Preliminary Study on Vision-based Pen-and-Ink Drawing by a Robotic Manipulator

Yan Lu, Josh H. M. Lam and Yeung Yam, *Senior Member, IEEE*

Abstract—**This paper presents a robotic system that can perform automated pen-and-ink drawing based on visual feedback, using a proposed algorithm for stroke trajectory planning. The algorithm first converts the outlines of an input image to stroke trajectory according to the structural importance. Then, iterative hatching is carried out to convey both the tone and textures of the original image; in this process, visual feedback is employed to determine stroke positions, and local gradient interpolation is applied to guide stroke orientations. Finally, the drawing automatically terminates at the minimum point of a proposed criterion function, so that the drawing performance is not only robust to various input image tones, but also convenient to be tuned for a different style. Experimental results demonstrate that the proposed algorithm can create desirable pen-and-ink works in a robust way.**

I. Introduction

Pen-and-ink drawing is a traditional but popular form of art, which can express a wealth of textures, tones, and styles just with monochromatic pen strokes. While creating a beautiful pen-and-ink work is a fond dream of many people, it's really painstaking to master the needed drawing skills, and it's also quite time-consuming even for artists to do such elaborate work. Therefore, a lot of studies have been conducted during the past decades, to help people create art works in an easy way.

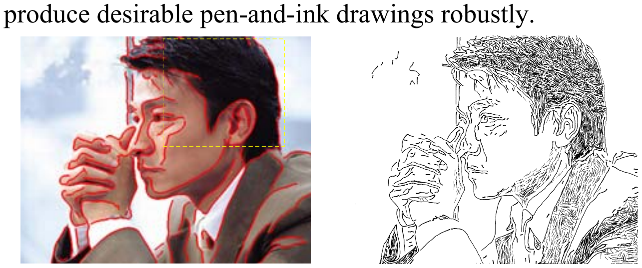
In computer graphics (CG), Non-Photorealistic Rendering (NPR) has developed quickly since its emergence in the early 1990s, which mainly deals with the computer generation of images and animations that appear to be made "by hand". Kinds of art forms have been studied [1]-[4], including pencil drawing, oil painting, pen-and-ink drawing, and so on. Although these techniques have brought quite compelling art works to people, a common limitation of them is that their productions are mostly digital images (or animations) displayed on computer. Even though they can be printed onto paper or canvas, the effects can hardly match those generated with real art media like pen, brush or ink, especially in a way that an artist does.

At the same time, in robotics field, many research projects have tried to build artistic robots. For example, ISAC [5] is a robot that can track and mimic a human's hand trajectory, and another robot AARON [6] can create art works by adapting a geometrical model of a subject. Draw-Bot [7] focused on force feedback to sketch a preprogrammed shape, and [8] further considered both visual and force feedback along with grasping technology. Calinon at al. constructed a humanoid robot capable of drawing human portraits [9]. Regardless of the art style or performance of these robots, they all succeeded somewhat in mimicking a human's manner to create art works, in the sense that they conduct drawing by manipulator with real media. However, a close examination of their productions reveals that while some are not monochromatic but colored paintings (e.g. AARON), other sketch works (e.g. ISAC and [9]) are just rudimentary to be regarded as pen-and-ink works.

Inspired from the advanced methodology of NPR, this paper presents an automated pen-and-ink drawing system based on a robotic manipulator [10]. The motivation of this project is to adopt robotic techniques to facilitate artwork creations. This would augment the ability to those people with few artistic skills, and would reduce the time-consumption for professional artists.

The main challenge is that only monochromatic strokes can be used, but three key elements of the original image need to be conveyed simultaneously, as illustrated in Fig 1 (a): (1) the structural contents (somewhat as the red lines indicate), (2) the original tone (i.e., brightness and darkness), and (3) textures (e.g., the hair texture encircled by the yellow dashed box).

In addition, actual drawing is different from NPR whose output is simulated in pixel-based representation. Therefore, visual feedback is needed to monitor and guide the drawing progress, together with a proposed algorithm for: outline drawing, tone and texture expression, and a stop criterion. Experimental results (e.g. Fig 1 (b)) prove our system can produce desirable pen-and-ink drawings robustly.



(a) (b)

Fig. 1. Main challenge: (a) a sample input, (b) a sample output. Note how the output represents the structural contents, tone and textures of the input.

The rest of this paper is organized as follows: Section II introduces our robot, visual feedback, and some pen-and-ink

Manuscript received February 15, 2009. This work is supported in part by a grant from the Research Grant Council of Hong Kong (Project No. 417606), and affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies. Yan Lu, Josh H.M. Lam and Y. Yam are with the Mechanical and Automation Engineering Department, the Chinese University of Hong Kong, Hong Kong, China (email: ylu, hmlam, yyam@mae.cuhk.edu.hk, phone: 31634227, fax: 26036002).

knowledge, Section III presents our algorithm, Section IV shows experiment results, and conclusion comes finally in Section V.

II. BACKGROUND

A. Robot Platform

Our lab has developed an Intelligent Robotic Art System (called IRAS) for replicating and creating art works (e.g. Chinese calligraphy). As shown in Fig 2, it has a 5-DOF manipulator: x, y, z, roll, and pitch, and only the first 3 DOFs are used in this work. The resolution of movement in each axis is 0.1 mm, which can provide precise and high quality drawing. A gripper is located at the manipulator tip to hold drawing tools like pen, pencil or paint brush for different use. Below the manipulator is a planar stage, where the canvas or paper is fixed. A PID controller is used in the system positioning to enhance the drawing smoothness. Moreover, a CCD camera is mounted at the top to provide visual feedback.

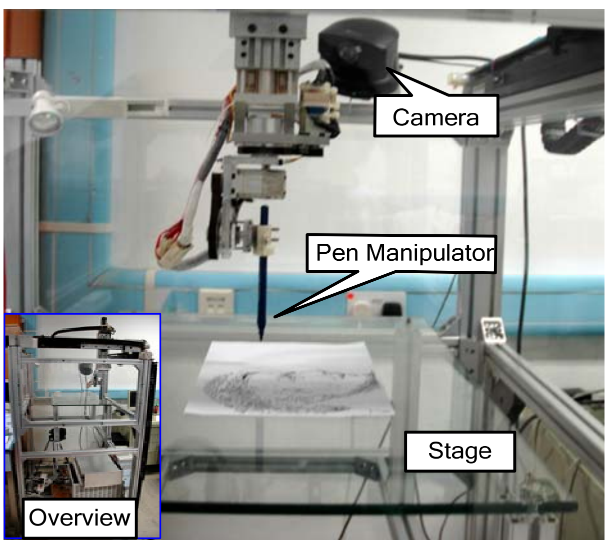


Fig. 2. IRAS: our drawing robot platform

B. Visual feedback

Since the camera is fixed at about 30 degrees looking downwards, the captured image is distorted as shown in Fig 3 (a). In theory, the projective distortion can be removed by selecting 4 reference points (totally 8 DOFs) on the drawing plane and mapping them to the 4 corners of the original image. Thus, homography is applied to re-project the captured image to the top view as if observed from above. The following is the equation for homography:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where (x, y) and (x', y') are respectively the coordinates of the captured image and the re-projected image, H is the projection matrix. To accurately obtain H, a GA-based approach proposed by our group [11] is used here, and the rectified image is shown in Fig 3 (b).

Fig. 3. Visual feedback: (a) captured image; (b) re-projected image

C. Pen-and-ink in NPR

Pen-and-ink has two major properties which distinguish it from other art forms. One is that a pen stroke contributes both tone and texture, so care must be taken to convey both these qualities simultaneously. The other is that strokes work collectively, that is, no single stroke is of critical importance; instead, strokes work jointly to express tone and texture [3].

The key to convey a tone is to place strokes accordingly. Salisbury et al. [4] proposed a concept of importance image to determine stroke positions by comparing the output and input. Suggestive as this method is, it still needs to be modified for our use, and visual feedback is also required in our approach, which will be discussed later.

While accurate stroke placement can facilitate the drawing to approximate the target tone, proper stroke orientations are necessary to express original textures. Traditionally, strokes are aligned orthogonal to the gradient directions at stroke positions. However, raw gradient data are often noisy, which tend to result in unpleasant patterns, especially in pen-and-ink drawing. To guide stroke orientations, Hays and Essa [13] applied radial basis function (RBF) to globally interpolate gradients from the strongest gradients. In this paper, we employ a new method using only local gradient information, which is more suitable for our work, as discussed later.

III. ALGORITHM

An overview of our algorithm is shown in Fig 4: we first extract and draw outlines, then convey original tone and textures by iterative hatching based on visual feedback, and finally terminate the drawing process with a criterion function.

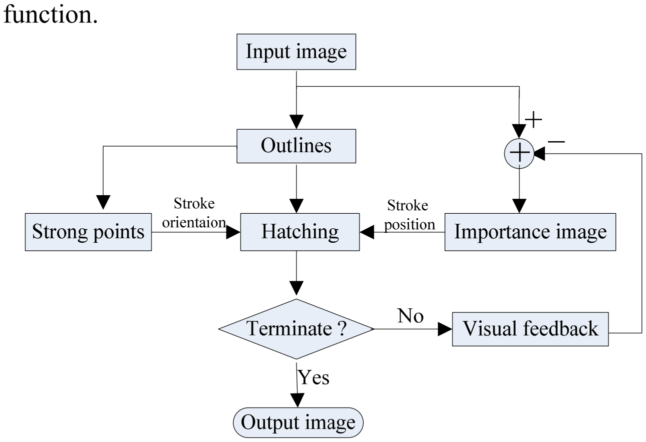


Fig. 4. Overview of our algorithm. Note that the calculation of "Importance image" is not only simple subtraction as illustrated here.

A. Outline drawing

Outlines play an important role in pen-and-ink drawing, for they can convey the structural contents of an image efficiently. In pen-and-ink, outline strokes are used not only for the contours of an object, but also for delineating the essentials of its interior [14]. Since they make different contributions to the structural contents, they should be drawn discriminatingly according to their structural importance.

Admittedly, gradient value is a good measure for the structural importance of an edge. However, if we use only gradient value for the measurement, it’s likely to omit some significant structural contents while giving unnecessary emphasis to some trivial details. This is because not all important structures in an image have large gradients.

In this paper, we employ the approach proposed by Orzan et al. [12], which measures the structural importance of edges by their “lifetime” in the Gaussian scale space of the image. This method first constructs a scale space for the image; at each scale, the image is convolved by the first order derivative of a Gaussian kernel of a certain variance; then the method extracts edges at all scales using Canny edge detector, and traces each edge from the finest scale to the scale where the edge disappears; finally the structural importance of an edge is measured by the “lifetime” the edge exists in scale space, as edges that live longer correspond to more stable structures.

Then, outlines of higher structural importance are drawn with thicker strokes than those of lower importance. From an example of this process in Fig 5, we can see that our approach conveys the image’s structures more explicitly than method that draws outlines with the same thick strokes. The reason is obvious: thicker stokes give emphasis to important (stable) structures, while thinner strokes act as necessary supplements to express fine structural information.

Fig. 5. Outlines: (a) original image; (b) structural importance (proportional to lightness); (c) outlines drawn with even thickness stroke; (d) outlines drawn by our method, strokes thickness is proportional to structural importance.

B. Tone expression

Hatching is further applied, which decorates the drawn image with fine strokes, so that the tone of input image can be expressed and approximated by the final drawing. In this process, the key is to locate each stroke at a right place.

One of the basic rules in pen-and-ink drawing is that strokes should be placed evenly: close together in dark areas, widely spaced in light areas. In order to drive the hatching process appropriately, we adopt and modify the notion of importance image introduced in [4].

At first, we need to define the notion of difference image I_{df} , whose value at each point is the tone (gray value) difference between two images, say one called object image and the other called operated image.

$$I_{df} = I_{oj} - I_{op} \tag{2}$$

where I_{oj} denotes the object image, and I_{op} denotes the operated image. In our case, the input image serves as the object image, and a blurred version of the drawn image is used as the operated image, so that the difference image can reflect the hatching progress. The drawn image is blurred by applying averaging filter of variable size, which increases with the lightness of the object area. This is based on the observation that each stroke has an effect of adding tone (darkness) to a local area; and the area size should be inversely proportional to the object gray value there, so that strokes in dark areas will be closely spaced and those in light areas will be sparse, as shown later.

In [4], the importance image I_{ip} is defined as:

$$I_{ip}^{i} = \begin{cases} I_{df}^{i} / I_{df}^{0} & I_{df}^{0} \neq 0 \\ 0 & I_{df}^{0} = 0 \end{cases} \tag{3}$$

where I_{ip}^{i} and I_{df}^{i} respectively denote the importance image and difference image after the ith stroke is drawn, and I_{df}^{0} denotes the initial difference image. By placing each stroke at the point whose value is the largest in current importance image, it’s expected that all regions approach their object tone at the same rate.

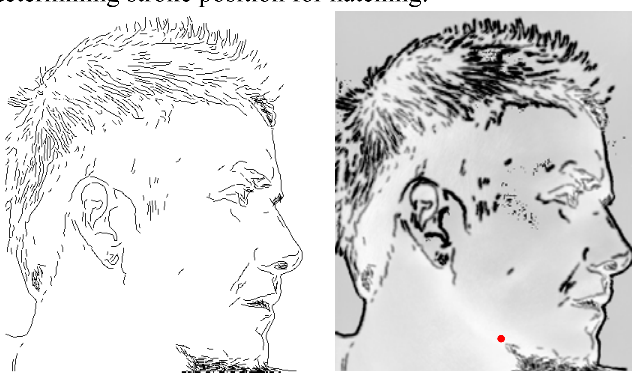
However, this method still suffers a problem. Consider the bright object areas, whose gray value is just a little larger than 0. Since the initial canvas is white (gray value is 0), when we try to place the 1st stroke, the importance value of these bright areas is all unit, equal to that of dark areas. Hence, it’s difficult to determine the stroke position, with so many points sharing the same largest value. This situation would not be alleviated as the hatching process continues.

To avoid this problem, we revise the definition of importance image as follows:

$$I_{ip}^{i} = \begin{cases} I_{df}^{i} / I_{df}^{0} + a \cdot I_{oj} & I_{df}^{0} \neq 0 \\ 0 & I_{df}^{0} = 0 \end{cases} \tag{4}$$

where I_{oj} denotes the object image, and a is a parameter adjusting the weight of I_{oj} . After modification, the importance image now takes the darkness of the object image into account, so that darker areas will take priority to be drawn when they have the same I^i_{df}/I^0_{df} as lighter ones. The parameter a is empirically set to be 0.1~0.5, because a too large value of a will fail areas to approach their object tone at the same or close rate, and this is also the reason for not directly using difference image to determine stroke position.

Whenever a stroke is drawn, its nearby points will have their values lowered in the importance image due to the blurring process before computing difference image. In this way, the next stroke is less likely to appear too close to the previous one, which further facilitates to maintain stroke separation needed in hatching. Fig 6 shows an example of determining stroke position for hatching.



(a) (b)

Fig. 6. Determine stroke position: (a) drawn image; (b) importance image (importance value is proportional to brightness). The next stroke position is marked by the red point, which corresponds to the largest importance value. The original image (i.e., the object image) is Fig 10 (a).

C. *Texture representation*

In pen-and-ink drawing, stroke orientations should be well organized, so that the drawn image can take on desirable textures. However, it's more complicated to determine stroke orientation than position. The reason lies in the fact that pen-and-ink, like any other form of art, allows a large variety of expression means. The same object can be represented in various ways by different artists, while none of the results can be simply judged to be the best. Even for the same drawing, two viewers may have contrary opinions. Therefore, what we've done is just a try to find a comparatively general way that can generate desirable textures with oriented strokes.

The gradient direction at point p in an image is:

$$\theta(p) = \tan^{-1}(Grad_y(p)/Grad_x(p)) \tag{5}$$

where $Grad_y$ and $Grad_x$ are the two components of the image gradient. However, gradient directions cannot be directly used to guide stroke orientations, because they are often noisy and therefore incredible especially when gradient magnitudes are small. An example is shown in Fig 7 (a) and (c), where stroke orientations are directly guided by gradient directions. It's obvious that the original textures (e.g. in the hair region) are disturbed by strokes with erratic orientations.

Since strokes work collectively, a local texture should be conveyed by a group of strokes which share consistent orientations. Thus, we need to reestimate the orientation for each stroke. As in [13], we accept the gradient data from the points with strong gradient magnitudes, and use them to calculate stroke orientations. In [13], radial basis function (RBF) is applied to globally interpolate gradients using all the strong gradients. However, common sense tells us that the orientation of a stroke should not be affected by points too far away; otherwise, the result would be somewhat unreasonable. This is the very problem involved in [13], and the method is also time-consuming due to global interpolation.

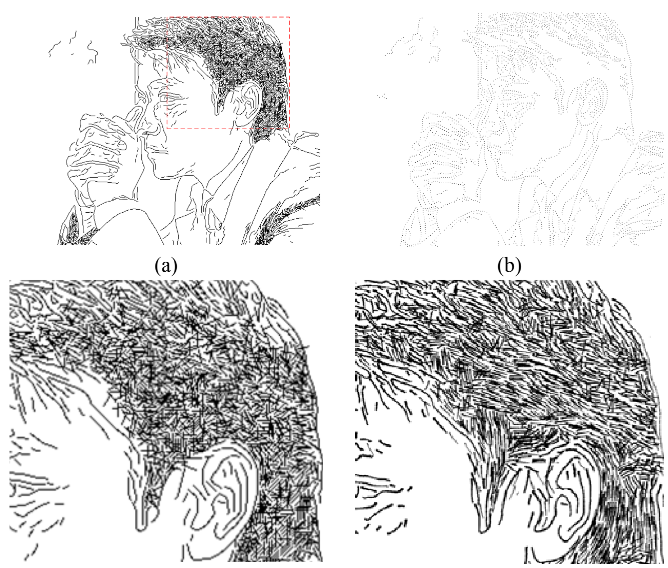
Our approach is to calculate the stroke orientation θ' at point p using only a few nearest points that have strong gradients. First, we select "strong points" (points with strong gradients) from edges of high structural importance (see section A) by sampling at regular intervals as shown in Fig 7 (b), then find out n (usually 2~4) points nearest to p, and compute θ' as follows:

$$\theta'(p) = \frac{\sum_{i=1}^{n} \theta(p_i) \cdot Grad(p_i)/d(p, p_i)}{\sum_{i=1}^{n} Grad(p_i)/d(p, p_i)} \tag{6}$$

$$with \qquad Grad(p_i) = |Grad_x(p_i)| + |Grad_y(p_i)|$$

where p_i is one of the n points nearest to p, $\theta(p_i)$ is the gradient direction at p_i, and $d(p, p_i)$ denotes the Euclidian distance between p and p_i. The reason why we do not select "strong points" by setting a gradient threshold is that this method will collect a series of points next to each other, resulting in information redundancy.

Since local strokes often share some "strong points", then they would appear more consistent with orientations gotten by our method. As shown in Fig 7 (d), the fair texture is more discernible than before.



(a) (b) (c) (d)

Fig. 7. Estimate stroke orientations: (a) image drawn following gradient directions; (b) "strong points"; (c) enlarged hair part of (a); (d) hair part drawn by our method, the texture looks more desirable now.

D. Stop Criterion

It's necessary to set a stop criterion for the iterative hatching process. In [4], hatching stops when the largest value in the importance image is below a termination threshold. However, it is difficult to find a threshold generally suitable for various input images, especially after our revising the definition of importance image.

Hence, a new criterion function is proposed to terminate the hatching process when the function reaches its minimum. The function is defined as follows:

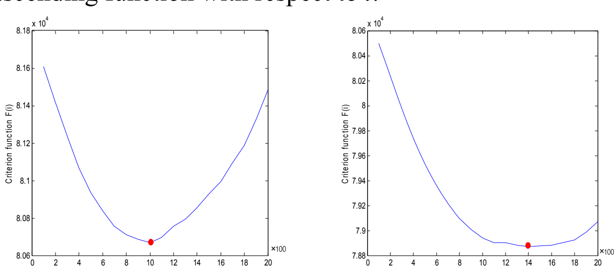
$$F(i) = \sum_{x \in R^i} (R^i(x))^2 + b \cdot \sum_{x \in I^i_{op}} (I^i_{op}(x))^2 \tag{7}$$

$$\text{with} \quad R^i(x) = \begin{cases} I_{oj}(x) - I^i_{op}(x) & I_{oj}(x) > I^i_{op}(x) \\ 0 & I_{oj}(x) \le I^i_{op}(x) \end{cases}$$

where X^i denotes the value of X after the *ith* stroke is drawn, R has the same size as I_{oj}, I_{oj} and I_{op} are the same as in equation (2), and b is a weight parameter.

The criterion function $F(i)$ is a sum of two components. The first one is a descending function with respect to i for two reasons: one is the number of points meeting the inequality $I_{oj}(x) > I^i_{op}(x)$ decreases as strokes increase; the other is the absolute values of $I_{oj}(x) - I^i_{op}(x)$ will reduce where a stroke is drawn nearby. In fact, this component behaves to indicate how much the darkness of the drawn image approximates the object image. The more strokes appear, the smaller this component gets.

In fact, iterative hatching will make the drawing darker and darker, eventually too dark in some regions. In order to control the final tone, the second component of $F(i)$ serves to restrain the total stroke number. As strokes always add darkness to the operated image I_{op}, this component is thus an ascending function with respect to i.



(a) (b)

Fig. 8. Criterion function and stop point when (a) b=1.0, and (b) b=0.9. Note that the stop point (in red color) comes earlier when b is larger, because the second component of $F(i)$ acts to restrict the total stroke number.

With an appropriate value of b, $F(i)$ will descend first and then ascend, with a minimum point depending on b as shown in Fig 8. No matter whether the input image tone is dark or light, the final drawing can always take on a desirable tone. This is because the stop point predicates a balance between the two components of $F(i)$, which keeps the drawing neither too dark nor too light. Therefore, the advantage of this stop criterion is that the drawing performance is robust to different input images, with a general value of b, e.g. b=1. Moreover, b's value can be adjusted slightly (between 0.8 and 1.2) to achieve another appearance, as long as you know that a larger value corresponds to a lighter final drawing, and vice versa. Examples will be shown in next section.

IV. Experiment Results

A. Experiment 1

This experiment exhibits the whole drawing process. Firstly, the original structural contents are represented by outlines of various stroke thicknesses, as shown is Fig 9 (a). Then, iterative hatching follows to convey the original tone and textures stroke by stroke, as shown in Fig 9 (b)-(f). As the stroke number i increases, the darkness at each area approaches its object tone at a similar rate, and local areas also take on desirable textures.

(a) i=0 (b) i=300

(c) i=700 (d) i=1026

(e) i=1283 (f) i=1502

Fig. 9. Whole drawing process: (a) outlines; (b)-(f) images in hatching, with stroke number i. (d) final drawing when b=1.05; (e) final drawing when b=1.0; (f) final drawing when b=0.9. Note the tone increases at a similar rate in each region, and the final drawings are different due to the values of b.

The experiment also demonstrates the effect of the parameter b in the criterion function. When b is 1.0, the drawing stops at i=1283 (Fig 9 (e)); when b is 0.9, the drawing stops at i=1502 (Fig 9 (f)); and when b is 1.05, the drawing stops at i=1026 (Fig 9 (d)). This reveals the relations between b's value and the final drawing effects, and we can easily adjust b slightly to get a desired style.

B. Experiment 2

This experiment tests the output robustness of our system to various input image tones. The value of b is set to be 1.0 in this experiment. Given two different input images, both the final drawings turn out to be desirable, as shown in Fig 10.

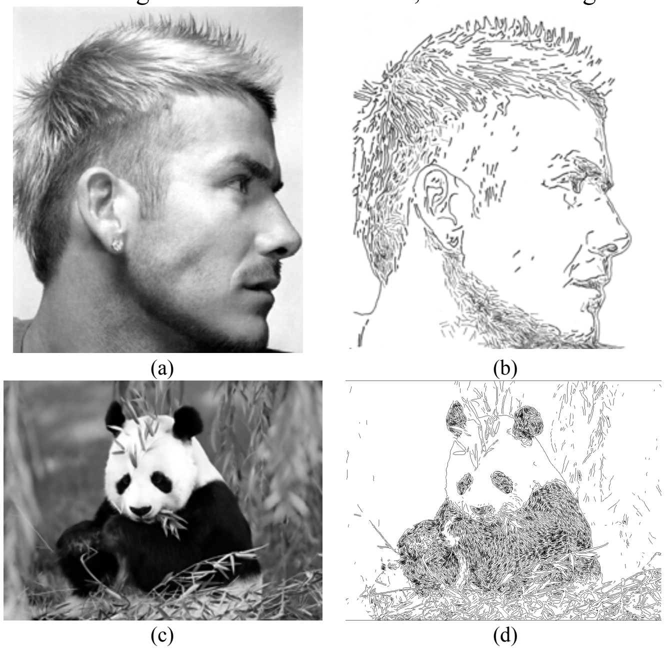


Fig. 10. Input images: (a) and (c); final drawings: (b) and (d).

V. CONCLUSION AND FUTURE WORK

We have developed a vision-based pen-and-ink drawing system based on a robotic manipulator. Given a grayscale image as input, the system first extracts the outlines from it and delineates them with different width strokes according to their structural importance. In the hatching process, stroke positions are determined based on visual feedback, and stroke orientations are calculated by local gradient interpolation. In this way, the drawing can approximate the original tone well, and can also take on desirable textures. We also proposed a criterion function to terminate the drawing process, and this makes the system robust to different input image tones. Experimental results prove that our system can produce relatively desirable pen-and-ink works. In future, we plan to utilize image segmentation to mine more information from the input image, and employ more kinds of strokes (like curves) to improve the drawing quality.

REFERENCES

[1] M. C. Sousa, J. W. Buchanan, "Computer generated graphite pencil rendering of 3d polygonal models," in *Computer Graphics Forum*, 1999, pp. 195–208.

[2] Hertzmann, A., Perlin, K. "Painterly rendering for video and interaction," In *Proceedings of International Symposium on Non Photorealistic Animation and Rendering*, 2000.

[3] Michael P.Salisbury, Sean E. Anderson, Ronen Barzel, and David H.Salesin. "Interactive pen-and-ink illustration," in *Proceedings of International conference on Computer graphics and interactive techniques*, July 1994, pp.101-108.

[4] Michael P. Salisbury, Michael T. Wong, John F. Hughes, David H. Salesin, "Orientable textures for image-based pen-and-ink illustration," in *Proceedings of International conference on Computer graphics and interactive techniques*, August 1997, pp.401-406.

[5] A. Srikaew, M. E. Cambron, S. Northrup, R. A. Peters II, M. Wilkes, and K. Kawamura, "Humanoid drawing robot," in *IASTED International Conference on Robotics and Manufacturing*, 1998.

[6] P. Monaghan, "An art professor uses artificial intelligence to create a computer that can draw and paint," The Chronicle of Higher Education, 1997.

[7] http://robix.com/drawbot.htm

[8] S. Kudoh, K. Ogawara, M. Ruchanurucks, and K. Ikeuchi, "Painting robot with multi-fingered hands and stereo vision," in *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp.127-132.

[9] Calinon, S., Epiney, J. and Billard, A., "A humanoid robot drawing human portraits," In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 161-166.

[10] Yam, Y, Lo, K.W, Kwok, K.W., "A robot drawing system: preliminary design and demonstration," in *Proceedings of the 2003 International Conference on Intelligent Technologies*, Chiang Mai, Thailand, Dec. 17-19, 2003, pp. 545-551.

[11] Ka Wai Kwok, Yeung Yam, Ka Wah Lo, "GA-based homography transformation for vision rectification in robot drawing System," in *Proceedings of IEEE Conference on Decision and Control and the European Control Conference*, Seville, Spain, 2005, pp. 2047-2052.

[12] A. Orzan, A. Bousseau, P. Barla, Joëlle Thollot, "Structure-preserving manipulation of photographs," in *International Symposium on Non-Photorealistic Animation and Rendering*, 2007, pp.103-110

[13] J. Hays, I. Essa, "Image and video based painterly animation," in *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, 2004, pp.113-120.

[14] Georges Winkenbach, David H. Salesin, "Computer-generated pen-and-ink illustration," in *Proceedings of International conference on Computer graphics and interactive techniques*, 1994, pp.91-100