SIGGRAPH'94 Technical Sketch Submission

Automatic Gap Closing for Freehand Drawing

Michel Gangnet
Jean-Manuel Van Thong

{gangnet, jmvt}@prl.dec.com
Digital Paris Research Laboratory
85, avenue Victor Hugo
92500 Rueil Malmaison, France

Jean-Daniel Fekete

jdf@two-oo-one.fr
2001 S.A.
2, rue de la Renaissance
92184 Antony Cedex, France

Summary

Automatic closing of the regions defined by freehand drawn strokes is mandatory for cel coloring in computer-assisted cartoon animation. We present a gap closing method based on connected component analysis inside square stamps located near potential gaps. This method is currently in industrial use.

If accepted, we will show a video tape.

1 Problem Description

TicTacToon is a vector-based paperless system for computer-assisted 2D cartoon animation developed by 2001 S.A. in collaboration with Digital PRL [1]. This technical sketch explains how we solve the gap detection and closing problem in TicTacToon's cel coloring application.

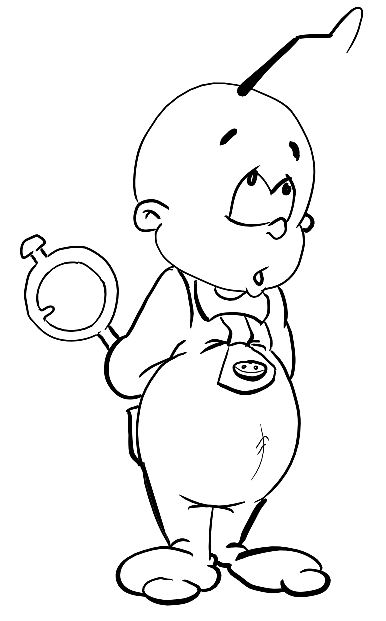


Fig. 1: A cel.

Artists draw with a pressure sensitive pen on a digitizing tablet, as in fig. 1. Using a new curve fitting method [2], the digitized points are used to compute a variable width stroke whose centerline and boundary is described by Bézier curve paths.

After cleaning, each drawing is prepared for coloring by computing the *planar map* defined by the centerlines [3]. A planar map is an object that contains a description of the regions, edges, and vertices, formed by a set of paths. Planar map methods allow the incremental insertion and removal of paths, and point location.

Coloring is thus very simple: on a mouse click, perform point location and set the color of the resulting region. However, strokes have width and seemingly closed regions may turn out to be actually open because of invisible gaps, as in fig. 2. Besides, small visible gaps should also be closed.

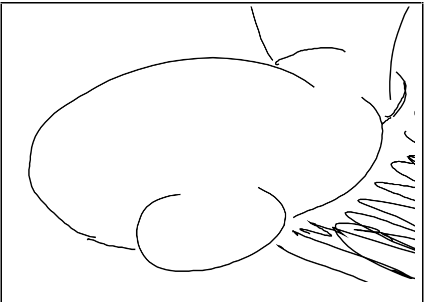
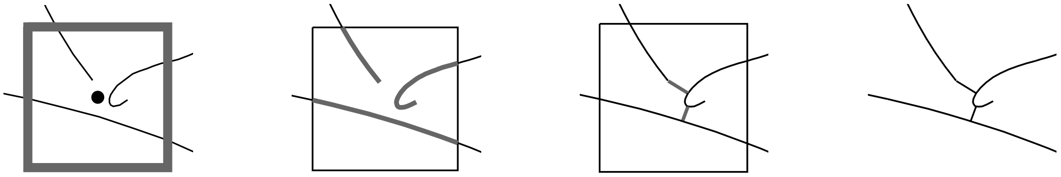


Fig. 2: Stroke centerlines and gaps.

What we need is a method to detect and close the gaps before the interactive coloring application is invoked. It is important to note that this problem is defined in object space, as opposed to image space where we could use a variation of the seed fill algorithm [4] to detect the gaps, e.g. by analyzing the boundary of the growing seed region.

2 Solution

First, we search the planar map and build a set of points which are candidates for gap detection. In our implementation, path endpoints and local x or y extrema are candidates.



Step 1. Step 2. Step 3. Step 4.

For each candidate point c, insert a square p with center c in the planar map (step 1); we call p a *stamp*. The stamp size defines the threshold gap size. For each region r interior to p, consider the set E of map edges which are not part of p and build the set C of E's connected components (step 2). Initialize a set of line segments, called *staples*, S to empty. While C has more than one element, add to S a line segment connecting the closest pair of components in C and update C. When done, insert all elements of S in the planar map (step 3). Clean up the interior of p, that is, remove very small regions and dangling edges smaller than a given size. Remove the square p from the map (step 4).

This algorithm is fast and efficient because stamp insertion in the map allows analyzing small areas where only a manageable subset of the map edges needs to be considered. In most cases, there are only two connected components per stamp. The method is also used by the *stapler tool* in the interactive coloring application. On a mouse click, we staple the connected components found inside the stamp centered at the click position. When the cel is rendered, staples are used to limit a region but are not drawn.

Note that this method can be easily extended to the image case: use component labeling within a square pixmap and draw line segments between the closest components.

3 References

[1] "TicTacToon Reference Manual V1.0", 2001 S.A., February 1994.

[2] T. Pudet, "Real Time Fitting of Hand-Sketched Pressure Brushstrokes", to appear in *EUROGRAPHICS'94 Conference Proceedings*, September 1994.

[3] M. Gangnet, J-C. Hervé, T. Pudet, J-M. Van Thong, "Incremental Computation of Planar Maps", *SIGGRAPH'89 Conference Proceedings*, *Addison Wesley*, July 1989.

[4] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, "Computer Graphics, Principles and Practice", *Second Edition, Addison Wesley*, 1990.