MANGAWALL: GENERATING MANGA PAGES FOR REAL-TIME APPLICATIONS

Zhipeng Wu and Kiyoharu Aizawa

The University of Tokyo, Tokyo, Japan

ABSTRACT

Recent advances in non-photorealistic rendering provide the best convenience for automatic image to manga conversion. However, we are facing a dilemma that the fine-grained manga conversion algorithms always involve computation intensive processes (e.g. image over-segmentation, brute-force feature matching, and energy optimization), which make it inapplicable to real-time applications. On the other hand, commercial manga Apps available for smart phones usually rely on simple edge detection and halftoning or hatching. Although the processing can be finished in several seconds, the conversion quality is not satisfactory yet. In this paper, we propose MangaWall to automatically convert and organize photos into manga pages. Our goal is to establish a lightweight manga rendition framework as well as generate high-quality images. To achieve this, the manga structure is enhanced by flow-based DoG operation and image vectorization. A multi-layer tone mapping and contrast-aware halftoning method is then proposed to render the manga-like screentone patterns. Besides, for multi-image input, a full binary tree-based layout representation is employed to efficiently organize manga images onto the same page canvas. Our MangaWall can be applied to real-time applications. It takes less than 0.7 second to convert a 1024×768 image into manga on Laptop PC.

***Index Terms*—** Non-Photorealistic Rendering, MangaWall, Image2Manga

1. INTRODUCTION

Originated in the late 19^{th} century, as a unique form of art, manga has won a worldwide popularity. Since 1950s, manga has steadily become the major part of Japanese publishing industry. It was reported that in Japan, the manga market size reached $5.5 billion in 2009. In Europe and the Middle East, the market was worth $250 million. And in the U.S. and Canada, it was also valued at $175 million. Encouraged by the huge number of fans, the endeavor of manga innovation has never stopped. Electronic manga drawing tools has emerged to help artists draw and organize manga on the computer. Since the introduction of such software greatly saves the workload and resources, computer-assisted manga design points out the future direction of manga.

Recently, with the rapid progress of smart-phone technology, amateur-oriented image to manga Apps quickly won significant reputation. The representative 'Manga Camera' [1] on iOS platform earned a burst of one million downloads in one week which made it the 1st-ranking-application on iOS App Store Japan. However, as shown in Fig.4, although it provides manga-like outputs and the introducing of frame templates further enhances the user experience, the manga conversion quality is not satisfactory. There is a need for more effective manga rendering algorithms as well as layout organization schemes to arrange multiple manga frames onto a single page canvas.

On the other hand, in research area, computational manga plays an emerging role recently. In [2], researchers proposed an effective method for colorizing black-and-white manga pages which contain intensive amount of strokes, hatching, halftoning and screening. Qu et al. proposed a novel approach for generating manga-style screen patterns with input color images [3]. Their goal is to preserve the visual richness in the original photograph by utilizing not only screen density, but also the variety of screen patterns. The combination of tone similarity, texture similarity, and chromaticity distinguishability ensures the high quality of image to manga conversion, especially the rendition of manga screening. In [4], a content-sensitive manga screening method was proposed as an alternative. This method improves the contrast-aware halftoning [5] with better screening pattern generation and edge exclusion. However, restricted to the time-consuming procedures such as image over-segmentation, these methods are infeasible for real-time applications. For instance, Given an 800×1000 input image, the processing time for [3] is about 4 minutes (Intel P4@3.2Ghz) and for [4] is 110 seconds (Intel Core Duo@3.0GHz).

While taking both of the efficiency and effectiveness into consideration, we propose a lightweight MangaWall system. It is superior to commercial Apps with a refined manga rendition framework. The core Image2Manga module is a combination of structure and texture processing. The structure extraction step aims to provide a visual abstraction of the image content (flow-based DoG) as well as filter out noise and trivial details (vectorization). In order to reproduce the bitonal texture pattern, we propose a multi-layer tone mapping and halftoning strategy. The manga texture is further obtained by merging multi-layer results with respect to manga images' characteristics. Note that all of the operations are performed on the whole image instead of local segmented regions, comparing with [3, 4], our approach is a 'global method' which strikes a good balance between manga conversion quality and computation cost. More comparisons and results can be found in section 4.

2. RELATION TO PRIOR WORK

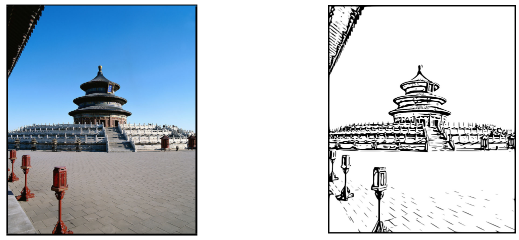
Non-Photorealistic Rendering Computer graphics algorithms that imitate non-photographic techniques such as painting or cartoon [6] were referred to as NPR [7]. To free manga artists from the labor-intensive and time-consuming screening process, in [3], an automatic yet controllable approach for screening stylish manga backgrounds from photographs was introduced. Recently, Lu et al. proposed an effective method for image to sketch conversion by dividing pencil sketch drawing into two sub-problems: stroke generation and tone drawing [8]. Motivated by Lu's work, our MangaWall algorithm first extracts line drawings to express general structure of the scene and then reproduces the manga shadow and shading effect by texture rendering.

Image Halftoning Halftoning refers to converting a continuous tone image into a pattern of black and white dots [9], which plays an indispensable role in natural image to manga conversion. Recently, with an impressive structure preservation capability, structure-aware halftoning (SAH) has come into notice [10]. In [11], Chang et al. proposed structure-aware error diffusion, a fast solution to solve the SAH problem. And in [5], Li et al. further presented a more efficient and effective structure-preserving halftoning method by enhancing the local contrast.

Manga Layout Generation Considering the problem to arrange several manga panels onto the page canvas, previous work introduced domain knowledge into the process of manga layout generation [12]. Based on a large manually labeled manga layout dataset, style models were firstly learned and further integrated under a generative probabilistic framework. This approach requires extensive user operations. The user needs to decide the region of interests, the semantically similar frames, and the number of the panels. On the other hand, for general photo layout generation, tree-based approaches advance others by providing a compact and non-overlapping layout organization [13]. In our previous work, we first generated full binary tree and then mapped it into the layout. To generate size-adjustable manga canvas, the tree generation was improved by using 'divide-and-conquer' paradigm as well as fast tree adjust algorithm [14].

3. IMAGE2MANGA

Structure Extraction: In general, manga production starts with drafting outlines, and further deepens the reader-impression



(a) Input image (b) Extracted structure

Fig. 1. Structure extraction.

by summarization, generalization, and exaggeration of daily scenes. Likewise, our structure extraction scheme should not only keep the major components of the original image but also eliminate the noises and trivial details. Comparing with the classical edge detection problem, our task requires two properties: (1) Filtering non-obvious lines to simplify the visual cues. (2) Enhancing major contours to convey scene aspects. Line drawing has drawn a lot of attention in recent NPR research. As proposed in [15], the main idea of line drawing was to take into account the 'flow' of edges. Rather than conducting classical 2-D DoG operations to extract edge responses, it was shown that we could receive the biggest contrast while applying 1-D DoG operator along the perpendicular direction to local 'edge flow' (Flow-based DoG, FDoG). Then, after flow-based DoG filtering, the edge line drawing is further vectorized to remove noise and trivial details. Fig. 1 shows an example of manga structure extraction.

Texture Rendering: Comparing with other art forms, manga has two common characteristics:

Screentone Pattern: After drawing structure lines, manga artists usually select appropriate pre-print screen sheets to fill regions in order to express shading, tone, texture, or atmosphere. This process is called screening which results in the screentone pattern in manga productions. In digital processing, the conversion from gray-scale image into binary image, namely halftoning, reproduces the screentone texture pattern. To best preserve the structure and tone similarity, we introduce the idea of contrast-aware halftoning [5] and implement it as algorithm 1.

As shown in Fig.2, to distribute the binary quantization error to nearby 7×7 neighbors, the basic idea is to let the darker pixels remain dark and brighter pixels remain bright. To be more specific, in the first step, we determine which color (black or white) should be chosen for the current $P_{x,y}$. Then, the quantization error is calculated by subtracting the chosen intensity from the original intensity. This error can be either negative or non-negative. For error distribution, we only consider $P_{x,y}$'s unvisited neighbors (Fig.2, pixels with yellow borders), as shown in algorithm 1.

White/Black Regions: Another key observation of manga images is the prevalent existence of solid white/black regions. Since manga is a highly abstract art form, these 'Flat Regions' are used to represent not only full white/black but also

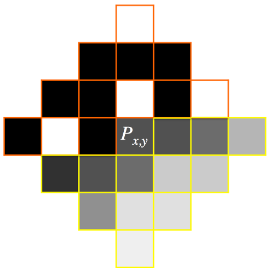


Fig. 2. Contrast-aware halftoning.

bright/dark areas. Noticing the fact that the tone distribution of manga images generally differs significantly from daily photos, we introduce a set of tone mapping functions [8]:

$$Bright: p_b = \frac{1}{z_b} exp(\frac{i-255}{9}) \tag{1}$$

$$Middle: p_m = \begin{cases} \frac{1}{z_m}(u_a - u_b) & \text{if} \quad u_a \le i \le u_b \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$Dark: p_d = \frac{1}{z_d} exp(\frac{-i}{9}) \tag{3}$$

Equation 1-3 refer to the parametric models for image's bright, middle, and dark parts, and they are further combined by taking weighted average. In practice, we set empirical parameters $\{w_b$:2, w_m:1, w_d:1$\}$ to extract the image's bright layer, $\{w_b$:1, w_m:2, w_d:1$\}$ for middle layer, and $\{w_b$:1, w_m:1, w_d:2$\}$ for dark layer (Fig.3 $b \sim d$). Then, we conduct contrast-aware halftoning for these layers (Fig.3 $f \sim h$) and merge them to get the final manga texture (Fig.3 k). Algorithm 2 shows our proposed layer fusion strategy. The basic idea is to find the 'prior stable pixels' which has a stable response in all of the three layers. For instance, if one pixel p from dark layer is white, then, it is stable and is believed to have a high probability to remain white in the following middle and bright layer. Similarly, the black pixels in bright layer are also regarded as 'stable'. In the layer fusion map (Fig.3 i), the green pixels refer to the stable pixels extracted from dark layer; the blue ones come from the bright layer, and the red ones are from the middle layer. Comparing with the bitonal image by directly halftoning the input image (Fig.3 e), our result can express a better sense of manga feeling with large white/black regions.

Algorithm 1: Contrast-aware Halftoning

foreach *y from top to bottom* **do**
  foreach *x from left to right* **do**
    $\hat{I_{x,y}}$ = BlackOrWhite($I_{x,y}$) $error = I_{x,y} - \hat{I_{x,y}}$
    foreach *unvisited neighbor pixel q* **do**
      if $error > 0$ **then** $weight_q = \frac{I_q}{dis(q)^k}$
      else $weight_q = \frac{255-I_q}{dis(q)^k}$
    end
    NormalizeAllWeights()
    foreach *unvisited neighbor pixel q* **do**
      $\hat{I_q} \Leftarrow I_q + error \times weight_p$
    end
  end
end

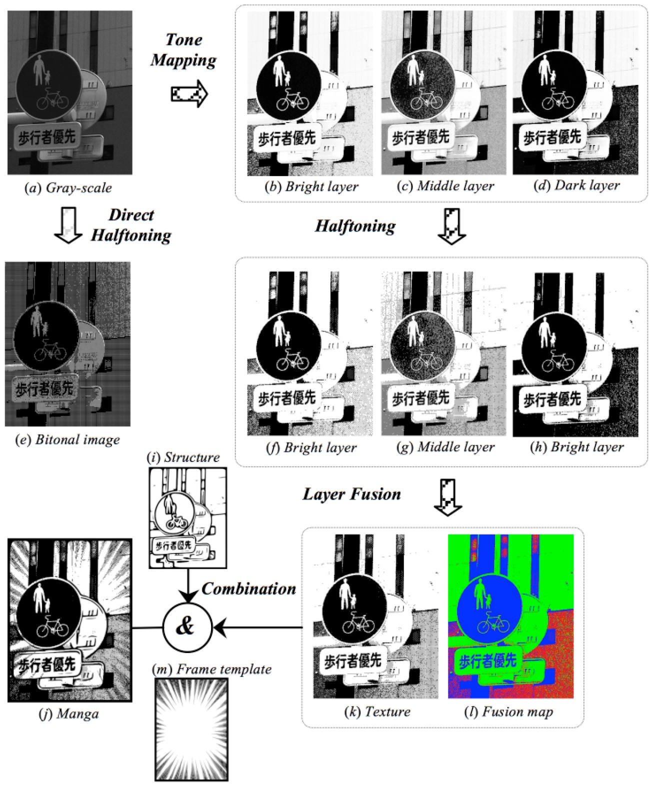


Fig. 3. Texture Rendering.

Algorithm 2: Multi-layer Fusion

foreach *row r and column c* **do**
  if $l_d(r, c)$ *is white* **then**
    $texture(r, c) \Leftarrow white$
  else if $l_b(r, c)$ *is black* **then**
    $texture(r, c) \Leftarrow black$
  else $texture(r, c) \Leftarrow l_m(r, c)$
end

Post-Processing: Since both of the structure (**S**) and texture (**T**) only contain white/black pixels, the final manga (M) can be efficiently obtained by using pixel-wise **&** operation (Fig.3 j). Besides, some pre-defined manga frame templates can also be added. For manga page layout, we employ our previous binary tree-based algorithm [14]. Given m-input images, we first construct a tree which occupies m leaves. Then we do tree adjustments to let the layout fit in the user's input canvas size. Please refer to [14] for more details.

Fig. 4. Visualized examples. (More examples: http://manga-on-the-fly.site11.com)

Table 1. Average processing time for image to manga.

Image Size	320×240	480×320	640×480	800×480
Proc. Time	$64ms$	$128ms$	$261ms$	$328ms$
Image Size	1024×768	1280×720	1440×900	1920×1080
Proc. Time	$679ms$	$798ms$	$1122ms$	$1826ms$

4. RESULTS

Efficiency: Table 1 shows the average processing time for 100 runs on our laptop PC equipped with 2.3GHz Intel Core i7 CPU. Noticing that it only takes less than 0.8 second to convert an HD image into manga, we believe our algorithm can be applied to real-time applications.

Effectiveness: Fig.4 a shows the comparison between MangaWall and [3, 4]. In [3], each of the segmented image regions is optimized to find the best matching texture pattern extracted from real manga database. Although this strategy may produce professional manga screening effect, the result image lacks a sense of abstraction and seems to be 'over-textured'. Instead of using real manga screening patterns, Li's approach [4] provides better flexibility by using automatically generated texture patterns. However, since both of [3] and [4] involves time-consuming procedures such as over-segmentation, these methods are not applicable to time-sensitive tasks. As discussed above, the proposed MangaWall is a fast approach based on 'global processing'. And the large number of white/black region provides better sense of manga-like content abstraction. Our work can be easily extended to create colored manga images. An example is illustrated here as well. Fig.4 b and c compare MangaWall with popular commercial softwares (Manga Camera [1], Comic Book Camera [16], Manga Me [17], and Comic Camera [18]). Our results provide clear and tidy structure lines as well as fine-grained texture patterns. Fig.4 d further shows the result of layout generation. The input sequence is guaranteed to be preserved in the generated manga page, following a natural reading order from top to bottom and from left to right (for Japanese manga, it is right to left). The page canvas is user-adjustable. The size parameters '$width$' and '$height$' are set by the user.

5. REFERENCES

[1] "Manga camera," `http://tokyo.supersoftware.co.jp/mangacamera/index_en.html`.

[2] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng, "Manga colorization," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1214–1220, July 2006.

[3] Yingge Qu, Wai-Man Pang, Tien-Tsin Wong, and Pheng-Ann Heng, "Richness-preserving manga screening," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 155:1–155:8, December 2008.

[4] Hua Li and David Mould, "Content-sensitive screening in black and white," in *GRAPP*, 2011.

[5] Hua Li and David Mould, "Contrast-aware halftoning," *Computer Graphics Forum*, vol. 29, no. 2, pp. 273–280, 2010.

[6] Holger Winnemöller, Sven C Olsen, and Bruce Gooch, "Real-time video abstraction," in *ACM Transactions On Graphics*. ACM, 2006, vol. 25, pp. 1221–1226.

[7] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen, "A non-photorealistic lighting model for automatic technical illustration," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 447–452.

[8] Cewu Lu, Li Xu, and Jiaya Jia, "Combining sketch and tone for pencil drawing production," in *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, 2012, NPAR '12, pp. 65–73.

[9] Robert Ulichney, *Digital halftoning*, 1987.

[10] Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, and Pheng-Ann Heng, "Structure-aware halftoning," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 89:1–89:8, 2008.

[11] Jianghao Chang, Benoît Alain, and Victor Ostromoukhov, "Structure-aware error diffusion," in *ACM Transactions on Graphics (TOG)*. ACM, 2009, vol. 28, p. 162.

[12] Y. Cao, A. B. Chan, and R. Lau, "Automatic stylistic manga layout," *ACM Transactions on Graphics*, vol. 31, 2012.

[13] C. Brian Atkins, "Blocked recursive image composition," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, MM '08, pp. 821–824.

[14] Zhipeng Wu and Kiyoharu Aizawa, "Picwall: Photo collage on-the-fly," in *APSIPA Annual Summit and Conference*. APSIPA, 2013.

[15] Henry Kang, Seungyong Lee, and Charles K Chui, "Flow-based image abstraction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 1, pp. 62–76, 2009.

[16] "Comic book camera," `https://itunes.apple.com/us/app/comic-book-camera-free/id593817786?mt=8`.

[17] "Manga me," `https://itunes.apple.com/us/app/manga-me/id397076266?mt=8`.

[18] "Comic camera," `https://itunes.apple.com/us/app/manga-comic-camera-processed/id578506865?mt=8`.