



HAL
open science

Stylisation d'objets éclairés par des cartes d'environnement HDR

Romain Vergne, Xavier Granier

► **To cite this version:**

Romain Vergne, Xavier Granier. Stylisation d'objets éclairés par des cartes d'environnement HDR. 20èmes Journées de l'Association Française d'Informatique Graphique (AFIG), Nov 2007, Marne la Vallée, France. inria-00300472v2

HAL Id: inria-00300472

<https://inria.hal.science/inria-00300472v2>

Submitted on 27 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stylisation d'objets éclairés par des cartes d'environnement HDR

Romain Vergne et Xavier Granier

INRIA LaBRI Universités de Bordeaux

Abstract

In this paper, we introduce a pipeline for an interactive and stylized rendering of High-Dynamic Range (HDR) environment-mapped objects. By using stylization on HDR images, the quality, the segmentation and the details are improved. Furthermore, this pipeline allows an easy combination of 2D stylization (on the environment map, and on the images) and 3D stylization (on the object). Based on this pipeline, we present new interactive styles that illustrate our approach. Through these styles, we detail the purpose and the action of each rendering step, in order to show its flexibility and to allow the reader the implementation of new styles.

Dans cet article, nous introduisons un pipeline de rendu permettant de styliser de manière interactive des objets éclairés par des cartes d'environnement à grande dynamique (High-Dynamic Range ou HDR). L'utilisation d'images HDR permet d'améliorer la qualité de certains traitements, comme les segmentations ou l'extraction des détails. De plus, cette architecture permet de combiner facilement des stylisations 2D (sur la carte d'environnement et sur les images) et 3D (sur les objets). Les nouveaux styles que nous présentons sont basés sur ce pipeline et illustrent la flexibilité de notre approche.

1. Introduction

Une carte d'environnement représente une solution simple et efficace pour simuler un éclairage distant. Un intérêt croissant s'est porté sur cette technique dans la communauté scientifique durant les dernières années (recherche sur le précalcul du transfert de la radiance [SKS02]) dans le but de développer des effets d'éclairage complexes et réalistes. Les cartes d'environnement sont aussi un outil de modélisation simple pour combiner des images capturées avec des données synthétiques [ALCS03], et sont intégrées dans la plupart des logiciels de modélisation et/ou de rendu.

Nous pensons que, grâce à cette adaptabilité, les cartes d'environnement peuvent être utilisées dans un contexte de rendu non-photoréaliste (Non-Photorealistic Rendering ou NPR) pour les mêmes raisons : elles représentent une solution basée-image simple et efficace. Elles permettent en particulier de combiner une stylisation 2D en précalcul sur la carte d'environnement avec une stylisation 3D interactive une fois la carte appliquée à un objet. Néanmoins, cette double stylisation peut provoquer des problèmes de cohérences et une des contributions de cet article est de les résoudre.

De plus, l'intérêt croissant apporté aux cartes d'environnement est combiné avec le développement de techniques utilisant des images à grande dynamique (High-Dynamic Range Imaging - HDRI) [RWPD06]. L'HDRI fournit un espace de travail plus précis et une intégration des données synthétiques plus facile et plus cohérente avec les images 2D. Même si ces qualités sont particulièrement avantageuses pour les rendus réalistes, elle peuvent aussi être très utiles pour les rendus expressifs. Par exemple, les images HDR peuvent permettre des segmentations plus précises, souvent utilisées dans les processus de stylisation. Les intensités des images HDR étant proches de celles de la réalité, il peut être plus facile de simuler le processus de création et de stylisation d'un artiste qui observe un environnement. Il est néanmoins nécessaire d'appliquer un algorithme de réduction de ton (tone-mapping) lorsque l'on veut afficher de telles images, et celui-ci doit être parfaitement intégré dans le processus de rendu. L'utilisation de l'HDR pour le NPR est ainsi un autre objectif de cet article.

Les principales contributions de cet article sont les suivantes : (i) Un pipeline de rendu pour faire du rendu expressif sur des objets éclairés par une carte d'environnement.



FIG. 1: Quelques exemples de rendus interactifs d'objets éclairés par des cartes d'environnement. (a) - style cartoon. (b) - style basé sur une détection des lignes caractéristiques. (c) - les luminances de l'environnement HDR ont été segmentées en six régions. Des canevas composés de points de densité plus ou moins grande sont utilisés et combinés avec la valeur de luminance de chacune de ces régions. En (d), une abstraction des couleurs a été mise en place pour les réflexions sur l'objet 3D, tandis qu'une simple détection des contours est utilisée pour l'arrière plan.

(ii) Une amélioration de la stylisation des images par l'utilisation de données HDR. Nous présentons aussi une brève étude de l'influence d'une réduction de ton sur le résultat final. (iii) Une analyse de l'influence de chaque étape de rendu, illustrée par des styles spécifiques dans ce contexte.

Cet article est organisé de la manière suivante. Nous présentons brièvement les travaux précédents, en nous focalisant sur les rendus expressifs hybrides 2D/3D et sur l'utilisation de l'HDR dans les stylisations. Nous décrivons ensuite l'architecture de notre pipeline, puis nous faisons une comparaison des stylisations entre des images HDR/LDR (Low-Dynamic Range). Grâce à ces stylisations et à notre pipeline de rendu, nous montrons quelques résultats sur des scènes éclairées par des cartes d'environnement, puis nous discutons de ces travaux.

2. Travaux précédents

L'objectif de cette section n'est pas de faire un état de l'art complet du NPR. Nous nous focaliserons principalement sur deux aspects : la stylisation de données 2D et 3D combinées (nécessaire pour travailler avec des cartes d'environnement), et l'utilisation d'images HDR pour le NPR. Le lecteur peut se référer à [GG01, SS02] pour un état de l'art du NPR.

Si des styles similaires existent dans des espaces 2D et 3D, comme les rendus en demi-teintes (2D [SHS02] - 3D [FMS02, FV03]), les rendus à base de traits (2D [Her01] - 3D [Chi06]), très peu de travaux ont été fait pour combiner des données 2D et 3D dans des processus de rendus expressifs interactifs. Les travaux qui s'en rapprochent le plus proviennent des recherches concernant la réalité augmentée. Fischer et al. [FBS05] et Haller et al. [HLB05] ont introduits des stylisations pour améliorer la cohérence entre des données acquises et des données virtuelles. Fischer et Bartz [FB05] utilisent un filtre bilatéral [TM98] combiné avec une détection des contours de Canny [Can86] pour l'arrière plan, et un *toon shading* [LMHB00] combiné avec un rendu des silhouettes [Lan00] pour l'objet 3D. Concernant

les rendus à base de traits, Fischer et al. [FBS05] utilisent un filtre simulant un pinceau sur l'image de la caméra et un système de particules pour l'objet 3D de manière à lui appliquer des coups de pinceau. Haller et al. [HLB05] utilisent un post-traitement sur l'image avec une segmentation des couleurs et une détection des contours. Si ces solutions peuvent être utilisées avec un placage d'environnement sur des scènes, nous pensons que des styles plus complexes et nombreux peuvent être mis en place avec une architecture de rendu plus adaptée, qui inclue l'éclairage des objets par l'environnement.

De nos jours, le placage d'environnement est fréquemment utilisé avec l'HDRI pour améliorer le réalisme, et notamment l'éclairage d'une scène. De plus, les publications récentes ont montrées qu'il était possible d'utiliser l'HDRI pour des travaux plus artistiques. Dans [LFUS06], les auteurs mettent en place une édition locale d'un opérateur d'adaptation de ton dans le but de s'adapter aux désirs de l'utilisateur. Pouvant être considérées comme réalistes, les changements de couleurs et d'intensités en font inexorablement des images non-photoréalistes. Dans ce domaine, l'HDR apporte une meilleure segmentation grâce à une amélioration du processus de contrôle de l'utilisateur. Les images HDR pourraient donc être utilisées pour les rendus NPR, et la grande précision qu'elles apportent pourrait définir une adaptation locale des stylisations. Pour un état de l'art plus précis concernant l'HDRI, l'auteur peut se référer au livre de Reinhard et al. [RWPD06].

3. Architecture

Notre architecture (cf. Figure 2) fournit un pipeline flexible pour du rendu NPR avec des objets sur lesquels on plaque une carte d'environnement. Le processus est basé sur les possibilités offertes par les GPU actuels de manière à subdiviser le rendu en étapes successives et plus spécialement décomposer la stylisation de l'environnement 2D et des données 3D. Cette section décrit les quatre étapes de ce pipeline et montre comment un style différent peut être calculé dans chacune d'entre elles, puis combinés dans l'image finale.

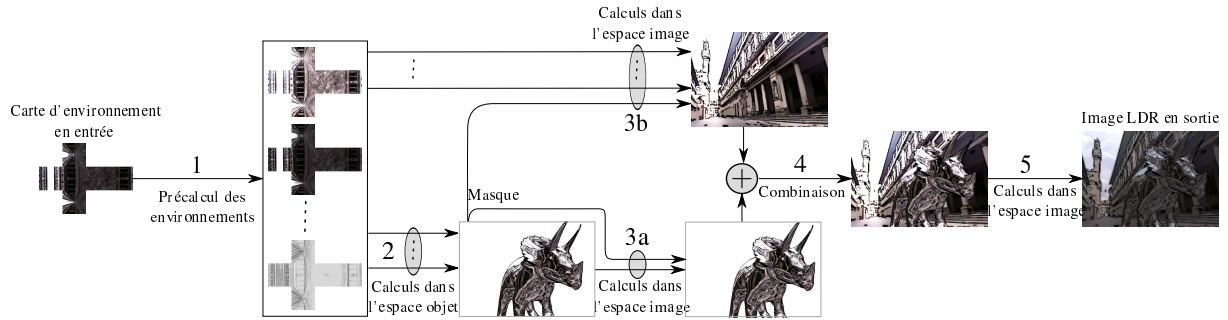


FIG. 2: Notre pipeline de rendu pour la stylisation de scènes éclairées par des cartes d'environnement

1. Précalcul de l'environnement : Le principal intérêt de cette étape est de préparer la carte d'environnement pour le rendu interactif. Elle peut être utilisée pour calibrer les images HDR [RWPD06], pour calculer les luminances utiles à la réduction de ton, pour styliser ces environnements ou même pour générer les niveaux de détail (mip-maps). Une fois cette étape terminée, on dispose alors d'une série de *cube maps* qui seront utilisés et/ou combinés pour représenter l'environnement et pour éclairer l'objet 3D.
2. Shading dans l'espace objet : Durant l'étape suivante, les stylisations 2D sont appliquées directement sur l'objet 3D. Une ou plusieurs *cube maps* de la pile peuvent alors être utilisées et combinées. La stylisation résultante peut être augmentée ou remplacée par des styles obtenus dans l'espace objet, indépendamment de l'environnement, comme la détection des silhouettes. Chaque composant du rendu peut être modifié par l'utilisateur dans le but d'obtenir le style désiré. Cette étape est essentielle pour la cohérence du style dans la scène.
3. Calculs dans l'espace image : dans cette étape, nous fournissons des outils permettant de créer des styles différents pour l'objet et la carte d'environnement afin de mieux les distinguer.
4. - 5. Combinaison et finalisation : L'objet et sa carte d'environnement sont ensuite combinées pour finaliser le rendu. A ce stade, un algorithme de réduction de ton peut être mis en place, ainsi qu'une stylisation globale dans l'espace écran, comme l'ajout d'une texture de papier [CTP*03] ou un filtre (un flou comme dans [HLB05]) pour augmenter la cohérence du rendu final. Une segmentation peut aussi être utilisée pour déterminer où les différentes stylisations doivent être appliquées (cf. Figure 4).

4. Stylisation d'images HDR

Le principal objectif de cette section est de montrer au lecteur une brève vue d'ensemble de ce que peut apporter l'H-DRI au NPR, et d'illustrer comment l'adaptation de ton

[RWPD06] peut interagir avec le processus de stylisation. En effet, la dynamique de ces images étant supérieure à celle des images traditionnelles, il est nécessaire d'utiliser un opérateur de réduction de ton pour les afficher sur nos écrans actuels.

4.1. Adaptation de ton et stylisation

La réduction de ton peut être appliquée avant le processus de stylisation - celle-ci est alors effectuée sur des images classiques, avec une faible dynamique - ou après la stylisation. L'adaptation réduisant la dynamique des images, il se peut alors que celle-ci enlève certains détails. D'un autre côté, le processus de création d'un artiste est généralement de se placer en face de scènes réelles (i.e., données HDR réelles) et ensuite de dessiner/peindre celles-ci. C'est un peu similaire à faire une adaptation de ton après la stylisation. Dans [SK06], les auteurs l'ont expérimenté avec leur propre opérateur [SKMS06]. Pour étendre leurs travaux, nous utilisons l'opérateur de Ward [War94] avant et après trois différentes stylisations : une simple détection de contours avec le filtre laplacien d' [Ope], un style *cartoon* [WOG06] et le style pointilliste [Her98] (cf. Figure 3). Cela confirme que le fait d'appliquer la stylisation directement sur les données HDR augmente la quantité de détails détectés. Les images résultantes contiennent plus de caractéristiques (cf. Figure 3).

Il existe un grand nombre de méthodes pour adapter la dynamique [RWPD06], et chacune d'entre elles possède ses propres qualités intrinsèques. Le choix de l'opérateur d'adaptation de ton produira donc un résultat visuel différent pour une stylisation donnée. La Figure 5 montre quelques exemples. Avec la même calibration [RWPD06], sans mise à l'échelle et avec les paramètres d'adaptation de ton par défaut, l'opérateur de Ward [War94] fournit une meilleure préservation des couleurs que celui de Ferwerda [FPSG96]. Ceci est dû à l'adaptation perceptuelle de la couleur du second opérateur, qui réduit la colorimétrie des scènes très peu éclairées (comme la pénombre). De la même manière, l'opérateur local de Reinhard [RSSF02] fournit une meilleure

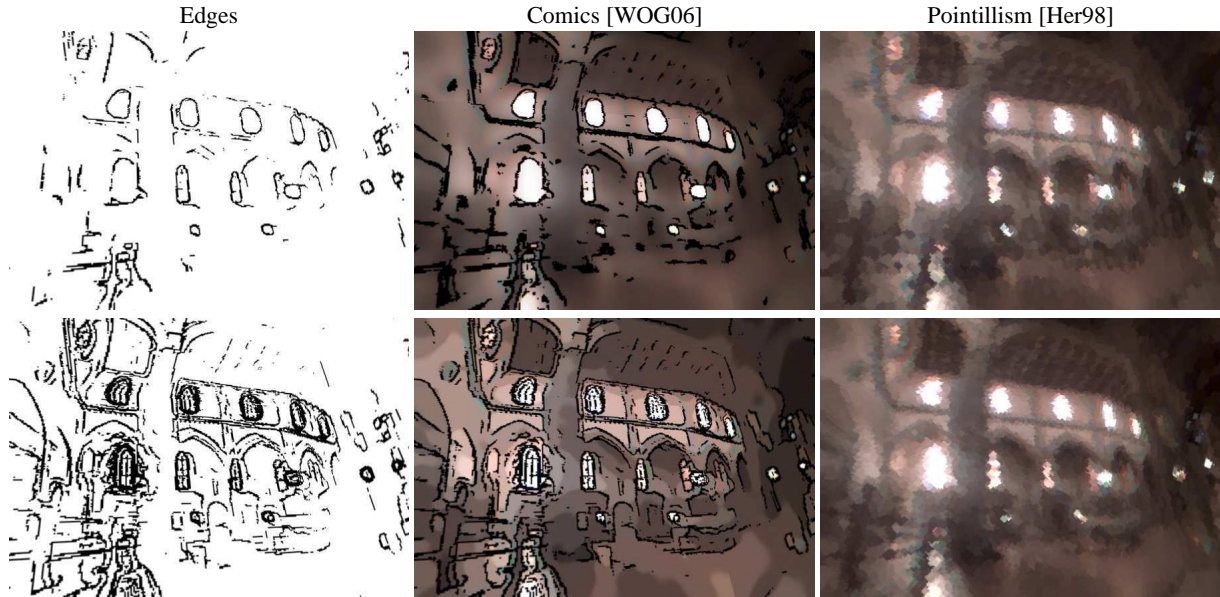


FIG. 3: Comparaison de différents styles avec la réduction de ton appliquée avant (en haut) et après (en bas) la stylisation.

préservation des contrastes qu'un opérateur global comme celui de Ward.

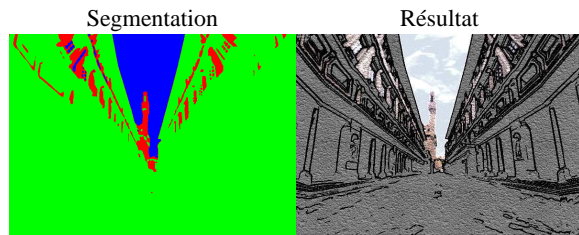


FIG. 4: A gauche : segmentation d'une image HDR basée sur les luminances (bleu → ciel (régions de hautes intensités), rouge → régions d'intensités moyennes, vert → régions de faibles intensités). A droite : stylisation basée sur cette segmentation (bleu → papier bleu, rouge → segmentation des couleurs + canevas, vert → contours + canevas)

4.2. Nouveaux styles avec les images HDR

Avec une grande dynamique, beaucoup plus de détails sont détectés et une segmentation plus précise peut être appliquée sur les images (comme on l'a vu dans la section précédente). Avec ces segmentations plus précises, il devient plus facile d'utiliser un style différent pour chacune des régions de la scène. Cette méthode est similaire à l'opérateur d'adaptation de ton local et paramétré par l'utilisateur développé par Lischinski et al. [LFUS06], mais appliqué dans le contexte du NPR. Dans la Figure 4, nous avons segmenté l'image en trois régions de manière très précise, avec un

simple seuillage sur la luminance. Les régions de fortes intensités correspondent au ciel et les régions de faibles intensités sont celles qui sont éclairées indirectement. Un simple canevas bleu et blanc à l'aspect d'un papier est utilisé pour le ciel, une segmentation des couleurs est appliquée dans les régions intermédiaires et une détection des contours est faite dans la dernière. Le style résultant produit un bon effet de profondeur sur l'environnement, avec cette transition du ciel coloré aux contours noirs.

5. Stylisation d'objets avec cartes d'environnement

Pour illustrer le fonctionnement de notre pipeline de rendu, nous décrivons deux nouveaux styles. Nous exposerons alors les objectifs et les actions de chacune des étapes associées à ceux-ci.

5.1. Dessin interactif de lignes

Durant les précalculs, nous commençons par extraire les contours du *cube map* en utilisant un simple filtre laplacien [Ope]. Nous créons alors les différents niveaux de détail, comme dans [KLK*00], avec une fonction spécifique qui assure une abstraction correcte des lignes durant les transitions. Nous voulons empêcher que trop de détails apparaissent lorsque les niveaux de résolution sont faibles. Nous avons choisis une solution simple qui consiste à créer un pixel noir si trois des quatre pixels de la résolution supérieure sont noirs, et de créer un pixel blanc dans le cas contraire. La solution de simplification progressive de lignes décrite dans



FIG. 5: Comparaison de quelques opérateurs d'adaptation de ton sur un style cartoon (en haut) et sur un style impressionniste (en bas). La même calibration, sans mise à l'échelle, a été utilisée pour toutes les images.

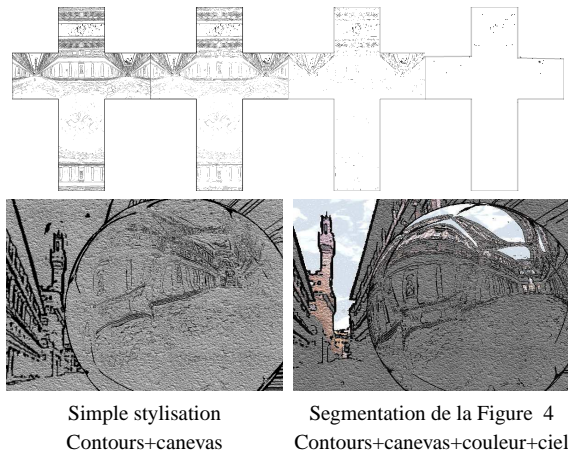


FIG. 6: Style basé sur une détection des contours. Les images du haut représentent les différents niveaux de détail des lignes. L'image en bas à gauche est un rendu utilisant un canevas. Il est possible d'améliorer ce rendu en utilisant la segmentation de la Figure 4 (image en bas à droite).

[BTS05] donnerait certainement de meilleurs résultats. Les niveaux de détail sont présentés dans la Figure 6.

Dans l'espace objet, nous combinons les réflexions du *cube map* stylisé avec une détection des silhouettes des données synthétiques en temps réel, afin d'augmenter la cohérence. Les niveaux de détail corrects sont sélectionnés durant le calcul des réflexions du *cube map*, de manière à préserver la

densité des lignes sur l'écran. La détection des contours de l'objet 3D peut facilement être mise en place dans un *fragment shader*, en utilisant les fonctions prédéfinies d'OpenGL GLSL. L'algorithme suivant montre le code correspondant :

```
float d = ratio*length(fwidth(unorm));
float l = cst*(1+dot(view,normal));
if(abs(dot(view,normal)) < d)
    // contour détecté
    gl_FragColor = vec4(0.0,0.0,0.0,1.0);
else
    // pas de contours
    gl_FragColor = textureCubeLod(...,...,l);
```

normal et *view* correspondent respectivement à la normale et au vecteur de vue projetés puis normalisés dans l'espace image. *unorm* correspond à la normale non-normalisée. Le *ratio* est un paramètre choisi par l'utilisateur qui contrôle la largeur des contours. Notez que nous avons implémenté notre propre sélection de niveaux de détail (*l* dans le code) de façon à forcer le choix d'un niveau moins détaillé lorsqu'un polygone devient tangent au point de vue. Cette sélection de niveau est contrôlée par la constante *cst*.

L'étape d'assemblage combine simplement l'environnement et l'objet stylisé. Pour améliorer la cohérence, un canevas de type papier est multiplié avec l'image combinée lors de la dernière étape. Nous utilisons actuellement une méthode simple pour déterminer les coordonnées de texture. Elle consiste à choisir l'intersection entre le vecteur de vue et le *cube map* pour obtenir les coordonnées du centre de la texture. Néanmoins, cette solution n'évite pas complètement

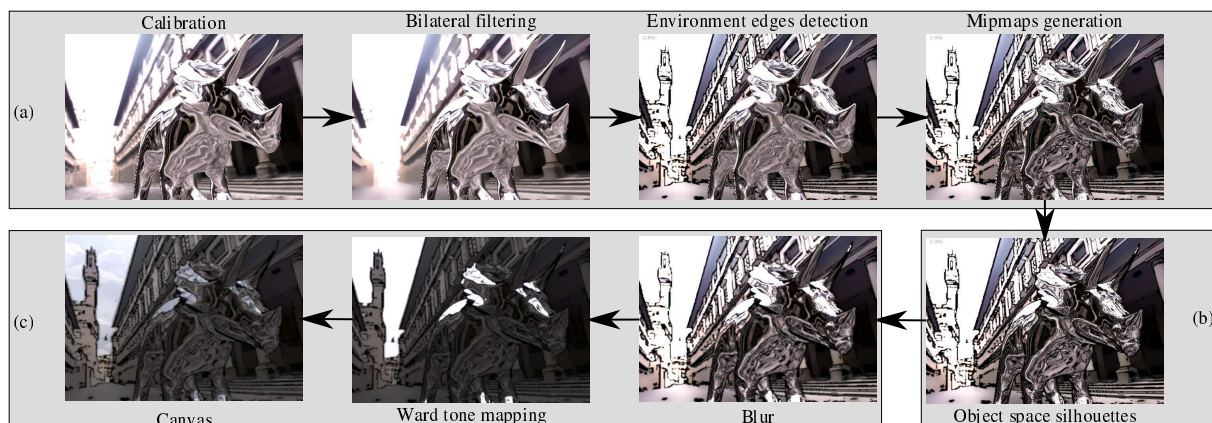


FIG. 7: Pipeline utilisé pour créer le style cartoon : (a) - précalculs, (b) - shading dans l'espace objet, (c) - post-traitements. Nous montrons l'influence de chacune des étapes pour le rendu de ce style.

l'effet "rideau de douche" produit par le placage du canevas. Les solutions décrites dans [CTP*03, BSM*07] pourraient améliorer la sensation de mouvement pendant le déplacement de la caméra. On peut voir le résultat final sur la Figure 6 (en bas à gauche).

Dans la Figure 6, l'image en bas à droite est une simple extension de ce style. Nous avons simplement décidé de segmenter l'image en trois régions avec un *fragment shader* (voir Figure 6), et d'appliquer un différent style/canevas sur chacune d'entre elles (cf. Figure 4). Pour les intensités les plus grandes (i.e., le ciel), nous utilisons un canevas représentant un papier bleu grossièrement froissé. Une simple abstraction des couleurs est utilisée pour le reste de la scène.

5.2. Style cartoon

Ce second style utilise plus d'étapes dans le pipeline, comme le montre la Figure 7. Il est basé sur le style présenté dans [WOG06] (pour l'abstraction de l'environnement).

Lors des précalculs, nous calibrons la carte d'environnement HDR originale [RWPD06] pour calculer un facteur global de mise à l'échelle qui sera utilisé dans les post-traitements pour l'adaptation de ton. Nous appliquons ensuite un filtre bilatéral [TM98] et une détection des contours laplacienne sur le *cube map*. La même méthode présentée dans la section précédente est utilisée pour générer les niveaux de détail des contours. Une moyenne est suffisante pour calculer les niveaux sur l'image couleur obtenue après application du filtre bilatéral.

Dans ce style, le shading de l'espace objet est similaire à celui vu précédemment. Les niveaux de détail corrects sont sélectionnés et reflétés sur l'objet, puis les silhouettes 3D sont ajoutées.

Après l'étape de combinaison, le post-traitement consiste

à appliquer un opérateur d'adaptation de ton, sans faire de lecture arrière (du GPU vers le CPU) pour éviter une perte des performances. Nous avons choisis l'opérateur de mise à l'échelle de Ward [War94] car il est simple et efficace. Une implémentation GPU [GWWH03] d'un opérateur local comme [RSSF02] pourrait néanmoins donner de meilleurs résultats. La principale difficulté de l'opérateur de Ward est de calculer un facteur prenant en compte l'ensemble des valeurs de l'image affichée, directement sur le GPU. Celui-ci est calculé avec un *fragment shader* qui parcourt partiellement l'image (par exemple, tous les 16 pixels) de manière à produire une texture composée seulement d'un pixel et contenant le facteur de mise à l'échelle requis. Il est maintenant facile de récupérer cette valeur dans une dernière passe pour modifier la luminance de l'ensemble des pixels. L'opérateur d'adaptation de ton est donc une simple mise à l'échelle linéaire des valeurs des pixels, et prend en compte ce facteur et celui qui a été calculé lors de la calibration. Enfin, un canevas est ajouté dans le ciel à la dernière étape du pipeline.

6. Resultats et analyse

Style	S1-640x480	S2-640x480	S2-1200x800
Contours	120	100	40
Cartoon	33	30	10

TAB. 1: Fréquence d'affichage pour les différents styles et configuration de scènes. L'objet est composé de 14064 polygones pour toutes les configurations. La configuration 2 (S2) correspond à une vue très proche de l'objet et la configuration 1 (S1) à une vue éloignée. (voir Figure 8).

Les fréquences d'affichage ont été testées sur un PC disposant d'un processeur AMD Turion 64 × 2 1.6 GHz et d'une carte graphique NVIDIA Geforce Go 7600.

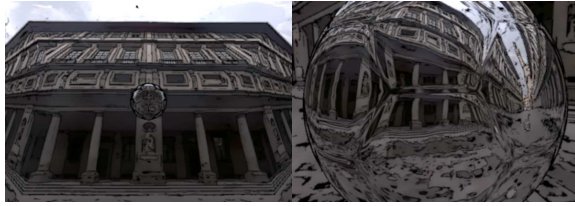


FIG. 8: A gauche : Configuration S1 - Vue large de la scène. A droite : Configuration S2 - Vue réduite.

Comme le montre le tableau 1, le temps de rendu est largement dépendant de la résolution de l'écran. La plupart des opérations de ces deux rendus étant faites dans le *fragment shader*, ce résultat est cohérent. De plus, on remarque le coût des post-traitements de la dernière étape du style *cartoon*. Pour le style détectant les contours, les calculs dans le *fragment shader* sont seulement faits sur les parties de l'image occupées par l'objet et il n'y a pas de post-traitements. Concernant le style *cartoon*, le post-traitement final est appliqué sur l'intégralité de l'image et dépend donc plus de la résolution. Notez que, pour ces deux styles, la résolution du *cube map* est de 6x512x512, et le temps de précalcul est environ de 20 secondes.

Nous nous sommes jusqu'ici focalisés sur la présentation de deux styles de rendu dont l'image finale est très cohérente (sensation qu'un seul style a été appliqué sur l'ensemble de l'image). Une plus grande variété de styles peuvent être mis en place comme on peut le voir dans la Figure 1, avec par exemple, des rendus différents pour l'objet et l'arrière plan. C'est le cas dans la dernière image : une abstraction des couleurs est utilisée pour l'objet et une détection des contours est faite sur l'arrière plan.

7. Conclusion

Dans cet article, nous avons introduits un pipeline flexible pour faire des rendus interactifs et stylisés d'objets sur lesquels on a plaqué un environnement à grande dynamique, combinant des stylisations 2D (sur la carte d'environnement et sur les images) et des stylisations 3D (sur l'objet). Chacune des étapes du pipeline peut facilement être modifiée par un utilisateur pour obtenir l'effet désiré. Nous avons ensuite introduits deux styles de rendu interactif basés sur ce pipeline : le premier est un simple style de détection de contours, et le second combine une détection des contours avec une abstraction des couleurs afin de produire un style de type *cartoon*. De plus, la stylisation d'images HDR apporte une qualité, une meilleure segmentation et extraction de détails, et permet d'élargir la gamme de styles NPR. Avec les styles présentés, nous avons montré que nous pouvons fournir un rendu cohérent (pour la lumière et dans l'espace image), même lorsqu'on sépare le processus pour l'objet et l'environnement.

Comme travaux futurs, nous aimerions améliorer d'une part l'interface pour obtenir un processus plus facile à utiliser. Pour le moment, tous les styles doivent être programmés. Le plus grand challenge, et le but de ce projet, est l'intégration d'objets synthétiques avec une capture de l'environnement en temps réel. Il faudra donc optimiser les précalculs.

References

- [ALCS03] AGUSANTO K., LI L., CHUANGUI Z., SING N. W. : Photorealistic rendering for augmented reality using environment illumination. In *ISMAR '03 : Proc. IEEE & ACM International Symposium on Mixed and Augmented Reality* (2003), p. 208.
- [BSM*07] BRESLAV S., SZERSZEN K., MARKOSIAN L., BARLA P., THOLLOT J. : Dynamic 2d patterns for shading 3d scenes. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2007)* 26, 3 (2007).
- [BTS05] BARLA P., THOLLOT J., SILLION F. : Geometric clustering for line drawing simplification. In *Proc. Eurographics Symposium on Rendering* (2005).
- [Can86] CANNY J. : A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8, 6 (1986), 679–698.
- [Chi06] CHI M.-T. : Stylized and abstract painterly rendering system using a multiscale segmented sphere hierarchy. *IEEE Trans. Visualization and Computer Graphics* 12, 1 (2006), 61–72.
- [CTP*03] CUNZI M., THOLLOT J., PARIS S., DEBUNNE G., GASCUEL J.-D., DURAND F. : Dynamic canvas for immersive non-photorealistic walkthroughs. In *Proc. Graphics Interface* (June 2003).
- [FB05] FISCHER J., BARTZ D. : Stylized augmented reality for improved immersion. In *VR '05 : Proc. IEEE Conference on Virtual Reality* (2005), pp. 195–202, 325.
- [FBS05] FISCHER J., BARTZ D., STRASSER W. : Artistic reality : fast brush stroke stylization for augmented reality. In *VRST '05 : Proc. ACM symposium on Virtual reality software and technology* (2005), pp. 155–158.
- [FMS02] FREUDENBERG B., MASUCH M., STROTHOTTE T. : Real-time halftoning : a primitive for non-photorealistic shading. In *EGRW '02 : Proc. Eurographics workshop on Rendering* (2002), pp. 227–232.
- [FPSG96] FERWERDA J. A., PATTANAIK S. N., SHIRLEY P., GREENBERG D. P. : A model of visual adaptation for realistic image synthesis. In *ACM SIGGRAPH '96* (1996), pp. 249–258.
- [FV03] FUNG J., VERYOVKA O. : Pen-and-ink textures for real-time rendering. In *Proc. Graphics Interface* (2003), pp. 131–138.
- [GG01] GOOCH B., GOOCH A. : *Non-Photorealistic Rendering*. A K Peters, 2001.
- [GWWH03] GOODNIGHT N., WANG R., WOOLLEY C., HUMPHREYS G. : Interactive time-dependent tone mapping using programmable graphics hardware. In *EGRW '03 : Proc. Eurographics workshop on Rendering* (2003), pp. 26–37.
- [Her98] HERTZMANN A. : Painterly rendering with curved brush strokes of multiple sizes. In *ACM SIGGRAPH '98* (1998), pp. 453–460.

- [Her01] HERTZMANN A. : Paint by relaxation. In *CGI '01 : Computer Graphics International 2001* (Washington, DC, USA, 2001), pp. 47–54.
- [HLB05] HALLER M., LANDERL F., BILLINGHURST M. : A loose and sketchy approach in a mediated reality environment. In *GRAPHITE '05 : Proc. international conference on Computer graphics & interactive techniques in Australasia and South East Asia* (2005), pp. 371–379.
- [KLG*00] KLEIN A. W., LI W., KAZHDAN M. M., CORRÊA W. T., FINKELSTEIN A., FUNKHOUSER T. A. : Non-photorealistic virtual environments. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 527–534.
- [Lan00] LANDER J. : Under the shade of the rendering tree. *Game Developer Magazine* 7, 2 (Feb. 2000), 17–21.
- [LFUS06] LISCHINSKI D., FARBMAN Z., UYTENDAELE M., SZELISKI R. : Interactive local adjustment of tonal values. *ACM Trans. Graph.* 25, 3 (2006), 646–653.
- [LMHB00] LAKE A., MARSHALL C., HARRIS M., BLACKSTEIN M. : Stylized rendering techniques for scalable real-time 3d animation. In *NPAR '00 : Proc. International Symposium on Non-Photorealistic Animation and Rendering* (2000), pp. 13–20.
- [Ope] OPENCV : Open computer vision library. www.opencv.org.
- [RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J. : Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21, 3 (2002), 267–276.
- [RWP06] REINHARD E., WARD G., PATTANAIK S., DEBEVEC P. : *High Dynamic Range Imaging - Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann Publishers, 2006.
- [SHS02] SECORD A., HEIDRICH W., STREIT L. : Fast primitive distribution for illustration. In *EGRW '02 : Proc. Eurographics workshop on Rendering* (2002), pp. 215–226.
- [SK06] SMITH K., KRAWCZYK G. : NPR for HDR, stylizing with high dynamic range photographs. In *NPAR 2006 Poster* (June 2006).
- [SKMS06] SMITH K., KRAWCZYK G., MYSZKOWSKI K., SEIDEL H.-P. : Beyond tone mapping : Enhanced depiction of tone mapped HDR images. *Comp. Graph. Forum (Proc. EUROGRAPHICS 2006)* 25, 3 (Sept. 2006), 427–438.
- [SKS02] SLOAN P., KAUTZ J., SNYDER J. : Precomputed radiance transfer for real-time rendering in dynamic, 2002.
- [SS02] STROTHOTTE T., SCHLECHTWEG S. : *Non-Photorealistic Computer Graphics*. Morgan Kaufmann Publishers, 2002.
- [TM98] TOMASI C., MANDUCHI R. : Bilateral filtering for gray and color images. In *ICCV '98 : Proc. the Sixth International Conference on Computer Vision* (1998), p. 839.
- [War94] WARD G. : *Graphics gems IV*. Academic Press Professional, Inc., 1994, ch. A contrast-based scalefactor for luminance display, pp. 415–421.
- [WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B. : Real-time video abstraction. *ACM Trans. Graph.* 25, 3 (2006), 1221–1226.



FIG. 1: Quelques exemples de rendus interactifs d'objets éclairés par des cartes d'environnement. (a) - style cartoon. (b) - style basé sur une détection des lignes caractéristiques. (c) - les luminances de l'environnement HDR ont été segmentées en six régions. Des canevas composés de points de densité plus ou moins grande sont utilisés et combinés avec la valeur de luminance de chacune de ces régions. En (d), une abstraction des couleurs a été mise en place pour les réflexions sur l'objet 3D, tandis qu'une simple détection des contours est utilisée pour l'arrière plan.



FIG. 5: Comparaison de quelques opérateurs d'adaptation de ton sur un style cartoon (en haut) et sur un style impressionniste (en bas). La même calibration, sans mise à l'échelle, a été utilisée pour toutes les images.

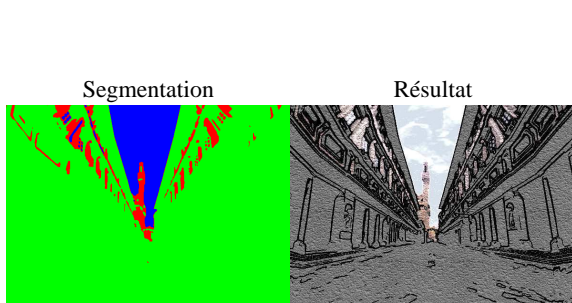


FIG. 4: A gauche : segmentation d'une image HDR basée sur les luminances (bleu→ciel (régions de hautes intensités), rouge→régions d'intensités moyennes, vert→régions de faibles intensités). A droite : stylisation basée sur cette segmentation (bleu→papier bleu, rouge→segmentation des couleurs+canevas, vert→contours+canevas)

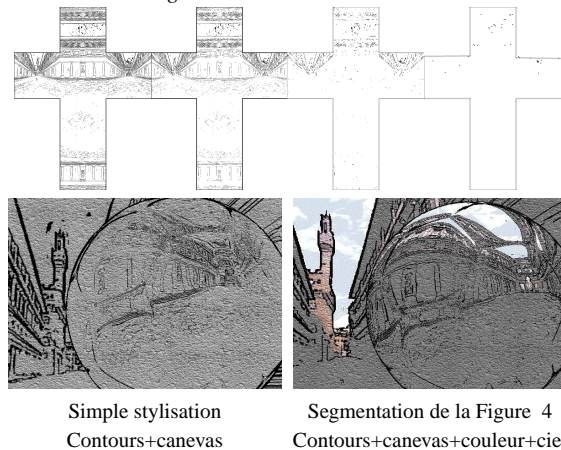


FIG. 6: Style basé sur une détection des contours. Les images du haut représentent les différents niveaux de détail des lignes. L'image en bas à gauche est un rendu utilisant un canevas. Il est possible d'améliorer ce rendu en utilisant la segmentation de la Figure 4 (image en bas à droite).