2005 International Conference on Control and Automation (ICCA2005)
June 27-29, 2005, Budapest, Hungary

Vision System and Projective Rectification For A Robot Drawing Platform[†]

Ka Wai Kwok, Yeung Yam, and Ka Wah Lo
Department of Automation and Computer-Aided Engineering
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong
kwkwok@acae.cuhk.edu.hk, yyam@acae.cuhk.edu.hk, kwlo@vtc.edu.hk

Abstract—**A robot drawing platform supporting five degrees of freedom (x, y, and z translation, z-rotation, and pitch) of a brush-pen movement is under development. The platform is aimed at the acquisition, learning, and execution of human techniques in Chinese brush pen painting and calligraphy. Both replication of existing works and rendition of new styles are planned. This paper describes the addition of vision-based capabilities to the platform, together with a demonstration of how the vision information can be used to rectify incorrect decision on branch points in iterative steps.**

I. INTRODUCTION

ROBOT drawings have been reported in numerous works [1]-[3]. Based on various principles and intelligent computing techniques, these systems come in different shapes and sizes, operating on different mechanisms and producing a variety of painting styles and artworks. One of the more famous artistic robots, also among the first to be introduced in the field, is Harold Cohen's Aaron [4]. Some of Aaron's works are so impressive that they are actually displayed in Museums. All in all, robot drawing is now recognized as a platform for studying relationships between artistic creativity and artificial intelligence [5]. However, the drawing systems reported so far are mostly focused on Western art, and for free style rendition only.

A robot drawing platform to study Chinese painting and calligraphy is presently under development in our laboratory. The platform supports five degrees of freedom (x, y, and z translation, z-rotation, and pitch) with the precision and repeatability needed for fine execution of brush strokes in Chinese art making. The eventual goal of the platform lies in the acquisition, learning, and execution of human techniques in Chinese brush pen painting and calligraphy. It's noteworthy that painting and calligraphy go naturally together in Chinese artistry. The spirit of a good Chinese painting is always accentuated if accompanied by a good poem expressed in good calligraphy. Preliminary descriptions of the platform have been reported in [6] and [7].

This paper describes the addition of a camera system to the platform and the resulting vision-based capabilities. They include rectification of an angled image captured by the camera to a perspective as viewed from above. Rectification is conducted using homography matrix computed upon 9 selected correspondence points on the drawing plane. The rectified version of the executed drawing is then readily compared to the original image to yield corrective actions for enhancement in the next iterative execution. Specifically, this work shows the example of generating corrective actions on drawing branch points of a line sketch.

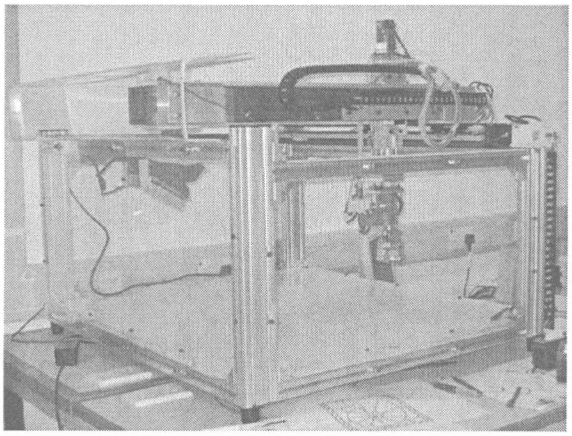


Figure 1. Hardware design of the Drawing Robot

II. ROBOT DRAWING PLATFORM

The drawing platform is shown in Figure 1. The platform consists of a x-y-z axis translational mechanism, and a robot gripper with a z-axis rotation and a pitching degree of freedom, making a total of 5-axis of degrees of freedom for the pen movement. Industrial grade components are utilized to achieve the

---

[†] This work was supported by the Hong Kong Research Grant Council under CUHK Direct Grant 2050317

high precision and repeatability needed for the full emulation of hand and wrist movement in the process of Chinese art making. The x- and y-translation are executed by two AC servomotors each with an angular torque of 0.51Nm and a travel length of 1m. Corresponding accuracy is ± 0.001mm. The z-axis AC motor has an angular torque of 0.08Nm, and a vertical load capability of 50N to support the specially designed robot gripper. The z-axis stroke length is 0.15m and the accuracy is ± 0.03mm. The overall dimension of the setup is 1.1m by 0.96m by 0.5m, with a drawing size of 0.8m by 0.7m. The five degrees of freedom of the brush pen are all independently commanded, doing away with the kinematics problems associated with many other robot-based drawing systems. The system supports two possible format to input an image; by an image file, or by direct hand drawing on a writing tablet.

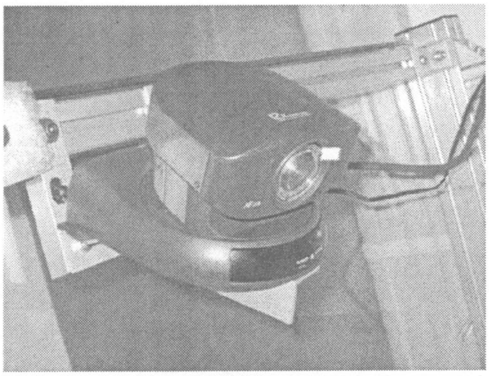


Figure 2. Camera looking down at drawing area for monitoring and visual-based capabilities

III. CAMERA SYSTEM

Figure 2 shows the recently installed camera system. The system uses a Sony EVI-D30/D31 Pan/Tilt/Zoom Color Video camera looking down at the drawing area at an angle of 30 degrees. The vision system serves to monitor the executed drawing on the drawing board and generate corrective actions upon comparing the executed drawing and the original image. In this work, we show the application of using the visual capability to make corrective action on drawing branch points. For simplicity, we use a line drawing for illustration. Line drawing means that the width of the strokes forming the lines is rather uniform and quite thin, hence executable with only the x- and y-axis degrees of freedom of the platform. This is in contrast to those with full brush strokes of thick and varying width, which would require pen motion control in other degrees of freedom as well. Also, corrective actions for the iterative execution here will be processed off-line. The full brush case with visual-based real-time on-line corrective capabilities will be our future goal.

Branch point decision in its own right is an important one in Chinese calligraphy, where the strokes forming any character should be executed in proper order. However, it is extremely difficult to accurately ascertain the correct order or sequence of stroke execution from the image of a Chinese character without knowing the particular character in concern. The present work is then also a starting attempt to tackle this issue for more general calligraphy execution.

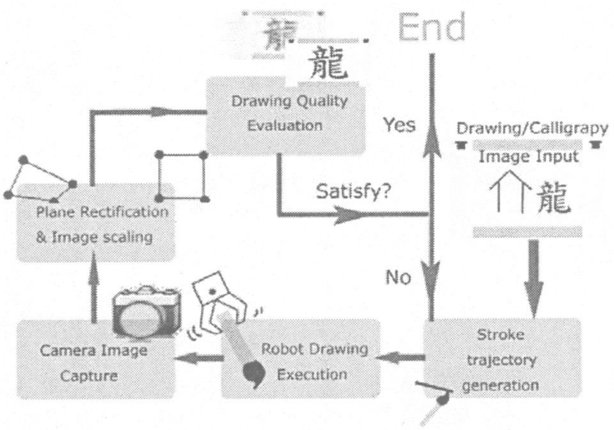


Figure 3. Schematic block diagram showing the process of visual-based corrective action

IV. VISUAL-BASED BRANCH POINT CORRECTION

Figure 3 depicts the schematic block diagram for the process of visual-based feedback and corrective action. Upon the acquisition of an image file, the following steps are to be conducted.

A. *Stroke Trajectory Generation*

This step concerns with the generation of line strokes to execute the drawing of the image. The corresponding JPEG or data file of the image is passed through a Matlab-based library containing user-designed algorithms to extract feature points and lines of the image. Figure 4 depicts the various algorithms in the process. They include

i) *Simple Global Thresholding:*

The original image is captured as a grayscale level data file is compared against a chosen threshold [8]. Any point on the image with pixel value higher than the threshold is regarded as a foreground (binary 1), otherwise it is regarded as a background (binary 0).

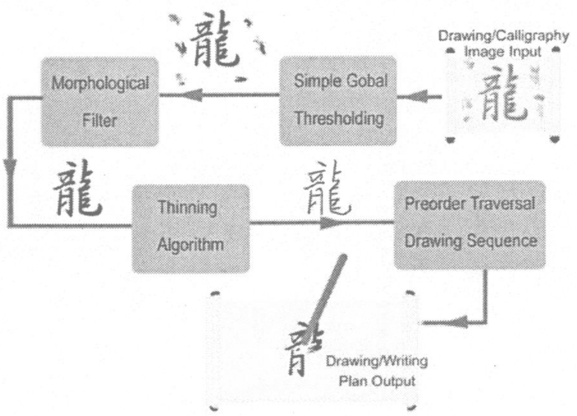


Figure 4. Algorithms for generating stroke trajectories

ii) *The Morphological Filter*

The morphological filter [9] deals with noises arising from surface reflectance and/or unclean drawing sources during image capturing. The filter performs two operations: binary closing to smooth over the gaps on objects, and binary opening to wipe out the spots in the background.

iii) *Thinning Algorithm*

The thinning algorithm [10] is an iterative object reduction technique resulting in a skeleton of the object of single pixel width. For the present case of line sketch, the resulting skeleton follows more or less the middle of the line segments. This gives the trajectories of the brush pen for execution.

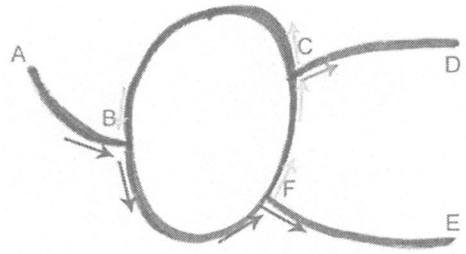


Figure 5. Preorder traversal sequence (Edges: A, D, E. Branches: B, F, C)

iv) *Preorder Traversal Point Sequence*

This part is concerned with the assignment of the drawing sequence. We adopted a simple rule to determine the start location consistent with most Chinese calligraphy: from Up to Down and Left to Right. After that, the Preorder Traversal Point sequence is adopted. Referring to Figure 5:

- The edge A is first chosen as the starting point (root) of the graph
- 1st line drawing segment: A-B-F-E. The branch list = {F,B} (the branch encountered last is placed at the head of the list)
- 2nd line drawing segment: F-C-B (start from the head element of the branch list). The branch list = {C} (Delete F and B from the list)
- 3rd line drawing segment: C-D. The branch list = {}
- If the branch list is empty, meaning that all connected skeleton in the graph have been visited, then start from the edge located at the left-top side of another one, if any.
- Drawing/writing plan terminates upon visiting all the graphs in image.

The preorder traversal point sequence is only somewhat effective even for calligraphy of formal Chinese characters, let alone the more free style of Chinese calligraphy and painting. The proper order of execution would be extremely difficult to pinpoint by any rule. The present work aims at developing visual-based corrective capability to tackle this problem.

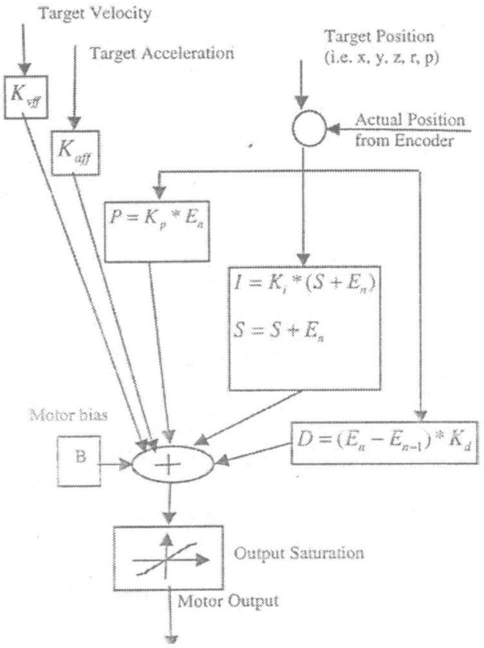


Figure 6. PID + K_{vff}+ K_{aff} controller

B. *Robot Drawing Execution*

The line trajectories and their order of execution as generated from the Stroke Trajectory Generation block are expressed in (x,y) coordinate sequences and forwarded to the motion controller of the drawing platform for execution. Execution is conducted via a PID type controller. Figure 6 shows the structure of the controller. The motor output value U_n for the axes of x or y is given by:

$$E_n = (P_{target})_n - (P_{actual})_n \qquad (1)$$

$$U_n = E_n K_n + (E_n - E_{n-1})K_d + \left(\sum_n E_n\right)\frac{K_t}{256} \qquad (2)$$
$$+ V_{target}K_{vff} + ACC_{target}K_{aff} + B,$$

where E_n, P_{target}, and P_{actual} are the position error, the target position, and the actual position, respectively, at sample time *n*; V_{target} is the current desired velocity in unit *counts/ST*, with *ST* being the sampling period; ACC_{target} is the current desired acceleration in unit *counts/ST*^{2}, *B* is the motor bias value, and $\left(\sum_n E_n\right)$ is the accumulated error at sample time *n*.

The filter gains of Figure 6 are programmable to provide good control accuracy and stability over a large range of operation. For the present execution, $K_P = 5$, $K_I = 2$, $K_D = 60$, and the velocity and acceleration gains K_{vff} and K_{aff} are zero. The filtered output saturates at $\pm 2^{15}$. The corresponding motor output is ±10V.

C. *Rectification and Image Scaling*

The camera captures the image of the executed drawing upon its completion. Figure 7 shows the captured image in the present study. The captured image is angled as the camera is mounted at 30 degrees looking downwards. To compare the executed drawing with the original image, the captured image needs to be transformed into a full plane view as if observed from above. The well known Direct Linear Transformation (DLT) [11] algorithm is adopted for this purpose. The algorithm determines a 3x3 homography matrix H upon given four or more given 2D to 2D point correspondences, $X_i \leftrightarrow X_i'$. The relationship can be expressed as $X_i' \times HX_i = 0$. In this case, the equation involves non-homogeneous vectors due to the fact that all correspondences are in the image coordinates, and hence the 3-vectors X_i' and HX_i are equal to each other. Specifically, upon given *n* correspondence pairs, $X_i = (x_i, y_i, 1)^T$ and $X_i' = (x_i', y_i', 1)$ for i= 1 to *n*, the cross product equation is:

$$X_i' \times HX_i = \begin{pmatrix} y_i' h^{3T} X_i - h^{2T} X_i \\ h^{1T} X_i - x_i' h^{3T} X_i \\ x_i' h^{2T} X_i - y_i' h^{1T} X_i \end{pmatrix} = 0 \qquad (3)$$

where $H = \begin{pmatrix} h^{1T} \\ h^{2T} \\ h^{3T} \end{pmatrix}$.

With $h^{jT} x_i = x_i^T h^j$, (3) can be written as:

$$\begin{bmatrix} 0^T & -X_i^T & y_i' X_i^T \\ X_i^T & 0^T & -x_i' X_i^T \\ -y_i' X_i^T & x_i' X_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \qquad (4)$$

The third equation in (4) can be omitted as it is linearly dependent. Each correspondence hence contributes to two linearly independent equations as:

$$\begin{bmatrix} 0^T & -X_i^T & y_i' X_i^T \\ X_i^T & 0^T & -x_i' X_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \text{ , or } A_i h = 0, \qquad (5)$$

where A_i is 2x9 matrix and *h* is 9x1 vector. Putting the *n* 2x9 matrices A_i into a single 2*n*x9 matrix *A* and generating its singular value decomposition (SVD), one obtains *h* as the unit singular vector to the smallest singular value, and hence *H* can be determined. Figure 8 shows the *n*=9 point correspondence picked for generating the homography matrix.

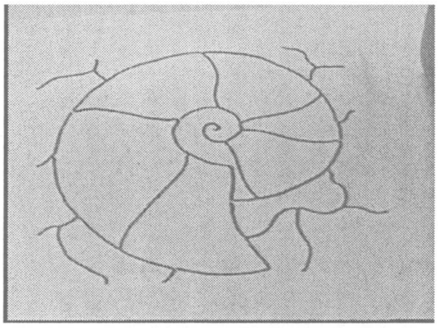


Figure 7. Captured image of the executed drawing

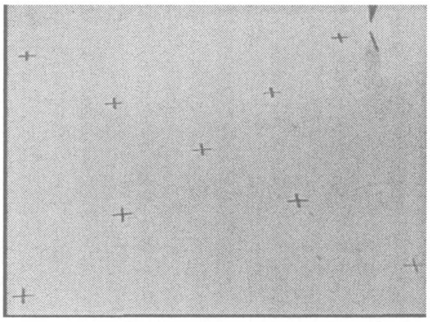


Figure 8. Point correspondences for generating homography matrix

The homography matrix allows the captured image to be rectified in a full plane view and same scale as the original input image so that the rectified image has the same pixel resolution as the original before the two can be compared. This homography transformation also increases the flexibility that the installed camera needs not to look down strictly vertical to the drawing plane. Figure 9 shows the overlapping of the rectified image of the executed drawing and the original image in the same pixel resolution.

D. Drawing Quality Evaluation/Corrective action

This step is aimed at extracting information [12] on branch point execution errors through analyzing the overlapped images of Figure 9. Focusing on the branch point regions, figure 10a) shows a circular disk centered in the stroke skeleton branch point of the original image, and figure 10b) shows the same disk formed with same coordinates in the rectified executed image. It can be seen that the stroke connectivity inside the disk between the executed image and the original image does not agree with each other. The branch point decision hence needs to be corrected. Figure 11 shows the five branch points on the original image (encircled) that have been executed improperly. This thus provides the needed information for correct branch point decision in a second robot execution of the same drawing. Figure 12 depicts the output of the second execution. All branch points are now properly executed. More iteration may be needed if the branch points are of more complicated nature.

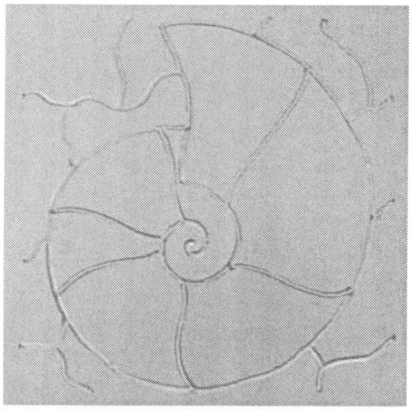
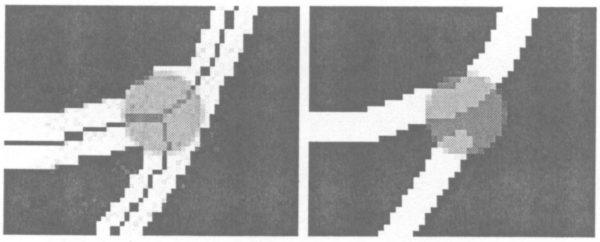


Figure 9. Overlapping of the rectified robot drawing and original image



a) b)

Figure 10a). A maximized circular disk is centered in the skeleton branch pixel inside the stroke. b). The same size disk with same coordinates is formed to detect the connectivity in the executed drawing

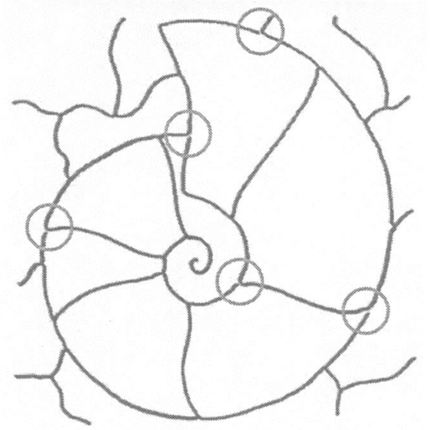


Figure 11. Improperly executed branch point locations

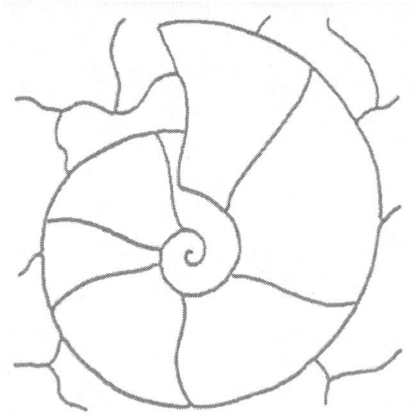


Figure 12. Second time executed image with corrective

V. CONCLUSIONS

A robot drawing platform aimed at studying Chinese painting and calligraphy is being developed in our laboratory. The platform supports a total of five degrees of freedom for brush-pen motion to emulate the needed hand and wrist movements in Chinese artistry. This paper describes the addition of a camera system to the platform and the development of visual-based capabilities for corrective action on improving the executed drawing by comparing with the original. Visual information is intuitive and natural as a child would learn to write and draw first by sight. The present work demonstrates that the capabilities developed can be successful in pinpointing the branch points of a line sketch that have been incorrectly executed, and then rectifying them in the next execution of the drawing. Branch point decision is important as it is related to the order of stroke execution, which plays a vital role in Chinese calligraphy. For simplicity, the present work utilizes only line drawings to illustrate the results. Also, processing for corrective actions is conducted off-line.

The full stroke case with on-line real-time visual-capabilities will be our future goal.

REFERENCES

[1] *Morgan, H., Harold Cohen's Aaron, The Robot as an Artist.*, http://www.scinetphotos.com/aaron.html

[2] Srikaew A., Cambron, M.E., Northrup S., Peters II, R.A., Wilkes, D.M., and Kawamura K., *Humanoid Drawing Robot*, IASTED International Conference on Robotics and Manufacturing, Banff, Canada, July 26-31, 1998.

[3] Morris, D., Phelps, K., and Joshi, N., http://techhouse.brown.edu/~neel/drawing_telerobot

[4] McCorduck, P., *Aaron's Code: Meta-art, Artificial Intelligence and the work of Harold Cohen*, W.H. Freeman & Co., New York, New York Copyright 1991.

[5] Pagliarini, L., and Lund, H.H., *Art, Robots, and Evolution as a Tool for Creativity*, Creative Evolutionary Systems, P. Bentley and D. Corne (eds), Morgan Kaufman, 2001.

[6] Yam, Y, Lo, K.W, Kwok, K.W., "A Robot Drawing System: Preliminary Design and Demonstration," Proceedings of the 2003 International Conference on Intelligent Technologies, Chiang Mai, Thailand, Dec. 17-19, 2003, pp. 545 -551.

[7] K.W. Lo, K.W. Kwok, and Y. Yam, "Automated Replication of Line Drawings By a Robot Drawing Platform", Proceedings of the 8*th World Multi-Conference on Systemics, Cybernetics and Informatics*}, Orlando, Florida, USA, July 18-21, 2004.

[8] Xiangyun Ye; Cheriet, M.; Suen, C.Y., "Stroke-model-based character extraction from gray-level document images", *Image Processing*, IEEE Transactions on, Volume: 10, Issue: 8 ,Aug. 2001,Pages:1152-1161

[9] Sedaaghi, M.H., "Morphological operators", *Electronics Letters*, Volume: 38 , Issue: 22 ,24 Oct 2002, Pages:1333-1335

[10] Mozer, M.C. and Smolensky, P., *Skeletonization: A technique for trimming the fat from a network via relevance assessment*, in Advances in Neural Information Processing Systems. Tourezky D.S., Ed. Vol. 1, Denver, CO. Morgan Kaufmann, pp. 107--115.

[11] Multiple View Geometry in Computer Vision Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004.

[12] X. Hilaire and K. Tombre, "Improving the Accuracy of Skeleton-Based Vectorization," in *Graphics Recognition - Algorithms and Applications*, D. Blostein and Y.-B. Kwon, editors, pp. 273-288. Springer Verlag, 2002.