The Visual Computer
https://doi.org/10.1007/s00371-021-02268-2

ORIGINAL ARTICLE

Color2Hatch: conversion of color to hatching for low-cost printing

Ryoma Isumi[1] · Kunio Yamamoto[2] · Tsukasa Noma[2]

Accepted: 15 July 2021

Abstract
In this paper, we propose Color2Hatch, a decolorization method for business/presentation graphics. In Color2Hatch, each region represented as a closed path and uniformly colored in scalable vector graphics (SVG) is converted to a region hatched in black and white. From the characteristics of business graphics, the hatching patterns are designed to represent mainly the hue in the region; additionally, lightness and saturation can also be reflected. To discriminate subtle differences between colors, attached short line segments, zigzag lines, and wave lines are used in hatching by analogy to a clock. Compared with the existing decolorization methods, for example, grayscale conversion and texturing, our method is superior in the discrimination of regions, suitable for low-cost black and white printing that meets real-world needs.

Keywords Decolorization · Color to hatching · Vector image processing · Low-cost black and white printing

1 Introduction

Printing on paper remains an important medium for communication despite the recent development of electronic devices; for example, smartphones and tablets, and even now, printing in black and white (B/W) are widely used. To include color figures and graphs in B/W printing, they should be converted to black and white.

A standard method for B/W conversion is color-to-grayscale conversion via a lightness channel. However, with this standard method, chromatic information is entirely ignored, and thus, information in the original figures/graphs is often lost.

Two approaches have already been proposed to solve this information-loss problem: contrast-enhancing color to grayscale conversion [1–14] and color-to-texture conversion [15–20]. Moreover, to keep the information converted by both approaches, the printing media should be of high quality so that the grayscale levels or fine textures are properly reproduced.

To make the B/W conversion more robust and usable even for low-quality printing, this study focuses on hatching, a technique of drawing parallel lines in close proximity to represent shading effects.

In this paper, we propose a conversion method from color in scalable vector graphics (SVG) figures to new extended hatching, representing color differences very well. While conventional hatching with parallel straight lines is inadequate for a variety of colors, our extended hatching with, e.g., lightning-like zigzag lines, can represent the differences in the original color even if the original tone is mostly lost.

This paper is organized as follows: Sect. 2 overviews the background of the study with related research. Section 3 provides an overview of the proposed approach. Section 4 elaborates on our approach and introduces some variations in our approach. The experiments are presented in Sect. 5, and Sect. 6 discusses the printing cost savings compared with existing methods, the limitations of our method, its applications, and its relation to visualization. Finally, Sect. 7 concludes the paper with some comments on future work.

2 Background and related work

To display color images on monochrome devices, the images are conventionally converted to grayscale images by lightness. For example, in ITU-R Recommendation BT.601,

✉ Tsukasa Noma
noma@ai.kyutech.ac.jp

[1] Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502, Japan

[2] Department of Artificial Intelligence, School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502, Japan

Published online: 10 August 2021

YUV (typical YUV), luma (gamma-corrected luminance) *Y'* is given by:

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

where R', G', and B' are the red, green, and blue signals.

However, with this standard method, chromatic information is entirely ignored, and thus, colors of similar lightness are indistinguishable even if their hues are distinguishable.

To preserve or enhance contrasts between colors of similar lightness, many authors have proposed decolorization methods. Bala and Braun [1] assigned distinct gray values to colors in business graphics. Gooch et al. [2] optimized the modulation of source lightness differences to reflect the gap in chrominance. Rasche et al. [3] extended the decolorization problem to reduce the gamut dimensions for color-deficient viewers. Grundenland and Dedgson [4] improved decolorization efficiency. Smith et al. [5] and Kuhn et al. [6] proposed other fast conversion methods. According to Gestalt principles, Wu et al. [7] decomposed the input image into soft segments, and Ji et al. [8] considered the boundary instead of segmentation. Cui et al. [9] used ISOMAP, a nonlinear dimensionality reduction method, and Zhu et al. [10] applied Broadbent's filter model of attention to the color-to-gray conversion. Recently, efficiency and contrast preserving properties have been improved (e.g., [11–14]).

Another approach for color-to-monochrome conversion is texturing. Roetling encoded eight basic colors as textures from unique RGB texture patterns for B/W dot printing [15]. Harrington mapped colors to visible B/W textures, where similar colors were mapped to similar textures [16, 17], and Bai et al. [18] improved the texture patterns. Queiroz and Braun [19] proposed a reversible method in which colors are mapped to gray textures and the color information is recovered from the texture. Guss used distinct line directions for RGB and represented colors using B/W textures [20].

In our view, neither of the above two approaches, grayscale conversion and texturing, are in practical use, particularly for low-cost B/W printing. Both these approaches rely on high-quality printing, where shading representation and pixel-by-pixel on/off should be accurate. These properties are not expected for conventional B/W printing.

There are two types of color images: pictorial images and business/presentation graphics images [17]. This study focuses on the latter, and we assume that each region in the image is uniformly colored. In color business graphics, hue plays a major role in discriminating regions, compared with lightness and saturation. Even if the image is converted to B/W, each colored area should be identifiable, as in color. The existing methods of grayscale conversion and texturing are inappropriate for this purpose.

We then focus on hatching. Hatching is an artistic technique for drawing closely spaced parallel lines. It is visually clear and superior in discriminating between regions. Existing hatching techniques in graphics community mainly reflect shapes, contours, shading, etc. (e.g., [21–23]). In technical drawings, the material of a part is indicated by hatching [24]. To the best of our knowledge, conversion from color to hatching has not yet been proposed.

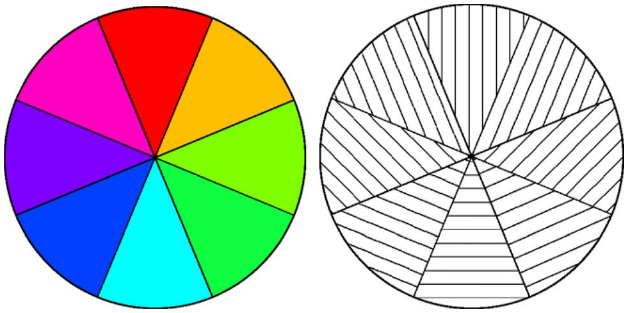


Fig. 1 Hue and hatching angles

As discussed above, hue is most critical in color business/presentation graphics and can be mapped to the direction of hatching lines (Fig. 1). If the difference between the hues of the two regions is large, discrimination is easy. For similar colors, however, the direction of the hatching lines is not sufficient for discriminating many regions.

In this study, we propose a conversion method from color to hatching, where even regions of similar colors are identifiable.

3 Overview

To improve the discriminability between hatching patterns from similar colors (hues), we focused on clocks (timepieces). Analog clocks display time according to the directions of their hands in 12 h. If they have only a single hand, the displayed time is ambiguous. Clocks usually have two hands: long (minute) hand and short (hour) hand, and the pair of hands enables us to identify time accurately in 12 h. This analogy can be applied to hatching.

We supposed that input images were in SVG format [25], where each region, for example a country in a world map or a slice in a pie chart, is represented as a closed path and colored uniformly. We then filled each region with a hatching pattern corresponding to its original color.

As discussed in the previous section, the hue of a region can be represented by the direction of the hatching lines (Fig. 2b). However, it is inappropriate for discriminating similar hues, such as a single-hand clock.

By analogy to a clock, attaching short line segments to hatching lines greatly improves the discriminability between similarly colored regions (Fig. 2c). By modifying

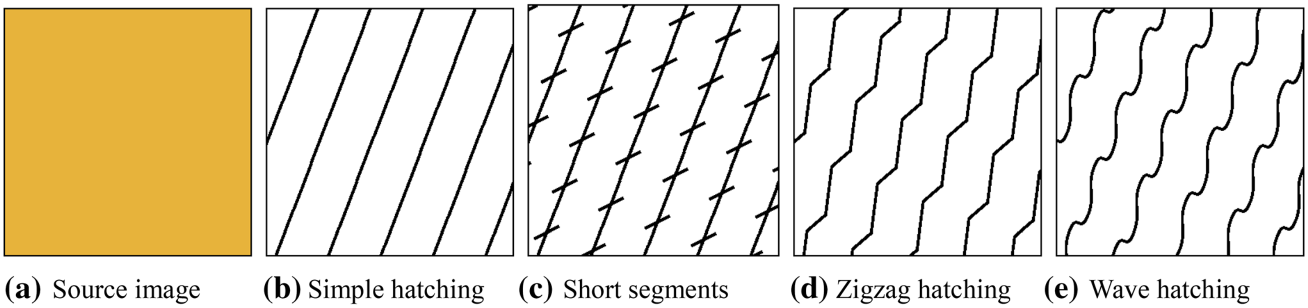


Fig. 2 Comparison of our hatching patterns

the geometry, zigzag hatching (Fig. 2d) and wave hatching (Fig. 2e) are also available for disambiguation. In the next section, we discuss how to generate the aforementioned hatching patterns.

4 Methods

The color of each region is first converted to a triplet of hue, saturation, and lightness. By default, we used CIELAB, and the hue (angle) and saturation/chroma were obtained using the cylindrical model. Other color models, such as HSV and HSL, can also be applied.

Each region (interior of the closed path) was tiled with a tilted hatching pattern. The hatching pattern and its tilt (rotation) angle were obtained from the hue of the region, as described below.

4.1 Simple hatching pattern

In this study, a hatching pattern is a rectangle whose top-left corner is the origin of the coordinate system, and the bottom-right corner is $\left(W_{\text{hatch}}, H_{\text{hatch}}\right)$. In our simple hatching pattern, a straight line was drawn vertically in the center to prevent overlapping with the border of the pattern (Fig. 3). W_{hatch} is determined by lightness if hatching reflects the lightness of the region color, as discussed in Subsection 4.6. H_{hatch} is typically equal to W_{hatch}.

4.2 Pattern angle

The above hatching pattern was rotated and laid over the region to represent the hue of the source color of the region. The simple hatching pattern rotated by 180° is the same as the original pattern, and then, our pattern rotation angle θ_{pattern} is given by

$$\theta_{\text{pattern}} = \frac{\theta_{\text{Hue}}}{2}$$

where θ_{Hue} is the hue angle of the source color $(0 \leq \theta_{\text{Hue}} < 360)$. Figure 1 shows a pair of sample color pie charts and their simple hatching images.

4.3 Segment attached hatching

As discussed in Sect. 2, simple hatching is weak in discriminating between regions of similar colors in the source image. By analogy to a clock, we attach short line segments to a simple hatching pattern (Fig. 4). The pattern angle, corresponding to the direction of (parallel) lines, roughly represents the hue, and the angle between the line and the attached segment, θ_{short}, shows subtle differences between hues. In other words, θ_{pattern} and θ_{short} correspond to the hour and minute hands in a clock, respectively.

The short segment angle θ_{short} is given by

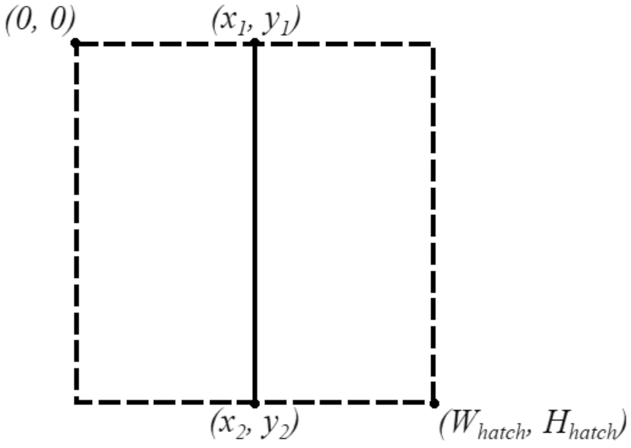


Fig. 3 Geometry of a simple hatching pattern

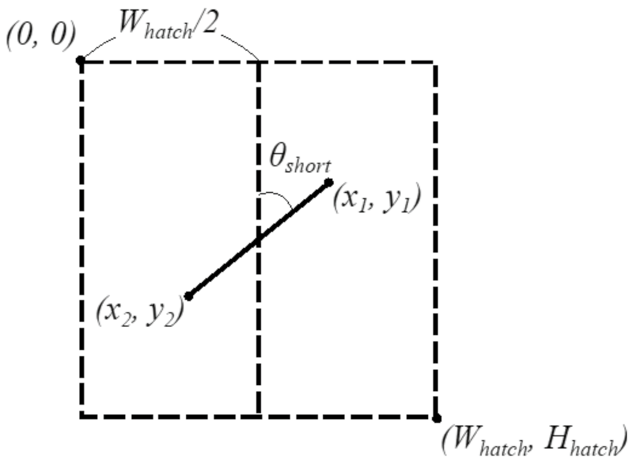


Fig. 4 Geometry of a short segment hatching pattern

$$\theta_{\text{short}} = 180\left(\frac{\theta_{\text{pattern}}}{\theta_{\text{OE}}} - \left\lfloor \frac{\theta_{\text{pattern}}}{\theta_{\text{OE}}} \right\rfloor\right)$$

θ_{OE} is a period of θ_{short}, which is determined so that the directions of the main hatching lines can be discriminated between θ and $\theta + \theta_{\text{OE}}$, and $\lfloor x \rfloor$ is a floor of x. In a clock, θ_{OE} corresponds to 30° (angle for an hour) of the hour hand. Figure 2c shows the segment attached to the hatching of the colored region in Fig. 2a.

4.4 Zigzag hatching

Modifying the geometry in the above hatching pattern yielded some variational hatching patterns. In zigzag hatching (Fig. 2d), zigzag lines replace straight lines in simple hatching to clarify subtle differences between similar colors.

Let l_{long} and l_{short} be lengths of segments where $l_{\text{short}} < l_{\text{long}}$, and the short segment angle θ_{short} is the same as in the previous subsection. In our experiment, $l_{\text{short}} = \frac{l_{\text{long}}}{2}$. The zigzag angle θ_{zigzag} is then given by

$$\theta_{\text{zigzag}} = \begin{cases} 180 - \theta_{\text{short}} & (0 \leq \theta_{\text{short}} < 90) \\ \theta_{\text{short}} & (90 \leq \theta_{\text{short}} < 180) \end{cases}$$

Two segments of lengths l_{long} and l_{short} form an included angle θ_{zigzag}, as shown in Fig. 5. From the law of cosines, the height of the Zigzag hatching pattern H_{zigzag} is given by

$$H_{\text{zigzag}} = \sqrt{l_{\text{long}}^2 + l_{\text{short}}^2 - 2 l_{\text{long}} l_{\text{short}} \cos \theta_{\text{zigzag}}}$$

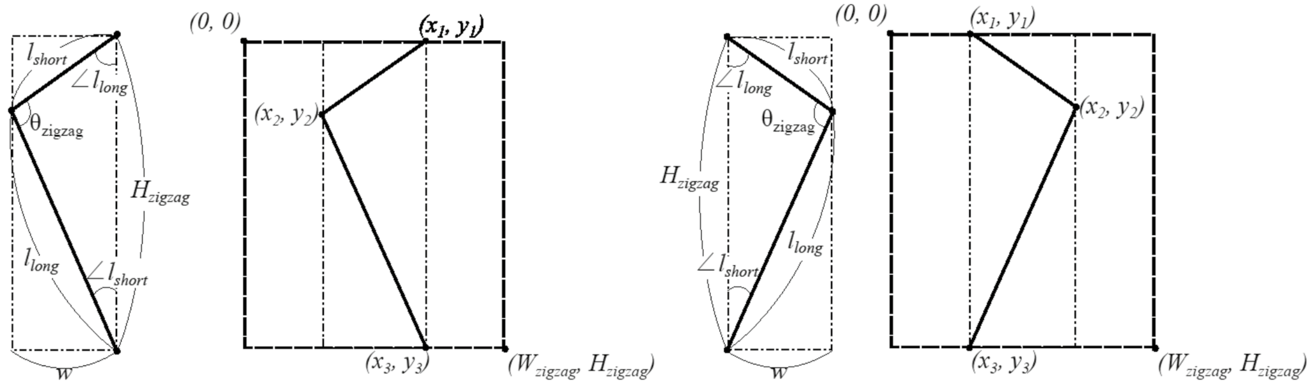
The width of the pattern W_{zigzag} is determined by lightness if the hatching is reflected by the lightness of the region color, as discussed in Subsection 4.6.

Note that the intersection point of the two segments is placed on the left side if $0 \leq \theta_{\text{short}} < 90$ (Fig. 5a), and on the right if $90 \leq \theta_{\text{short}} < 180$ (Fig. 5b).

4.5 Wave hatching

Smoothly connecting the endpoints of the two segments in zigzag hatching yields wave hatching (Fig. 2e). Suppose that the intersection point (and also the endpoints) of the two segments in zigzag hatching is (x_2, y_2), and the other endpoints of the short and long segments are (x_1, y_1) and (x_3, y_3), respectively (see Fig. 5). Then, we draw two cubic Bézier curves. One has four control points: (x_1, y_1), (x_1, y_2), (x_2, y_1), and (x_2, y_2), and the control points of the other are (x_2, y_2), $(x_2, 2y_2)$, $(x_3, y_3 - y_2)$, and (x_3, y_3) (Fig. 6).

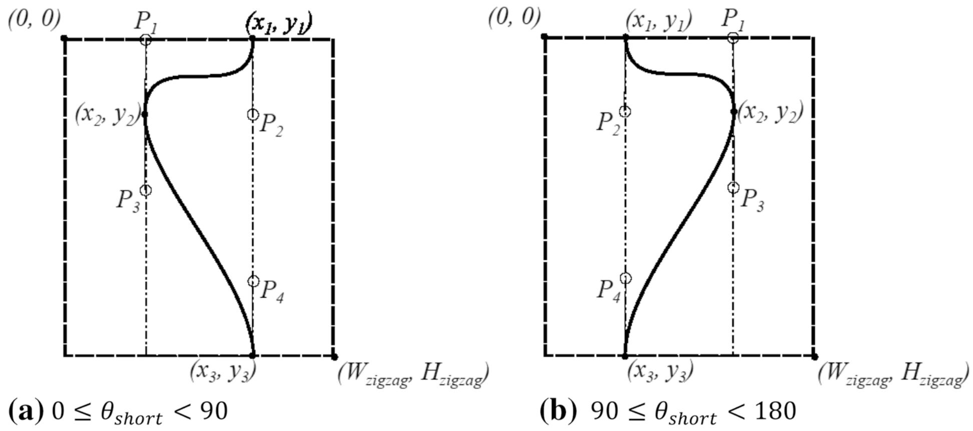
Similar to zigzag hatching, the curves are drawn on the left side if $0 \leq \theta_{\text{short}} < 90$ (Fig. 6a), and on the right if $90 \leq \theta_{\text{short}} < 180$ (Fig. 6b).



(a) $0 \leq \theta_{short} < 90$

(b) $90 \leq \theta_{short} < 180$

Fig. 5 Geometry of a zigzag hatching pattern



(a) $0 \le \theta_{short} < 90$ **(b)** $90 \le \theta_{short} < 180$

Fig. 6 Geometry of a wave hatching pattern

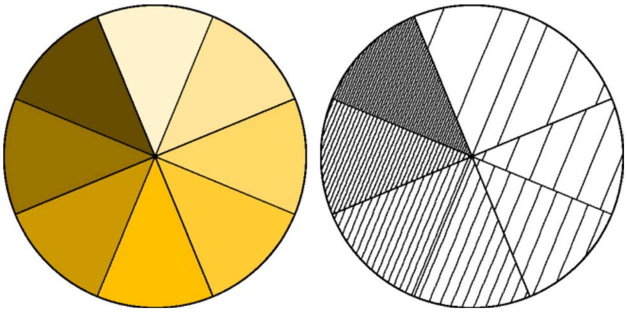


Fig. 7 Lightness and line intervals

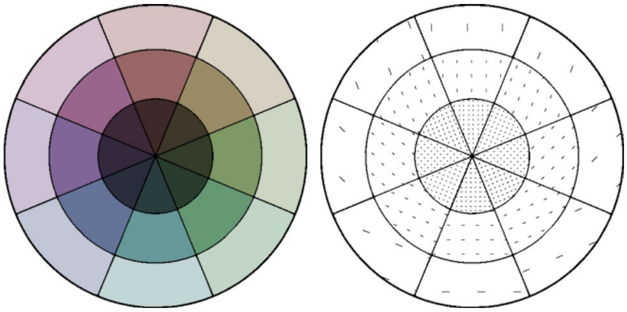


Fig. 8 Saturation and dashed lines

4.6 Mapping lightness to intervals between lines

The representation of hue has been discussed thus far in this section, and this subsection focuses on lightness. From our hatching mechanism, where rectangular hatching patterns are laid over regions, lightness is controlled by the interval between hatching lines, and the interval is equal to the width of the hatching pattern W_{hatch}.

Any positive monotonically increasing function can be mapped from the lightness value of a region to its W_{hatch}. We then use

$$W_{\mathrm{hatch}} = W_{\mathrm{min}} \left(1 + L^{\prime}\right)^{p}$$

where W_{min} is the minimum interval between hatching lines such that the hatching direction is identifiable, L' is the lightness normalized to $0 \le L^{\prime} \le 1$, and p is an arbitrary parameter. Figure 7 compares a sample pie chart composed of slices of similar hue and different lightness values and its simple hatching result.

4.7 Mapping saturation to dashed lines

If required, the proposed method represents saturation via dashed lines. From the saturation S', normalized to $0 \le S^{\prime} \le 1$, the lengths of the stroke and the break are

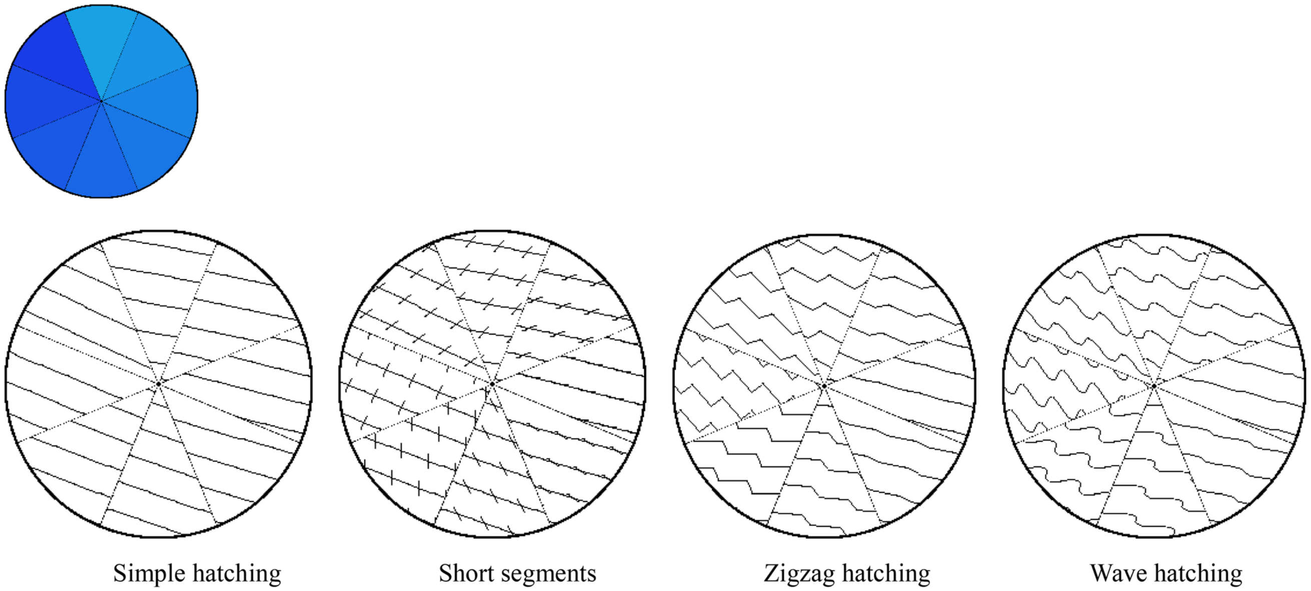
$$l_{\mathrm{stroke}} = S^{\prime} L_{\mathrm{base}}$$

$$l_{\mathrm{break}} = \left(1 - S^{\prime}\right) L_{\mathrm{base}}$$

where the base length L_{base} is usually used as W_{hatch}. Figure 8 compares a sample pie chart composed of slices of different saturations and its simple hatching result.

5 Results

We implemented a color-to-hatching conversion system. Figure 2 illustrates the hatching patterns with a source color. Figure 9 shows how our hatching clarifies the differences


Simple hatching

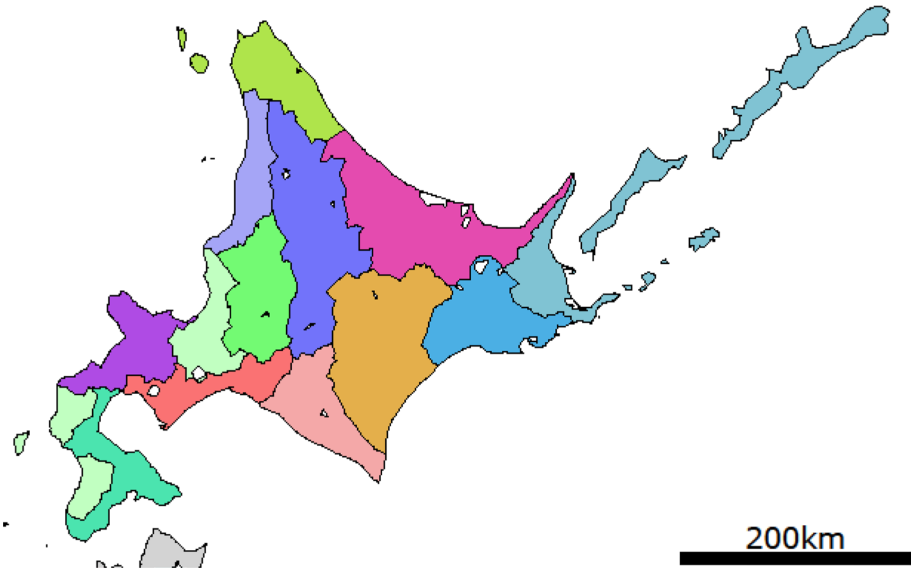
Short segments

Zigzag hatching

Wave hatching

Fig. 9 Comparison of patterns for similar colors (top-left: original color image; bottom row: hatching samples)

Fig. 10 A color image of subprefectures in Hokkaido Island (Subprefectures of Hokkaido.svg created by Lincun at Japanese Wikipedia, under CC BY-SA 3.0 license (http://creativecommons.org/licenses/bysa/3.0/))



between regions of similar colors in the original color pie chart. Figures 11 and 12 compare our hatching and Color2Gray [2] results converted from a map of subprefectures on Hokkaido Island in Japan (Fig. 10).

To mimic a low-cost B/W printing process, the images in Figs. 11 and 12 were printed on a laser printer (Canon LBP7200CN), photocopied by an electrostatic copier (Xerox ApeosPort-IV 4471) and then scanned (Canon CanoScan LiDE 400) (Fig. 13). In Fig. 14, a stencil printer (Riso Kagaku Risograph RZ770) replaces the copier. The features of hatching survive, and the regions are still discriminable even in low-cost B/W printing.

From these results, Color2Hatch can clarify subtle differences in the color of regions compared with the existing methods, as discussed in Sect. 2. In addition, the results of the proposed method do not appear dark; in other words, they consume much less ink/toner in printing. This reduces the printing cost, as discussed in the next section.

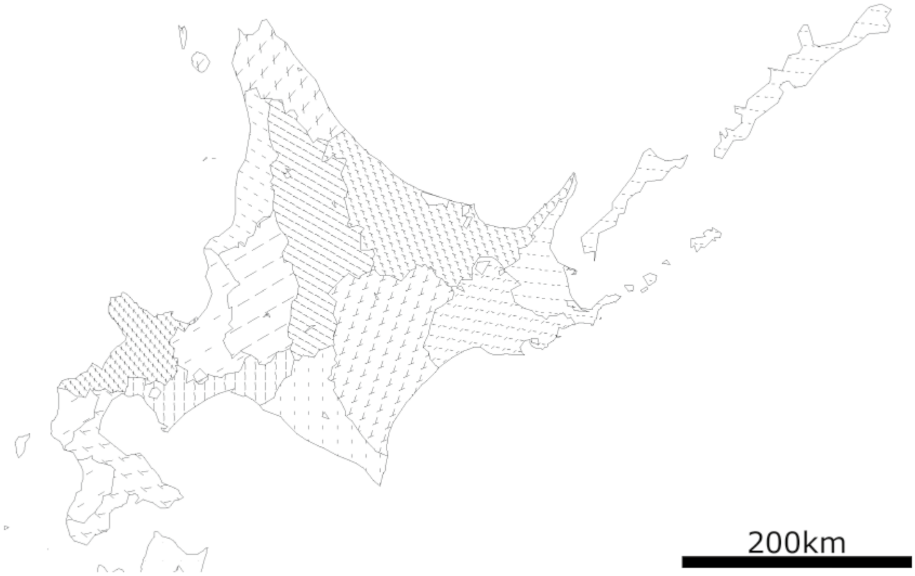


Fig. 11 A hatched image converted by our method (segment attached hatching + dashed lines)

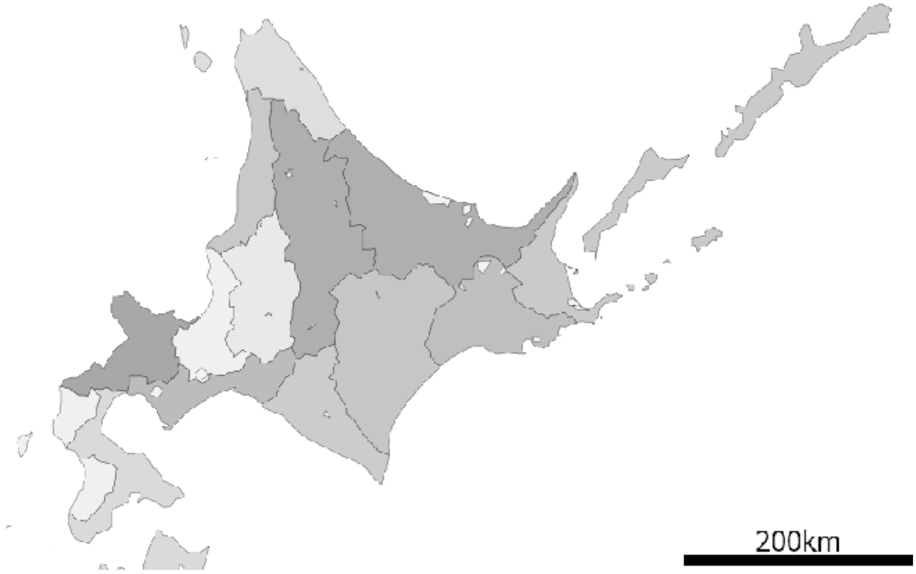


Fig. 12 A grayscale image converted by Color2Gray [2]

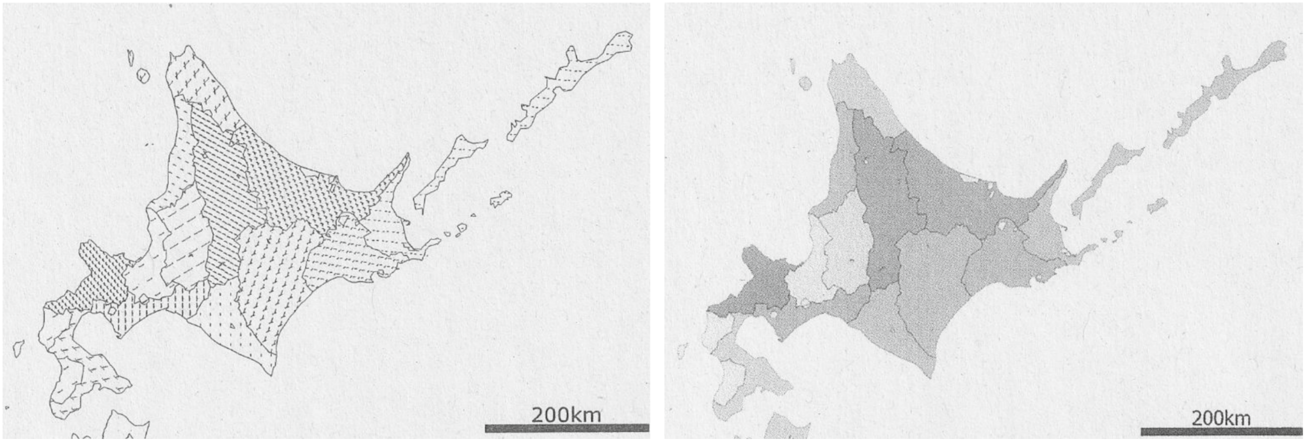


Fig. 13 Images photocopied by an electrostatic copier (left: ours (segment attached hatching + dashed lines); right: Color2Gray [2])

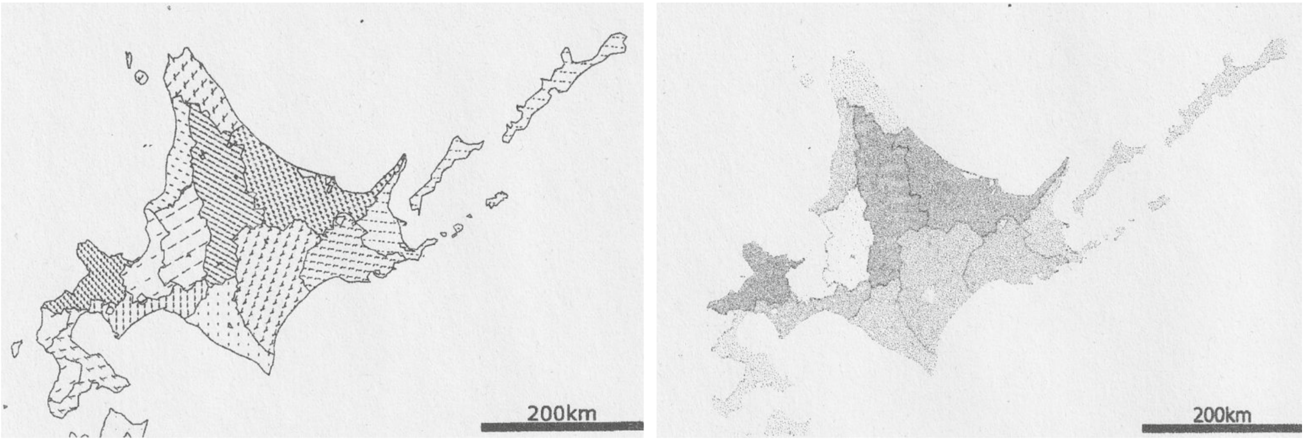


Fig. 14 Images printed by a stencil printer (left: ours (segment attached hatching + dashed lines); right: Color2Gray [2])

6 Discussions

6.1 Printing cost savings

In this section, we compare the printing costs of our method with those of existing printing methods. Both mass printing and small quantity printing with PC printers are discussed below.

6.1.1 Mass printing

A printing price list for estimation is the price of mass printing, by a printing company. The printing price[1] is proportional to the number of color inks in both sheet-fed printing and rotary printing, and color materials are usually printed with four types of ink (C(cyan), M(magenta), Y(yellow), and K(black)) [26]. If color images are translated to our hatched representation, the printing cost is reduced by one-fourth.

6.1.2 Small quantity printing

We often use PC printers such as laser or inkjet printers for small quantity printing, and most of the printing cost of these printers is that of toners and inks.

Laser printers. A laser printer prints an image by transferring and fixing the toner (powdered ink) to a paper, and four types of toners (CMYK) are used in color laser printers. Although accurately predicting toner usage is an ongoing research problem [27], we make a simple estimation of toner usage based on its area coverage. The color shades are represented by the quantity of the toner fixed on a paper, and the relation between the area coverage of the toner and lightness is represented by a tone reproduction curve. In our estimation, from the real tone reproduction curve of a color laser printer [28], we model a simple tone reproduction curve, as shown below (Fig. 13):

$$L^* = \begin{cases} 100 - c & \text{(black)} \\ 100 - 0.5c & \text{(cyan, magenta, yellow)} \end{cases}$$

where c is the area coverage of the toner ($0 \leq c \leq 100$), and L^* is the lightness ($0 \leq L^* \leq 100$).

To map an RGB color to the quantity of the four types of toner, we first convert RGB to CMYK. Although the current color printing devices/drivers usually use 3D LUT for RGB–CMYK conversion, our estimation uses a simple conversion method, which was used in early color printers [29]. The conversion method from (R, G, B) is ($0 \leq R, G, B \leq 255$):

Step 1. C_1, M_1, and Y_1 are obtained as.
$C_1 = 255 - R, M_1 = 255 - G, Y_1 = 255 - B.$
Step 2. K is obtained as.
$K = \mathrm{Min}\left(C_1, M_1, Y_1\right).$
Step 3. C_2, M_2, and Y_2 are obtained as.
$C_2 = C_1 - K, M_2 = M_1 - K, Y_2 = Y_1 - K.$

The resultant $\left(C_2, M_2, Y_2, K\right)$ is then used to obtain the toner quantities matching the toner reproduction curve.

Table 1 compares the quantity of the toner in color printing, grayscale printing, and our hatching for a few sample colors.[2] We suppose that our hatching uses segment-attached hatching, where the length of the short segments is half of W_{hatch} (and H_{hatch}), the width of the long/short segments is one-tenth that of W_{hatch}, the long/short segments are in pure black, and lightness and saturation are not considered in hatching.

[1] The printing cost in a narrow sense. DTP, stencil production, papers, and bookbinding are not considered.

[2] The values of toners of color printing are adjusted so that the lightness values match.

Table 1 Comparison of estimated toner/ink consumptions

Colors				Toners for color printing					Black toner for grayscale printing	Black toner for our hatching
Name	R	G	B	C	M	Y	K	Total		
red	255	0	0	0.0	0.70	0.70	0.0	1.4	0.70	0.15
green	0	255	0	0.41	0.0	0.41	0.0	0.82	0.41	
blue	0	0	255	0.89	0.89	0.0	0.0	1.78	0.89	
yellow	255	255	0	0.0	0.0	0.23	0.0	0.23	0.11	
magenta	255	0	255	0.0	1.0	0.0	0.09	1.09	0.59	
cyan	0	255	255	0.60	0.0	0.0	0.0	0.60	0.30	
pink	255	127	127	0.0	0.35	0.35	0.0	0.70	0.35	

The value 1 of toner consumption means that a single type of toner fully covers an area

As presented in Table 1, except for light colors (e.g., yellow), our hatching method consumes much less toner than color printing and grayscale printing, even if the price difference between black (K) toner and color (cyan, magenta and yellow (CMY)) toners is not taken into consideration.

Inkjet printers. An inkjet printer prints an image by propelling droplets of ink onto paper, and four types of ink (cyan, magenta, yellow and black (CMYK)) are used in typical color inkjet printers.[3] Similar to laser printers, the shades of colors are represented by the quantity of ink on a paper, and it is controlled by halftoning, for example, Bayer dithering or error-diffusion methods. Assuming the ink has a tone reproduction curve similar to that shown in Fig. 15, the estimation in Table 1 is valid for inkjet printers, and our hatching method has a cost advantage.

Note that the above estimations of toner/ink consumption are based on simple assumptions and ignore many factors in the real world, for example the order of toners/inks, the overlapping of dots, dot gain, etc.

6.2 Limitations

In our current method, there is still much room for improvement. As shown in Figs. 10, 11, 12, to identify the color (or its corresponding hatching) of a region, each region should be considerably larger in our method.

One of the problems lies in the design of the shape of zigzag hatching and wave hatching. In the current design, a discontinuity exists when $\theta_{\text{short}} = 90$. This may result in two different shapes from the same color because of, for example, round-off errors.

Another problem is the lack of representation of achromatic colors. A simple solution is to convert achromatic colors (colors whose saturation is below a threshold) to another type of hatching, for example cross-hatching.

6.3 Applications

Many printers have a B/W button, and/or their driver windows have a toggle switch between color printing and B/W printing. They enable us to print color documents instantly in grayscale. Their printer output, however, is in conventional grayscale and has the problem discussed in Sect. 2.

If a printer receives page description language commands and also has our Color2Hatch functions in its firmware, pages with uniformly colored regions can be properly hatched in B/W printing. For example, in a page described by Postscript [31], regions surrounded by a closed path are hatched instead of being filled with a color. This is an extended/enforced version of the existing B/W button and achieves better discriminability and toner saving.

Our Color2Hatch can be applied to electronic and conventional papers. Compared to liquid crystal displays, electronic papers are superior in terms of high readability and low power consumption. To achieve high readability, however, many electronic paper devices are monochrome, and their users have difficulty reading color materials. If an electronic paper device has Color2Hatch functions, it can

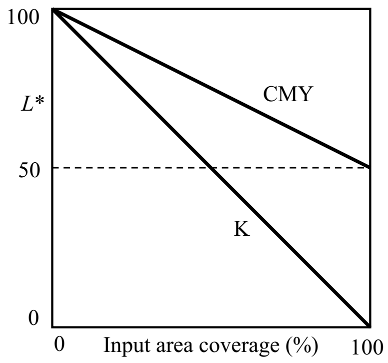


Fig. 15 A model tone reproduction curve

[3] Many inkjet printers for photo-printing have six or more types of ink because the use of dark and light colors contributes to the achievement of both high tone reproduction and low graininess [30]. Its color model is often refereed as CcMmYK color model. In this paper, we make estimations only for printers of four types of ink.

convert color materials to hatched representations in B/W. This will improve the availability and usefulness of electronic paper devices.

6.4 Relation to visualization

Our method is related to glyph-based visualization [32]. As shown in Fig. 11, the segment attached hatching integrated with dashed lines appears to be an iconographic display of multidimensional data [33]. This is reasonable considering that color is a kind of multidimensional data. Based on hatching, however, our alignment of icons/glyphs is regular and appropriate for representing the uniform color of each region.

7 Conclusions

We proposed a novel decolorization method for color business graphics in vector format. Unlike the existing grayscale conversion and texturing methods, our Color2Hatch method is visually clear, provides greater distinguishability, and meets real-world needs even for low-cost printing. Our approach to robust conversion from color to hatching will be useful for saving the printing costs associated with preparation and setup.

One interesting problem is to match the difference (distance) of the two colors and their corresponding hatching from a cognitive perspective. Future work will be applied to non-uniformly colored regions.

We hope that document processors and/or printing systems in the future can instantly prepare B/W printing data from color documents.

Declarations

Conflict of interest The authors have no conflicts of interest to declare relevant to the contents of this article.

References

1. Bala, R., Braun, K.M.: Color-to-grayscale conversion to maintain discriminability. In: Proceedings of the SPIE 5293, Color Imaging IX: Processing, Hardcopy, and Applications (2003). Doi:https://doi.org/10.1117/12.532192
2. Gooch, A.A., Olsen, S.C., Tumblin, J., Gooch, B.: Color2Gray: salience-preserving color removal. ACM Trans. Graph. **24**(3), 634–639 (2005)
3. Rasche, K., Geist, R., Westall, J.: Re-coloring images for gamuts of lower dimension. Comput. Graph. Forum. **24**(3), 423–432 (2005)
4. Grundland, J., Dodgson, N.A.: Decolorize: fast, contrast enhancing, color to grayscale conversion. Pattern Recognit. **40**(11), 2891–2896 (2007)
5. Smith, K., Landes, P.-E., Thollot, J., Myszkowski, K.: Apparent greyscale: a simple and fast conversion to perceptually accurate images and video. Comput. Graph. Forum. **27**(2), 192–200 (2008)
6. Kuhn, G.R., Oliveira, M.M., Fernandes, L.A.: An improved contrast enhancing approach for color-to-grayscale mappings. Vis. Comput. **24**(7), 505–514 (2008)
7. Wu, J., Shen, X., Liu, L.: Interactive two-scale color-to-gray. Vis. Comput. **28**(6), 723–731 (2012)
8. Ji, Z., Fang, M.E., Wang, Y., Ma, W.: Efficient decolorization preserving dominant distinctions. Vis. Comput. **32**(12), 1621–1631 (2016)
9. Cui, M., Hu, J., Razdan, A., Wonka, P.: Color-to-gray conversion using ISOMAP. Vis. Comput. **26**(11), 1349–1360 (2010)
10. Zhu, W., Hu, R., Liu, L.: Grey conversion via perceived-contrast. Vis. Comput. **30**(3), 299–309 (2014)
11. Zhao, H., Zhang, H., Jin, X.: Efficient image decolorization with a multimodal contrast-preserving measure. Comput. Graph. **70**, 251–260 (2018)
12. Wang, W., Li, Z., Wu, S.: Color contrast-preserving decolorization. IEEE Trans. Image Process. **27**(11), 5464–5474 (2018)
13. Zhang, X., Liu, S.: Contrast preserving image decolorization combining global features and local semantic features. Vis. Comput. **34**(6), 1099–1108 (2018)
14. Liu, Q., Li, S., Xiong, J., Qin, B.: WpmDecolor: weighted projection maximum solver for contrast-preserving decolorization. Vis. Comput. **35**(2), 205–221 (2019)
15. Roetling, P.G.: Pseudo-gray techniques for the black and white display of color images. In: Proceedings of the SPIE 0310, Image Quality (1981) Doi:https://doi.org/10.1117/12.932860
16. Harrington, S.J.: Mapping of color images to black-and-white textured images. US Patent 5,153,576 (1992)
17. Harrington, S.J.: Algorithmic mapping of colors to textures. Proc. SPIE 2171, Color Hard Copy and Graphic Arts III (1994) Doi:https://doi.org/10.1117/12.175319
18. Bai, Y., Harrington, S.J., Taber, J.: Improved algorithmic mapping of color to texture. In: Proceedings of the SPIE 4300, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI (2000) Doi:https://doi.org/10.1117/12.410819
19. Queiroz, R.L., Braun, K.M.: Color to gray and back: color embedding into textured gray images. IEEE Trans. Image Process. **15**(6), 1464–1470 (2006)
20. Guss, H.: Texturing coloured images in black and white. Proc. Bridges **2016**, 493–496 (2016)
21. Saito, T., Takahashi, T.: Comprehensible rendering of 3-D shapes. In: Proceedings of SIGGRAPH '90, 197–206 (1990)
22. Salisbury, M.P., Wong, M.T., Hughes, J.F., Salesin, D.H: Orientable textures for image-based pen-and-ink illustration. In: Proceedings of the SIGGRAPH '97, 401–406 (1997)

23. Webb, M., Praun, E., Finkelstein, A., Hoppe, H.: Fine tone control in hardware hatching. In: Proceedings of NPAR '02, 53-ff (2002)
24. Institute, A.N.S.: Line conventions and Lettering. Y **14**, 2M (1979)
25. Scalable Vector Graphics (SVG), https://www.w3.org/Graphics/SVG/
26. Economic Research Association (ed): Printing Price Data 2021 (Sekisan Shiryou Insatsu Ryoukin 2021). Economic Research Association (2021) (in Japanese)
27. Wang, L., Abramsohn, D., Ives, T., Shaw, M., Allebach, J.: Estimating toner usage with laser electrophotographic printers. In: Proceedings of the SPIE 8652, Color Imaging XVIII: Displaying, Processing, Hardcopy, and Applications (2013)
28. Hara, T.: Current status and issues of print quality in color laser printers. J. Imag. Soc. Jpn **40**(3), 231–236 (2001)
29. The Imaging Society of Japan (ed): Inkjet (revised version). Tokyo Denki University Press (2018) **(in Japanese)**
30. Makita, T., Kutsuma, T.: Image processing design optimized for high definition inkjet printers considering human visual system. J. Imag. Soc. Jpn **41**(4), 358–367 (2001)
31. Adobe Systems Inc.: PostScript Language Reference (3rd ed.). Addison-Wesley, New York (1999)
32. Borgo, R., Kehrer, J., Chung, D.H., Maguire, E., Laramee, R.S., Hauser, H., Ward, M., Chen, M.: Glyph-based visualization: foundations, design guidelines, techniques and applications. Eurographics 2013, State of the Art Reports (2013) doi:https://doi.org/10.2312/conf/EG2013/stars/039-063
33. Pickett, R.M., Grinstein G.G.: Iconographic displays for visualizing multidimensional data. In: Proceedings of IEEE SMC '88, 514–519 (1988)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ryoma Isumi received his M.Eng degree in computer science and systems engineering from Kyushu Institute of Technology in 2019. His research interests include computer graphics and functional programming.

Kunio Yamamoto is an assistant professor in the Department of Artificial Intelligence at Kyushu Institute of Technology. He received his M.Eng degree in computer science and systems engineering from Kyushu Institute of Technology in 2000. His research interests include computer graphics and human computer interaction.

Tsukasa Noma is currently a professor in the Department of Artificial Intelligence at Kyushu Institute of Technology. His research interests include computer graphics, animation, and human computer interaction. He received his B.Sc. degree in mathematics from Waseda University in 1984, and the M.Sc. and D.Sc. in information science from the University of Tokyo in 1986 and 1989.