Image Stylization using Structure Tensor

Lucas Foucault, Ific Goudé

***Résumé*—We used the work of a former student based on the calculation of pictures' tensors. Those tensors are then reused in order to style the picture. Using this method, we convert the tensors into lines, creating a specific sort of picture imitating artists' pencil's strokes. Then, we refined his work by adding new styles. First we used the structure tensors in order to make a low poly effect on the pictures. To this end we used triangulation methods like Delaunay's or Voronoi's, consisting of dividing the image in several polygons and coloring each of them by the dominant color of the area where they are located. Then, we created a style involving to scatter the particles on the picture to make them follow a flow determined by the structure tensors. Every particle leaves a trace of its passage drawing the different shapes constituting the picture.**

***Index Terms*—Stylization, Structure Tensor, Stroke, Low Poly, Particules Flow**

I. Introduction

THANKS to Remi Morillon work, a newly ESIR graduate, we had the basis of the image style. Indeed, his program computed the structure tensors present in the picture and reused them in order to draw strokes in the desired locations **[1]**. Our goal was to improve Remi's work by adding new styles different to those tapped by our predecessor. Of course, we have also adjusted the architecture of the code. We have developed two new features : the polygon based style and the flow of particle based style. This document purpose is to explain how to compute those tensors which represent the foundation of our different applications. Then to show the way the program works to design the three different image styles. In order to exemplify our results, we'll use our processing on the pictures figure 1.

Those two pictures have specific characteristics. Lena.jpg poses irregularities in both fore and background whereas Albert.jpg is more homogeneous which would allow a better detection of the contours. Regarding Egypt.jpg, the shapes in background and the two de details in foreground might lead one to believe that the stylizations could be interesting.

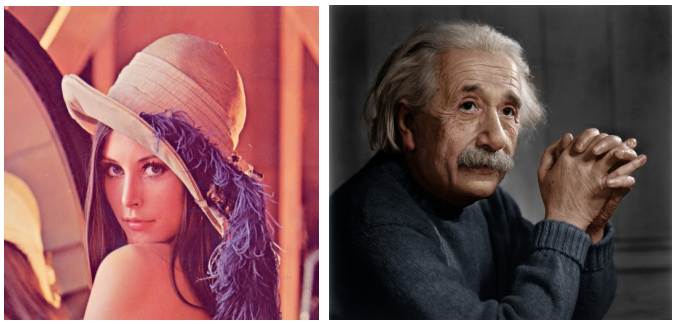

FIGURE 1. From left to right : Lena.jpg, Albert.jpg

II. Structure tensor computation

A Structure tensor is a second-moment matrix derived from the gradient of a function. It represents the directions of the gradient taken the contribution of a specific neighbourhood. The structure tensor is composed of the squared partial derivation of the gradient in x, the squared partial derivation of the gradient in y, and the multiplication of the partial derivation of the gradient in x and y. As the structure tensor is positive and symmetric, the eigenvalues and eigenvectors exist. Larger eigenvalues are, stronger the local image edge is. One of the eigenvectors represents the direction of the gradient (normal to the edge) and the other, called isophote, is in the direction of the edge.

$$S_\sigma = \Big(\sum_{i=1}^{n} \bigtriangledown_i I \times \bigtriangledown_i^T I\Big) * G_\sigma$$

$$\lambda\pm = \frac{g_{11} + g_{22} \pm \sqrt{\triangle}}{2}$$

$$\theta\pm = \begin{pmatrix} 2g_{12} \\ g_{22} - g_{11} \pm \sqrt{\triangle} \end{pmatrix}$$

$$\triangle = (g_{11} - g_{22})^2 + 4g_{12}^2$$

FIGURE 2. S : Smoothed 2D structure tensor, lambda : Eigenvalue, theta : Eigenvector

The eigenvalues can be used to compute a norm (different norm exists, representation figure 3). The value of this norm gives us the strength of the edge at the current pixel. If this value is upper than a fixed threshold, it is considered as an available edge. In our process, the first step is to compute a structure tensor map of the image in input. For each pixel, we obtain the associated eigenvalues and eigenvectors.

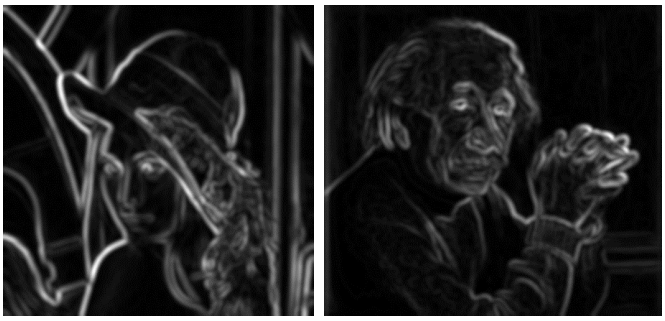

FIGURE 3. Structure tensor map, here the norm is represented.

For each eigenvalue we compute the norm and make the associated histogram. In this way, we can fix simply the number of points we want to conserve. We fixed the percent of point we want to draw in the image and the threshold is defined in function. After that, a random selection is done to create a non-determinist stylization (figure 4). If the norm of the selected point is upper than the defined threshold, we can draw ; else, it is considered as a smooth area.

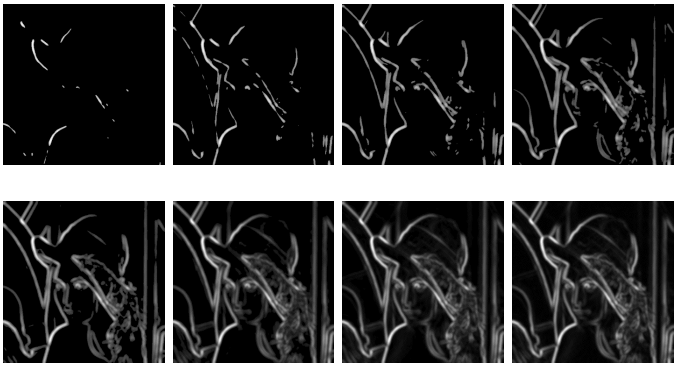

FIGURE 4. Valid points representation for different percent from left to right and up to down : 1%, 5%, 10%, 20%, 30%, 50%, 80%, 100%

III. STYLIZATION BASED STROKE

THIS is the basis style in stylization. This part will not be precised in details because it is not the fruit of our work, this method was mostly implemented by our colleague Remi. This is the basis style in stylization. The simple idea is to draw lines on the edges. For a valid point selected randomly, we have the associated structure tensor. We said before that the isophote gives us the direction of the edge, we simply need to draw a line in this direction to obtain a stroke. Multiple variations are inspired by this method. First, a simple black stroke with defined thickness and length for each valid point. The color can be added reusing the color of the same pixel in the original image. Rather than using simple line, brush effect can be drawn too. Curved stroke are also implemented. Inspired by Bezier's curves, those strokes give a more artistic effect. Finally, the pyramidal stroke wants to accentuate strong edges. This last has to be more explain. A Gaussian pyramid is used to down sample the image, it produces a low frequency image, and then just the strongest gradients are conserved. We apply the algorithm on the down sampled image to draw strong edges, and we repeat this process for each level of the pyramid. It produces a charcoal style (figure 5 and 6).

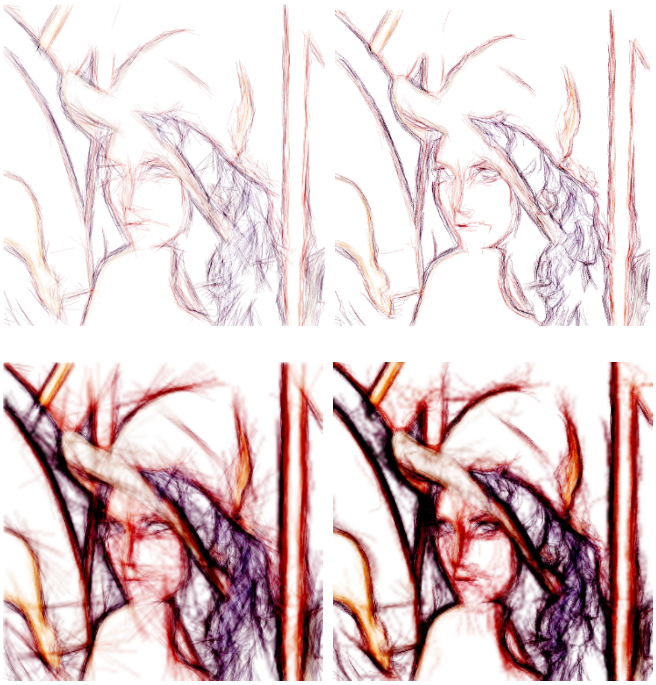

FIGURE 5. Style description from left to right and up to down : straight stroke, curve stroke, pyramidal straight stroke, pyramidal curve stroke

IV. STYLIZATION BASED LOW POLY

WE will now talk about the stylization based on Low Poly. To understand and develop this

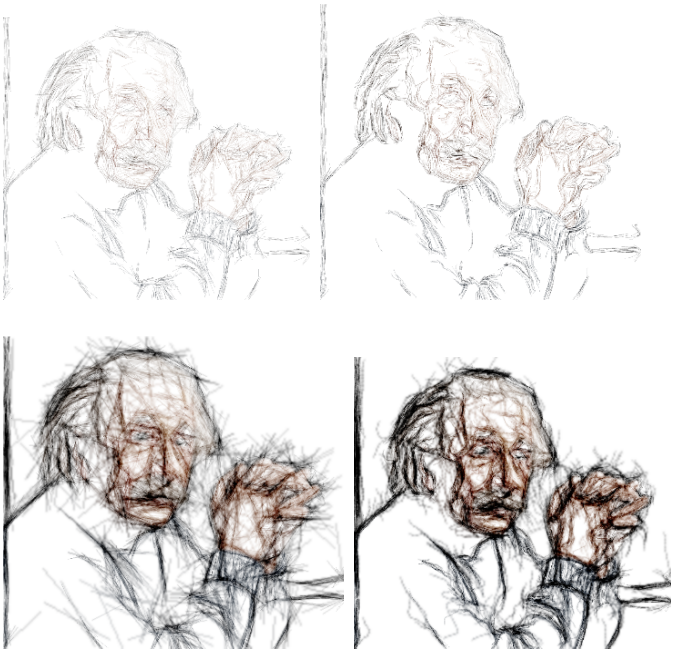

FIGURE 6. Style description from left to right and up to down : straight stroke, curve stroke, pyramidal straight stroke, pyramidal curve stroke

FIGURE 7. Style description from left to right : delaunay, voronoï

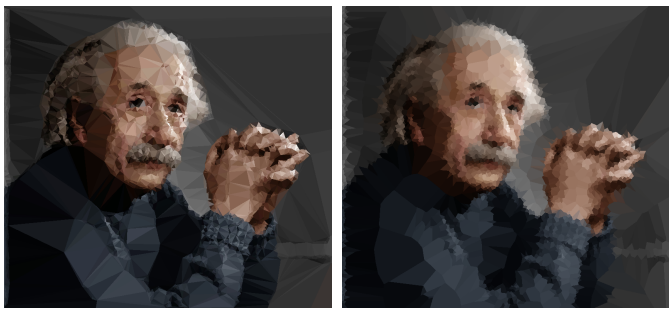

FIGURE 8. Style description from left to right : delaunay, voronoï

style we took an interest in triangulation. Indeed, in order to observe this aspect of the picture less detailed, we must divide the picture in several polygons. In order to do this, we must identify the dots we need to save, those key dots must set the picture in the best possible way. To choose the significant dots we must calculate the structure tensors, we call them ValidPoints. We then reuse those points in order to create a rectangular subdivision representing the picture called Subdiv2d on OpenCV. Thereafter, depending on the method used, there are two different technics : the Delaunay's triangulation or the Voronoi's diagram. We can easily apply these two methods with OpenCV. Regarding Delaunay's triangulation, thanks to the previous subdivision, we can get a distribution of the triangles in the picture, thus build a list of triangles. Each one is then filled with the average color of the dots it contains. With reference to Voronoi's diagrams, the obtained polygons are not necessary triangles but facets. Then we must build a list of polygons and fill theme the same way as the triangulation method. Here are some results obtained with this stylization figure 7 and 8.

V. STYLIZATION BASED PARTICLES FLOW

WE will now talk about the last created style : the stylization based on the flow of particles. To make it possible we had to develop a new class simulating a particle's behavior because this stylization uses particles in order to draw the main strokes of the picture. Like the stylization based on strokes, this method calculates the structure's tensors in order to find the direction of the picture's contours given by the isophotes. More specifically, the program works in this way :

1) Initialize a specific number of particles with random positions on the picture (the number is chosen by the user before the launch).
2) Chose the essential isophotes to get a result (you must not chose the whole isophotes obtained with the tensors calculus because the operation will become too long in regards to the final result).
3) Move each particle one by one by giving it a power making it run to the closest isophote, increasing its velocity this way.
4) Ensure that none of the particles gets out of the picture, if this happens make it reappear on the opposite side.

5) Display a line between the new calculated particle's point and its previous position. This line can be colored with the mean shade between the two dots in the picture.

This way, the flow of particles appears, each particle following the way given by its nearest isophote.

Here are the different results achieved thanks to this algorithm figure 9 and 10.

FIGURE 9. Style description from left to right : particles flow based on interval tensor, particles flow based on main tensor

FIGURE 10. Style description from left to right : particles flow based on interval tensor, particles flow based on main tensor

VI. CONCLUSION

TO conclude we saw how to use structure tensors to get the key points and applied different styles using them. The first part of our work was about understanding the algorithm already written by our colleague and improve it by changing some feature. The main modifications were about defining threshold depending on a percentage of wanted points and improving the render quality of the pyramidal style. After that, we would use those points in a different way. The Delaunay's triangulation appears obvious to us. It really changes the application of the stylization at the moment. Voronoï's diagram follows logically our main idea of a low poly style. The particles' flow comes later with the inspiration of the procedural generation of a perlin noise and the way particles could navigate in an image. The result was unexpected and we enjoy to present this style in this paper.

RÉFÉRENCES

[1] David Tschumperlé, *Tensor-directed simulation of strokes for image stylization with hatching and contours*. 18th IEEE International Conference on Image Processing, Sept 2011.