Une méthode de rendu d'image à partir du modèle de courbe de diffusion[1]

Projet de spécialité ENSIMAG 2009 Filière imagerie

Cédric ZANNI & Maxime QUIBLIER

Sommaire

1 Introduction

Dans le cadre du projet de spécialité de 2nde année, nous avons choisi de travailler sur les courbes de diffusion, un nouvel objet de dessin vectoriel. Adrien Bousseau était notre tuteur pour ce projet. Ayant lui même participé à l'élaboration de l'article [1], il nous a proposé d'analyser une autre méthode de rendu dont il avait eu l'idée. Cette dernière méthode se base sur une technique de construction de fonction par Moving Least Square, présentée en détail dans l'article [2].

Nous devions dans un premier temps tester l'adaptation la plus évidente de la méthode pour effectuer le rendu d'une image décrite par des courbes de diffusion. En cas de résultats non satisfaisants, nous avions carte blanche pour explorer les possibilités offertes. Nous nous étions fixé pour objectif de travailler en priorité sur la qualité du rendu, et éventuellement sur la vitesse de calcul.

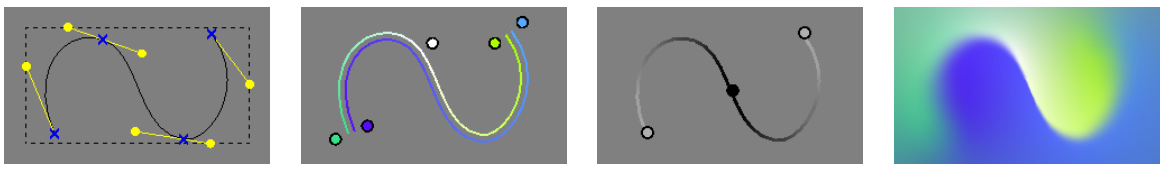
Après une rapide présentation des articles précédemment cités, le lecteur trouvera dans le présent rapport les résultats de l'adaptation directe ainsi que ceux des améliorations que nous avons imaginé et testé. La durée du projet n'était pas suffisante pour explorer toutes les pistes, nous terminons donc ce document par une présentation exhaustive de nos idées pour les travaux futurs.

2 Moving Least Square et courbes de diffusion

2.1 Le modèle des courbes de diffusion

Dans l'article *"Diffusion Curves : a vector representation for smooth-shaded images"*[1], les auteurs présentent un nouvel objet de dessin vectoriel : la courbe de diffusion. Une courbe de diffusion est une courbe de Bézier dont les points de contrôle sont agrémentés d'une couleur gauche, une couleur droite et une valeur de flou. Ces valeurs déterminent respectivement la couleur du plan de dessin d'un coté et de l'autre de la courbe et la manière dont elles se mélangent au niveau de la courbe. Les couleurs et la valeur de flou varient continuement le long d'une courbe entre les points de contrôle où l'utilisateur impose des valeurs.

A partir d'un ensemble de courbes de diffusion, l'image finale est calculée en résolvant une équation de Poisson. Les contraintes de cette équation sont données par l'ensemble des contraintes de couleurs et de flou sur les courbes de diffusion. Le principal intérêt réside dans les variations progressives de couleurs qu'engendre le modèle. Tous les avantages classiques de la représentation vectorielle restent valables (indépendance vis-à-vis de la résolution, aide à l'animation...).



(a) (b) (c) (d)

FIGURE 1 – Structure de donnée et rendu pour la courbe de diffusion : (a) une courbe de Bézier, (b) des couleurs et (c) des valeurs de flou définies arbitrairement et variant linéairement le long de la courbe, (d) image finale

Le système de dessin est intuitif pour les artistes et les résultats obtenus sont assez convain-

quant comme le montre la Figure 1. De plus, l'implémentation sur carte graphique proposée permet une efficacité de calcul suffisante pour avoir un rendu temps-réel.

Cependant, certains points demanderaient à être améliorés :

- Le flou est calculé indépendamment de l'image nette et demande un temps de calcul conséquent malgré la qualité de l'implémentation
- Même si ce modèle est un des plus flexible existant, l'utilisateur aimerait profiter d'un peu plus de contrôle, par exemple sur les dégradés entre deux courbes distinctes, ou sur les directions de diffusion.

2.2 La méthode d'approximation par Moving Least Square

Dans l'article *"'Interpolating and approximating implicit surfaces from polygon soup"*, les auteurs présentent une méthode pour construire une surface implicite interpolant ou approximant un modèle polygonal. Pour construire la fonction implicite nécessaire à la définition de la surface, ils utilisent une méthode appelée "Moving Least Square" (MLS).

Soit N points P_i, $i \in [\![1; N]\!]$, l'objectif est de construire une fonction $f(\mathbf{x})$ approchant les valeurs ϕ_i en ces points. Une approximation classique par les moindres carrés revient à résoudre le système :

$$\begin{bmatrix} b^T(P_1) \\ \vdots \\ b^T(P_N) \end{bmatrix} \mathbf{c} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

- $b(\mathbf{x})$ est le vecteur de fonctions de bases. Par exemple $b(\mathbf{x}) = [1, x, y, z]$ pour approcher un plan, $b(\mathbf{x}) = [1]$ pour approcher une fonction constante.
- $\mathbf{c}$ est le vecteur de coefficients cherché.

Dans la formulation des MLS, l'approximation change en fonction du point $\mathbf{x}$ où est évaluée f de sorte que $\mathbf{c}$ varie avec $\mathbf{x}$. Pour cela, une fonction $w(\mathbf{x}, P_i)$ est utilisée pour donner un poids variable à chaque ligne :

$$\begin{bmatrix} w(\mathbf{x}, P_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, P_N) \end{bmatrix} \begin{bmatrix} b^T(P_1) \\ \vdots \\ b^T(P_N) \end{bmatrix} \mathbf{c} = \begin{bmatrix} w(\mathbf{x}, P_1) & & \\ & \ddots & \\ & & w(\mathbf{x}, P_N) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

En donnant des noms aux matrices, l'équation précédente devient :

$$W(\mathbf{x})\ B\ c(\mathbf{x}) = W(\mathbf{x})\ \phi$$

D'où :

$$B^T\ (W(\mathbf{x}))^2\ B\ c(\mathbf{x}) = B^T\ (W(\mathbf{x}))^2\ \phi$$

Finalement, il est possible d'évaluer $f(\mathbf{x})$ par la formule :

$$f(\mathbf{x}) = b^T(\mathbf{x})\ H^{-1}\ B^T\ (W(\mathbf{x}))^2\ \phi \qquad \text{avec} \qquad H = B^T\ (W(\mathbf{x}))^2\ B$$

Les auteurs de l'article affirment que d'après leurs tests, une fonction de base constante et la fonction de poids définie par $w(r) = \frac{1}{(r^2+\epsilon^2)}$ donnent de bons résultats.

D'autre part, il est aisé d'étendre les équations matricielles présentées ci-dessus pour obtenir un modèle de contraintes continues.

2.3 Idée de départ et objectifs

A partir du modèle des courbes de diffusion, nous cherchons à obtenir une image complète, c'est-à-dire une fonction de $\mathbb{R}^2$ dans $[0; 255]$ interpolant les couleurs définies au niveau des courbes. Nous souhaitons de plus que cette fonction soit continue partout sauf éventuellement sur les courbes.

Les MLS correspondent justement à une méthode pour construire une fonction de ce type. La fonction de poids pourrait permettre d'ajouter un contrôle sur l'influence relative des courbes, ce que le rendu par l'équation de diffusion ne permet pas.

Le premier objectif de notre travail était donc d'adapter la méthode des MLS et de l'utiliser pour effectuer le rendu de l'image décrite par les courbes de diffusion. L'interêt du modèle réside notamment dans le contrôle donné par la fonction de poids, nous avions donc prévu de faire quelques essais dessus pour améliorer les rendus.

Dans notre cas l'adaptation donne :

- les p_i sont les points de contrainte de couleur. Théoriquement, il devrait y en avoir une infinité : deux pour chaque point d'une courbe (un pour la couleur à droite et un pour la couleur à gauche). Notons que nous sommes obligés de décaler les points pour exprimer séparément les contraintes à gauche et à droite. Ce problème technique était déjà présent dans l'implémentation originale. En pratique, l'implémentation existante approxime les courbes de béziers par des segments. Nous avons donc essayé d'une part un modèle discret avec pour seuls points de contrainte les extrémités de ces segments, et d'autre part un modèle continu en intégrant les contraintes sur ces segments.
- les ϕ_i correspondent aux couleurs à interpoler en chaque point. Dans le cas continu, nous utilisons une variation linéaire le long de chaque segment.
- $b(x)$ est le vecteur de fonctions de base qui donne dans $\mathbb{R}^2$: $b(x) = [1]$ (cas d'une constante) et $b(x) = [1, x, y]$ (cas d'une droite).
- $\omega(x, P_i)$ est la fonction de poids déterminant l'influence relative de P_i en x.

2.4 Résultats de l'adaptation directe

2.4.1 Résultats principaux

L'implémentation utilisant l'approximation par des droites donne des résultats inutilisables. Ceci peut facilement se comprendre en considérant l'interprétation suivante : nous cherchons en dimension 3 un plan d'équation $z = ax+by+d$ tel que ce plan passe par les points de coordonnées (x_{P_i}, y_{P_i}, C_i) où P_i et C_i sont respectivement les coordonnées d'un point de contrainte et sa couleur. Il n'y a alors aucune raison d'obtenir des valeurs de z dans $[0; 255]$.

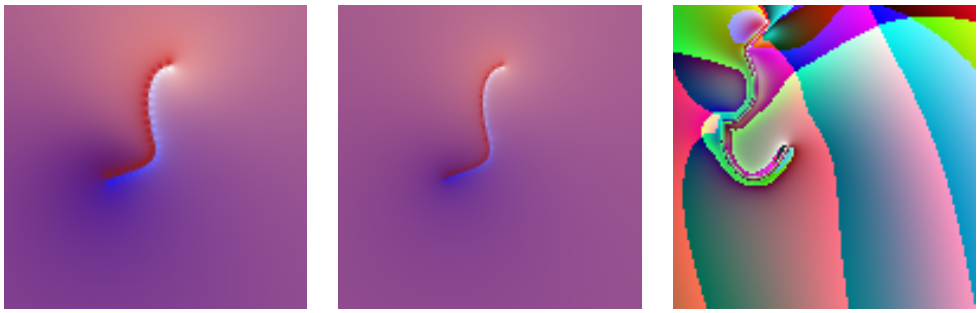
En revanche en utilisant l'approximation par une constante (donc un plan horizontal), nous avons la garantie de rester dans le bon intervalle. Dans ce dernier cas, les résultats sont plus intéressant et l'implémentation matricielle devient obsolète. La simplification des équations mène à une moyenne pondérée classique, beaucoup plus efficace en temps de calcul :

$$color(x) = \frac{\sum_{P_i} \omega^2(x,P_i).\phi_i}{\sum_{P_i} \omega^2(x,P_i)}$$

Au voisinage de la courbe, les couleurs obtenues sont correctes tandis qu'à partir d'une distance relativement faible, les couleurs droite et gauche de contrainte sont totalement mélangées.

Notons que l'utilisation du modèle continu s'avère superflue : le résultat est peu amélioré pour un temps de calcul nettement plus important.

Tous ces résultats sont illustrés par la Figure 2.



(a) (b) (c)

FIGURE 2 – Exemple de résultat de l'implémentation direct. Approximation par constante : (a) modèle discret ; (b) modèle continu ; Approximation par droite : (c) modèle discret

Pour la suite, nous avons donc abandonné l'approximation par des droites et le modèle continu par approximation d'intégrale.

2.4.2 Résultats pour différentes fonctions de poids

Le rendu avec la fonction de poids initial n'étant pas satisfaisant nous avons étudié plusieurs variantes. Ces variantes nous ont permis de localiser plus précisément les configurations à problèmes. La suite de ce rapport montrera que la fonction de poids est primordiale dans le modèle, c'est pourquoi il nous semble intéressant de rendre compte içi des tentatives effectuées.

L'essai qui c'est averé le plus satisfaisant à été l'utilisation d'une fonction de poids dépendant de la position du point par rapport à la normale à la courbe.

Le problème est symétrique. Nous considèrons ici que le point de calcul x est du côté droit de la courbe.
Soit P le point de contrainte dont le poids est recherché, et soit N la normal à la courbe (en ce point). Nous avons :

$$w(x,P,N) = \begin{cases} w_{aux} \ si \ < \vec{N}.\vec{Px} >\geqslant 0 \\ sin(acos(\frac{\vec{N}.\vec{Px}}{\|\vec{Px}\|^2})).w_{aux} \ sinon \end{cases}$$

avec w_{aux} une fonction de poids dépendant uniquement de la distance $\left\|\vec{Px}\right\|$.

Quand le point en cours de calcul est du côté droit de la courbe, l'influence de la contrainte du coté gauche est diminuée, et inversement.

Nous avons effectués des tests sur 3 types de dessins : une seule courbe simple, une courbe fermée et deux courbes simples. Nous avons pu relever les problèmes suivants :

- Toutes les fonctions de poids testées entrainent un mélange des couleurs à "longue" distance et l'atténuation de couleur est trop rapide au voisinnage de la courbe : les couleurs le long

d'une courbe influence donc des deux côtés. Les fonctions les moins affectés par ce problème sont les fonction à décroissance rapide.

- Aucune des fonctions de poids n'est "universelle", au sens où la plupart d'entres elles présentent un problème grave dans au moins un cas particulier. Par exemple, concernant le dégradé entre deux courbes, chacune des deux fonctions suivantes résoud un problème au prix de la création d'un autre tout aussi important.
  - $\omega(r) = \frac{1}{r^2+\epsilon^2}$ ne permet pas un bon dégradé entre les courbes mais donne un résultat satisfaisant en dehors.
  - $\omega(r) = \frac{1}{r+\epsilon^2}$ donne un résultat convenable entre les courbes mais déplorable en dehors. Les couleurs le long d'une courbe influence des deux côtés. Notons de plus que le résultat est catastrophique dans une image plus complexe.
- Un problème récurrent concerne la diffusion d'une couleur en dehors d'une courbe fermée.

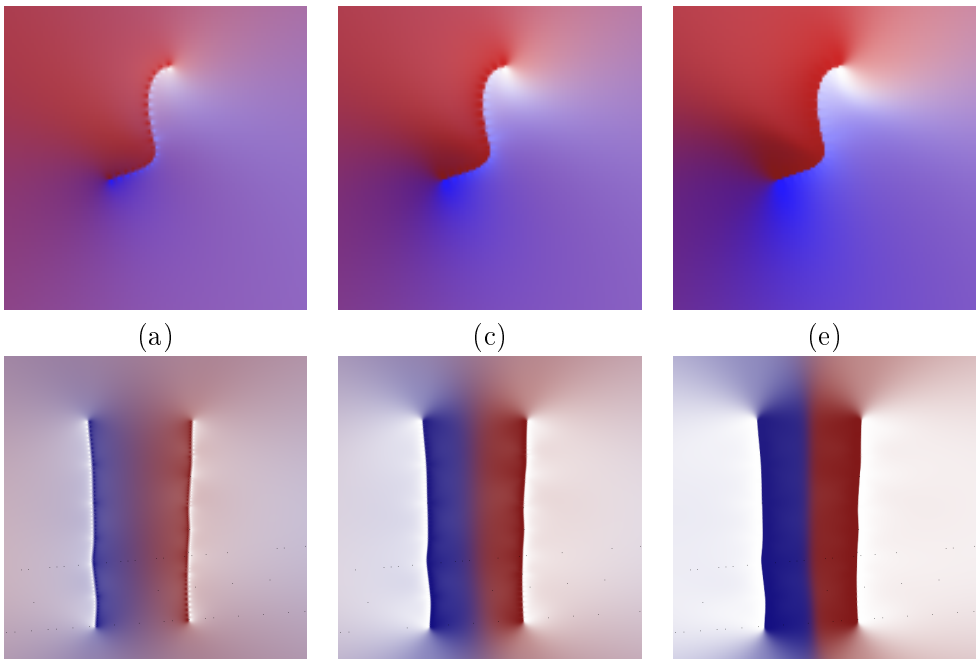


FIGURE 3 – Exemple de travail sur la fonction de poids pour le modèle issu de l'adaptation directe. (a) et (b) : $w_{aux} = \frac{1}{r+\epsilon^2}$, (c) et (d) : $w_{aux} = \frac{1}{r^2+\epsilon^2}$, (e) et (f) : $w_{aux} = \frac{1}{r^4+\epsilon^2}$

2.5 Conclusion

Finalement, nous pouvons noter qu'un travail soutenu sur la fonction de poids donne des résultats intéressant, bien que nettement insuffisant.

Pour rendre ce modèle viable, une réflexion sur cette fonction est donc nécessaire. Or sa forme actuelle rend son design délicat car les effets d'une modification sont difficiles à appréhender par l'intuition.

Nous avons donc eu l'idée de décomposer la fonction de poids, ce nouveau modèle fait l'objet de la section suivante.

3 Modèle dissociant couleur diffusée et zone d'influence

3.1 Expression formelle et motivation

Suite aux différents problèmes rencontrés lors de l'adaptation directe exposée précédemment, nous avons modifié le modèle pour le rendre plus intuitif et plus facile à manipuler.

Dans le modèle précédent, les points gauche et droit sont traités de manière indépendante. Or pour un point donné, ces deux contraintes sont étroitement liées. En effet, les deux points sont théoriquement confondus et il est plus juste de dire qu'il s'agit d'un point diffusant une couleur différente selon la direction.

Notre idée consiste donc à introduire une fonction de couleur déterminant la couleur que chaque point de diffusion veut imposer à un point du plan. Cette couleur dépend logiquement de la couleur à droite, de la couleur à gauche, de la valeur de flou et de la position du point de calcul relativement au point de diffusion. Une fois cette couleur déterminée, la fonction de poids arbitre l'influence de chacune des contraintes.

Dans le but de donner une impression de continuité, nous avons choisi un système adaptatif qui discrétise chaque courbe en une suite de points dont le nombre dépend de sa longueur. Ceci permet d'obtenir un rendu de qualité tout en limitant les calculs.

Finalement, la couleur d'un point x du plan est calculée par la formule suivante (pour chaque canal R,G,B) :

$$color(x) = \frac{\sum_{P_i} \omega(x, P_i)\phi(x, P_i, N_i, Cl_i, Cr_i, Bl)}{\sum_{P_i} w(x, P_i)}$$

où :

P_i sont les points des courbes

N_i est la normale en P_i

Cl_i,Cr_i sont les couleurs respectivement gauche et droite de la courbe en P_i

Bl_i est la valeur de flou en P_i

$\omega(x, P_i)$ est la fonction de poids

Ce modèle divise le nombre de points par deux mais impose de traiter différemment les points aux extrémités.

La fonction de couleur donne un contrôle plus intuitif de la diffusion et permet de dissocier influence de couleur et influence relative. En particulier, la transition au travers de la courbe est gérée par la fonction de couleur ce qui nous laisse plus de liberté sur la fonction de poids.

3.2 Conception de la fonction de couleur

Tout comme la fonction de poids, la fonction de couleur est ici primordiale. Nous avons donc consacré plusieurs jours à sa conception.

Notons que le traitement est symétrique à droite et à gauche de la courbe, nous ne présentons donc ici que les équations sur la moitiée droite, les graphiques montre par contre les 2 cotés.

Pour le travail sur la fonction de couleur, nous nous sommes placés dans le repère $(P, \vec{N}, \vec{T})$ où P est le point de contrainte, $\vec{N}$ et $\vec{T}$ respectivement la normale et la tangeante à la courbe en P. Pour tout point (x, y) dans ce nouveau repère on note $r^2 = x^2 + y^2$ et $\theta = (\vec{N}, (x, y))$.

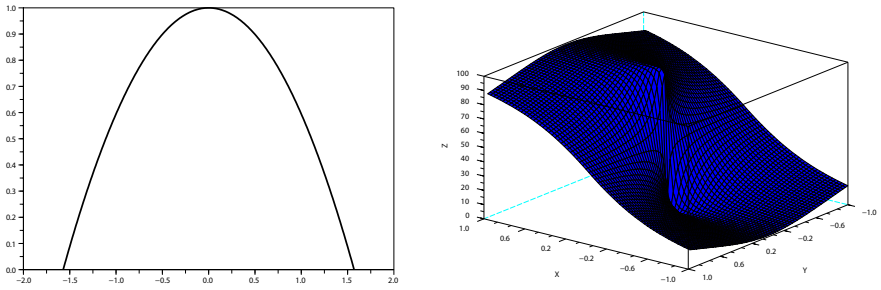
La fonction *step* suivante représente une variation brusque entre les 2 couleurs. Elle donne de bons résutats au voisinage des points de contrainte mais de fortes discontinuités sont visibles.

$$step(x,y) = \begin{cases} Cr \text{ si } x > 0 \\ Cl \text{ sinon} \end{cases}$$

La fonction *varAngle* ci-dessous représente une variation quadratique de la couleur en fonction de l'angle à la normale $\vec{N}$. Elle donne globalement de bons résultats mais est responsable d'un mélange trop rapide des couleurs au voisinage des points de contraintes.

$$\forall x > 0,\ varAngle(x,y) = Cr \times p(\theta) + \tfrac{Cr+Cl}{2} \times (1 - p(\theta))$$

où $p(\theta) = -\frac{4}{\pi^2}(\theta - \frac{\pi}{2})(\theta + \frac{\pi}{2})$



(a) (b)

FIGURE 4 – Graphes des fonctions utilisées : (a) $p(\theta)$ sur $[-\frac{\pi}{2}; \frac{\pi}{2}]$; (b) $varAngle(x,y)$ sur $[-1;1]^2$ avec $Cr = 100, Cl = 0$

Puisque chacune des deux fonctions est intéressantes à une certaine distance, nous construisons une 3ème fonction en les mélangeant : cette fonction utilise *step* comme modèle local et *varAngle* comme modèle à plus longue distance. Pour effectuer la transition entre les deux, nous utilisons $g(x,y) = e^{-150r^2}$.

La fonction mélange est obtenue par la formule suivante :

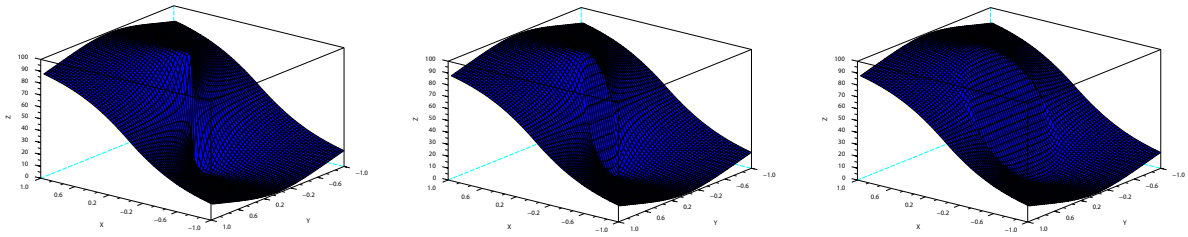
$$fm(x,y) = g(x,y) \times Cr + (1 - g(x,y)) \times varAngle(x,y)$$

Le flou est supposé atténuer les couleurs et les faire tendre vers $\frac{Cr+Cl}{2}$ au voisinage de la courbe. Il est donc tout à fait logique de vouloir utiliser la fonction de couleur pour le gérer. Nous avons utilisé le modèle suivant :

- Bl est un réel positif (non nul pour éviter les singularités). Plus Bl est grand, plus le flou est important
- Si $r > Bl$, $\phi(x,y) = fm(x,y)$
- si $r \leqslant Bl$, $\phi(x,y) = q(x)$ où : q est le polynôme de degré 2 tel que :

$$\begin{cases} q(R_y) = fm(\sqrt{Bl^2 - y^2}, y) \\ q(0) = \frac{Cr+Cl}{2} \\ q'(R_y) = 0 \end{cases}$$

$x \leqslant \sqrt{Bl^2 - y^2} = R_y$ car dans le cas contraire nous aurions $r > Bl$. Ainsi sur la droite $Y = y$, ϕ varie polynomialement par rapport à x de telle sorte que $\phi(R_y, y) = fm(R_y, y)$, ce qui garantit la continuité sur le cerlce de rayon Bl.



(a) (b) (c)

FIGURE 5 – Graphes de la fonction ϕ sur $[-1;1]^2$ avec différentes valeurs de flou : (a) $Bl = 0.01$; (b) $Bl = 0.2$; (c) $Bl = 0.5$

La valeur de flou représente le rayon du disque de centre P sur lequel l'expression de ϕ change.

Nous aurions pu imposer pour le polynôme q la contrainte $Q'(0) = \frac{\partial fm(x,y)}{\partial x}$ pour avoir une fonction plus lisse. Cependant, ceci aurait occasionné des calculs supplémentaires et le rendu actuel nous parait suffisant.

La fonction ϕ laisse apparaitre une faible discontinuité aux extrémités des courbes, nous traitons donc celles-ci de manière différente (abandon de la contrainte local).

3.3 Résultats

Les résultats obtenus sont globalement plus satisfaisants que ceux obtenus par l'implémentation directe.

- Les mélanges de couleur de part et d'autre d'une courbe sont moins important (en abscence de flou).
- Le rendu du flou est correct.

Neanmoins le mélange des couleurs à grande distance est encore présent. De plus, les dégradés ne sont corrects que dans le cas des fonctions à décroissance lente mais ces fonctions sont les plus touchées par le problème précédent. Notons qu'avec cette méthode, le caractère discret des contraintes est moins visible. Avec une discrétisation adaptative dépendante de la longueur de la courbe, il devient impossible de distinguer les contraintes sur l'image.

3.4 Conclusion

Bien qu'il reste de nombreux problèmes, ce modèle apporte plus de flexibilité et augmente les pistes d'amélioration. Il serait sans doute possible d'éliminer la plupart des artefacts visuels en définissant avec soin les fonctions de couleurs et de poids.

Cependant, à cette étape du développement, l'importance du temps de calcul devient un facteur limitant. En effet, la conception des fonctions exige de nombreux tests. Dans notre cas, pour une image 512×512, sur un MacBookPro munit d'un processeur Core2Duo 2.4 Ghz, il faut compter environ 45 secondes pour une courbe simple et 10 minutes pour un dessin tel que le poivron. La complexité du problème est de l'orde de $h \times l \times L$ où h est la largeur de l'image, l la largeur et L la somme des longueurs des courbes qui détermine le nombre de points de contrainte.

Conscients que cette limitation ralentirait les développements futurs, nous avons passé les derniers jours du projet à réfléchir sur les moyens d'accélérer le rendu de l'image. En particulier,

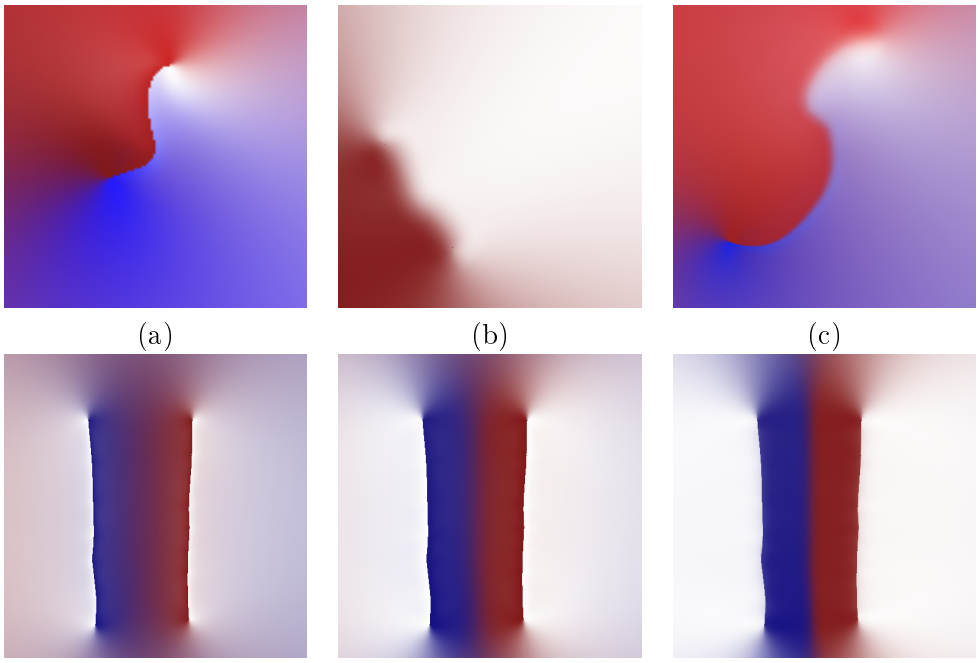


FIGURE 6 – Exemple : fonction de poids utilisée pour les trois premières images : $\frac{1}{r^2+\epsilon^2}$, (a) courbe simple ; (b),(c) exemple de flou (constant le long d'une courbe ou progressif) ; (d),(e),(f) fonction de poids $\frac{1}{r^d+\epsilon^2}$ avec d=1,2,4 montrant les différentes qualités de dégradé obtenu dans le cas de deux courbes

nous avons tenté d'utiliser un maillage. La section suivante présente le rapport sur cette tentative. Les autres idées de solutions envisageables sont présentées dans la section sur les pistes de recherche.

4 Maillage

Le temps de calcul nécéssaire à l'obtention d'une image étant très important et il est possible d'introduire l'utilisation d'un maillage. Le solveur calcule les couleurs des points sur les noeuds du maillage et l'image est ensuite complétée par openGL. Notre implémentation montre que cette méthode présente des problèmes et limites non négligeables.

Tout d'abord il faut créer un maillage adapté au problème. Nous avons choisi d'utiliser le mailleur Triangle développé par Jonathan Richard Shewchuk. Dans notre cas nous fixons comme contrainte en entrée du mailleur le cadre de l'image ainsi que l'ensemble des contraintes de couleur (courbes parallèle et proche des courbes de contrôle).
Les maillages ainsi construis ne sont pas parfaitement adaptés au problème. En effet le mailleur peut construire plusieurs triangles qui couvrirons une aire ne représentant qu'un pixel. Il y a alors perte de temps de calcul et le rendu peut ne pas être localement satisfaisant. De plus le mailleur ne parvient pas à construire les maillages pour les images les plus complexes.

Bien que les maillages ne soit pas totalement adaptés au problème cette méthode semble assez efficace. Certains problèmes restent à régler pour obtenir un rencu aussi bon que celui du calcul pixel par pixel.

Avec ce mailleur non optimal, et suivant la qualité du rendu souhaité, la méthode permet un

FIGURE 7 – Exemple de résultat pour une image plus complexe

gain de temps plus ou moins important. Pour un rendu très grossier le temps peut être divisé par 60, pour un rendu convenable le temps est divisé par 10 (meilleur rendu actuel par la méthode de maillage). Le gain de temps pourrait être plus important avec un maillage adapté au problème. Ces chiffres sont obtenu pour une image ne possédant pas un nombre très élevé de courbes (dans le cas contraire l'efficacité de la méthode sera réduite).

Les résultats obtenus montrent que l'utilisation d'un maillage pour réduire le temps de calcul est une méthode à approfondir.

5 Pistes de recherche

L'implémentation actuelle n'est qu'une première vu sur ce qu'il est possible de faire en utilisant la méthode des "moving least squares"[2] pour calculer le rendu des "diffusions curves"[1]. En effet il est encore possible d'augmenter grandement l'efficacité de la résolution du problème et il est de plus possible d'améliorer la qualité du rendu. Voici les principales pistes que nous avions en vu pour effectuer des améliorations que ce soit du point de vu rendu dans un premier temps, puis du point de vu efficacité de calcul dans un second temps.

Une des pistes principales, et nécessaire à certaines autres est l'introduction d'une fonction de distance donnant pour chaque point la distance à la courbe la plus proche. Comme nous allons le voir, cette information pourrait permettre des améliorations à la fois du rendu et du temps de calcul.

5.1 Insuffisances actuelles

- Dans les cas de discontinuités de la tangente le long d'une courbe (phénomène de pointe, voir Figure 9), on observe un mélange non souhaité des couleurs.

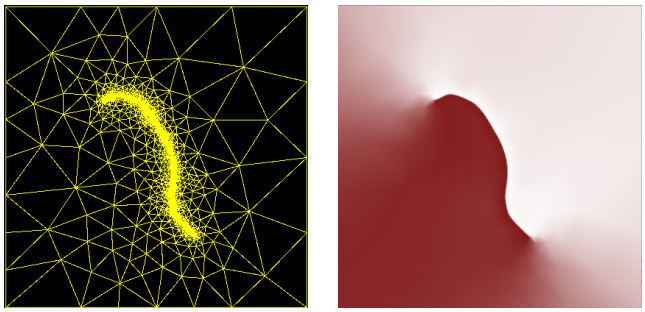
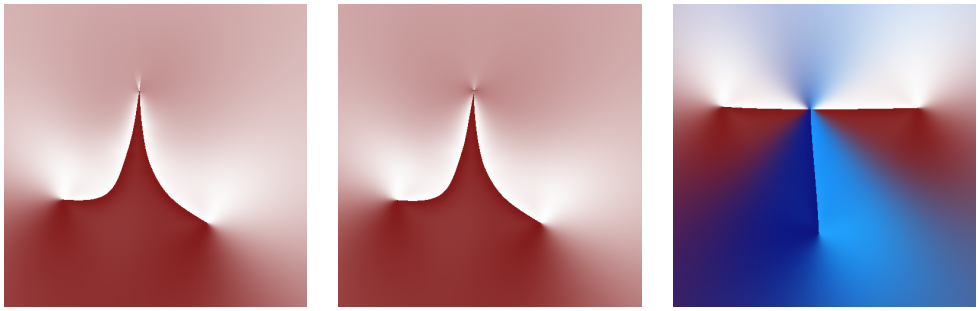


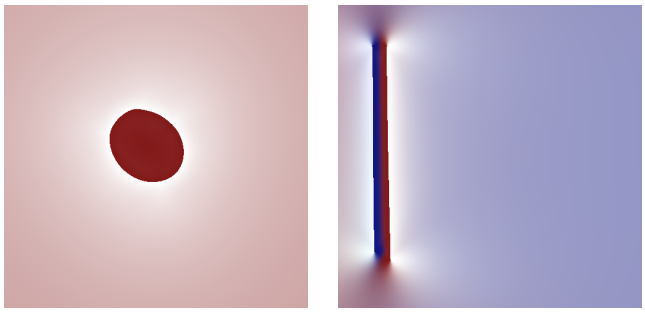
FIGURE 8 – Exemple maillage

- Phénomène de T (voir Figure 9) : les couleurs de la première courbe traversent la seconde.
- Mélange des couleurs à longue distance et diffusion en dehors d'une courbe fermée (voir Figure 10).
- Qualité des dégradés dépendante de la distance entre les courbes.
- Temps de calcul catastrophique.



(a) (b) (c)

FIGURE 9 – (a) Effet de pointe sur une courbe continue ; (b) Effet de pointe sur deux courbes à extrémités proches ; (c) Phénomène de T



(a) (b)

FIGURE 10 – (a) Diffusion en dehors d'une courbe fermée ; (b) Mélange de couleurs à longue distance

5.2 Modèle et rendu

▷ Il est possible de redéfinir la fonction de couleur pour obtenir un flou plus joli ou tout au moins différent.

▷ Possibilité d'améliorer les dégrader entre les courbes par l'application d'un filtre de lissage dont le rayon dépendrait de la distance à la courbe la plus proche. Ceci pourrait être donné par la fonction de distance). Problème : cette application pourrait prendre un temps conséquent, sans doute comparable à celui nécessaire à l'application du flou dans l'implémentation par équation de diffusion.

▷ Amélioration de la diffusion aux extrémités par un travail plus poussé sur la fonction de couleur en ces points. Par exemple, donner un poids plus important dans la direction tangentielle.

▷ Amélioration (ou paramétrisation) de la fonction de couleur. Définir un angle de diffusion pour la couleur gauche pure (respectivement droite) avant d'amorcer la décroissance quadratique vers la couleur moyenne.

▷ Possibilité d'utiliser des fonctions de poids à support compacte. L'utilisateur pourrait alors fixer des supports différente suivant les courbes. Cela permettrait de régler tous les problèmes de mélange des couleurs (courbe trop proche, pointes, mélange à distance...).

Ceci pourrait être fait de manière très ergonomique : lors de la sélection d'une courbe l'utilisateur verrait apparaitre une enveloppe délimitant l'influence de la courbe. Il pourrait alors la modifier. Il est même possible d'envisager de donner à l'utilisateur un contrôle sur le profil de la fonction de poids (La forme des dégradés serait alors totalement maitrisable par le dessinateur).

Cette méthode présente aussi l'avantage de permettre une réduction du temps de calcul (La fonction de couleur n'as pas besoin d'être calculer si le poids trouvé est nul).

▷ L'utilisation du canal alpha et de calques permettrait d'intégrer facilement les dessins créés à une image préexistante.

▷ Etant donner que les résultats avec différentes fonctions de poids sont valides dans différentes partie de l'image, il serait possible de paramétrer celle-ci en fonction de la distance. Cependant, la paramétrisation pourrait s'avérer être un problème particulièrement ardu et il n'est pas certain qu'une telle approche offre des résultats satisfaisants.

5.3 Implémentation et temps de calcul

▷ Amélioration du maillage : la construction d'un maillage parfaitement adapté au problème est nécessaire.

▷ Evaluation de la fonction de couleur uniquement si la distance du point à la courbe n'est pas beaucoup plus grande que la distance à la courbe la plus proche (fonction de distance). Notons que le même gain pourrait être réalisé à l'aide de fonctions de poids à support compact (cf paragraphe précédent).

▷ Organisation des contraintes par blocs dans l'image plutôt que par courbe. Nous n'utilisons pour le calcul que les blocs les plus proches en fonction de la distance à la courbe la plus proche. Ceci permet donc une forte diminution du nombre de contraintes à prendre en compte pour le calcul de la couleur en un point donné. Cette structure de données ne prend pas nécessairement beaucoup de temps à mettre en place et peut faire gagner un temps très important (surtout si l'on travail sur carte graphique car la gestion de la mémoire pourrait être optimisée).

▷ Remplacement des exponentielles par des fonctions plus simples et relecture/optimisation du code qui est actuellement très basique. En particulier, optimisation et/ou simplification de la

fonction de couleur.

▷ Implémentation de l'algorithme sur GPU avec le langage de programmation CUDA. En effet, le problème actuel est très facilement parallélisable, le gain de temps pourrait donc être significatif.

▷ Une dernière piste pourrait être l'optimisation du nombre de points sur chaque courbe en fonction de la longueur et de la résolution. Actuellement, seule la longueur est prise en compte, et les paramètres ont été choisis de manière empirique.

6 Conclusion

Les résultats obtenus sont satisfaisants bien qu'il présente un certain nombre de défauts à corriger. De plus en améliorant l'implémentation de la résolution basée sur les Moving Least Squares, il devrait être possible de donner plus de contrôle à l'utilisateur. Pour le moment le temps de calcul pour obtenir un rendu correct est beaucoup trop important mais il devrait être possible de le réduire considérablement.

Références

[1] Orzan, A., Bousseau, A., Winnemoller, H., Barla, P., Thollot, J., and Salesin, D. 2008. *Diffusion Curves : a vector representation for smooth-shaded images*

[2] Shen, C., J.F.O'Brien, Shewchuk, J.R. 2004. *Interpolating and approximating implicit surfaces from polygon soup*