

# Texture Mapping for Cel Animation

Kevin Bollini

9 novembre 2012

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Contexte</b>	<b>3</b>
<b>3</b>	<b>Problématiques et objectifs</b>	<b>3</b>
3.1	Problématiques . . . . .	3
3.2	Objectifs . . . . .	3
<b>4</b>	<b>Méthode</b>	<b>4</b>
4.1	Spécification de Marqueurs . . . . .	4
4.1.1	Détection de silhouette . . . . .	4
4.1.2	Curve fitting . . . . .	4
4.1.3	Marqueurs du Dessin "Manuel" . . . . .	4
4.2	"Warp" . . . . .	5
4.2.1	Warp avec une paire de marqueurs . . . . .	5
4.2.2	Warp avec de multiples paire de marqueurs . . . . .	5
4.3	Contrôle du Warp . . . . .	5
4.3.1	Paramètres . . . . .	5
4.3.2	Modelisation et Caméra . . . . .	5
4.3.3	Ajout de marqueurs . . . . .	5
<b>5</b>	<b>Résultats et Limites</b>	<b>6</b>
5.1	Résultats . . . . .	6
5.2	Limites . . . . .	7
<b>6</b>	<b>Discussion</b>	<b>7</b>

# 1 Introduction

Ce commentaire portera sur la publication "Texture Mapping for Cel Animation", papier de 1998, par Wagner Toledo Corrêa, Robert J. Jensen, Craig E. Thayer, Adam Finkelstein pour l'Université de Princeton et Walt Disney feature Animation. Il a pour but d'exposer une technique permettant l'utilisation de dessin manuel et de modélisation 3D pour de l'animation 2D, par déformation d'un modèle 3D vers la forme d'un croquis manuel.

## 2 Contexte

La publication de cet article survient en 1998, à une période de grand progrès de l'animation 3D, on peut citer le film d'animation Toy Story en 1995 par exemple. Cependant, à l'époque de l'article, l'animation 2D par la modélisation 3D est une pratique marginale, non appréciée des artistes qui ne veulent pas perdre du temps à animer de cette façon. De plus, l'industrie de l'animation 2D "manuelle" est forte de 80 ans d'expérience alors que l'animation 3D n'en est qu'à ses balbutiements.

## 3 Problématiques et objectifs

Dans le contexte de l'article, l'utilisation de 3D dans l'animation 2D est marginale. Le problème étant que les artistes ne veulent pas perdre beaucoup de temps dans la modélisation 3D alors qu'ils peuvent faire plus rapidement à la main, tout en pouvant faire passer plus d'émotion.

### 3.1 Problématiques

L'animation 3D pose plusieurs problèmes, le temps de modélisation, la difficulté de la tâche et la qualité du rendu. En effet, il est clairement plus rapide de créer des formes à la main. D'autre part, un des problèmes de l'animation "manuelle" est celle d'objets fortement texturés, comme par exemple le tapis d'Aladdin. La texture se déforme d'image en image, le dessin et la déformation de la texture étant complexes. Cela cause alors un effet de "flottement". Ce même tapis pour le long métrage "Aladdin" en 1992, a été une tâche dantesque en terme d'animation, nécessitant un "rotoscoping" méticuleux pour chaque frame. Pour rappel, le rotoscoping est une méthode d'animation se basant sur l'utilisation d'images réelles. Ce problème ne se pose pas avec la modélisation 3D, la texture étant alors toujours strictement la même.

### 3.2 Objectifs

Voici alors les objectifs soulignés par les chercheurs :

- Simplifier la création animation 3D
- Exploiter un dessin manuel pour créer un modèle 3D
- Améliorer le rendu des textures entre chaque frames

## 4 Méthode

La méthode proposée se déroule en plusieurs étapes :

- Création et numérisation d'un dessin
- Modélisation 3D approximative du dessin à "main levée"
- Détection de Marqueurs
- "Warp" : déformation, ondulation du modèle approximatif vers le dessin manuel
- Rendu final de l'objet

Il faut donc dessiner d'abord, à la façon d'un simple croquis, l'entité que l'on souhaite créer, il faut ensuite modéliser de façon simple, position et forme, de l'objet. On déduit ensuite des marqueurs sur le dessin 2D et le modèle 3D afin de pouvoir "warper" le modèle 3D vers la position et déformation du croquis. Il ne restera plus qu'à texturer et à faire le rendu de l'objet pour obtenir une frame.

### 4.1 Spécification de Marqueurs

Cette étape permet de créer des "marqueurs" sur le modèle 3D et sur l'ébauche. Ces marqueurs, des courbes paramétriques des contours des objets, sont indispensables à l'étape suivante du pipeline, à savoir, le morphing de l'entité 3D vers la forme du croquis.

#### 4.1.1 Détection de silhouette

A cet étape, il faut extraire du modèle 3D ses contours. Cette extraction est basée sur une image, l'uv-image, obtenue par rendu de la scène 3D par ombrage Gouraud. Cette image est colorée en  $(R,V,B) = (u,v,ID)$ , où  $u$  et  $v$  sont les coordonnées paramétrique de chaque sommet du modèle et  $ID$  l'identifiant de la texture. A partir de cet uv-image, on construit un graphe orienté où chaque sommet est un coin de pixel et chaque arrêtes une frontière entre deux pixels de couleur différente. On obtient alors un graphe permettant la déduction de contours du modèle, il suffit de trouver les composantes connexes du graphe, par un parcours en profondeur par exemple.

#### 4.1.2 Curve fitting

Il faut exploiter les données du graphe précédement calculées pour pouvoir en déduire les marqueurs. Les données du graphe permettent d'obtenir des chaînes de pixels, données trop lourdes à traiter, il faut alors les approximer par une courbe B-spline. En effet, on peut représenter les données par une spline avec un minimum de point de contrôle, les données seront plus légères, en taille et en difficulté de traitement. La méthode utilisée est donc de calculer une spline d'un minimum de point de contrôle et d'erreur avec les données originales.

#### 4.1.3 Marqueurs du Dessin "Manuel"

Ici, principalement pour éviter plus d'intervention de l'utilisateur, les auteurs utilisent des méthode existantes (Gleicher et Mortensen) pour extraire les contours du dessin, on obtient alors un jeu de données comparable à celui obtenu après la détection de contours du modèle 3D, et on peut donc en déduire de la même façon une courbe paramétrique.

## 4.2 "Warp"

Cette étape permet le gondolement, la déformation de l'entité 3D vers la forme du croquis à l'aide des marqueurs de la façon suivante :

### 4.2.1 Warp avec une paire de marqueurs

Cette opération est basée sur le calcul d'une équation, appelée  $q(t)$ , permettant de déterminer l'image d'un point  $p(t)$  du modèle 3D vers sa position "warpée". Seulement,  $q(t)$  d'obtenir l'image en ne considérant rien d'autre que le paramètre  $t$ . Il faut donc pondérer, on utilise alors  $q$ , une moyenne pondérée de  $q(t)$ . Cette équation est pondérée d'une fonction  $c(t)$ , permettant de régler l'intensité de la déformation en fonction de la distance,  $d(t)$ . On voit donc que l'on recalcule ici la position de chaque pixel du modèle 3D en mappant leur position sur le modèle 3D avec celle sur le dessin.

### 4.2.2 Warp avec de multiples paire de marqueurs

On généralise ensuite cette méthode pour le warp avec plusieurs marqueurs, la déformation n'étant pas forcément, et même rarement due qu'à un seul marqueur. Ici, le calcul est assez simple, on utilise les différents marqueurs pour définir leurs fonctions  $q_i(t)$  associées, avec  $i$  allant de 1 à  $m$ , le nombre de marqueurs.  $q$  devient donc la moyenne pondérée des sommes des équations  $q_i(t)$  de chaque marqueur, pondérées par un poids  $w_i$  et par leurs fonctions  $c_i(t)$ . Ces poids sont spécifiés par l'utilisateur. On obtient alors un modèle 3D déformé correctement.

## 4.3 Contrôle du Warp

Maintenant que l'on sait comment "warper" notre modèle, nous pouvons regarder les divers paramètres pouvant influencer sur le comportement de cette déformation.

### 4.3.1 Paramètres

On peut faire varier la fonction  $c(t)$  en modifiant la valeur d'une variable de "smoothness" et d'une variable de vitesse d'atténuation en fonction de la distance.

### 4.3.2 Modelisation et Caméra

Le warp change selon le modèle 3D de base réalisé et la position de la caméra. En effet, la déformation de la texture ne sera pas la même selon comment on regarde l'objet.

### 4.3.3 Ajout de marqueurs

On peut ajouter des marqueurs supplémentaires pour ajouter du détail à la déformation ne pouvant être défini que grâce aux marqueurs de contours. Par exemple, si l'objet est déformé en son centre, on aurait besoin d'ajouter un marqueur à cet endroit.

## 5 Résultats et Limites

### 5.1 Résultats

L'objectif posé en introduction est atteint, ce protocole permet en effet de créer un modèle 3D, texturé et déformé correctement, par l'utilisation d'un dessin au crayon, explicitant la forme que l'on souhaite donner à notre objet. On peut voir par l'exemple ci-dessous une illustration de la méthode, ici appliquée à l'animation d'une balle rebondissant sur le sol et se déformant au fur et à mesure du temps.

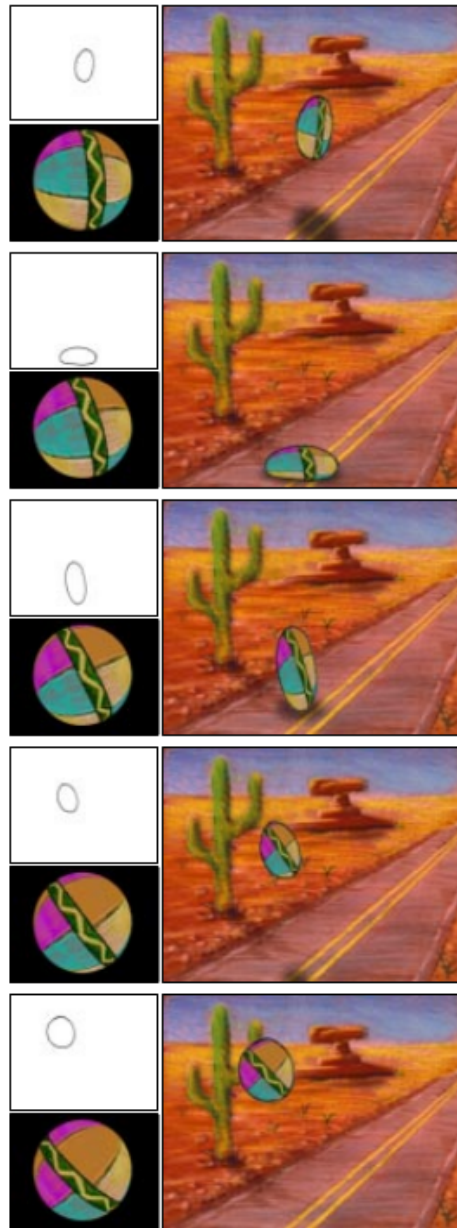


Figure 11: Bouncing ball animation.

On constate donc ici l'efficacité de la méthode développée. On voit bien que la déformation de la balle, et surtout de sa texture, est convaincante à l'oeil. D'autre part, la forme donnée à la balle est une forme qui est objectivement plus complexe à modéliser en 3D que celle d'une balle parfaitement ronde, on peut donc aussi en conclure qu'il est bien plus simple d'utiliser cette méthode pour l'animation de la balle plutôt que d'essayer de modéliser directement en 3D la balle déformée.

## 5.2 Limites

Cette méthode connaît tout de même des limites sur son application à certains types d'objets. Tout d'abord, les fourrures ou cheveux, ces items sont souvent dessinés à la main par des séries de petits traits et non par des lignes franches, ce qui rend leur interprétation complexe par l'algorithme.

Ce problème se pose aussi pour les textiles, dessinés en représentant souvent des plis et courbures complexes à modeler en 3D.

D'autre part, la méthode sera difficile à appliquer à des formes géométriques non logiques, ne pouvant donc pas être modélisées correctement en 3D. Prenons par exemple le croquis d'un visage de face, où le nez serait dessiné de profil, ce croquis n'aurait pas de "bonne" modélisation, l'application de la méthode en deviendrait alors caduque.

Aussi, les objets en plusieurs morceaux, tel un personnage constitué d'un corps et d'un tête, ne serait pas non plus un objet adapté à ce protocole. En effet, il n'est pas possible d'appliquer ceci à des objets divisés, ceux ci ne pouvant pas raisonnablement être modélisés en un seul objet 3D. Il faudrait mieux traiter chaque morceaux indépendamment, mais ce n'est ni pratique, ni optimal.

## 6 Discussion

L'idée développée ici par les chercheurs cherche à allier les avantages du dessin manuel, à savoir, la simplicité, aux avantages de la modélisation 3D. Je pense que le résultat obtenu est assez intéressant comme on a pu le voir sur l'exemple précédent, mais il peut être amélioré. En effet, il reste des interactions humaines dans le processus qu'il serait avantageux de pouvoir supprimer. Principalement, il faudrait pouvoir éviter de devoir modéliser à la main un premier modèle 3D pour chaque frame, c'est une étape très chronophage, qui limite grandement l'intérêt de la méthode décrite ici. Aussi, l'algorithme est limité à des objets simples et monopatches, ce qui est bien dommage, car dans l'animation on cherche souvent à animer des personnages, chose ici impossible.

Ceci étant, Le résultat est tout de même très probant, la déformation des textures étant visuellement réaliste, tout en supprimant le problème de flottement qu'on pouvait trouver dans l'animation classique. Même si la méthode reste chronophage, elle est tout de même bien plus efficace, en terme de temps et de résultat visuel, que la méthode utilisée pour le long-métrage aladdin, qui comme je l'ai dit dans l'introduction, avait nécessité un travail rotoscopique d'orfèvre des animateurs.