

Chapitre 5

Modification des propriétés d'une scène réelle connue sous différents éclairages

Dans le chapitre précédent, nous avons présenté une méthode permettant le ré-éclairage interactif de scènes réelles. Cette méthode est basée uniquement sur l'utilisation de la radiosité hiérarchique, dont la structure de subdivision de polygone nous permet d'utiliser le matériel d'accélération graphique pour l'affichage. Même si les résultats sont satisfaisants, il arrive parfois que les textures ne soient pas correctement corrigées, et que certains artefacts soient perceptibles, notamment dans les régions d'ombre originales. De plus, les paramètres radiométriques ne sont pas toujours estimés correctement, à cause d'une part des manipulations de textures au niveau des éléments de radiosité plutôt que sur chaque pixel, et d'autre part de l'utilisation d'une seule photographie pour l'estimation.

Dans l'approche présentée dans ce chapitre, nous essayons de retrouver des paramètres radiométriques plus fiables, ce qui nous permet d'étendre les possibilités d'interaction à l'enlèvement virtuel d'objets réels par exemple. Une idée simple pour résoudre ce problème de fiabilité est de prendre plusieurs photographies (au lieu d'une seule pour la solution précédente), afin d'avoir chaque zone de la scène sous éclairage direct sur au moins une photographie. Dans ce chapitre, nous décrivons une méthode permettant de retrouver les propriétés radiométriques d'une scène réelle en utilisant plusieurs photographies prises sous le même point de vue, mais sous des éclairages différents. Comme pour la méthode précédente, notre objectif est d'obtenir une méthode simple et rapide, utilisable par un utilisateur ayant un matériel non professionnel. Ce que nous attendons de cette méthode sont des résultats comparables à la réalité, sans exiger des valeurs exactes. De plus, il nous paraît indispensable de conserver l'aspect interactif lors de la phase de manipulation virtuelle. Notre objectif est toujours de permettre à un utilisateur de modifier *interactivement* et de façon virtuelle la géométrie et l'éclairage de scènes réelles.

Contrairement à la méthode précédente, nous travaillons pixel par pixel, pour estimer les réflectances et effectuer le rendu. Nous utilisons la méthode de radiosité uniquement pour calculer l'éclairage indirect. Le maillage utilisé est un maillage bien plus grossier que celui qui était nécessaire pour la méthode précédente, ce qui permet d'avoir un temps

de calcul très rapide pour la radiosité. Le travail pixel par pixel présente plusieurs avantages. Pour l'affichage, ceci nous permet d'éviter un affichage multi-passes utilisé pour la première méthode. De plus, cette nouvelle méthode nous permet d'enlever des objets réels en comblant l'information manquante par des textures générées et appliquées pixel par pixel, et se prête mieux aux améliorations (voir le chapitre 7).

5.1 Description générale de la méthode

Cette nouvelle approche comporte quatre étapes :

1. reconstruction géométrique et capture des différentes conditions d'éclairage pour un point de vue unique,
2. estimation de la réflectance pour chaque condition d'éclairage,
3. combinaison, par moyenne pondérée, des estimations précédentes, pour obtenir une estimation plus fiable de la réflectance,
4. initialisation du système d'éclairage pour permettre les modifications interactives.

Nous utilisons tout d'abord deux ensembles distincts de photographies. Le premier ensemble sert à retrouver un modèle géométrique simplifié de la scène réelle. Le deuxième ensemble permet d'estimer les propriétés de réflectance des objets réels contenus dans cette scène. Les photographies de cet ensemble sont du même point de vue, mais les conditions d'éclairage sont changées pour chacune d'elles, en déplaçant une source de lumière unique et connue. L'étape d'acquisition est développée dans la section 5.2. Nous appelons les photographies du deuxième ensemble des *images de luminance*. Cette appellation n'est pas strictement justifiée car il s'agit d'images « brutes » RGB. Nous levons cette restriction dans le chapitre suivant 6. Nous disposons pour chaque image de la géométrie de la scène réelle, ainsi que du modèle de la source de lumière. Par conséquent, nous avons accès à certaines informations telles qu'une valeur approximée de radiosité pour chaque pixel, les informations concernant la source de lumière, et tous les facteurs dépendants de la géométrie. À partir de ces informations, nous calculons une *image de réflectance* pour chaque position de la source de lumière. Nous n'avons cependant pas d'information sur l'éclairage indirect. Pour cette raison, la réflectance estimée par image est considérée comme approximative et inexacte dans les zones d'ombre, où l'influence de l'éclairage indirect est très importante.

Nous effectuons ensuite une moyenne pondérée des images individuelles de réflectance. La pondération se fait à partir de la confiance qu'on a pixel par pixel en ces estimations individuelles. Des filtres nous permettent d'évaluer cette confiance avant le calcul de la réflectance finale. Toute cette phase d'estimation de la réflectance est décrite dans la section 5.3. Nous supposons que la scène est diffuse. En particulier, nous supposons que les réflectances des objets sont diffuses, et que les lampes émettent de façon diffuse.

Le modèle géométrique et les propriétés de réflectance sont ensuite utilisés pour initialiser un système d'éclairage qui simule l'éclairage original de la scène (voir la section 5.4). Le rendu virtuel utilise les paramètres estimés, et re-projette les ombres des objets réels dues à la lampe réelle. L'image ainsi produite est comparable à la photographie originale.

Pour permettre une bonne définition des ombres, l'éclairage direct est calculé pixel par pixel en utilisant un lancer de rayons depuis le point de vue fixe. En revanche, l'éclairage indirect est simulé par une solution de radiosit  hi rarchique, am lior e par une structure d'acc l ration pour des modifications dynamiques. L' clairage indirect est indispensable pour avoir une simulation r aliste de l' clairage. Cependant l'exactitude pixel par pixel de cet  clairage indirect ne nous a pas sembl  n cessaire, et nous avons pr f r  une approximation par petites r gions ; en revanche, la d finition des ombres directes depuis les sources de lumi re est visuellement tr s importante. Le choix de deux repr sentations diff rentes pour chaque type d' clairage nous permet d'avoir   la fois une connaissance tr s fine de l' clairage direct, tout en ayant une solution de radiosit  grossi re simulant l' clairage indirect.

Lorsque ces quatre premi res  tapes de pr -calcul sont termin es, tout est en place pour modifier la g om trie et l' clairage de la sc ne r elle mod lis e. Les algorithmes de modification interactive adapt s   cette m thode sont pr sent s dans la section 5.5.

5.2 Acquisition

Afin de reproduire et de modifier l' clairage d'une sc ne r elle, nous avons besoin d'un mod le g om trique en trois dimensions de cette sc ne ainsi qu'une connaissance de ses propri t s radiom triques. Pour cela, nous avons besoin d'acqu rir les donn es de la sc ne r elle. Cette acquisition se fait en deux  tapes.

Une premi re consiste   prendre quelques photographies qui servent   reconstruire un mod le g om trique en trois dimensions de la sc ne r elle sur laquelle nous voulons travailler. Nous utilisons pour la reconstruction l'outil *Rekon* [POF98]. Cet outil est d crit en d tail dans le chapitre 3 section 3.2.2. Les photographies doivent  tre prises sous des angles de vue diff rents pour permettre un bon calibrage. L'ensemble de ces photos, typiquement entre sept et neuf, ayant servi   la reconstruction d'une de nos sc nes est pr sent  sur la figure 5.1. Le mod le extrait est montr  sur la figure 5.2. Les coordonn es des murs sont mesur es et leur mod le reconstruit   la main. Il est important de noter que le mod le extrait est un mod le simplifi  ; la sc ne r elle n'est pas reconstruite dans tous ses d tails. Ceci serait beaucoup trop co teux en temps de mod lisation et ne s'av re pas n cessaire dans notre but de r - clairage interactif.

La deuxi me  tape d'acquisition doit nous fournir un ensemble de donn es   partir desquelles nous pourrions estimer la r flectance diffuse des objets r els. Pour r aliser cette estimation, nous d sirons avoir une bonne connaissance de l' clairage de la sc ne. Pour cela, nous prenons un ensemble de photographies, prises depuis un seul et m me point de vue, mais sous des  clairages diff rents. Pour faciliter l'estimation, nous n'avons qu'une seule source de lumi re manipulable. Notre objectif est de permettre   un designer de r aliser cette  tape d'acquisition dans un temps raisonnable, et en une seule fois. Le r sultat ne doit pas d pendre de la qualit  exhaustive de l'acquisition. De plus, le mat riel utilis  doit  tre facilement manipulable et peu co teux. Nous avons donc utilis  une lampe de jardin, tr s facile   d placer et   mod liser. Cette lampe est illustr e sur la figure 5.3. Afin d'avoir un  clairage le plus diffus possible, nous avons couvert la lampe par une feuille de papier



FIG. 5.1: Photographies servant à la reconstruction 3D de la scène réelle.

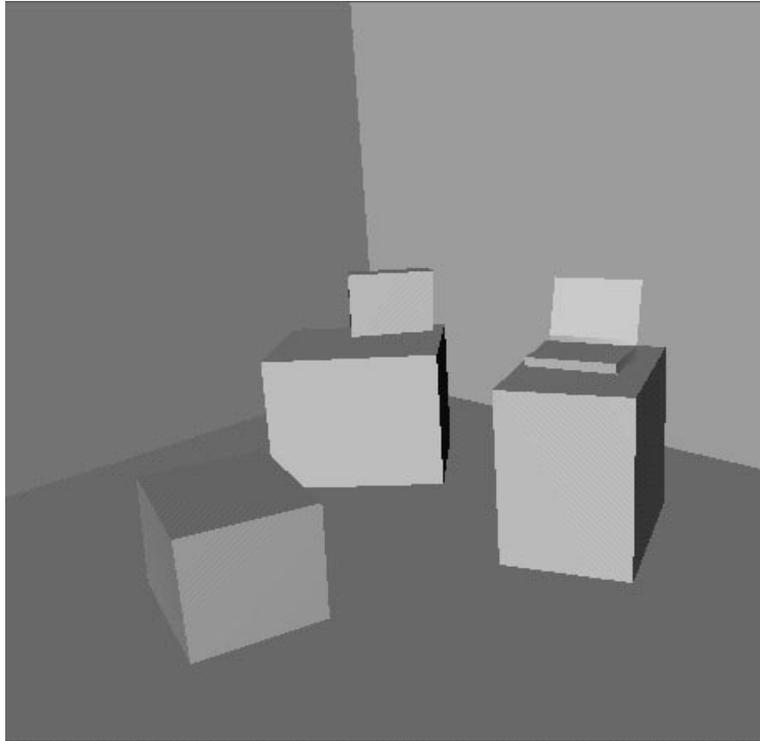


FIG. 5.2: Modèle 3D reconstruit.



FIG. 5.3: Lampe utilisée pour éclairer les scènes lors de nos tests. Elle est recouverte d'un papier de soie, permettant de se rapprocher du caractère d'émission diffuse.



FIG. 5.4: Images de luminance servant à l'estimation de la réflectance.

de soie. La lampe est placée dans une position différente pour chacune des photographies. Cette position ainsi que l'inclinaison de la lampe, sont mesurées à la main. Elle fournit le seul éclairage de la pièce. La figure 5.4 montre 7 photographies, appelées « images de luminance ». On voit clairement que les ombres sont différentes sur chacune de ces images. Pour chaque pixel, nous essayons d'avoir au moins une image où l'objet lui correspondant est éclairé directement par la lampe.

5.3 Estimation de la réflectance

Une fois la géométrie reconstruite, nous sommes prêts à procéder à l'estimation radiométrique de la scène réelle. Ceci commence par l'estimation de la réflectance dont nous présentons la méthode dans cette section.

5.3.1 Approche générale

L'estimation de la réflectance se fait d'abord pour chacune des images de luminance. Puis, pour chaque pixel, une valeur de confiance est calculée. Cette valeur reflète notre confiance en la précision de l'estimation de la réflectance pour ce pixel. Ensuite, une valeur finale de réflectance est calculée par une moyenne des valeurs estimées sur chacune des images de luminance, pondérées par la valeur de confiance associée.

Si nous supposons que toutes les surfaces de la scène sont diffuses, il existe une relation simple entre la luminance L en un point 3D P , la réflectance ρ , et la lumière incidente en ce point :

$$L = \rho \left(\hat{B} + \sum_i F_i V_i E_i \right) \quad (5.1)$$

où $F_i V_i E_i$ est l'éclairage provenant directement de la source de lumière i , et \hat{B} est l'éclairage indirect provenant du reste de la pièce par inter-réflexions. E_i est l'existance de la source i , c'est-à-dire l'énergie émise, F_i est le facteur de forme entre la source i et le point P , et V_i est la valeur de visibilité entre la source i et le point P . Le facteur de forme définit la portion d'énergie arrivant sur une surface, et la visibilité décrit la portion d'énergie non arrêtée par des obstacles (voir chapitre 2).

Structure de données

Pour estimer la réflectance et effectuer le rendu, nous travaillons dans l'espace de l'image. Ce choix permet d'avoir une estimation de la réflectance plus précise, en gardant les détails sans avoir à manipuler des polygones texturés. De plus, la gestion dans l'espace image permet d'utiliser des structures de données efficaces. A chacun des pixels, nous associons une structure augmentée stockant l'information nécessaire, ce qui évite des calculs inutiles. De plus, cette base de pixels nous permet de réaliser des localisations dans l'espace image (2D) plutôt que dans l'espace de la scène 3D.

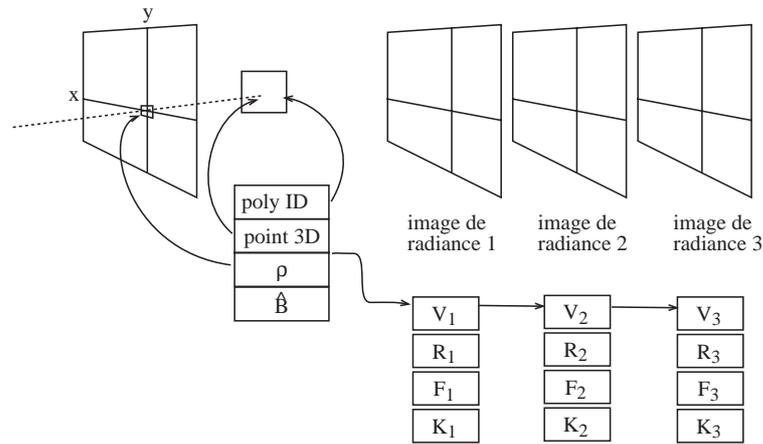


FIG. 5.5: Structure de données pour l'estimation de la réflectance.

Pour l'estimation de la réflectance, nous utilisons une structure de données pixel par pixel, pour le point de vue donné par les images de luminance. Nous créons une structure de données décrite par la figure 5.5. Dans cette structure, nous stockons pour chaque pixel :

- le point 3D P re-projeté au centre du pixel,
- un numéro d'identification (ID) unique pour chaque polygone,
- la réflectance finale ρ en ce point,
- une liste d'information pour chacune des images de luminance.

Pour chaque image de luminance, nous stockons pour chacun des pixels :

- le facteur de forme F_i entre la source de lumière i et le point 3D P ,
- le facteur de visibilité V_i depuis la source de lumière i et le point P , déterminant la quantité de lumière parvenant sans obstacle en ce point,
- l'éclairage indirect \hat{B} arrivant en ce point,
- la réflectance estimée ρ_i en ce point,
- le coefficient K_i représentant la justesse d'estimation de la réflectance.

Le numéro d'identification est déterminé en utilisant un « item buffer » [WHG84]. Un exemple est illustré par la figure 5.6, où chacune des couleurs différentes sont associées à un numéro d'identification. Le point 3D P est calculé en intersectant une droite passant par le point de vue, et le centre du pixel correspondant, avec le polygone identifié par l'« item buffer »*.

Le facteur de forme F_i est calculé en utilisant des techniques standard de calcul entre point et polygone [BRW89] (voir la section 2.2.1). La visibilité V_i est la fraction du polygone de la source de lumière visible depuis le point P . La valeur de V_i est comprise

*Cependant des erreurs de précision numérique peuvent survenir, entraînant notamment une non-intersection entre le rayon et le polygone. Ce phénomène apparaît surtout sur les contours des polygones. Dans ce cas-là, nous déplaçons le rayon vers les pixels voisins pour trouver une intersection.

entre '0' lorsque le polygone est entièrement invisible depuis le point P , et '1' lorsqu'il est complètement visible. L'estimation se fait par lancer de rayons, entre la source et le point. La valeur de visibilité est égale au pourcentage de rayons n'ayant rencontré aucun obstacle. Le nombre de rayons que l'on utilise varie entre 4 et 64, selon la rapidité et la qualité des calculs désirées.

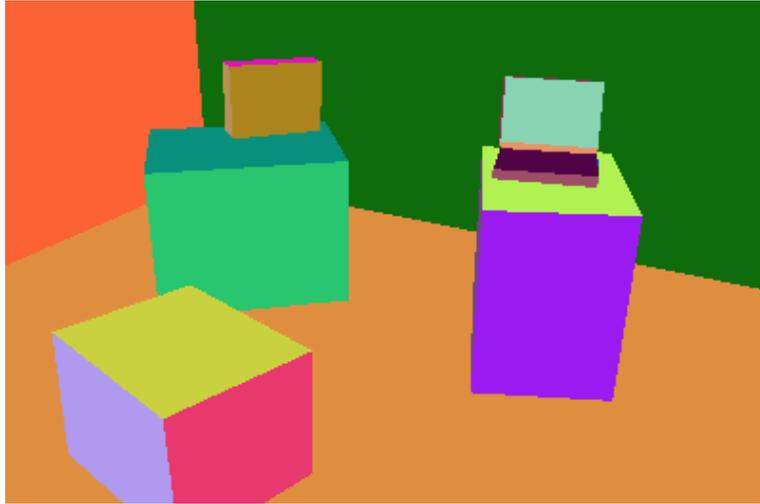


FIG. 5.6: Numérotation des polygones en utilisant un « item buffer ».

La connaissance géométrique de la scène nous permet de calculer certains des coefficients nécessaires à l'estimation de la réflectance. Nous n'avons cependant pas de connaissance radiométrique pour la luminance L , l'existance de la source E_i et l'irradiance indirecte \hat{B} . Nous utilisons alors une approximation pour chacune de ces valeurs.

La scène n'est éclairée que par une seule lampe qui est la même pour toutes les images de luminance. Si son existance est inexacte, la répercussion de cette erreur se fait dans la solution de radiosité. Cependant, comme il n'y a qu'une seule source, l'erreur sur la solution de radiosité n'est qu'à un facteur multiplicatif près. L'existance E_i de la source est donc fixée par l'utilisateur[†]. Nous ne connaissons pas non plus la valeur d'irradiance indirecte \hat{B} . Cette valeur est fixée à la moyenne des luminances pour les pixels de l'image de luminance, divisée par la réflectance moyenne (ceci est basé sur l'équation (2.7)). Cette réflectance moyenne est également déterminée par l'utilisateur (comme pour le chapitre 4). La luminance L est calculée directement pour chacun des pixels de l'image par :

$$L = T^{-1}(C_i) \quad (5.2)$$

où T est la fonction de transformation de la caméra, et C_i la couleur du pixel correspondant au point P dans l'image de luminance associée à la lampe i . Nous avons dans un premier temps travaillé sur des images RGB avec une fonction de transformation identité, n'ayant aucun effet sur la couleur. Nous discutons ce choix dans le chapitre suivant (chapitre 6).

[†]Nous montrons dans le chapitre suivant comment retrouver une valeur d'existance à partir d'images « High Dynamic Range ».

Nous estimons grâce à tous ces paramètres une valeur de réflectance par pixel et par image de luminance, en utilisant l'équation (5.1). Cette estimation n'est pas forcément valable pour tous les pixels à cause de nos approximations sur les valeurs radiométriques, ainsi que les approximations sur la reconstruction géométrique (la scène n'est pas entièrement modélisée). Pour surmonter ces limitations, nous allons créer une image de réflectance en faisant une moyenne pondérée des images de réflectance associées à *chaque* condition d'éclairage. Une valeur de confiance est alors associée à chacun des pixels, et elle servira de valeur de pondération. Cette valeur de confiance est représentée par le coefficient de validité K_i . Une opération de filtrage permet de déterminer et d'étendre les valeurs de confiance. Nous réalisons alors un filtre associé à chacune des images de luminance, qui sera composé de coefficients de validité K_i , reflétant la confiance que nous avons dans l'estimation de la réflectance pour chacun des pixels. Une valeur finale de réflectance est alors calculée à partir des valeurs estimées pour chacune des images de luminance, pondérées par les coefficients de validité. Nous détaillons toutes ces étapes dans les paragraphes suivants.

5.3.2 Estimation de la réflectance pour chaque image de luminance

Pour chacune des images de luminance capturées, la scène n'est éclairée que par une seule lampe. Les valeurs F_i et V_i peuvent être calculées (d'une façon approximative) car nous avons une représentation géométrique de la scène. Comme nous l'avons expliqué précédemment, nous ne connaissons pas exactement l'exitance E_i de la source, ni la réflectance ρ et l'éclairage indirect \hat{B} au point P . Les approximations déjà décrites sont utilisées.

Pour chacune des images de luminance associée à une position différente i de la lampe, nous estimons, pour chaque pixel de l'image correspondant à un point 3D de la scène, une réflectance ρ_i :

$$\rho_i = \frac{T^{-1}(C_i)}{F_i V_i E_i + \hat{B}}. \quad (5.3)$$

Le réflectance obtenue n'est représentative de la réalité qu'à un facteur près étant donné que nous ne connaissons pas l'exitance de la source lumineuse.

Les résultats de cette estimation sont montrés par la figure 5.7. Dans la colonne de gauche, sont représentées les images de luminance, et dans la colonne du milieu, les images de réflectance associées. Il est à noter que la qualité des résultats n'est à cette étape pas satisfaisante. L'estimation est notamment assez mauvaise dans les régions d'ombre. Ceci est dû à l'approximation faite sur l'éclairage indirect (voir le chapitre 6 pour des améliorations sur le traitement de l'éclairage indirect).

5.3.3 Filtrage des valeurs de réflectance

L'éclairage indirect n'est pas le même pour tous les pixels ; l'utilisation du terme ambiant pour représenter l'éclairage indirect est donc une approximation insuffisante. De plus, la couleur du pixel n'est pas une valeur de luminance. D'une part, les valeurs de luminance réelles ont été transformées par des fonctions de la caméra. D'autre part, le bruit



FIG. 5.7: À gauche, les images de luminance (photographies). Au milieu, les images de réflectance associées à chacune des images de luminance. À droite, les images de coefficients de validité associées (après filtrage). Les pixels en blanc équivalent à une grande confiance.

introduit par la compression et la transformation de la lumière en valeurs RGB font que ces valeurs de luminance sont dans certains cas incorrectes. Cette méconnaissance à la fois de l'éclairage indirect et d'une valeur correcte des luminances réelles, entraîne une mauvaise connaissance de la réflectance, en particulier dans les zones d'ombre où il n'y a pas d'éclairage direct. Finalement, l'hypothèse d'avoir une scène entièrement diffuse ne permet pas de tenir compte des effets spéculaires de certaines surfaces, comme le reflet d'objets sur le sol de notre scène ou les taches de lumière sur l'écran de l'ordinateur portable sur les images 5.7.

Pour éviter que tous ces artefacts ne se retrouvent dans la réflectance finale, nous utilisons une moyenne pondérée de ces images de réflectances. Les valeurs de pondération sont déterminées par un processus de filtrage, pour éviter les problèmes dus aux approximations. Pour cela, nous associons à chaque pixel (x, y) de chaque image de luminance i un facteur de validité $K_i(x, y)$ représentatif de la confiance que l'on porte à l'estimation de la réflectance pour ce pixel. Les valeurs $K_i(x, y)$ sont comprises entre un (lorsque nous avons très confiance dans la validité de l'estimation) et zéro (lorsque nous n'avons aucune confiance dans la validité de l'estimation). Le processus complet de filtrage est décrit par la figure 5.8.

<p>Algorithme de filtrage Initialisation : $K_i(x, y) = V_i(x, y)$ Filtre par valeur minimum : $K_i(x, y) = \min (V_i(x_x, y_y))_{5 \times 5}$ Lissage : si $\text{moy}(K_i(x_x, y_y)) < K_i(x, y)$ alors $K_i(x, y) = \text{moy}(K_i(x_x, y_y))$ Filtrage des valeurs atypiques : si $\rho_i - \text{moy}(\rho_n) > \varepsilon$ alors $K_i = 0$ Lissage : si $\text{moy}(K_i(x_x, y_y)) < K_i(x, y)$ alors $K_i(x, y) = \text{moy}(K_i(x_x, y_y))$</p>

FIG. 5.8: Algorithme de filtrage.

Pour représenter le fait que l'estimation soit souvent erronée dans les régions d'ombre, les valeurs K_i sont initialisées à partir des valeurs de visibilité V_i . Un exemple de valeurs de visibilité est illustré par la figure 5.9 (a). La confiance est ainsi très élevée dans les zones d'éclairage direct sans obstruction, et basse dans les zones d'ombre. Le long des zones d'ombre, la précision de la visibilité dépend beaucoup de l'exactitude de la géométrie et donc de la qualité de la reconstruction géométrique. Une première étape de filtrage consiste à prendre en compte cette inexactitude, de façon indépendante pour chacune des images de luminance. Les valeurs basses des facteurs de validité sont étendues aux pixels voisins en utilisant un filtre par valeur de minimum. Le pixel sur lequel est centré le filtre prendra la valeur minimale des valeurs à l'intérieur du support du filtre ((x_x, y_y) dans la figure 5.8). Dans nos tests, nous utilisons par exemple un filtre de 5×5 ce qui reflète notre estimation de

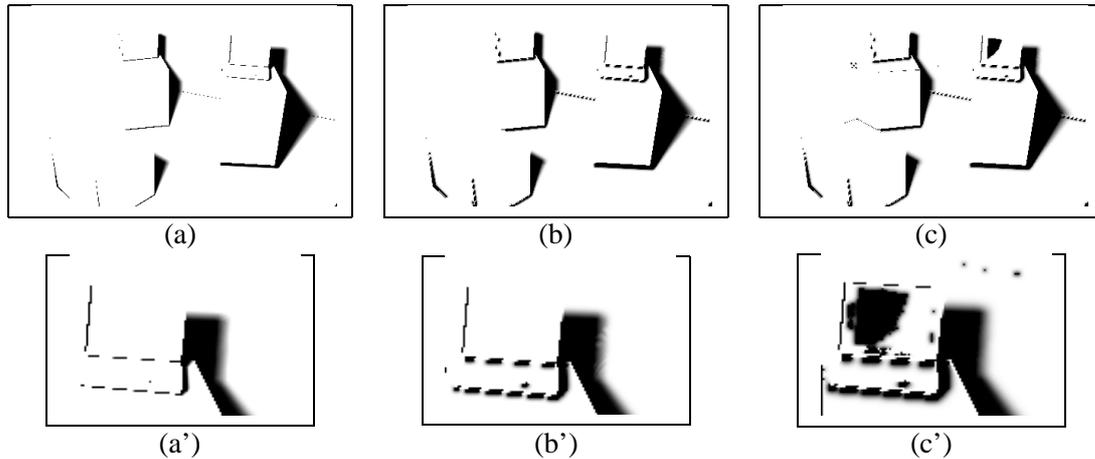


FIG. 5.9: Résultats à chaque étape du filtrage. (a) Initialisation des valeurs de validité par les valeurs de visibilité. (b) Élargissement des zones des valeurs basses. (c) Détection des valeurs singulières. (a'), (b') et (c') sont un zoom sur les images (a), (b) et (c).

la précision de la reconstruction à deux pixels près. La taille du filtre peut donc être ajustée si l'on considère que l'erreur est plus ou moins grande. Pour éviter des changements trop abrupts dans les valeurs de validité, un filtre est appliqué pour lisser les transitions. Ce filtre est également de taille 5x5. Le principe est de remplacer la valeur au centre du filtre par la moyenne des valeurs à l'intérieur du filtre si cette valeur est inférieure à la valeur initiale. Il n'est appliqué cependant que si tous les pixels observés dans le filtre appartiennent au même polygone. Les valeurs de validité ainsi étendues et lissées sont illustrées sur la figure 5.9 (b).

Les valeurs de réflectance très éloignées des autres valeurs estimées pour chacune des images de luminance sont détectées dans une deuxième étape de filtrage. Ceci permet d'enlever les taches de lumière dues à la spécularité d'un objet, comme la spécularité de l'écran du portable, ou d'enlever par exemple les pieds de la lampe utilisée, visibles sur deux des images de la colonne de gauche de la figure 5.7. Nous utilisons pour cela un nouveau filtre qui détecte les valeurs singulières (« outliers »). Pour chacun des pixels, la valeur médiane des réflectances est calculée sur les réflectances ayant une valeur de validité supérieure à un seuil, par exemple de 0.75. Ceci permet d'éliminer du calcul de la médiane les valeurs qui n'ont pas un degré de confiance suffisant pour être prises en compte dans le calcul final. Aux valeurs de réflectance dont l'écartement à la valeur médiane est supérieur à un certain seuil, on associe un facteur de validité de zéro. Ce seuil est déterminé par l'utilisateur. Il ne doit pas être trop bas, car les valeurs de réflectance initiales sont calculées comme nous l'avons vu de façon approximative et ne sont pas égales les unes aux autres. Dans nos tests, nous avons utilisé un seuil de 0.3. À nouveau, pour éviter des discontinuités abruptes, un filtre (de 3x3) lissant les contours est appliqué.

À la fin de cette étape, nous avons les valeurs finales de validité qui serviront à pondérer les valeurs de réflectances pour calculer la réflectance moyenne. Cette opération est expliquée dans la section suivante 5.3.4. Ces valeurs sont illustrées sur la figure 5.9 (c).

Les images composées des coefficients de validité sont présentées dans la colonne de droite de la figure 5.7 pour chacune des images de luminance. Les valeurs sont basses dans les régions d'ombre, et basses pour les valeurs singulières comme pour la tache de specularité visible sur la première ligne d'images.

5.3.4 Estimation finale de la réflectance

Nous avons estimé une réflectance diffuse pour chacun des pixels des images de luminance. À chacune de ces valeurs, nous avons associé une valeur de confiance K_i . Nous avons vu que l'estimation est approximative sur chacune des images prises indépendamment les unes des autres. Par contre, en combinant ces valeurs, nous pouvons avoir une estimation convaincante, diminuant ainsi les artefacts. Pour cela, nous calculons une moyenne sur chacun des pixels, en pondérant les valeurs de réflectance par leur coefficient de validité K_i . Pour chaque pixel, la réflectance associée ρ est :

$$\rho = \frac{\sum_i K_i \rho_i}{\sum_i K_i} \quad (5.4)$$

Les résultats sont montrés sur la figure 5.10. Les images (a) et (b) sont des images de réflectances estimées pour deux des images de luminance, auxquelles sont associées des valeurs de validité, montrées respectivement en (c) et (d). En (e) est montrée la réflectance finale estimée par une moyenne pondérée des sept images de réflectance (voir figure 5.7). Alors que les réflectances sont estimées d'une façon moins précise dans les images (a) et (b) (comme pour l'effet spéculaire sur le portable, ou dans les régions d'ombre), l'image de réflectance (e) finale contient très peu d'artefacts.

5.4 Initialisation du système d'éclairage

Lorsque les réflectances sont calculées, le système d'éclairage peut être initialisé. L'éclairage est calculé en combinant deux méthodes. L'éclairage direct est calculé pixel par pixel et l'éclairage indirect par une solution de radiosité optimisée. Ce choix permet d'avoir une bonne définition de l'éclairage direct, dont les variations sont souvent abruptes, dans les régions d'ombre par exemple. En revanche, les variations de l'éclairage indirect sont souvent moins significatives et une approximation par une solution de radiosité est souvent suffisante. Deux structures sont ainsi initialisées pour calculer l'éclairage. Ceci nous permet d'avoir une bonne définition des contours des ombres, tout en ayant un rendu et une mise à jour interactifs.

5.4.1 Éclairage direct

L'éclairage direct est calculé pixel par pixel. Ce calcul est stocké au fur et à mesure dans une structure qui servira à réaliser un rendu accéléré. Pour cela, nous utilisons une deuxième structure de données où nous stockons à nouveau de l'information pixel par pixel :

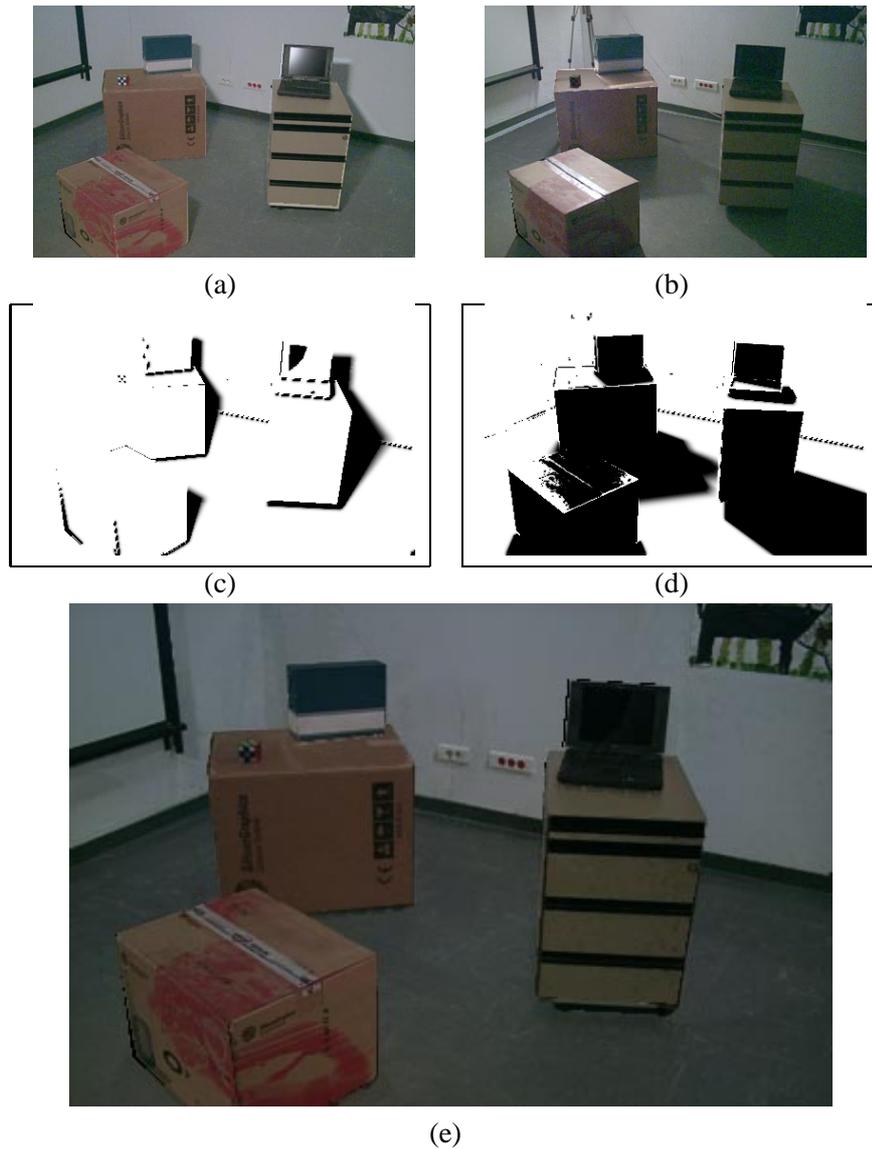


FIG. 5.10: (a) et (b) Réflectances estimées pour deux images de luminance. (c) et (d) Valeurs de validité associées à ces deux mêmes images de luminance. (e) Réflectance finale.

- le point 3D P re-projeté au centre du pixel,
- le numéro d'identification (ID) du polygone correspondant,
- l'éclairage indirect \hat{B} arrivant en ce point,
- la réflectance ρ_i en ce point,
- une liste d'informations pour chacune des sources de lumière s de la scène :
 - le facteur de forme F_s entre la source de lumière s et le point 3D P ,
 - le facteur de visibilité V_s depuis la source de lumière s et le point P ,
 - l'exitance E_s de la source s .

La structure de données complète est illustrée figure 5.11. Cette deuxième structure de données n'était pas indispensable. Nous aurions pu travailler avec une structure unique, groupant les informations de cette structure et de la première structure illustrée par la figure 5.5.

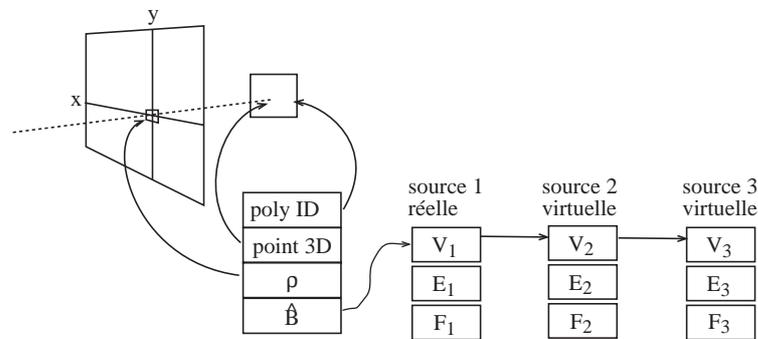


FIG. 5.11: Structure de donnée utilisée pour le rendu.

Cette structure est initialisée pixel par pixel. Les sources de lumière présentes dans la scène ne sont plus celles utilisées pour l'estimation de la réflectance. Lors de l'initialisation du système, une seule source est conservée. Cette source correspond à la source 1 dans la structure de données sur la figure 5.11. L'utilisateur choisit une image de référence, et la source de lumière associée. Les sources dont l'indice est supérieur à un sont des sources de lumière virtuelles ajoutées dans la scène. Le nombre total de sources n_s variera selon les manipulations de l'utilisateur.

Pour chaque pixel, les champs de réflectance sont initialisés, et la liste des sources de lumière comprend un élément, avec l'exitance de la source, les facteurs de forme et la visibilité. Cette initialisation se fait directement à partir de la table ayant servi à estimer les réflectances (voir section 5.5).

5.4.2 Éclairage indirect

Pour permettre le calcul de l'éclairage indirect, nous utilisons un système de radiativité optimisé. Tout d'abord, pour chaque polygone visible sur l'image de référence (depuis

le point de vue choisi), la réflectance est initialisée. Pour cela, les polygones sont pré-subdivisés. Pour chaque carreau, la réflectance est rendue égale à la moyenne des réflectances sur les pixels correspondant à partir de l'image de réflectance finale. Pour les carreaux non visibles associés à un polygone visible, la réflectance est initialisée à la moyenne des réflectances de ce polygone. L'initialisation de la réflectance dans le système de radiosit  pour ces  l ments visibles depuis l' il est illustr e figure 5.12.

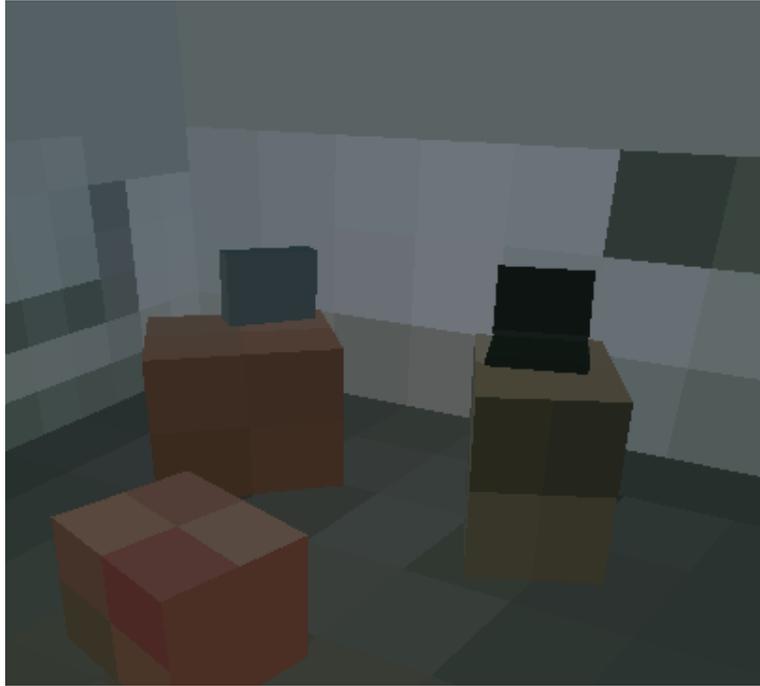


FIG. 5.12:  l ments de la radiosit  dont la r flectance est initialis e   partir de l'image de r flectance (voir figure 5.10 (e)).

Pour les polygones non visibles sur l'image de r f rence, un traitement sp cial est effectu . Ces polygones sont montr s sur la figure 5.13. Des photographies additionnelles, montr es sur la figure 5.14, sont prises lors de la capture. En utilisant l' quation (5.3), nous retrouvons pour un  chantillon de pixels des valeurs de r flectance qui sont ensuite moyenn es. Cette valeur moyenne sert   initialiser la r flectance du polygone. Par exemple, sur la figure 5.15, la texture en (a) sert   estimer une r flectance approximative pour le mur en (b). Les polygones pour lesquels nous n'avons pas d'image de luminance comme base de calcul, sont initialis s   une valeur de r flectance fix e par l'utilisateur.

Lorsque l'ensemble des r flectances de la sc ne est initialis , une solution de radiosit  peut  tre calcul e. La subdivision est pr fix e sur un  chantillonnage grossier, car nous n'avons pas besoin d'avoir une d finition pr cise des contours des ombres. La solution de radiosit  est calcul e par des algorithmes bas s sur la radiosit  hi rarchique avec « clustering » (voir section 2.2.1). Cette solution de radiosit  nous fournit un  clairage global. Pour en extraire l' clairage indirect, nous proc dons en deux  tapes. Dans la premi re  tape,

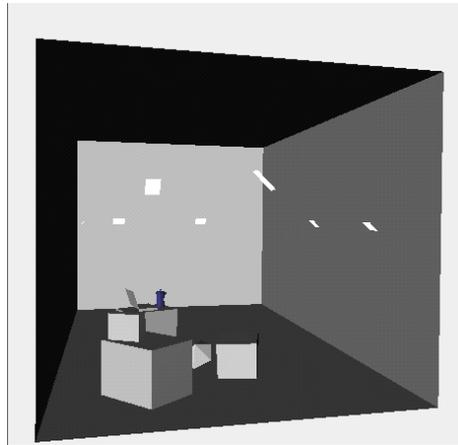
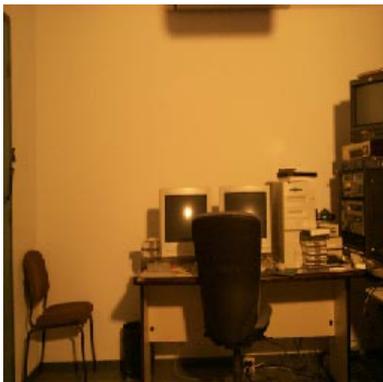
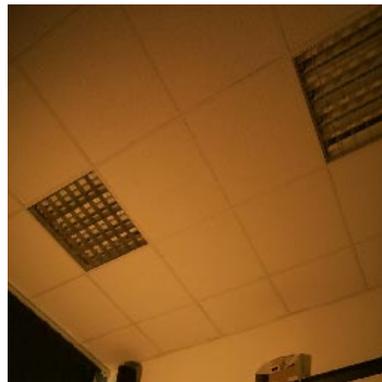


FIG. 5.13: Polygones invisibles depuis l'image originale. Dans cette scène, les polygones invisibles reconstruits sont les murs et le plafond.



(a)



(b)



(c)

FIG. 5.14: Photographies additionnelles permettant d'avoir une texture plus significative pour estimer la réflectance des murs, non visibles sur l'image d'origine. (a) Mur de droite, (b) Plafond, (c) Mur de derrière.



FIG. 5.15: (a) Texture servant à l'estimation plus approximative de la réflectance d'un mur non visible depuis le point de vue choisi (celui des images de luminance). (b) Réflectance du mur ainsi estimée.

nous calculons la radiosité globale ; dans la deuxième, nous ne calculons que l'éclairage direct (une seule itération dans la solution). L'éclairage indirect est ensuite estimé pour chaque élément (et pour chaque sommet d'élément), en soustrayant l'éclairage direct de l'éclairage global.

L'éclairage indirect ainsi calculé est ensuite utilisé pour mettre à jour la structure d'affichage, composée en partie par la structure représentant l'éclairage direct (figure 5.11). On utilise pour cela le matériel d'accélération graphique des stations de travail. Les valeurs de l'éclairage indirect sont affichées sur un tampon de mémoire auxiliaire, avec un lissage par affichage avec des valeurs aux sommets et interpolation. Cet affichage est récupéré pixel par pixel pour initialiser \hat{B} dans la structure de données.

En prévision d'éventuelles modifications, une structure de « shaft » est créée [HW94]. Suivant l'idée de la méthode de Drettakis et Sillion [DS97], un shaft est associé à chaque lien, et pour chaque shaft, la liste des polygones intersectés est maintenue. Ceci nous permettra par la suite d'optimiser les changements de visibilité et de localiser les modifications dans la structure des liens.

5.4.3 Rendu

Le rendu est réalisé pixel par pixel en utilisant la structure d'affichage 5.11. Lors de l'affichage, la couleur C est calculée pour chaque pixel par l'équation suivante :

$$C = \rho \left(\hat{B} + \sum_{s=1..n_s} F_s V_s E_s \right) \quad (5.5)$$

où F_s, V_s, E_s sont respectivement le facteur de forme, la visibilité, et l'existance de la source s , n_s est le nombre total de sources dans la scène, ρ la réflectance associée au pixel, et \hat{B} l'irradiance indirecte. Avant toute insertion de source virtuelle, il n'y a qu'une seule

source, représentant la source originale. Pour accélérer l'affichage, les composantes de l'équation (5.5) sont groupées et pré-calculées. Typiquement l'affichage d'une image 300 par 400 pixels prends moins d'un dixième de seconde.

L'image de la figure 5.16 (b) est le résultat d'un rendu après initialisation de tous les paramètres de la scène originale, et en utilisant l'équation (5.5). Cette image ne contient plus d'informations de la photographie originale, mais n'utilise que celle fournie par l'image de réflectance, dans laquelle il n'y a plus les effets d'éclairage. Les ombres sont à nouveau projetées grâce au terme de visibilité V_s . La qualité du rendu, tenant compte des détails, est due à la qualité de la réflectance ρ , différente pixel par pixel. L'image simulée n'est pas identique à la photographie originale montrée figure 5.16 (a). En effet, la réflectance estimée est supposée parfaitement diffuse, et en conséquence, les effets spéculaires ne sont pas affichés. De plus, la reconstruction des objets non visibles depuis le point de vue est très approximative ce qui implique des erreurs dans le calcul de l'éclairage indirect. Cette image simulée est cependant convaincante car elle est similaire et cohérente par rapport à l'originale, ce qui la rend acceptable pour l'utilisation voulue.



FIG. 5.16: (a) Photographie originale. (b) Reprojection après simulation de l'éclairage.

5.5 Ré-éclairage et modification de la géométrie

Lorsque les réflectances sont estimées et le système d'éclairage initialisé, nous pouvons modifier interactivement sur ordinateur les propriétés des scènes réelles reconstruites. Les modifications permises par le système sont au niveau de la géométrie et de l'éclairage. Nous pouvons ajouter, déplacer et enlever des objets virtuels. Nous pouvons également enlever des objets réels. De plus, nous pouvons modifier l'intensité des sources de lumière et ajouter des nouvelles sources virtuelles.

5.5.1 Modification de la géométrie

Nous désirons pouvoir interactivement ajouter, déplacer ou enlever des objets. Si nous refaisons entièrement le calcul de radiosité et de mise à jour de la table d'affichage, cette

modification géométrique prendrait de l'ordre de plusieurs minutes. Pour permettre un temps de mise à jour interactif, nous voulons localiser les modifications. Pour cela, nous utilisons une structure de shafts qui ne nous permet non seulement d'accélérer les calculs de visibilité par une localisation spatiale, mais aussi d'identifier les pixels directement affectés par des modifications de positions d'ombres (éclairage direct).

Insertion d'un objet virtuel

Lors de l'insertion d'un objet virtuel, la solution de radiosit  hi rarchique est mise   jour par l'algorithme suivant. Tout d'abord, l'objet est attach    la racine de la hi rarchie (« cluster » englobant toute la sc ne). Ensuite, des liens sont  tablis entre cet objet et chacune des lampes. Puis des it rations de raffinement sont faites pour  tablir les liens aux niveaux satisfaisant le crit re BF. Lors de cette passe de modification de la structure de radiosit , les shafts modifi s sont marqu s, de fa on   pouvoir plus tard localiser rapidement les changements. Un shaft peut  tre marqu  « modifi  » soit parce qu'il vient d' tre cr  , soit parce qu'il y a eu un changement de visibilit  (l'objet nouvellement ins r  intersecte ce shaft). Dans ce dernier cas, les polygones de l'objet intersectant le shaft sont ins r s dans la liste des polygones intersect s associ e au shaft. Le nouvel objet ins r  est test  pour chacun des shafts, et de nouveaux calculs de visibilit  dans la solution de radiosit  sont effectu s seulement s'il y a intersection. Un exemple de modification des shafts est illustr  sur la figure 5.17. En (a), le shaft est initialis . La liste des polygones intersect s est montr e par des traits clairs. Lors de l'insertion d'un objet virtuel en (b), le cluster englobant l'objet virtuel intersecte ce shaft et est ins r    la liste des polygones intersect s. Ce cluster est dessin  en traits clairs.

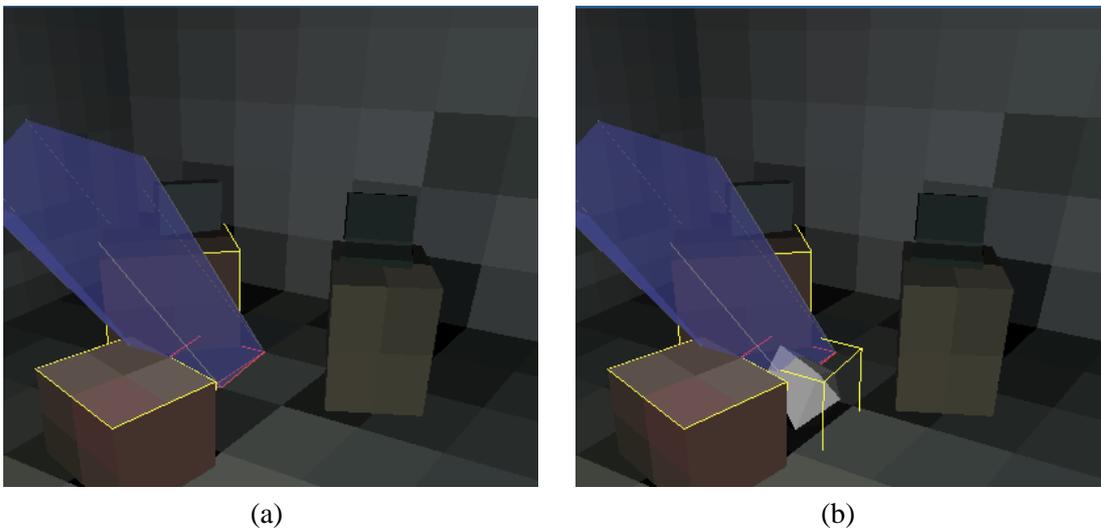


FIG. 5.17: (a) Shaft associ    un  l ment de la hi rarchie, avec en traits clairs la liste des polygones qu'il intersecte. (b) Un objet virtuel en gris est ins r  dans la sc ne. Le cluster dans lequel il est inclu intersecte le shaft et est ins r    la liste des objets intersect s.

Une fois la nouvelle solution de radiosité mise à jour, l'éclairage indirect est calculé et la table d'affichage est mise à jour. La mise à jour se fait en deux passes différentes successives.

Tout d'abord, les changements visuels dus directement à l'insertion de l'objet, sans tenir compte de l'éclairage, sont localisés et mis à jour. Pour cela, la boîte englobante de l'objet inséré est projetée sur la fenêtre de vue. Ensuite, nous calculons le rectangle englobant de cette projection pour déterminer sur quels pixels les modifications doivent avoir lieu. Lors de l'insertion d'un objet, les pixels ainsi localisés sont visibles en gris clair sur la figure 5.18 (a). Lors du parcours de ces pixels potentiellement modifiés, les opérations suivantes ont lieu :

- Un nouvel « item buffer » est réalisé pour mettre à jour le numéro d'identification des polygones dans la zone de modification.
- Si le numéro d'identification est différent de celui anciennement affecté au pixel, la réflectance est remplacée par la réflectance du nouvel objet. Un nouveau point 3D est aussi calculé, en utilisant l'intersection d'un rayon depuis le point de vue avec cet objet (ce calcul est déjà utilisé dans la section 5.3.1).
- De plus, pour chacune des structures de source de lumière, un nouveau facteur de forme entre chaque source et le nouveau point 3D est calculé. Si le polygone associé au point 3D n'a pas été marqué modifié (c'est-à-dire qu'il n'a pas de shaft associé marqué modifié), une nouvelle valeur de visibilité est calculée. Ce cas peut apparaître plutôt lors du déplacement d'un objet ou lors de son enlèvement, quand des pixels deviennent à nouveau visibles depuis l'œil, mais pour lesquels il n'y pas de modification d'éclairage direct. Dans le cas où le polygone est marqué « modifié », la visibilité est seulement calculée lors de la deuxième passe, que nous allons présenter dans la suite. Cette distinction évite une redondance des calculs.



FIG. 5.18: Un objet virtuel est inséré dans la scène. (a) Les pixels localisés en gris clair sont ceux sur lesquels il y a des changements visuels potentiels à cause de la projection de l'objet virtuel sur l'écran. (b) Les pixels localisés en gris clair sont ceux pour lesquels il y a des changements de visibilité potentiels, dus aux ombres.

La deuxième passe consiste à mettre à jour les valeurs de visibilité pour les pixels af-

fectés par une modification d'éclairage direct. Lorsque l'objet est inséré dans la scène, certains shafts sont touchés, car l'objet virtuel les intersecte. L'objet intersecte donc des volumes source-récepteur, et dans ce cas, il y a un changement de visibilité, et notamment de nouvelles ombres. Les polygones récepteurs associés aux shafts touchés sont marqués de telle façon qu'on puisse rapidement les identifier dans la table de pixels. Pour mettre à jour la visibilité dans la table de pixels, nous parcourons tous les pixels de la table. Si le polygone associé à un pixel est marqué modifié, alors nous réalisons un nouveau calcul de visibilité pour chacune des sources associées à un shaft marqué modifié. Dans ce cas, la liste des objets intersectés associée au shaft permet d'accélérer les calculs. Ensuite, les nouveaux pixels sont affichés. Cette deuxième partie dans la mise à jour comprend des calculs de visibilité qui sont très coûteux. La localisation permet d'éviter des calculs inutiles compromettant les possibilités d'interactivité.

Déplacement d'un objet virtuel

Lors du déplacement d'un objet, le processus est pratiquement identique. Les différences se situent d'abord lors de la modification de la solution de radiosité, quand la hiérarchie des liens associés à l'objet est modifiée. Dans un souci d'efficacité, nous avons conservé l'objet et les structures associées dans la hiérarchie, plutôt que de les enlever puis les rajouter. Les liens attachés à l'objet virtuel sont conservés, mais les informations de visibilité et de facteur de forme associées sont mises à jour. Par contre, les shafts associés à ces liens sont détruits puis recréés car ils ont une composante géométrique. Puis la boîte englobante est projetée sur la fenêtre d'affichage, dans les positions avant et après le déplacement. Le rectangle contenant les pixels à traiter a donc une taille supérieure à celui calculé pour l'insertion de l'objet. La mise à jour des réflectances utilise la réflectance originale des objets réels, qui a été conservée. La suite de l'algorithme pour la mise à jour de la visibilité est identique à celle de l'insertion d'un objet. Les shafts intersectés par la position avant et après déplacement permettent d'identifier les régions potentiellement affectées.

Enlèvement d'un objet virtuel

L'algorithme est pratiquement identique pour enlever un objet que pour en insérer un. Au lieu d'insérer un nouvel objet dans la hiérarchie, l'objet en est enlevé. Tous les liens et les shafts attachés à cet objet sont détruits, l'objet enlevé de la hiérarchie (il n'est plus attaché à la racine) et une nouvelle solution de radiosité est calculée. L'algorithme de mise à jour de la table d'éclairage direct reste inchangé.

Ces algorithmes sont présentés sur la figure 5.19. Ils sont valables pour ajouter, déplacer ou enlever un objet virtuel. Les algorithmes utilisés pour manipuler les objets réels sont équivalents, mais comportent des différences importantes car nous ne connaissons pas ce qu'il y a derrière un objet réel enlevé.

Algorithme général

Modifier le système de radiosité
 Mise à jour de l'éclairage indirect dans le système de radiosité
 Modifier la table d'affichage

(a)

Modification du système de radiosité

si insertion
 Accrocher l'objet à la racine
 Créer les liens entre l'objet et les sources de lumière
sinon si déplacement
 Modification des coordonnées des polygones
 Mise à jour de l'information sur tous les liens de cet objet
sinon si suppression
 Retirer l'objet de la racine
 Enlever tous les liens de cet objet
Itérer
 Raffiner - « push-pull » - « gather »
 Modification des informations de visibilité sur les shafts (touchés)
 Marquage des polygones dont les shafts ont été modifiés

(b)

Modification de la table d'affichage

Localisation des pixels affectés
 Mise à jour des pixels affectés
 - polygone ID
 - Point d'intersection
 - Facteurs de forme
 - Réflectance
 Mise à jour de la visibilité pour les pixels marqués
 Mise à jour de l'éclairage indirect pour tous les pixels

(c)

FIG. 5.19: (a) Algorithme général. (b) Algorithme de mise à jour du système de radiosité. (c) Algorithme de mise à jour de la table d'affichage, pour l'insertion, la suppression, ou le déplacement d'un objet virtuel.

Enlèvement d'un objet réel

Nous avons vu comment enlever un objet virtuel. L'enlèvement d'un objet réel est un peu plus difficile car la connaissance des réflectances de la scène n'est valide que depuis un seul point de vue. Si nous enlevons un objet contenu dans la photographie initiale, des objets cachés vont devenir visibles depuis l'œil. Enlever un objet réel va donc produire des *trous*, c'est-à-dire des régions non connues car invisibles dans les images réelles. Pour pallier ce problème, nous utilisons un algorithme de remplissage par génération de texture. Cet algorithme n'est pas très coûteux mais n'est cependant pas interactif. Pour cette raison, nous le réalisons en pré-calcul avant la phase d'interactivité. La technique de génération de textures que nous utilisons est celle de Heeger et Bergen [HB95]. Nous utilisons en particulier l'implantation de El-Maraghi [EM]. Ces techniques de génération de textures sont décrites dans la section 5.5.2. Nous nous sommes restreints à des textures simples. Cependant, des algorithmes de génération de textures plus complexes pourraient être utilisés dans le cadre de notre problème [De 97].

Pour synthétiser les textures qui servent à masquer les régions inconnues, nous utilisons l'image de réflectance. L'avantage de l'image de réflectance est qu'elle ne contient plus d'effets dus à l'éclairage (ou qui sont au moins très diminués), comme les ombres ou la variation due aux facteurs de forme. Pour remplir la région non connue par une nouvelle texture, nous nous sommes inspirés de l'algorithme d'Igehy et Pereira [IP97]. L'application de l'algorithme est cependant différent car nous travaillons directement à partir d'une texture privée des effets d'éclairage, et nous connaissons la géométrie de la scène. Nous extrayons un échantillon de texture de réflectance sur les polygones ayant des régions potentiellement masquées. L'extraction est manuelle. Cet échantillon est ensuite passé en paramètre au programme de génération de texture, ainsi que la taille de la région à couvrir. La taille de cette dernière est calculée automatiquement, comme expliqué par la suite. Cette texture sert dans la phase interactive à initialiser les nouvelles réflectances. Ces textures générées sont stockées pour les objets marqués « enlevables » par l'utilisateur.

Lors de la phase d'interaction, l'utilisateur choisit l'objet à enlever. Les régions masquées par cet objet sont trouvées par un « item buffer ». Les nouveaux numéros d'identification sont calculés dans la zone masquée par l'objet. À nouveau, cette zone est déterminée par la projection de la boîte englobante de l'objet sur la fenêtre de vue. Dans les régions ainsi identifiées, deux masques sont créés. Le premier est un masque de pondération, le deuxième un masque de texture. Le masque de texture est rempli par les numéros d'identification que nous venons de calculer. Ce masque est utilisé pour déterminer quelle texture utiliser pour remplir le trou. Un exemple est illustré par la figure 5.20 (a). Le masque de pondération sert à déterminer la zone à traiter. Les pixels marqués '1' sont ceux à remplir, les pixels marqués '0' sont à laisser tels quels. Le masque est étendu à quelques pixels autour de la zone à modifier pour éviter des problèmes dus à une inexactitude géométrique, de façon à ne plus avoir les couleurs du contour de l'objet enlevé. Un exemple est montré sur la figure 5.20 (b). Les bords du masque, c'est-à-dire les frontières 0/1, sont rendus flous en établissant un fractionnement des valeurs (c'est-à-dire entre 0 et 1).

Dans la région identifiée par la projection de la boîte englobante de l'objet réel, les numéros d'identification des polygones maintenant visibles servent à remplir le masque

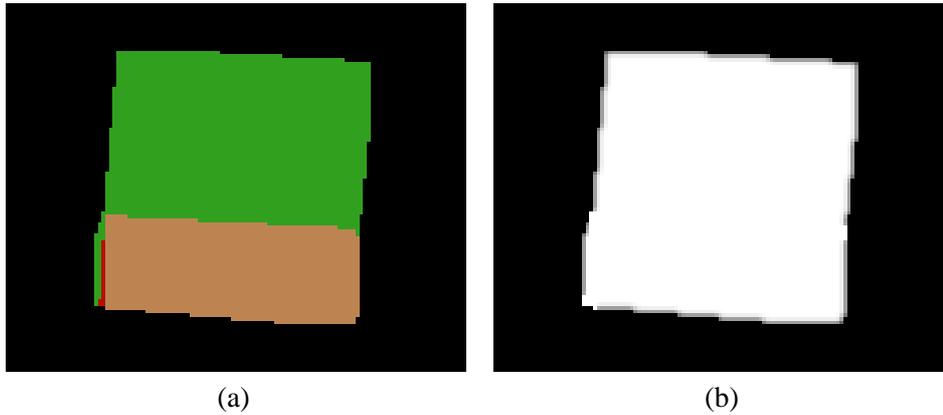


FIG. 5.20: Masques utilisés pour enlever l'ordinateur portable (voir la figure 5.21). (a) Masque représentant les numéros d'identification des polygones masqués par l'objet réel enlevé. (b) Masque indiquant en blanc la région touchée par l'enlèvement de l'objet. La région en noir n'est pas modifiée par le remplissage de texture.

de texture. Le calcul des nouvelles réflectances se fait en utilisant les deux masques. Les couleurs des pixels sont remplies :

- depuis les valeurs des textures générées, sélectionnées à partir du masque de texture, si les valeurs du masque de pondération est '1',
- depuis les anciennes valeurs de réflectance, si les valeurs du masque de pondération est '0',
- par une combinaison des deux, textures générées et anciennes réflectances, si les valeurs du masque de pondération est entre '0' et '1'.

Le fractionnement des valeurs sur les bords du masque est réalisé par un filtre appliqué sur chaque pixel. La valeur au centre du filtre est égale à la moyenne pondérée des pixels voisins.

Ensuite, le même algorithme que celui utilisé pour les objets virtuels est appliqué. Un exemple de génération d'une nouvelle image de réflectance est montré figure 5.21 (b). Le résultat après simulation de l'éclairage est montré en (c).

5.5.2 Génération de textures

De nombreux algorithmes existent pour générer des textures à partir d'échantillons. La difficulté du problème incite à définir des algorithmes adaptés à certains types de texture. La technique de génération de texture de Heeger et Bergen [HB95] par exemple fonctionne mieux sur des textures stochastiques, comme une texture de granite par exemple, que sur des textures déterministes, comme une image de mur en briques. En revanche, cette méthode a été testée sur un grand nombre d'échantillons, et a donné pour la majorité de bons résultats. L'algorithme est basé à la fois sur une pyramide directionnelle inversable (« self-inverting steerable pyramide »), et sur une procédure d'ajustement des couleurs. El-Maraghi a implanté cette méthode et son programme est accessible sur internet [EM].

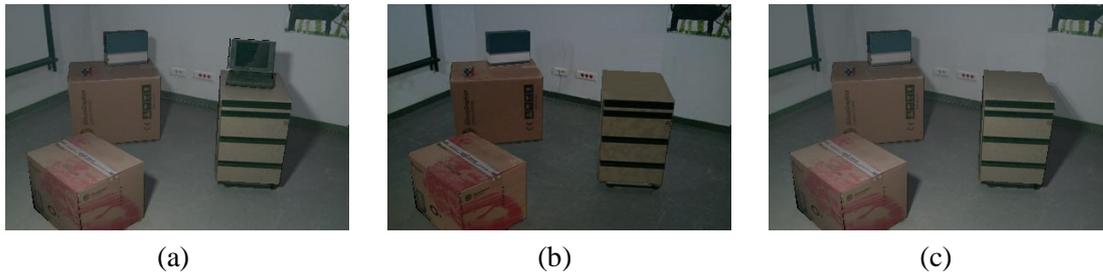


FIG. 5.21: L'ordinateur portable est enlevé virtuellement de la scène. (a) Scène originale. (b) Les réflectances sont mises à jour après un remplissage par génération de textures des régions précédemment cachées par l'ordinateur. (c) En utilisant ces nouvelles valeurs de réflectance, l'éclairage est simulé en modifiant localement l'éclairage direct.

Dans notre algorithme d'enlèvement d'objets réels, nous nous sommes inspirés des techniques utilisées par Igehy et Pereira [IP97]. Leur objectif est d'enlever une partie de l'image, et de remplir le trou laissé par une texture. Le résultat doit paraître réaliste. Pour atteindre cet objectif, ils modifient l'algorithme de Heeger et Bergen [HB95] de génération de texture, en y incluant des procédures spécifiques. Les données de l'algorithme de Heeger et Bergen sont un échantillon et un bruit. Igehy et Pereira ajoutent en entrée un masque et une image originale. Le masque est créé à la main. Ses valeurs sont de '0' dans la partie à enlever, des valeurs fractionnées autour, et de '1' dans la partie à conserver. L'échantillon de texture est également choisi par l'utilisateur. Contrairement à notre approche où la texture est synthétisée puis composée à l'image, la composition se fait au fur et à mesure de la génération de texture. Les limitations de cet algorithme sont celles de l'algorithme de génération de texture de Heeger et Bergen qui ne traite que les textures stochastiques. De plus, la région à modifier ne doit correspondre qu'à une seule surface, et les différences d'éclairage ne sont pas prises en compte.

Il existe une autre méthode efficace d'enlèvement de « défauts », comme par exemple un fil électrique ou une inscription sur un mur qu'on ne voudrait pas voir. Cette méthode est développée par Hirani et Totsuka [HT96]. Elle tient compte de l'image originale ce qui permet un résultat cohérent au niveau de l'éclairage. De plus, cette méthode semble permettre de travailler sur n'importe des types variés de texture, y compris les textures déterministes. L'algorithme modifie l'image bruitée en s'appuyant sur une image modèle. Les deux images sont dans un premier temps transformées par des opérations de transformation de Fourier (FFT). Les amplitudes et les phases ainsi déterminées vont servir à créer une nouvelle image, avec une transformée de Fourier Inverse (IFFT), où le bruit sera diminué. Cette nouvelle image est ensuite composée avec l'image originale, où un masque binaire détermine les régions où les pixels de l'image générée doivent être utilisés. Cette image composée sert ensuite de nouvelle entrée à l'algorithme itératif. Les exemples décrits par les auteurs utilisent typiquement 10 itérations pour enlever le bruit. Pour des textures déterministes, l'image contenant le défaut est découpée, et des échantillons adaptés à chaque découpage sont choisis. Les régions à modifier et les échantillons modèles associés sont choisis à la main. Cet algorithme présente l'avantage sur le précédent de pou-

voir travailler sur des types variés de texture, et d'enlever des défauts même à cheval sur différentes surfaces. De plus, la composition permet de conserver les aspects d'éclairage, s'il est similaire dans le reste de l'image. Par contre, cet algorithme ne peut fonctionner que si les défauts sont de petite taille par rapport à l'image. Cet algorithme pourrait éventuellement être appliqué pour étendre l'enlèvement d'objet dans n'importe quelle zone de la scène, où les textures seraient plus variées.

5.5.3 Modification de l'éclairage

Nous avons auparavant décrit les algorithmes nous permettant d'ajouter, de déplacer ou d'enlever des objets virtuels, et réels. Nous décrivons maintenant les algorithmes permettant de modifier l'éclairage, en changeant l'intensité des sources de lumière, ou en ajoutant des lampes virtuelles.

Modification de l'intensité des sources

Lorsqu'on modifie l'intensité d'une source de lumière, deux opérations doivent être réalisées :

- le calcul de l'éclairage indirect avec la nouvelle valeur d'exitance, pour mettre à jour le terme d'éclairage indirect \hat{B} dans la structure de données. Pour cela, on utilise le système de radiosité puis on fait un rendu dans un tampon mémoire, comme précédemment (voir section 5.4.3). Comme il n'y a aucun changement de visibilité, seules des itérations de « gather » et « push-pull » sont réalisées par le système de radiosité (voir la section 2.2.1 à propos de la radiosité). Ce calcul est très rapide étant donnée la subdivision grossière choisie.
- le changement de l'exitance de la source en question dans la structure de données, servant à calculer l'éclairage direct.

Le temps mis pour la modification dépend de la taille de l'image et de la rapidité de la solution de radiosité. Il n'y a aucun calcul supplémentaire en ce qui concerne la visibilité ou la structure de la solution de radiosité.

Addition de sources de lumière virtuelles

Lors de l'insertion d'une source de lumière, il y a plus d'opérations à faire. Il faut d'abord modifier la structure de liens dans le système de radiosité hiérarchique, et créer les « shafts » associés aux nouveaux liens. Ceci nous permet à la fois de calculer un éclairage indirect tenant compte de la nouvelle lampe virtuelle, mais également d'avoir une structure efficace pour faciliter les calculs de visibilité.

Comme lors de l'insertion d'un objet virtuel, un test est réalisé pour connaître les pixels visuellement modifiés, en projetant la boîte englobante de la source sur la fenêtre de vue. Très souvent dans nos scènes, la source de lumière ajoutée est hors du champ visuel.

En ce qui concerne l'éclairage, pour chaque pixel, chaque champ de la nouvelle structure de données associée à la nouvelle lampe s est mis à jour :

- l'exitance E_s ,
- la visibilité V_s entre la source et le point 3D associé à ce pixel. Pour cela, nous utilisons les « shafts » créés lors de la solution de radiosité. Si un shaft ne contient aucun polygone dans sa liste de polygones intersectés, il n'y a pas de calcul de visibilité à faire et le facteur de visibilité est mis à '1' (complètement visible). Sinon, un calcul de visibilité est réalisé par un lancer de rayons entre la source et le point 3D. Comme nous sommes dans une session d'interaction, nous n'utilisons que quatre rayons, ce qui introduit de « l'aliasing » dans la pénombre. Nous limitons les calculs d'intersection entre rayons et polygones aux polygones de la liste associée au « shaft ».
- le facteur de forme F_s entre la source s et le point 3D associé au pixel. Ce facteur de forme est calculé comme lors de la phase d'initialisation.

Une nouvelle structure de lampe est ensuite ajoutée dans la liste des structures associées aux lampes dans la structure de données d'affichage (et éclairage direct) (voir la figure 5.11).

5.6 Résultats

Nous avons implanté et testé ces algorithmes interactifs sur plusieurs scènes, sur une station de travail *SGI Onyx2 Infinite Reality*, avec un processeur R10000, à 195 Mhz. Les résultats sont montrés sur des vidéos accessibles sur internet[‡]. La première scène testée correspond à celle présentée tout au long du chapitre. L'image originale est présentée sur la figure 5.22 (a), la reprojction virtuelle simulée par notre programme est en (b). Cette image est de 400 par 300 pixels. Nous avons tout d'abord enlevé l'ordinateur portable posé sur un meuble à tiroir en (c). Cet enlèvement d'objet prend 1.2 secondes. Ce temps ne prend pas en compte la génération de texture qui a été réalisée en pré-calcul (30 secondes pour chaque texture générée). En revanche, les masques pour remplacer l'ancienne réflectance par la nouvelle sont calculés dans cette phase d'interactivité. Nous avons ensuite inséré deux objets, une sphère (polyèdre) en (d) comprenant 10 polygones en 1.54 secondes, et une cafetière en (e) comprenant 200 polygones en 5.65 secondes. La cafetière correspond à un objet réel reconstruit auparavant en utilisant le système de reconstruction à base d'images *Rekon* [POF98]. Nous l'avons placée à l'endroit où se trouvait l'ordinateur portable. Nous avons ensuite déplacé la sphère, en (f) et (g). La mise à jour se fait en moins de 1 seconde par image. Le déplacement de la cafetière met un temps supérieur (3 secondes par image).

Sur la figure 5.23, nous avons ensuite ajouté une première lampe à droite dans la scène en (a), puis une deuxième à gauche dans la scène en (b). Chaque insertion de lampe met 12 secondes. Ce temps est supérieur à celui réalisé pour l'insertion d'un objet car nous devons calculer des facteurs de forme et des facteurs de visibilité pour tous les pixels de l'image. Enfin, nous avons éteint la lampe créant l'éclairage original en (c). L'extinction de la lampe se fait en 0.4 secondes. Le déplacement d'un objet comme la sphère (en (d) et (e)) met un peu plus longtemps avec les nouvelles lampes, typiquement 4.5 secondes, car une mise à jour des paramètres est nécessaire pour chacune des lampes.

[‡]<http://www-imagis.imag.fr/Membres/Celine.Loscos/relight.html>

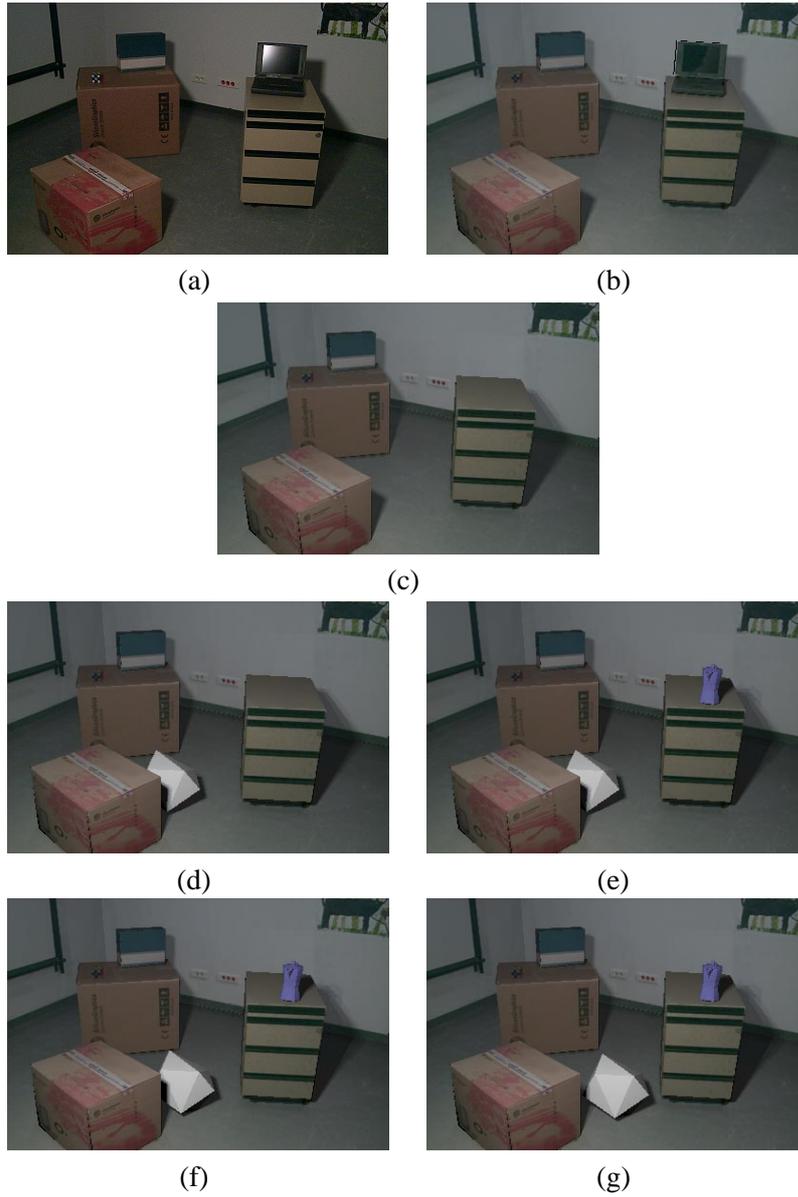


FIG. 5.22: Modifications de la géométrie pour la première scène testée.

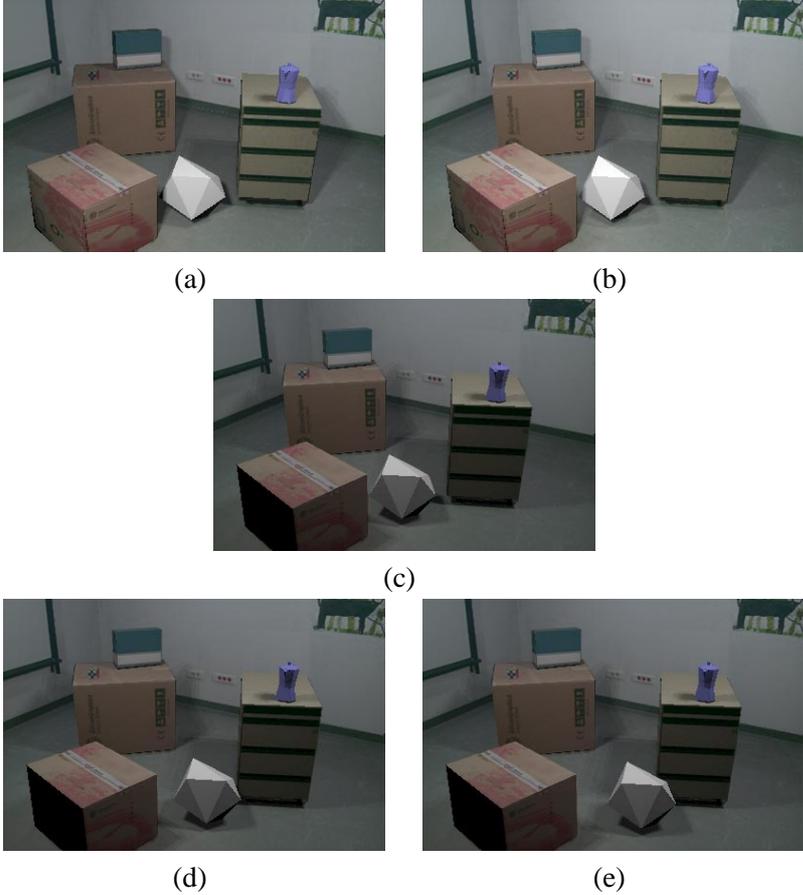


FIG. 5.23: Modifications de l'éclairage pour la première scène testée.

Nous avons testé une autre scène dont l'image originale (512 par 480 pixels) est présentée par la figure 5.24 (a). La réflectance estimée pour cette scène est présentée en (b). Nous avons utilisé six positions de lampes différentes pour calculer cette image de réflectance. La simulation d'éclairage par le système est présentée en (c). A nouveau, cette image virtuelle est similaire à celle d'origine. Avec cette scène, nous voulions montrer que nous pouvions réaliser des modifications très difficiles à faire dans la réalité. Nous avons donc enlevé la porte virtuellement, en (d). Cette opération prend 3.3 secondes, sans compter le temps de génération de textures calculée antérieurement. Cette opération met plus longtemps que l'enlèvement du portable dans la scène précédente. La taille de la zone à traiter est en effet plus grande, de même que la taille de l'image. L'affichage sur cette scène prend 0.17 secondes (le temps d'affichage pour la première scène prend 0.1 secondes). Nous avons ensuite ajouté une chaise virtuelle en (e), ce qui a pris 5.37 secondes. Cette chaise comporte 185 polygones. Le temps de mise à jour est à nouveau supérieur à celui du temps d'insertion des objets dans la scène précédente, car la zone à traiter a une taille supérieure. Enfin nous avons inséré une lampe virtuelle dans cette scène en (f), projetant les ombres de la chaise sur le mur où se trouvait la porte. Le temps mis pour l'insertion est de 7.6 secondes.

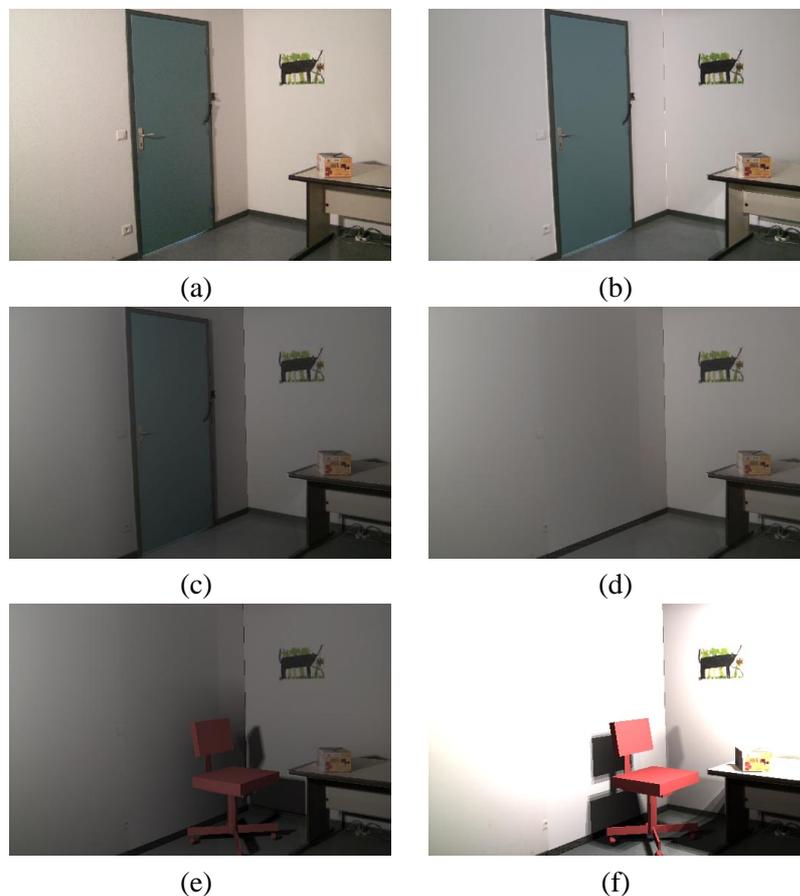


FIG. 5.24: Résultats sur une deuxième scène testée.

5.7 Conclusion

Nous avons présenté dans ce chapitre une nouvelle approche pour ré-éclairer et modifier géométriquement les scènes réelles. Notre approche est basée sur un algorithme comprenant une phase de pré-calcul dans laquelle les propriétés de réflectance diffuses des surfaces réelles sont approximativement retrouvées. Nous utilisons pour cela une séquence de photographies, appelées images de luminance, prises sous un même point de vue, mais sous des éclairages différents. Après avoir reconstruit un modèle 3D simplifié de la scène réelle, les images de réflectance associées à chaque image de luminance sont créées en utilisant les informations géométriques fournies par le modèle, et les informations de couleur contenues dans les photographies. Ces images de réflectance contiennent des imperfections, notamment dans les régions d'ombre. Elles sont combinées en calculant une moyenne des pixels pondérée par la confiance que nous avons en chacun des pixels. Cette confiance a été calculée par un algorithme de filtrage, utilisant les valeurs de visibilité, et en détectant les valeurs singulières. L'image obtenue par combinaison des images de réflectance est une image de réflectance, dont les couleurs sont homogènes et les effets de lumières sont estompés.

Les valeurs de réflectance sont associées à chaque pixel d'une structure de données augmentée. Le rendu de l'éclairage se fait en combinant un lancer de rayons pour l'éclairage direct, et une méthode de radiosit  pour l'éclairage indirect. Le rendu se fait pixel par pixel en utilisant la structure de données augment e.

Lorsque le pr -calcul est termin , nous pouvons modifier de fa on r aliste la g om trie et l' clairage de la sc ne r elle en temps interactif. Des algorithmes adapt s sont utilis s pour localiser efficacement les pixels affect s par les modifications.

Dans cette m thode nous avons lev  le probl me du traitement des r gions d'ombre en les compensant par une meilleure connaissance de l' clairage de la sc ne. Ceci nous a permis d'estimer les r flectances de fa on plus pr cise que dans la premi re m thode pr sent e au chapitre 4. Les temps de mises   jour lors des modifications sont cependant  quivalents ou plus lents. De plus nous sommes toujours restreints par un point de vue fixe. Par contre, les structures de donn es utilis es nous permettent d'enlever automatiquement les objets r els de la sc ne de fa on convaincante. Cette m thode s'appr te plus   des extensions, et le contr le pixel par pixel permet une estimation des param tres et une simulation plus fines.

Nous pensons que l'estimation des r flectances pourraient  tre am lior e par l'utilisation d'images HDR, et par une it ration sur le calcul de la r flectance, en utilisant au fur et   mesure un nouveau calcul de l' clairage indirect. Nous discutons de ces am liorations dans le chapitre suivant.

