Acquisition de l'environnement pour le ré-éclairage et le positionnement d'objets virtuels dans une scène réelle

François Fouquet, Jean-Philippe Farrugia, Brice Michoud, Sylvain Brandel

Revue Électronique Francophone d'Informatique Graphique, Volume 4, Numéro 1, pp. 1–12, 2010

Acquisition de l'environnement pour le ré-éclairage et le positionnement d'objets virtuels dans une scène réelle

François Fouquet, Jean-Philippe Farrugia, Brice Michoud, Sylvain Brandel

Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
{francois.fouquet, jean-philippe.farrugia, brice.michoud, sylvain.brandel}@liris.cnrs.fr

Résumé

L'objectif de la réalité augmentée est d'insérer des objets virtuels dans une scène réelle. Afin d'obtenir une intégration réaliste et cohérente, il est nécessaire de ré-éclairer ces objets en fonction de leur position et des conditions d'éclairage réelles, et de tenir compte des occultations provoquées par des objets réels plus proches que les objets virtuels. Dans cet article, nous adaptons les méthodes de reconstructions d'images HDR et d'estimations de la profondeur pour un contexte temps réel. Nous présentons leurs limitations ainsi que les paramètres qui influencent les temps de calcul et la qualité des images de manière significative. Nous montrons comment modifier ces paramètres pour accélérer les calculs et évaluer les impacts sur la qualité du résultat. Enfin, pour notre approche de la réalité augmentée, nous proposons une extraction temps réel de ces informations d'un flux vidéo, en un seul traitement.

Mots clé : Imagerie HDR, Vision par ordinateur, Carte de profondeur, Réalité augmentée, Rendu temps réel

1. Introduction

La réalité augmentée est un ensemble de techniques issues de l'imagerie qui consiste à mêler des éléments synthétiques et des éléments réels dans une scène cohérente. Généralement, l'utilisateur peut se déplacer librement dans un environnement réel pour lequel il lui est proposé une visualisation augmentée d'objets synthétiques. Les applications de cette technique sont multiples allant des loisirs numériques (jeux vidéo) à la simulation de situations d'urgence pour la formation des personnels dans les industries à risques (nucléaire, médical...). Très souvent la mise en place de la réalité augmentée nécessite du matériel spécialisé voire des lieux aménagés exclusivement pour cela (salle immersive). Dans le cadre de notre projet, un des objectifs est d'ouvrir au plus grand nombre l'accès à ces techniques. Nous avons donc pris le parti de n'utiliser que du matériel non spécifique et déjà très répandu (webcam, GPU, vidéo-projecteurs...).

Augmenter la réalité signifie mettre en correspondance la représentation que l'on a de l'environnement et celle de l'objet synthétique qu'on souhaite y insérer. En fonction de l'application visée, cette fusion des représentations sera plus ou moins précise ou complète. Par exemple, pour une application d'assistance à la réparation de photocopieur, il faut placer les éléments virtuels très précisément (ie., alignés avec les différentes pièces mécaniques de la machine). En revanche, le ré-éclairage et la projection des ombres des éléments ne sont pas cruciaux pour le bon fonctionnement de l'application. A contrario, la reconstruction d'un monument historique détruit a besoin d'être visuellement convaincante. Dans ce cas, le placement peut être moins précis sans que cela dérange l'utilisateur. En revanche, une erreur d'occultation ou d'éclairage affectera grandement le réalisme.

Pour obtenir une réalité augmentée réaliste, nous nous plaçons dans ce deuxième cas. Actuellement, nos travaux se focalisent sur l'acquisition de deux types d'informations sur l'environnement nécessaires pour effectuer un ré-éclairage simple : la géométrie et l'éclairage. Un grand nombre de méthodes ont été proposées pour acquérir ces informations à partir d'images d'un ou plusieurs points de vue mais peu d'entre elles sont adaptées à la réalité augmentée et à une contrainte de faible coût matériel.

Dans cet article, nous proposons de sélectionner deux mé-

thodes, une pour la photométrie et l'autre pour la géométrie, et de tester l'influence de certains paramètres d'entrée sur le temps de calcul. Nous utilisons pour cela les mêmes données en entrée et leur appliquons simultanément les deux techniques en un unique processus. L'objectif final est d'obtenir un bon compromis entre temps de calcul et qualité afin d'appliquer ces méthodes en temps réel.

La suite de cet article comporte trois sections. La première présentera les travaux existants et les méthodes que nous avons développées pour accélérer l'acquisition de l'environnement photométrique à partir de jeux d'images et de flux vidéo. La section suivante présentera, de manière similaire, l'acquisition de la géométrie. La troisième section présente un exemple d'incrustation d'un objet virtuel dans une scène réelle. Enfin, nous conclurons en synthétisant les résultats importants puis nous présenterons certaines perspectives de ce travail.

2. Acquisition de la photométrie

2.1. Travaux existants

Les images obtenues avec un capteur classique (appareil photo ou caméra) fournissent généralement des informations suivant trois canaux (rouge, vert et bleu) qui correspondent d'une part à la quantité de lumière renvoyée au capteur par la zone de l'environnement située dans son champ de vision et d'autre part à la couleur de ces zones. Cependant, ces appareils ne permettent pas d'obtenir une image contenant toute la dynamique lumineuse de la scène en une seule prise de vue. Si la scène que l'on observe contient des zones de luminance très fortes (sources lumineuses) ou au contraire très faibles (ombres), les images obtenues présenteront des zones sur-exposées ou sous-exposées pour lesquelles la quantité de lumière réellement présente dans la scène n'est pas déterminable. Par ailleurs, les capteurs CCD ont généralement des fonctions de réponse non linéaires et la quantité de lumière qu'ils reçoivent va elle-même varier en fonction des paramètres de la caméra (temps d'exposition, ouverture...). Déterminer la luminance arrivant réellement au capteur à partir d'une image peut donc s'avérer complexe.

Pour lever ce problème nous utilisons des images HDR (*High Dynamic Range*) qui permettent de stocker l'ensemble des détails de luminance d'un environnement quelle que soit l'intensité de son éclairage. Ces images nécessitent généralement une opération de réduction de dynamique appelée reproduction de tons (*tone mapping*) pour être visualisées sur un écran. Il existe différentes techniques pour reconstruire les images HDR, reposant toutes sur une même idée introduite par Mann et Picard [MP95] d'une part et Madden [Mad93] d'autre part. Le principe est de combiner plusieurs images prises d'un même point de vue mais avec des expositions différentes afin de reconstruire une image contenant tous les détails de luminance des différentes expositions. Debevec et Malik [DM97] proposent d'utiliser des images pour lesquelles ils font uniquement varier le temps d'exposition. Ils déterminent dans un premier temps la fonction de réponse f de l'appareil puis ils reconstruisent ensuite l'image HDR en faisant une moyenne pondérée des images d'entrée transformées par f^{-1}. Mitsunaga et Nayar ont proposé une méthode similaire [MN99] mais faisant en plus l'hypothèse que f est un polynôme. Ils réalisent ensuite une recherche itérative de l'ordre et des coefficients du polynôme correspondant le mieux aux données observées. En plus de déterminer f, cette deuxième méthode présente l'avantage de permettre de déterminer des ratios d'exposition entre les différentes images de départ. Les expositions de ces images peuvent donc être inconnues, contrairement à la méthode de Debevec et Malik. La plupart des techniques actuelles d'imagerie HDR sont rassemblées dans l'ouvrage *"High dynamic range imaging"* [RWPD05]. L'acquisition d'images HDR proposée par Debevec et Malik reste la référence du domaine, et est à la base de nombreux travaux, citons par exemple la construction de vidéos restituant l'ensemble de la dynamique HDR à partir d'un simple flux vidéo en entrée [KUWS03].

2.2. Reconstruction d'images HDR

Dans le cadre de notre étude, orientée vers la réalité augmentée, les données d'entrée sont des flux vidéo, avec éventuellement des variations de temps d'exposition d'une image à l'autre. Nous avons donc choisi d'utiliser la méthode proposée par Debevec et Malik [DM97], qui utilise des images prises depuis un même point de vue, avec des temps d'exposition différents. Elle propose une méthode pour déterminer la fonction de réponse propre à la caméra, puis une méthode pour combiner les images en entrée en utilisant cette fonction. Il est ainsi possible de déterminer tous les niveaux de luminance contenus dans le jeu d'images. Les images prises avec un temps d'exposition court (1/1000s. par exemple) contiennent de l'information sur les zones de forte énergie de la scène, alors que les images prises avec un temps d'exposition long procurent de l'information sur les zones de faible énergie. Pour déterminer la fonction de réponse de la caméra, les auteurs sélectionnent manuellement des points de l'image. Pour chacun de ces points, ils enregistrent sur chaque image la valeur de luminance et le temps d'exposition correspondant. Toutes ces observations forment un échantillon de valeurs de la fonction de réponse de la caméra dépendant du temps d'exposition. Cet échantillon est ensuite mis sous la forme d'un système d'équations fortement sur-déterminé. Une décomposition en valeurs singulières permet ensuite d'estimer la fonction de réponse de la caméra qui correspond au mieux au jeu de données. Pour que cette estimation soit utilisable, il faut que la scène observée contiennent initialement une dynamique élevée (typiquement on voit à la fois une source de lumière et une zone d'ombre) et que les localisations de point choisies sur l'image rendent compte de cette grande dynamique. En effet, si le jeu de points échantillonne des zones de luminances

proches, la fonction est alors estimée très précisément dans ce domaine de luminance, mais devient erronée dès qu'on s'en éloigne. Pour la construction des images HDR proprement dite, les auteurs appliquent l'inverse de la fonction de réponse à toutes les images du flux d'entrée. L'image finale est obtenue en calculant la moyenne pondérée de ces images transformées.

La valeur de luminance Z_{ij} de chaque pixel i d'une image ayant un temps d'exposition T_j est obtenue à partir de l'énergie lumineuse E_i arrivant à cet endroit du capteur et étant transformée par la fonction de réponse f du capteur : $Z_{ij} = f(E_i T_j)$. Pour construire l'image HDR, nous cherchons à déterminer l'irradiance E_i au pixel i :

$$E_i = \frac{f^{-1}(Z_{ij})}{T_j} \quad (1)$$

La fonction de réponse f doit être déterminée en même temps que la valeur E_i en chaque point de l'image. En vue de la reconstruction de l'image HDR, nous recherchons à déterminer la fonction g, logarithme népérien de la fonction inverse de f. g est une fonction discrète définie sur l'intervalle [0-255], il faut donc déterminer les valeurs de $g(0)$ à $g(255)$ pour déterminer intégralement g :

$$g(Z_{ij}) = \ln(f^{-1}(Z_{ij})) = \ln(E_i) + \ln(T_j) \quad (2)$$

La fonction g étant spécifique à chaque capteur, pour retrouver la fonction g de la caméra que nous utilisons ainsi que les E_i, nous cherchons $g(Z_{ij})$ tel que l'équation 2 soit vérifiée pour tous les points de toutes les images d'entrée. Ceci nous donne un système à $P \times N_t$ équations, P étant nombre d'images en entrée et N_t le nombre total de points sur chaque image :

$$\sum_{i=1}^{N_t} \sum_{j=1}^{P} [g(Z_{ij}) - \ln(E_i) - \ln(T_j)] = 0 \quad (3)$$

Ce système d'équations étant largement surdéterminé, seule une résolution approchée est envisageable. Debevec et Malik ont décidé de ne pas utiliser tous les points de l'image pour le calcul de g et des E_i mais d'en sélectionner à la main quelques uns qu'ils jugent intéressants. Ils ont également décidé de contraindre la fonction g à être *au mieux* continue et monotone en forçant $g''(z) = g(z-1) - 2g(z) + g(z+1) = 0$, z appartenant à l'intervalle [1-254]. Ils ajoutent enfin une fonction de pondération $w(z)$ afin de donner une importance plus faible aux z situés aux limites de la gamme de luminance (susceptibles d'être bruités). Cette fonction vaut $w(z) = z$ pour z appartenant à l'intervalle [0-127] et $w(z) = 256 - z$ pour z appartenant à [128-255]. En prenant en compte toutes ces modifications, il reste finalement à minimiser le système suivant, N étant le nombre de points pris sur les photographies pour l'estimation de la fonction de réponse et z les valeurs de luminance des images d'entrée (niveaux de gris entre $Z_{min} = 0$ et $Z_{max} = 255$) :

$$\theta = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij})[g(Z_{ij}) - \ln(E_i) - \ln(T_j)]\}^2 + \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z)g''(z)]^2 \quad (4)$$

Il reste alors à reconstruire l'image des radiances pour chaque pixel, à partir de g :

$$\ln(E_i) = \frac{\sum_{j=1}^{P} w(Z_{ij})[g(Z_{ij}) - \ln(T_j)]}{\sum_{j=1}^{P} w(Z_{ij})} \quad (5)$$

Avec notre implémentation (Quadcore 2.5GHz avec 2 Go de RAM), cette méthode construit une image HDR (640×480) en 4,4s., à partir de 12 images dont les temps d'exposition varient de 1/1000s. à 2s. (figure 1). Ce temps de calcul, qui n'inclut pas le temps de prise des photos, est divisé en deux parties : 3,2s. pour calculer le temps de réponse de la caméra et les 1,2s. restantes pour calculer l'image HDR.

Comme nous l'avons évoqué précédemment, le choix des points et de la scène utilisé pour le calcul de la fonction de réponse de la caméra est primordial pour garantir de bons résultats. Dans la méthode originale, la sélection est faite par un tirage pseudo-aléatoire de points dans le plan de l'image, sans implémenter aucune stratégie de sous-échantillonnage spatial. Nous avons donc ajouté une sélection automatique des points afin de couvrir au mieux toute la dynamique présente dans la scène. L'idée utilisée pour cette sélection est qu'un point de faible luminance dans la scène sera observé avec des faibles valeurs quel que soit le temps d'exposition de l'image, alors qu'un point de forte luminance aura des valeurs élevées sur toutes les images. En effet, les capteurs classiques que nous utilisons ne permettent de récupérer que des images dont les dynamiques sont faibles. Les niveaux d'éclairement récupérés en une seule prise de vue sont limités d'une part par la quantité minimum d'énergie lumineuse que le capteur est capable de différencier du bruit, et d'autre par la quantité maximum d'énergie qu'il peut détecter sans saturer pendant son temps d'exposition à la lumière. Nous pouvons ainsi considérer que pour un point de l'espace (et donc des images d'entrée), la moyenne des valeurs observées pour chaque temps d'exposition est un bon indicateur de la luminance réelle du point. L'ensemble des points peut alors être construit progressivement en suivant une règle d'échantillonnage simple. Nous commençons par choisir un premier point aléatoirement sur l'image et nous calculons la moyenne des luminances de toutes les images pour ce point. Puis, nous recommençons avec un second point. Si sa valeur moyenne est suffisamment éloignée de celle calculée au premier point, nous l'ajoutons à l'échantillon, sinon ce point est rejeté et nous passons au suivant. Nous réitérons ce processus en comparant systématiquement la moyenne à celle de tous les points précédemment retenus jusqu'à obte-

nir un échantillon couvrant l'ensemble des valeurs possible de l'estimateur. Les essais réalisés montrent que l'utilisation d'une différence de moyenne supérieure à 5 niveaux de gris comme critère de rejet des points donne un échantillonnage satisfaisant. Dans ces conditions, la sélection comporte en général une trentaine de points. Notons cependant que, dans certains cas, cette méthode de sélection va conduire au parcours exhaustif des points de l'image d'entrée sans pour autant renvoyer un échantillon de points suffisant. Si ce cas se présente, la dynamique de la scène observée est trop réduite pour déterminer correctement la fonction de réponse. Il faudra alors compléter les observations avec d'autres jeux d'images sur lesquelles la dynamique manquante sera observable.

2.3. Réduction des temps de calcul

L'objectif de notre étude étant d'évaluer des méthodes existantes afin de les adapter à la réalité augmentée, nous allons maintenant nous intéresser à identifier leurs limitations et notamment le temps de calcul. Ainsi, nous allons chercher à modifier les paramètres afin de diminuer ce temps tout en perdant le minimum de qualité possible.

Dans un premier temps, remarquons que la fonction de réponse de la caméra peut être considérée comme constante. Elle peut donc n'être estimée qu'une seule fois puis conservée pour la suite, ce qui permet de réduire significativement les temps de calcul. Ensuite, nous pouvons remarquer que le nombre d'images prises en entrée est en lien direct avec le temps de calcul. Nous avons donc effectué un test pour estimer l'influence de ce paramètre sur les reconstructions. Pour commencer, la fonction de réponse de la caméra est pré-calculée sur le nombre maximum d'images à notre disposition (12 dans notre exemple). Puis nous effectuons des reconstructions avec 2, 4 et 6 images en entrée ainsi qu'une reconstruction de référence utilisant les 12 images (figure 2). Le choix des images à conserver se fait de manière homogène : dans la liste des images triées par temps d'exposition, nous en gardons une sur 2 (6 images sur 12), une sur 3 (4 images sur 12) et une sur 6 (2 images sur 12). Les reconstructions sont enfin affichées avec un outil de visualisation par plages colorées où chaque demi-ordre de grandeur de luminance correspond à une couleur. Cet outil permet d'apprécier rapidement la cohérence d'une reconstruction par rapport à la composition de la scène. Nous avons également proposé un outil permettant de comparer les différentes reconstructions avec la référence. Il renvoie une image où la couleur de chaque pixel correspond à la distance relative (exprimée en pourcentage) entre l'image et la référence. Les résultats présentés sur la figure 3 soulignent qu'avec la méthode utilisée, les reconstructions restent cohérentes quel que soit le nombre d'images en entrée. Sur notre exemple, en utilisant plus de 4 images, les écarts à la référence sont limités. Cette quantité dépendant du type de scène et d'éclairage, nous pouvons utiliser cet outil pour décider du nombre d'images

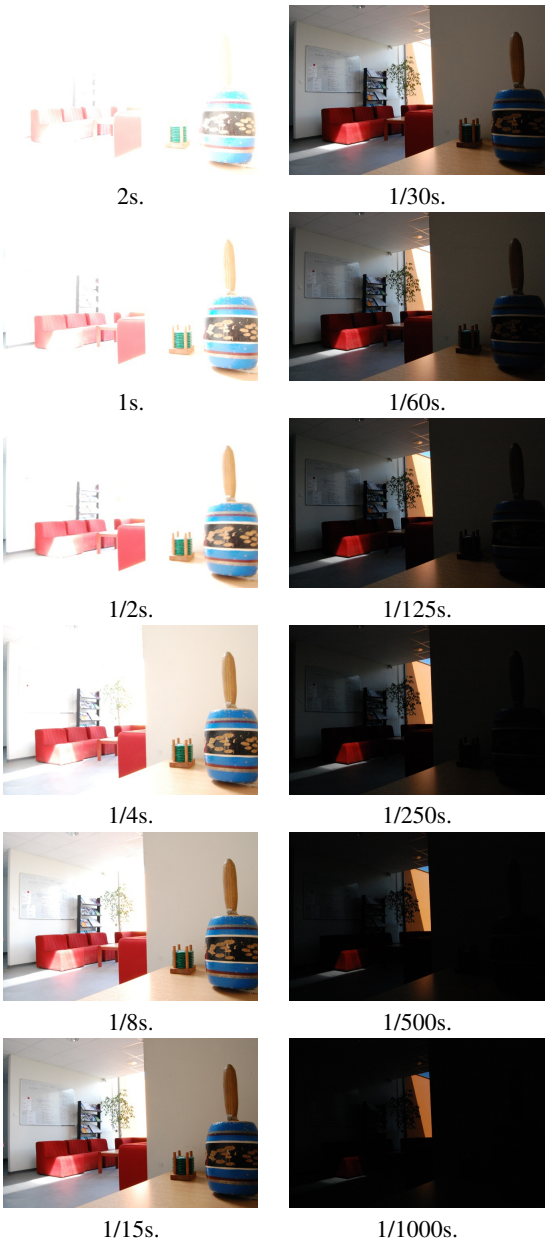

Figure 1: *Jeu d'images en entrée.*

en entrée à garder en fonction du niveau de cohérence souhaité (par exemple moins de 1% de pixels différant de plus de 20% par rapport à l'image de référence).

La réduction du nombre d'images en entrée implique inévitablement de faire un choix dans les temps d'exposition sélectionnés pour faire la reconstruction. Bien que n'ayant aucune influence sur le temps de calcul, ce choix joue un rôle primordial dans la qualité des images HDR reconstruites. Seule une partie de ce domaine est intéressante pour la réa-

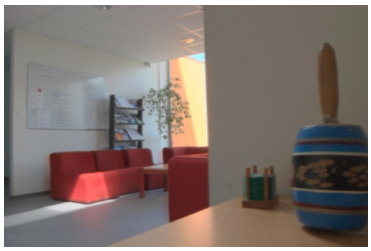

Figure 2: *Tone mapping de l'image HDR de référence.*

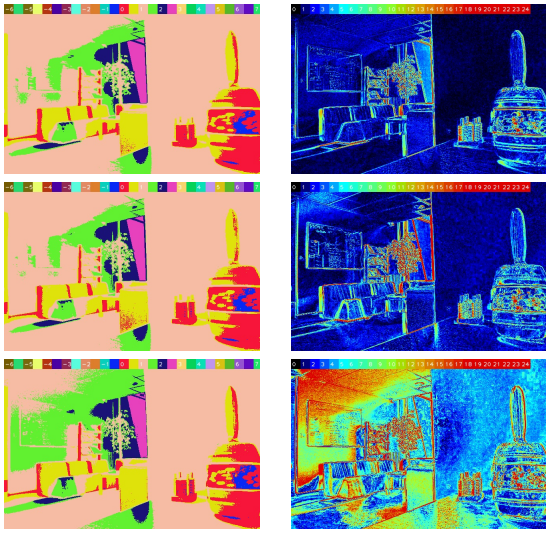

Figure 3: *Influence du nombre d'images en entrée. À gauche : visualisation en plages colorées. À droite : comparaison avec l'image de référence. Ligne du haut : 6 images. Ligne du milieu : 4 images. Ligne du bas : 2 images. L'échelle s'étend du noir pour les pixels ayant des valeurs identiques jusqu'au rouge pour des pixels ayant 25% ou plus de différence.*

lité augmentée : le ré-éclairage d'objets est fait à partir des sources de lumière de l'environnement qui sont observables sur les images à expositions courtes, les détails dans les zones de faible intensité ne sont donc pas importants pour cette application. Cela nous permet d'enlever les images à temps d'exposition élevé (qui sont donc très longues à acquérir) des données d'entrée. Cette action n'est bien entendu possible que si les zones faiblement éclairées sont effectivement inutiles pour le ré-éclairage de l'objet virtuel à insérer dans une application de réalité virtuelle. Si ça n'est pas le cas, il faudra conserver les informations provenant d'images en entrée à temps d'exposition longs. Ce test met également en évidence le fait que les pertes de qualité que nous observons sont dans une certaine mesure prévisibles. En effet, les erreurs observées sont systématiquement dans des zones où l'image portant le plus d'information a été enlevée : par exemple, si une image avec un temps d'exposition court est enlevée, il apparaît des erreurs dans les zones de forte luminance. De plus, un test similaire au précédent a permis de vérifier que le choix de temps d'exposition régulièrement répartis sur le domaine de ceux qui sont potentiellement porteurs d'informations offre les meilleurs résultats.

2.4. Reconstruction à partir de flux vidéo

Pour tester la faisabilité d'une acquisition HDR en utilisant des webcams, nous avons évalué la vitesse de l'algorithme en lui fournissant plusieurs flux vidéo en entrée avec des temps d'expositions différents. Ce test montre que la reconstruction s'exécute à 10 frames par seconde (FPS) pour 2 flux en entrée, 7 FPS pour 4 flux et 5 FPS pour 6 flux. Le nombre de flux conditionne la diversité de temps d'expositions différents, et donc la qualité de la reconstruction. Nous avons utilisé des vidéos de résolution 320×240, prises l'une après l'autre avec une même caméra. Le temps de reconstruction étant directement dépendant du nombre de pixels des images à reconstruire, ceci divise d'emblée le temps de calcul par 4 par rapport à l'implémentation décrite dans la section 2.2, où nous utilisons des images de résolution 640×480. Il est important de noter que les résultats obtenus sont réalisés uniquement avec le CPU et sans optimisation algorithmique ou matérielle particulière. Enfin, il faut remarquer que lors de ce test, aucun recalage des images d'entrée n'a été nécessaire, les scènes étant statiques. Or, dans une application réelle, la scène comporte des objets en mouvement et le recalage devient alors indispensable (figure 4). Cette phase ajoutera un temps de calcul non négligeable. En effet, les méthodes de recalage classiques basées sur les contours sont inadaptées car des images d'une même scène ayant des expositions différentes n'ont pas forcément de contours communs. La méthode de recalage *Mean Threshold Bitmap Alignment Technic* [War03] proposée pour répondre spécifiquement au problème précédent est également peu satisfaisante car elle est trop lente (environ 1s. pour la reconstruction) et ne tient pas compte d'une éventuelle rotation entre les images.

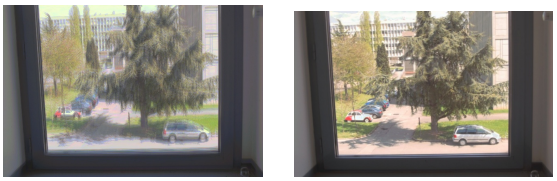

Figure 4: *Tone mapping d'une image reconstruite à partir d'une séquence d'images en mouvement. À gauche : sans recalage. À droite : avec recalage.*

Pour éviter d'avoir à effectuer ce recalage, nous avons réalisé un test avec des images provenant d'un seul flux vidéo à temps d'exposition fixe. Il est important de noter que cet essai revient à étaler la dynamique de ces images en entrée

en les passant dans la fonction de réponse de la caméra sans avoir de données complémentaires permettant de lisser les imperfections de l'acquisition ou les erreurs dans la détermination de la fonction. Cette reconstruction à partir d'une seule image a également été proposée par Debevec et Malik [DM97]. Pour cette essai, la fonction de réponse de la caméra est préalablement calculée à partir de plusieurs images provenant d'un flux vidéo dans lequel nous avons fait varier les temps d'exposition car, avec un seul temps d'exposition, le système à résoudre devient sous-déterminé et il n'est plus possible de déterminer la fonction de réponse. Cet essai a été réalisé dans les mêmes conditions que les autres tests de réduction du nombre d'images d'entrée et le paramètre d'exposition a été choisi au milieu de la gamme disponible. Les résultats de cette reconstruction sont disponibles en figure 5. Ces résultats nous montrent qu'étaler la dynamique de l'image LDR permet une approximation correcte de l'image HDR mais présente, comme on pouvait s'y attendre, des erreurs très importantes dans les zones de faibles et fortes luminances. En effet, ces zones correspondent aux parties de l'image sous et sur-exposées de l'image d'entrée pour lesquelles les informations dont on aurait besoin pour reconstruire l'image HDR complète ne sont pas présentes. Le temps de calcul est de 0,15s. pour une image en 640×480, et de 0,038s. en 320×240, ce qui correspond à 26 FPS et confirme le fait que le temps de calcul des reconstructions est directement proportionnel à la résolution des images traitées. Ceci est très encourageant car nous sommes très proche de la fréquence d'acquisition d'une caméra (30FPS) et donc de la possibilité de reconstruire en temps réel.

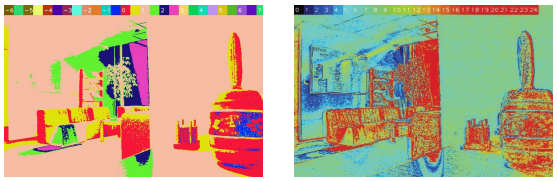

Figure 5: *Reconstruction avec une seule image d'entrée. À gauche : visualisation en plages colorées. À droite : Image des distances relatives de l'images à la référence.*

Une dernière méthode expérimentée consiste à utiliser un flux vidéo unique dans lequel les temps d'exposition varient d'une image à l'autre. Nous disposons alors d'un jeu d'images avec une plage de temps d'exposition variés, nous replaçant dans les mêmes conditions que lors de l'expérimentation de la méthode initiale, avec toutes les optimisations qui ont suivies. Mais les temps d'acquisition sont alors à prendre en compte (0,1s. pour 10 images en faisant varier les temps d'exposition), nous contraignant à effectuer l'acquisition d'une scène réelle puis l'incrustation d'objets virtuels dans cette même scène, mais statique (voir section 4 pour une illustration de ce principe). Pour les mêmes raisons, nous n'effectuons pas de calculs d'estimation de pose et considérons que le point de vue du rendu final correspond au point de vue de l'acquisition. Ainsi, nous orientons nos objectifs exclusivement vers la réalité augmentée sur flux vidéo (ou avec des dispositifs de type *see through*) et non vers des possibilités de réalité augmentée par projection (vidéo-projecteurs, écrans LCD...).

3. Acquisition de la géométrie

3.1. Travaux existants

Il existe de nombreuses méthodes visant à retrouver une géométrie à partir d'images. Deux descriptions de cette géométrie sont possibles. La première est subjective, les données géométriques sont alors représentées sous la forme d'une carte de profondeur (*depth map*) donnant les distances des objets depuis le point de vue de l'utilisateur. La deuxième, qualifiée d'objective, consiste à décrire la géométrie dans un repère 3D global.

Cette représentation globale est utilisée par les méthodes se basant sur la notion d'enveloppe visuelle (*visual hull*) [Lau94]. Elles permettent de déterminer une enveloppe de la forme 3D d'un objet à partir de ses silhouettes 2D issues de plusieurs caméras calibrées. Le principal défaut de cette méthode dite de *shape from silhouette* est qu'il demeure des ambiguïtés quant aux placements des objets reconstruits dans l'espace, ambiguïtés qui peuvent être réduites en augmentant le nombre de caméras ou en utilisant des méthodes de suppression des objets fantômes [MGB*09]. Elle nécessite également de pré-positionner et calibrer plusieurs appareils d'acquisition dans l'environnement afin d'obtenir des résultats ayant une bonne précision. Ceci en fait également une méthode difficile à envisager pour l'exploration d'un environnement inconnu.

Parmi les techniques utilisant des images provenant de plusieurs points de vue, nous pouvons également citer l'ensemble des techniques de stéréo-vision [DA89] [Fau93] [KSK06] qui présentent l'avantage d'être adaptées à une représentation subjective. À partir de deux images d'entrée ayant des points de vue proches, ces techniques vont mettre en correspondance des points de la première et de la deuxième image puis vont déduire la profondeur en fonction du déplacement de ces points entre les deux images et des paramètres intrinsèques et extrinsèques des caméras.

Les méthodes utilisant des images prises d'un même point de vue sont quant à elles mieux adaptées à une réalité augmentée subjective. Nous pouvons là encore distinguer plusieurs approches différentes. Certaines reposent sur un matériel particulier qui va donner de lui-même un moyen de connaître la profondeur des objets présents dans la scène. Parmi ces appareillages spéciaux, nous pouvons notamment citer les systèmes d'ouvertures codées (*coded aperture*) [LFDF07] et les caméras à temps de vol (*Time Of Flight Camera*) [RGY03]. Étant pour la plupart des modèles expérimentaux, ces appareils sortent pour l'instant du cadre de cette étude bien qu'ils présentent un grand intérêt. Une autre

méthode, le *shape from shading* [ZTCS99], consiste à analyser les variations d'éclairement à la surface des objets 3D afin de déterminer leur géométrie. Cette méthode nous est cependant difficile à mettre en œuvre car elle nécessite la connaissance (voire la maîtrise) des sources de lumière présentes dans la scène. Plus récemment, Saxena et al. ont proposé une méthode de détermination des cartes de profondeur par apprentissage à l'aide de champs de Markov [SCN08]. Cette technique, qui fait écho à la méthode d'analyse de scène proposée par Hoiem [Hoi07], demande toutefois de définir et de calculer un grand nombre de descripteurs sur les images et nécessite une base d'apprentissage conséquente.

Enfin, il existe également des méthodes de reconstruction de la géométrie se basant sur la profondeur de champ de la caméra. En effet, les objets qui se situent dans le plan de focalisation sont nets tandis que ceux situés devant ou derrière ce plan sont flous. De plus, la quantité de flou présente en chaque point est directement dépendante de la distance des objets observés au plan de focalisation. Cette approche a été proposée dans un premier temps par Pentland [Pen87] qui donna également les bases de la méthode *depth from defocus*. L'auteur y modélise la quantité de flou présente en chaque point par une fonction de diffusion (*Point Spread Function*) qu'il estime en analysant les images dans le domaine fréquentiel. La connaissance des paramètres de prise de vue (ouverture et distance focale) permet ensuite d'obtenir la distance du point dans l'espace. Cette méthode a été reprise par la suite dans des nombreux travaux, notamment ceux de Xiong et Shafer [XS93] qui proposent l'utilisation de filtres de Gabor pour lever le problème lié au fenêtrage dans la transformée de Fourier locale. D'autres limitations apparaissent lors de l'utilisation de ces méthodes et notamment leur incapacité à déterminer la profondeur des surfaces ne présentant pas de contours marqués (ou de textures). Pour palier cette limitation, Pentland a suggéré l'utilisation d'une source de lumière structurée projetant ainsi la texture nécessaire à l'étude sur les zones uniformes. Cette idée a également été reprise par Nayar et al. [NWN95] qui proposent l'étude de structures de lumière particulières permettant d'estimer des cartes de profondeur denses en temps réel.

Une dernière approche, appelée *depth from focus*, travaille à partir d'un jeu d'images acquises d'un même point de vue en faisant varier la distance de focalisation. Un estimateur local de focalisation est ensuite calculé sur les points des images afin de déterminer l'image sur laquelle chaque zone de la scène est la mieux focalisée ainsi que la profondeur correspondante via le calibrage de la caméra. De nombreux travaux s'appuient sur cette méthode dont ceux de Grossmann [Gro87] et ceux de Nayar et Nakagawa [NN89].

3.2. Construction de la carte de profondeur

Notre environnement de réalité augmentée est constitué essentiellement de matériel standard à bas prix : ordinateurs classiques, webcams, ordinateurs portables avec dispositifs d'acquisition embarqués... Pour extraire l'information géométrique des images, nous avons donc écarté les méthodes utilisant plusieurs caméras, y compris la stéréo-vision, ainsi que les caméras spécifiques, telles que caméras à temps de vol, en raison de leur coût. Nous disposons de caméras nous permettant de modifier la longueur focale, nous avons donc étudié les approches *depth from defocus* et *depth from focus*. La première présente l'avantage de ne prendre que deux images en entrée, avec des paramètres de focalisation différents, mais est complexe à mettre en œuvre, et comporte une ambiguïté dans le résultat : les distances obtenues correspondent à des distances relatives par rapport à un plan de focalisation engendrant le flou moyen, et non à des distances au point de vue, et il n'est pas possible de déterminer dans l'absolu si les objets correspondant sont devant ou derrière ce plan.

Nous avons donc choisi une approche par *depth from focus*. Le principe de cette méthode est d'acquérir une série d'images du même point de vue avec différentes valeurs de distance focale ; puis d'appliquer, sur chaque pixel, un estimateur de focale qui détermine une valeur représentant le niveau de flou ; enfin de calculer une image dans laquelle chaque pixel fournit la distance focale correspondant à l'image la plus nette en ce pixel. Finalement, cette image est convertie en carte de profondeur à partir d'une table de correspondance (ou LUT : *Look Up Table*) calculée lors d'une étape de calibrage et dépendant uniquement du matériel utilisé.

Si une scène présente beaucoup de contours marqués, un moyen simple d'estimer si une image de cette scène est plus ou moins focalisée qu'une autre est d'en étudier le gradient pour tout point des images. Plus le gradient a une valeur élevée, plus nous pouvons considérer que les pentes des contours sont fortes et donc que l'image est nette en ce point. Pour notre estimateur de focalisation, nous avons choisi comme critère la norme du gradient de Sobel : $S(x,y) = \sqrt{(S_h(x,y))^2 + (S_v(x,y))^2}$, $S_h(x,y)$ (resp. $S_v(x,y)$) étant le résultat de la convolution de l'image au point (x,y) par un masque de Sobel horizontal (resp. vertical). Cependant, l'opérateur laplacien peut également être utilisé, soit tel quel, soit modifié comme le proposent Nayar et Nakagawa [NN89]. En effet, le gradient de Sobel va permettre de détecter les variations de niveaux de gris et renverra des valeurs fortes même sur l'intérieur des contours lorsqu'ils sont épais tandis que le laplacien reverra des valeurs fortes uniquement lors des brusques changements de pente des contours et renverra 0 lors d'un changement régulier de niveau de gris. Or, lorsqu'on défocalise une caméra, les pentes des contours seront plus adoucies aux endroits où elles présentent de fortes variations. La valeur renvoyée par le laplacien (en valeur absolue) sera donc elle aussi plus faible sur les points défocalisés. Nous avons donc décidé d'implémenter les deux méthodes afin d'en comparer les résultats.

L'estimateur ainsi défini nous permet de comparer le niveau de flou en un point sur des images ayant des paramètres de focus différents. Nous pouvons associer des paramètres de distance focale à des points de la scène situés à des distances connues de la caméra pour que ces points soient nets. Nous avons utilisé une webcam logitech 9000Pro, qui permet de faire varier la distance focale sur une échelle de 256 valeurs. Les distances de focalisation correspondantes n'étant pas fournies par le constructeur, nous avons réalisé un calibrage avec une mire afin de donner pour chaque valeur du paramètre de focus une estimation de la distance de focalisation correspondante. Nous avons placé la mire à une distance connue de la caméra, puis nous avons pris un jeu d'images en faisant varier le paramètre de focus sur toute la gamme disponible (de 0 à 255). En émettant l'hypothèse réaliste que que tous les points du plan de la mire sont à la même distance de la caméra, nous pouvons calculer pour toutes les images la somme des estimateurs locaux de focalisation (gradient ou laplacien). Nous obtenons alors l'évolution de cette somme en fonction du paramètre de focus, nous permettant d'associer la valeur du paramètre de focus à la distance à la mire. En réitérant cette manipulation pour d'autres distances de mire, nous construisons le tableau de correspondance (LUT) entre la distance et le paramètre de focus. Cependant, au delà d'une certaine distance entre la mire et la caméra (600mm avec nos caméras), il apparait un palier pour lequel toutes les valeurs de focus (comprises entre 0 et 50) renvoient un estimateur de focalisation proche du maximum. On peut donc considérer qu'en deçà de 50, le paramètre de focus de la caméra correspond à une focalisation *à l'infini* et qu'il n'est donc plus possible de déterminer de correspondance avec des distances supérieures à 600mm. De manière similaire, nous avons considéré que les valeurs 253 à 255 correspondent à une distance de 20mm, en sachant que cette valeur signifie uniquement que l'objet est très près. À cette distance, l'hypothèse que tous les points de la mire sont à une même distance est discutable, mais cette approximation reste raisonnable car, en pratique, très peu de scènes contiendront des objets à moins de 20mm de la caméra. Nous avons ensuite décidé de compléter la table de correspondance en interpolant linéairement les valeurs entre les points connus (les paramètres de focus déterminés pour les distances pour lesquelles nous avons des données). Cette interpolation fait l'hypothèse réaliste que la courbe donnant la distance en fonction du focus est strictement monotone. En revanche, il est clair qu'à un paramètre de focus donné ne correspond pas une unique distance mais une plage de distances situées autour de la distance donnée dans la LUT. Ceci est inévitable et est simplement dû à la discrétisation du réglage de focus sur 256 niveaux. Enfin, les différences observées entre les estimations faites avec le laplacien et le gradient sont peu influentes sur le résultat du tableau de correspondance final. Nous n'avons donc implémenté dans l'algorithme effectuant la reconstruction de la carte de profondeur qu'une seule table de correspondance qui s'appliquera aux deux types d'estimation.

Pour reconstruire les cartes de profondeur, nous commençons par réaliser un jeu d'images en faisant varier le paramètre de focalisation de la caméra. Le temps d'acquisition de ce jeu d'images dépend exclusivement de la vitesse de changement de focale propre à la caméra. Dans notre cas, le temps d'acquisition de 10 images à distances focales variables est de 0,18s. Il nous faut ensuite déterminer, pour chaque point, dans quelle image il est le mieux focalisé. Pour cela, nous calculons en chaque point de chaque image de l'ensemble d'entrée l'estimateur de focalisation, puis nous ne conservons que la valeur du paramètre de focalisation de l'image pour lequel cet estimateur est maximum. Cette méthode présente cependant l'inconvénient majeur de calculer des estimations pour tous les points, y compris ceux situés sur des zones non texturées. L'information de contour recherchée n'étant pas présente en ces points, l'estimateur renvoie des valeurs très faibles pour toutes les images. C'est alors le paramètre de focalisation pour lequel le bruit est maximum qui sera conservé. De ce fait, nous avons décidé d'ajouter un seuillage sur l'estimateur qui permet d'éliminer les points sur lesquels l'attribution d'un plan de profondeur n'est pas fiable. Dans cette étude, la valeur du seuil est choisie manuellement. Il est possible d'automatiser ce choix en se basant sur les moyennes et écarts types des estimateurs sur les différentes images. Pour finir, nous reconstruisons la carte de profondeur en remplaçant les valeurs des paramètres de focalisation enregistrés par les valeurs de distances correspondantes dans la LUT calculée lors du calibrage. La figure 6 présente un exemple de carte de profondeur calculé en 9s à partir de 256 images. La figure 7 présente des cartes de profondeur obtenues à partir d'autres jeux d'images.

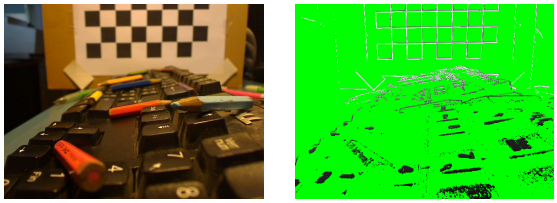

Figure 6: *Exemple de construction de carte de profondeur. À gauche : extrait du jeu d'images en entrée 640×480. À droite : carte de profondeur correspondant à la même résolution. L'échelle de couleurs varie linéairement du noir (2cm ou moins) au blanc (60cm ou plus). Les pixels verts correspondent aux points rejetés lors du seuillage.*

Il est important de remarquer dans cet essai que la qualité de la carte de profondeur obtenue est directement dépendante du seuil choisi et que le seuil optimal va varier d'un jeu d'images à l'autre. Il est donc nécessaire d'avoir un autre indicateur permettant d'anticiper le comportement du couple "jeu d'images / seuil". Nous avons choisi, pour faire cette vérification, de calculer le pourcentage de points déterminés. Il correspond au nombre de points ayant passé au moins une fois le seuil sur le nombre de points total

Figure 7: *Exemples de construction de cartes de profondeur. À gauche : extrait d'autres jeux d'images en entrée 640×480. À droite : cartes de profondeur correspondant à la même résolution.*

de l'image. Cet indicateur va notamment nous permettre de comparer les images de profondeur entre elles en considérant que deux images sont comparables si elles ont des taux de points conservés identiques. Même si cet indicateur est plus cohérent que la valeur de seuil pour l'évaluation des images, sa valeur optimale ne peut pas être déterminée une fois pour toute car elle dépend du pourcentage de points de contours présents dans les images d'entrée. Celui-ci va lui-même grandement varier en fonction de la scène observée. Prendre un pourcentage de points conservés plus élevé que le pourcentage de points de contours fera alors apparaître des contours fantômes à côté des contours réels, ainsi que du bruit. Enfin, nous supposons actuellement qu'une autre source d'erreur de cette méthode pourrait être la présence de maxima locaux dus au bruit et qui produisent des maxima globaux à des endroits où ils ne devraient pas se trouver. Une méthode intéressante pour supprimer ce type d'erreur serait de prendre pour chaque point la valeur médiane de tous les paramètres de focalisation conservés lors du seuillage. Il faut néanmoins pour cela faire l'hypothèse que l'estimateur de focalisation suit approximativement une évolution gaussienne de part et d'autre du paramètre de focalisation le maximisant.

La carte de profondeur ainsi construite n'est renseignée qu'au niveau des contours des objets et n'est donc pas suffisante pour effectuer des calculs d'occlusion. Nous avons donc expérimenté des techniques permettant de créer une carte dense à partir de la carte clairsemée préalablement construite. La méthode la plus convaincante comporte deux étapes. La première effectue un balayage horizontal (resp. vertical) de la carte de profondeur pour remplir les données manquantes entre arêtes renseignées. Chaque pixel non valué entre le bord de la carte et une arête est valué avec la valeur correspondant à l'arrière plan. Chaque pixel non valué entre deux arêtes est valué avec la plus grande des valeurs des deux arêtes (correspondant à l'arête la plus loin). Nous n'effectuons pas d'interpolation linéaire pour éviter la création d'artéfacts entre objets de l'arrière-plan et objets principaux. La seconde étape est une simple moyenne des cartes construites par balayage horizontal et vertical. La figure 8 montre un exemple de remplissage de carte de profondeur. Il existe des techniques plus élaborées, qui tiennent compte des contraintes de lissage entre pixels voisins et effectuent des optimisations globales, mais ces méthodes sont généralement lentes et donc non compatibles avec nos contraintes de temps réel. Notons toutefois qu'il existe une méthode temps réel pour construire une carte de profondeur dense, utilisant la programmation GPU et des sources stéréo [YWY*06].

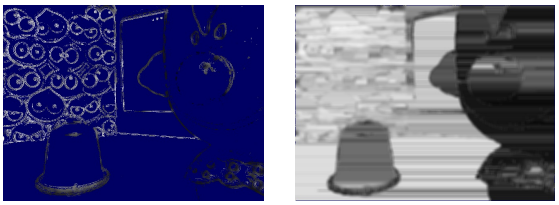

Figure 8: *Remplissage d'une carte de profondeur. À gauche : carte clairsemée. À droite : carte dense.*

3.3. Réduction des temps de calcul

Une première méthode pour réduire les temps de calcul est de réduire le nombre d'images. Lors de la phase de calibrage, nous avons remarqué qu'en raison de limitations matérielles, les images dont le paramètre focal correspond aux extrêmes de la plage des focales ne contiennent pas d'informations pertinentes pour le calcul des profondeurs. Nous pouvons donc réduire le volume des données en entrée en supprimant des images aussi extrêmes, et en enlevant régulièrement certaines images du jeu d'entrée. Pour évaluer les conséquences de ces suppressions, nous avons effectué des tests avec 3 nouveaux jeux d'images. Le premier jeu conserve 1 image sur 10, le second 1 image sur 20, le troisième 1 image sur 40. Le tableau 1 montre les temps de calcul correspondant aux trois jeux d'images. Il est important de remarquer que la suppression d'images du jeu d'entrée influence directement la précision des cartes de profondeur résultantes. En effet, la méthode de reconstruction calcule un nombre de plans de profondeur égal au nombre d'images en entrée (respectivement 20, 10 et 5 dans notre cas). Notons de plus que la sélection effectuée ne correspond pas à une discrétisation régulière de l'espace de profondeur : la LUT

fournit une correspondance non linéaire entre les paramètres de focale et les distances, ce qui est généralement suffisant pour reconstruire une carte de profondeur et évaluer sa cohérence.

Nombre d'images en entrée	Temps de calcul
Jeu complet (256)	9s
20 images	1.5s
10 images	0.8s
5 images	0.5s

Table 1: *Temps de calcul des cartes de profondeur en fonction du nombre d'images en entrée (640×480).*

Ces résultats montrent que la diminution du nombre d'images en entrée entraîne une diminution du temps de calcul qui lui est quasiment proportionnelle. Les cartes de profondeur que nous reconstruisons restent visuellement cohérentes quel que soit le nombre d'images en entrée. La prise en compte d'un nombre moindre d'images en entrée semble donc un bon moyen de gagner du temps afin d'adapter cette méthode à la réalité augmentée.

Une autre méthode pour réduire les temps de calcul consiste à réduire la résolution des images d'entrée et évaluer les conséquences sur les cartes de profondeur. En effet, tout comme nous l'avons supposé pour la luminance, l'information de profondeur nécessaire à l'incrustation d'objets n'a pas besoin d'être très précise et il doit donc être possible d'acquérir des images de résolution moindre pour les applications de réalité augmentée. Nous avons donc refait un test de temps de calcul sur 2 jeux d'images construits en réduisant la résolution des images d'un des ensembles. Les deux nouveaux jeux comportent 256 images faisant respectivement 320×240 et 160×120. La carte de profondeur est calculée en 4,1s. pour le premier jeu, 1,1s. pour le second (figure 9).

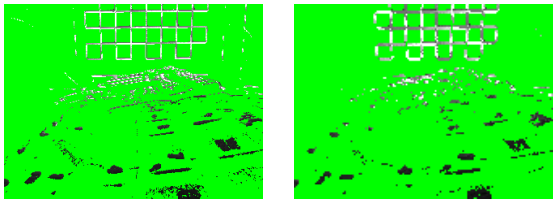

Figure 9: *Reduction de la taille des images avec 256 images en entrée. À gauche : 320×240. À droite : 160×120.*

De même que lors du test précédent, nous observons une réduction de temps de calcul significative, rendant le procédé attractif pour la réalité augmentée. En revanche, nous remarquons que si la réduction à 320×240 donne des cartes qui restent cohérentes, la réduction à 160×120 donne des résultats pour lesquels des erreurs sont très visibles. L'utilisation d'une réduction trop importante n'est donc peut être pas adaptée en l'état mais pourra être testée en implémentant des estimateurs de focalisation différents.

Pour conclure cette étude, nous avons voulu faire un essai en combinant les deux types de réduction. Nous avons donc réalisé une reconstruction sur une sélection de 5 images en 320×240. Le temps de calcul de l'image présentée en figure 10 est de 0,17s. Ce résultat est très encourageant car nous pouvons construire 6 cartes de profondeur par seconde avec 5 images en entrée pour chaque carte, ce qui correspond à la limite d'acquisition des données de 30 images par seconde d'une webcam usuelle.

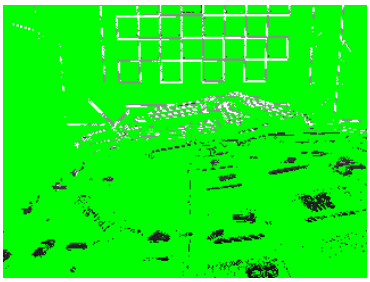

Figure 10: *Exemple de combinaison des réductions de la résolution et du nombre d'images.*

4. Résultats

Afin d'illustrer notre méthode, nous montrons un exemple d'application (figures 12, 13 et 14) en insérant un objet virtuel dans une scène réelle à partir d'une seule capture avec une simple webcam. Nous commençons par capturer la carte de profondeur et une image HDR de la scène en utilisant les techniques que nous avons présentées. Nous capturons une image HDR depuis le point de vue, ainsi que six images HDR prises approximativement du centre de la scène dans les six directions de base (vers l'arrière, l'avant, la gauche, la droite, le haut et le bas). Ces six images sont ensuite converties en *cubemap* (figure 11).

Figure 11: *Cubemap (vue du dessus) construite à partir de 6 image HDR et placage sur une sphère (vue de face).*

Le processus de rendu est décomposé en deux passes.

Pendant la première, nous utilisons une technique de ré-éclairage très simple en plaquant les données HDR directement sur l'objet, puis nous stockons l'image résultante dans une texture. Notons qu'il ne s'agit pas d'un *environment mapping* réel, mais juste d'un simple processus pour affecter des luminances crédibles sur la surface de l'objet. Pendant la seconde phase, nous calculons entre la scène réelle et l'objet virtuel en comparant les profondeurs virtuelles calculées lors de la phase précédente avec les valeurs stockées dans la carte de profondeur. Pour ce test, toutes les images acquises et résultantes sont en 640×480. L'acquisition de l'environnement est effectuées en 0,09s. à partir de 9 images pour l'image HDR et en 0,16s. à partir de 9 images pour la carte de profondeur. Le calcul de l'image HDR prend 2,5s. et la calcul de la carte de profondeur prend 5s. Le rendu de la scène est effectué en temps réel.

Figure 12: *Rendu d'un objet virtuel (bigguy) dans une scène réelle. En haut : flux d'entrée et incrustation d'un objet virtuel, sans ré-éclairage ni calcul d'occlusions. En bas : incrustation d'un objet virtuel, avec ré-éclairage et calcul d'occlusions.*

Figure 13: *Rendu d'un objet virtuel (sphère) dans une scène réelle.*

Figure 14: *Rendu d'un objet virtuel (bunny) dans une scène réelle.*

5. Conclusion et perspectives

Cette étude a montré qu'il existe de nombreuses méthodes d'acquisition de la profondeur et de l'éclairage. Nous avons également vu que beaucoup d'entre elles sont difficiles à exploiter dans un contexte de réalité augmentée parce qu'elles requièrent trop de calculs ou que les données utilisées sont incompatibles avec les besoins. Nous avons donc sélectionné plusieurs méthodes que nous jugeons bien adaptées, et nous les avons expérimentées de manière à évaluer leur potentiel pour la réalité augmentée. Ces tests ont montré qu'une acquisition d'images HDR et de cartes de profondeur est possible en temps interactif avec du matériel standard.

Nos travaux ouvrent également de nombreuses perspectives pour l'acquisition d'environnement. Dans un premier temps, il est possible d'envisager l'optimisation algorithmique et matérielle des méthodes existantes, afin de les adapter aux processeurs graphiques et centraux multi-cœurs. L'acquisition de jeux de données étalons par l'intermédiaire de matériel spécialisé sera nécessaire pour tester les méthodes que nous employons. Nous envisageons également d'acquérir de nouvelles données sur l'environnement, telles que les caractéristiques de surface des matériaux composant les objets de la scène.

Enfin, il faut être conscient que les informations que nous récupérons à l'heure actuelle ne sont pas directement intégrables dans une application de réalité augmentée. En effet, les données photométriques reconstruites représentent l'ambiance lumineuse du point de vue de l'utilisateur. Or, pour ré-éclairer un objet, il faut connaître la lumière arrivant au point de la scène où il doit être inséré. Dans la suite de nos travaux, il nous faudra donc nécessairement rechercher des méthodes permettant d'obtenir cette information à partir de celles que nous connaissons. De même pour les données géométriques, une étape de mise en correspondance entre l'échelle de l'environnement et de celle de l'ob-

jet sera inévitable. Cependant, le contexte de la réalité augmentée est également un avantage car il donne accès à un certain nombre d'informations complémentaires (la position de l'utilisateur, l'orientation de l'axe optique des caméras, de multiples prises de vue proches) sur lesquelles nous pourrons nous appuyer pour créer des méthodes d'acquisitions plus robustes et plus rapides.

6. Remerciements

Ces travaux sont soutenus par l'Agence Nationale pour la Recherche (ANR) dans le cadre du projet ANR-07-MDCO-001. Ils bénéficient également du soutien du projet LIMA du cluster ISLE de la région Rhône-Alpes.

Références

[DA89] DHOND U., AGGARWAL J. : Structure from stereo : A review. *IEEE Transactions on System, Man and Cybernetics. Vol. 19*, Num. 6 (1989), 1489–1510.

[DM97] DEBEVEC P. E., MALIK J. : Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97)* (1997), pp. 369–378.

[Fau93] FAUGERAS O. : *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1993.

[Gro87] GROSSMANN P. : Depth from focus. *Pattern Recognition Letters. Vol. 5*, Num. 1 (1987), 63–69.

[Hoi07] HOIEM D. : *Seeing the World Behind the Image : Spatial Layout for 3D Scene Understanding*. PhD thesis, Carnegie Mellon University, 2007.

[KSK06] KLAUS A., SORMANN M., KARNER K. : Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR06* (2006), pp. III : 15–18.

[KUWS03] KANG S. B., UYTTENDAELE M., WINDER S., SZELISKI R. : High dynamic range video. In *SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers* (2003), ACM, pp. 319–325.

[Lau94] LAURENTINI A. : The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 16*, Num. 2 (1994), 150–162.

[LFDF07] LEVIN A., FERGUS R., DURAND F., FREEMAN W. T. : Image and depth from a conventional camera with a coded aperture. In *Proceedings of the 34th annual conference on Computer graphics and interactive techniques (SIGGRAPH '07)* (2007), pp. 70–78.

[Mad93] MADDEN B. : *Extended Intensity Range Image*. Tech. rep., University of Pennsylvania, 1993.

[MGB*09] MICHOUD B., GUILLOU E., BOUAKAZ S., BARNACHON M., MEYER A. : Towards Removing Ghost-Components from Visual-Hull Estimations. In *ICIG* (septembre 2009).

[MN99] MITSUNAGA T., NAYAR S. : Radiometric Self Calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '99)* (1999), vol. 1, pp. 374–380.

[MP95] MANN S., PICARD R. W. : Extending dynamic range by combining different exposed pictures. In *Proceedings of the 48th Annual Conference of The Society for Imaging Science and Technology (IS&T)* (1995), pp. 442–448.

[NN89] NAYAR S., NAKAGAWA Y. : *Shape from Focus*. Tech. rep., The Robotics Institute, Carnegie Mellon University, 1989.

[NWN95] NOGUCHI M., WATANABE M., NAYAR S. : Real-time focus range sensor. In *Procedings of the 5th International Conference on Computer Vision (ICCV '95)* (1995), pp. 995–1001.

[Pen87] PENTLAND A. P. : A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 9*, Num. 4 (1987), 523–531.

[RGY03] R. GVILI A. KAPLAN E. O., YAHAV G. : Depth keying. In *Proceedings of SPIE Electronic Imaging Conference* (2003), vol. 5006, pp. 564–574.

[RWPD05] REINHARD E., WARD G., PATTANAIK S., DEBEVEC P. : *High Dynamic Range Imaging : Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann Publishers Inc., 2005.

[SCN08] SAXENA A., CHUNG S., NG A. : 3-d depth reconstruction from a single still image. *International Journal of Computer Vision. Vol. 76*, Num. 1 (2008), 53–69.

[War03] WARD G. : Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *Journal of Graphics Tools. Vol. 8*, Num. 2 (2003), 17–30.

[XS93] XIONG Y., SHAFER S. : Depth from focusing and defocusing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '93)* (1993), pp. 68–73.

[YWY*06] YANG Q., WANG L., YANG R., WANG S., LIAO M., NISTER D. : Real-time global stereo matching using hierarchical belief propagation. In *The British Machine Vision Conference (BMVC)* (2006).

[ZTCS99] ZHANG R., TSAI P., CRYER J., SHAH M. : Shape from shading : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 21*, Num. 8 (1999), 690–706.