

## Chapitre 4

# Modification des propriétés d'éclairage d'une scène réelle à partir d'un seul éclairage connu

Comme nous l'avons souligné dans les chapitres précédents, l'objectif général de cette thèse est d'apporter des solutions au problème du ré-éclairage interactif des scènes réelles. Un exemple d'application de telles méthodes est la compositions d'éclairage pour des scènes d'intérieur. Nous souhaitons fournir des solutions pour permettre la modification interactive des propriétés d'éclairage d'une scène réelle.

Dans ce chapitre, nous décrivons une méthode permettant d'atteindre cet objectif, avec une méthode de capture simple et l'utilisation des cartes d'accélération graphique. Nous utilisons en tout une douzaine d'images d'entrée (quatre mosaïques) pour créer un modèle géométrique de la scène réelle. Des textures sont ensuite extraites depuis une seule mosaïque, avec l'éclairage d'origine. Nous utilisons ensuite ce modèle texturé pour simuler l'éclairage et le modifier, ce qui permet la manipulation virtuelle et interactive de l'intensité des lampes réelles présentes dans notre modèle, et l'insertion d'objets et de lampes virtuels. La modification de l'intensité des lampes réelles est un problème difficile, car les textures contiennent déjà l'éclairage réel initial, et en particulier des ombres.

Parallèlement à notre travail, d'autres solutions ont été publiées [YDMH99, YM98]. Nous les avons décrites précédemment dans la section 2.4. Toutes nécessitent une connaissance de la scène sous plusieurs éclairages, et la résolution se fait pixel par pixel. Nous proposons une approche moins complexe, en nous contentant d'une seule image en entrée du système d'éclairage. Pour permettre le ré-éclairage, nous modifions les textures du modèle pour qu'elles ne prennent plus en compte les occlusions dans l'éclairage qu'elles représentent. Pour cela, nous utilisons une méthode de radiosité hiérarchique\*, et nous nous appuyons sur les travaux de Fournier *et al.* [FGR93] et de Drettakis *et al.* [DRB97]. En utilisant un système de radiosité, nous nous restreignons à traiter le cas des scènes diffuses (émissions et réflectances diffuses). Les textures originales comprenant les effets d'éclairage tels que les ombres, sont transformées en textures correspondant à un éclairage

---

\*Pour le lecteur qui ne connaît pas la radiosité, nous suggérons la lecture préalable de la section 2.2 du chapitre 2

calculé sans prise en compte de la visibilité. Pour modifier les textures, nous utilisons le matériel d'accélération graphique en affichant les polygones texturés aux feuilles de la subdivision de la radiosité hiérarchique. La phase interactive permet de modifier l'éclairage de la scène, d'insérer et de déplacer des objets virtuels.

## 4.1 Présentation générale du système

Notre objectif est de permettre la modification interactive de l'intensité des lampes réelles (et notamment l'extinction totale de l'éclairage réel), l'insertion des lampes virtuelles et la modification de leurs intensités, ainsi que l'insertion et le déplacement interactif des objets virtuels. Toutes les mises à jour doivent permettre d'obtenir un éclairage commun cohérent et précis.

L'algorithme que nous présentons dans ce chapitre comporte trois parties : une partie pour reconstruire et à initialiser le système, une partie de pré-calcul, qui permet en particulier de modifier les textures originales, et une partie d'interaction avec le système. La figure 4.1 présente l'algorithme complet.

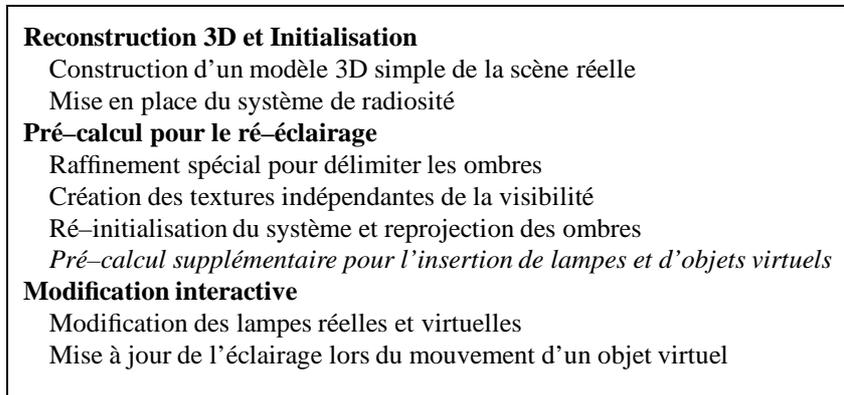


FIG. 4.1: Algorithme complet.

Nous reconstruisons tout d'abord un modèle géométrique texturé représentant la scène réelle. Ce modèle est utilisé pour initialiser le système de radiosité. Cette initialisation est présentée dans la section 4.2. La deuxième partie de l'algorithme est une phase de pré-calcul qui permet de corriger les textures, et de créer des textures d'éclairage sans occlusion, puis qui ré-actualise le système à partir de ces nouvelles textures (voir section 4.3). Enfin, nous utilisons des algorithmes adaptés (voir section 4.4), afin de modifier interactivement la scène réelle. Les résultats sont présentés dans la section 4.5.

## 4.2 Initialisation du système

Dans un premier temps, nous retrouvons un modèle géométrique texturé de la scène réelle, à partir de photographies de celle-ci. Nous utilisons pour cela un des outils présentés

dans le chapitre précédent en section 3.2. Les textures utilisées sont dépendantes du point de vue. Ceci présente un grand avantage, qui est de préserver les détails de perspective tout en ayant un modèle simple. En revanche, le modèle ne peut être visualisé dans notre système que d'un seul point de vue. Le modèle est décomposé en deux ensembles. Le premier ensemble contient les objets visibles sur la mosaïque servant à la reconstruction. Cette mosaïque est appelée *photographie originale*. Pour une de nos scènes de test, c'est par exemple la mosaïque en haut à gauche sur la figure 4.2. Les objets non visibles sur cette photographie, sont placés dans un deuxième ensemble et sont traités différemment.

Le modèle visible sur la photographie originale est reconstruit en utilisant les outils semi-automatiques de reconstruction à partir d'images. Sa reconstruction par l'outil *TotalCalib* [BR97] est illustrée par la figure 4.2. Le reste de la scène est reconstruit de façon grossière à la main, et texturé par des textures représentatives, de basse résolution. Les sources de lumière sont modélisées à la main. Les objets du reste de la scène ainsi que les lampes auraient pu être modélisés en utilisant un outil de reconstruction. Cependant, pour nos besoins, il s'avère que le modèle non visible sur l'image n'a pas besoin d'être précis. Seule une géométrie grossière suffit pour simuler les échanges lumineux. De plus, ceci nous évite de multiplier les données photographiques. Un exemple de modèle reconstruit est présenté sur la figure 4.3. Dans ce modèle, on distingue le modèle texturé dans le fond, les murs et un ensemble de bureaux reconstruits grossièrement sur le devant, et quatre lampes à néons (A, B, C, D) représentées par des rectangles.

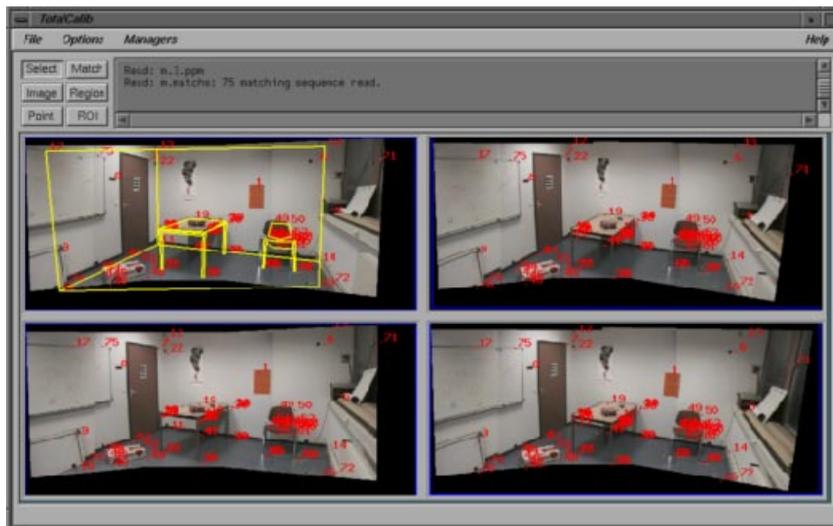


FIG. 4.2: Le modèle reconstruit par un outil de reconstruction à base d'images, ici *TotalCalib*.

### 4.2.1 Initialisation de la radiosité hiérarchique

Le modèle 3D ainsi reconstruit est utilisé dans notre système de radiosité hiérarchique. Avant de calculer une solution de radiosité pour le modèle de la scène réelle, nous devons estimer plusieurs paramètres : la réflectance des objets et l'intensité des sources. Ces

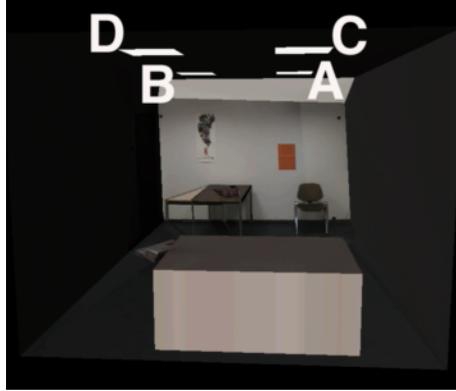


FIG. 4.3: Le modèle complet incluant quatre lampes réelles (A, B, C, D).

paramètres sont fixés en utilisant les équations de Fournier *et al.* [FGR93] et de Drettakis *et al.* [DRB97], qui estiment les paramètres directement à partir des textures (c'est-à-dire des images d'entrées). Pour chaque polygone, une valeur de réflectance diffuse est déterminée en utilisant la moyenne des intensités des pixels. L'équation est basée sur la définition de la réflectance, égale à l'énergie émise divisée par l'énergie reçue. Pour chaque polygone  $i$ , la réflectance  $\rho_i$  est :

$$\rho_i = \frac{\hat{B}_i}{\hat{B}_A} \times \hat{\rho} \quad (4.1)$$

où  $\hat{B}_i$  est la moyenne de l'intensité des pixels identifiés par la projection de cet objet sur l'image d'origine (photographie). à partir de la texture de  $i$ ,  $\hat{B}_A$  est l'intensité moyenne de toute la scène, calculée à partir de la photographie originale, et  $\hat{\rho}$  est la réflectance moyenne de la scène. Dans un premier temps,  $\hat{\rho}$  est fixée par l'utilisateur. D'après l'équation d'estimation (4.1), les valeurs de réflectance dépendent de la couleur des textures, et sont donc plus sombres dans les régions d'ombre. Dans un deuxième temps, la réflectance moyenne  $\hat{\rho}$  est ré-estimée en faisant la moyenne des valeurs de réflectance calculées. Puis, les réflectances sont mises à jour en utilisant l'équation (4.1) pour cette nouvelle réflectance moyenne.

Ensuite, l'exitance  $E_i$  des sources de lumière réelles est estimée en utilisant l'équation :

$$\sum_i E_i A_i = (1 - \hat{\rho}) \hat{B}_A \sum_i A_i \quad (4.2)$$

où  $A_i$  est l'aire du polygone  $i$ . Cette approximation est basée sur l'estimation d'un terme ambiant dans l'algorithme de radiosit  progressive [CCWG88]. Pour simplifier la r solution, et  tant donn  que c'est le cas dans nos sc nes, les sources de lumi re sont consid r es comme ayant la m me exitance. Un syst me d' quations pourrait  tre r solu pour le cas de lampes ayant des exitances non-homog nes.

Une fois que la r flectance et l'intensit  des sources ont  t  estim es, le syst me de radiosit  hi rarchique est initialis . La sc ne est subdivis e et les liens sont cr es. Lors de la subdivision, les r flectances sont mises   jour pour chaque nouvel  l ment de subdivision

en utilisant l'équation (4.1), à partir des textures correspondantes. De cette initialisation est issue une radiosité  $B_i$  pour chaque élément. Cette radiosité ainsi calculée sert à initialiser un facteur d'affichage  $D_i$  ( $D_i = B_i$ ). Si la radiosité de la scène est modifiée, ce facteur permet de moduler les textures et ainsi de prendre en compte les modifications d'éclairage. Ceci se fait en utilisant l'équation suivante pour chaque élément  $i$  du maillage de radiosité<sup>†</sup> :

$$\frac{B_i}{D_i} \times T_i \quad (4.3)$$

où  $T_i$  est la texture de l'élément  $i$ . Un exemple de maillage est montré plus loin sur la figure 4.6. Par exemple, si un objet virtuel est inséré dans la scène, la valeur de radiosité dans ses ombres sera inférieure au facteur d'affichage  $T_i$ . À cet endroit, la texture sera donc assombrie. Ceci est montré par la figure 4.4. En (a), la radiosité n'est pas modifiée. Le rapport d'affichage permet de visualiser le modèle texturé sans modification d'éclairage dans notre système. En (b), un objet virtuel a été inséré en utilisant la méthode de radiosité dynamique de Drettakis *et al.* [DRB97]. Les ombres de l'objet sont projetées sur la table en utilisant le rapport d'affichage décrit par l'équation (4.3) (voir le zoom sur l'image en (c)).

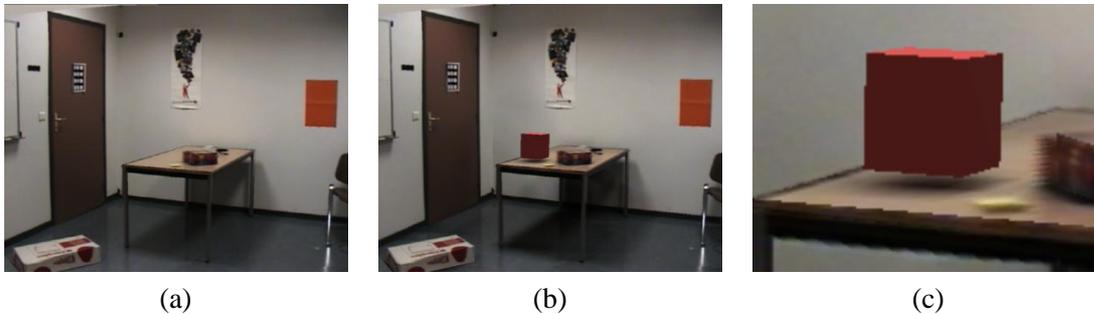


FIG. 4.4: (a) La scène vue dans notre système de radiosité hiérarchique. (b) Un objet virtuel est inséré dans la scène. Les ombres de l'objet virtuel sont projetées sur la table. (c) Zoom sur la projection des ombres virtuelles.

Il est important de noter que les valeurs de radiosité n'ont pas besoin d'être très précises. Comme l'affichage se fait grâce à une modulation par un rapport qui est de un s'il n'y a pas de modifications, ce qui importe est que la modification de la valeur de radiosité  $B_i$  soit cohérente avec les valeurs précédentes. De plus, l'œil humain perçoit surtout les différences d'intensités, plutôt que les intensités elles-mêmes.

#### 4.2.2 Limitations du système ainsi initialisé

En utilisant le système initialisé avec les équations de Fournier *et al.* [FGR93] et de Drettakis *et al.* [DRB97], nous ne pouvons pas modifier les lumières réelles de façon satisfaisante. L'effet est illustré par la figure 4.5.

Dans la figure (b), la scène n'est éclairée que par une seule lampe, celle de droite. Pourtant, les ombres dues à la lampe de gauche sont toujours perceptibles. Ce résultat est

<sup>†</sup>En pratique, les valeurs sont interpolées aux sommets de chaque élément pour donner un aspect lisse

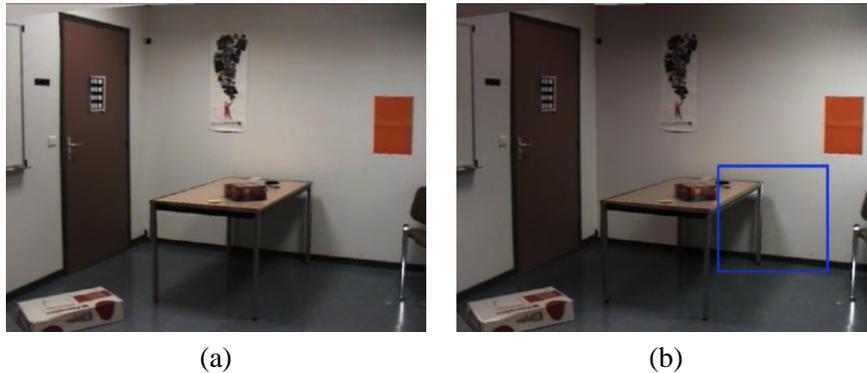


FIG. 4.5: (a) Éclairage original de la scène. Deux lampes projettent les ombres de la table sur le mur qui nous fait face. (b) En utilisant la méthode de Drettakis *et al.* [DRB97], nous éteignons la lampe de gauche. Les ombres dues à cette lampe sont toujours visibles sur le mur, alors qu'elles ne le devraient pas.

prévisible si on utilise la méthode de Drettakis *et al.* [DRB97]. En effet, l'affichage est réalisé directement en utilisant les textures de la photographie originale. Les textures contiennent les effets d'éclairage originaux, tels que les ombres des objets réels. Lors de l'affichage, elles sont modulées par un facteur qui est la radiosité potentiellement modifiée, divisée par la radiosité originale (c'est-à-dire le facteur d'affichage).

Dans la figure 4.5 (b), la région encadrée contient l'ombre de la table, due à la lampe de gauche, et une région qui ne contient pas d'ombre. En éteignant la lampe de gauche, la radiosité courante dans cette région varie peu, en fonction des valeurs de facteurs de forme. La texture dans les deux régions est modulée par la radiosité modifiée divisée par la radiosité originale. Comme la région d'ombre sur la texture est plus sombre que la région qui ne l'est pas, elle reste plus sombre après la modification de l'éclairage, car le rapport radiosité modifiée sur radiosité originale est similaire dans les deux régions. Pour avoir un résultat correct, il faudrait que la texture affichée ait une couleur similaire dans les deux régions. Les deux méthodes précédentes ne peuvent donc pas traiter de façon satisfaisante la modification de l'éclairage due aux lampes réelles.

Nous levons dans ce chapitre la limitation des deux méthodes précédentes, en transformant les textures pour qu'elles représentent un éclairage calculé sans prise en compte de la visibilité.

### 4.3 Pré-calcul pour permettre le ré-éclairage

Dans notre méthode, nous utilisons le rapport d'affichage décrit dans l'équation (4.3), permettant d'exploiter les capacités d'accélération graphique du matériel. Nous initialisons les paramètres du système à partir des textures, comme nous l'avons présenté dans la section précédente. Afin de permettre le ré-éclairage de la scène en modifiant l'intensité des lampes réelles, nous ajoutons un algorithme de pré-calcul (la méthode est très différente des méthodes précédentes [FGR93, DRB97]). Grâce à cet algorithme, nous construisons des textures correspondant à un éclairage indépendant de la visibilité, c'est-à-dire

que la lumière ne rencontre pas d'obstacle avant d'arriver sur l'objet.

La création de ces textures se fait en deux étapes. Dans une première étape, nous ajoutons la lumière bloquée à celle déjà présente dans les textures, en utilisant l'information contenue dans la structure de la radiosit  hi rarchique.   la fin de cette premi re  tape, la texture comporte des imperfections, principalement dues aux nombreuses approximations faites dans notre syst me. Pour r soudre ces probl mes, nous appliquons, dans une deuxi me  tape, une correction bas e sur la couleur d'un  l ment de texture de r f rence, donnant une indication sur la couleur de texture   obtenir.   la fin de cette  tape, les textures obtenues repr sentent effectivement un  clairage calcul  sans tenir compte de la visibilit  ou «sans occlusion».

Pour chaque  tape, il est important d'avoir un maillage adapt , d limitant correctement les r gions d'ombre. Nous avons pour cela d velopp  un raffinement bas  sur la couleur des textures que nous d crivons dans la section suivante. Ensuite nous d crivons les deux  tapes pour la modification des textures. Nous terminons par l'initialisation du syst me de radiosit  bas  sur les nouvelles textures.

#### 4.3.1 Raffinement bas  sur les textures pour d limiter les ombres

Nous d sirons appliquer des modifications pour obtenir des textures repr sentant l' clairage sans ombres. Les r gions o  il y a le plus de modifications   faire sont les r gions o  l'on per oit les ombres. Pour mieux d limiter les zones d'ombre, nous avons besoin d'un maillage adapt , localement plus fin dans les r gions d'ombre, surtout dans les zones de p nombres. Si on utilise les m thodes standard de raffinement, telles que le raffinement   crit re BF [HSA91] ou dirig  par le contr le de l'erreur [GH96a], nous n'obtenons pas un maillage adapt    nos besoins. Le probl me principal de ces approches est qu'elles ne garantissent pas une bonne d limitation des ombres, m me en utilisant un facteur de visibilit  [HSA91]. De plus, les limites de ces raffinements se per oivent bien dans notre cas, o  la g om trie reconstruite et la visibilit  calcul e par lancer de rayons sont approximatives. Le maillage de discontinuit  [LTG93] se r v le tout aussi inadapt , et serait incorrect  tant donn  que la g om trie de nos sc nes n'est pas pr cise. En cons quence, nous gardons la subdivision par maillage de type «quadtree» (voir section 2.2.1), mais nous utilisons un nouveau raffinement bas  sur la texture.

L'id e g n rale est d'utiliser l'information de couleur contenue dans les textures, qui sont extraites directement des photographies, ainsi que l'information de visibilit  accessible par le syst me de radiosit  hi rarchique. Les ombres des objets r els sont incluses dans les textures, et sont distingu es par des r gions de couleur plus sombre. En utilisant le type de la visibilit  (VISIBLE, PARTIEL ou INVISIBLE) associ  aux liens, et la diff rence de couleur entre deux  l ments voisins, nous pouvons forcer la subdivision dans les r gions d'ombre.

Ce raffinement a lieu apr s la phase d'initialisation d crite dans la section pr c dente (section 4.2.1), qui a permis d'initialiser les param tres de la sc ne, tels que la r flectance, et l'intensit  des sources de lumi re. La subdivision initiale, obtenue avec un crit re de raffinement BF, est grossi re, comme le montre la figure 4.6 (a).   la fin de cette solution, tous les liens depuis les sources de lumi re sont attach s aux feuilles de la hi archie

des éléments, afin de procurer une meilleure précision sur l'information de visibilité (voir figure 4.6 (b)).

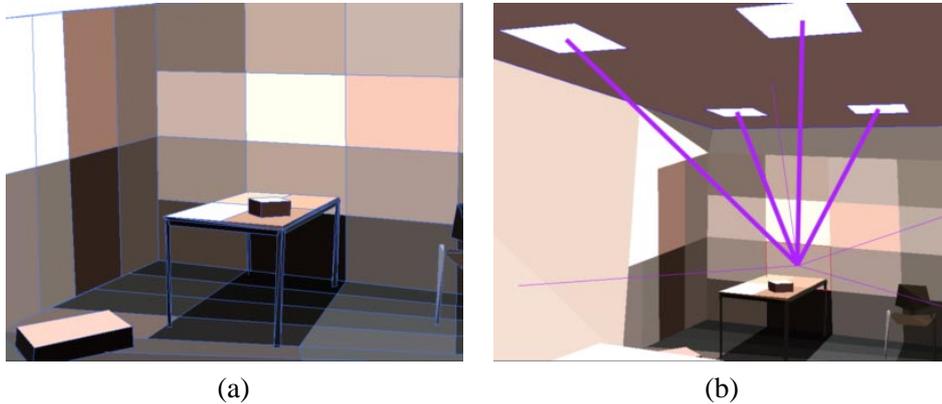


FIG. 4.6: (a) Subdivision grossi re apr s la premi re solution de radiosit , initialisant les param tres. (b) Tous les liens depuis les sources de lumi res sont attach s aux feuilles de la hi rarchie pour assurer une meilleure pr cision dans les valeurs de visibilité.

L'algorithme de raffinement bas  sur la texture compare la visibilité et la couleur de deux  l ments feuilles voisins. La visibilité doit  tre coh rente avec les diff rences de couleur. Nous consid rons deux cas, pour un  l ment et ces voisins :

1. Si les deux  l ments ont des couleurs similaires, ils doivent avoir une visibilité similaire. Si ce n'est pas le cas, on subdivise.
2. Si les deux  l ments ont des couleurs diff rentes, ils doivent avoir une visibilité diff rente. Si ce n'est pas le cas, on subdivise.

Nous consid rons que la visibilité est similaire, si la visibilité est de m me type (VISIBLE, PARTIEL, INVISIBLE) et si la diff rence entre les facteurs d'att nuation est inf rieure   un certain seuil fix  par l'utilisateur. De la m me fa on, nous consid rons que deux  l ments ont une couleur similaire si leur diff rence de couleur est inf rieure   un certain seuil fix  par l'utilisateur. Pour calculer la diff rence de couleur, nous convertissons les valeurs RGB en valeurs CIE LAB [Fai98]. En effet, les couleurs RGB ne sont pas facilement comparables. Le mod le CIE LAB est plus repr sentatif des variations de couleurs telles qu'elles sont per ues par le syst me visuel humain. La distance de couleur calcul e dans ce mod le est  galement plus similaire   l'interpr tation de distances de l' il humain. La transformation des couleurs RGB en couleurs CIE LAB est ex cut e en deux temps. Les valeurs sont d'abord converties dans l'espace de couleurs CIE XYZ, puis dans l'espace CIE LAB [Fai98]. La distance est ensuite calcul e simplement par une  quation de distance euclidienne :

$$Distance_{couleur}(i, n) = \sqrt{(L_i - L_n)^2 + (A_i - A_n)^2 + (B_i - B_n)^2}. \quad (4.4)$$

L'algorithme complet de raffinement est montr  sur la figure 4.7. Cet algorithme traverse la hi rarchie des  l ments de fa on r cursive et compare chaque paire d' l ments voisins. Un  l ment est subdivis  s'il ne respecte pas les conditions du cas 1 et 2 ou si

dans le cas 2, le type de visibilité est PARTIEL. Cette dernière condition nous permet de mieux raffiner dans les régions de pénombre. L'algorithme examine ensuite les éléments fils créés. Si aucun changement de visibilité n'est détecté, la subdivision est annulée et l'élément devient à nouveau une feuille de la hiérarchie.

**Raffinement pour délimiter les ombres**

**Pour** chaque feuille  $i$ , comparaison avec les éléments feuilles voisins  $n$   
**si**  $i$  a une couleur similaire par rapport à  $n$   
     **et** une visibilité depuis les sources de lumière différente  
**alors** subdiviser  $i$   
**sinon si**  $i$  a une couleur différente de celle de  $n$   
     **et** une visibilité depuis les sources de lumière similaire  
**alors** subdiviser  $i$   
**sinon si** type = PARTIEL **alors** subdiviser  $i$   
**si**  $i$  a été subdivisé **alors**  
     **si**  $i$  n'a pas de différence de visibilité depuis  
         les sources de lumière avec ses éléments fils  
     **alors** annuler la subdivision de  $i$  ( $i$  est à nouveau une feuille)  
**sinon pour** chaque fils de  $i$   
     recommencer l'algorithme

FIG. 4.7: Algorithme du raffinement basé sur les textures pour délimiter les régions d'ombre.

Le cas numéro 1 apparaît dans les limites d'ombre. Cette subdivision aide à avoir un maillage plus fin dans les limites entre les régions d'ombre, et les régions éclairées directement par les sources de lumière. La subdivision d'un élément est arrêtée lorsque le niveau maximum de subdivision (basé sur la taille minimum de l'élément) est atteint, ou si les conditions des cas 1 et 2 sont respectées.

Le cas numéro 2 apparaît quand le lancer de rayons a échoué pour identifier correctement la visibilité. L'élément peut être non-subdivisé si la différence de couleur n'est pas due à une région d'ombre, mais à un changement de couleur dans la texture, comme c'est le cas pour le poster orange sur le mur sur la figure 4.8 (a).

À la fin du raffinement, la subdivision est fine dans les régions d'ombre (surtout de pénombre), et moins fine dans les régions visibles. La subdivision est montrée sur la figure 4.8 (b). Durant le raffinement, les réflectances ont été mises à jour.

Pour préparer la phase de modification de texture, les liens depuis toutes les sources (sources de lumière, et autres éléments qui sont des sources secondaires) sont attachés aux feuilles de la hiérarchie des éléments. Des itérations de *gather* et *push-pull* sont réalisées avec cette nouvelle subdivision, ce qui permet de mettre à jour les valeurs de radiosités  $B_i^{orig}$  et les valeurs de facteurs d'affichage  $D_i^{orig}$  ( $D_i^{orig} = B_i^{orig}$ ), en fonction de la nouvelle subdivision.



FIG. 4.8: (a) L'élément encadré a une couleur différente par rapport à ses voisins, à cause de la couleur de l'affiche. En revanche, il n'y pas de différence de visibilité entre eux. L'élément n'est pas subdivisé. (b) Maillage résultant du raffinement basé sur la texture. Les régions d'ombre sont délimitées correctement.

### 4.3.2 Création de la texture représentant l'éclairage indépendant de la visibilité

Comme nous l'avons vu précédemment, nous désirons créer des nouvelles textures à partir des textures d'origine. Ces nouvelles textures doivent représenter un éclairage calculé sans occlusion. La création de nouvelles textures se fait en deux étapes successives.

#### Première étape pour la création des nouvelles textures

La première étape consiste à amasser l'énergie bloquée par des objets réels, et de l'ajouter aux textures d'origine  $T^{orig}$  qui correspondent à la photographie originale. Pour cela, nous utilisons le nouveau raffinement présenté ci-dessus, et nous définissons une nouvelle quantité d'énergie  $\bar{E}_i$ . Cette nouvelle quantité est l'irradiance d'un élément  $i$  depuis les sources  $s$ , bloquée par des objets réels. Une source  $s$  est soit une source de lumière, soit un autre élément de la hiérarchie (source d'éclairage secondaire). Cette irradiance additionnelle est la somme des radiosités  $E_{is}$  de chaque source, multipliée par les facteurs de forme  $F_{is}$  correspondants, ainsi que le complément du facteur d'atténuation  $(1 - V_{is})$ .

En considérant chaque source  $s$ , l'irradiance additionnelle  $\bar{E}_i$  pour chaque élément  $i$  est égale à :

$$\bar{E}_i = \sum_s F_{is}(1 - V_{is})E_{is}. \quad (4.5)$$

Le fait que tous les liens soient attachés aux feuilles assure un calcul plus juste de l'énergie additionnelle. Pour plus de précision, nous avons tenu compte de l'irradiance indirecte bloquée. Cependant, étant donné que la géométrie de la scène est simplifiée et non précise, nous ne considérons pas cette irradiance indirecte bloquée comme très fiable.

Pour créer les nouvelles textures en considérant la lumière bloquée, la texture originale est modulée par un facteur de correction calculé à chaque sommet des éléments feuille de la hiérarchie. La modulation de la texture aux sommets des éléments permet une modification lissée. Le facteur de correction utilise l'irradiance additionnelle décrite dans l'équation (4.5).

Pour inclure cette énergie bloquée dans les textures et créer les nouvelles textures  $T^{inter}$ , nous modulons la texture  $T_i^{orig}$  associée à chaque élément  $i$  par le facteur de correction suivant :

$$T_i^{inter} = \frac{\rho_i E_i + \rho_i \bar{E}_i}{D_i^{orig}} \times T^{orig}. \quad (4.6)$$

Dans cette équation,  $\bar{E}_i$  est l'irradiance bloquée par des objets réels, et la radiosité  $B_i^{orig} = D_i^{orig} = \rho_i E_i$ . La texture intermédiaire  $T_i^{orig}$  est générée en rendant chaque feuille de la hiérarchie avec une modulation appropriée comme la décrit l'équation (4.6). Si le facteur de modulation est supérieur à un, nous utilisons un affichage en passes multiples, décrit dans l'annexe A. Un exemple de modification de texture est montré sur la figure 4.9 (b), pour la texture du sol d'une de nos scènes. La reprojection des textures modifiées sur les polygones de la scène est montrée sur la figure 4.9 (d). Comme le montre cet exemple, la modification de texture est inexacte, ce qui produit des zones trop claires. Ceci est dû à l'estimation de l'énergie bloquée qui est inexacte, étant donné les approximations réalisées sur les paramètres de la radiosité, comme l'intensité des sources de lumière, les réflectances, les facteurs de forme et les facteurs d'atténuation, représentant les valeurs de visibilité.

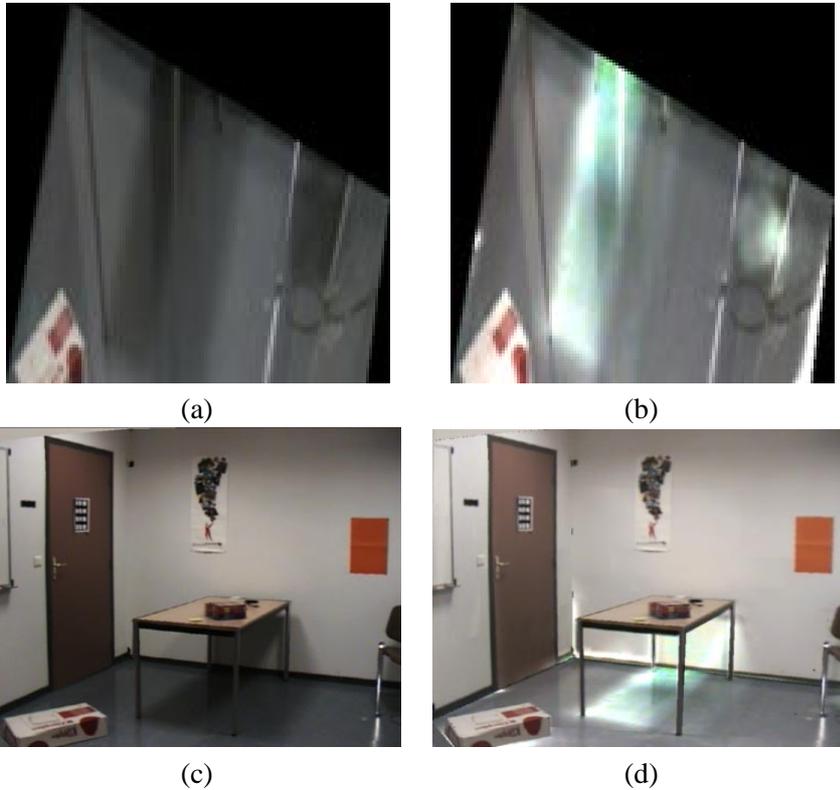


FIG. 4.9: (a) Texture d'origine pour le sol d'une de nos scènes. (b) Texture  $T^{inter}$  résultant de l'addition de l'énergie lumineuse bloquée par des objets réels. (c) Modèle 3D avec les textures d'origine reprojétées. (d) Modèle 3D avec les textures modifiées en ajoutant l'énergie bloquée.

La texture obtenue après cette étape de modification est utilisée pour mettre à jour des

valeurs de réflectance  $\rho_i^{inter}$ , et des valeurs de radiosité  $B_i^{inter}$ . Le facteur d'affichage est mis à jour en fonction de ce que représente la nouvelle texture, c'est-à-dire l'énergie totale, comprenant l'énergie bloquée par les objets réels ( $D_i^{inter} = \rho_i^{inter}(E_i^{inter} + \bar{E}_i^{inter})$ ).

Comme la figure 4.9 (d) le montre, nous ne pouvons pas utiliser les textures ainsi modifiées. Pour compenser les erreurs de modification, nous appliquons une correction, basée sur une heuristique prenant en compte la couleur des textures.

### Deuxième étape pour la création des nouvelles textures

Nous voulons maintenant corriger les textures nouvellement générées en ajoutant la lumière bloquée. L'idée générale de la correction est que l'on peut estimer la couleur désirée d'un élément dans l'ombre à partir de la couleur d'un élément complètement visible depuis les sources de lumière, appartenant au même objet. Nous aimerions donc que la couleur de texture d'un élément dans l'ombre soit similaire à celle d'un élément directement éclairé. La similarité correspond à une égalité de couleur, au rapport de facteurs de forme près. En effet, la texture que nous voulons obtenir doit représenter un éclairage direct complet (non bloqué), et non une réflectance indépendante de l'éclairage. La texture voulue  $T_i$  d'un élément  $i$  dans l'ombre doit être égale à la texture d'un élément voisin  $r$  directement éclairé, au rapport de facteurs de forme près :

$$T_i = T_r \times \frac{\sum_s F_{is}}{\sum_s F_{rs}}. \quad (4.7)$$

Comme nous l'avons fait précédemment, nous modifions la texture élément par élément, en la modulant. Pour corriger les textures, nous associons à chaque élément des régions d'ombre, un *élément de référence* qui servira à corriger la texture. Pour chaque élément du maillage dans l'ombre, nous cherchons donc un élément appartenant à une région éclairée. Cet élément doit être à la fois proche dans le sens de distance euclidienne, et d'une couleur similaire.

Pour cela, nous sélectionnons tout d'abord une liste d'éléments appartenant à une région éclairée, et voisins d'éléments appartenant aux régions d'ombre. Ces éléments forment une liste appelée *frontière*. Comme les liens depuis les sources de lumière sont attachés aux feuilles de la hiérarchie, la classification *VISIBLE*, *PARTIEL*, et *INVISIBLE* est relativement fiable. Grâce à cette classification, il est facile d'établir automatiquement la liste des éléments *frontière*, composée des éléments ayant tous les types des liens associés aux sources de lumière *VISIBLE*, et ayant un voisin au moins dont le type de lien associé à une source soit *PARTIEL* ou *INVISIBLE*. Cette frontière délimite la région où se trouvent les éléments à modifier lors de la correction des textures. Notre algorithme n'a cependant pas besoin que la région ainsi délimitée soit fermée. Nous appliquons une deuxième sélection sur ces éléments appartenant à la frontière, en ne conservant que ceux qui sont visibles depuis le point de vue. Cette restriction est justifiée par l'algorithme d'extraction de texture lors de la construction du modèle : il n'y a pas de calcul de visibilité réalisé, et la texture extraite pour un polygone contient les textures des éléments qui sont devant lui par rapport au point de vue. Nous avons affiché les éléments de la frontière ainsi sélectionnés

pour une subdivision grossière, pour le mur du fond, masqué par la table (voir figure 4.10 (a)).

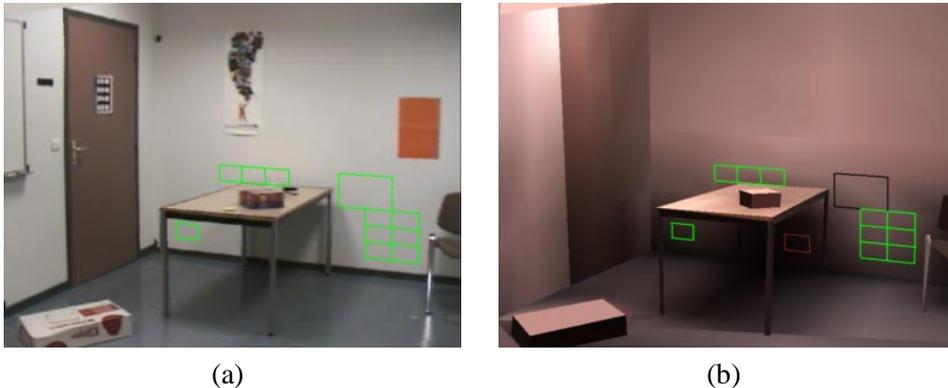


FIG. 4.10: (a) Frontière composée d'éléments appartenant à une région complètement visible depuis les sources de lumière, et voisins d'éléments de régions d'ombre. (b) L'élément de subdivision en noir de la frontière a été sélectionné comme élément de référence pour l'élément dans l'ombre de la table.

Pour chaque élément  $i$  dans l'ombre, un élément de référence  $r$  est choisi parmi les éléments de la frontière. L'élément choisi est l'élément le plus proche ayant une couleur similaire. Par exemple, pour l'élément dans l'ombre de la table sur la figure 4.10 (b), l'élément en noir a été choisi comme élément de référence parmi la liste d'éléments composant la frontière. Le choix de l'élément de référence se fait selon l'algorithme décrit à la figure 4.11.

```

Pour chaque feuille  $i$ 
   $min_{distance} = \infty$ 
   $min_{couleur} = \varepsilon$ 
   $Reference = i$ 
  Pour chaque élément  $n$  de la frontière
    si  $Distance(i, n) < min_{distance}$ 
      et  $Distance_{couleur}(n, i) < min_{couleur}$ 
      alors
         $Reference = i$ 
         $min_{distance} = Distance(i, n)$ 
         $min_{couleur} = Distance_{couleur}(i, n)$ 

```

FIG. 4.11: Algorithme de sélection d'un élément de référence.

Pour comparer les couleurs, nous utilisons les valeurs de réflectance, calculées directement depuis les textures et accessibles dans le système de radiativité. Comme nous l'avons fait précédemment lors du raffinement, nous comparons les couleurs dans l'espace CIE-LAB. Les couleurs entre deux éléments sont considérées comme similaires, si la distance euclidienne (équation (4.4)) est inférieure à un certain seuil  $\varepsilon$  fixé par l'utilisateur. Si aucun élément de référence n'est trouvé par l'algorithme, l'élément de référence affecté est

un élément par défaut choisi manuellement auparavant par l'utilisateur.

Une fois qu'un élément de référence est choisi pour chacun des éléments dans les régions d'ombre, nous utilisons l'équation (4.7) pour déterminer le facteur de correction à appliquer sur la texture. Au lieu d'utiliser les textures comme le suggère l'équation (4.7), nous utilisons les réflectances qui représentent à un facteur près la couleur des textures dans le système de radiosité. Comme l'élément de référence est à une certaine distance de l'élément dans l'ombre, la réflectance de l'élément de référence est modulée par le rapport des facteurs de forme des sources de lumière pour ces deux éléments,  $\sum_s F_{is}$  pour l'élément  $i$  dans l'ombre,  $\sum_s F_{rs}$  pour l'élément de référence  $r$ . En tenant compte de cette distance, la réflectance corrigée (que nous désirons obtenir) pour l'élément  $i$  est :

$$\rho_i^{corr} = \rho_r \frac{\sum_s F_{is}}{\sum_s F_{rs}}. \quad (4.8)$$

En utilisant la réflectance corrigée  $\rho_i^{corr}$ , les textures finales correspondant à un éclairage indépendant de la visibilité sont créées. Les feuilles de la hiérarchie sont affichées avec un facteur de correction appliqué aux sommets pour moduler la texture, comme pour la génération de la texture intermédiaire. Pour les éléments dans l'ombre seulement, la texture  $T_i^{inter}$  est modulée par l'équation suivante pour générer la texture finale  $T_i^{final}$  :

$$T_i^{final} = \frac{\rho_i^{corr}}{\rho_i^{inter}} \times T_i^{inter} = \frac{\rho_r}{\rho_i^{inter}} \frac{\sum_s F_{is}}{\sum_s F_{rs}} \times T_i^{inter}. \quad (4.9)$$

Le facteur  $\frac{1}{\rho_i^{inter}}$  est nécessaire car il permet d'atténuer l'effet contenu dans la texture  $T_i^{inter}$  et dû à la modification précédente. Si  $\rho_i^{corr}$  est supérieur à  $\rho_i^{inter}$ , un affichage à plusieurs passes est utilisé (voir annexe A), comme nous l'avons fait lors de la première modification des textures.

Le résultat de la texture corrigée pour le sol d'une de nos scènes est illustré sur la figure 4.12 (b). En (d), les textures ont été modifiées et reprojétées sur les polygones. Il est à noter que les ombres ont été relativement bien enlevées des textures, qui représentent, comme attendu, un éclairage indépendant de la visibilité.

À partir de ces nouvelles textures, correspondant à un éclairage indépendant de la visibilité, de nouvelles valeurs de réflectance  $\rho_i^{final}$  sont calculées pour chacun des éléments dans les régions d'ombre de la scène. Avec ces nouvelles réflectances, un calcul de radiosité est effectué en conservant les liens et la subdivision pour mettre à jour la radiosité  $B_i^{final}$ , et le facteur d'affichage  $D_i^{final} = \rho_i^{final} (E_i^{final} + \bar{E}_i^{final})$  (voir l'équation (4.5)). Le nouveau facteur d'affichage correspond à l'éclairage des nouvelles textures. Sur la figure 4.13, la radiosité  $B_i^{final}$  est montrée en (a), et le facteur d'affichage  $D_i^{final}$  est montré en (b).

### 4.3.3 Reprojection des ombres

Nous avons créé les textures indépendantes des valeurs de visibilité. Nous devons maintenant reprojeter les ombres afin d'avoir une représentation de l'éclairage initial de la scène.

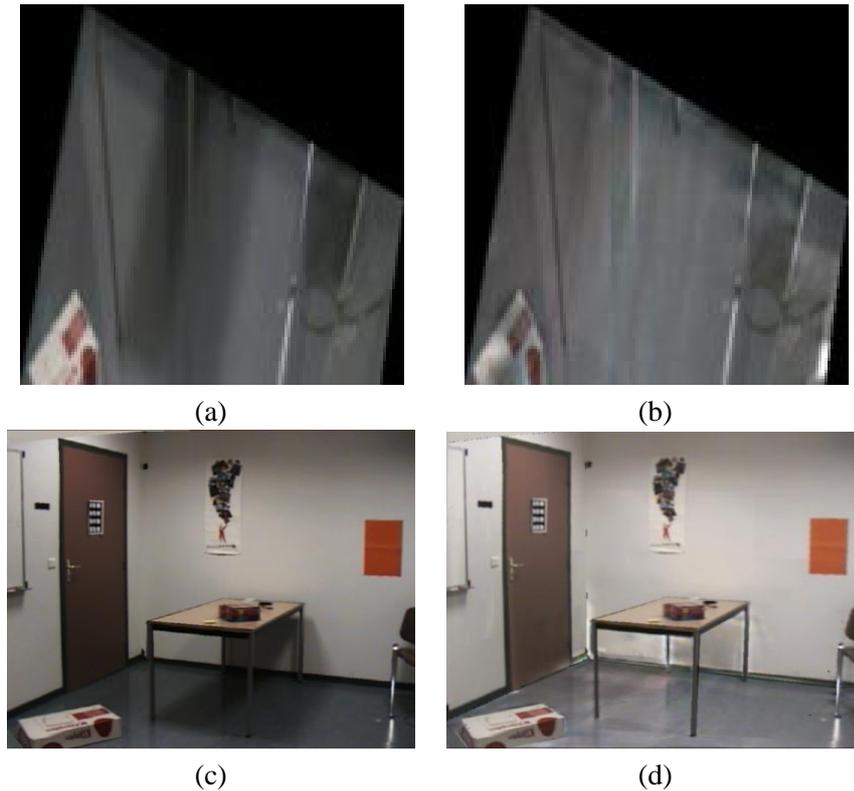


FIG. 4.12: (a) Texture d'origine pour le sol d'une de nos scènes. (b) Texture corrigée par les deux passes de l'algorithme de modification de textures. Comme attendu, cette texture correspond à un éclairage indépendant de la visibilité. (c) Textures d'origine projetées sur les polygones de la scène. (d) Textures finales corrigées  $T^{final}$ , projetées sur les polygones de la scène.

Nous utilisons pour cela l'équation d'affichage (4.3), avec les nouvelles valeurs de radiosité  $B_i^{final}$ , de facteur d'affichage  $D_i^{final}$ , et de texture  $T_i^{final}$ . Dans les régions d'ombre, la radiosité  $B_i^{final}$  est inférieure au facteur d'affichage  $D_i^{final}$ . Le rapport  $\frac{B_i^{final}}{D_i^{final}}$  est inférieur à 1, ce qui a pour effet d'assombrir la texture. La projection des ombres ainsi réalisée produit l'image 4.14 (b). Cet éclairage simulé est très similaire à l'éclairage contenu dans les textures d'origine en (a).

Il n'est souvent pas nécessaire de conserver la subdivision utilisée pour la création des nouvelles textures. Ce maillage est souvent trop fin dans les régions d'ombre, et pas suffisamment dans les régions directement éclairées. De plus nous avons attaché tous les liens aux feuilles de la hiérarchie ce qui peut être un handicap lors de la modification interactive de l'éclairage. Dans certains cas cependant, le maillage peut être conservé.

Dans la majorité des cas, nous finissons l'étape de pré-calcul par la génération d'une nouvelle solution de radiosité complète, utilisant un critère de raffinement BFV [HSA91]. Comme attendu, le maillage est moins fin dans les régions d'ombre, plus fin dans les régions éclairées, et il y a moins de liens actifs. La radiosité et le facteur d'affichage sont calculés en deux passes indépendantes.



FIG. 4.13: (a) Radiosité  $B^{final}$  correspondant à la nouvelle texture corrigée  $T^{final}$ . (b) Facteur d'affichage  $D^{final}$  correspondant à la nouvelle texture corrigée  $T^{final}$ .

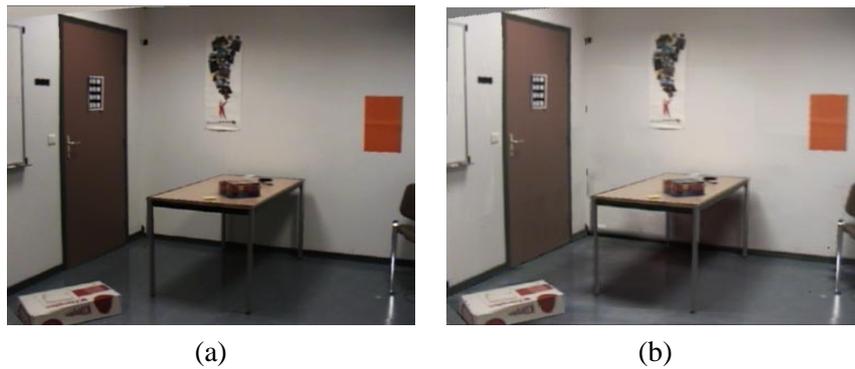


FIG. 4.14: (a) Scène originale vue dans notre système avec les textures d'origine. (b) Éclairage re-projeté sur les nouvelles textures  $T^{final}$ , correspondant à un éclairage indépendant de la visibilité. Les ombres sont projetées en utilisant le rapport d'affichage (équation (4.3)) appliqué sur les nouvelles valeurs de radiosité et de facteur d'affichage.

Dans un premier temps, tout le maillage et tous les liens sont enlevés. Les valeurs de réflectance, de radiosité et de facteur d'affichage sont initialisées à zéro. Une première solution de radiosité est ensuite calculée en utilisant le critère BFV. Cette solution produit un nouveau maillage, illustré sur la figure 4.15 x(a). Les valeurs de réflectance pour chacun des éléments de radiosité sont calculées durant la subdivision, et correspondent à la nouvelle texture  $T^{final}$ . Après avoir attaché tous les liens aux feuilles de la hiérarchie, nous calculons un nouveau facteur d'affichage, correspondant comme précédemment à une valeur de radiosité totale, c'est-à-dire sans prise en compte de la visibilité.

Ensuite, nous conservons le maillage, les valeurs de réflectance associées, et le facteur d'affichage nouvellement calculé pour chacun des éléments. Les liens sont détruits, et les valeurs de radiosité annulées. Une nouvelle solution de radiosité est calculée, en utilisant les mêmes critères de raffinement que ceux de la précédente. De nouveaux liens hiérarchiques sont également créés. La reprojection des ombres, en utilisant la radiosité et le facteur d'affichage ainsi calculés, est montrée sur la figure 4.15 (b).

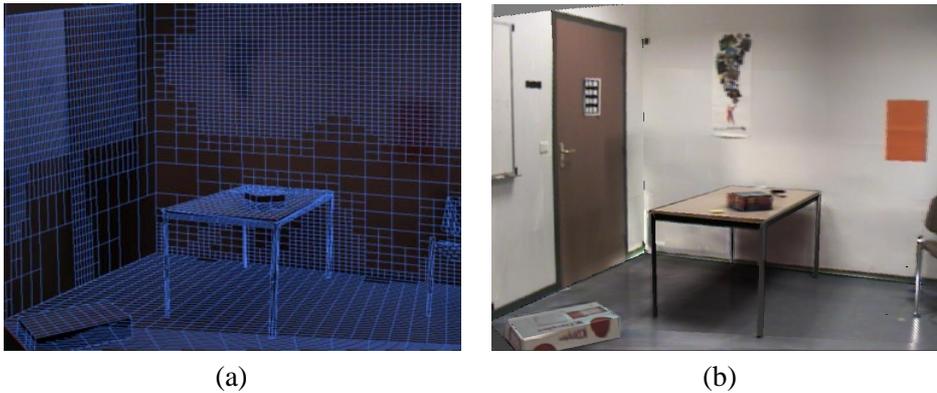


FIG. 4.15: (a) Après la modification des textures, une nouvelle solution de radiosité est calculée, en utilisant un critère de raffinement BFV. Le maillage résultant est moins fin dans les régions d'ombre que celui utilisé pour la modification des textures. (b) Reprojection des ombres sur les nouvelles textures, en utilisant la nouvelle solution de radiosité basée sur le critère de raffinement BFV.

## 4.4 Éclairage commun et ré-éclairage interactif

Après avoir modifié les textures originales pour créer de nouvelles textures sans ombres, et après avoir ré-initialisé la solution de radiosité à partir de ces nouvelles textures, nous pouvons modifier interactivement l'intensité des sources de lumière réelles, et ajouter des objets et des lampes virtuelles. Nous expliquons dans cette section quels algorithmes nous utilisons pour réaliser ces modifications interactivement. Les résultats sont présentés ensuite, dans la section 4.5.

### 4.4.1 Insertion d'objets et de lampes virtuels

Pour insérer des objets virtuels et les déplacer dans la scène, nous utilisons l'algorithme de radiosité hiérarchique dynamique de Drettakis et Sillion [DS97], déjà utilisé dans le cadre de la réalité augmentée par Drettakis *et al.* [DRB97]. Nous avons adapté cet algorithme pour l'insertion des lampes virtuelles. L'influence des lampes virtuelles est très importante. C'est pourquoi nous avons forcé le raffinement en créant directement des liens entre la source de lumière et les polygones visibles depuis le point de vue de la scène, au lieu d'avoir ces liens au niveau des clusters. Ensuite, nous utilisons l'algorithme de radiosité hiérarchique dynamique, en autorisant la création d'autant de liens que le demande le raffinement. L'insertion d'une lampe virtuelle s'accompagne d'un éclairage général de la scène. Notre rendu est réalisé par le matériel d'accélération graphique, en affichant les polygones subdivisés et texturés de la scène, plutôt que pixel par pixel [FGR93]. Pour éclaircir les textures par le facteur d'affichage, nous utilisons un rendu en multi-passes, comme nous l'expliquons en annexe A.

#### 4.4.2 Ré-éclairage interactif

Pour réaliser un ré-éclairage interactif de la scène, nous conservons le maillage et les liens établis dans la solution de radiosité calculée précédemment. Lors de la modification de l'intensité d'une source de lumière, la radiosité est mise récursivement à zéro, puis une nouvelle solution de radiosité est calculée itérativement en conservant la subdivision et les liens. La radiosité est accumulée aux feuilles en utilisant les liens (*gather*), et la solution est rendue cohérente par un *push-pull*. La solution itérative est arrêtée lorsque l'éclairage global est stable.

Comme nous ne modifions que l'intensité des sources de lumière, la subdivision et les liens sont toujours cohérents, même après la modification. Dans certains cas, conserver la même hiérarchie peut amener à une subdivision trop fine, lorsque les lampes sont éteintes. Comme l'utilisateur peut vouloir rallumer ces lampes, nous préférons conserver la subdivision.

Raffinement			
Diminution de l'intensité de la lampe			
Temps de modification	0.2 sec.	0.3 sec.	0.7 sec.
Temps d'affichage	0.2 sec.	0.2 sec.	0.6 sec.
Nombre de feuilles/liens	3486/11246	5781/16807	8247/50787

FIG. 4.16: Modification interactive de l'intensité d'une lampe virtuelle. Le temps de mise à jour dépend du niveau de subdivision et du nombre de liens actifs établis.

Cet algorithme est interactif car nous évitons de calculer un nouveau raffinement, et en particulier de nouvelles valeurs de visibilité et de facteurs de forme, ce qui serait coûteux (en temps de calcul). Le temps de mise à jour dépend du niveau de subdivision et du nombre de liens. Il ne varie pas beaucoup si on modifie plusieurs intensités de lampe en même temps. Des exemples de temps de calcul sont montrés sur la figure 4.16. Comme prévu, ces temps augmentent en fonction du nombre de feuilles de la hiérarchie des éléments, ainsi que du nombre de liens. Dans ce tableau, nous mettons en évidence le coût (en temps passé) de l'affichage réalisé en plusieurs passes pour permettre l'éclaircissement des textures. Ce coût dépend du nombre d'éléments à afficher.

Cependant, la qualité de la solution varie peu lorsqu'on augmente la subdivision. L'ef-

fet général est le même pour les trois exemples. De plus, les temps restent interactifs, même dans le cas d'une forte subdivision.

## 4.5 Résultats

Nous avons testé nos algorithmes sur plusieurs scènes. Pour chaque scène, nous présentons des résultats interactifs de ré-éclairage et d'insertion de lampes et d'objets virtuels. Les temps sont calculés sur une station de travail *SGI Onyx2 Infinite Reality*, avec un processeur R10000, à 195 Mhz<sup>‡</sup>.

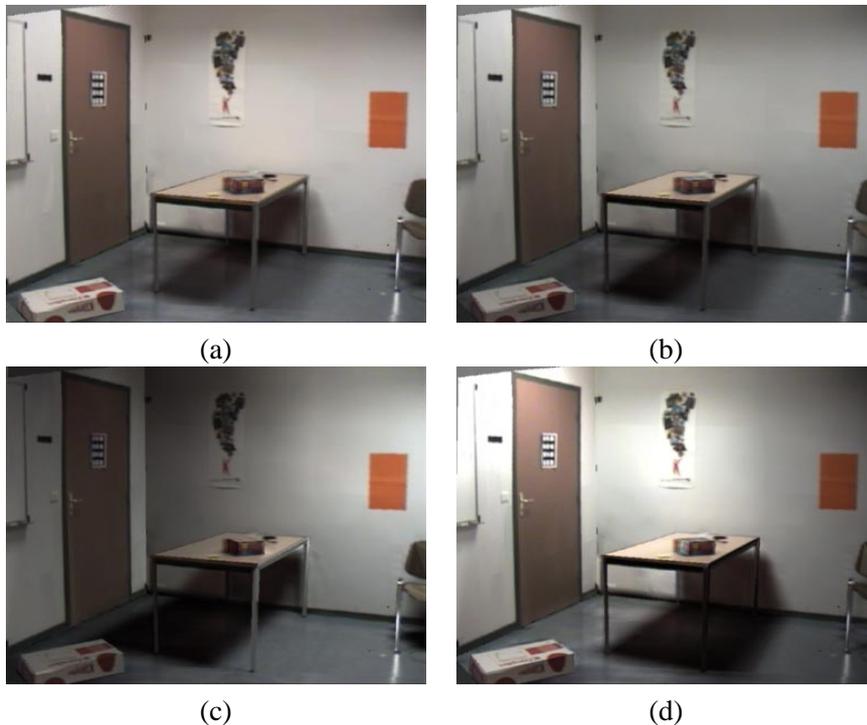


FIG. 4.17: (a) Scène avec son éclairage original reprojété sur les textures. (b) Les deux lampes du fond ont été éteintes. (c) La lampe de gauche a été éteinte, et l'intensité de la lampe de droite doublée. (d) La lampe de droite a été éteinte, et l'intensité de la lampe de gauche doublée.

La première scène testée est présentée sur la figure 4.17 (a), sous son éclairage d'origine. Nous éteignons d'abord les deux lampes du fond (C et D) (voir figure 4.3). Comme le montre le résultat en (b), la scène est globalement plus sombre, mais il n'y a pas de modification perceptible dans les ombres. Nous éteignons ensuite la lampe A à gauche, et nous doublons l'intensité de la lampe B à droite. Le résultat est montré en (c). L'ombre de la table due à la lampe de gauche a disparu, et la partie de la scène éclairée par cette lampe est plus sombre. Le résultat est meilleur que celui obtenu sans modification de texture en

<sup>‡</sup>Les résultats sont également montrés en vidéo sur internet :  
<http://www-imagis.imag.fr/Membres/Celine.Loscos/CAR/car.html>.

utilisant les algorithmes de Fournier *et al.* [FGR93] ou de Drettakis *et al.* [DRB97] (voir figure 4.5). Nous ré-éclairons ensuite la lampe de gauche, en doublant son intensité initiale et nous éteignons la lampe de droite (voir figure 4.17 (d)). Comme attendu, l'ombre de la table due à la lampe de droite a disparu et la partie de la scène éclairée par cette lampe s'est assombrie. Avec la subdivision utilisée pour cet exemple, le temps total de mise à jour pour chaque modification est de 0.8 secondes, temps d'affichage compris.

Une source virtuelle peut être insérée dans cette scène en utilisant l'algorithme présenté dans la section 4.4.1. L'insertion de la lampe virtuelle prend 7.8 secondes. Dans la scène montrée sur la figure 4.18 (a), toutes les lampes réelles sont éteintes laissant la scène éclairée uniquement par la lampe virtuelle. Les ombres dues aux lampes réelles ont disparu. L'éclairage est cohérent ; par exemple, les ombres de la table dues à la lampe virtuelle sont projetées sur le sol et sur les murs. Nous pouvons également modifier l'intensité de la lampe virtuelle, comme nous le montrons sur la figure 4.18 (b), en utilisant le même algorithme que pour les lampes réelles (voir section 4.4.2).



FIG. 4.18: (a) Insertion d'une lampe virtuelle. Toutes les lampes réelles ont été éteintes. La scène n'est éclairée que par la source de lumière virtuelle. (b) L'intensité de la lampe virtuelle est diminuée.

Avec ce nouvel éclairage, un objet virtuel est inséré en 1.42 secondes dans la scène et peut être déplacé interactivement. Dans la figure 4.19 (a), on voit la subdivision relative à l'insertion de la lampe virtuelle et de l'objet virtuel. En (b), (c), et (d), l'objet se déplace à raison de 3 images par secondes (temps d'affichage compris). Les ombres de l'objet dynamique, dues à la fois à la lampe virtuelle et aux lampes réelles, sont mises à jour en utilisant l'algorithme de Drettakis et Sillion [DS97]. On peut également remarquer dans ces images que l'effet spéculaire du pied de la chaise sur le sol est conservé car il est contenu dans les textures d'origine. Même si l'effet spéculaire n'est parfois plus cohérent avec l'éclairage, il permet de maintenir un effet visuel réaliste dont l'œil humain ne perçoit pas l'inexactitude. De plus un tel effet serait difficile à simuler avec des algorithmes de radiosité.

Nous avons également testé notre méthode sur une autre scène, reconstruite avec l'outil *Rekon* [POF98]. Cette scène est montrée sur la figure 4.20 (a) où les ombres réelles ont été reprojétées sur les textures modifiées, montrées en (b). Comme pour la scène précédente, les ombres ont été correctement enlevées des textures qui représentent un éclairage indé-

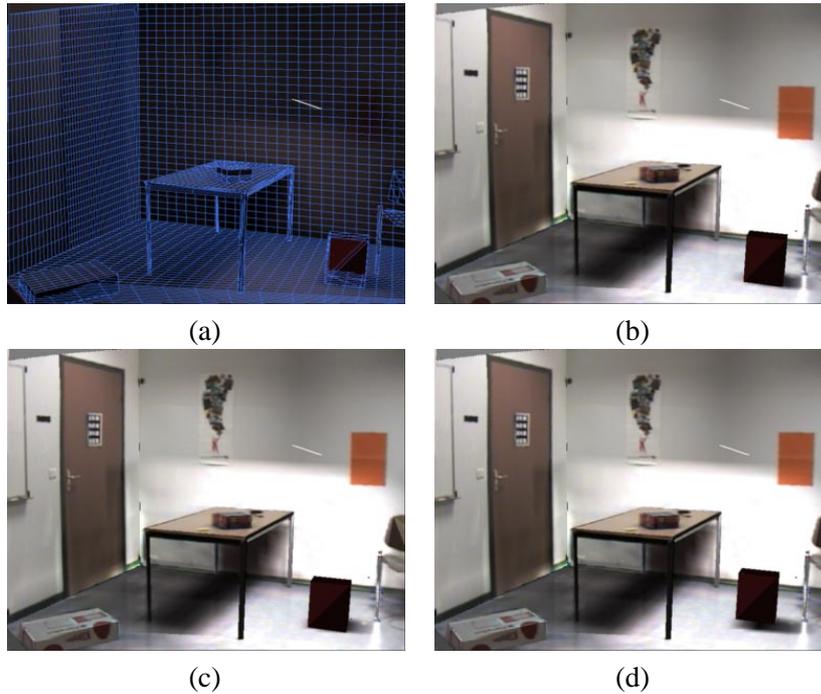


FIG. 4.19: (a) Insertion d'un objet virtuel en 1.42 secondes, et la subdivision conséquente. (b), (c), (d) L'objet virtuel (boîte) est déplacé interactivement en 3 images par seconde.

pendant de la visibilité.



FIG. 4.20: (a) Deuxième scène testée. Les ombres ont été reprojétées en utilisant notre système de radiosité. (b) Textures modifiées, représentant un éclairage indépendant de la visibilité.

Nous avons modifié l'intensité des lampes réelles de cette scène, et comparé le résultat avec des photographies prises dans les mêmes conditions d'éclairage. Dans la figure 4.21, la première colonne de gauche correspondant aux images (a), (b), (c), et (d), sont des photographies. La colonne du milieu (images (e), (f), (g) et (h)) correspond aux résultats simulant les mêmes conditions d'éclairage par notre système. L'effet général est similaire, mais les couleurs sont légèrement différentes. Dans la colonne de droite (images (i), (j) et (k)), nous avons corrigé les images en appliquant une correction gamma [War94a]. Nous aurions également pu appliquer un « Tone Mapping » [TR93]. Ces transformations perme-

ttent de transformer l'image pour faire ressortir les contrastes comme le ferait l'œil. Les résultats sont très similaires aux photographies originales. Nous avons d'abord éteint les lampes du fond (figures (a), (e) et (i)), puis nous avons éteint la lampe de gauche, laissant la scène éclairée uniquement par la lampe de droite (figures (b), (f), et (j)). Ensuite, nous avons éteint la lampe de droite et rallumé celle de gauche (figures (c), (g), (k)).

Sur la dernière ligne, nous avons comparé l'insertion d'une lampe virtuelle dans la scène avec toutes les autres lampes éteintes (en (h)). La lampe virtuelle a été placée au même endroit qu'une lampe utilisée pour les photographies (en (d)).

Nous pensons que la différence entre l'image simulée et l'image photographiée est due à une erreur dans l'estimation de l'intensité des sources. Deux raisons justifient cette observation. La première est qu'un simple gamma suffit pour rendre l'image photographiée et l'image simulée similaires (voir figure 4.21). La deuxième est que nous obtenons un très bon résultat pour la lampe virtuelle, pour laquelle l'intensité a été fixée à la main.

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté une solution pour résoudre le problème de ré-éclairage de scènes réelles. Dans notre approche, nous utilisons un modèle texturé pour représenter la scène réelle. Les textures sont extraites depuis une photographie de la scène prise sous des conditions d'éclairage fixes. La résolution du problème de ré-éclairage est difficile car l'éclairage réel, y compris les ombres, est déjà inclus dans les textures. Ces effets d'éclairage doivent être correctement identifiés avant d'être modifiés.

Nous avons présenté une solution permettant une modification interactive de l'éclairage, pour des sources de lumières réelles et virtuelles. L'algorithme est composé de trois parties. Dans la première, un modèle 3D texturé de la scène réelle est reconstruit, et le système de radiosité initialisé avec les algorithmes développés par Fournier *et al.* [FGR93] et Drettakis *et al.* [DRB97]. Dans une deuxième phase de pré-calcul, nous préparons la scène pour que l'intensité des lampes réelles puisse être modifiée. Dans un premier temps, un raffinement basé sur la texture délimite finement les zones d'ombre. Les textures sont ensuite modifiées en incluant la lumière bloquée par des objets réels, puis en se basant sur une couleur de texture de référence. Les textures ainsi obtenues représentent une solution de radiosité calculée sans occlusion. À la fin de cette deuxième partie de l'algorithme, un nouveau raffinement utilisant un critère standard peut-être effectué, et les ombres sont re-projetées sur les textures, par un affichage utilisant le matériel d'accélération graphique des stations de travail. Les images simulées à la fin de cette étape sont très similaires aux photographies originales.

Ensuite, des algorithmes de mise à jour interactive sont appliqués pour modifier l'éclairage de la scène en changeant l'intensité des lampes réelles, et en insérant des objets et des lampes virtuelles. Les résultats sont convaincants, et la comparaison avec des photographies valide nos résultats.

Cependant, notre méthode présente certaines limitations que nous aimerions améliorer.

Dans un premier temps, l'affichage en multi-passes limite les temps d'interactivité. Une optimisation de l'affichage général de notre système devrait améliorer ce problème.

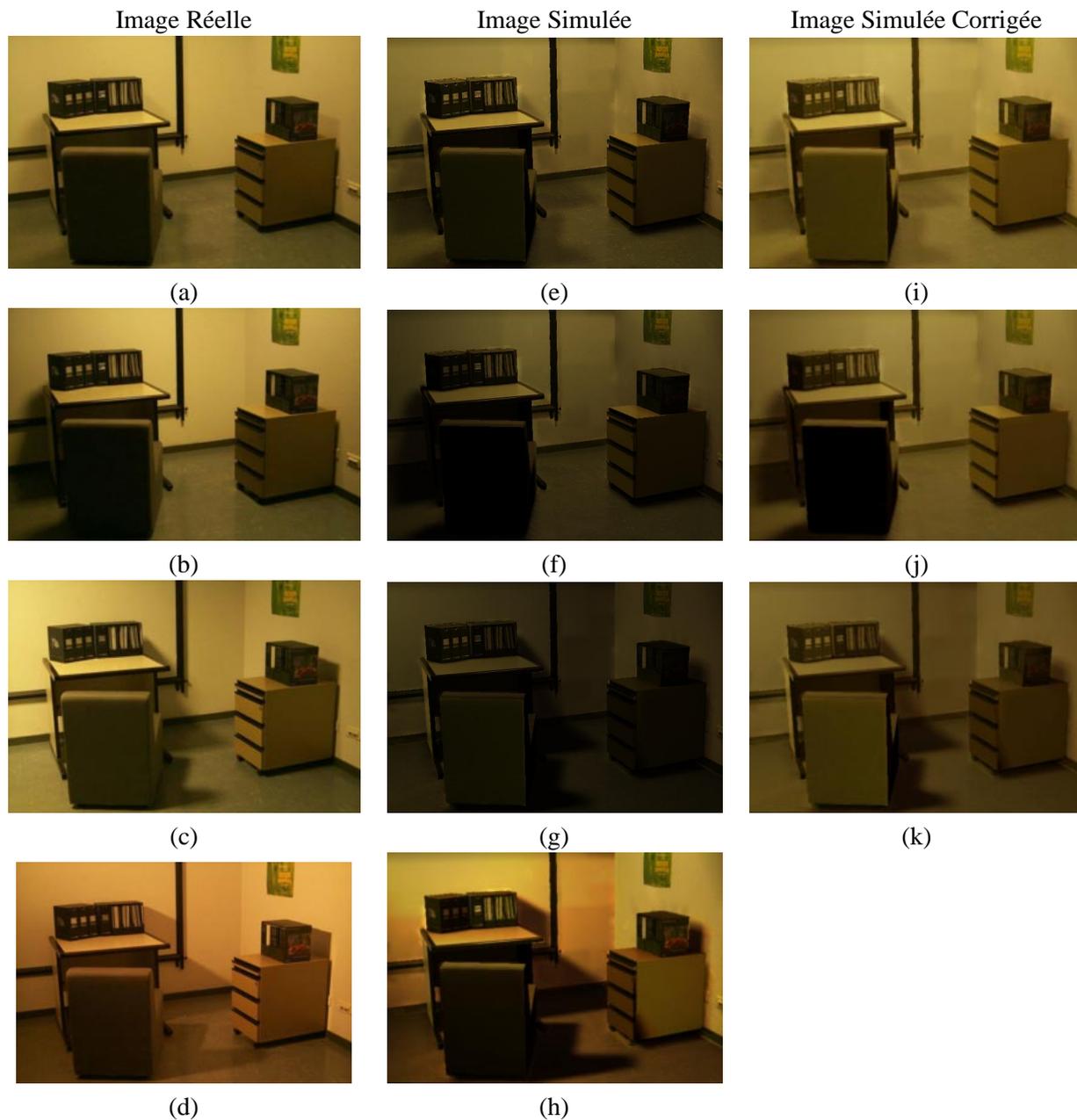


FIG. 4.21: (a), (b), (c), (d) sont des photographies prises sous différentes conditions d'éclairage. (e), (f), (g) et (h) sont des images simulées selon les mêmes conditions d'éclairage. (i), (j), (k) et (l) sont les mêmes images que dans la colonne du centre, mais sur lesquelles nous avons appliqué une correction gamma.

De plus, les algorithmes utilisés pour mettre à jour la radiosité lors de la modification de l'intensité des lampes, pourraient être optimisés en utilisant des techniques dans l'esprit de celles de Drettakis et Sillion [DS97].

Une limitation importante de notre système est l'obligation de visualiser la scène depuis un point de vue fixe. Un utilisateur apprécierait de pouvoir se déplacer dans la scène pour en évaluer l'éclairage. Une solution pourrait être développée en utilisant les techniques de texture indépendantes du point de vue [POF98], ou de mise à jour interactive des textures [DBY98].

Enfin, la qualité des textures obtenues par notre méthode est limitée par la connaissance que nous avons de la scène. Travailler depuis une seule photographie et permettre quand même le ré-éclairage s'est avéré très difficile. Les difficultés rencontrées se situent surtout pour les régions d'ombre. Dans ces régions, la qualité de la simulation de l'éclairage est très importante pour retrouver une couleur de texture correspondant à un éclairage indépendant de la visibilité. De plus, la qualité des couleurs dans ces régions est limitée. La phase de correction basée sur la couleur d'un élément totalement éclairé nous a paru indispensable. En travaillant pixel par pixel, plutôt que sur les éléments de radiosité, les manipulations de textures pourraient être plus fines, et les valeurs de réflectance mieux estimées. Ceci permettrait notamment d'enlever des objets réels.

Dans le chapitre suivant, nous tenons compte des observations faites sur cette solution, pour en proposer une nouvelle, pour laquelle les données dans les régions d'ombre seront plus précises.