Vis Comput (2012) 28:657–668
DOI 10.1007/s00371-012-0689-9

ORIGINAL ARTICLE

Automatic generation of accentuated pencil drawing with saliency map and LIC

Michitaka Hata · Masahiro Toyoura · Xiaoyang Mao

Published online: 24 April 2012

Abstract An artist usually does not draw all the areas in a picture homogeneously but tries to make the work more expressive by emphasizing what is important while eliminating irrelevant details. Creating expressive painterly images with such accentuation effect remains to be a challenge because of the subjectivity of information selection. This paper presents a novel technique for automatically converting an input image into a pencil drawing with such emphasis and elimination effect. The proposed technique utilizes saliency map, a computational model for visual attention, to predict the focus of attention in the input image. A new level of detail controlling algorithm using multi-resolution pyramid is also developed for locally adapting the rendering parameters, such as the density, orientation and width of pencil strokes, to the degree of attention defined by saliency map. Experimental results show that the images generated with the proposed method present the visual effect similar to that of the real pencil drawing and can successfully direct the viewer's attention toward the focus.

Keywords Pencil drawing · Saliency map · LIC · Focus of attention · Multi-resolution pyramid

1 Introduction

Compared with photographs, paintings and drawings can often convey information more effectively through emphasizing what is relevant while omitting extraneous details. A skilled artist can successfully direct the viewer's attention to the focus by accentuating the focus through changing the rendering parameters in the picture. This paper presents a novel technique for automatically converting an input photograph into a pencil drawing image with such kind of accentuation effect. Pencil drawings are excellent both as the preparatory sketches and as finished renderings. The monochrome tone and the specific mood of "unfinishedness" can often stir up one's curiosity and imagination, and hence they are also widely used as an expressive medium for apparel or architecture designs and scientific or technical illustrations. Being monochrome, the effect of accentuation is actually particularly important in achieving the charm appearance, as well as the expressiveness of pencil drawings [1]. Figure 1(d) shows an example of pencil drawing by an artist. To direct the viewer's attention toward the squirrel, irrelevant details are eliminated on outer area and the rendering parameters, such as the texture of strokes, are also changed accordingly to accentuate the squirrel. Such accentuation effect is not achieved in Fig. 1(b) which is generated with the existing automatic pencil drawing generation method [2]. Figure 1(c) is the result generated with the proposed method. We can see the squirrel is expressively rendered in a similar way as that of Fig. 1(d).

The advance of non-photo-realistic rendering technology in the past decade has made it possible to simulate almost all kinds of traditional artistic media including the pencil drawing. However, automatically creating true expressive painterly images with such accentuation effect remains to be a challenge. The difficulty mainly lies in the subjectivity of information selection. Given the same scene, different artists may create very different works by accentuating different areas or objects, depending on how they perceive or understand the scene as well as what kind of subjects they want to express. A known perspective to analyze

M. Hata · M. Toyoura · X. Mao (✉)
Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi, 4-3-11, Takeda, Kofu, Yamanashi,
400-8511, Japan
e-mail: mao@yamanashi.ac.jp

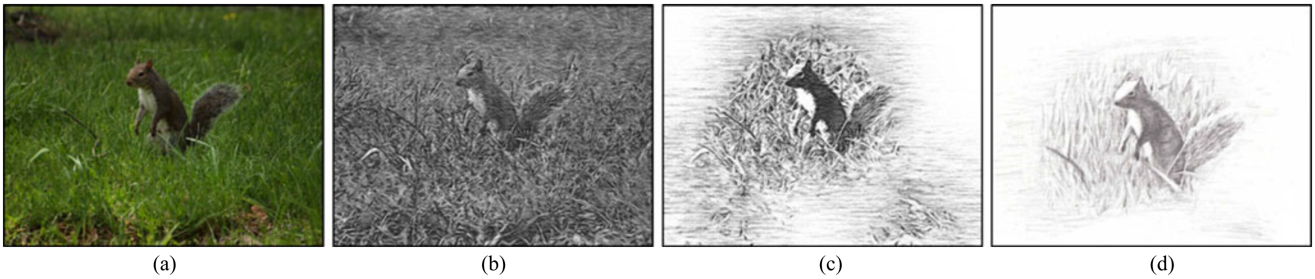


Fig. 1 Accentuated pencil drawing generation using saliency map and LIC (Line Integral Convolution). (**a**) Input image. (**b**) Result of existing method [2]. (**c**) Result of the proposed method. (**d**) Work drawn by an artist

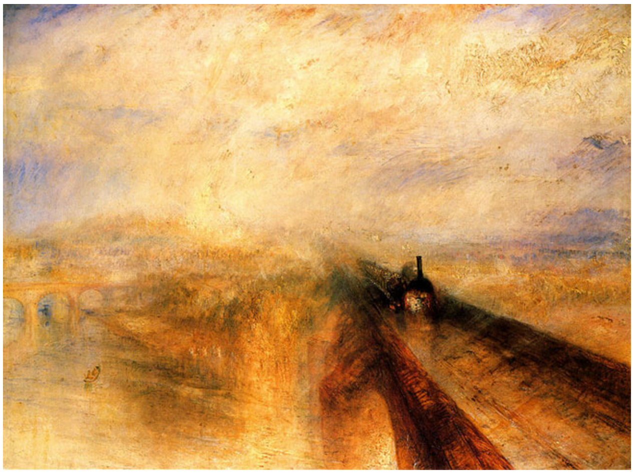


Fig. 2 A viewer-centered painting by J.M.W. Turner "Rain Steam and Speed the Great Western Railway"

the difference among pictures is whether the representation is viewer-centered or object-centered [3]. Object-centered pictures, as typified by Picasso's works, are the interpretation of what artists know, namely the brain image, while the viewer-centered pictures are the interpretation of what artists see, namely, the retinal image. While information selection in an object-centered picture can be purely subjective, a salient feature or location on a retinal image is very likely to be chosen as the subject to be emphasized in a viewer-centered picture. Figure 2 is a typical viewer-centered work by the famous English artist J.M.W. Turner. We can see that he has made the front of train the focus by blurring out the details at other areas. The front of the train is likely to be what has drawn his attention at his first glance of the scene. Inspired by Turner's work, we came to the idea of employing saliency map [4], a computational model of visual selective attention, for automatically predicting the focus of attention in a viewer-centered picture. To accentuate the focus, we also developed a technique for controlling the level of details as well as the appearance of pencil strokes according to the degree of attention given by saliency map. Our major contributions can be summarized as following:

- A novel framework combining saliency map and LIC (Line Integral Convolution) for automatically generating viewer-centered pencil drawings with the accentuation effect.
- A new multi-resolution scheme for achieving accentuation effect through locally adapting various rendering parameters to the degree of attention.
- New techniques for improving the image quality of existing LIC-based pencil drawing generation method.
- An evaluation experiment for validating the effect of accentuation in inducing the viewer's visual attention.

Our experimental results show that the images generated with the proposed method present the similar accentuation effect as in the real pencil drawing and can successfully direct the viewer's attention toward the focus.

The remainder of the paper consists of the following sections. Section 2 reviews the related works. Section 3 briefly introduces the existing LIC-based pencil drawing generation technique and saliency map. Section 4 presents the details of the proposed technique by showing the basic idea, the overall procedure and the details of newly proposed techniques. Section 5 describes the evaluation experiment after showing some results. Section 6 concludes the paper.

2 Previous work

In this section, we review the related work from two aspects: pencil drawing generation and the application of visual attention models in image abstraction.

2.1 Pencil drawing generation

As one of the most popular artistic media, pencil drawing was also well explored for non-photorealist rendering in the past decade. There are mainly two approaches to addressing the pencil drawing computationally. The first approach is to provide physical simulation to the materials and skills, and has been mainly combined with interactive painting or 3D non-photorealistic rendering for generating realistic pencil drawing images. The second approach is the painterly filtering, which involves taking an image and applying some kind of image processing or filtering techniques to convert it into an image visually resembling a pencil drawing. Our technique takes the second approach. It takes a 2D image as the input and converts it into a pencil drawing like image automatically.

As simulation-based approach, Sousa and Buchanan developed an observation model of pencil drawings [5, 6] by investigating real pencil drawing with scanning electron microscope. Takagi et al. proposed to model the paper micro-structure and color pigment distribution as 3D volume data and use volume ray-tracing for rendering color pencil drawings [7]. Semet et al. used an interactive artificial ant approach to generate a pencil drawing from a reference image [8]. Melikhov et al. used a disk B-spline curves for converting an image into a pencil drawing strokes with physically simulated interior [9]. Lee et al. developed a GPU-based real-time technique for rendering 3D meshes using in the pencil drawing style [10].

More recently, AlMeraj et al. [11] designed an algorithm for creating lines that perceptually resembles human drawn pencil strokes, based on a physical model of human arm movements as well as an observational analysis of human drawn lines.

Among large number of pencil drawing generation techniques based on image processing or filtering [12–18], a major scheme is to use LIC (Line Integral Convolution). LIC is a texture-based flow visualization technique which visualizes a flow field by low-pass filtering a white noise along the local streamlines of the vector field. Inspired by the visual similarity between the pencil strokes and the traces of streamlines in LIC image, Mao et al. proposed a technique for automatically converting a 2D input image into a pencil drawing by convolving the dithered version of the input image with a vector field defining the directions of strokes [2]. The method has been extended for colored pencil drawing [12] and video [13, 14]. Recently, several more works have been done to add further improvements to the original LIC-based method [15–17].

To the best of our knowledge, however, all those previous works on pencil drawing put their efforts mainly on how to generate realistic strokes or hatching, and the issue of how to automatically generate the accentuated pencil drawing has not been explored yet. The proposed method extends the existing LIC scheme for automatically generating accentuated pencil drawings. We employ a new multi-resolution pyramid based approach to locally adapt the parameters of LIC to the degree of relevance so as to accentuate the focus of attention.

2.2 Application of visual attention models in image abstraction

Visual attention models have been explored by many researches for achieving the abstraction effect in painterly rendering. DeCarlo and Santella proposed a method for controlling the level of details in stylized images according to user's gaze information [19]. This method has the advantage of being able to reflect the real intention of a user, but has the shortage of requiring the eye-tracking which limits its usefulness in real applications. Our method uses saliency map to predict user's attention and hence it is fully automatic. Saliency map is a spatial map showing the strength (degree of saliency) of the visual attention. Since there is a limit in the throughput of the brain, the information of broad perspective cannot be processed in an instant. So human aggregates multiple features and then turns attention to where it unconsciously acquired as a more important place.

Saliency map has been used for generating other kinds of painterly images. Collomosse and Hall used saliency map to generate the images in oil-painting style [20]. The visual attention of an edge is detected and only those edges with high attention are drawn later. Winnemöller et al. proposed an automatic real-time video abstraction framework [21], in which areas with lower frequency are further filtered out so that the most informative area (the area consisting of higher frequency) can be further emphasized. Kyprianidis and Kang have published a series of excellent papers for high quality real-time image and video abstraction using different filtering kernels and edge-enhancement techniques [22–26]. Zhao et al. used saliency map to control the degree of abstraction [27]. Although our technique can be viewed as a kind of image abstraction technique and shares the same idea of using saliency map to define the degree of abstraction, we have developed a set of new techniques for applying the saliency map to the specific application of pencil drawing generation.

3 Preliminaries

As mentioned in previous sections, saliency map and LIC-based pencil drawing generation scheme are two important elemental techniques upon which we build the proposed techniques. As the preliminaries required for understanding the algorithm of the proposed technique, we briefly describe in this section the procedures for generating pencil drawing with LIC and for computing saliency map.

3.1 Pencil drawing generation using LIC

As depicted in Fig. 3, given a photograph, Mao et al.'s technique generates a pencil drawing image in the following 7 steps [2]:

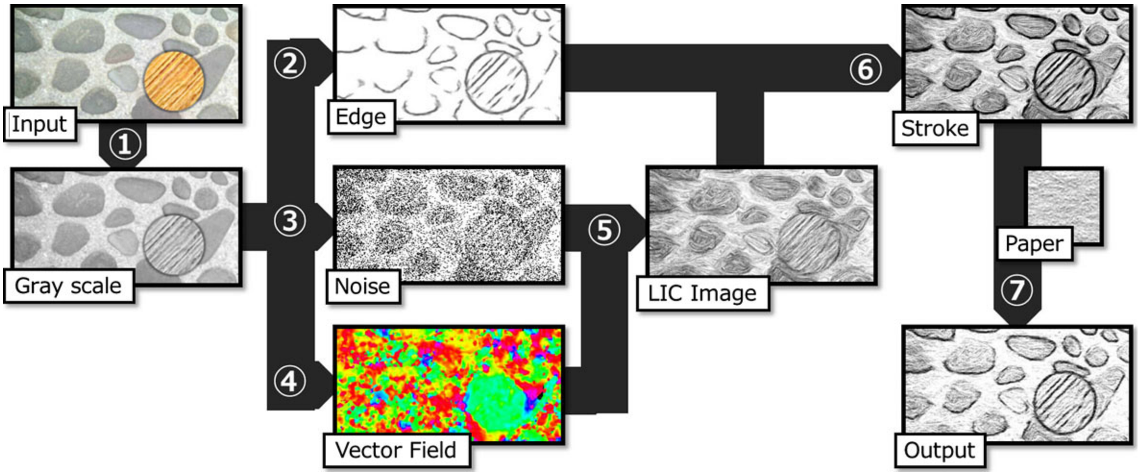


Fig. 3 Pencil drawing generation using LIC

1. Generate a gray scale image from the input image.
2. Generate an edge image from the grayscale image.
3. Generate the noise image by applying random dithering to the grayscale image generated at Step 1.
4. Generate the vector field either by assigning a random direction to each segmented region or using the gradient of the input image.
5. Applying LIC to the white noise and the vector field.
6. Generate stroke image by adding the edge image to the LIC image.
7. Obtain the output image (pencil drawing) by compositing the stroke image with the paper sample.

By using the dithered image as the noise, the resulting pencil drawing has a tone matching that of the input image. Changing the granularity of the noise or the length of the 1D convolution kernel in LIC can result in pencil strokes of different widths or lengths. We will describe in the next section how we adapt those parameters to the local saliency value to achieve the accentuation effect.

3.2 Saliency map

We use Itti et al.'s model [4] to compute the saliency map.

As depicted in Fig. 4, firstly, intensity, color (red, green, blue and yellow) and orientation (degrees of 0°, 45°, 90°, and 135°) are extracted in multi-resolution from the Gaussian pyramid and Gabor pyramid. Then a Feature Map is generated for each of them by computing the difference between the layers of the pyramids, which imitates the center-surround type receptive field. The multi-resolution layers of each Feature Map are combined into a Conspicuity Map where similar features are suppressed and distinct features are further emphasized nonlinearly. Finally, the intensity, color and orientation of Conspicuity Map are unified to obtain the Saliency Map which topographically encodes for conspicuity (or "saliency") at every location in the input image.

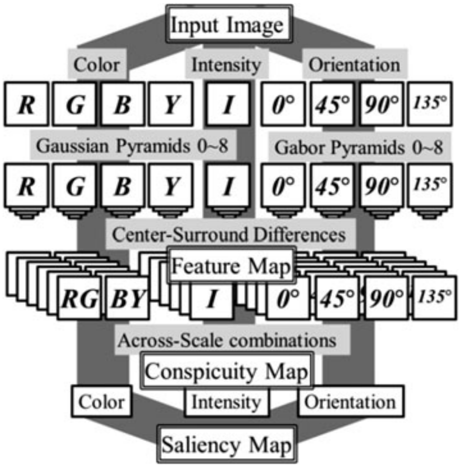


Fig. 4 Computational model of saliency map

4 Proposed method

4.1 Basic idea

In the real pencil drawings, artists achieve the effect of accentuation by controlling not only the level of details but also the appearance of strokes. Figure 5 is an example of pencil drawing by an artist. The focusing region (area enclosed by the dotted line) is drawn in fine details with clear and sharp strokes. The periphery area between the dotted line and dashed line is drawn with rough and flat strokes and the outer region is simply omitted. To achieve the similar effect, the proposed method uses a Gaussian filtered version of saliency map, called Draw Map, to define the relevance of each area. To locally adapt the density and appearance of strokes to the relevance, we first build a Gaussian pyramid from the input image, and then select appropriate resolution from the pyramid according to the saliency value in the Draw Map for computing the noise, edge and vector field.

4.2 Overall procedure

Given an arbitrary input image, the proposed method automatically converts it into an accentuated pencil drawing in the following 9 steps (Fig. 6):

1. Generate a saliency map from the input image.Smooth the saliency map to obtain the Draw Map.
2. Generate a grayscale image from the input image. Build a multi-contrast Gaussian pyramid with descending contrast along with the decreasing of resolution.
3. Generate a Multi-Resolution Image by selecting pixels from different layers of the multi-contrast Gaussian pyramid according to the pixel's saliency value in the Draw Map.
4. Generate an edge image from the Multi-Resolution Image.
5. Generate a vector field with the Gabor pyramids which have been used for extracting orientation features when computing saliency map.
6. Generate a noise pyramid by applying random dithering to each layer of the Gaussian pyramid.
7. Apply LIC to the noise pyramid and vector field to generate a LIC pyramid. Then generate a multi-resolution LIC image by selecting the pixels at appropriate layers of the LIC pyramid based on the Draw Map.
8. Generate stroke image by adding multi-resolution edge image to the multi-resolution LIC image.
9. Obtain the Output Image (pencil drawing) by composite of the stroke image and the paper sample.

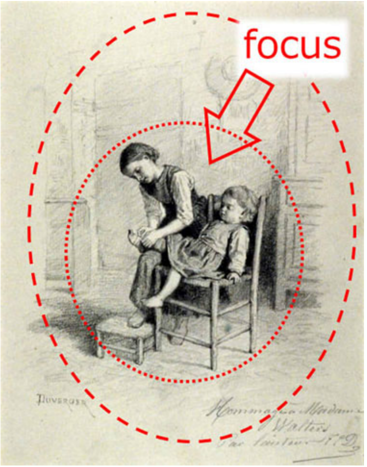


Fig. 5 A focus and its rendering. An accent is attached by the contrast of strokes. The inner side of the *dotted line* is distinctly drawn compared with the area between the *dotted line* and the *dashed line*. And the outside of the *dashed line* is omitted. From The Walters Art Museum (http://art.thewalters.org/detail/15810/little-girl-dressing-her-little-brother/) (CC BY-NC-SA 3.0)

4.3 Techniques

4.3.1 Draw Map

In saliency map, the saliency value is nonlinearly emphasized or suppressed to simulate the lateral inhibition mech-

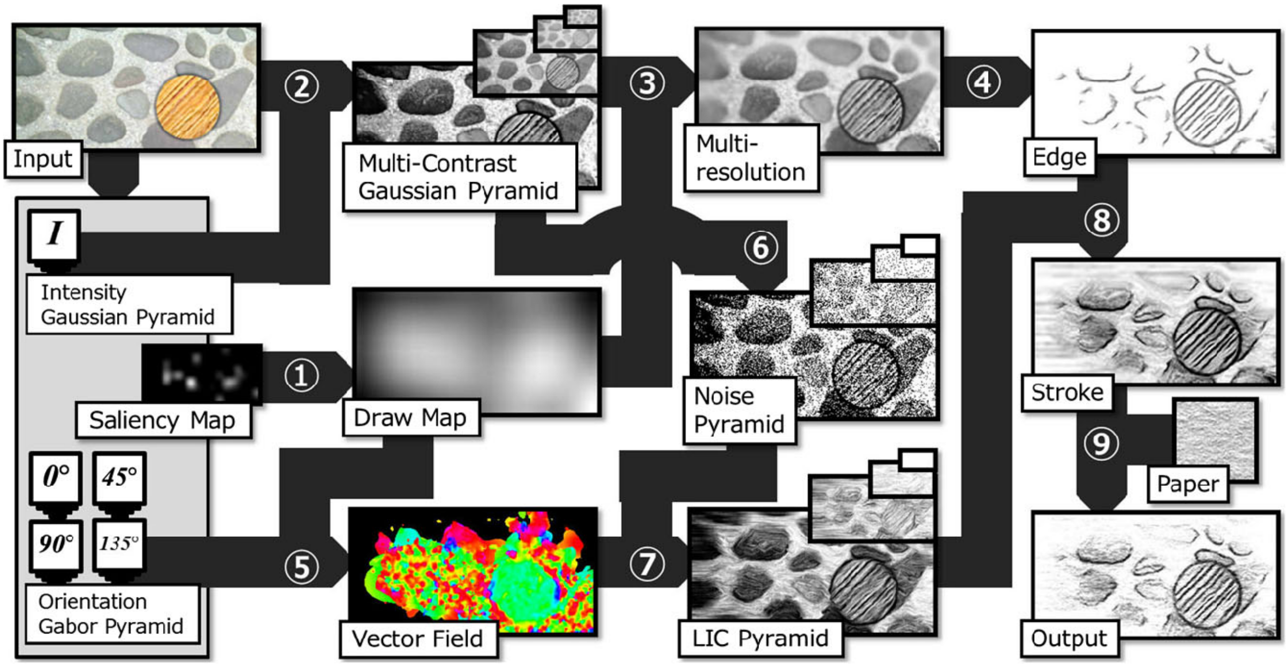


Fig. 6 Overall procedure of the proposed method

anism (Fig. 11(b)) [4]. For this reason, if we simply use the original saliency map for controlling the local rendering parameters, almost no strokes would be drawn for most areas of the image except for a very small region around the position with highest saliency value (Fig. 11(d)). However, as shown in Fig. 5, the periphery of the focus is usually also drawn to provide the context for the focusing area in real pencil drawings. It is psychologically known that periphery vision is very important in scene recognition [28]. Here the context in periphery, rendered with strokes of different appearance can provide a further cue to emphasize the focusing region.

To solve this problem, we first suppress the nonlinear effect of saliency map by taking a square root of the saliency value and then further smooth it with a Gaussian filter. Since it is known that the best periphery size is about the half of the scene [28], we set the standard deviation σ of the Gaussian filter to be $1/12$ of the image size, so that the filter supports half of the image size. We call the resulting smoothed map Draw Map and use it instead of the original saliency map for locally controlling the parameters of strokes. Figure 11(e) is a pencil drawing image generated by using the Draw Map of Fig. 11(c), where we can see an accentuation effect similar to Fig. 5 has been achieved.

4.3.2 Multi-Contrast Gaussian Pyramid

To make the strokes look sharper and clearer in the focusing area, we use an image of enhanced contrast for computing the parameters of strokes for the focusing area and use images of low contrast for other areas. To get the local control over the contrast, we process the Gaussian pyramid to have a descending contrast along with the decreasing of resolution to get a Multi-Contrast Gaussian Pyramid. Thus, by using the appropriate layers from the Multi-Contrast Gaussian Pyramid for generating noise as well as for detecting edges, we can locally adapt the sharpness of strokes to the saliency value.

4.3.3 Multi-Resolution Image (integration of pyramid layers based on Draw Map)

For keeping a smooth transition between the areas of different level of detail, instead of computing edges from Multi-Contrast Gaussian Pyramid directly, we first generate a single Multi-Resolution Image by choosing pixels from the suitable layer of the pyramid according to the saliency value in the Draw Map. Denoting the depth of the layer with the highest resolution as N, the layer r from which the pixel (x, y) in the Multi-Resolution Image should be chosen is calculated by the following equation:

$$r = DM(x, y) \cdot N \tag{1}$$

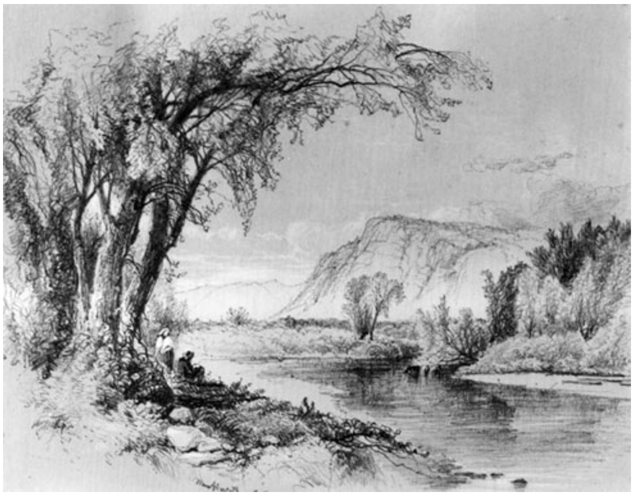


Fig. 7 Edges in real pencil drawing. From The Walters Art Museum (http://art.thewalters.org/detail/11007/mountain-and-river-scene/) (CC BY-NC-SA 3.0)

where $DM(x, y) \in [0, 1]$ is the saliency value in the Draw Map. Since r is usually a floating number, we calculate $MI(x, y)$, the value of the pixel (x, y) in Multi-Resolution Image, by linearly interpolating the values of the pixels on the two adjacent layers of the pyramid:

$$\begin{aligned} MI(x, y) = {} & (1 - a) \cdot PI\big(floor(r), x, y\big) \\ & + a \cdot PI\big(floor(r) + 1, x, y\big), \\ a = {} & r - floor(r), \end{aligned}$$

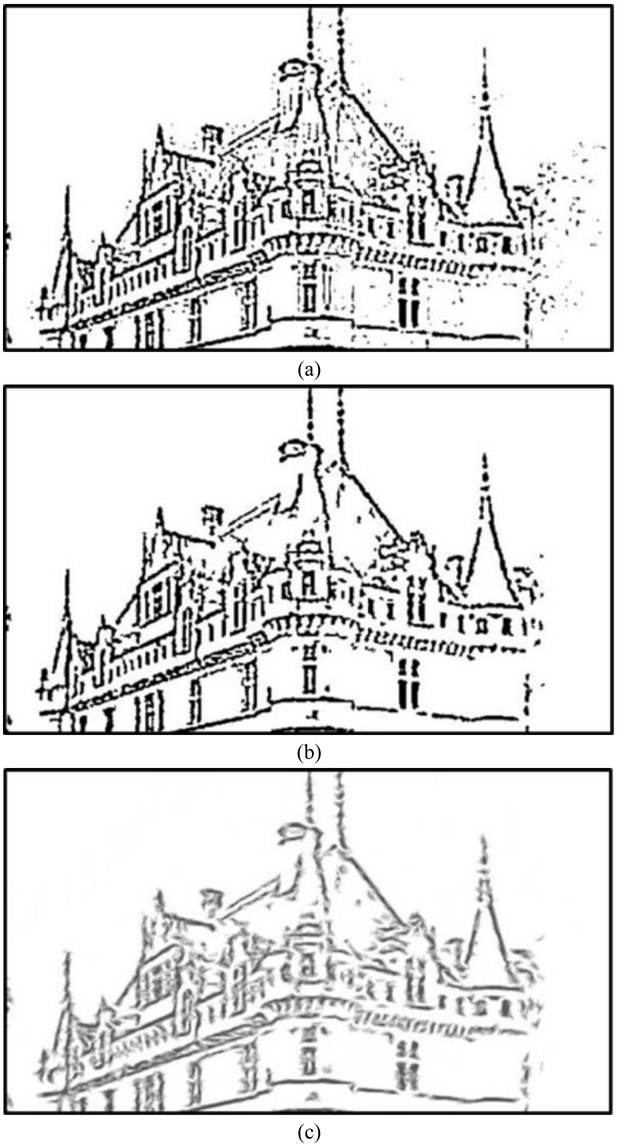
where $PI(n, x, y)$ is the value of the pixel (x, y) at the nth $\in [0, N]$ layer of pyramid.

4.3.4 Edge

As shown in Fig. 7, edge is also an important visual factor contributing to the effect of accentuation. The density and strength of the edge should decrease as getting further from the focus. This effect can be achieved naturally by detecting the edges from the Multi-Resolution Image on which the frequency and contrast decrease with the decrease of saliency value. DoG filter is commonly used for the edge detection in non-photorealistic rendering applications [22, 23, 27] because it is relatively insensitive to noise compared with differential filters. But we choose *adaptive thresholding* method [29] for achieving an easy control over the width of edges. The edge image $ATh(x, y)$ is calculated in the following way:

$$ATh(x, y) = \begin{cases} 1, & MI(x, y) > Ave(x, y) - E \\ 0, & \text{otherwise} \end{cases}$$

where $Ave(x, y)$ is the average of the $5 * 5$ neighborhood of pixel (x, y). E is an offset to control the density and width


(a)

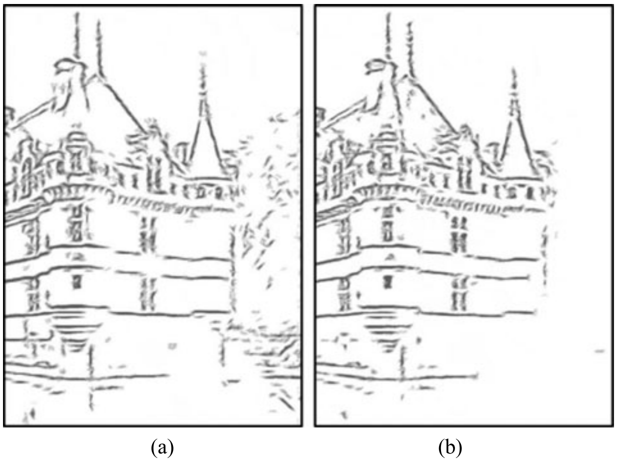
(b)

(c)

Fig. 8 Noise removal of edge refinement. (**a**) Result of the adaptive threshold. (**b**) Noise removal. (**c**) LIC enhancement

of the edges. By increasing E, fewer and narrower edges are detected.

The edge image obtained by the adaptive thresholding may contain short edges and noise as shown in Fig. 8(a). We remove those short edges and noise by eliminating the edge pixel whose neighborhood has low average edge strength. Figure 8(b) is the result obtained by eliminating the edge pixels whose $5 * 5$ neighborhood has small edge strength. Finally Line Integral Convolution is performed on the edge image to produce coherent edges with the vector field described in the next paragraph. Since LIC blurs the image along local streamlines of the vector field, applying LIC to the edge image results in the edge enhancement in a way similar to flow-based edge enhancement approaches [22, 23, 27]. Furthermore, LIC filtering gives the edge the appearance of pencil stroke matching the strokes in other interior regions. Figure 8(c) shows the result of applying LIC along the edge.


(a)

(b)

Fig. 9 Accentuation of edges. (**a**) Uniform edges. (**b**) Accentuated edges with their density, width and strength adapted to local saliency value

By applying the above edge detection as well as the edge enhancement technique to the Multi-Resolution Image generated according to Draw Map, we can obtain the edge images with the density and appearance of edges that are adapted to local saliency value. Figure 9(b) is an example of such a result. Compared with the edge image generated without using Draw Map (Fig. 9(a)), we can see the result of proposed technique (Fig. 9(b)) which resembles the accentuation effect of the work drawn by artist (Fig. 7).

4.3.5 Vector field

Existing LIC-based methods used gradient and Fourier analysis for defining the vector field representing local texture directions. We chose to use the result of Gabor filtering for three reasons: (1) Gradient method is very sensitive to noise; (2) Fourier analysis method usually cannot generate a correct vector field except for the case when the local texture has a very uniform direction; (3) Gabor filter is known to be biologically correct texture descriptor and we already have the multi-resolution Gabor pyramid for 4 orientations (0°, 45°, 90°, 135°) available during the process for generating Saliency Map.

Denoting by λ the wavelength, σ the standard variation, θ the orientation, ψ the phase offset, and γ the spatial aspect ratio, the Gabor filtering of image I can be represented as the following convolution:

$$GF_{\lambda,\sigma,\theta,\varphi}(x,y) = e^{-\frac{\acute{x}^2+\gamma^2\acute{y}^2}{2\sigma^2}} \cos\left(2\pi\frac{\acute{x}}{\lambda}+\varphi\right)$$

$$\acute{x} = x\cos\theta \ + y\sin\theta, \qquad \acute{y} = -x\sin\theta + y\cos\theta$$

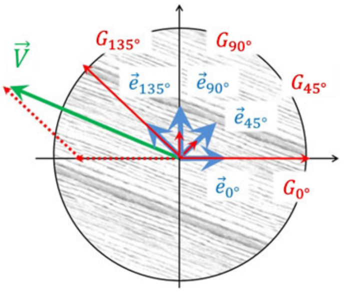


Fig. 10 Calculating local orientation by compositing the 4 vectors detected by Gabor filtering

We can get the strength of the various orientation components by changing θ of Gabor filter.

Gabor energy is obtained by calculating the root-sum-square between a Gabor and a 90°-phase-shifted version:

$$G_\theta(x,y) = \sqrt{\left(GI_{\theta,0}(x,y)\right)^2 + \left(GI_{\theta,-\frac{\pi}{2}}(x,y)\right)^2}$$

$$GI_{\theta,\varphi}(x,y) = I(x,y) * GF_{\theta,\varphi}(x,y)$$

$$= \sum_k \sum_l I(k,l) GF_{\theta,\varphi}(x-k, y-l)$$

As shown in Fig. 10, we detect the direction vector at each pixel as the composition of vectors obtained by scaling the four unit vectors in the directions of 0°, 45°, 90°, and 134°.

By subtracting the Gabor energy in each of the four directions with the minimum energy among the four, we can avoid false detection of a vector in case the pixel does not have an explicit direction.

As shown in Fig. 10, when the direction of the texture is between 135° and 180°, instead of 0°, unit vector of 180° should be used for the composition.

$$\vec{V} = \begin{cases} \sum_{\substack{\theta=180°,45°,\\ 90°,135°}} (G_\theta - m)\vec{e}_\theta, \\ \quad G_{135°} > max(G_{45°}, G_{90°}) \\ \sum_{\substack{\theta=0°,45°,\\ 90°,135°}} (G_\theta - m)\vec{e}_\theta, \quad \text{otherwise} \end{cases}$$

$$m = min(G_{0°}, G_{45°}, G_{90°}, G_{135°})$$

$$G_{180°} = G_{0°}$$

Again, using the Gabor pyramid we can adapt the level of details of the orientations to that of the saliency value. As we can see in a rabbit image in Fig. 10, with our Gabor pyramid based orientation detection method we can generate the strokes well depicting the orientation of local texture.

4.3.6 *Noise*

In order to achieve the omission of a stroke in the irrelevant areas, we reduce the density of black pixels at the low-resolution layer when generating the noise. This is realized by adjusting the intensity of Multi-Contrast Gaussian Pyramid according to the depth of the layer before performing dithering. Thus the value at pixel (x,y) on the nth layer of noise pyramid NP is calculated in the following way:

$$NP(n,x,y) = \begin{cases} 0, & 1 + \frac{n}{N}(PI(n,x,y) - 1) < T \\ 1, & \text{otherwise} \end{cases}$$

where T is the threshold of random dithering.

4.3.7 *LIC image*

We apply LIC to the vector field and noise pyramid to get an LIC pyramid, and then generate the multi-resolution LIC from the LIC pyramid by referring to the Draw Map. Since the process of generating the multi-resolution LIC requires to up-sample the low resolution layer to have the same size as the highest resolution, it results in rough and wide strokes naturally at the area with low saliency value.

5 Results

5.1 Implementation

We implemented the proposed technique using C# and Emgu CV (wrapper to the OpenCV). We used iLab Neuromorphic Vision C++ Toolkit VirtualBox (http://ilab.usc.edu/toolkit/) for computing the saliency map. Depth of the pyramid is 4 layers ($N = 3$). The length of the 1D filter kernel used in LIC is 10 for the layer of highest resolution. Part of images are from Berkeley Segmentation Data Set (http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/) and PublicDomainPictures.net (http://www.publicdomainpictures.net/).

5.2 Results

Figure 11 demonstrates the effect of adapting the rendering parameters to the saliency value. Figure 11(b) is generated by only adapting the level of details to the saliency value while Fig. 11(a) also changes the directions, width and length of strokes, the density and strength of edges. It is obvious that by changing the rendering parameters, the focusing area can be further accentuated making the image more expressive and charming.

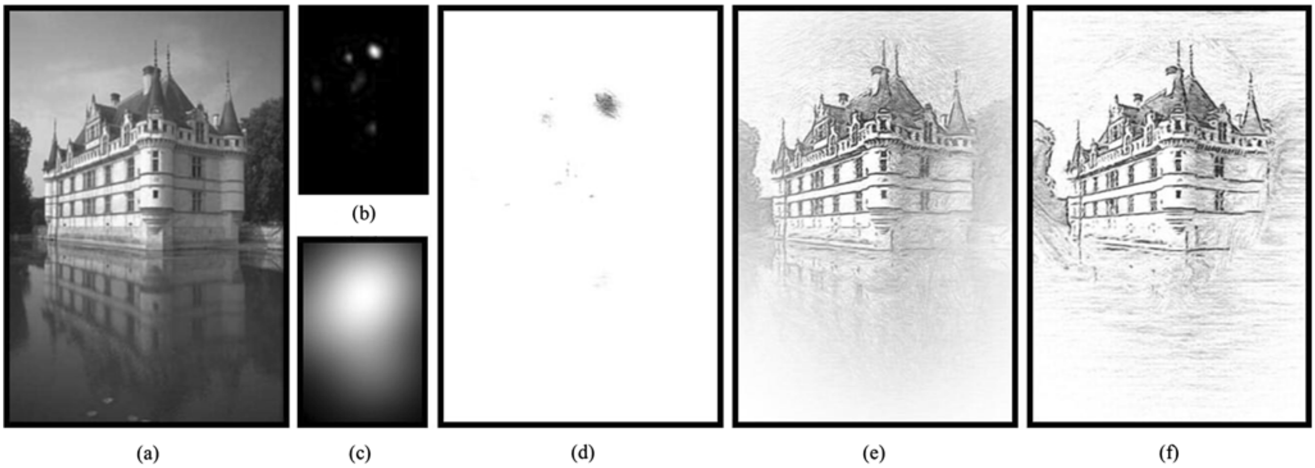


Fig. 11 Draw Map and rendering parameter control. (**a**) Input image. (**b**) Saliency Map. (**c**) Draw Map. (**d**) A result generated with the proposed scheme but using the original saliency map. (**e**) A result generated with the proposed scheme but only adjusting the intensity with Draw Map. (**f**) A result generated with the proposed method which adjusting all the rendering parameters with Draw Map

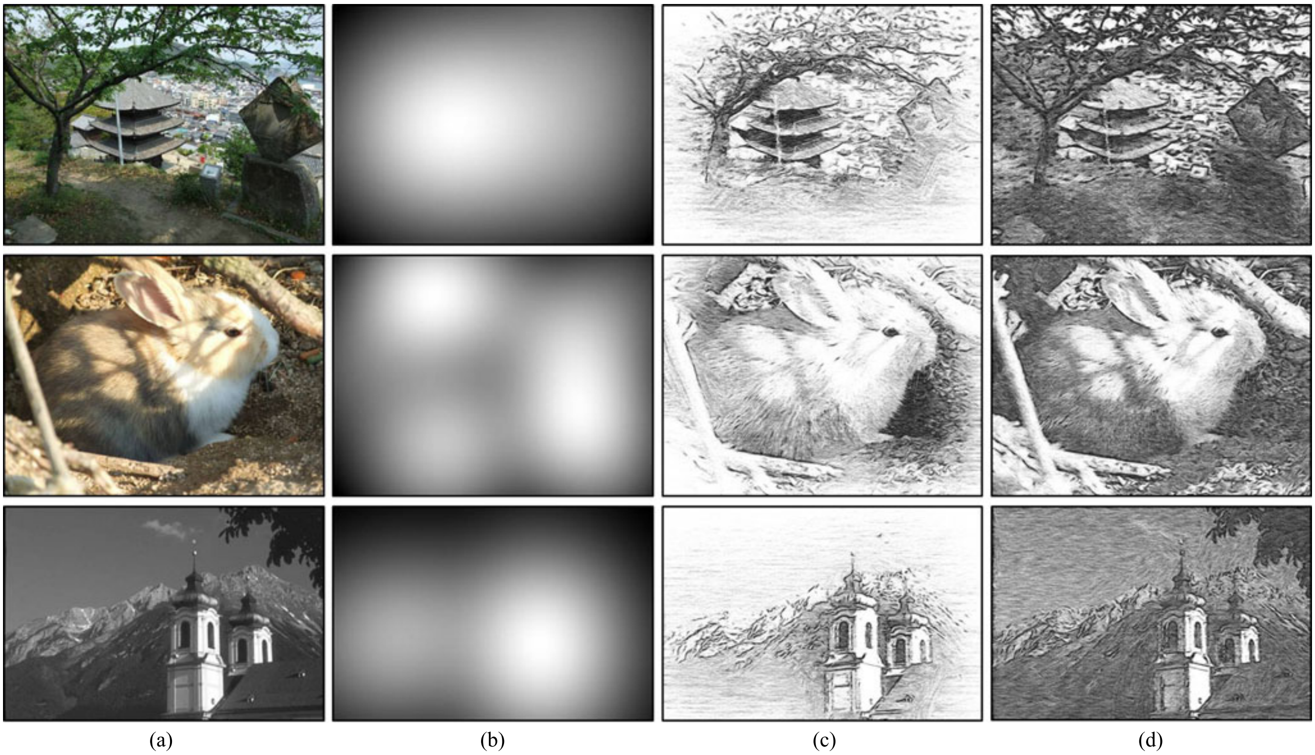


Fig. 12 Comparison of results. (**a**) Input images. (**b**) Draw Map. (**c**) Results of proposed method. (**d**) Results of Mao et al.'s method [2]

Figure 12 shows several more results generated with our new techniques. For comparison, the images generated with the existing LIC-based method [2] are also shown. We can see from the images that compared with the result of existing method, the focus area has been emphasized by eliminating irrelevant details in the remaining area. The change of rendering parameters also makes the image more charming and pleasing as an art work.

The time required of generating a $962 * 642$ image is 51 minutes on a Core2Duo 2.26-GHz PC. This does not include the time for computing saliency map. The most time-consuming parts are the local orientation detection based on Gaussian pyramid and LIC. Since fast LIC [30] and Gabor filtering algorithm using GPU [27] are already available, we can expect to improve the speed of the proposed technique by utilizing those acceleration techniques.

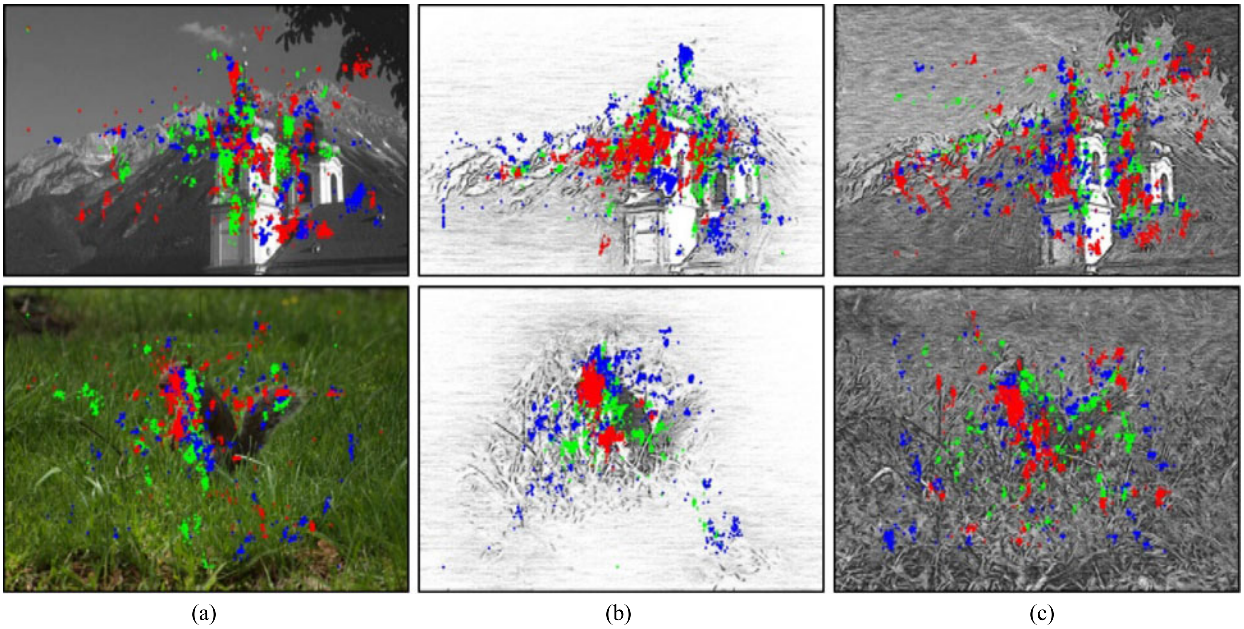


Fig. 13 Eye-tracking data (30 seconds). (**a**) Input. (**b**) Results of proposed method. (**c**) Results of Mao et al.'s method [2]

5.3 Evaluation

We have conducted an experiment to validate whether the pencil drawing images generated with the proposed technique have the effect of directing the viewer's attention to the focus area. The subjects are divided into 3 groups. Group 1 is presents the original images, Group 2 is presented with the pencil drawing images generated with the existing LIC-based method, and Group 3 is presented with the images generated with the proposed method. Each group has 3 subjects and there is no overlapping of subjects among these groups. In other words, each subject watched only one version for each image. We recorded the eye positions of subjects for the first 30 seconds using EMR-AT VOXER eye-tracker with a sampling rate of 60 Hz from NAC Image Technology, Inc. Figure 13 visualizes the eye positions of the 3 subjects (each in different color) over the images. We can see, for all 3 different images that the eye positions on the images generated with the proposed technique are more concentrated around the focusing area. Note that the eye positions on the images generated with the existing method are even more scattered than those on the original images. From the evaluation results we can conclude that the images generated with the proposed technique do have the effect of drawing the viewer's attention toward the focusing area.

5.4 Discussion

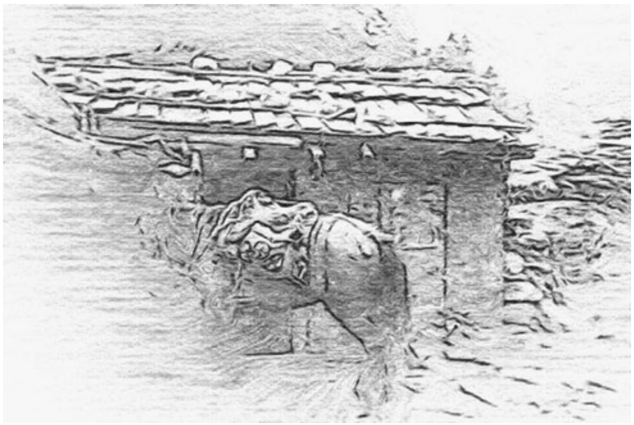


Fig. 14 An example of failure

Figure 14 shows an example of failure. Only half of the horse's face was drawn. Saliency map is a bottom-up model based on low-level visual cues and hence it may fail to predict human attention correctly since our attention is also top-down and task-based. Some other factors may influence the visual attention. For example, it is known that humans tend to attend the pattern of a face. Combining the newest face detection technique may help us improve the results for an image including human objects. Recently, several more sophisticated visual attention models have been developed. Judd et al. created a saliency model which uses machine learning technique to learn the visual attention from the eye-tracking results [31]. Since this model could partially learn the top-down visual attention, we can expect to generate pencil drawings more similar to human drawn works by employing their model.

6 Concluding remarks

We proposed a novel method for automatically converting an input image into pencil drawing. By using saliency map to predict the attention of a viewer and by adapting the local rendering parameters to the saliency value, we succeeded in generating viewer-centered pencil drawing images simulating the effect of emphasis and eliminations found in human pencil drawings. We are now improving the method by employing a more sophisticated visual attention model and some newest computer vision technologies. Through such an effort we expect to be able to partially simulate the effect found in object-centered pencil drawings.

Acknowledgement The authors thank Mr. T. Sawada from University of Yamanashi for his help in conducting the evaluation experiment. This work was supported by KAKENHI 21300033 Grant-in-Aid for Scientific Research (B), Japan Society for the Promotion of Science (JSPS).

References

1. Guptill, L.A.: Rendering in Pencil. Watson-Guptill, New York (1977)
2. Mao, X., Nagasaka, Y., Imamiya, A.: Automatic generation of pencil drawing from 2D images using line integral convolution. In: Proc. of the 7th International Conference on Computer Aided Design and Computer Graphics (CAD/GRAPHICS'01), pp. 240–248 (2001)
3. Willats, J.: Art and Representation: New Principles in the Analysis of Pictures. Princeton University Press, Princeton (1997)
4. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
5. Sousa, M.C., Buchanan, J.W.: Observational model of blenders and erasers in computer-generated pencil rendering. In: Graphics Interface, vol. 1999, pp. 157–166 (1999)
6. Sousa, M.C., Buchanan, J.W.: Computer-generated graphite pencil rendering of 3D polygonal models. Comput. Graph. Forum **18**(3), 195–208 (1999)
7. Takagi, S., Fujishiro, I., Nakajima, M.: Volumetric modeling of colored pencil drawing. In: Pacific Graphics '99 Conference Proceedings, pp. 250–258 (1999)
8. Semet, Y., O'Reilly, U., Durand, F.: An interactive artificial ant approach to non-photorealistic rendering. In: GECCO (1), pp. 188–200 (2004)
9. Melikhov, K., Tian, F., Xie, X., Seah, H.S.: DBSC-based pencil StyleSimulation for line drawings. In: Proc. of 2006 International Conference on Game Research and Development, pp. 17–24 (2006)
10. Lee, H., Kwon, S., Lee, S.: Real-time pencil rendering. In: Proc. NPAR, pp. 37–45 (2006)
11. AlMeraj, Z., Wyvill, B., Isenberg, T., Gooch, A., Richard, G.: Automatically mimicking unique hand-drawn pencil lines. Comput. Graph. **33**(4), 496–508 (2009)
12. Yamamoto, S., Mao, X., Imamiya, A.: Colored pencil filter with custom colors. In: Proc. of Pacific Graphics 04, pp. 329–338 (2004)
13. Yamamoto, S., Mao, X., Imamiya, A.: Enhanced LIC pencil filter. In: Proc. of the International Conference on Computer Graphics, Imaging and Visualization 04, pp. 251–256 (2004)
14. Xie, D., Zhao, Y., Xu, D., Yang, X.: Convolution filter based pencil drawing and its implementation on GPU. In: Lecture Notes in Computer Science, vol. 4847, pp. 723–732 (2007)
15. Chen, Z., Zhou, J., Gao, X., Li, L., Liu, J.: A novel method for pencil drawing generation in non-photo-realistic rendering. In: Lecture Notes in Computer Science, vol. 5353, pp. 931–934 (2008)
16. Yang, H., Min, K.: Feature-guided convolution for pencil rendering. In: TIIS, pp. 1311–1328 (2011)
17. Kwon, Y., Min, K.: Texture-based pencil drawings from pictures. In: ICHIT (2), pp. 70–77 (2011)
18. Matsui, H., Johan, J., Nishita, T.: Creating colored pencil images by drawing strokes based on boundaries of regions. In: Proc. of Computer Graphics International 05, pp. 148–155 (2005)
19. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. ACM Trans. Graph. **21**(3), 769–776 (2002)
20. Collomosse, J.P., Hall, P.M.: Painterly rendering using image salience. In: Proceedings 20th Eurographics UK Conference, pp. 122–128 (2002)
21. Winnemöller, H., Olsen, S., Gooch, B.: Real-time video abstraction. ACM Trans. Graph. **25**(3), 12–26 (2006)
22. Kang, H., Lee, S., Chui, C.: Flow-based image abstraction. IEEE Trans. Vis. Comput. Graph. **15**(1), 62–76 (2009)
23. Kyprianidis, J.E., Döllner, J.: Image abstraction by structure adaptive filtering. In: Proc. EG UK Theory and Practice of Computer Graphics, pp. 51–58 (2008)
24. Kyprianidis, J.E., Kang, H., Döllner, J.: Image and video abstraction by anisotropic Kuwahara filtering. Comput. Graph. Forum **28**(7), 1955–1963 (2009)
25. Kyprianidis, J.E.: Image and video abstraction by multi-scale anisotropic Kuwahara filtering. In: Proc. 9th Symposium on Non-Photorealistic Animation and Rendering (NPAR) (2011)
26. Kyprianidis, J.E., Kang, H.: Image and video abstraction by coherence-enhancing filtering. In: Computer Graphics Forum, vol. 30, pp. 593–602 (2011). Special issue on the Eurographics
27. Zhao, H., Jin, X., Shen, J., Mao, X., Feng, J.: Real-time feature-aware video abstraction. Vis. Comput. **24**(7), 727–734 (2008)
28. Saida, S., Ikeda, M.: Useful visual field size for pattern perception. Percept. Psychophys. **25**(2), 119–125 (1979)
29. Linda, G.S., George, C.S.: Computer Vision. Prentice Hall, New York (2001)
30. Qin, B., Wu, Z., Su, F., Pang, T.: GPU-based parallelization algorithm for 2D line integral convolution. In: ICSI, vol. 1, pp. 397–404 (2010)
31. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV) (2009)

Michitaka Hata received the B.Sc. and M.Sc. degrees in Engineering from University of Yamanashi, Japan. He is a Ph.D. student in Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Japan. His research interests include non-photorealistic rendering and visualization.

Masahiro Toyoura received the B.Sc. degree in Engineering, M.Sc. and Ph.D. degrees in Informatics from Kyoto University. He is currently an Assistant Professor at Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi, Japan. His research interests are augmented reality, computer and human vision. He is a member of IEEE Computer Society.

Xiaoyang Mao received her B.Sc. in Computer Science from Fudan University, China, M.Sc. and Ph.D. in Computer Science from University of Tokyo. She is currently a Professor at Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi, Japan. Her research interests include texture synthesis, non-photo-realistic rendering and their application to scientific visualization. She is a member of ACM SIGRRAPH and IEEE Computer Society.