).
- Tous les autres noeuds $p \in \nu$ du graphe sont liés à S et à T par un lien $t_{p,S} \in \varepsilon$ et $e_{p,T} \in \varepsilon$, dotés d'un poids qui est le coût de données qui serait utilisé pour calculer l'énergie totale du graphe si l'on attribuait à p la configuration décrite par S ou T
- Les noeuds p sont reliés à leurs voisins (en 4-connectivité dans notre implémentation) par des liens $e_{p,q}$, $q \in \mathcal{N}_p$. Le poids attribué à ces noeuds correspond à l'énergie de lissage entre chaque pixel.

Il apparaît alors évident que ce type de graphe ne permet d'optimiser la distribution de labels qu'entre deux valeurs. Si ce n'est pas problématique dans le cas d'une segmentation arrière-plan / avant-plan, ça l'est évidemment pour la reconstruction stéréo où le nombre de label dépend directement de la disparité maximale entre les deux images. Boykov et al. [17] ont proposé des algorithmes permettant de résoudre cette difficulté, en considérant deux labels à la fois et en itérant avec des couples de labels différents, jusqu'à ce qu'un minimum d'énergie soit atteint.
L'algorithme dit *swap move* consiste à considérer les échanges entre les ensembles P_α et P_β contenant les noeuds ayant l'un des deux labels α et β, ceux-ci pouvant prendre n'importe quelle valeur dans la liste de labels $\mathcal{L}$ pourvu que l'ensemble des couples possibles soient évalués. Les noeuds ayant un label courant différent de α ou β ne sont pas modifiés. Cette procédure est répétée autant de fois que nécessaire jusqu'à ce que l'énergie d'une nouvelle itération n'amène pas d'amélioration à l'énergie du graphe.

Algorithme 4.4 Algorithme *Swap move*

```
Initialiser avec une configuration f
succes := true
tantque succes = true faire
  succes := false
  pour tout {α,β} ⊂ L faire
    Trouver f̂ = arg min E(f') parmi les configurations f' accessibles par un échange
    α − β dans f (par un algorithme de graph-cut)
    si E(f̂) < E(f) alors
      f := f̂
      succes := true
    finsi
  fin pour
fin tantque
Renvoyer f
```

L'algorithme dit *expansion move* consiste à considérer un label α et à évaluer les possibilités d'expansion de l'ensemble P_α sur l'ensemble de tous les autres noeuds. Cette évaluation est faite pour l'ensemble des valeurs de la liste de labels, et de la même manière cette procédure est répétée jusqu'à atteindre un minimum.

Algorithme 4.5 Algorithme *Expansion move*

Initialiser avec une configuration f
succes := true
tantque succes = true **faire**
  succes := false
  pour tout $\alpha \in \mathcal{L}$ **faire**
    Trouver $\hat{f} = \arg\min E(f')$ parmi les configurations f' accessibles par une expansion de α à partir de f (par un algorithme de *graph-cut*)
    si $E(\hat{f}) < E(f)$ **alors**
      $f := \hat{f}$
      succes := true
    finsi
  fin pour
fin tantque
Renvoyer f

Pour la reconstruction stéréo, nous avons choisi d'utiliser l'algorithme *swap move* du fait de la simplicité de son implémentation. Le graphe est en effet exactement celui décrit plus haut (c'est à dire un noeud par pixel, en plus des deux noeuds terminaux), et est doté des poids suivants (figure 4.6) :

lien	poids	cas
t_p^α	$C_{stereo}(p,\alpha) + \sum_{q \in \mathcal{N}_p, q \notin \mathcal{P}_{\alpha\beta}} V_{p,q}(\alpha, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
t_p^β	$C_{stereo}(p,\beta) + \sum_{q \in \mathcal{N}_p, q \notin \mathcal{P}_{\alpha\beta}} V_{p,q}(\beta, f_q)$	$p \in \mathcal{P}_{\alpha\beta}$
$e_{\{p,q\}}$	$V_{\{p,q\}}(\alpha,\beta)$	$q \in \mathcal{N}_p, \{p,q\} \in \mathcal{P}_{\alpha\beta}$

La segmentation a été faite, quant à elle, en optant pour une méthode semblable à celles proposées par Boykov et al. [14] et Duchenne et al. [30]. Les premiers proposent un graphe, destiné à être optimisé par l'algorithme *expansion move*, contenant des contraintes dures c'est-à-dire forçant la configuration de certains pixels. Ceci est intéressant puisque notre graine a entre autres buts de spécifier avant la segmentation l'appartenance de certains pixels. Pour ce faire, des poids particuliers sont donnés aux liens de ces pixels.
Cet algorithme impose également l'ajout de noeuds intermédiaires entre les noeuds (représentant les pixels) n'ayant pas le même label. Un exemple d'un tel noeud est nommé "a" dans la figure 4.7.

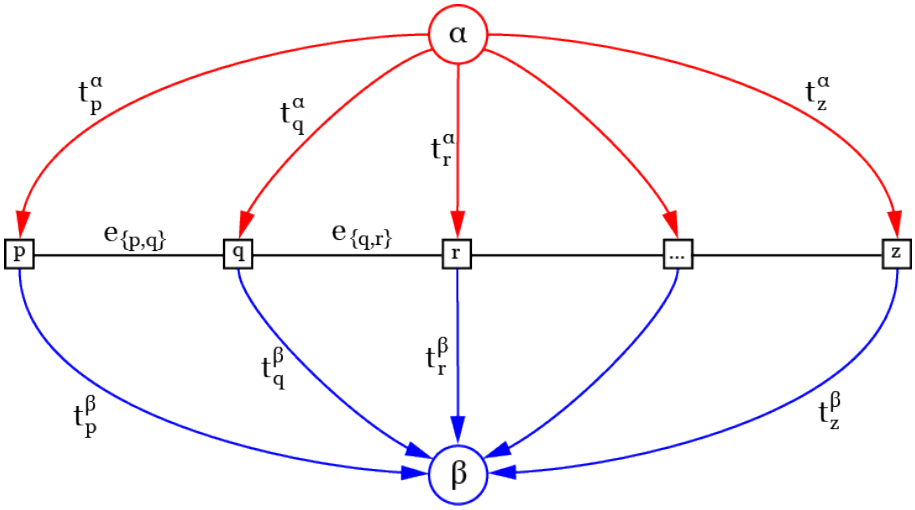


Figure 4.6 – Structure du graphe pour l'algorithme *swap move*, ici dans le cas simplifié d'une image 1D

lien	poids	cas
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$C_{FG}(p, f_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$C_{BG}(p, \alpha)$	$p \in \mathcal{P}$
$e_{\{a,q\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$q \in \mathcal{N}_p, f_p \neq f_q$
$e_{\{a,q\}}$	$V_{\{p,q\}}(\alpha, f_q)$	$q \in \mathcal{N}_p, f_p \neq f_q$
t_a^{α}	$V_{\{p,q\}}(f_p, f_q)$	$q \in \mathcal{N}_p, f_p \neq f_q$
$e_{\{p,q\}}$	$V_{\{p,q\}}(f_p, \alpha)$	$q \in \mathcal{N}_p, f_p = f_q$

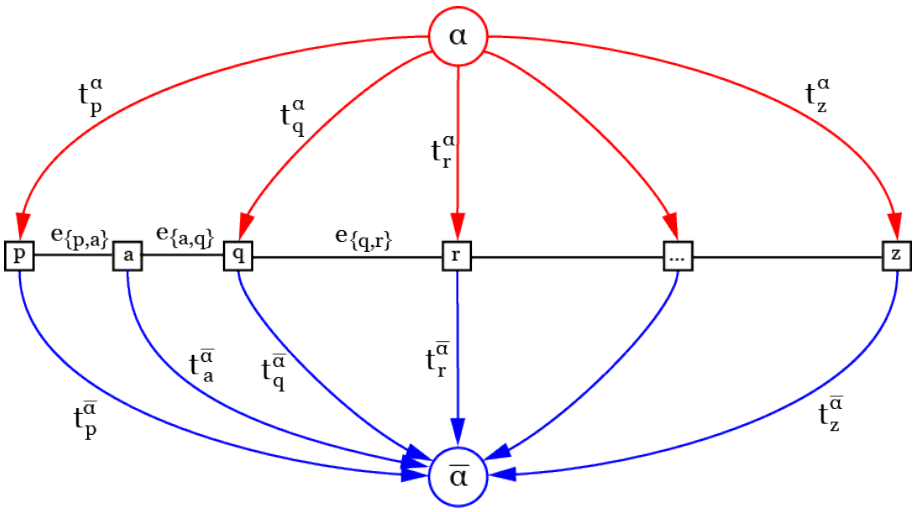


Figure 4.7 – Structure du graphe pour l'algorithme *expansion move*, ici dans le cas simplifié d'une image 1D

La résolution de chaque graphe par un algorithme de *graph-cut* consiste à trouver la coupe de coût minimale séparant les noeuds en deux groupes distincts. Le coût d'une coupe $\mathcal{C}$ (figure 4.8) est calculé comme suit :

$$E(\mathcal{C}) = \sum_{t \in \mathcal{C}} t + \sum_{e \in \mathcal{C}} e$$

Le label est alors attribué à un noeud de la manière suivante, pour la reconstruction stéréo et pour la segmentation :

Reconstruction stéréo	**Segmentation**
si $t_p^\alpha \in \mathcal{C}$ avec $p \in \mathcal{P}_{\alpha\beta}$ alors $f_p^\mathcal{C} = \alpha$	$t_p^\alpha \in \mathcal{C}$ avec $p \in \mathcal{P}$ alors $f_p^\mathcal{C} = \alpha$
si $t_p^\beta \in \mathcal{C}$ avec $p \in \mathcal{P}_{\alpha\beta}$ alors $f_p^\mathcal{C} = \beta$	$t_p^{\bar{\alpha}} \in \mathcal{C}$ avec $p \in \mathcal{P}$ alors $f_p^\mathcal{C} = f_p$
si $p \in \mathcal{P}$ et $p \notin \mathcal{P}_{\alpha\beta}$ alors $f_p^\mathcal{C} = f_p$	

Ainsi, le label attribué à un noeud sera celui correspondant au lien qui aura été coupé par $\mathcal{C}$: si t_p^α est coupé, alors le label α sera attribué au pixel p.

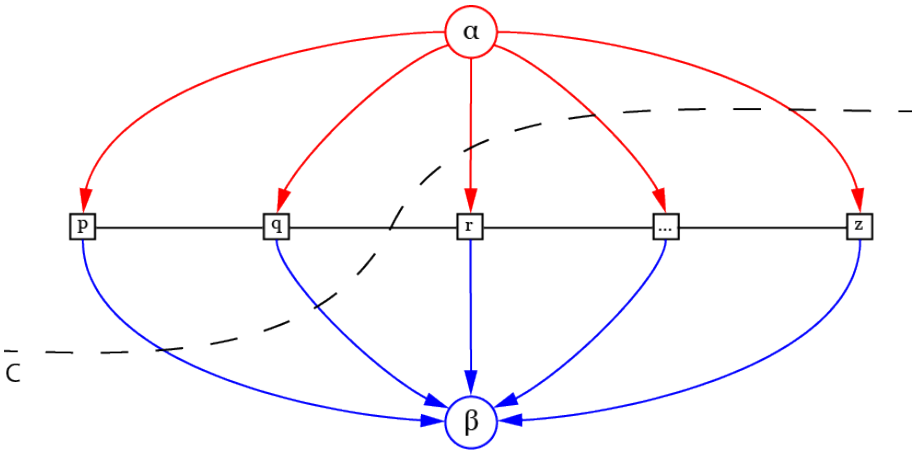


FIGURE 4.8 – Exemple d'une coupe $\mathcal{C}$ dans le graphe de l'algorithme *swap move*

Les algorithmes utilisés pour résoudre ces graphes ne seront pas détaillés ici (voir Ford et Fulkerson [34], Goldberg et al. [38], Cherkassky et al. [20], Thalwitzer [97]), et si des implémentations de certains de ces algorithmes ont été faites, c'est une version intégrée à la librairie NPP (*NVidia Performance Primitives*) qui a finalement été utilisée. Celle-ci permet en effet un gain extrêmement important de temps de calcul puisque fonctionnant sur GPU (carte graphique), contrairement à notre implémentation sur CPU. A titre de comparaison, notre implémentation sur CPU rendait impossible toute utilisation d'images d'une résolution supérieure à 320x240 pour la reconstruction stéréo du fait de temps de calcul trop important (de l'ordre de plusieurs heures pour notre première implémentation en Python), tandis que la librairie NPP nous permet de travailler confortablement sur des images en 640x480 (résultats en quelques dizaines de secondes pour notre code peu optimisé).
A titre d'information, toutes ces implémentations sont basées sur les travaux de Boykov et al. [15] (algorithme *augmenting path*) et Golberg et al. [38] (algorithme *push-relabel*). Le premier a été implémenté sur CPU du fait de ses bonnes performances pour les problèmes liés à l'informatique graphique, tandis que le second a été choisi pour les NPP car plus simple à paralléliser (voir Vineet et al. [103]).

Cependant, l'implémentation de *graph-cut* dans les NPP impose un certain format pour le graphe qui doit être régulier, c'est à dire que chaque ligne du graphe a le même

nombre d'éléments, qui sont tous reliés à leurs voisins en 4-connectivité. Ce format est tout à fait adapté à l'algorithme *swap move*, mais impose quelques aménagements au graphe de l'algorithme *expansion move* pour pouvoir le résoudre à l'aide des NPP. Notre solution a été d'ajouter des noeuds "factices" afin de compléter le graphe, ces noeuds étant liés à leurs voisins et aux noeuds terminaux par des liens nuls, qui ne modifient donc pas le coût global d'une coupe s'ils se trouvent lui appartenir. Ils n'ont donc pas d'influence sur le résultat final tout en permettant d'utiliser les NPP (figure 4.9).

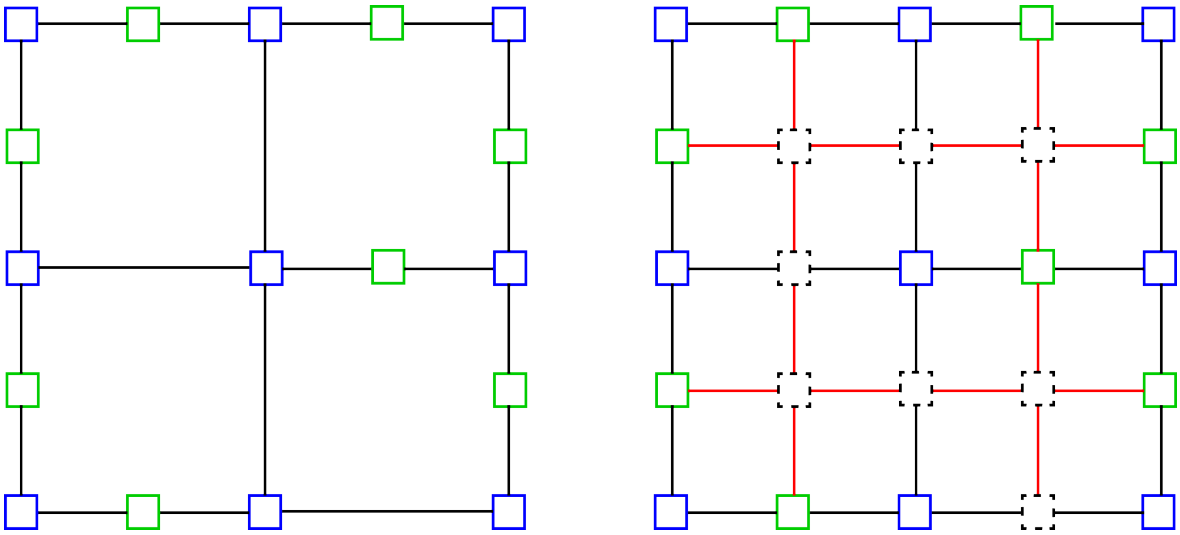


Figure 4.9 – Adaptation du graphe de l'algorithme *expansion move* pour être résolu par les NPP :
- Droite - Graphe d'origine : noeuds des pixels en bleu, noeuds additionnels en vert
- Gauche - Graphe adapté : noeuds des pixels en bleu, noeuds additionnels en vert, noeuds factices en pointillés. Les liens en rouge sont associés à un coût nul.
- Les liens terminaux ne sont pas représentés pour une question de lisibilité.

4.2.4 Modèle pour la mesure de la qualité de la segmentation

Deux caractéristiques sont à l'étude dans ce travail, liées au cadre de la réalité mixte que nous nous sommes imposé. Tout d'abord, la segmentation doit se faire à une fréquence suffisante et avec une latence réduite pour permettre l'utilisation de notre méthode dans ce cadre. Ces deux caractéristiques constituent nos deux premières métriques qualifiant notre méthode.

Les métriques utilisées pour évaluer la qualité de la segmentation sont celles utilisées couramment dans la littérature : la précision (*precision*) et le rappel (*recall*) [83]. La précision est la probabilité qu'un pixel évalué comme appartenant à l'avant-plan le soit effectivement (selon la vérité-terrain). Le rappel est la probabilité qu'un pixel appartenant à l'avant-plan (selon la vérité-terrain donc) soit détecté comme tel par la segmentation. Ces deux mesures sont calculées de la manière suivante :

$$recall = \frac{TP}{TP+FN} \qquad precision = 1 - \frac{FP}{TP+FP}$$

Avec TP (pour *True Positive*) le nombre de pixels justement détectés comme appartenant à l'avant-plan, FP (*False Positive*) les pixels faussement détectés comme étant dans l'avant-plan, et FN (*False Negative*) les pixels de l'avant-plan non détectés comme tels.

4.3 Approche proposée

Nous allons aborder dans cette partie la phase intermédiaire de l'approche proposée dans ce travail, qui lie l'acquisition de la carte de profondeur et la segmentation finale : la création de la graine. Pour rappel, celle-ci est destinée à alimenter l'algorithme de segmentation semi-automatique et remplacer l'intervention humaine.

La graine est donc issue de la carte de profondeur de la scène, que celle-ci soit créée à partir d'une carte de disparité ou qu'il s'agisse de la mesure d'une z-caméra comme le Kinect. Pour créer la graine, nous commençons par faire une segmentation sur la carte de profondeur (a priori de moindre qualité que ce que sera la segmentation finale). Un modèle statistique de la scène "vide" est créé, puis chaque nouvelle capture est comparée à ce modèle pour en tirer une segmentation. Comme l'une des caractéristiques des méthodes d'obtention de la carte de profondeur est qu'elle peut contenir des zone d'ombre (le fait qu'il n'y en ait pas est rare), la segmentation se décompose en trois zones : arrière-plan, avant-plan, et non défini.

Pour répondre aux exigences de l'algorithme de segmentation, la graine devra avoir les caractéristiques suivantes :

(1) Sans bruit : pas de blobs isolés issus d'erreurs dans les cartes de profondeur

(2) Sans faux positif : les zones considérées comme appartenant au BG ou au FG doivent être certaines

(3) Etre résistante aux modifications légères de l'arrière plan (feuillage par exemple)

(4) Fournir suffisament d'informations pour la création d'un modèle valable pour la seconde étape de segmentation.

L'approche que nous avons utilisée pour répondre au mieux à ces impératifs est la suivante. Pour commencer, le modèle mis en oeuvre pour caractériser l'arrière-plan consiste en deux informations pour chaque pixel de la carte de profondeur de la scène "vide" : la moyenne et l'écart type. Ces données sont évaluées sur une durée dépendant de la présence d'éléments mobiles en arrière plan, et de la période (éventuelle) de leur mouvement. Les données peuvent être considérées valides si la durée de capture est supérieure ou égale à la période la plus longue observable dans la scène. Finalement, nous connaissons pour chaque pixel p de la scène sa moyenne $\bar{z}$ ainsi que son écart type $\sigma(z)$.

Lors de la phase de segmentation, après cette première phase de création du modèle, les pixels de chaque capture sont divisés en trois groupes comme dis précédemment : l'arrière-plan noté $\mathcal{B}_0$; l'avant-plan noté $\mathcal{F}_0$; la zone d'incertitude notée $\mathcal{O}_0$. La règle permettant de classer les pixels dans un de ces trois groupes est la suivante, z étant la profondeur du pixel considéré p dans la capture actuellement étudiée :

$$|z - \bar{z}| \leq 2\sigma(z) \Leftrightarrow p \in \mathcal{B}_0$$

$$|z - \bar{z}| \geq 3\sigma(z) \Leftrightarrow p \in \mathcal{F}_0$$

$$2\sigma(z) \leq |z - \bar{z}| \leq 3\sigma(z) \Leftrightarrow p \in \mathcal{O}_0$$

Un seul de ces trois groupes est destiné à être segmenté une nouvelle fois dans la suite de notre méthode, et il s'agit du groupe $\mathcal{O}_0$. Les deux autres groupes, en plus d'être utilisés pour créer les modèles de l'arrière et de l'avant-plan, ne peuvent qu'accueillir de nouveaux pixels, et non en perdre. Il s'agit d'un élément important de l'algorithme de segmentation par *graph cuts* mis en oeuvre après la création de la graine.

Cette première segmentation produit, au sein du groupe $\mathcal{F}_0$, un certain nombre d'éléments au sens de pixels appartenant à ce groupe et à un voisinage commun. Avant de passer à la création du modèle, nous procédons à un classement de ces éléments, que nous nommerons blobs. Comme cela a été expliqué précédemment (voir section 4.2.1), les modèles de l'arrière et de l'avant-plan sont constitués d'un nombre défini de gaussiennes, exprimant la probabilité d'appartenance d'une couleur à l'un ou l'autre. Une limitation de cette modélisation est que dans le cas où l'ensemble des couleurs à modéliser est trop important, elle peut ne pas être suffisament discriminante pour mener à une segmentation correcte. C'est notamment l'un des risques lorsque l'on tente de segmentation plusieurs objets différents, mais représentés par un seul modèle.

Ainsi, nous allons procéder à la segmentation de chacun des blobs du groupe $\mathcal{F}_0$ individuellement. Chaque blob sera dorénavant noté $\mathcal{F}^i$, i étant l'index du blob courant. A partir d'ici, la suite de notre méthode ne s'adressera qu'à un blob à la fois. De plus, s'assurer que les points (2) et (3) de l'énumération précédente sont respectés, nous avons choisis d'imposer une taille minimale aux blobs, taille en dessous desquels nous considérons qu'il s'agit de faux positifs.

Chaque $\mathcal{F}^i$ est encadré par un ensemble de pixels appartenant à $\mathcal{O}_0$ ainsi qu'à $\mathcal{B}_0$. Une portion de ces deux ensembles est sélectionnée, au voisinage de $\mathcal{F}^i$, de telle manière qu'elle inclue l'ensemble des pixels de $\mathcal{O}_0$ dans ce voisinage, ainsi qu'un nombre suffisant de pixels de $\mathcal{B}_0$ pour que le modèle de l'arrière-plan soit cohérent. Nous avons choisi comme règle de sélectionner dans ce dernier groupe un nombre de pixels équivalent à celui de $\mathcal{F}^i$. Les deux nouveaux ensembles ainsi formés sont notés $\mathcal{O}^i$ et $\mathcal{B}^i$.

Finalement, au lieu d'une seule graine nous avons autant de graines que d'objets. Chaque graine est composée des ensembles $\mathcal{F}^i$ d'où est tirée la modélisation de l'avant-plan (c'est à dire de l'objet), $\mathcal{B}^i$ d'où est tirée la modélisation de l'arrière-plan, et $\mathcal{O}^i$ sur lequel sera faite la segmentation finale.

4.4 Expérimentations

4.4.1 Mise en oeuvre du matériel

Afin d'évaluer notre algorithme dans de multiples conditions, nous avons mis en place un montage comprenant deux caméra identiques pour en tirer une reconstruction stéréo. Les caméras RGB sur système stéréo sont des GC1910C de Prosilica, dotées d'un

capteur d'une résolution de 1920x1080 et associées à des optiques de 16mm offrant un champ de vision de 35 degrés.

En sus de l'utilisation d'un couple de caméras en stéréo pour produire la carte de profondeur de la scène, nous avons utilisé une caméra mesurant directement la profondeur des objets à son plan focal, en l'occurrence un Kinect de Microsoft. Outre la facilité d'accès à cette caméra (s'agisant d'un produit grand public), le Kinect a les caractéristiques suivantes :

- Résolution de la carte de profondeur de 640x480, codée sur 12 bits
- Caméra RGB d'une résolution maximale de 1280x800 (non utilisée)
- Champ de vision horizontal de 58 degrés
- Portée de 3.5m

La résolution en particulier est très intéressante puisque bien supérieure aux z-caméras habituelles. Ceci est du à la technologie utilisée : alors qu'en général le principe de la mesure du temps de vol est implémenté (temps pour qu'un signal fasse un aller-retour entre l'émetteur, l'objet distant et le capteur), le Kinect mesure la déformation d'un motif projeté (en infra-rouge) sur les objets distants. Nous n'avons pas utilisé la caméra RGB puisque nous souhaitions mettre en relation les résultats de l'utilisation du Kinect avec ceux mettant en oeuvre la reconstruction stéréo, qui nécessite donc deux caméras.

4.4.1.1 Calibrage

Avant de pouvoir utiliser la mesure fournie par la z-camera comme une graine (en lieu et place de la reconstruction stéréo), l'ensemble de l'assemblage des trois caméras a été calibré. Ce calibrage consiste à déterminer les paramètres suivants :

- Paramètres intrinsèques de chaque caméra (focale et centre optique)
- Paramètres extrinsèques de chaque caméra (déformations optiques)
- Matrices de translation décrivant les positions des caméras les unes par rapport aux autres
- Matrices de rotation décrivant l'orientation des caméras

Deux outils sont couramment utilisés pour calibrer des caméras : le *Camera Calibration Toolbox* pour Matlab ([2]), et les outils dédiés dans la librairie OpenCV (accessible en C/C++ et Python). La méthode est la même dans les deux cas : il s'agit de prendre un ensemble vues d'un damier (connu) dans diverses positions, du moment que celui-ci est visible sur l'ensemble des caméras à calibrer. Nous avons utilisé OpenCV, par simplicité du fait que l'ensemble du code est en C++.
L'opération a été faite en deux temps, les algorithmes implémentés dans OpenCV ne permettant de calibrer que deux caméras ensemble. Comme cela a été précisé auparavant, la caméra de gauche a été choisie comme référence pour la reconstruction stéréo. Les deux autres caméras ont donc été calibrées successivement avec la caméra de gauche.
Précisons que dans le cas de la z-caméra, le calibrage s'est fait à partir de l'image infra-rouge, le damier étant alors éclairé par une lampe à incandescence. De plus, le champ de vision de la z-caméra étant bien supérieur à ceux des caméras RGB (58 contre 35 degrés), nous sommes contraint au moment du calibrage de la z-caméra avec la caméra de gauche de placer le damier dans la zone centrale de la z-caméra sans quoi

il sortirait du champ de vision de l'autre caméra. Les déformations optiques ont donc été ré-évaluées séparément en considérant la z-caméra seule.
Les captures issues de chaque caméra sont corrigées optiquement (selon les paramètres intrinsèques et extrinsèques) avant d'être utilisées ultérieurement.

4.4.1.2 Correction de la carte de profondeur

La carte de profondeur issue de la reconstruction stéréo a pour référence le point focal de la caméra RGB de gauche. Par défaut, la carte de profondeur de la z-caméra a pour référence le point focal de cette même z-caméra. Pour pouvoir utiliser cette dernière carte de profondeur comme une graine dans la segmentation d'images issues de la caméra RGB de gauche, il est nécessaire de corriger le point de référence.
Le calibrage précédent nous fournit l'ensemble des éléments dont nous avons besoin pour y parvenir, c'est à dire les matrices de translation et de rotation entre les deux caméras. La position de chaque point de la carte de profondeur est calculée dans le repère de la z-caméra, pour être projetée ensuite selon les caractéristiques de la caméra RGB gauche.

La carte de profondeur étant donnée pour être la distance des objets au plan focal de la caméra, la position des points détectés dans le repère de la caméra est la suivante :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{zcam} = \begin{bmatrix} \frac{X.D}{f} \\ \frac{Y.D}{f} \\ D \end{bmatrix}_{zcam}$$

On obtient finalement dans le repère de la caméra RGB :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{RGB} = \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix}_{RGB} * \begin{bmatrix} \frac{X.D}{f} \\ \frac{Y.D}{f} \\ D \end{bmatrix}_{zcam} + \begin{bmatrix} x_{zcam} \\ y_{zcam} \\ z_{zcam} \end{bmatrix}_{RGB}$$

La projection sur la caméra RGB (simulée) se fait alors de la manière suivante :

$$\begin{bmatrix} Y \\ X \\ 0 \end{bmatrix}_{RGB} = \begin{bmatrix} \frac{x.f}{z} \\ \frac{x.f}{z} \\ 0 \end{bmatrix}_{RGB}$$

Dans le cas de notre montage, le calibrage indique que la rotation entre les caméras est inférieure à 1 degré sur chacun des trois axes, équivalent à un décalage d'une dizaine de pixels ce qui est non négligeable et mérite d'être corrigé. La figure 4.10 est une illustration de cette correction : la capture issue de la z-caméra est corrigée pour obtenir une carte de profondeur dans le repère de la caméra RGB.

4.4.2 Conditions expérimentales et hypothèses

La caractéristique sur laquelle nous souhaitons mettre l'accent, et dont la prise en compte a mené à la méthode présentée dans ce travail, est la robustesse de la segmentation vis à vis des changement d'éclairage. Dans une moindre mesure, la robustesse

Figure 4.10 – Gauche : carte de profondeur issue de la z-caméra non corrigée.
Centre : carte de profondeur de la z-caméra corrigée
Droite : carte de profondeur issue de la stéréo, pour référence

aux modifications mineures de l'arrière-plan est également considérée. Les expérimentations visent donc à mettre en lumière ces deux caractéristiques, et deux scènes ont été capturées dans ce but.

La première scène se concentre sur une modification de l'éclairage sur une scène immobile, dans laquelle est placée un objet devant être détouré. Les conditions d'éclairage suivantes ont été capturées (voir également figure 4.11), la première étant celle utilisée comme condition de référence, à partir de laquelle les modèles ont été créés (mélange de gaussiennes et *codebook*) :

1. Un spot doté d'un filtre (pour se rapprocher d'une lumière blanche).
2. Ce même spot, sans filtre.
3. Eclairage naturel
4. Lampes de droite du banc de mesure
5. Toutes les lampes du banc de mesure
6. Lampes de gauche du banc de mesure

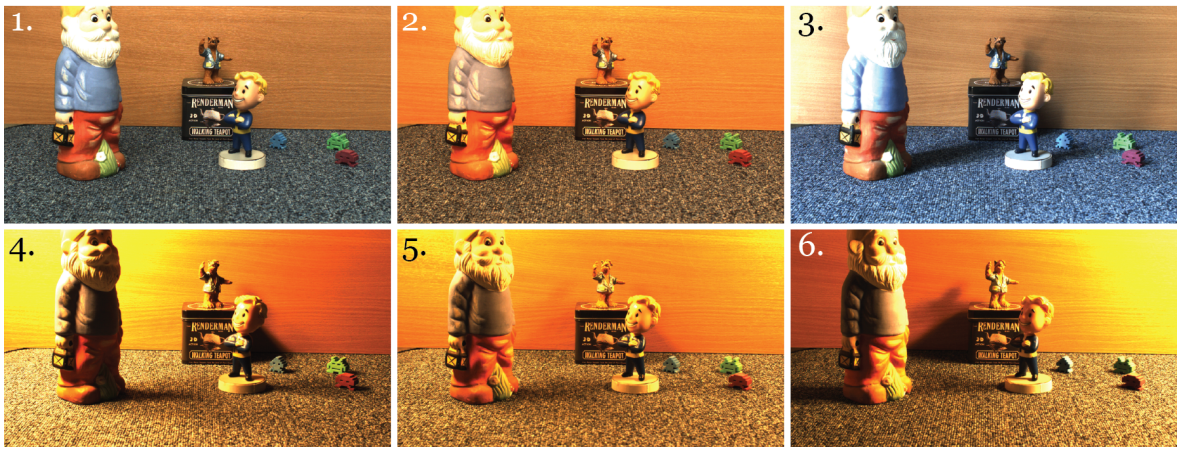


Figure 4.11 – Conditions d'éclairage pour l'évaluation de la segmentation
1. Spot filtré ; 2. Spot non filtré ; 3. Naturel
4. Lampes de droite du banc ; 5. Toutes les lampes du banc ; 6. Lampes de gauche

La seconde scène met en oeuvre un arrière-plan mobile. Pour ce faire, nous avons ajouté à notre scène un écran sur lequel nous avons diffusé deux vidéos (figure 4.12).

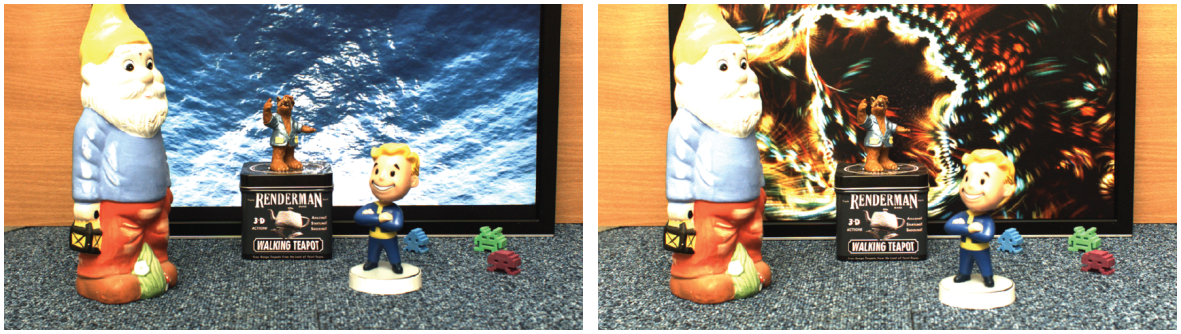


Figure 4.12 – Segmentation avec un arrière-plan mobile

La première est une boucle d'une simulation de vagues [1]. Chaque pixel prend au cours de cette boucle un nombre limité de valeurs, ce qui est adapté à une modélisation par un mélange de gaussiennes ou un dictionnaire. La seconde vidéo est une animation de fractales. Les pixels peuvent prendre dans ce dernier cas des valeurs semblant aléatoires. Les résultats sont regroupés dans la figure 4.15 (voir également la figure 4.14).

En supplément de ces deux tests synthétiques, des expérimentations dans le cadre de scène "naturelles" ont été menées afin de mesurer les performances de l'implémentation de notre méthodes, pour juger de son utilisabilité dans des applications de réalité mixte. Pour nous assurer de mesurer les performances de notre algorithme et non celles de celui-ci ajouté à l'algorithme de reconstruction stéréo, nous avons fait ces essais exclusivement dans le cadre de l'utilisation conjointe de la z-caméra à l'une des caméra RGB.

Cependant, afin de déterminer l'impact de la qualité de la carte de profondeur sur la segmentation finale, un test synthétique a été fait spécifiquement.

4.4.3 Mesures expérimentales

4.4.3.1 Variations de l'éclairage

La figure 4.13 présente les résultats de ces essais. En dehors de la qualité de la segmentation, qui est pour les conditions de départ de qualité sensiblement identique pour les trois méthodes employées, on remarque immédiatement que les méthodes à base de mélange de gaussiennes et de *codebook* supportent très mal les changements d'éclairage. Au contraire, notre méthode hybride présente des résultats constants quel que soit l'éclairage. A noter que les paramètres utilisés sont les mêmes pour toutes les conditions lumineuses.

1. Simulation d'eau par TimmY, http ://vimeo.com/13355187

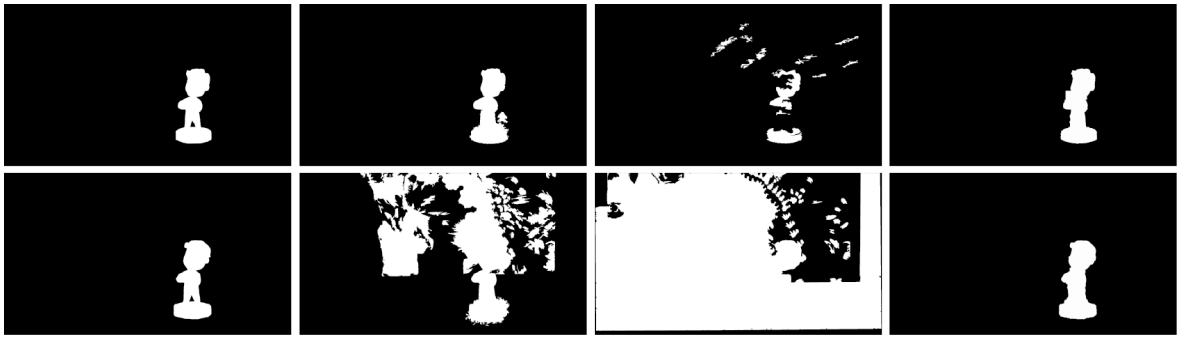
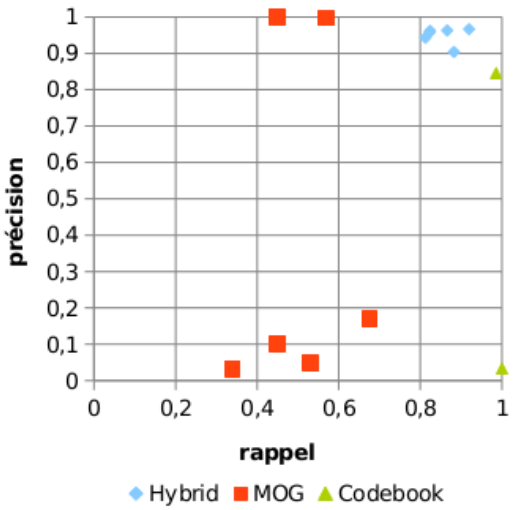


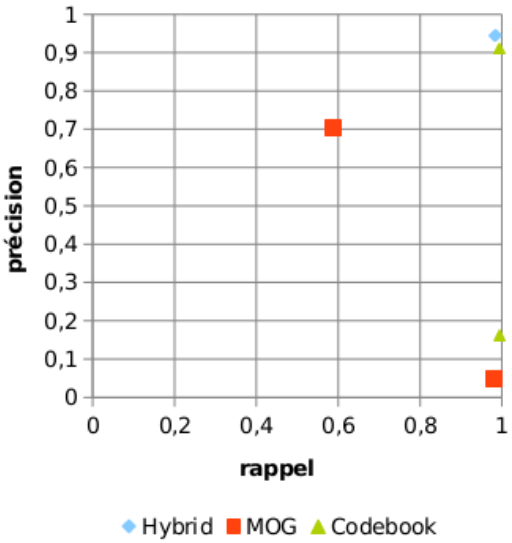
Figure 4.14 – Haut : simulation d'eau ; Bas : fractales
De gauche à droite : vérité terrain / dictionnaire / mélange de gaussiennes / hybride



	Hybrid		MOG		Codebook	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
spot_blue_filter	0,957275	0,822269	0,999598	0,447796	0,845837	0,98602
spot_no_filter	0,961818	0,824221	0,997311	0,567971	0,0334496	1
bench_left	0,903701	0,882559	0,047843	0,531977	0,033444	1
bench_right	0,943213	0,812584	0,0333077	0,33911	0,0334393	1
bench_both	0,962891	0,866146	0,102575	0,450334	0,0334403	1
natural	0,966397	0,919408	0,17028	0,675636	0,033446	1

Figure 4.13 – Résultat de la segmentation après changement d'éclairage

4.4.3.2 Arrière-plan mobile



	Hybrid		MOG		Codebook	
	Précision	Rappel	Précision	Rappel	Précision	Rappel
water	0,943987	0,982244	0,704545	0,587377	0,910506	0,99374
fractals	0,945274	0,985413	0,0493559	0,980896	0,16216	0,994634

Figure 4.15 – Résultat de la segmentation avec un arrière-plan mobile

Comme attendu, en ce qui concerne les modélisations par gaussiennes et par dictionnaire, les résultats sont de beaucoup moins bonne qualité lorsque la vidéo de fractales est diffusée par rapport à la vidéo de vagues. Notre approche ne souffre pas de ce problème et la qualité de la segmentation est sensiblement identique.

4.4.3.3 Influence de la carte de profondeur

Du fait de la conception même de notre méthode, la qualité de la carte de profondeur a une importance capitale sur la segmentation finale. Une carte de profondeur parfaite permettrait de segmenter l'avant-plan, sans même avoir besoin de passer par une segmentation dans l'image RGB pour l'améliorer. Une carte de profondeur d'excellente qualité (sans être parfaite) permet elle de réduire l'espace entre la graine de l'avant-plan et celle de l'arrière-plan avec pour intérêt de limiter les éventuelles erreurs dues à un contraste trop faible.

Dans notre cas, la carte de profondeur est issue soit de la reconstruction stéréo, soit de la captation par une z-caméra. Ces deux techniques restent limitées notammment lorsqu'il s'agit de décrire correctement les contours des objets. Elles sont de plus très sensibles au matériau des objets, les surfaces très spéculaires étant propices à des erreurs de mesure. D'où l'intérêt de filtrer notre graine issue de la différence entre la carte de profondeur mesurée à l'instant t et celle mesurée à l'instant 0, qui décrit l'arrière-plan.

Notre méthode hybride peut se contenter d'une graine très éloignée de la segmentation cible (c'est à dire de la vérité terrain). Le filtrage que nous appliquons à la différence entre les cartes de profondeur permet d'éliminer les faux positifs de celle-ci,

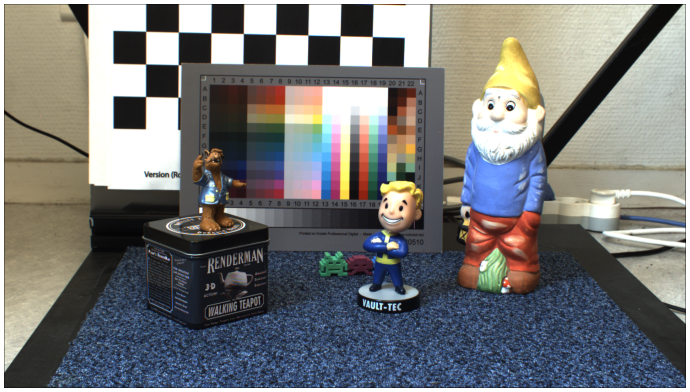


FIGURE 4.16 – Scène pour les tests de l'influence de la qualité de la carte de profondeur

ce qui est très important puisque ces faux positifs seraient conservés dans la segmentation finale (en plus de dégrader la justesse de la modélisation par des gaussiennes de l'avant-plan). Les figure 4.17 et 4.18 présentent les résultats d'un essai où le lissage de la graine (érosion) varie, en prenant comme source pour la carte de profondeur la reconstruction stéréo et la z-caméra. Nous avons fait ces essais sur une scène légèrement différente des deux essais précédents, présentant d'importants applats de couleur peu texturés pour mieux mettre en avant les problèmes liés aux faibles contrastes 4.16.

	Stereo		Z-camera	
	Précision	Rappel	Précision	Rappel
1	0,580566	0,894976	0,6554	0,876221
3	0,62337	0,869502	0,672376	0,857097
5	0,657317	0,869256	0,705997	0,836201
7	0,705175	0,85215	0,730204	0,808708
9	0,723526	0,845209	0,759894	0,762288
11	0,742126	0,811317	0,796276	0,732679
13	0,779984	0,811219	0,802566	0,717566
15	0,780714	0,788132		
17	0,762817	0,664698	0,848494	0,337714
19	0,81582	0,643014		
21	0,816085	0,64872		
23	0,794011	0,440498		

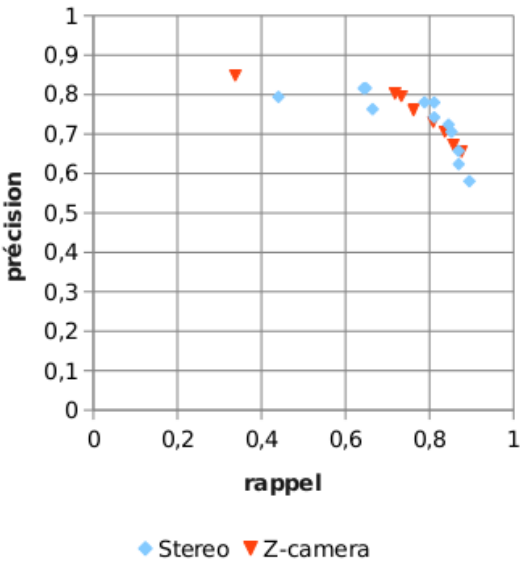


FIGURE 4.17 – Influence de la carte de profondeur sur le résultat final. Le paramètre modifié est l'érosion de la graine (de 1 à 23 pixels)

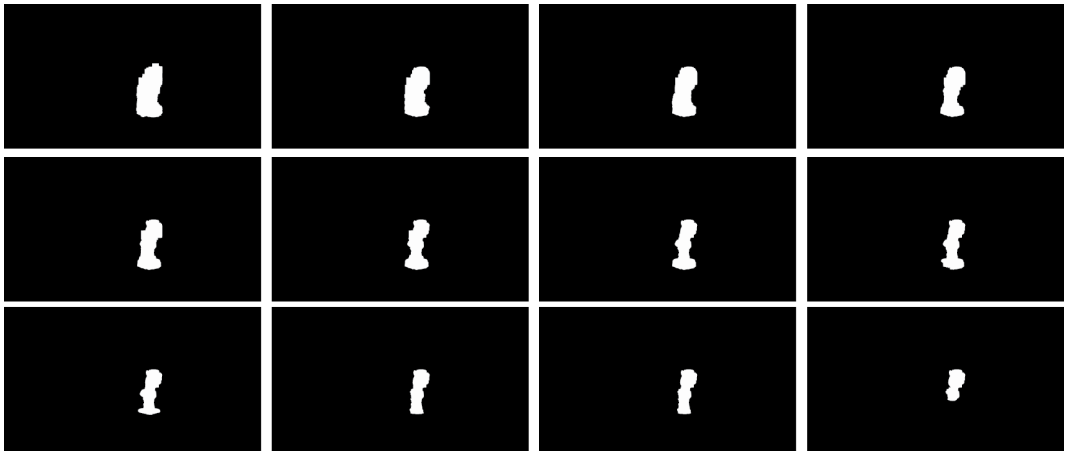


Figure 4.18 – Résultat de la segmentation, selon l'érosion de la graine (de 1 à 23 pixels par pas de 2)

On voit clairement que nos deux sources imposent de filtrer la graine pour maximiser la qualité de la segmentation. Une érosion trop importante présente cependant le risque de devoir créer une modélisation avec peu de pixels (graine de l'avant-plan peu représentative de l'ensemble), et d'avoir une graine de l'arrière-plan qui inclut des éléments de l'avant-plan (selon la distance choisie entre les graines de l'arrière et de l'avant plan).

Au sujet de la distance entre les graines de l'arrière-plan et de l'avant-plan, il est nécessaire qu'elle soit aussi proche que possible, et ce pour deux raisons. La première concerne principalement les arrière-plans chargés de détails : une graine de l'arrière-plan trop éloignée risque de ne pas décrire correctement l'arrière-plan proche de l'objet à segmenter. La seconde raison est que les coûts liés aux données sont pondérés selon la distance du pixel considéré aux graines de l'arrière-plan et de l'avan-plan. Le changement de la distance entre ces deux graines modifie donc les coûts des pixels intermédiaires et influe sur la segmentation.
A titre d'indication, la figure 4.19 illustre l'évolution de la qualité de la segmentation en fonction de la distance entre les graines, sur la même scène que pour le test précédent, et pour une valeur de lissage de 13. Le test précédent a été réalisé avec une distance fixe de 40 pixels.

	Stereo	
	Precision	Recall
10	0,844191	0,594501
20	0,779374	0,753132
30	0,779984	0,811219
40	0,74087	0,823402
50	0,705318	0,857565
60	0,696678	0,857811
70	0,63743	0,868075
80	0,555342	0,868148
90	0,475084	0,871963

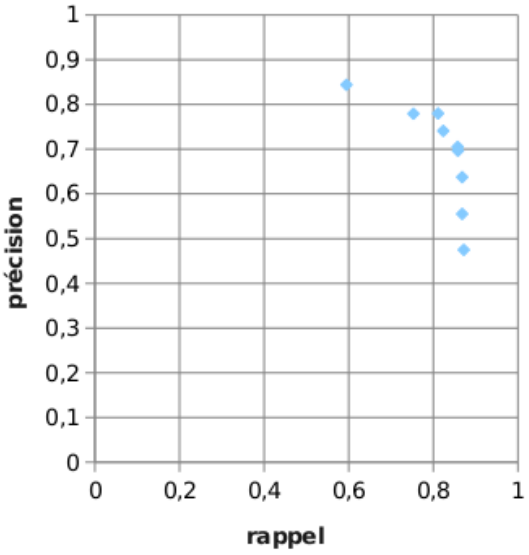


Figure 4.19 – Influence de la distance entre les graines de l'arrière et de l'avant-plan (donnée en pixels)

4.4.3.4 Performances : observations et discussion

En gardant à l'esprit que le but premier de notre approche est d'être utilisée dans le cadre d'applications temps réel, nous avons soumis notre implémentation de celui-ci à un test de performance dans des conditions réelles. Ce test a consisté à exécuter la segmentation pendant une durée relativement longue (environ une heure), en mesurant la durée de chacune des phases de la segmentation qui sont les suivantes :

- capture
- pré-segmentation selon la carte de profondeur
- création du mélange de gaussienne
- segmentation finale.

La durée d'un cycle de segmentation dépendant de la surface de l'objet à segmenter, les mesures relevées ont été évaluées selon ce paramètre principalement. Sur la figure 4.20 la durée totale d'un cycle est également donnée. L'ordinateur ayant servi pour ce test est composé principalement d'un processeur Intel Core i5 à 3.4GHz ainsi que d'un processeur graphique de type GTX 670, de NVidia.

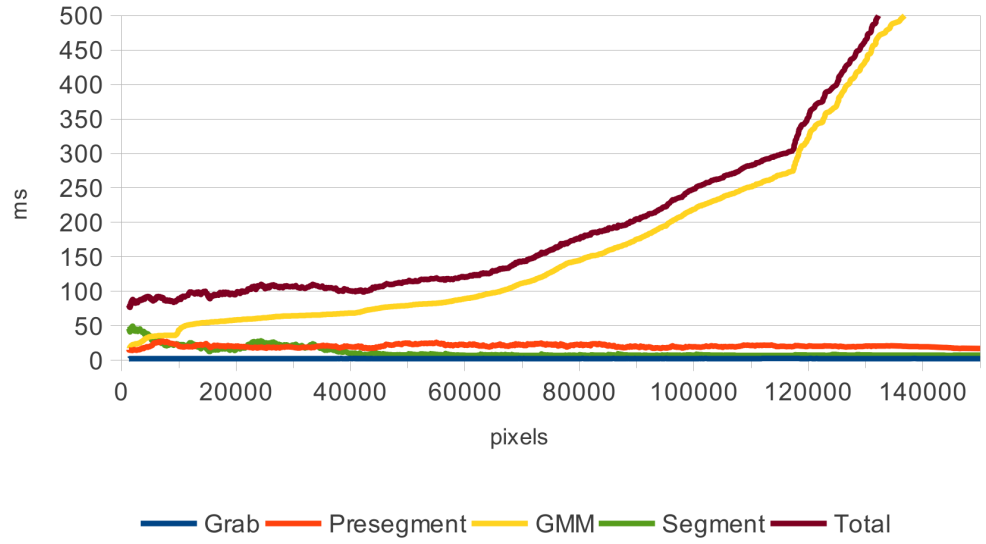


Figure 4.20 – Temps de chacune des étapes de la segmentation, selon la surface de la zone considérée

Il est clair à la vue de la figure 4.20 que la phase de création des modèles de l'arrière et de l'avant-plan sont de loin les plus longues, avec une augmentation exponentielle avant la surface de l'objet à segmentation. En revanche, les autres phases ont des durées beaucoup plus réduites et une durée totale cumulée (hors création des modèles) d'environ 30ms. Notons également que les trois phases de capture, pré-segmentation et segmentation finale affichent des durée globalement constantes, si ce n'est des durées anormalement élevées pour la segmentation finale lorsque l'objet est particulièrement petit. Ceci est sans doute à mettre sur le compte de modèles peu représentatifs des avant et arrière plans du fait d'un nombre de pixels trop réduits pour les construire.

4.5 Analyse et discussion

Nous avons présenté dans ce chapitre une méthode de segmentation tirant profit de la disponibilité de la carte de profondeur associée à un point de vue pour automatiser une classe d'algorithmes de segmentation, qui dans leurs implémentations standards imposent l'intervention de l'utilisateur.

Notre proposition est un premier jet qui pose les bases de ce type de méthodes. Les premiers résultats sont encourageants et démontrent la capacité de cette méthode à passer outre les problèmes habituels liés aux changements lumineux et aux arrière-plans mobiles, là où des méthodes plus courantes ne permettent pas de segmenter correctement les objets de l'avant-plan.

De nombreuses pistes d'améliorations sont envisageables. Outre l'utilisation de sources de meilleures qualité pour la carte de profondeur, ou d'un algorithme de segmentation de meilleure qualité, la création de la graine peut être largement améliorée en adaptant par exemple ses paramètres aux dimensions de l'objet détecté. Ou en utilisant mieux la carte de profondeur pour diriger la segmentation : en considérant un

objet caché partiellement par l'arrière-plan (derrière une colonne par exemple), une amélioration serait de forcer l'appartenance de cette colonne à l'arrière-plan lors de la segmentation.

L'implémentation en particulier mériterait de voir l'ensemble de l'algorithme exécuté sur le processeur graphique. Actuellement seule la segmentation finale est exécutée sur le processeur graphique, ce choix ayant été fait après avoir observé qu'il s'agissait de la phase la plus gourmande de notre méthode. Maintenant, la phase de création des modèles semble être le nouveau point limitant et mériterait certainement de profiter de la puissance offertes par ces architectures massivement parallèles. C'est d'autant plus envisageable que la phase de segmentation semble en l'état avoir une durée constante quelque soit la taille de l'objet, ce qui suggère que le processeur graphique est sous utilisé.

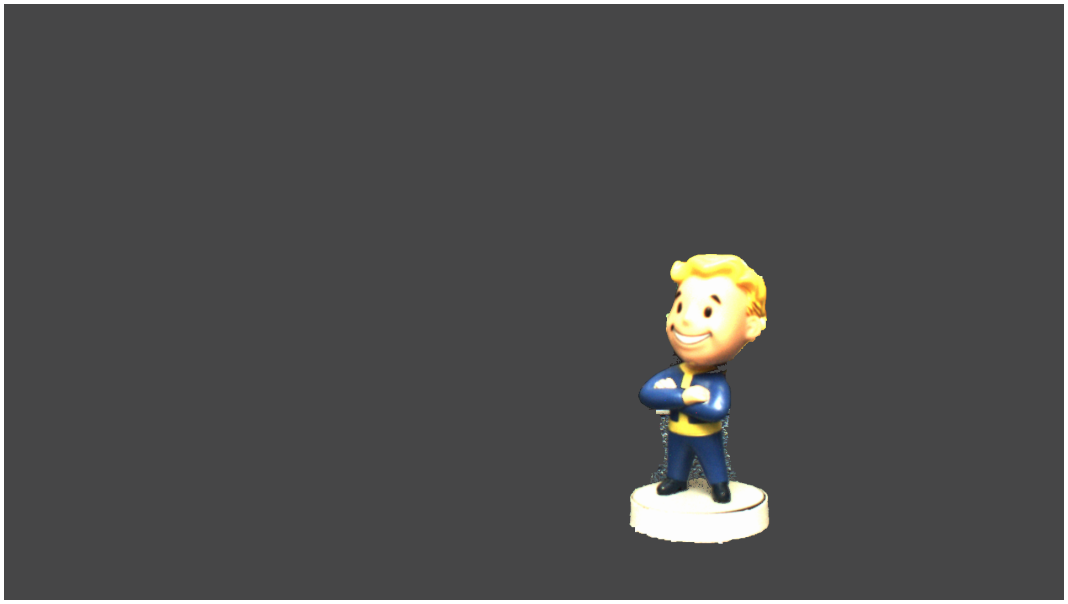


Figure 4.21 – Masque de segmentation appliqué à l'image RGB, issue de la scène avec arrière-plan mobile "fractals"

Soulignons enfin que notre méthode permet, outre l'automatisation d'algorithmes de segmentation parmi ceux offrant la meilleure qualité, de connaître la profondeur des objets segmentés. La segmentation associée à l'information de position dans l'espace des objets offrent alors la possibilité d'intégrer les objets en question dans un environnement virtuel de manière cohérente et réaliste, du moins en ce qui concerne la géométrie.

Conclusion

Pour reprendre les observations faites dans la description de notre problématique, une meilleure connaissance de l'environnement réel de l'utilisateur permet d'envisager des applications de réalité mixte plus avancées. Nous nous sommes intéressé dans ce mémoire à deux aspects de cette environnement :

- son éclairage
- la segmentation entre l'arrière-plan et les objets mobiles qui constituent l'avant-plan.

Le travail de thèse ayant pris place dans le cadre du développement du dispositif ray-on, une hypothèse simplificatrice très importante est la fixité du point de vue en translation. Cette hypothèse nous a permis d'envisager des méthodes que la détection de la position aurait largement complexifiées, d'autant plus que cet aspect constitue un domaine de recherche à part entière. Quoi qu'il en soit, le champ des applications de la réalité mixte utilisant un point de vue fixe est suffisamment vaste pour justifier ce choix.

Concernant le premier aspect, nous avons choisi d'étendre les capacités des algorithmes dits *Image Based Rendering*, en les combinants avec une mesure photométrique de l'éclairage ambiant. Notre méthode consiste à associer à chaque direction mesurable sur notre capture de l'éclairage une image haute dynamique de notre scène, d'un point de vue fixe, éclairée par une lumière directionnelle unitaire (dans notre cas $1W.m^{-2}.sr^{-1}$). L'ensemble des images ainsi créées forme une base de données.

La reproduction de l'éclairage réel dans notre scène virtuelle se fait alors en ajoutant l'ensemble des images de la base de données, pondérées par la valeur photométrique (toujours en $W.m^{-2}.sr^{-1}$) de l'éclairage mesurée dans la direction associée. Le résultat est alors une image haute dynamique de la scène dont l'unité d'expression des valeurs des pixels est cohérente avec la réalité physique.

Nos expérimentations ont démontré l'apport d'une reproduction précise de l'éclairage, tout en illustrant les limites de notre approche. Une reproduction photométrique de l'éclairage peut faciliter les opérations de compositing entre sources réelles et virtuelles, en particulier dans le cadre de conditions d'éclairage difficiles liées à une dynamique importante ou une température d'éclairage très éloignée de 5454°K qui est la température de l'illuminant neutre E.

Cependant des limitations sont apparues dans les possibilités de reproduction de l'éclairage, visibles en particulier dans les tons bleus lors de nos mesures. La piste la plus évidente et la plus simple à vérifier est qu'un gammut limité des caméras empêche la capture et la reproduction correcte de ces tons. Il serait intéressant de vérifier cette piste soit en utilisant des caméras plus performantes, soit en utilisant comme sources

un appareil photo.

La segmentation, qui dans notre cas consiste en la détection des objets mobiles c'est à dire n'appartenant pas à l'environnement, a été considérée selon le même parti pris du point de vue fixe. Nous nous sommes concentrés sur le problème de la gestion des changements d'éclairage dans la scène, afin d'obtenir une méthode donnant des résultats très proches quel qu'il soit.

Notre méthode se déroule en deux temps. Tout d'abord, une approximation de la segmentation est créée en effectuant une opération de type soustraction de l'arrière plan (*background subtraction*) dans le domaine de la profondeur au lieu de traditionnels canaux de couleur ou niveaux de gris. L'information de profondeur, capturée par une z-camera ou un couple de caméras stéréo, est totalement indépendante de l'éclairage et donc cette approximation de la segmentation l'est également.

Le résultat de cette première opération est utilisée comme une graîne pour créer une modélisation colorimétrique de l'arrière et de l'avant-plan. Celle-ci est alors utilisée conjointement avec la segmentation approximative dans le cadre d'un algorithme de segmentation basé sur les *graph-cuts* afin de produire une segmentation plus précise. La modélisation étant produite à chaque itération à partir de la graîne, sans tenir compte des itérations précédentes, la segmentation finale reste indépendante des modifications d'éclairage.

Par extension, notre méthode ne détecte un élément comme appartenant à l'avant-plan que lorsque sa profondeur est différente de celle normalement mesurée pour l'arrière-plan. Ainsi si ce dernier est mobile dans les limites du seuil de détection, notre méthode ne produira pas de faux positifs. Cela vaut pour les écrans diffusants du contenu visuel comme décrit dans les expérimentations, mais également pour les feuillages ou tout élément dont on considère qu'une distance excessive au point de vue implique l'appartenance à l'arrière-plan.

Les méthodes proposées permettent d'envisager des applications de réalité mixte réalistes et immersives, où il serait peu évident de discrimminer le virtuel du réel. La principale contrainte de ces méthodes est que le point de vue est nécessairement fixe. Les possibilités offertes par l'utilisation conjointe des deux méthodes sont par exemple, en excluant les applications non photoréalistes :

- l'intégration d'un objet virtuel dans une scène réelle, avec une reproduction de l'éclairage réel sur cet objet
- la segmentation d'un objet réel pour son intégration dans une scène virtuelle, celle-ci pouvant être éclairée à l'identique pour une meilleure intégration.

En sus des améliorations dans la qualité de la segmentation et dans la reproduction de l'éclairage (des pistes ayant été données à ce sujet dans les chapitres correspondants), il serait intéressant de profiter de la connaissance de l'éclairage pour annuler l'éclairage porté sur les objets segmentés, afin de les éclairer avec de nouvelles sources. Ceci ouvrirait la voie à de nouvelles applications photoréalistes de réalité mixte.

De même, une évolution de la reproduction de l'éclairage pourrait être de passer à une forme moins contraignante de *Image Based Rendering*, permettant par exemple un

léger déplacement dans le plan de l'écran. En captant le déplacement de l'utilisateur face à l'écran, il serait alors possible de lui faire ressentir la profondeur de la scène.

Enfin, l'extension à des méthodes d'éclairage indépendantes de la position n'a pour l'instant que peu d'intérêt, du moins tant que les techniques de segmentations ne permettent pas de déplacer le point de vue à volonté. Cependant, on peut envisager d'utiliser notre méthode de segmentation conjointement avec un modèle géométrique précis de l'environnement dans lequel on souhaite se déplacer. En considérant que la détermination de la position de l'utilisateur a un niveau de précision équivalent à celui de la carte de profondeur, il deviendrait possible de procéder à une soustraction entre la carte de profondeur mesurée en temps réel et la simulation de l'arrière-plan, ce qui de proche en proche permettrait la segmentation en tout point de l'espace.

Glossaire

BRDF *Bidirectionnal Reflectance Distribution Function.* Voir fonction de distribution de la réflectance bidirectionnelle.. 6, 27

capture de l'éclairage (ou *light probe*). Image de l'environnement lumineux, en format hémisphérique ou latitude-longitude la plupart du temps. Pour de meilleurs résultats, on utilise un format de pixel haute dynamique. 21, 30, 41, 106

carte de profondeur Une image représentant, pour chaque pixel, la distance de l'objet correspondant au plan focal de la caméra. 73, 78, 79

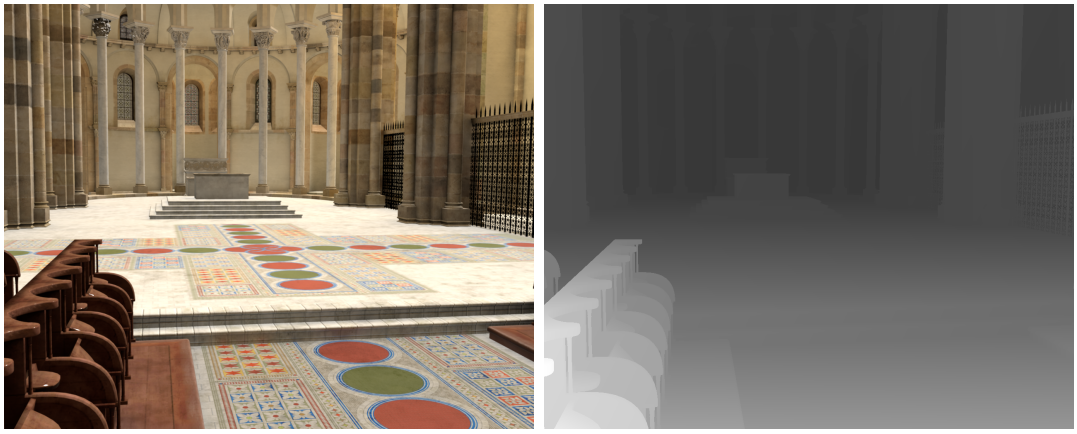
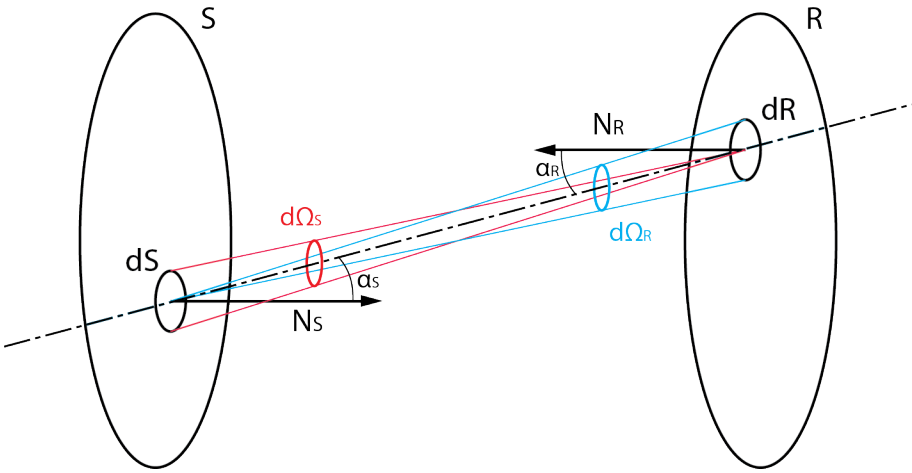


Image capturée par une caméra virtuelle, et carte de profondeur correspondante

éclairement Flux lumineux par unité de surface, arrivant en un point, en $lm.m^{-2}$. 41, 44

étendue géométrique Grandeur permettant de caractériser la dispersion d'un faisceau lumineux entre sa source et un récepteur. En posant :

- dS et dR les éléments de surface appartenant respectivement à la source lumineuse S et au récepteur R
- $\vec{\alpha_S}$ et $\vec{\alpha_R}$ les angles entre la direction allant de dS à dR et les normales à ces mêmes éléments de surface, $\vec{N_S}$ et $\vec{N_R}$
- $d\Omega_S$ et $d\Omega_R$ les angles solides de chacun des éléments de surface dS et dR vus depuis le centre de l'autre élément.



On a par définition l'étendue géométrique :

$$d^2G = d_S.\cos\vec{\alpha_S}.d\Omega_S = d_R.\cos\vec{\alpha_R}.d\Omega_R$$

7

exitance Flux lumineux par unité de surface, partant d'un point, en $lm.m^{-2}$. 41

exitance énergétique Quantité d'énergie rayonnée partant d'un point, par unité de surface. Elle s'exprime en $W.m^{-2}$. 6

flux lumineux Grandeur caractérisant la puissance lumineuse telle qu'elle est perçue par le système visuel humain, en lumen (lm). Il est calculé comme étant le produit du flux énergétique et de la réponse de l'oeil humain à ce flux. 7, 29, 34

flux énergétique Flux des radiations électromagnétiques émises par une source, ne se limitant pas aux longueurs d'onde visibles (en W). 6, 7

fonction de distribution de la réflectance bidirectionnelle Voir BRDF. Fonction à quatre dimensions décrivant la manière dont une surface réfléchit la lumière (direction de la réflexion) selon la direction du rayon lumineux incident. Elle s'exprime de la façon suivante :

$$f_r(\vec{\theta_{in}}, \vec{\theta_{out}}) = \frac{dL_{out}(\theta_{out})}{dE_{in}(\theta_{in})}$$

avec :
- $f_r(\theta_{in}, \theta_{out})$: en sr^{-1}
- $\vec{\theta_{in}}, \vec{\theta_{out}}$: directions incidente et réfléchie
- E_{in} : luminance énergétique incidente sur la surface selon la direction θ_{in}, en $W.sr^{-1}.m^{-2}$
- L_{out} : exitance énergétique réfléchie par la surface dans la direction θ_{out}, en $W.m^{-2}$

5, 6, 26, 27

lieu des corps noirs Aussi nommé lieu de Planck, ensemble des chromacités que prend la lumière émise par un corps noir à mesure que sa température augmente. 38

luminance Flux lumineux passant par une surface selon une direction donnée, exprimée en $lm.sr^{-1}.m^{-2}$, ou $cd.m^{-2}$ Elle est définie par :

$$L = \frac{d^2\Theta}{d^2G}$$

avec :
- L : luminance
- $d^2\Theta$: flux lumineux dans la direction considérée capté par la surface réceptrice, en lm
- d^2G : étendue géométrique reliant les deux surfaces émettrice et réceptrice.

7, 22, 23, 27, 28, 30

luminance énergétique Quantité d'énergie rayonnée passant par une surface selon une direction donnée, exprimée en $W.sr^{-1}.m^{-2}$. Elle est définie de manière similaire à la luminance, mais en considérant le flux énergétique en place du flux lumineux. 6

métrique En considérant un espace E, une métrique $d(.,.)$ (ou distance) est une fonction de E dans $\mathbb{R}^+$ respectant les contraintes suivantes :

$$\forall x, y \in E, d(x, y) = d(y, x)$$

$$\forall x, y \in E, d(x, y) = 0 \leftrightarrow x = y$$

$$\forall x, y, z \in E, d(x, y) < d(x, z) + d(z, y)$$

7, 87

ouverture Désigne, dans le cas d'une optique d'appareil photo ou de caméra, le rapport entre le diamètre de l'entrée et la focale : $N = \frac{f}{D}$ (également nommé *f-number* en anglais, ou f/N). Elle conditionne la quantité de lumière arrivant sur le capteur, et ce de manière indépendante par rapport à la focale de l'optique : deux optiques de focales différentes conduiront la même quantité de lumière au capteur si elles ont la même ouverture. 23

réflectance Proportion de flux réfléchi par rapport au flux reçu par une surface, sans unité. 28, 44

segmentation Consiste, en informatique graphique, à séparer les éléments composants une image selon des caractéristiques lexicales telles que le type d'objet, la couleur ou la texture, le comportement dans une suite d'images (en mouvement ou non par exemple), la position dans l'image ou dans la scène représentée par cette image, etc. 71

semi-métrique Voir métrique. Une semi-métrique est semblable à une métrique, à ceci prêt qu'elle n'impose pas la dernière relation (à savoir l'inégalité triangulaire). 87

sensibilité ISO Exprime la capacité d'un film argentique ou d'un capteur CCD ou CMOS à fournir une image selon les conditions d'illuminations : une sensibilité plus élevée indique que l'appareil sera plus à même de transcrire des faibles luminances qu'une sensibilité moindre. 23

shader Suite d'instructions destinée à être exécutée sur un processeur graphique. Un shader peut concerner par exemple la description du comportement d'un matériau, ou un ensemble de traitements à appliquer sur une image avant son affichage. Les implémentations recentes de *shaders* permettent également de modifier ou de créer de la géométrie. Les shaders sont aussi fréquemment utilisés pour profiter de la puissance des processeurs graphiques dans le cadre de calculs scientifiques. 16

vérité-terrain ou *ground truth*. En informatique graphique, la vérité-terrain désigne le résultat recherché lors d'une opération de traitement d'image. Par exemple, il peut s'agir de la carte de profondeur réelle d'une scène, ou de la segmentation réelle d'un objet placé devant un arrière-plan. 92

Table des figures

Liste des équations

Bibliographie

[1] International color consortium.

[2] Matlab camera toolbox.

[3] Microvision.

[4] Total immersion.

[5] ray-on, 2008.

[6] Layar, 2009.

[7] StreetMuseum, 2010.

[8] Wikitude, 2010.

[9] The stanford 3D scanning repository, 2011.

[10] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 1 :3–20, 1991.

[11] E. H. Adelson and J. Y. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), Feb. 1992.

[12] M. Aittala. Inverse lighting and photorealistic rendering for augmented reality. *The Visual Computer*, 26(6) :669–678, 2010.

[13] A. Bleiweiss and M. Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. *DAGM Workshop*, 2009.

[14] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *Computer Vision*, 2004.

[15] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[16] Y. Boykov and O. Veksler. Graph cuts in vision and graphics : Theories and applications. 2006.

[17] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE International Conference on Computer Vision*, 1999.

[18] A. Bugeau and P. Pérez. Detection and segmentation of moving objects in complex scenes. *Computer Vision and Image Understanding*, 113(4), 2008.

[19] J. Chen, G. Turk, and B. MacIntyre. Watercolor inspired non-photorealistic rendering for augmented reality. *VRST*, pages 231–234, 2008.

[20] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. 1997.

[21] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[22] M. Corsini, M. Callieri, and P. Cignoni. Stereo light probes. *Eurographics*, 2008.

[23] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. *CVPR*, 2008.

[24] M. Cristani and V. Murino. Background subtraction with adaptive spatio-temporal neighborhood analysis. *VISAPP International Conference on Computer Vision Theory and Applications*, 2 :484–489, 2008.

[25] G. Cutri, G. Naccarato, and E. Pantano. Mobile cultural heritage : The case study of locri. *International conference on Technologies for E-Learning and Digital Entertainment*, 2008.

[26] P. E. Debevec. Rendering synthetic objects into real scenes : Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *SIGGRAPH*, 1998.

[27] P. E. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *Siggraph*, pages 145–156, 2000.

[28] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. *Computer Graphics*, 1997.

[29] N. Dixit, R. Keriven, and N. Paragios. GPU-Cuts : combinatorial optimisation, graphic processing units and adaptive object extraction. 2005.

[30] O. Duchenne and J.-Y. Audibert. Fast interactive segmentation using color and textural information. 2006.

[31] A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision*, pages 751–767, 2000.

[32] H. Farid and E. P. Simoncelli. Range estimation by optical differentiation. *Journal of the Optical Society of America*, 1997.

[33] J. Fischer, D. Bartz, and W. Strasser. Stylized augmented reality for improved immersion. *Virtual Reality*, pages 195–202, Mar. 2005.

[34] L. Ford and D. Fulkerson. *Flows in network*. 1962.

[35] Y. Fu, J. Cheng, Z. Li, and H. Lu. Saliency cuts : An automatic approach to object segmentation. 2008.

[36] J. Gallego, M. Pardàs, and G. Haro. Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model. 2009.

[37] G. Gill. Argyllcms, May 2012.

[38] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *JACM*, 1988.

[39] G. Gordon, T. Darrel, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. *Proceedings of CVPR*, 1999.

[40] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *SIGGRAPH*, pages 43–54, 1996.

[41] W. Grimson. Computational experiments with a feature based stereo algorithm. *MIT*, 1984.

[42] P. Haeberli. Synthetic lighting for photography, 1992.

[43] U. Hahne and M. Alexa. Combining time-of-flight depth and stereo images without accurate extrinsic calibration. *ACM*, 2008.

[44] M. Haller. Photorealism or/and non-photorealism in augmented reality. *VRCAI*, 2004.

[45] M. Harville, G. Gordon, and J. Woodfill. Adaptive video background modeling using color and depth. 2001.

[46] M. Harville, G. Gordon, and J. Woodfill. Foreground segmentation using adaptive mixture in color and depth. 2001.

[47] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. *IEEE International Conference on Computer Vision*, 1999.

[48] M. Inanici and J. Galvin. Evaluation of high dynamic range photography as a luminance mapping technique. *Lawrence Berkeley National Laboratory*, 2004.

[49] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. 1998.

[50] P. KaewTraKulPong and R. Rowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Proceedings of the Second European Workshop on Advanced Video Based Surveillance Systems*, pages 149–158, 2001.

[51] T. Kakuta, T. Oishi, and K. Ikeuchi. Fast shading and shadowing and handling occlusions for asuka-kyo MR contents. *Information Processing Society of Japan*, 2009.

[52] T. Kanade, H. Kano, and S. Kimura. Development of a video-rate stereo machine. *IROS*, 1995.

[53] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window : theory and experiment. 1994.

[54] D. A. Kerr. New measures of the sensitivity of a digital camera. 2007.

[55] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. 2003.

[56] Y.-S. Kim, B.-h. Cho, B.-s. Kang, and D.-I. Hong. Color temperature conversion system and method using the same, 2006.

[57] T. Ko, S. Soatto, and D. Estrin. Warping background subtraction. 2010.

[58] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. 2005.

[59] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. *IEEE International Conference on Computer Vision*, 2001.

[60] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts ? 2004.

[61] J.-F. c. Lalonde, A. A. Efros, and S. G. Narasimhan. Estimating natural illumination from a single outdoor image. *IEEE International Conference on Computer Vision*, 2009.

[62] D.-Y. Lee, J.-K. Ahn, and C.-S. Kim. Fast background subtraction algorithm using two-level sampling and silhouette detection. IEEE, 2009.

[63] J. Leens, S. Piérard, and O. Barnich. Combining color, depth, and motion for video segmentation. *Computer Vision Systems*, 2009.

[64] V. Lempitsky, P. Kohli, and T. Sharp. Image segmentation with a bounding box prior. 2009.

[65] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007.

[66] J. Lewis. Fast normalized cross-correlation. *Vision Interface*, 1995.

[67] D. Liu, K. Pulli, L. G. Shapiro, and Y. Xiong. Fast interactive image segmentation by disciminative clustering. ACM, 2010.

[68] Y. Liu, X. Qin, S. Xu, E. Nakamae, and Q. Peng. Light source estimation of outdoor scenes for mixed reality. *The Visual Computer*, 25(5) :637–646, 2009.

[69] M. Maria. Little CMS, 2012.

[70] J. Marsot, F. Gardeux, and V. Govaere. Réalité augmentée et prévention des risques. *INRS*, 2009.

[71] L. Matthies and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *Computer Vision*, 1989.

[72] P. Milgram and H. C. Jr. A taxonomy of real and virtual world display integration. *International Symposium on Mixed Reality*, 1999.

[73] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 302–309, 2004.

[74] T. Muller, J. Roger, and J.-M. Sanchez. Illumination photo-realiste interactive en environnement distant. 2003.

[75] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. 2005.

[76] J. S. Nimeroff, E. P. Simoncelli, and J. Dorsey. Efficient re-rendering of naturally illuminated environments. *Eurographics Workshop*, page 106, 1994.

[77] P. Noriega and O. Bernier. Real time illumination invariant background subtraction using local kernel histograms. *British Machine Vision Conference*, 3 :979–988, 2006.

[78] S. Pessoa, G. Moura, J. Lima, V. Teichried, and J. Kelner. Photorealistic rendering for augmented reality : A global illumination and BRDF solution. In *Virtual Reality Conference*, pages 3–10, Mar. 2010.

[79] S. A. Pessoa, E. L. Apolinario, G. de S. Moura, and J. P. S. do M. Lima. Illumination techniques for photorealistic rendering in augmented reality. *ISMAR*, pages 223–232, 2008.

[80] J. Pilet, C. Strecha, and P. Fua. Making background subtraction robust to sudden illumination changes. 2008.

[81] M. Planck. The theory of heat radiation. 1914.

[82] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF : a stereo correspondance algorithm using a disparity gradient limit. *Perception*, 1985.

[83] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows : Algorithms and evaluation. *Transaction on Pattern Analysis and Machine Intelligence*, 2003.

[84] G. Reitmay and T. W. Drummond. Going out : Robust model-based tracking for outdoor augmented reality. *International Symposium on Mixed and Augmented Reality*, pages 109–118, 2006.

[85] M. Robart. *Simulation des interactions lumière - matière pour la modélisation de la réflectance par ondelettes en synthèse d'images réalistes*. PhD thesis, Université de Toulouse 3, 1999.

[86] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut" - interactive foreground extraction using iterated graph cuts. 2003.

[87] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. 1996.

[88] P. Schröder and W. Sweldens. Spherical wavelets : Efficiently representing functions on the sphere. *SIGGRAPH*, pages 161–172, 1995.

[89] D. Sharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. 2001.

[90] J. Shin, R. Grasset, H. Seichter, and M. Billinghurst. A mixed-reality rendering framework for photorealistic and non-photorealistic rendering. Technical report, HITLabNZ, 2008.

[91] B. Shoushtarian and H. E. Bez. A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking. *Pattern Recognition Letters*, 1 :5–26, 2004.

[92] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighthing environments. 2002.

[93] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2 :246–252, 1998.

[94] L. D. Stefano, M. Marchionni, S. Mattoccia, and G. Neri. A fast area-based stereo matching algorithm. 2002.

[95] J. K. Suhr, H. G. Jung, G. Li, and J. Kim. Mixture of gaussians-based background subtraction for bayer-pattern image sequences. *Transactions on Circuits and Systems for Video Technology*, 2011.

[96] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. *European Conference on Computer Vision*, pages 628–641, 2006.

[97] T. Thalwitzer. *Max-Flow Min-Cut*. VDM, 2009.

[98] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 1986.

[99] D. van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 2010.

[100] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography : Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *SIGGRAPH*, 2007.

[101] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. 2008.

[102] J. A. Vijverberg, M. J. Loomans, C. J. Koeleman, and P. H. de With. Global illumination compensation for background subtraction using gaussian-based background difference modeling. IEEE, 2009.

[103] V. Vineet and P. Narayanan. CudaCuts : fast graph cuts on the GPU. 2008.

[104] L. Wang, C. Zhang, R. Yang, and C. Zhang. TofCut : towards robust real-time foreground extraction using a time-of-flight camera. 2009.

[105] T.-T. Wong, P.-A. Heng, S.-H. Or, and W.-Y. Ng. Image-based rendering with controllable illumination. *Eurographics Workshop*, 97 :13–22, 1997.

[106] J. Xiao and M. Shah. Accurate motion layer segmentation and matting. IEEE, 2005.

[107] I. M. Zendjebil, F. Ababsa, J.-Y. Didier, J. Vairion, L. Frauciel, M. Hachet, P. Guitton, and R. Delmont. Réalité augmentée en extérieur : enjeux et état de l'art. *AFRV*, 2007.

[108] B. Zhong, S. Liu, H. Yao, and B. Zhang. Multi-resolution background subtraction for dynamic scenes. 2009.

[109] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[110] M. Zoellner. Virtual museum guide, 2010.

[111] M. Zoellner, D. Stricker, G. Bleser, and Y. Pastarmov. iTacitus - novel interaction and tracking paradigms for mobile AR. *VAST*, 2007.

Amélioration de la cohérence visuelle pour la réalité mixte appliquée au patrimoine

Résumé : Le travail présenté dans ce mémoire a pour cadre le dispositif de réalité mixte ray-on, conçu par la société on-situ. Ce dispositif, dédié à la mise en valeur du patrimoine architectural et en particulier d'édifices historiques, est installé sur le lieu de l'édifice et propose à l'utilisateur une vision uchronique de celui-ci. Le parti pris étant celui du photo-réalisme, deux pistes ont été suivies : l'amélioration du mélange réel virtuel par la reproduction de l'éclairage réel sur les objets virtuels, et la mise en place d'une méthode de segmentation d'image résiliente aux changements lumineux.
Pour la reproduction de l'éclairage, une méthode de rendu basé-image est utilisée et associée à une capture haute dynamique de l'environnement lumineux. Une attention particulière est portée pour que ces deux phases soient justes photométriquement et colorimétriquement. Pour évaluer la qualité de la chaîne de reproduction de l'éclairage, une scène test constituée d'une mire de couleur calibrée est mise en place, et capturée sous de multiples éclairages par un couple de caméra, l'une capturant une image de la mire, l'autre une image de l'environnement lumineux. L'image réelle est alors comparée au rendu virtuel de la même scène, éclairée par cette seconde image.
La segmentation résiliente aux changements lumineux a été développée à partir d'une classe d'algorithmes de segmentation globale de l'image, considérant celle-ci comme un graphe où trouver la coupe minimale séparant l'arrière plan et l'avant plan. L'intervention manuelle nécessaire à ces algorithmes a été remplacée par une pré-segmentation de moindre qualité à partir d'une carte de profondeur, cette pré-segmentation étant alors utilisée comme une graîne pour la segmentation finale.
Mots clés : réalité mixte, rendu photo-réaliste, rendu basé-image, calibrage colorimétrique, segmentation d'image

Improvements to the visual consistency of mixed reality applied to cultural heritage

Abstract : The work described in this report has as a target the mixed reality device ray-on, developed by the on-situ company. This device, dedicated to cultural heritage and specifically architectural heritage, is meant to be installed on-site and shows the user an uchronic view of its surroundings. As the chosen stance is to display photo-realistic images, two trails were followed : the improvement of the real-virtual merging by reproducing accurately the real lighting on the virtual objects, and the development of a real-time segmentation method which is resilient to lighting changes.
Regarding lighting reproduction, an image-based rendering method is used in addition to a high dynamic range capture of the lighting environment. The emphasis is put on the photometric and colorimetric correctness of these two steps. To measure the quality of the lighting reproduction chain, a test scene is set up with a calibrated color checker, captured by a camera while another camera is grabbing the lighting environment. The image of the real scene is then compared to the simulation of the same scene, enlightened by the light probe.
Segmentation resilient to lighting changes is developed from a set of global image segmentation methods, which consider an image as a graph where a cut of minimal energy has to be found between the foreground and the background. These methods being semi-automatic, the manual part is replaced by a low resolution pre-segmentation based on the depthmap of the scene which is used as a seed for the final segmentation.
Keywords : mixed reality, photo-realistic rendering, image-based rendering, colorimetric calibration, image segmentation

ARTS ET MÉTIERS ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY