LETTER

Cartesian Resizing of Line Drawing Pictures for Pixel Line Arts

Zhongying HU¹, Nonmember and Kiichi URAHAMA(¹)a, Member

SUMMARY We propose a method for downsizing line pictures to generate pixel line arts. In our method, topological properties such as connectivity of lines and segments are preserved by allowing slight distortion in the form of objects in input images. When input line pictures are painted with colors, the number of colors is preserved by our method.

key words: pixel line art, line drawings, Cartesian resizing, topological property

1. Introduction

Making pixel arts requires laborious work by skilled artists. Non-photorealistic rendering (NPR) techniques for automatically generating them from photographs or pictures have recently been presented. Approaches [1]–[3] of downscaling color photographs are useful, but they need color quantization and edge detection by image filters which are hard to work stably. Hence an approach simplifying the NPR problem by limiting input images to line drawings has been proposed [4]. We, in this letter, also consider the downsizing of line pictures. Inglis and Kaplan [4] assumed vector images to be inputted, hence their method cannot be applied to bitmap images made by paint software. They attempted to maintain the form of objects while the topological properties of images such as connectivity of lines and segments are not preserved in their method. On the contrary, in this letter, we allowed deformation of objects while preserving the topology of lines and segments. This preservation of the topology of lines and segments is needed if input line pictures are painted with colors. In order to evaluate the topology of lines and segments, our method deals with bitmap images. Similar method for topology-preserving downscaling of binary images has been presented [5], but our method is different from it. In hand-made pixel arts, deformation of objects is commonly observed, as occurs in our method.

2. Connectivity of Lines and Segments

We deal with bitmap images with black lines drawn on a white background. The connectivity of pixels is evaluated in 8-neighborhood for black pixels and 4-neighborhood for white ones. For instance, in Fig. 1, there are four black lines labeled $B_1$ to $B_4$ and four white regions labeled $W_1$ to $W_4$ ($W_1$ is the background).

We consider downsizing of such line pictures. When this image is colored as the right one in Fig. 1, if the circle labeled $B_2$ is shrunk to only one pixel, then the red region disappears. Similarly, if the upper-right two lines are adhered to the line below them or lines are cut or disappear, the image is changed to a different one. Such topological change occurs in the method by Inglis and Kaplan [4].

3. Downsizing Procedure

We downsize input line pictures in a similar way of the Cartesian resizing of images [6]. First, we label every connected component, i.e. lines and regions in an input image by 8-neighborhood labeling for black pixels and 4-neighborhood labeling for white ones.

In the Cartesian image resizing method, columns and rows are sequentially selected and removed. Candidates for removal are columns and rows including less black pixels. In this procedure, simple removal of the center column in Fig. 2 (a) leads to Fig. 2 (b) where the black line is cut. To fill this gap for keeping the line connectivity, we copy the center black pixel to its left one and then remove the center column to get Fig. 2 (c). This cut of black lines occurs if the neighborhood around the center pixel matches one of nine patterns in Fig. 3 where gray pixels are possibly black or white, i.e. it is don’t cared. If a black pixel on the candidate column matches one of patterns in Fig. 3, that pixel is copied to its left and removed. Another case requiring copy of black pixels is the patterns in Fig. 4 where black pixels continue below the center. In this case, all of continuing...
black pixels are copied to their left-side. This operation of copying black pixels runs in parallel, i.e. a copying operation of some pixels does not affect other pixels in the same column/row. Accompanying this copy of black pixels, their label is also copied. After copy of labels, we check label connectivity. If different labels are fused by this column removal, we abandon this removal and proceed to the next candidate column of removal.

Similar check is needed for white pixels. Removal of the second rightmost column in Fig. 5 (a) with copying black pixels leads to Fig. 5 (b) where the white pixel at the second column from the right and the third row from the top becomes isolated because the connectivity of white pixels is evaluated with 4-neighborhood. This isolation of white pixels can be found by matching the $3 \times 3$ block to patterns in Fig. 6. If a white pixel on the column matches one of these patterns, we abandon the removal of that column and proceed to the next candidate column. Additionally, if white pixels of different labels are fused by the column removal, we also do not remove such column and go to the next candidate.

These operations are summarized to the following procedure for the reduction of the image size:
1) Count the number $n_i$ of black pixels in each column $i$. Add a random number $r_i$ in $[0,1]$ to $n_i$ and attach the weight $n_i + r_i$ to each column $i$.
2) Select the column of the least weight. If the $3 \times 3$ block around a black pixel on this column matches one of pattern in Fig. 3, copy that black pixel to its left one. If the block matches one of patterns in Fig. 4, copy all pixels under the center to their left ones. After the copy of these black pixels, if one of patterns in Fig. 6 appears or the labeling of images changes, i.e. a label disappears or different labels are fused, abandon the removal of this column and increase the weight of the column to $2c_i$. Go to the column with the next least weight.
3) Increase the weight of two neighbors of the removed column to $2c_{i-1}$ and $2c_{i+1}$.
4) After shrinking the image width by one, remove one row from the image for reducing its height by one.
5) Repeat steps 2~4 until the width and height of the image reach the desired size.

4. Experiments

We have experimented this method for eight images in Fig. 7 which are reproduced from the paper by Inglis and Kaplan [4] by kind permission from the Eurographics Association. Their sizes are $108 \times 108$ for the upper-left one and two lower-left ones, and $128 \times 128$ for the remaining five pictures. We set their reduced size to the same as those in [4] where the upper-left two and lower left two are reduced to $20 \times 20$ and the remaining four are reduced to the size $30 \times 30$ except for only the third one in the bottom (T-rex) is reduced to $33 \times 33$ which is the lowest limit of our method for preserving the connectivity of segments. The results of the above downsizing procedure are shown in Fig. 8. Though some objects are deformed from Fig. 7, the labels of black pixels are preserved.
These results of our method are compared with other methods in Figs. 9 and 10 where the leftmost column is results shrunk by the Adobe Illustrator, the second column is by the Adobe Photoshop, the third column is by Inglis and Kaplan [4] and the last rightmost column is by our method. While other methods convert vector images to raster ones, our method converts raster images to raster ones, hence they deal with a different problem. Some topological errors are observed in the previous methods such as line breaks (e.g., left two images in the bottom of Fig. 9), line adhesion (e.g., eyes and face contours in the left three images of the bottom in Fig. 10), shrinking of white regions to black pixels (e.g., teeth in the third image at the third row and the eye in the second image at the bottom row in Fig. 10). These errors lead to problems if these line pictures are painted with colors. Some colors disappear in such cases while no problem occurs in our method where colors are kept as is shown in Figs. 11 and 12.

5. Conclusion

We have proposed a method for downscaling line pictures. Contrary to the method by Inglis and Kaplan where topological properties are not preserved, our method keeps every line and region in the pictures. Hence line pictures of the original size can be restored from small-sized images obtained by our method if object deformation is allowed, while restoration is impossible in the previous methods. This is the benefit of our method, but object deformation is its deficiency. Improvement for reducing object deformation is under study.

References

[1] K. Suzuki, S. Saito, Y. Chang, K. Kondo, and M. Nakajima, “Pixel art generation from photograph,” IEICE Trans. Inf. & Syst. (Japanese Edition), vol. J91-D, no.12, pp. 2961–2972, Dec. 2008.
[2] H. Takimoto, S. Yoshimori, and Y. Mitsukura, “Method for automatic generation of pixel art based on color-difference tolerance,” J. ITE, vol.66, no.11, pp. 399–406, 2012.
[3] T. Gerstner, D. DeCarlo, M. Alexa, A. Finkelstein, Y. Gingold, and A. Nealen, “Pixelated image abstraction,” Proc. NPAR, pp. 29–36, 2012.
[4] T. Inglis and C.S. Kaplan, “Pixelating vector line art,” Proc. NPAR, pp. 21–28, 2012.
[5] L.R. Morales-Manilla, G. Sanchez-Diaz, and R. Soto, “An image resizing algorithm for binary maps,” Proc. ENC, pp. 126–132, 2004.
[6] X. Fu, K. Inoue, and K. Urahama, “Improvement of photo composition by partitioned Cartesian resizing of images,” J. ITE, vol.64, no.12, pp. 1960–1963, 2010.