

Chaîne de Traitement pour la Numérisation et le Rendu Réaliste de Peintures d'Art

Frédéric Larue[†] Lucas Ammann[†] Jean-Michel Dischler[†]

LSIIT UMR 7005 CNRS-ULP - Université Louis Pasteur Strasbourg I, France

Résumé

La numérisation et la visualisation sont aujourd'hui deux domaines d'importance dans le cadre de l'héritage culturel, pour la conception de galeries d'art virtuelles par exemple. Malgré de nombreux travaux, permettre la navigation temps réel autour de copies numériques complexes demeure, dans le cas général, un problème difficile du fait de la complexité de la mesure et de la quantité de données à traiter. Dans ce papier, nous introduisons une nouvelle chaîne de traitement dédiée autant à la numérisation qu'au rendu réaliste de peintures d'art. Nous exploitons le fait que les variations géométriques pour de tels objets, même si elle sont généralement faibles, restent non négligeables d'un point de vue visuel. Contrairement à la plupart des méthodes existantes, nous proposons donc d'acquérir la géométrie en plus de la texture couleur. Ces deux informations sont exploitées lors du rendu en utilisant, pour la couleur, un modèle analytique estimé à partir des mesures et, pour la géométrie, une approche hybride qui combine deux techniques de rendu de relief selon l'échelle, fournissant ainsi un schéma adaptatif permettant un rendu temps réel, quelque soit le point de vue. Notre protocole garantit donc un rendu préservant le relief de la toile, ainsi qu'une représentation très compacte de la copie numérique finale.

1. Introduction

Dans le domaine de l'informatique graphique, de nombreux travaux s'évertuent, depuis de nombreuses années, à améliorer la qualité des images de synthèse grâce à des objets de plus en plus complexes, comme ceux produits par les dispositifs de numérisation par exemple. Mais la qualité visuelle n'est pas le seul objectif : la vitesse des algorithmes de rendu est également un élément d'intérêt en informatique graphique, et de nombreux chercheurs tentent de trouver des compromis acceptables entre vitesse et qualité visuelle.

Dans le domaine de l'héritage culturel, l'acquisition numérique est désormais couramment utilisée pour la conservation et la diffusion de pièces d'art. Le projet Michelangelo [LPC*00], par exemple, s'est fixé pour objectif de numériser les principales sculptures de Michel-Ange, avec pour résultat une base de données de maillages 3D très détaillés. En effet, les scanners 3D fournissent aujourd'hui des données qui sont de plus en plus précises, mais donc forcément de plus en plus complexes à manipuler. Du fait de

la grande variété d'objets susceptibles d'être numérisés, il est intéressant de se poser la question de l'usage de méthodes d'acquisition et de rendu dédiées à certaines classes d'objets. Dans cet article, nous nous sommes concentrés sur la numérisation et le rendu réaliste temps réel de peintures d'art. Nous souhaitons offrir une méthode simple d'un point de vue technique mais qui permette la conception de galerie d'art numérique, c'est à dire en préservant une certaine précision visuelle.

L'état de l'art sur la numérisation de tableaux reste assez pauvre. De manière générale, la géométrie est approximée par un simple plan et l'information chromatique capturée par un ensemble de photographies. Cependant, dans la réalité, la surface d'une toile peinte peut présenter de nombreuses aspérités, dont certaines sont suffisamment significatives pour devoir être prises en compte lors du rendu si l'on souhaite garantir un certain réalisme. En fait, le relief joue un rôle important, et les effets qu'il induit sur la toile sont même parfois souhaités par le peintre lui-même, comme pour la peinture au couteau par exemple. Ainsi, l'information géométrique doit être considérée autant lors de l'acquisition que pour ce qui est du rendu.

[†] e-mail: {larue, ammann, dischler}@lsiit.u-strasbg.fr



Figure 1: Gauche: *photographie du Tableau du Port*. Milieu: *rendu par plaquage de texture classique sur un simple plan*. Droite: *rendu à l'aide notre méthode hybride, adjoint d'une texture bidirectionnelle*.

Nous proposons une nouvelle méthode pour acquérir précisément les informations géométriques et chromatiques d'une peinture d'art, ainsi qu'un algorithme pour la visualisation des copies numériques obtenues. Nous utilisons un modèle analytique pour représenter la texture de manière réaliste et une approche hybride pour afficher les détails géométriques, accélérée par un schéma adaptatif. Nous montrerons que le rendu temps réel est atteint et que les données représentant la copie finale sont particulièrement compactes. Nos principales contributions sont donc :

- Une chaîne de traitement, allant de l'acquisition du relief et de la texture jusqu'à leur rendu final,
- Un jeu de pré-traitements spécifiquement dédiés aux données provenant de la numérisation de tableaux,
- Un algorithme efficace permettant, pour un faible coût mémoire, d'obtenir un rendu réaliste en temps réel grâce à un mécanisme adaptatif spécifique.

Après une brève discussion concernant les travaux antérieurs, nous présentons le principe de notre chaîne de traitement (section 3), incluant la numérisation et l'algorithme de rendu, respectivement décrits dans les sections 4 et 5. Ensuite, nous présentons quelques résultats en section 6 et concluons avec des perspectives possibles pour ces travaux en section 7.

2. Travaux antérieurs

Concernant la numérisation de tableaux d'art, nous pouvons citer Tominaga *et al.* [TTK04], qui ont développé une technique permettant l'acquisition de la couleur d'une peinture à l'aide d'une caméra multi-bande. Dans ces travaux, le relief de la surface n'est considéré qu'au moment de l'acquisition : la rugosité n'est utilisée que pour déterminer la réflectance spectrale de la peinture et non durant la phase de rendu. Les

travaux de Grattoni *et al.* [GS03] présentent une nouvelle méthode pour numériser des surfaces peintes, en termes de géométrie et de couleur, à l'aide d'un outil optique. Cependant, la géométrie considérée est celle du support de la peinture (et non celle de la peinture elle-même) puisqu'il s'agit ici de numériser des surfaces telles que des fresques dans un souci de conservation et de restauration. Par rapport à la restauration, les auteurs de [GAL03] montrent comment ils numérisent à l'aide d'une caméra 3D une peinture sur bois de Léonard De Vinci. Ce projet utilise la géométrie pour fournir un diagnostic sur l'état du support en bois de la peinture, mais ne considère pas le relief à des fins de visualisation, contrairement à notre approche.

Pour produire un rendu réaliste de surface complexe présentant un relief de faible amplitude, il existe de nombreuses solutions. La première est, bien évidemment, l'utilisation d'un maillage extrêmement détaillé, mais cette solution est généralement peu attrayante pour le rendu temps réel. Une seconde peut être la simulation visuelle de ce relief, qui fournit de très bonnes performances d'affichage par rapport à l'utilisation de maillages. Le *bump mapping* [Bli78] a été la première méthode de ce genre. Un plaquage de texture modifié est utilisé pour perturber les normales à la surface de l'objet, simulant le relief par les modifications induites sur le calcul d'illumination. Les résultats visuels sont bons, les performances excellentes, et certains travaux ont déjà proposé de faire l'acquisition de textures *bump* à partir d'objets réels [RTG97]. Malheureusement, certains problèmes de parallax peuvent survenir pour les angles d'observation rasants, d'autant plus lorsque le relief à simuler est important. Des techniques comme le *displacement mapping* [Coo84] ont alors proposé de supprimer ces problèmes en modifiant directement la géométrie sous jacente. Plus récemment, le *relief map-*

ping [OBM00,POC05,BD06] et le *parallax mapping* [Tat06] ont été introduits pour ajouter de manière purement visuelle des détails géométriques à l'aide de techniques complexes de plaquage de textures, ne nécessitant que des objets 3D basiques comme support de rendu, et fournissant donc des rendus rapides et de relativement bonne qualité.

En plus de la géométrie, l'autre caractéristique importante d'une peinture est, bien entendu, sa couleur, qui doit également être acquise très précisément afin d'être restituée plus tard de manière fidèle. Les changements de couleur sur la surface sont liés à la variation spatiale de la réflectance induite par les différentes peintures utilisées. De nombreux travaux se sont attachés à la mesure et au rendu de réflectances de surfaces. La *fonction de texture bidirectionnelle (BTF)* [DNGK97] a été introduite dans ce but. Une BTF encode les variations spatiales d'un matériau, incluant les changements de BRDF, sur la surface d'un objet. Comme la BTF est une fonction à six dimensions, peu attrayante pour le rendu temps réel dû à sa grande consommation mémoire, des méthodes ont cherché à compresser les données ainsi acquises [LFTG97, MWL*99, NDM05]. Pour plus d'informations, un survol des BTF a été publié par Müller *et al.* [MMS*05]. A l'instar de McAllister *et al.* [MLH02], notre méthode utilise un modèle de Lafortune [LFTG97] pour représenter la réflectance en chaque point de la surface du tableau. Comme souligné dans [MMK03], la mesure de BTF pour des surfaces à relief important introduit de grandes imprécisions. C'est pourquoi notre méthode se concentre à la fois sur l'acquisition et le rendu du relief de la peinture au lieu d'utiliser directement une BTF.

La plupart des techniques précédemment citées sont, sous certaines conditions, peu adaptées du fait de leur coût en temps de calculs ou de leur qualité de rendu inadéquate. Les méthodes hybrides peuvent alors fournir une alternative intéressante. De tels mécanismes consistent à utiliser des algorithmes de rendu différents en fonction de certains paramètres, comme la position relative du point de vue ou la taille de l'objet projeté à l'écran. Becker *et al.* [BM93], par exemple, utilisent alternativement trois algorithmes différents, selon le point de vue et sa distance à la scène. Le principal problème des méthodes hybrides réside dans les transitions entre les différents modes de rendu, qui sont souvent enclines à générer des artefacts visuels. Comme le souligne Heidrich *et al.* [HDKS00], les artefacts les plus importants proviennent des différences d'illumination entre les différents algorithmes. Généralement, une composition alpha (parfois améliorée, comme dans [GW07]) permet d'éviter ces effets de 'popping' en proposant des transitions progressives. Ces problèmes mis à part, les méthodes hybrides restent des outils puissants pour accélérer le processus de rendu. Cependant, il n'existe pas de solutions générales : chaque technique est adaptée à une situation spécifique. C'est pourquoi nous avons choisi d'introduire notre propre méthode, spécifiquement conçue pour le rendu de peintures numérisées.

3. Principe

La méthode proposée ici est basée sur le fait que la plupart des peintures d'art présentent souvent des caractéristiques géométriques qui méritent d'être préservées. Une partie du relief au dessus de la toile est, en effet, généralement significative d'un point de vue visuel. Mais la grande majorité du relief reste néanmoins négligeable et peut donc être approximée par une technique de rendu simpliste. Nous proposons alors de découpler le relief significatif du reste.

Tout d'abord, la surface du tableau est approximée par un champ de hauteurs, c'est à dire une carte définissant en chaque point une élévation par rapport au plan de la toile. Cette représentation est particulièrement adaptée aux tableaux, du fait de la forme hautement planaire de ce type d'objets. Le relief est alors classifié par seuillage du champ de hauteurs. Les points en dessous du seuil sont affichés à l'aide d'un simple *bump mapping* et les autres, assimilés au relief significatif, sont représentés par un ensemble de boîtes situées au dessus du plan de la toile et affichées à l'aide d'un rendu de reliefs plus complexe. La fusion des deux techniques est gérée par un mécanisme adaptatif, accélérant de manière drastique la vitesse de rendu.

Le rendu couleur du tableau est finalement obtenu grâce à une description des variations spatiales de son matériau, représentée par une texture de lobes de Lafortune [LFTG97] estimés à partir des mesures et évalués à la volée par le matériel graphique.

4. Acquisition et Pré-Traitements

Dans cette section, nous explicitons nos choix concernant l'acquisition de la géométrie et de la texture. Nous présentons également les pré-traitements nécessaires à notre algorithme hybride de rendu de peintures numérisées.

4.1. Acquisition 3D

La géométrie est acquise à l'aide d'un scanner à lumière structurée. Etant donné que les détails chromatiques peuvent présenter de forts contrastes, nous capturons plusieurs cartes 3D, avec différents temps d'exposition, afin de garantir la bonne acquisition des régions claires et sombres. Du fait de la forme particulière de la classe d'objets que nous nous proposons de numériser, un seul point de vue est utilisé pour l'acquisition de l'information 3D, choisi perpendiculairement au plan de la toile. La paramétrisation 2D de la carte 3D et l'angle de vue quasi-orthogonal nous fournissent une représentation géométrique déjà proche du champ de hauteurs que nous désirons obtenir. Voyons comment extraire proprement ce dernier.

4.2. Extraction automatique de la toile

Selon l'application, on peut vouloir supprimer l'information qui ne concerne pas directement la surface peinte. Il est alors

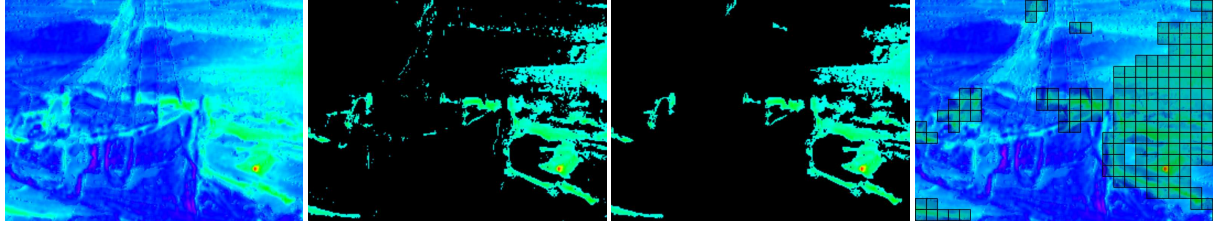


Figure 2: Classification du relief. De gauche à droite: 1) champ de hauteurs résultant de l'alignement par ACP de la carte de profondeurs acquise, mise en valeur par un gradient de couleur, 2) seuillage des coordonnées de hauteurs, 3) nettoyage pour réduire l'influence du bruit, 4) boîtes créées autour des composantes connexes restantes et utilisées plus tard pour le rendu du relief significatif.

intéressant de pouvoir disposer d'un traitement permettant d'extraire la toile en ignorant le cadre et toute autre information géométrique non pertinente. La particularité d'un tableau numérisé est la forme hautement plane de sa géométrie. En effet, son relief peut être vu comme une variation de hauteur par rapport à un plan de référence. Pour récupérer le champ de hauteurs correspondant au tableau, nous effectuons une analyse en composantes principales (ACP) de la carte 3D afin d'extraire un nouveau repère local $(\vec{X}, \vec{Y}, \vec{Z})$, où \vec{X} et \vec{Y} sont les axes de principale dispersion, tangent au plan de la toile, et \vec{Z} est l'axe orthogonal. Les positions des points de la carte 3D sont alors recalculées par rapport à ce nouveau repère, et la coordonnée associée à l'axe \vec{Z} est appelée *coordonnée de hauteur*. En considérant ce nouveau repère, la transition entre la toile et le cadre provoque nécessairement une discontinuité importante de la hauteur comparé aux petits détails géométriques de la toile elle-même. Ainsi, nous appliquons l'opérateur de Sobel sur les coordonnées de hauteur afin de mettre en valeur ces fortes discontinuités. Pour détecter la partie intérieure de la peinture (la toile), nous appliquons un algorithme de croissance de région qui s'arrête lorsqu'une zone de gradient trop important est rencontrée. Nous choisissons comme point de départ de la croissance le centre de la carte 3D, puisqu'il est évident qu'un tableau correctement mesuré se trouve bien centré dans le champs du dispositif d'acquisition. Seule la région résultant de ce remplissage est conservée, ignorant ainsi les points qui n'appartiennent pas à la toile. Une érosion de quelques pixels est ensuite appliquée afin d'éviter la présence d'artefacts résiduels. Une fois que seule l'information pertinente demeure, une nouvelle ACP est appliquée pour recalculer un repère local $(\vec{X}, \vec{Y}, \vec{Z})$ qui soit plus représentatif des seules données de la toile.

4.3. Classification du relief

Notre but est maintenant d'extraire la partie du relief de la toile qui est suffisamment significative, d'un point de vue visuel. Nous définissons un seuil δ_0 en dessous duquel la hauteur est considérée comme négligeable. Actuellement, ce seuil est fourni manuellement. Durant nos tests, nous avons fixé sa valeur entre 20% et 40% de la hauteur maximale

de la toile. Tous les pixels de la carte 3D dont la hauteur est supérieure à δ_0 sont marqués *valides*, et les autres *non valides*, de manière à construire un masque binaire \mathcal{M}_v . Les composantes connexes de \mathcal{M}_v qui sont trop petites (quelques pixels) sont supprimées de manière à ignorer la possible influence du bruit de numérisation.

La grille de pixels de la carte 3D est ensuite subdivisée en un ensemble de cellules de $n \times n$ pixels. Nous verrons plus tard, dans la section 5, que la valeur de n est importante pour notre algorithme de rendu hybride puisqu'il permet de contrôler la granularité du mécanisme adaptatif. Les cellules qui ne contiennent aucun pixel valide dans \mathcal{M}_v sont supprimées. Pour les autres, une boîte est créée, dont les dimensions sont définies par la taille n en pixel de la cellule, et par la hauteur maximale de la toile. Le relief segmenté est alors totalement inclus dans l'ensemble final de boîtes, comme l'illustre la figure 2. Cet ensemble sera utilisé pour rendre séparément la partie significative du relief. De plus, nous calculons également pour chaque boîte la hauteur h_{max} et les coordonnées image (u_{max}, v_{max}) de son point le plus élevé, ce qui nous sera utile pour le rendu adaptatif, décrit plus tard en section 5.3. Il est évident, mais important, de noter que plus n est petit, plus il y aura de boîtes qui seront créées. Le choix de n est discuté dans la section résultats (section 6).

4.4. Acquisition de la texture bi-directionnelle

Concernant l'information chromatique, l'apparence de la toile est capturée à l'aide de plusieurs photographies, prises pour des points de vue et des éclairages différents. La localisation de la caméra est effectuée à l'aide de la paramétrisation par lumière structurée décrite dans [LD06], et la localisation de la source lumineuse est obtenue par une structure mécanique rigide permettant de couvrir l'ensemble de l'hémisphère au dessus de l'objet.

Une fois les photographies recalées sur la géométrie, des échantillons de luminance sont extraits pour chaque point de la carte 3D par projection dans l'espace image de chaque photographie. Ces échantillons sont composés des directions locales d'observation et d'illumination et de la couleur du pixel atteint. Parmi ces échantillons, certains sont sujets aux

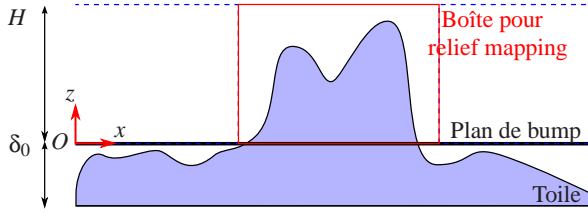


Figure 3: Positions relatives du plan de bump et des boîtes. La boîte rouge contient une partie du relief significatif.

auto-occlusions, c'est à dire lorsque la lumière est stoppée par la géométrie avant d'arriver en ce point. Certains modèles analytiques de représentation de texture sont sensibles, en termes de précision, aux fortes discontinuités que peuvent représenter de tels effets d'ombre. Comme l'information de relief est connue dans notre cas, nous choisissons d'éliminer les échantillons qui ne correspondent pas à de l'illumination directe. Ceux-ci sont détectés par lancé de rayons.

Un modèle de Lafortune est finalement estimé en chaque point de la carte 3D. Nous utilisons l'approximation de [MLH02] à un lobe spéculaire, comme dans l'équation 1 :

$$f_r(\vec{v}, \vec{l}) = \rho_d + \rho_s (C_x v_x l_x + C_y v_y l_y + C_z v_z l_z)^k \quad (1)$$

où ρ_d et ρ_s sont des vecteurs RVB décrivant respectivement les contributions diffuses et spéculaires à l'énergie lumineuse sortante, et \vec{v} et \vec{l} sont les directions locales d'observation et d'incidence lumineuse. Nous avons choisi ce modèle pour sa simplicité, permettant une évaluation directe par le matériel graphique, et pour sa compacité, étant donné que seules trois textures 2D sont nécessaires : deux pour ρ_d et ρ_s et une pour les paramètres (C_x, C_y, C_z, k) de forme du lobe. Evidemment, d'autres modèles de représentation peuvent être utilisés, mais nous rappelons que l'un de nos objectifs est la compacité de la copie numérique finale.

5. Rendu

Après classification du relief, nous obtenons deux sortes de relief, le *négligeable* et le *significatif*, chacun étant affiché à l'aide d'une technique différente : le *bump mapping* et le *relief mapping*, respectivement. Ces deux techniques sont fusionnées grâce à un mécanisme qui choisit automatiquement le meilleur algorithme selon les conditions d'observation, accélérant ainsi le rendu. Le relief significatif est représenté par l'ensemble de boîtes déterminé au cours des traitements précédents. En plus de ces boîtes, nous créons une texture contenant le champ de hauteurs défini précédemment ainsi que les vecteurs normaux en chaque point de la toile. Ces normales sont calculées en considérant l'ensemble des triangles qu'il est possible de construire avec le 8-voisinage d'un point dans la carte 3D. Seules les composantes n_x et n_y sont stockées, la troisième étant recalculée à la volée. Cette texture est utilisée pour le rendu des deux types de relief.

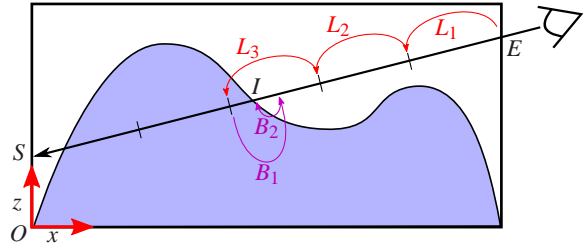


Figure 4: Recherche d'intersection entre un rayon et un champ de hauteurs effectué sur GPU par l'algorithme du relief mapping.

5.1. Rendu du relief négligeable

Pour le relief négligeable, la surface de la toile est approximée par un plan qui est rendu en utilisant un *bump mapping* classique. Comme la partie significative du relief ne commence qu'au delà du seuil δ_0 , un espace existe entre la hauteur minimum de la toile et la hauteur de la partie basse des boîtes. Pour palier à ce problème, nous déplaçons le plan de bump jusqu'à ce seuil. Comme l'illustre la figure 3, la vraie toile se trouve en fait sous le plan de bump, mais la hauteur maximale de la toile reste la même. Le relief de la peinture reste ainsi totalement respecté.

5.2. Rendu du relief significatif

L'algorithme de rendu utilisé pour le relief significatif est une variante de l'algorithme du *relief mapping* décrit par Policarpo *et al.* [POC05]. Les zones rendues à l'aide de cet algorithme sont superposées au plan de bump, ce qui permet de corriger les erreurs de parallax introduites par ce dernier lorsque le relief devient trop important.

Rappel sur le relief mapping Le *relief mapping* est une extension du plaquage de textures classique. Il utilise la GPU pour modifier la géométrie de l'objet rendu. Pour chaque pixel rendu d'un polygone simple, un rayon est lancé pour déterminer une intersection avec la surface réelle de l'objet, qui est définie par un champ de hauteurs dont l'intensité des pixels est interprétée comme une variation de hauteur au lieu d'une simple couleur. Cette intersection est trouvée par une recherche linéaire, suivie d'un raffinement binaire, respectivement représentés par les étapes L_i et B_j de la figure 4. Lorsque l'intersection entre le rayon et la surface réelle est trouvée, l'illumination du pixel peut être calculée.

Relief mapping à partir de boîtes Habituellement, l'algorithme du *relief mapping* utilise un simple polygone comme support de rendu. Notre méthode utilise des boîtes définies dans l'espace 3D, chacune étant définie par sa position (u_b, v_b) et sa taille (w_b, h_b) dans l'espace texture du champ de hauteurs. Chaque boîte est donc associée à une portion de la texture, qui est, par conséquent, partagée entre toutes les boîtes. Les hauteurs minimales et maximales sont



Figure 5: Comportement de notre rendu adaptatif. Pour les gros plans (gauche) ou les angles rasants (droite), un nombre plus important de boîtes est automatiquement sélectionné afin de compenser l'erreur de parallax introduite par le bump.

les mêmes pour toutes les boîtes, et correspondent respectivement au niveau du plan de bump et à la hauteur H la plus élevée de toute la toile.

Pour un rendu efficace du *relief mapping* à partir de boîtes, nous associons des coordonnées de texture 3D à chaque sommet afin de définir un repère local à chaque boîte (O_x et O_z sur la figure 4) à l'intérieur duquel le lancer de rayon sera effectué. Au cours du rendu, ces coordonnées de texture fournissent immédiatement le point d'entrée du rayon dans la boîte, quelqu'en soit la face considérée. L'algorithme de *relief mapping* procède alors ainsi : pour chaque pixel atteint par la boîte actuellement rendue, les points d'entrée E et de sortie S du rayon d'observation sont déterminés à l'aide des coordonnées de texture 3D. Ces deux points nous donne le rayon de traversée le long duquel la recherche d'intersection avec le champ de hauteurs doit être faite. Les phases de recherche linéaire et binaire sont alors effectuées de la même manière que pour le *relief mapping* standard.

5.3. Mécanisme de rendu adaptatif

Une peinture observée selon une direction proche de la normale à sa surface ne présente quasiment aucune différence visuelle entre le relief significatif et le relief négligeable. Cela est également valable lorsque le point de vue est éloigné, c'est-à-dire lorsque les variations de hauteur deviennent négligeables par rapport à leur taille projetée à l'écran. En effet, la distance entre la surface réelle et le plan de bump devient alors trop petite pour être visible.

Nous avons donc développé un algorithme pour déterminer automatiquement si une boîte doit être affichée ou non. De plus, une composition alpha est effectuée pour éviter d'éventuels artefacts de 'popping'. L'état d'une boîte est donné par un coefficient α_{box} calculé à partir du plus haut point I de la boîte, dont les coordonnées dans la scène sont facilement récupérées à partir des valeurs h_{max} and (u_{max}, v_{max}) calculées précédemment (voir section 4.3). En considérant la projection B de I sur le plan de bump le long de la direction d'observation \vec{v} , nous pouvons voir que le ratio $r = \|\vec{IB}\| / \|\vec{CI}\|$, où C est la position de la caméra, dépend directement de l'angle de vue, mais également de la distance au point de vue, comme l'illustre la figure 6.

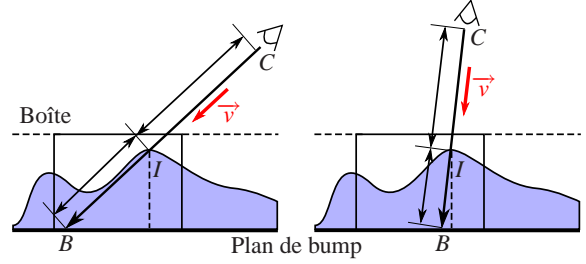


Figure 6: Calcul du coefficient de transition α_{box} à partir du point le plus haut de chaque boîte I . Le ratio $\|\vec{IB}\| / \|\vec{CI}\|$ dépend de l'angle de vue et de la distance à la surface.

Ce ratio tend vers zéro pour des points de vues éloignés ou frontaux, et augmente lorsque l'on se rapproche ou que l'angle de vue devient rasant. Ce ratio est donc pertinent pour notre mécanisme adaptatif. Le calcul par boîte du coefficient α_{box} est donné par l'équation 2:

$$\alpha_{box} = \begin{cases} 0 & \text{si } Kr < \epsilon_{min} \\ 1 & \text{si } Kr > \epsilon_{max} \\ \frac{Kr - \epsilon_{min}}{\epsilon_{max} - \epsilon_{min}} & \text{sinon} \end{cases} \quad (2)$$

où ϵ_{min} et ϵ_{max} sont deux valeurs seuil et K est un facteur défini par l'utilisateur pour contrôler la sensibilité de la transition entre les deux algorithmes de rendu. Selon la valeur de α_{box} , il y a trois cas à considérer. Si α_{box} est nul, la boîte n'est pas dessinée. Si $\alpha_{box} = 1$, la boîte est affichée sans composition alpha. Si $\alpha_{box} \in]0, 1[$, la boîte est composée avec le plan de bump.

Généralement, des inconsistances lumineuses peuvent apparaître lorsque différents algorithmes de rendu sont combinés. Dans notre cas, la même normale, la même information de couleur et le même modèle d'illumination sont utilisés pour les deux algorithmes. Ainsi, aucune différence d'illumination n'est visible au niveau des transitions, comme on peut le voir sur la figure 7.

5.4. Illumination

Comme décrit en section 4.4, nous utilisons un modèle de Lafortune pour calculer l'illumination. L'équation 1 est di-

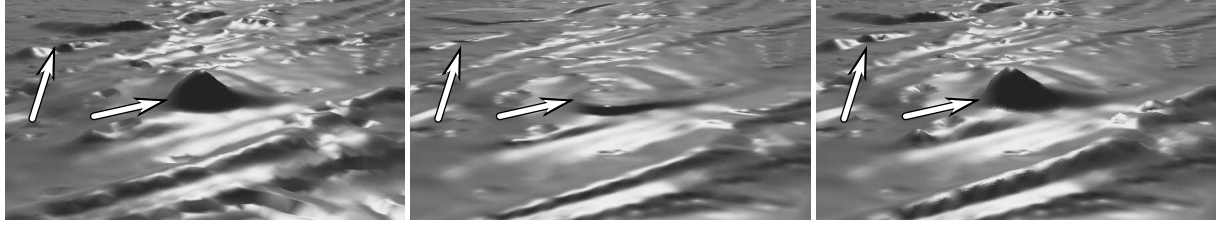


Figure 7: Comparaison entre le rendu maillage (gauche), le bump mapping (milieu) et notre rendu hybride (droite). L'erreur de parallax introduite par le bump mapping est corrigée par l'utilisation de relief mapping là où le relief devient significatif.

rectement évaluée par la carte graphique. L'information de couleur est disponible via deux textures RGB et une RGBA. Les direction d'illumination et d'observation sont toutes deux corrigée à l'aide de la coordonnée de hauteur, même pour le bump, de manière à prendre en compte la vraie position de la surface, et le vecteur normal est recalculé à la volée à partir des coordonnées n_x et n_y estimées précédemment (voir section 5).

6. Résultats et Discussions

Nous avons testé notre méthode sur le Tableau du Port, illustrée en figure 1 par une photographie originale et un rendu de notre copie numérique. Le principal avantage de notre méthode est l'absence de maillage pour représenter la géométrie, ce qui engendre un faible coût mémoire. En fait, la carte 3D capturée par le scanner est directement traitée et utilisée comme champ de hauteurs, travaillant donc directement sur l'échantillonnage géométrique initial. Seules quatre textures flottantes 16bits (une RGBA et trois RGB) et une liste de boîtes sont nécessaires pour représenter la copie numérique finale, incluant la texture bidirectionnelle. La taille totale après traitement est d'environ 11Mo. Le maillage correspondant contient 815K triangles.

Les performances obtenues avec notre méthode hybride, reportées dans la table 1, sont temps réel. Les fréquences d'affichage ont été mesurées sur un AMD Athlon X2 4200+ doté d'une carte NVIDIA GeForce 7900GTX. On peut voir que l'accélération apportée par l'adaptativité est significative, surclassant systématiquement le *relief mapping* standard. Notre approche est cependant moins efficace qu'un rendu de maillages pour des points de vue très proches. En effet, la vitesse du rendu adaptatif dépend fortement du point de vue. A angles rasants ou en gros plans, le relief est beaucoup plus visible et l'adaptativité requière plus de boîtes, comme le montre la figure 5, diminuant d'autant les performances. Ce mécanisme dépend aussi du nombre total de boîtes, qui est directement lié à leur taille. Pour une petite taille, les boîtes approximent bien le relief mais le calcul de visibilité par boîte devient plus coûteux. Au contraire, une taille de boîte plus grande entraîne un nombre de pixels plus important à rendre par *relief mapping*, algorithme plus coûteux que le simple *bump mapping*. Des tests de performance sont donnés dans la table 2 pour différentes tailles de boîte.

Dist.	Angle de vue	Maillage	RM	Hybride	Hybride + Adapt.
proche	rasant	71	36	33-48	36-57
	face	85	32	39-48	56-72
cadré	rasant	60	100	61	330
	face	52	74	60	350
loin	rasant	96	280-450	82	650
	face	96	155-230	80	1250

Table 1: Vitesse d'affichage (Hz) pour le Tableau du Port avec différents algorithmes de rendu et pour différentes conditions d'observation.

Taille boîte	# Boîtes	Face	Interm.	Rasant
5 × 5	2073	305	142	18
10 × 10	714	300	185	38
20 × 20	253	330	250	75
50 × 50	81	240	175	145
100 × 100	35	135	130	125

Table 2: Vitesse de rendu (Hz) de notre méthode hybride adaptative pour différentes tailles de boîte et différents angles d'observation.

Le *relief mapping* est particulièrement adapté au rendu de peintures d'art du fait de leur nature planaire. En effet, les peintures traditionnelles peuvent être correctement représentées par des champs de hauteur. L'autre avantage à utiliser le *relief mapping* est la correction des erreurs de parallax introduites par le *bump mapping*, comme le montre la figure 7. Cependant, il est important de remarquer que le rendu de champs de hauteur texturés impose certaines limitations. En effet, si un saut important se produit entre les hauteurs de deux pixels adjacents, la couleur le long de la falaise générée est simplement interpolée entre ces deux points. Ainsi, plus les discontinuités dans le relief sont importantes, plus la texture est étirée. Cette limitation rend difficile l'extension de notre méthode au rendu de bas-reliefs. Un autre problème du *relief mapping* est, comme le souligne [BD06], la présence d'artefacts dus à l'échantillonnage discret du rayon lors de la recherche d'intersections. Dans notre cas, le relief est vraiment faible par rapport à ce qui est généralement rendu à l'aide de cet algorithme, et ces artefacts sont donc limités. Cela nous permet d'ailleurs de réduire le nombre de pas pour la recherche linéaire afin d'accélérer encore un peu le rendu.

7. Conclusions et Perspectives

Dans ce papier, nous avons introduit une nouvelle chaîne de traitement pour numériser et restituer des peintures d'art. Cette méthode permet d'acquérir la géométrie et la couleur d'un tableau. Nous utilisons un rendu hybride basé sur deux algorithmes différents pour tenir correctement compte du relief de la peinture, selon son importance. Un mécanisme adaptatif est utilisé pour combiner les deux algorithmes. Le rendu temps réel est atteint, et pour un faible coût mémoire car la peinture n'est représentée que par un champ de hauteurs et trois textures pour son matériau bidirectionnel.

Actuellement, nous n'avons travaillé que sur un petit tableau. Notre chaîne pourrait être étendue pour permettre l'acquisition de grandes toiles, ne rentrant pas entièrement dans le champs du scanner, ce qui pose actuellement problème pour certains des pré-traitements. Enfin, étant donné que nous avons remarqué que le rendu par maillage était plus efficace pour les points de vue proches, nous voudrions ajouter un troisième niveau de rendu, utilisant le maillage lui-même pour l'échelle la plus fine.

Références

- [BD06] BABOUD L., DÉCORET X.: Rendering geometry with relief textures. In *Proc. of the 2006 conference on Graphics interface* (2006), pp. 195–201.
- [Bli78] BLINN J. F.: Simulation of wrinkled surfaces. In *Proc. of SIGGRAPH '78* (1978), pp. 286–292.
- [BM93] BECKER B. G., MAX N. L.: Smooth transitions between bump rendering algorithms. In *Proc. of SIGGRAPH '93* (1993), pp. 183–190.
- [Coo84] COOK R. L.: Shade trees. In *Proc. of SIGGRAPH '84* (1984), pp. 223–231.
- [DNGK97] DANA K. J., NAYAR S. K., GINNEKEN B. V., KOENDERINK J. J.: Reflectance and texture of real-world surfaces authors. In *Proc. of the Conference on Computer Vision and Pattern Recognition* (1997), p. 151.
- [GAL03] GUIDI G., ATZENI C., LAZZARI S.: 3d optical scanning diagnostics for Leonardo Da Vinci's "Adorazione dei Magi" conservation. In *3DIM* (2003), pp. 110–115.
- [GS03] GRATTONI P., SPERTINO M.: A mosaicing approach for the acquisition and representation of 3d painted surfaces for conservation and restoration purposes. *Mach. Vision Appl.* 15, 1 (2003), 1–10.
- [GW07] GIEGL M., WIMMER M.: Unpopping: Solving the image-space blend problem for smooth discrete lod transitions. *Computer Graphics Forum* 26 (2007), 46–49.
- [HDKS00] HEIDRICH W., DAUBERT K., KAUTZ J., SEIDEL H.-P.: Illuminating micro geometry based on pre-computed visibility. In *Proc. of SIGGRAPH '00* (2000), pp. 455–464.
- [LD06] LARUE F., DISCHLER J.-M.: Automatic registration and calibration for efficient surface light field acquisition. In *Proc. of VAST06* (2006), pp. 171–178.
- [LFTG97] LAFORTUNE E. P. F., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. In *Proc. of SIGGRAPH '97* (1997), pp. 117–126.
- [LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital Michelangelo project: 3d scanning of large statues. In *Proc. of SIGGRAPH '00* (2000), pp. 131–144.
- [MLH02] MCALLISTER D. K., LASTRA A., HEIDRICH W.: Efficient rendering of spatial bi-directional reflectance distribution functions. In *Proc. of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2002), pp. 79–88.
- [MMK03] MESETH J., MÜLLER G., KLEIN R.: Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium* (2003), pp. 89–96.
- [MMS*05] MULLER G., MESETH J., SATTLER M., SARLETTE R., KLEIN R.: Acquisition, synthesis, and rendering of bidirectional texture functions. *Computer Graphics Forum* 24, 1 (2005), 83–109.
- [MWL*99] MARSCHNER S. R., WESTIN S. H., LAFORTUNE E. P., TORRANCE K. E., GREENBERG D. P.: Image-based BRDF measurement including human skin. In *Rendering Techniques '99, Proc. of the Eurographics Workshop* (1999), pp. 131–144.
- [NDM05] NGAN A., DURAND F., MATUSIK W.: Experimental analysis of BRDF models. In *Eurographics Symposium on Rendering* (2005), pp. 117–126.
- [OBM00] OLIVEIRA M. M., BISHOP G., MCALLISTER D.: Relief texture mapping. In *Proc. of SIGGRAPH '00* (2000), pp. 359–368.
- [POC05] POLICARPO F., OLIVEIRA M. M., COMBA J. L. D.: Real-time relief mapping on arbitrary polygonal surfaces. In *Proc. of the 2005 symposium on Interactive 3D graphics and games* (2005), pp. 155–162.
- [RTG97] RUSHMEIER H. E., TAUBIN G., GUÉZIEC A.: Applying shape from lighting variation to bump map capture. In *Proc. of the Eurographics Workshop on Rendering Techniques '97* (1997), pp. 35–44.
- [Tat06] TATARCHUK N.: Dynamic parallax occlusion mapping with approximate soft shadows. In *Proc. of the 2006 symposium on Interactive 3D graphics and games* (2006), pp. 63–69.
- [TTK04] TOMINAGA S., TANAKA N., KOMADA T.: Imaging and rendering of oil paintings using a multi-band camera. In *Southwest Symposium on Image Analysis and Interpretation* (2004), pp. 6–10.



Figure 8: Gauche: copie numérique du Tableau du Port dans une galerie virtuelle. Droite: illustration de notre mécanisme adaptatif. Pour les angles rasants, plus de boîtes sont sélectionnées pour éviter les erreurs de parallax dues au bump mapping.