

Tensor-Directed Simulation of Strokes for Image Stylization with Hatching and Contours



David Tschumperlé

Image Team, GREYC / CNRS (UMR 6072) IEEE ICIP'2011, Brussels/Belgium, September 2011



- Context : Non-Photorealistic Rendering (NPR).
- Motivation : Render a photograph as a sketch or a pencil drawing.
- Method : B&W stroke-based algorithm. Place (black) graphic primitives on an white canvas, according to the geometry of the initial image. Then, colorize the result.



Original color image



Final rendering

<ロト < 同ト < 目ト < 目 > 、 目 、 の へ つ >

Overview : State of the art of B&W NPR





Deussen-Hiller-etal (stipple drawing)



Hertzmann-Zorin (from 3d models)



Durand-Ostromoukhov-etal (interactive method)



Winkenbach-Salesin (from 3d models)



Salisbury-Wong-etal (oriented textures)



Salisbury-Anderson-etal (interactive method)



We propose (yet) another stroke-based algorithm with :

- Straight or curved lines as graphic primitives.
- 2nd-order tensors for analyzing the local image geometry and modeling the pencil paths.
- Ability to render image contours and hatching within the same single process.
- Use the flexibility of the tensor model to allow multiple drawing styles with the same algorithm and from the same input image.
- \implies Try to simulate pencil strokes as they would be drawn by an artist !

Illustration of the proposed method





Same input image leads to multiple styles of pencil rendering

▲ロト ▲圖ト ▲画ト ▲画ト 三回 - のへの

A first naive approach



- Compute the image luminance $Y : \Omega \to \mathbb{R}$, and its smoothed gradient $\nabla Y_{\sigma} = \nabla Y * G_{\sigma}$.
- **2** Create an empty (white) canvas $S : \Omega \to \mathbb{R}$.
- Repeat N times :
 - Pick one random position $\mathbf{X} = (x, y)$ in Ω .
 - 2 Draw a semi-transparent straight line $\mathbf{X} L\xi_{(\mathbf{X})} \rightarrow \mathbf{X} + L\xi_{(\mathbf{X})}$ on S,

where $\xi_{(\mathbf{X})} = \frac{\nabla Y_{\sigma(\mathbf{X})}^{\perp}}{\|\nabla Y_{\sigma(\mathbf{X})}\|}$ (contour direction at **X**).





Color image I

Luminance Y

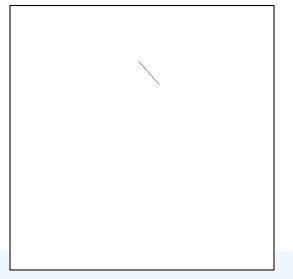
Smoothed gradient ∇Y_{σ}

(ロ) (日) (日) (日) (日) (日) (日)

After 1st iteration



(ロ)



After 200th iterations





After 600th iterations





After 1200th iterations

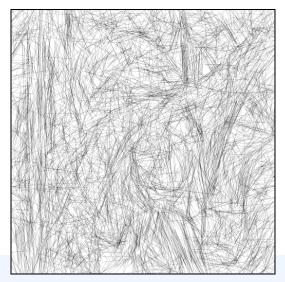




After 4800th iterations



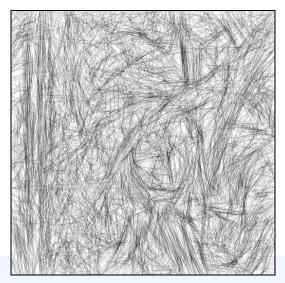
◆ロト ◆聞 と ◆注 と ◆注 と □ 目



After 10000th iterations



<ロ> <四> <四> <三> <三> <三> <三> <三> <三



After 10000th iterations (colorized)



イロト イポト イヨト イヨト 三日

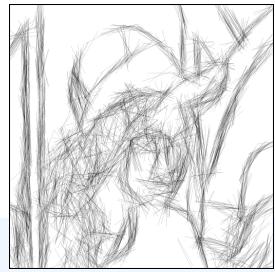


Canvas S + Colorization

Edge-focused modification



Draw a pencil stroke at X only when ||∇Y_{σ(X)}|| > ε
 (i.e. X is on a significant edge).



Edge-focused modification (colorized)



Draw a pencil stroke at X only when ||∇Y_{σ(X)}|| > ε
 (i.e. X is on a significant edge).





・ ロ ト ス 厚 ト ス ヨ ト ・

- Only straight lines.
- Poor analysis of the image geometry (simple gradient, no analysis of the color variations).
- No spatial coherence (or no drawing) on flat image regions.



⇒ Can be easily improved by considering tensor models !



 Inspired by the formalism of diffusion tensors, used for directing PDE-based anisotropic smoothing processes.

[Weickert:98,Tschumperle-Deriche:03,...]

$$\frac{\partial I}{\partial t} = \operatorname{div} (\mathbf{D} \nabla I)$$
 or $\frac{\partial I}{\partial t} = \operatorname{trace} (\mathbf{D} \mathbf{H})$

 Based on the analysis of the local image geometry through the computation of the structure tensor [DiZenzo:86].

$$\mathbf{G}_{lpha,\sigma} = (\sum_{i}
abla \mathbf{I}_{ilpha} \
abla \mathbf{I}_{ilpha}^{T}) * G_{\sigma}$$

Eigenvalues/eigenvectors of G_{α,σ} are robust local geometric estimators of the image structures. Diffusion tensors D are built upon G_{α,σ} and direct the smoothing process → Anisotropic diffusion.





 \implies Diffusion tensors model the way a digital painter would apply the "smudge tool" to smooth the noise and remove image artefacts.

...to stroke tensors

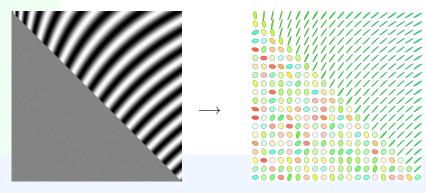


- Structure tensors are able to locally distinct between different types of image regions :
 - Flat regions : Small isotropic tensors.
 - 2 Contours : Anisotropic tensors, oriented \perp to the contour.
 - Small-scale texture : Large isotropic tensors.
- From an "sketching" point of view, this would correspond to differents drawing behaviors :
 - Flat regions or textures : Regular strokes forming a coherent hatching pattern.
 - Contours : Oriented strokes, along the contour direction.
- ⇒ Strokes geometry can be successfuly modeled by a tensor field T that depends on $G_{(\alpha,\sigma)}$!

[See paper content for more details about our proposal]

...to stroke tensors

- crrs 👔 📠
- From an "sketching" point of view, this would correspond to differents drawing behaviors :
 - Flat regions or textures : Regular strokes forming a coherent hatching pattern.
 - Contours : Oriented strokes, along the contour direction.



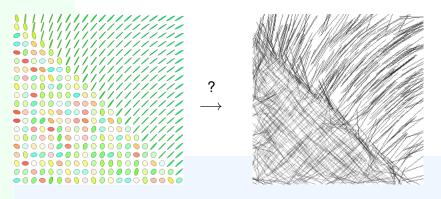
 \implies Stroke tensors variability will lead to different drawing styles.

< □ > < @ > < @ > <

From stroke tensors...



- **Question** : How to draw strokes on canvas *S* with respect to the stroke tensor field **T**, such that :
 - Small isotropic tensors locally render very few strokes.
 - 2 Large isotropic tensors locally render hatches.
 - Anisotropic tensors locally render oriented contour strokes.





• Stroke throwing algorithm : Initialize an empty canvas *S* and compute the stroke tensors **T** from the input image **I**.

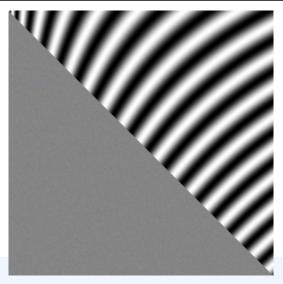
For each angle γ from a given set $[\gamma_0, \gamma_1, ..., \gamma_N]$, do :

- **O** Compute vector field $\mathbf{w}_{\gamma} = \sqrt{\mathbf{T}} a_{\gamma}$ with $a_{\gamma} = (\cos \gamma \ \sin \gamma)^{T}$.
- Repeat N times :
 - **1** Pick one random position $\mathbf{X} = (x, y)$ in Ω .
 - **2** Draw a semi-transparent straight line $\mathbf{X} L\xi_{(\mathbf{X})} \rightarrow \mathbf{X} + L\xi_{(\mathbf{X})}$ or a streamline of \mathbf{w}_{γ} starting from \mathbf{X} and of length *L*, on *S*.

- The stroke tensors are used to distort each direction a_γ, according to the geometry of the input image.
- The choosen number of angles γ set the style of the hatches.
- Streamlines allow drawn strokes to be curved.



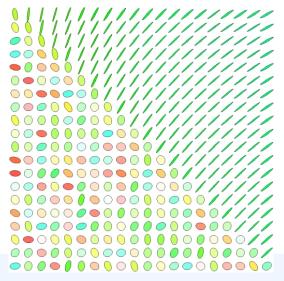
▲□▶ ▲□▶ ▲ヨ≯ ▲ヨ≯ ヨー のへで



Input image (synthetic).



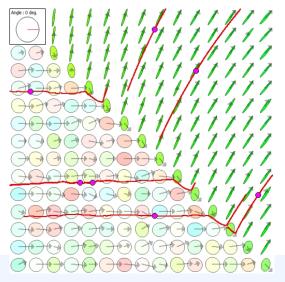
<ロト < 同ト < 目ト < 目 > 、 目 、 の へ つ >



Computed field of stroke tensors.



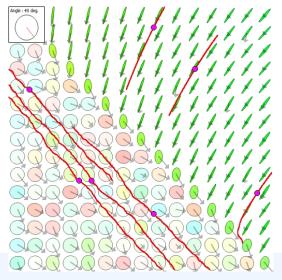
<ロト < 同ト < 回ト < 回ト = 三日



Some streamline strokes drawn for $\gamma = 0^{o}$.

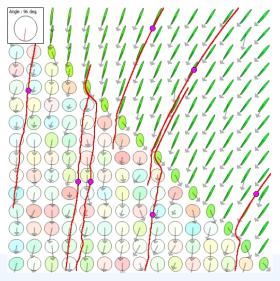


▲□ ▶ ▲□ ▶ ▲□ ▶ ▲□ ▶ □ ● のへで



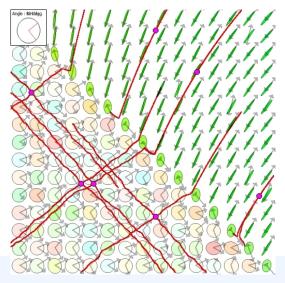
Some streamline strokes drawn for $\gamma = 48^{o}$.





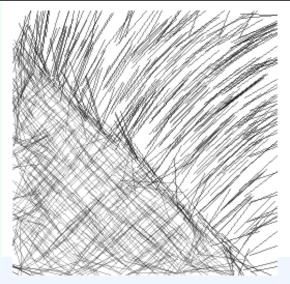
Some streamline strokes drawn for $\gamma = 96^{\circ}$.





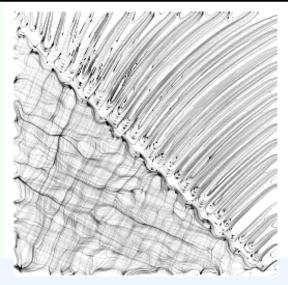
Some streamline strokes drawn for $\gamma = 45^{\circ}$ and $\gamma = 135^{\circ}$.





Final result with straight lines, $\gamma = 45^{o}$ and $\gamma = 135^{o}$.





Final result with curved lines, $\gamma = 45^{o}$ and $\gamma = 135^{o}$.

Colorization step



- Colorization combines the B&W sketch canvas and the colors of the input image, to add colors to the generated sketch.
- Here, we tried very simple combinations only, also known as layer blending modes : Soft light, hard light, overlay,....
- All results shown afterwards are using these simple coloring techniques.



Input image

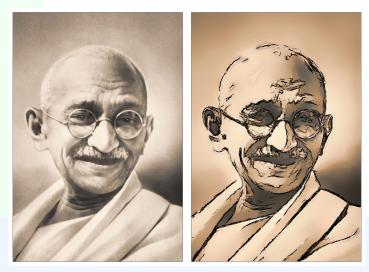
B&W sketch

"Colorized" result

<ロト < 同ト < 目ト < 目 > 、 目 、 の へ つ >



 Some straightforward applications of the sketch algorithm, with simple colorization steps...

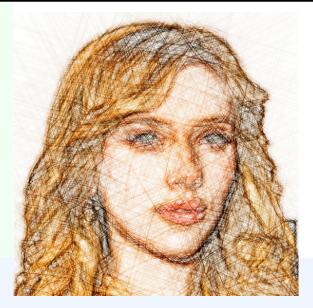






◆□▶ ◆□▶ ◆ □ ▶ ◆ □ ▶ ◆ □ ● ● ● ● ●







• Can be used to simulate several different drawing steps towards a final painting.



Step 1 (straight strokes)



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ



Step 2 (curved strokes)

Some results



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ



Step 3 (curved strokes + colorization)

Some results



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ



Step 4 (curved strokes + colorization)



(ロ)、



Final painting (the only input image of the algorithm !)

Some results



(ロ)



(Colors do not have to match the input image)



 Several sketch results can be also merged together to render even more complex rendering (courtesy of Tom Keil).





 Several sketch results can be also merged together to render even more complex rendering (courtesy of Tom Keil).



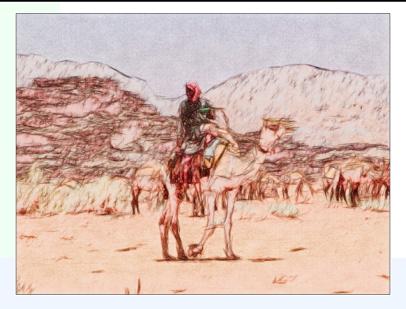








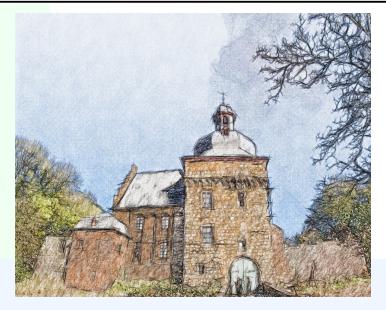














• Conclusions :

We have proposed a B&W stroke-based algorithm for simulating the drawing of lines, to convert photographs into sketchs.

- It uses tensor fields to deal with the geometry both of the image and the strokes.
- It defines a way to draw strokes according to the geometric shape of the tensor field.

Perspectives :

- Find the best ways to model the stroke tensors according to a given artistic style.
- 2 Improve the colorization step.
- Model the thickness of strokes.
- Deal with image sequences (temporal coherence).

Do it yourself !



• The algorithm code is available in **G'MIC**, a powerful framework for image processing, developed at the GREYC/Image lab.

http://gmic.sourceforge.net

• Try our sketch algorithm, with the G'MIC plug-in for GIMP.

Prive > Arrays & frame > Arrays & frame > Arrays & frame > Arrays & frame > Stating angle > Black & white > Stating angle > Black & white > Stating angle > Diractal > Stating angle > Charcoal > Stating angle > Procil Name > Convort Stroke > Dogradations > Patterns > Patterns > Uptut Procis > Name > Patterns > Tating > Convort Stroke > Convort Stroke > Convort Stroke > Convort Stroke > Convort Stroke > Convort Stroke Convort	G'MIC for GIMP - 1,5,0,1 x					
Arrys & frame Arrys &	Letore	Available filters (259)	Sketch :			
> Artistic Starting angle 4.00.0 > Artistic Starting angle 4.00.0 > Artistic Angle range 100.0 > Churcol Starting angle 100.0 > Churcol Pencli Opacity 0.00.0 > Pencli Name 0.00.0 0.00.0 > Mission Pencli Opacity 0.00.0 > Pencli Nampless 0.00.0 0.00.0 > Pencli Starting angle 0.00.0 0.00.0 > Pencli Nampless 0.00.0 0.00.0 > Contours Pencli Startpress 0.10.0 > Contours > Degradations Conternistics 0.00.0 > Layers > Degradations Conternistics 0.00.0 > Layers > Layers 0.00.0 1.00.0 > Layers > Patterns Patterns Conternistics 0.00.0 > Layers > Patterns Tartistic Conternistics 0.00.0 > Layers > Tartistics Conternistics Conternistics 0.00.0 > Layers > Tartistics		2010	Number of orientations		2	
Proteine	1 and 1		Starting angle		45.00 🗘	
Orbarcoal Solder intgin Solder intgin <th>1-90 AC</th> <th></th> <th>Angle range</th> <th></th> <th>180.00 😜</th>	1-90 AC		Angle range		180.00 😜	
Sterici Contrau threshold 100 G Opacity 000 Contraut threshold 100 G Opacity 000 Contraut threshold 000 Contraut	CONT X MA		Stroke length		30.86	
Percil Percil Intravisity Intravisity Hough stetch Hough stetch Hough stetch Hough stetch Preview Intravisity Input / Output : Intravisity Input : Intravisity		7 -	Contour threshold		1.00	
Ink wash Background intersity 0.00 ° Hordy's ketch Hordy's ketch Dersity 1.02 ° Y Broker > Colors Anisotropy 0.00 ° Y Broker > Colors Anisotropy 0.00 ° Iput / Output : > Deformations Octors Smoothress 0.25 ° Input / Output : > Deformations Porteresc 1.00 ° Dogs table Output mode > Degradations Patterns Octors table Color mode Dogs table Output mode > Layers > Layers Color mode Black on white Color mode Output preview > Patterns > Inner: Over Broken quettal Anisotropy Color mode Output preview > Patterns > Color mode Black on white Color mode Output preview > Preview tyce Internet Aniver: Over Broken quettal Color mode Black on white Color mode		8	Opacity			
Hard sketch Density 102 °S Starpness 0.10 °S Perview > Anisotropy Perview > Octoors Portrainations Controns Sonothress Input Joyens > Degradations Output mode > Degradations Output mode > Degradations Output mode > Degradations Patterns > Patterns Patterns > Perview			Background intensity		0.00	
Imput / output : > Colors Sharpness 0.10 € // preview > Colors Anisotropy 0.80 € // preview > Deformations Smoothvess 0.25 € // preview > Deformations Smoothvess 0.25 € // preview > Deformations Boots stroke 0.00 € // preview > Layers O drow from dell Boots stroke 0.00 € // preview > Parterns Preview type Conved writical 0 // preview > Preview type Forward vertical 0 Amisr : David throumperf. Latest update: :2010/22/9		-	Density			
> Colors > Colors > Perview > Deformations Input / dupt: > Deformations Input / syers > Enhancement Output review > Deformations > Deformations > Boost stroke Output preview > Patterns > Preview Type Forward vertical Output preview > Preview Type Officience > Interns Preview Type Convert Router Rout Router Officience > Interns		Sketch	Sharpness			
Preview		KOP V COIOIS				
Input / Output : Input / Output : Output mode		Contours				
Input, Byess > Boost stroke Output messages > Layers Output messages > Patterns Precised Strating Precised Strating Aumor : David Inchumpent. Latest update : 2010/22/9 Aumor : David Inchumpent. Latest update : 2010/22/9	Input / Output :		Coherence			
Output message Uights & Shadow Uights & Shadow Output preview > > Parterns Preview type Forward vertical © Aumor : David Tichumpent. Laest update : 2010/22/9	Input layers 0				1.00	
Output messages Color model Black on white Color model Output preview Output preview Forward vertical Color model Color model Output preview Amer: Down Roumpersk. Lasers upskie Color model Color model	Output mode 0		Curved stroke			
Output preview	Output messages 0		Color model	Black on white	0	
	Output preview 0		Preview type	Forward vertical	0	
	Author : David Tschumperié. Latest update : 2010/12/29.					
			& Cancel	Beset Maximize Apply	 Дек 	

Questions?



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ○臣○



(Don't shoot the speaker !)