

# Peinture Virtuelle : modélisation et interaction

Anais Atencia, Jean-Jacques Bourdin

Laboratoire d'Intelligence Artificielle

Université Paris 8

2, rue de la Liberté

93526 Saint-Denis Cedex, FRANCE

anais@ai.univ-paris8.fr, jj@ai.univ-paris8.fr

**Résumé :** *Nous proposons un modèle de peinture virtuelle intégrant non seulement la simulation physique mais aussi le geste du peintre et son positionnement. Ce modèle inclut les déplacements (aller-retour) indispensables à la peinture impressionniste. Nous verrons que les effets de zooms liés à des capteurs améliorent l'interactivité du modèle.*

**Mots-clés :** rendu non photoréaliste, peinture virtuelle, haute résolution, palette graphique.

## 1 Introduction

Un des axes de recherche en rendu non photoréaliste (NPR) [GG01] est de simuler les styles de peinture en s'inspirant des effets des peintures impressionnistes [Hae90, Her98, Mei96, Lit97]. De nombreux travaux de peinture virtuelle ne modélisent qu'un effet comme par exemple la peinture à l'eau [CAS<sup>+</sup>97]. Nous présentons un modèle de peinture virtuelle plus général qui permet d'intégrer les différents arts graphiques : huile, aquarelle, encre, graphite. . . Nous utiliserons néanmoins le terme générique de peinture car il correspond à notre exemple récurrent et au mode le plus général. L'application intègre aussi la simulation de l'action de peindre. Peindre, c'est une succession de gestes fins et mesurés, de grands coups de brosses [Her98], de déplacements et de changements de perspectives et d'éclairage. Nous recréons le mouvement de recul que peut avoir un peintre pour visualiser son œuvre. Ce mouvement est modélisé à l'aide d'une succession de zooms arrière ou avant. Ces zooms sont commandés via une interface basée sur la position et les mouvements de l'utilisateur. Notre but est que l'utilisateur puisse visualiser l'image et travailler de la même manière que dans son atelier. En utilisant une tablette *wacom* avec visualiation (de type cintiq [Wac]), de même que sur la toile, il peint directement sur l'image. Les différentes étapes de ce travail sont donc : une simulation physique de la peinture, un système de changement de point de vue et une interface. Nous présenterons dans un premier temps le modèle physique, puis l'interface homme machine et enfin nous détaillerons les techniques utilisées pour la détection de la position de l'utilisateur et pour les zooms.

## 2 Modèle physique

Dans cette partie, nous détaillons le modèle 3D de peinture créé par Sobczyk et al. [SBB02b] que nous utilisons. Ce modèle de peinture virtuelle utilise des images en très haute résolution [SBB02a, Cai00]. Il s'inspire des méthodes de Curtis [CAS<sup>+</sup>97] et Buchanan [SB99a, SB99b, SB99c, SB00]. Fredo Durand [Dur02] a noté que la peinture virtuelle était plus qu'une simple projection 3D d'une image 2D. La peinture a une épaisseur. Le modèle de Sobczyk prend en compte trois éléments de la peinture : le support, les instruments et le médium.

- Le **support** est l'objet sur lequel on peint. Nous avons défini différents supports utilisés par un peintre : le bois, le métal, le papier ou la toile. À chaque support correspond plusieurs caractéristiques comme la couleur et le taux d'absorption de la peinture. Chez Sobczyk, le support est représenté comme un objet 3D. Il est constitué de cellules. Une scène comporte  $5 \times 10^7$  cellules.
- Le modèle gère les différents **instruments** utilisés par un peintre comme le pinceau ou le couteau. À chaque outil on applique une direction, une force et une quantité de peinture.
- Le **médium** correspond à ce qui est appliqué : charbon du crayon, pâte à peinture, vernis, encre. . . Il est défini par une couleur et un facteur de viscosité.

La palette prend en compte l'interaction entre le support et le médium. En fonction de la force appliquée aux pinceaux, le support peut être altéré. Sur le support, nous appliquons plusieurs couches de peinture. Ces différentes couches augmentent l'épaisseur de la cellule. Chaque cellule est donc caractérisée par une épaisseur variable.

Le tableau est alors un paysage et est codé comme un objet *OpenGL*. Nous pouvons visualiser la peinture selon différents point de vue ou avec différents éclairages.

Pour passer d'un détail à l'image dans son ensemble, nous avons besoin d'outils et de techniques spécifiques.

### 3 Interface homme machine

Une fois que la toile virtuelle est créée comme un paysage 3D nous devons fournir les outils permettant de la visualiser en prenant en compte le positionnement et les mouvements de l'utilisateur.

Nous avons choisi un système non intrusif c'est-à-dire un système qui laisse l'utilisateur libre de ses mouvements et ne nécessite pas de porter sur la tête le matériel de capture [Col99]. Il est en effet difficile de réaliser des gestes naturels et spontanés si l'on doit porter un équipement pour les mesures. Un système non intrusif permet de laisser l'utilisateur libre de ses mouvements. Il est aussi important que l'utilisateur ne soit pas gêné dans son activité par du bruit ou les éventuels déplacements de l'outil de capture.

Notre but est que l'utilisateur puisse visualiser l'image de la même manière qu'il regarderait un tableau. Plus la peinture est éloignée de l'observateur et plus les contours s'estompent. Par exemple, lorsqu'on regarde un tableau impressionniste en prenant du recul, les contours n'apparaissent plus et ce qui ne semblait n'être que des taches de rouge et de bleu devient un violet. Avec l'éloignement, le cerveau opère la synthèse et il restitue les couleurs qui sont juxtaposées sur la toile selon les lois optiques de leur complémentarité.

Les premiers visiteurs des expositions impressionnistes mettaient le nez sur les tableaux, comme ils le faisaient pour les peintures miniatures de Meissonier [Gom90]. Ils ne pouvaient percevoir qu'un mélange confus de couleurs posées au hasard. Il faudra un peu de temps pour que le public comprenne que pour apprécier une peinture impressionniste il est nécessaire de prendre du recul. De loin, un tableau comme [Sis73] est quasiment photoréaliste. Sur la même idée, avec notre palette l'utilisateur doit pouvoir passer de manière très simple de l'image dans son ensemble à un détail de cette image. Par exemple dans *le Bal du Moulin de la Galette de Renoir* [Ren76], l'utilisateur doit pouvoir passer du tableau dans son ensemble au couple au fond sur le banc. L'utilisateur pourra alors se demander si ce couple est en phase de dispute ou de séduction.

Les images sur lesquelles nous travaillons sont en très haute résolution (30 pixels représentant 1 millimètre) et en très haute définition (6000×9000) selon les souhaits exprimés à Eurographics [ABB<sup>+</sup>01]. On ne peut donc visualiser sur un écran d'ordinateur qu'une partie de l'image. Pour pouvoir visualiser les images très haute définition dans leur ensemble sur l'écran, on applique des zooms arrière sur une partie ou sur l'ensemble de l'image.

Nous avons besoin d'une interface simple et intuitive pour utiliser les zooms. L'un des moyens les plus naturels est de retranscrire le mouvement de l'utilisateur devant la tablette. En fonction du mouvement avant arrière et de la distance parcourue par l'utilisateur, on applique un zoom.

Le principe de cette interface est :

- quand l'utilisateur s'approche de l'écran, on effectue un zoom avant sur l'image.
- quand l'utilisateur recule, on effectue un zoom arrière sur l'image.

Nous allons maintenant présenter cette interface.

## 4 Technique de zoom

Le système de zoom est composé de deux parties : la mesure de distance et le calcul du zoom adéquat. C'est dans cet ordre que nous traiterons ces deux parties.

### 4.1 Détection de la position

Nous utilisons des télémètres à ultrasons pour déterminer la position de l'utilisateur. Ces télémètres sont basés sur la mesure du temps écoulé entre l'émission d'un ultrason et le retour de l'écho. L'onde ultrason se propage à la vitesse du son. Dès qu'un obstacle est rencontré, l'écho revient. Le chronomètre s'interrompt à la réception du signal. L'utilisation des télémètres à ultrasons pose certains problèmes :

- précision. Les mesures ne sont pas très précises.

- interférence. Si l'utilisateur fait des gestes brusques ou passe sa main devant son visage, les mesures sont parasitées.
- obstacles. Quand nous mesurons des grandes distances, rien ne doit se trouver entre les télémètres et l'utilisateur.
- minima. Nous ne pouvons détecter précisément les positions très proches de l'écran car les télémètres ne vont pas mesurer la position des yeux mais celle du front.

Nous avons positionné ces télémètres sur la partie supérieure de la tablette. Afin de s'affranchir des perturbations extérieures, nous lançons une série de cinq mesures sur deux télémètres puis nous effectuons un calcul de moyenne. Même ainsi le problème du minima reste posé, nous effectuons donc un calibrage. L'utilisateur détermine les positions "proche" et "loin" et il calibre ses propres positions "proche" et "loin". Ensuite on calcule l'image à afficher en fonction du calibrage de l'image, du calibrage de la position de l'utilisateur, du mouvement et de la distance parcourue par l'utilisateur. Le ratio de distance obtenue sert de valeur d'entrée du zoom. Dans un premier temps, en considérant la scène à afficher, nous avons utilisé les variations de point de vue d'*OpenGL*.

Pour l'instant, notre interface a seulement besoin de connaître le positionnement de l'utilisateur par rapport à l'écran. Mais nous envisageons de mettre en place des systèmes de capture du regard à l'aide d'une caméra [SD02, SYW96] pour traiter seulement la partie de l'image que l'utilisateur regarde.



FIG. 1 – Zoom *aléatoire* sur un détail du Bal du Moulin de la Galette

## 4.2 Zooms

Les applications de zooms déjà existantes sont essentiellement basées sur un calcul de moyenne. Elles ont un système d'antialiassage qui semble préjudiciable au traitement qu'on effectue sur les images. Par exemple, un lissage sur l'Église d'Auvers sur Oise de Van Gogh [Gog90] dénature la toile. Dans ce tableau, le rapprochement des couleurs vert et bleu pour l'herbe, bleu vif et noir pour le ciel, bleu et rouge pour le toit créent des effets de saturation et de stridence générateurs de malaise. Avec un zoom qui lisse cette image, le contraste des couleurs est atténué et la violence qui traverse la toile originale disparaît, les couleurs apparaissent uniformes, l'étirement nerveux de la touche est estompé. Pour éviter cela, nous avons écrit nos propres zooms. Ces zooms prennent en compte les caractéristiques de différents modèles de couleurs (HSV, RGB...)

Principe de ces zooms arrière : un zoom arrière 4 représente la division par 4 des dimensions de l'image.



FIG. 2 – Zoom *moyenne* sur un détail du Bal du Moulin de la Galette

Soient  $I_O$  l'image d'origine et  $I_Z$  l'image zoomée. Le zoom  $z$  est une transformation :

$$I_Z = z(I_O)$$

16 pixels de  $I_O$  sont représentés par 1 pixel de  $I_Z$ . Pour chaque groupe de  $4 \times 4$  pixels de  $I_O$ , sont appliqués des algorithmes pour obtenir la valeur la plus représentative. Pour le moment, nous appliquons ces algorithmes sur des matrices disjointes.

Nous avons classé nos zooms en catégorie :

- les zooms qui sélectionnent les données de manière arbitraire. En prenant la valeur centrale, une valeur aléatoire ou une valeur différente pour chaque matrice. Avec ces zooms, l'image a un aspect flou et granuleux (cf. image 1).
- les zooms qui recherchent des valeurs communes. On calcule la valeur moyenne, la valeur médiane qui partagent la matrice en deux parties d'égale fréquence ou la valeur la plus fréquente présente dans une matrice. Les images resultantes de ces zooms sont moins contrastées que les images originales (cf. image 2).
- les zooms qui sélectionnent les valeurs extrêmes. On garde la valeur maximum, la valeur minimum ou on calcule la moyenne extrême pour chaque matrice. Ces zooms modifient les couleurs. En prenant les valeurs maximums, on augmente le contraste (cf. image 3).

On choisit ces différents zooms en fonction des caractéristiques qu'on veut faire ressortir sur l'image. En prenant les valeurs moyennes de l'image, nous diminuons le contraste alors qu'en sélectionnant les données extrêmes nous modifions les couleurs. Par exemple pour l'Église d'Auvers sur Oise de Van Gogh, nous utiliserons un zoom qui renforce l'opposition des couleurs. Nous sélectionnons dans le système de couleur HSV, les valeurs moyennes de H et les valeurs maximums de S et de V. L'image 4 présente le résultat ce zoom appliqué sur un détail du tableau de l'Église d'Auvers sur Oise de Van Gogh. La version papier de cet article étant imprimée en noir et blanc, le lecteur trouvera la version couleur des images en [Ate02].

## 5 Conclusion

L'utilisateur de notre système peut appliquer ses touches de peinture au plus près et en s'éloignant de l'écran obtenir une vue globale du tableau. Nous avons vu comment est détectée la position de l'utilisateur à l'aide de télémètres



FIG. 3 – Zoom *maximum* sur un détail du Bal du Moulin de la Galette

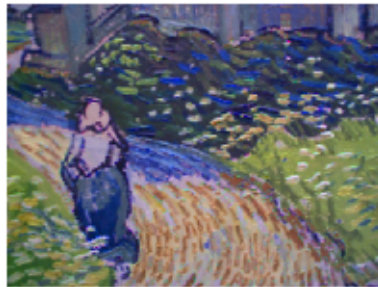


FIG. 4 – Zoom sur un détail de l'Eglise d'Auvers sur Oise

à ultrasons et de quelle manière sont appliqués les zooms. Les télémètres à ultrasons ne sont pas très précis et comportent de nombreux inconvénients. Nous allons prochainement tester notre système avec des télémètres infrarouges pour avoir des mesures plus précises. Nous travaillons sur la mise en place de zooms en 3D qui permettront de visualiser le volume de l'image. Un système de capture du regard doit également être expérimenté. Ce système nous permettra d'appliquer des zooms sur la partie regardée par l'utilisateur. Un de nos axes de recherche est d'améliorer la visualisation des images, en modélisant les effets de craquelure ou de vernis que nous percevons lorsque nous regardons une toile.

## Références

- [ABB<sup>+</sup>01] A. Atencia, J.-J. Bourdin, V. Boyer, T. Pissard, and D. Sobczyk. Scalable impressionist rendering. In *Eurographics 2001*, 2001.
- [Ate02] Anaïs Atencia. Page web, 2002.  
<http://www.ai.univ-paris8.fr/anais>.

- [Cai00] R. Caillou. Principles of computer graphics : the experience of a class a user. *Proceedings of EUROGRAPHICS'00*, September 2000.
- [CAS+97] C.J. Curtis, S.E. Anderson, J.E. Seim, K.W. Fleischer, and D.H. Salesin. Computer-generated watercolor. *Proceedings of SIGGRAPH'97*, pages 421–430, August 1997.
- [Col99] C. Collet. Capture et suivi du regard par un système de vision dans le contexte de la communication homme-machine. Master's thesis, Ecole Normale Supérieure de Cachan, September 1999.
- [Dur02] F. Durand. An invitation to discuss computer depiction. *NPAP 2002 : Second International Symposium on Non-Photorealistic Animation and Rendering*, June 2002.
- [GG01] B. Gooch and A. Gooch. *Non-Photorealistic Rendering*. A K Peters, 2001.
- [Gog90] Vincent Van Gogh. L'Église d'Auvers sur Oise, 1890.  
<http://www.abcgallery.com/V/vangogh/vangogh43.html>.
- [Gom90] E. Gombrich. *Histoire de l'art*. Flammarion, 1990.
- [Hae90] Paul E. Haeberli. Paint by numbers : Abstract image representations. *Proceedings of SIGGRAPH'90*, August 1990.
- [Her98] A. Hertzmann. Painterly rendering with curved brush strokes of multiples sizes. *Proceedings of SIGGRAPH'98*, July 1998.
- [Lit97] P. Litwinowicz. Processing images and video for an impressionist effect. *Proceedings of SIGGRAPH'97*, pages 407–414, August 1997.
- [Mei96] B. Meier. Painterly rendering for animation. *Proceedings of SIGGRAPH'96*, pages 477–484, August 1996.
- [Ren76] Pierre-Auguste Renoir. Bal du Moulin de la galette, 1876.  
<http://www.abcgallery.com/R/renoir/renoir86.html>.
- [SB99a] M.C. Sousa and J.W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. *Proceedings of EUROGRAPHICS'99*, September 1999.
- [SB99b] M.C. Sousa and J.W. Buchanan. Computer-generated pencil drawing. *Western Computer Graphics Symposium*, March 1999.
- [SB99c] M.C. Sousa and J.W. Buchanan. Observational model of blenders and erasers in computer-generated pencil rendering. *Proceedings of Graphics Interface*, June 1999.
- [SB00] M.C. Sousa and J.W. Buchanan. Observational model of graphite pencil materials. *Computer Graphics Forum*, 19(1), March 2000.
- [SBB02a] D. Sobczyk, V. Boyer, and J.-J. Bourdin. Impressionist rendering, a high resolution approach. *3DD : International Workshop on 3D Digitization*, February 2002.
- [SBB02b] D. Sobczyk, V. Boyer, and J.-J. Bourdin. Virtual painting : Model and results. *ICCVG'02*, September 2002.
- [SD02] A. Santella and D. DeCarlo. Abstracted painterly renderings using eye-tracking data. *NPAP 2002 : Second International Symposium on Non-Photorealistic Animation and Rendering*, pages 75–82, February 2002.
- [Sis73] A. Sisley. Chemin de la Machine, Louveciennes, 1873.  
<http://www.abcgallery.com/S/sisley/sisley14.html>.
- [SYW96] R. Stiefelhagen, J. Yang, and A. Waibel. A model based gaze tracking system. *Proc. of IEEE International Joint Symposia on Intelligence and Systems*, pages 304–310, November 1996.
- [Wac] Wacom.  
<http://www.wacom.com>.