Stylisation d'image basé contours et hachures par simulation de tracés de traits à géométrie tensorielle

David Tschumperlé

Stylisation d'image basé contours et hachures par simulation de tracés de traits à géométrie tensorielle

David TSCHUMPERLÉ

Laboratoire GREYC - Equipe IMAGE (UMR CNRS 6072)
6 Bd Maréchal Juin, 14050 Caen Cedex, France
David.Tschumperle@greyc.ensicaen.fr

Résumé – Nous abordons le problème de la transformation automatique de photographies numériques sous forme de dessins ou de croquis stylisés. Nous proposons un algorithme de rendu, simple et rapide à mettre en oeuvre, basé sur une simulation de successions de traits de crayon dirigé par un champ de tenseurs du 2ème ordre. Il permet de simuler de manière réaliste un crayonnage tel que pourrait l'exécuter un artiste souhaitant reproduire une photographie couleur sous forme de croquis. Une étape de colorisation simple, utilisant les couleurs de l'image d'entrée, permet de finaliser ce rendu non-photoréaliste.

Abstract – We describe a simple but capable algorithm to generate sketches from color images. The method is based on the drawing of dark strokes on a white background, directed by an image-dependent tensor-valued geometry. It is able to produce various sketching styles with different kind of hatching and contours. Combining these grayscale sketches with the colors of the input images allows to produce a wide variety of image stylization renderings.

1 Introduction

Dans le domaine du rendu non-photoréaliste, les méthodes basées "traits" (*strokes* en anglais) [8] servent à définir des processus de transformation d'images numériques, par exemple, sous forme de peintures [9], croquis [7], gravures [5], comics [3], ou encore de lavis [6, 10, 14]. Des techniques semblables ont été utilisées en visualisation d'image, pour la représentation de champs de vecteurs [2], de tenseurs [11] ou d'images médicales [1]. Ces algorithmes diffèrent principalement par le placement spatial ainsi que par la forme des primitives choisies comme briques de bases pour générer le rendu. Ici, nous proposons un algorithme de *simulation de traits de crayon*, ayant la particularité de générer *à la fois* une stylisation des zones de contours *et* des zones d'intensités plus homogènes. Ceci est rendu possible par l'utilisation de *champs de tenseurs du* 2^{eme} *ordre* qui permettent de modéliser finement la géométrie locale des traits que l'on souhaite tracer. Notre technique se base sur des "lancers" successifs de traits de crayon, à partir de positions aléatoires dans l'image, de telle manière que chacun de ces rendus soit consistant avec la géométrie tensorielle des traits prédéfinie, c-à-d en faisant apparaître des hachures spatialement régulières sur les régions homogènes (cas des tenseurs *isotropes*), et en surlignant nettement les contours lorsqu'il y en a (cas des tenseurs *anisotropes*). Notre méthode est particulièrement simple à implémenter et rapide à exécuter, tout en étant capable de générer de multiples styles différents de dessins, grâce à la variabilité naturelle du modèle tensoriel considéré (Fig.1). Des résultats additionnels d'application de notre algorithme après colorisation closent cette publication.

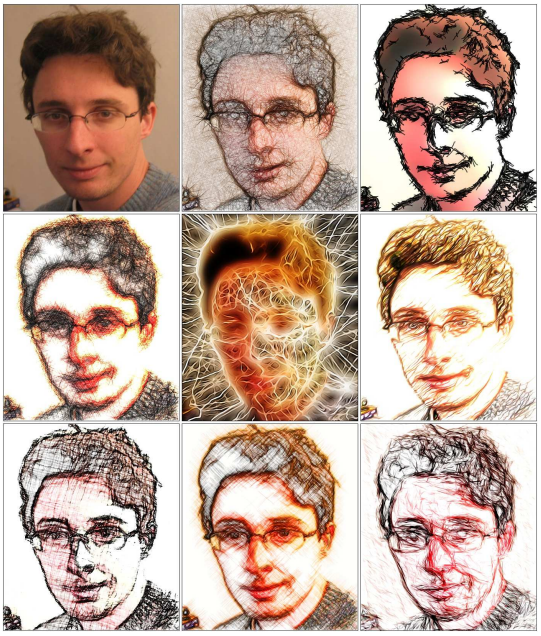


FIGURE 1 – *Un algorithme de simulation de traits de crayon est proposé, pour le rendu stylisé automatique de photographies numériques couleurs. Notre méthode est capable de générer des styles de dessins variés à partir d'une même image d'entrée (ici, en haut à gauche).*

2 Une première approche naïve

Pour simuler la conversion d'une image couleur $\mathbf{I} : \Omega \to \mathbb{R}^3$ sous forme d'un crayonnage de traits noirs sur fond blanc, nous suggérons dans un premier temps de déterminer un ensemble

de segments représentant les structures les plus significatives de $\mathbf{I}$. Comme la plupart des informations de contours sont présentes dans le gradient lissé de la composante de luminance $Y : \Omega \to \mathbb{R}$ de $\mathbf{I}$, on peut assez naturellement suggérer un premier algorithme naïf, pour la stylisation d'image par traits de crayon :

1. Calculer le champ de luminance Y de l'image couleur $\mathbf{I}$, puis le gradient de sa version lissée $\nabla Y_\sigma = Y * G_\sigma$, où G_σ est un noyau gaussien isotrope d'écart type $\sigma \in \mathbb{R}^+$.
2. Initialiser une image constante, à fond blanc $S : \Omega \to \mathbb{R}$, qui va contenir le rendu de l'algorithme de croquis automatique.
3. Tirer au sort une position $(x, y) \in \Omega$, telle que $\|\nabla Y_{(x,y)}\| \geq \epsilon$. Choisir un paramètre $\epsilon \in \mathbb{R}^+$ important permet ici de focaliser le rendu sur les contours les plus contrastés de l'image $\mathbf{I}$.
4. Tracer dans S le segment de couleur noire (partiellement transparent) de $(x-u, y-v)$ à $(x+u, y+v)$, où $(u, v) = \frac{L}{2} \frac{\nabla Y^\perp}{\|\nabla Y\|}$. La longueur du trait $L \geq 0$ peut être éventuellement définie comme une fonction de $\|\nabla Y_{(x,y)}\|$ (contraste local).
5. Retour à l'étape 3, jusqu'à ce qu'un nombre significatif de traits aient été tracés dans S (dépend de la taille de Ω, et de L).

Cette approche rudimentaire permet déjà de générer des esquisses réalistes, mais où plusieurs caractéristiques "artistiques" typiques sont manquantes : D'abord, on ne permet que le tracer de traits de crayon *droits*, qui limitent la diversité de stylisation des contours (ainsi que l'intervalle de valeurs acceptables pour le choix de L). Ensuite, on ne génère pas de traits spatialement harmonieux dans les régions homogènes de l'image (c-à-d lorsque l'on choisit $\epsilon \to 0$). L'orientation $\frac{\nabla I}{\|\nabla I\|}$ n'ayant aucune cohérence dans ces zones ($\|\nabla I\| \to 0$), les segments résultants s'entrecroisent de manière quasi-aléatoire, contrairement à un motif de hachures par exemple, qui serait ici plus adapté. Dans ce papier, notre contribution principale consiste donc à proposer un *algorithme de simulation de traits dirigé par un modèle tensoriel*, permettant de maîtriser plus finement les comportements locaux de ces tracés.

3 Géométrie tensorielle des traits

Les tenseurs du second ordre sont des matrices symétriques définies-positives, adaptées pour modéliser des caractéristiques géométriques locales de natures isotropes ou anisotropes. En traitement d'image, des champs de tenseurs ont déjà été utilisés avec succès pour modéliser la géométrie de structures locales (*tenseurs de structure*) [4, 13], ou pour décrire le comportement de processus de diffusion dans le cadre du débruitage d'image (*tenseurs de diffusion*) [12, 13]. Ici, nous considèrons de manière similaire, que les traits de crayon à simuler peuvent avoir des configurations soit isotropes (*hachures*), soit anisotropes (*contours*), selon que le point de départ du trait est situé sur une région quasi-homogène ou sur un contour de l'image. Modéliser cette variété de configurations par un champ tensoriel semble donc assez naturel. L'approche en deux étapes que nous proposons est dans l'idée assez proche de celles définies dans [13, 12] pour la régularisation d'image : Nous calculons, dans un premier temps, le champ de tenseurs de structure lissé :

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

En tout point $(x, y) \in \Omega$, les valeurs propres $\lambda_+, \lambda_- \geq 0$ de $\mathbf{G}_{\alpha,\sigma(x,y)}$ sont représentatives de la variation locale des couleurs dans $\mathbf{I}$. $(\lambda_+ + \lambda_-)$ peut être alors considérée comme une mesure fiable pour déterminer le type de structure géométrique sur lequel (x, y) est situé (contour, région homogène ou coin). La base de vecteurs propres $\theta_+ \perp \theta_-$ donne quant à elle, l'orientation prédominante des structures locales (θ_- est parallèle au contour). La Fig.2a expose une illustration de $\mathbf{G}_{\alpha,\sigma}$ sous forme d'un champ d'ellipses de rayons $\lambda_{+/-}$ orientées par $\theta_{+/-}$. L'utilisation des tenseurs (1) assure que la corrélation inter-canaux est bien prise en compte (contrairement à l'approche naïve précédente), et ses paramètres de lissage permettent une estimation de la géométrie locale plus robuste au bruit et/ou à un espace échelle quelconque. Une fois que $\mathbf{G}_{\alpha,\sigma}$ est estimé, nous pouvons modéliser la géométrie des traits de crayon par *un autre champ tensoriel* $\mathbf{T} : \Omega \to \mathrm{P}(2)$ (*tenseurs de traits*) :

$$\forall (x, y) \in \Omega, \quad \mathbf{T}_{(x,y)} = c_+ \theta_+ \theta_+^T + c_- \theta_- \theta_-^T \quad (2)$$

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

Ces tenseurs de traits $\mathbf{T}$ favorisent les comportements suivants :
- Lorsque $\mathbf{T}_{(x,y)}$ est anisotrope, il est orienté le long de $\theta_{-(x,y)}$, et le point (x, y) est probablement localisé sur un contour de l'image ($\lambda_{+/-}$ forts). Chaque trait de crayon partant de (x, y) devrait donc effectivement suivre la direction $\theta_{-(x,y)}$ du contour.
- Lorsque $\mathbf{T}_{(x,y)}$ est isotrope, le point (x, y) se trouve probablement sur une région d'image quasi-homogène ($\lambda_{+/-}$ faibles), et les traits dessinés ici devraient générer des hachures, spatialement cohérentes, selon plusieurs directions du plan.
- Lorsque $\mathbf{T}_{(x,y)}$ définit une configuration intermédiaire (tenseurs "gonflés", mais pas complètements isotropes), on souhaite qu'ils génèrent des traits représentant un mélange continu entre contours nets et hachures.

Les valeurs propres $c_{+/-}$ de $\mathbf{T}_{(x,y)}$ seront logiquement relatives à la longueur du trait tracé en (x, y), à la fois pour les contours et les hachures, et dépendent plus ou moins du contraste local de $\mathbf{I}$. Notons qu'avec seulement deux paramètres p_1, p_2 pour $\mathbf{T}$, nous autorisons déjà une grande variété de styles de crayonnage, car on peut moduler de manière globale, le profil préférentiel des tenseurs (isotropes partout avec $p_1 \approx p_2$, anisotropes partout avec $p_1 \gg p_2 \gg 0$), ainsi que la dépendance des tenseurs au contraste local $(\lambda_+ + \lambda_-)$. La Fig.2b expose un champ $\mathbf{T}$ de tenseurs de traits, pour une image couleur, avec les paramètres $(p_1, p_2) = (1.2, 0.5)$ et $(\alpha, \sigma) = (0.5, 1.2)$.

4 Lancer de traits à géométrie tensorielle

L'algorithme naïf de tracé de traits (section 2) ne sait gérer qu'une unique orientation $\nabla Y_{(x,y)} / \|\nabla Y_{(x,y)}\|$ pour tracer un segment en un point (x, y). Ici, nous étendons cette technique à une géométrie tensorielle, ç-à-d le rendu d'un ou de plusieurs segments (éventuellement orientés différemment) selon

que $\mathbf{T}_{(x,y)}$ est plutôt anisotrope ou isotrope. Pour cela, nous proposons de décomposer le champ de tenseurs de traits $\mathbf{T}$ en plusieurs champs de vecteurs $\mathbf{w}_\gamma : \Omega \rightarrow \mathbb{R}^2$, exprimant les quotités de $\mathbf{T}$ suivant les orientations $\gamma \in [0, \pi]$ du plan. En remarquant que $\int_{\gamma=0}^{\pi} a_\gamma a_\gamma^T \, d\gamma = \frac{\pi}{2} \mathbb{I}_d$, avec $a_\gamma = (\cos\gamma \;\; \sin\gamma)^T$, nous pouvons écrire $\mathbf{T} = \frac{2}{\pi} \sqrt{\mathbf{T}} \left(\int_{\gamma=0}^{\pi} a_\gamma a_\gamma^T \, d\gamma \right) \sqrt{\mathbf{T}}$, et donc : $\mathbf{T} = \frac{2}{\pi} \int_{\gamma=0}^{\pi} \mathbf{w}_\gamma \mathbf{w}_\gamma^T \, d\gamma$, avec $\mathbf{w}_\gamma = \sqrt{\mathbf{T}} a_\gamma$ ($\sqrt{\mathbf{T}}$ dénote ici la racine carré matricielle). Chaque $\mathbf{w}_\gamma \mathbf{w}_\gamma^T$ est un champ tensoriel unitaire purement anisotrope, orienté selon $\mathbf{w}_\gamma$.
- Lorsque $\mathbf{T}_{(x,y)}$ est parfaitement anisotrope ($c_+ \approx 0$), alors $\mathbf{w}_{\gamma(x,y)} = (\theta_-.a_\gamma) c_-^{\frac{1}{2}} \theta_-$ est toujours parallèle à θ_-. Un trait de crayon orienté par $\mathbf{w}_\gamma$ sera par conséquent toujours tracé dans la direction du contour, quelque soit l'angle γ considéré.
- Lorsque $\mathbf{T}_{(x,y)}$ est parfaitement isotrope ($c_+ \approx c_-$), alors $\mathbf{w}_{\gamma(x,y)} = c_- a_\gamma$ reste orienté par a_γ, ce qui veut dire que dans les régions quasi-homogènes, de multiples traits orientés différemment suivant les $\mathbf{w}_\gamma$ vont être tracés à partir d'un même point (x, y). On permet donc ici de créer un rendu localement *hachuré*, et spatialement cohérent, avec un type de hachures paramétrable, et uniquement dépendant de la façon dont γ est discrétisé (c-à-d par trois paramètres $\gamma_{min}, \gamma_{max}$ et $\delta\gamma$).
- Lorsque $\mathbf{T}_{(x,y)}$ définit une configuration intermédiaire, il va plus ou moins *attirer* les orientations a_γ vers ses vecteurs propres $\theta_{+/-}$, suivant son degré d'anisotropie. Cela va permettre la génération de transitions continues entre les deux configurations extrêmes correspondants aux cas isotropes (hachures) et anisotropes (contours).

La décomposition de $\mathbf{T}$ en de multiples champs vectoriels $\mathbf{w}_\gamma$ suggère également d'exprimer un trait non plus comme un simple segment orienté par $\mathbf{w}_\gamma$, mais comme la ligne de champ $\mathcal{C} : [-\frac{L}{2}, \frac{L}{2}] \rightarrow \mathbb{R}^2$ de $\mathbf{w}_\gamma$, partant de (x, y) et de longueur L :

$$\frac{\partial \mathcal{C}_{(l)}}{\partial l} = \mathbf{w}(\mathcal{C}_{(l)}) \; , \quad \text{où} \quad \mathcal{C}_{(0)} = (x, y) \tag{3}$$

Considérer ces lignes de champ comme primitives de trait va permettre de mieux "coller" aux contours des images, quelque soit leur courbure. Les rendus de stylisation qui en découlent paraissent ainsi plus délicats. Finalement, notre algorithme de simulation de tracé de traits à géométrie tensorielle s'écrit :
1. Calculer le champ de tenseurs de traits $\mathbf{T}$ (2) à partir de la photographie numérique couleur $\mathbf{I}$.
2. Décomposer $\mathbf{T}$ en plusieurs champs de vecteurs $\mathbf{w}_\gamma = \sqrt{\mathbf{T}} a_\gamma$, pour γ discrétisé dans $[\gamma_{min}, \gamma_{max}]$ avec un pas δ_γ.
3. Initialiser une image constante à fond blanc $S : \Omega \rightarrow \mathbb{R}$ qui contiendra le rendu final du crayonnage (en niveau de gris).
4. Tirer au sort une position (x, y) dans Ω, telle que $\sqrt{\lambda_+ + \lambda_-} \geq \epsilon$. Notons que ϵ peut maintenant être ici choisi proche de 0.
5. Pour chaque γ, tracer dans S le segment (partiellement opaque) de $(x, y) - L \; \mathbf{w}_{\gamma(x,y)}$ à $(x, y) + L \; \mathbf{w}_{\gamma(x,y)}$, *ou* la ligne de champ (3) de $\mathbf{w}_\gamma$, partant de (x, y), de longueur $L\mathbf{w}_{\gamma(x,y)}$.
6. Retour à l'étape 4, jusqu'à ce qu'un nombre significatif de traits aient été tracés dans S.

Une illustration du fonctionnement de l'algorithme est présenté en Fig.2c sur une image synthétique comprenant des parties distinctes isotropes et anisotropes, avec des primitives en forme de lignes de champs. Notons la façon dont différents styles de hachures sont générés à partir de différents jeux de paramètres $(\gamma_{min}, \gamma_{max}, \delta_\gamma)$.

5 Colorisation finale du rendu

Une fois le rendu du croquis en niveau de gris obtenu, nous pouvons optionnellement le composer avec les couleurs originales de l'image d'entrée pour générer un rendu de dessin final, coloré. Pour générer les résultats présentés en Fig.3 et 4, nous avons utilisé des techniques de composition automatiques simples de calques, telles que celles utilisées en routine par les infographistes (multiplication, overlay, hard light, ...). Ces traitements additionnels sont triviaux, et peuvent se réaliser de manière automatique, l'étape essentielle de transformation de l'image étant réalisé par notre algorithme de simulation de trait de crayon. Cette technique de crayonnage automatique s'avère particulièrement flexible et générique, et sait générer des rendus de croquis réalistes, où les structures contenues dans des photographies couleurs sont naturellement extraites et mises en avant. L'algorithme est simple et peut être potentiellement parallélisé pour des applications temps réelles, ce qui ouvre des perspectives intéressantes pour l'application de conversion automatique de vidéos réelles en dessins animés par exemple.

Références

[1] S. Bruckner and M. Eduard Gröller, "Style transfer functions for illustrative volume rendering," Computer Graphics Forum, 26(3) :715-724, September 2007.

[2] B. Cabral and L.C. Leedom, "Imaging vector fields using line integral convolution," in *SIGGRAPH*, 1993, vol. 27.

[3] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," in *SIGGRAPH*, 2002, pp. 769–776.

[4] S. Di Zenzo, "A note on the gradient of a multi-image," *Comput. Vision Graph. Image Process.*, vol. 33, pp. 116–125, 1986.

[5] O. Deussen, S. Hiller, C. van Overveld, and T. Strothotte, "Floating points : A method for computing stipple drawings," *Computer Graphics Forum*, vol. 19, pp. 40–51, 2000.

[6] O. Deussen and T. Strothotte, "Computer-generated pen-and-ink illustration of trees," in *SIGGRAPH*, 2000, pp. 13–18.

[7] F. Durand, V. Ostromoukhov, M. Miller, F. Duranleau, and J. Dorsey, "Decoupling strokes and high-level attributes for interactive traditional drawing," in *Eurographics Workshop on Rendering Techniques*, 2001, pp. 71–82.

[8] A. Hertzmann, "A survey of stroke-based rendering," *IEEE Comp. Graphics and Applications*, vol. 23, pp. 70–81, 2003.

[9] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *SIGGRAPH*, 1998, pp. 453–460.

[10] M. Salisbury, C. Anderson, D. Lischinski, and D.H. Salesin, "Scale-dependent reproduction of pen-and-ink illustrations," in *SIGGRAPH*, 1996, pp. 461–468.

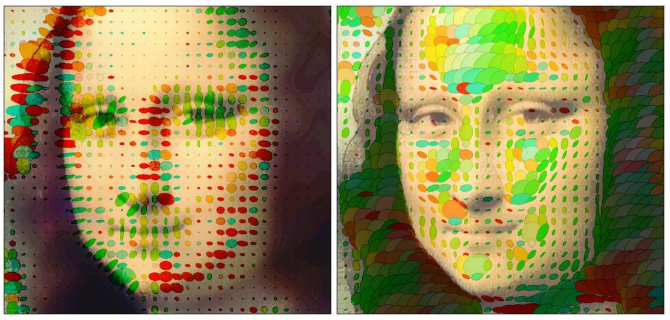
[11] D. Tschumperlé and R. Deriche, "Tensor field visualization with pde's and application to dt-mri fiber visualization," in *IEEE Work. on Variational and Level Set Methods*, 2003.

[12] D. Tschumperlé and R. Deriche, "Anisotropic diffusion partial differential equations for multichannel image regularization : Framework and applications," vol. 145 of *Advances in Imaging and Electron Physics*, pp. 149 – 209. Elsevier, 2007.

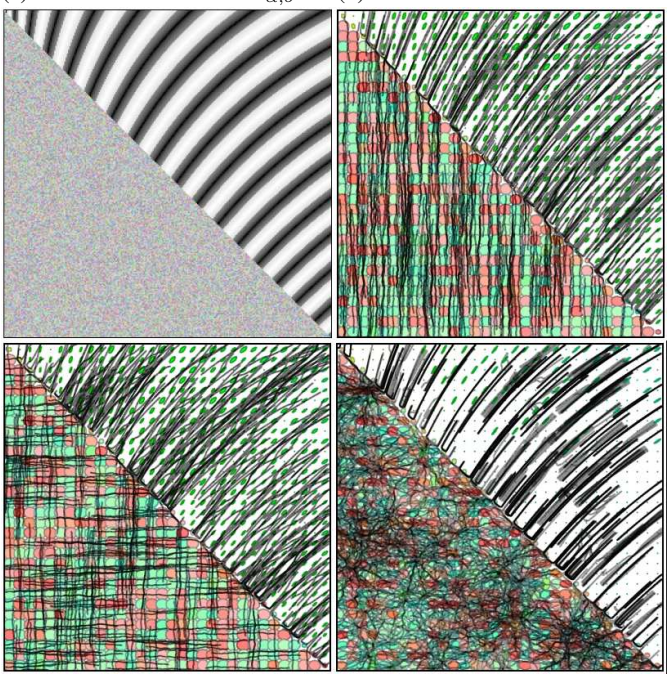
[13] J. Weickert, "Coherence-enhancing diffusion filtering," *Int. Journal. Comput. Vision*, vol. 31, pp. 111–127, April 1999.

[14] G. Winkenbach and D.H. Salesin, "Computer-generated pen-and-ink illustration," in *SIGGRAPH*, 1994, pp. 91–100.

Remerciements : *L'auteur tient à remercier chaleureusement Tom Keil, pour son aide à la génération des résultats présentés dans ce papier (Fig.3).*



(a) Tenseurs de structure $\mathbf{G}_{\alpha,\sigma}$ (b) Tenseurs de traits $\mathbf{T}$



(c) Image synthétique (haut-gauche) et simulations de traits générées

FIGURE 2 – *Principe de l'algorithme de lancer de traits. En haut, les champs de tenseurs de structure (a) et de traits (b) pour une image couleur. En bas, deux résultat (d),(e) de simulation à partir d'une image synthétique (c). Les paramètres* $(\gamma_{min}, \gamma_{max}, \delta_\gamma)$ *choisis sont respectivement* $(90^o, 90^o, 0)$ *et* $(0^o, 90^o, 90^o)$.

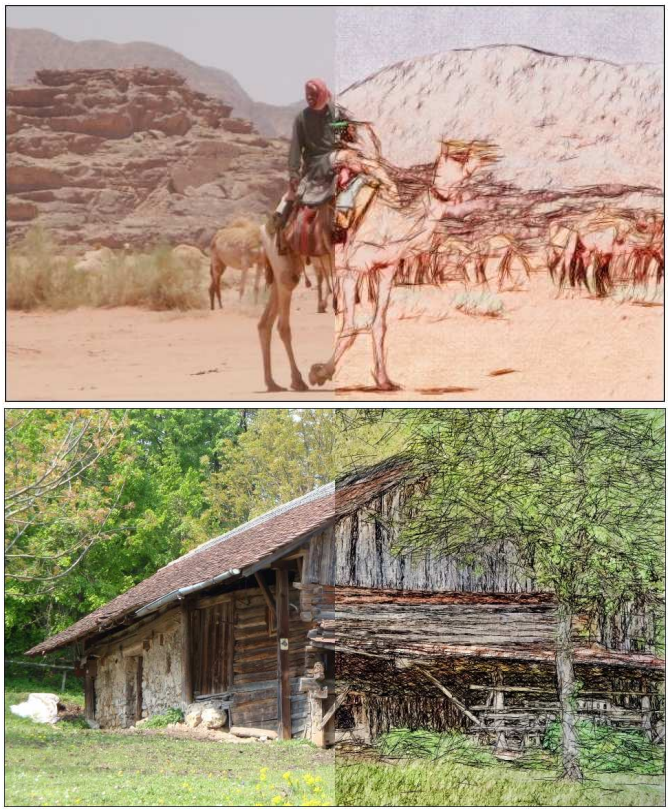


FIGURE 3 – *Exemples de générations automatiques de dessins stylisés (parties droites) à partir de photographies couleurs (parties gauche).*

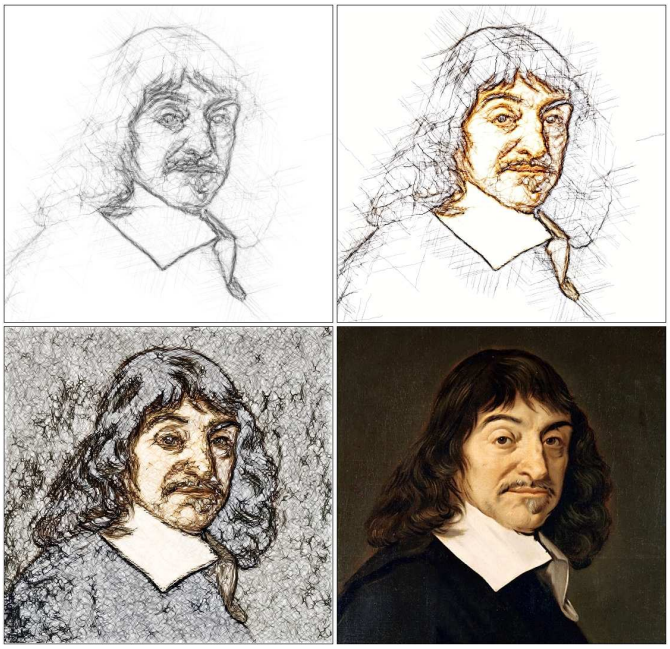


FIGURE 4 – *Simulation d'étapes chronologiques de dessin pour aboutir à une peinture finale (la dernière image est l'unique image d'entrée de notre algorithme de génération de croquis, exécuté avec des jeux de paramètres différents).*