

# Projet CAV : Tensor Directed Stylization

BERTHET Vincent, QUERO Benoit      Enseignant : Olivier LE MEUR

[https://github.com/RealVincentBerthet/Tensor\\_directed\\_stylization](https://github.com/RealVincentBerthet/Tensor_directed_stylization)

Fevrier 2020

## 1 Introduction

L'article « Tensor-directed simulation of strokes for image stylization with hatching and contours » de David Tschumperlé, aborde le problème de génération automatique de dessins et de croquis à partir de photographies. La méthode est basée sur une simulation de successions de traits de crayon dirigés par un champs de tenseurs. Elle permet de créer différents styles de dessins avec différentes sortes de hachures et de contours selon les paramètres en entrée de l'algorithme.

La particularité de cet algorithme est l'utilisation de tenseurs. Ces tenseurs permettent de modéliser la géométrie locale des traits et donc de générer à la fois une stylisation sur les zones de contours ainsi que sur les zones d'intensités plus homogènes. Cette méthode permet de faire apparaître des hachures dans les régions homogènes de notre image et des traits plus marqués sur les contours, renforçant le côté artistique du résultat. En faisant varier les nombreux paramètres, il est possible d'obtenir une multitude de styles de crayonnage différents. Pour implémenter cet algorithme, nous avons notamment utilisé les technologies Python et OpenCV.

## 2 Approche Naïve

L'approche naïve est une approche simple de la stylisation en dessin puisqu'elle ne se base que sur le gradient de l'image pour la position des traits à tracer. Comme les structures les plus importantes de notre image se trouvent dans le gradient, cette méthode permet d'obtenir des résultats assez réalistes.

Ce premier algorithme consiste donc à :

- Calculer la luminance  $Y$  de notre image  $I$ , puis le gradient de sa version lissée  $\nabla Y_\sigma = Y * G_\sigma$  où  $G_\sigma$  est un noyau gaussien isotrope d'écart type  $\sigma$ .
- Initialiser une image blanche qui va contenir le résultat final

Pour tracer les traits :

- Tirer au sort une position  $(x, y)$  telle que  $\|\nabla Y(x, y)\| \geq \epsilon$ .  $\epsilon$  est un seuil nous permettant de ne garder que les contours les plus importants de  $I$ .
- Tracer un segment  $(x - u, y - v)$  à  $(x + u, y + v)$ , où  $(u, v) = \frac{L}{2} \frac{\nabla Y^\perp}{\|\nabla Y\|}$ .  $L$  est la longueur du trait que nous définissons.



Figure 1: Pyramide traçage approche naïve

Comme on peut le voir sur les résultats, les contours sont stylisés mais les zones d'intensités homogènes comme le ciel ne le sont pas car leurs gradients faibles ne permettent pas à l'algorithme de prendre en compte ces zones lors du lancer de traits.

### 3 Approche Tenseurs

#### 3.1 Calcul des tenseurs de structure

Pour pouvoir styliser les régions homogènes d'une image, cette approche propose un algorithme de lancer de traits dirigé par un modèle tensoriel. Pour chaque pixel de notre image, un tenseur  $y$  est attribué. Si ce pixel est sur une zone homogène, le tenseur est isotrope et nous voulons donc que les traits de crayon soient également isotropes (hachures). De même, dans le cas d'un contour, le tenseur est anisotrope et nous voulons des traits anisotropes, suivant les contours.

Dans un premier temps, nous calculons le champs de tenseurs de structure lissé  $G$  :

$$G_{\alpha,\sigma} = \left( \sum_{i=1}^n \nabla I_{i\alpha} \nabla I_{i\alpha}^T \right) * G_{\sigma} \text{ où } I_{i\alpha} = I_i * G_{\sigma}$$

Pour chaque tenseur  $(x, y)$  de ce champs, les valeurs propres  $\lambda_+$  et  $\lambda_-$  sont représentatives des variations de couleurs de  $I$  et donc représentatives du type de structure géométrique sur lequel le pixel  $(x, y)$  se situe (contour ou région homogène). Les vecteurs propres sont quant à eux représentatifs de l'orientation de ces structures.

#### 3.2 Calcul des tenseurs de traits

Dans un second temps, nous calculons le champs de tenseur de traits  $T$  :

$$\forall (x, y) \in \Omega, T_{(x,y)} = c_+ \theta_+ \theta_+^T + c_- \theta_- \theta_-^T$$

avec  $c_+ = \frac{1}{(1+\lambda_++\lambda_-)^{p_1}}$ ,  $c_- = \frac{1}{(1+\lambda_++\lambda_-)^{p_2}}$  et  $p_1 \geq p_2 \geq 0$

Lorsque le pixel est situé sur un contour,  $T(x, y)$  est anisotrope, les traits seront donc tracés suivant l'orientation de  $\theta_-$ . Lorsque le pixel est situé en zone homogène,  $T(x, y)$  est isotrope, les traits tracés dans cette zone seront en forme de hachures.

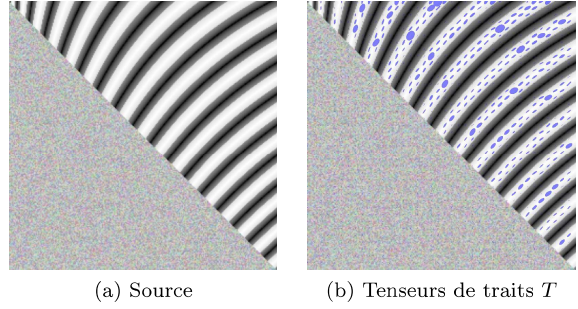


Figure 2: Principe de l'algorithme sur une image synthétique

La Figure 2(b) représente des ellipses sur une image synthétique. Ces ellipses sont de rayons  $c_+$  et  $c_-$  et orientées par  $\theta_-$ . Ce que l'on peut remarquer, c'est que les ellipses sont plus grandes dans les zones homogènes (les parties blanches) et plus petites dans les zones avec fort gradient (lignes et bruit). On remarque également qu'elles sont orientées parallèlement aux contours. Ces informations sont importantes car les styles des traits dessinés sont fonctions de  $c_+$ ,  $c_-$  et  $\theta_-$ .

$c_+$  et  $c_-$  sont les valeurs propres du tenseur de traits. Comme elles contiennent  $\lambda_+$  et  $\lambda_-$ , elles sont également représentatives de la structure géométrique sur laquelle se trouve le pixel associé. La seule différence est qu'il est possible de modifier le style de crayonnage en faisant varier les valeurs  $p_1$ ,  $p_2$  des valeurs propres  $c_+$  et  $c_-$ . Lorsque  $p_1 \approx p_2$ , les tenseurs et donc les traits de crayon sont isotropes partout. Lorsque  $p_1 \gg p_2 \gg 0$ , les tenseurs et les traits de crayon sont anisotropes partout.

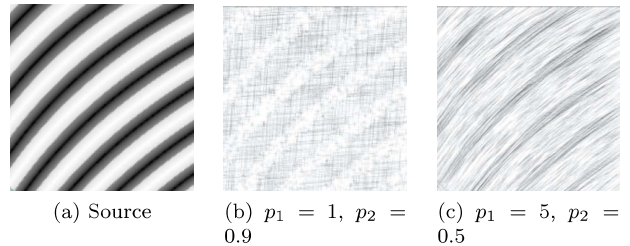


Figure 3: Variation de styles liés à  $p_1$  et  $p_2$

La Figure 3 nous montre les variations de style liés à  $p_1$  et  $p_2$ . Sur une image source avec un faible gra-

dient, si  $p_1 = p_2$ , alors  $c_+ \approx c_-$  et la zone est considérée comme homogène. Lorsque  $p_1 \gg p_2 \gg 0$ , alors  $c_+ \gg c_-$  et la zone est considérée comme contenant des contours.

Avec le modèle tensoriel, nous traçons des traits dans plusieurs directions, contrairement à l'algorithme naïf où nous traçons les traits uniquement dans la direction du gradient. Nous décomposons donc le champ de tenseurs de traits  $T$  en plusieurs champs de vecteurs.

$$w_\gamma = \sqrt{T} a_\gamma \text{ avec } \gamma \in [0, \pi] \text{ et } a_\gamma = (\cos\gamma \sin\gamma)^T$$

Lorsque  $T(x, y)$  est anisotrope ( $c_- \approx 0$ ), les traits de crayon orientés par  $w_\gamma$  sont toujours dans le sens du contour, peu importe les  $\gamma$  choisis. Lorsque  $T(x, y)$  est isotrope ( $c_+ = c_-$ ), les traits sont orientés suivant  $a_\gamma$  et suivent donc les angles  $\gamma$  choisis. Dans notre implémentation, nous avons choisi des hachures en quadrillage en utilisant un  $\gamma = [0, \frac{\pi}{2}]$ . Sur l'exemple suivant, nous avons utilisé deux  $\gamma$  différents sur une image assez homogène. Pour  $\gamma = [0, \frac{\pi}{2}]$ , on remarque bien que deux axes ont été choisis car les lignes sont horizontales et verticales. Pour  $\gamma = [0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$ , quatre axes sont choisis, nous avons donc en plus des lignes diagonales.

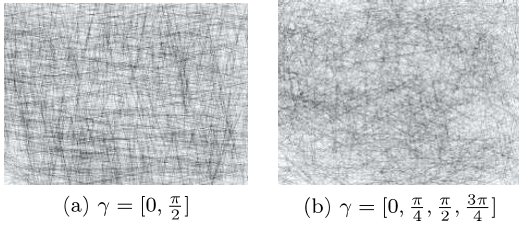


Figure 4: Deux styles de hachures en fonction des  $\gamma$  choisis

### 3.3 Traçage des traits de crayon

Nous initialisons tout d'abord une image blanche. Comme pour l'algorithme naïf, nous tirons au hasard une position  $(x, y)$  dans l'image. Nous regardons ensuite si  $\sqrt{\lambda_{+(x,y)} + \lambda_{-(x,y)}} > \varepsilon$ . Si elle est supérieure, nous traçons alors un segment allant de  $(x, y) - Lw_{\gamma(x,y)}$  à  $(x, y) + Lw_{\gamma(x,y)}$  avec  $L$  une valeur nous permettant de faire varier la longueur du trait à notre guise.

La figure suivante nous montre le résultat obtenu sur une image synthétique présentant des contours et des zones homogènes. Nous pouvons clairement remarquer sur cette image que l'approche tenseurs permet d'avoir des zones homogènes stylisées avec des hachures, contrairement à l'approche naïve.

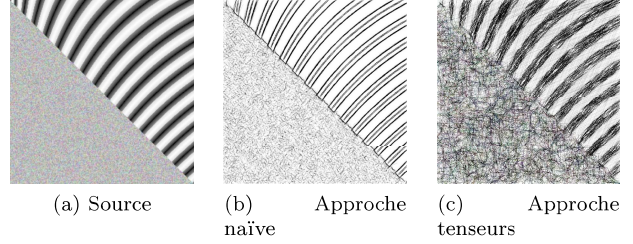


Figure 5: Comparaison des approches sur une image synthétique

Pour que le rendu soit plus joli, nous choisissons d'ajouter une couleur aux traits en lui assignant la couleur du pixel  $(x, y)$  de  $I$ . Dans l'article, l'auteur utilise également d'autres techniques de colorisation comme la technique de la multiplication, de l'overlay ou de l'hard light.

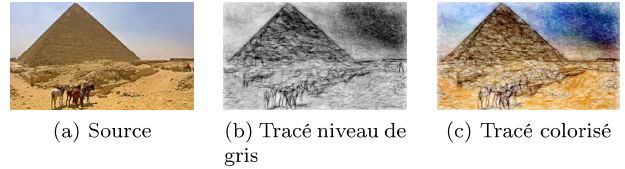


Figure 6: Pyramide traçage tenseurs



## 2 Galerie



(a) Source

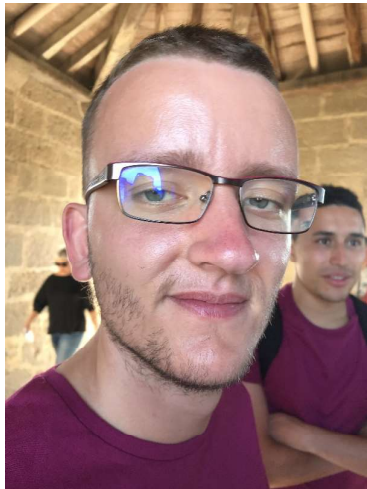


(b) Sketch 1



(c) Sketch 2

Figure 1: Obama



(a) Source



(b) Sketch 1



(c) Sketch 2

Figure 2: Matthias



(a) Source



(b) Sketch 1



(c) Sketch 2

Figure 3: Remi Cozot



(a) Source



(b) Sketch 1



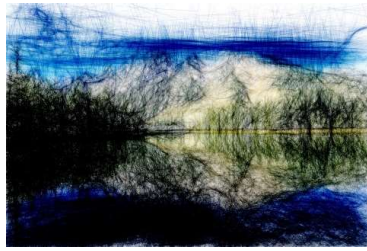
(c) Sketch 2

Figure 4: Olivier Le Meur





(a) Source



(b) Sketch 1



(c) Sketch 2

Figure 5: Montagne



(a) Source

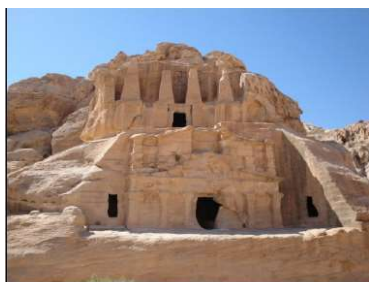


(b) Sketch 1



(c) Sketch 2

Figure 6: Lena



(a) Source



(b) Sketch 1

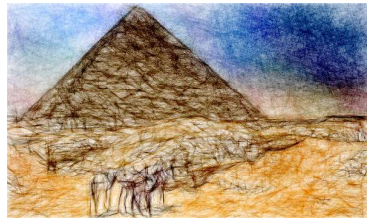


(c) Sketch 2

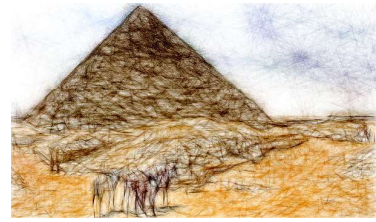
Figure 7: Desert



(a) Source



(b) Sketch 1



(c) Sketch 2

Figure 8: Pyramide



(a) Source



(b) Sketch 1



(c) Sketch 2

Figure 9: Ratatouille