Extraction of a Cartoon's Topology

Morgane Rivière * — The University of Tokyo
Makoto Okabe † — The University of Electro-Communications

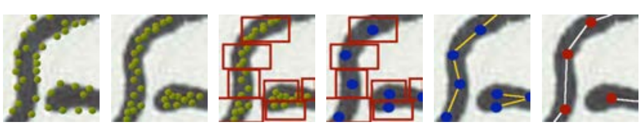
Abstract

The vectorization process transforms an image in the algebraic representation of if its contours. In the case of hand-drawn cartoons, the pen stroke made by the artist defines such a contour. That's why drawing can be interpreted as a series of junctions between nodes, or, in other words, as a topological graph.

Many softwares tackle this subject (Adobe Live Trace, WinTopo...). We will focus here on a program developed in 2013 by a team of researchers from the ETH Zurich and Disney Studios [Noris et al. 2013]. Their method was very efficient for solving ambiguous cases in the drawing, however their implementation was very slow : the vectorization of a 2048x2048 cartoon could need more than 3 minutes of computation. We have improved their method achieving interactive speeds.

1 Method

Our goal is to compute the topological graph corresponding to a given cartoon. We expect the output result to be faithful to the original drawing on the one hand, but, on the other hand, we want it to have as few nodes as possible in order to be easily modifiable by the user.
First, we compute the gradient of the original raster. Then, we select specific points in the image whose gradient and intensity are above a fixed threshold, then we make them move along the gradient lines (1(a) and 1(b)). This way, we obtain a first scatter plot : we lighten by computing the average of its point on a smart shifting window (1(c)). Using the resulting plot, we compute local Minimal Spanning Trees (MST): we divide the raster in several small windows and compute the MST corresponding to our scatter plot within this window (1(e)). However, we forbid edges which cross more than a certain amount of white pixels. Those local MST will prevent the final graph from having unnecessary micro-cycles. We link those different local MST together: at most one edge can make the junction from a local MST to another, and all edges must be smaller than a fixed limit. Finally, we smooth the result and discard unnecessary nodes (1(f)).



(a) (b) (c) (d) (e) (f)

Figure 1: *Overview of the vectorization process*

*morgane.riviere@gmail.com
†m.o@acm.org

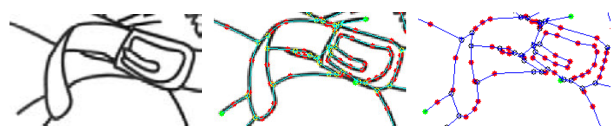
SIGGRAPH 2014, August 10 – 14, 2014, Vancouver, British Columbia, Canada.

ACM 978-1-4503-2958-3/14/08

The greedier step of the method is the computation of the optimal graph. Our improvement mostly holds in two key points : the reduction of the number of nodes, and a simplier method to compute the graph. Indeed, the original method computed the global MST for the scatter plot and refined the results afterward by generating several smaller MST. Here we do the contrary : we start by computing local MST and then we merge them.

2 Results

As far as speed is concerned, our method is efficient : the vectorization of a noisy 2048x2048 raster takes about one second rather than 3 minutes. Besides, the fewer the nodes are, the easier it will be to change the output : on our examples, we generate 30% less nodes than Adobe Illustrator and a about 30% more nodes than WinTopo. Furthermore, our method is faster than Adobe Illustrator on our test samples.



(a) (b) (c)

Figure 2: *(a) The original raster ; (b) comparison with the output graph ; (c) the output graph*

But the main stake is the precision of the output. Our method provides satisfying results in most cases but fails to capture small details where local MST are no longer a correct model. To solve this problem, we could detect areas necessitating a specific handling by computing the Histogram of Orientied Gradients (HOG) on the raster. Indeed : ambiguous zones are charaterized by pen strikes going in almost all directions and so, the local gradient is remarkable.

3 Application

Our algorithm produce a graph which can manipulated by any user. This output is specific to our method. With this graph, the artist can create a large panel of short animations without altering the topology of his drawing. The next stage of our work will be to build the platform to design such animations.

References

NORIS, G., HORNUNG, A., SUMNER, R. W., SIMMONS, M., AND GROSS, M. 2013. Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics (TOG) 32*.