Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

GA-based Homography Transformation for Vision Rectification in Robot Drawing System[1]

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 motion has been under development in our laboratory. The platform is aimed at studying Chinese painting and calligraphy. Both replication of existing works and rendition of new are planned. This paper describes the addition of vision-based capabilities in the platform. They include projective rectification of the executed work by an overlooking camera, and the corrective drawing actions for iterative improved drawing. For enhanced performance, rectification here is implemented automatically via GA-based Homography transformation. A demonstration of how the vision information can be used to evaluate and improve the drawing is presented.**

I. Introduction

Robot drawing is not new, and has been reported in numerous works [1]-[6]. However, meet of the works reported are focused on the free style drawing of Western art. In contrast, we have designed and constructed a robot drawing platform aimed at studying Chinese painting and calligraphy in our laboratory. The platform supports five degrees of freedom of a pen motion for the full emulation of hand and wrist movement in the process of Chinese art making. The eventual goal will be on the acquisition, learning, and execution of human techniques in Chinese brush pen painting and calligraphy. Preliminary capabilities of the platform are reported in [7] and [8].

This paper describes the addition of a camera system and the incorporation of vision-based capabilities to the platform. They include rectification of an angled image captured by the camera to a full plane view. Rectification is conducted using homography matrix generated by Genetic Algorithms (GA) upon initialization by 4 manually selected correspondence points. The incorporation of GA relaxes the stringent requirement on the accuracy of the manual selected points, and yield high quality overlapping of the executed drawing with the original. The comparison readily allows corrective actions to be performed on the next execution for improved drawing quality. A demonstration of this improvement is included.

II. Robot Drawing Platform

Figure 1 shows the drawing platform constructed in our laboratory. 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 degrees of freedom for the pen movement. The platform utilized industrial grade components to achieve the high precision and repeatability needed for the fine execution of brush strokes. The x and y-translation are executed by two AC servomotors each with an angular torque of 0.51 Nm and a length of travel of 1m. The corresponding accuracy is $\pm$ 0.001 mm. The z-axis AC motor has an angular torque of 0.08 Nm, and a vertical support load of 50 N to provide the needed support for the specially designed robot gripper. The z-axis stroke length is 0.15 m and the accuracy is $\pm$ 0.03 mm. The overall dimension of the setup is 1.1 m by 0.96 m by 0.5 m, with a drawing size of up to 0.8 m by 0.7 m. The five degrees of freedom of the brush pen are independently commanded, doing away with the kinematics problems associated with many other robot-based drawing systems.

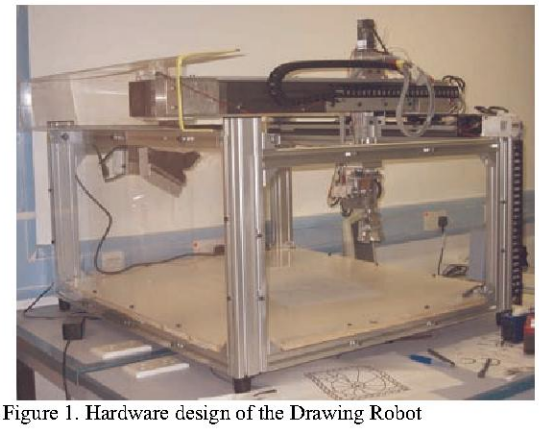


Figure 1. Hardware design of the Drawing Robot

The present work utilizes line calligraphy for demonstration of vision capabilities. Line calligraphy means that the width of strokes forming the words will be quite uniform and rather thin. This contrasts to those with full brush strokes of thick and varying width, which would require pen motion control other than (x, y) degrees of freedom. Hence, the line trajectories and their order of execution are only expressed in the form of (x, y) coordinate sequences. They are forwarded to the motion controller of the drawing platform for execution. The motion controller performs the execution through the implementation of a PID type controller. Detail description of this part can be found in [7] and [8].

Also, corrective actions here will be processed off-line for improved drawing. Eventually, with the full brush capabilities, we hope to be able to use visual feedback to

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

correct the thickness and line path of the brush strokes in real time and online in the future.

III. Line Trajectory Generation

The system accommodates two possible ways to input an image. One is through an image file, and the other by direct hand drawing on a writing tablet. The corresponding JPEG image or data file then passes through a Matlab-based library containing user-designed algorithms to analyze the input data and to extract feature points and lines of the image. The process involves raster to vector conversion which include thresholding to grayscale image [9], image noise reduction filter [10] and thinning algorithm [11]. The conversion results in one-pixel-wide skeletons of the image foreground. For the present case of line calligraphy, the resulting skeleton would follow more or less the middle of the line segments which constitute the trajectories in line calligraphy

Assignment of the drawing sequence is then generated using Preorder Traversal Point Sequence (PTPS). As baseline, a simple rule is adopted to determine the start location consistent with most Chinese calligraphy: from Up to Down and Left to Right. After that, the PTPS as described below is used. Figure 2 gives an example of the application:

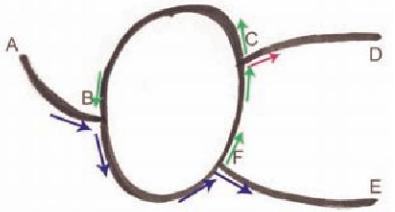


Figure 2. An example of Preorder traversal point sequence (Edges: A, D, E. Branches: B, F, C)

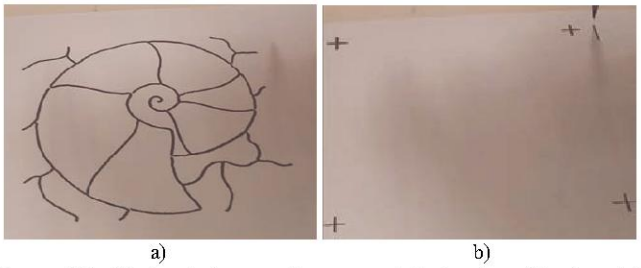
- The edge A is first chosen as the starting point (root) of the graph
- 1st line drawing segment: A-B-F-E. The branch list = {F,B} (the last passing branch placed at the head of the list)
- 2nd line drawing segment: F-C-B (start the point from the head element of the branch list). The branch list = {C} (Delete F and B from the list after all possible ways of those branch are visited)
- 3rd line drawing segment: C-D. The branch list = {}
- If the branch list is empty, that means all connected skeleton in the graph are visit, then try to start from the edge which locates at the left-top side of another graph.
- Until all graphs in the same picture are visited, the drawing/writing plan would be finished.

The PTPS is not all consistent to human drawing and writing sequence, especially in free style Chinese calligraphy and painting. Proper order of execution, however, would be extremely difficult to set by any rule. In some cases, the improper drawing and writing sequence will highly affect the quality of executed work. The vision-based capabilities to be incorporated later are aimed at improving this problem.

IV. Camera System for Image Capture and Rectification

Figure 3 shows the installed camera system looking down at the drawing area at an angle of roughly 30 degrees. The system uses a Sony EVI-D30/D31 Pan/Tilt/Zoom Color Video camera. The camera system serves to monitor the executed drawing on the drawing board and generate corrective actions upon comparing the executed drawing with the original image. Specifically, we show in this work the example of using the visual capability to make corrective action on drawing the branch points. Branch point decision is an important one in Chinese calligraphy, where the strokes forming the character must be executed in proper order. The correct ordering is difficult to generate just by the image of the Chinese character, especially where the calligraphy is more of free style. The present work is an attempt to tackle this issue via visual-based techniques.

Figure 3. Camera looking down at drawing area for monitoring and visual-based capabilities



a) b)

Figure 4a). Captured image of an executed drawing. b). 4 points corresponding to the 4 corner of the original image for generating homography matrix

The image of the executed work is captured by the camera must first undergo a projective rectification [12]. The aim of the projective rectification is to remove the projective distortion in the perspective image of the drawing plane to the extent that similarity properties such as angles or ratios of lengths could be visualized at the original plane. In theory, projective distortion can be completely removed by selecting 4 reference points (totally 8 degrees of freedom) on the drawing plane, and compute the homography transformation map them to the 4 corners of the original images. Figure 4 shows the captured and angled image in our case study and the 4 selected points for homography transformation.

V. HOMOGRAPHY TRANSFORMATION

In other 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 atop. The well known Direct Linear Transformation (DLT) [13] algorithm is adopted for this purpose. The algorithm determines a 3x3 homography matrix upon given at least four 2D to 2D point correspondences, $\mathrm{X}_i \leftrightarrow \mathrm{X}_i'$. The equation can be expressed as $\mathrm{X}_i' \times H\mathrm{X}_i = 0$. In the present case, the equation involves non-homogeneous vectors as all correspondences are in the image coordinates, and hence, the 3-vectors X_i' and $H\mathrm{X}_i$ are equal. Specifically, given n correspondence pairs, $\mathrm{X}_i = (x_i, y_i, 1)^T$ and $\mathrm{X}_i' = (x_i', y_i', 1)^T$ for i= 1 to n, the cross product equation is:

$$\mathrm{X}_i' \times H\mathrm{X}_i = \begin{pmatrix} y_i' h^{3T}\mathrm{X}_i - h^{2T}\mathrm{X}_i \\ h^{1T}\mathrm{X}_i - x_i' h^{3T}\mathrm{X}_i \\ x_i' h^{2T}\mathrm{X}_i - y_i' h^{1T}\mathrm{X}_i \end{pmatrix} = 0 \qquad (1),$$

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

With $h^{jT} x_i = x_i^T h^j$, (1) can be written as:

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

The third equation in (2) is omitted as it is linearly dependent. Each correspondence hence contributes to two linearly independent equations as:

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

Where A_i is the 2×9 matrix and h is 9×1 vector. Then, putting the n 2×9 matrices A_i into a single $2n\times 4$ matrix A and generating the singular value decomposition (SVD) of A, the unit singular vector corresponding to the smallest singular value then yields h, and hence H. Figure 4b) shows the n=4 correspondences picked for generating the rectifying matrix.

The homography matrix allows the captured image to be rectified in a full plane view with proper scaling so that the rectified image has the same pixel resolution as the original input image before the two can be compared. It increases the flexibility that the installed camera needs not to look down strictly vertical to the drawing plane. However, it highly demands the accuracy of the homography transformation for two images overlapping exactly.

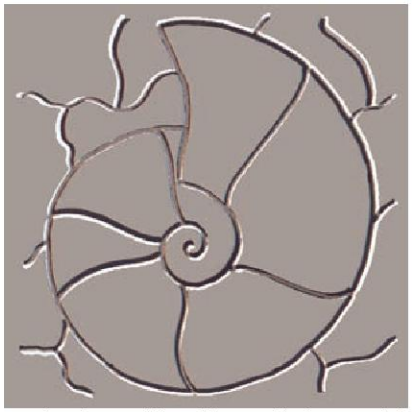


Figure 5. Poor overlapping qualities of the rectified executed image (in white) and the original image (in black)

VI. GENETIC ALGORITHMS FOR HOMOGRAPHY CORRECTION

Figure 5 shows the overlap between the rectified image of the executed drawing in figure 4a) and the original image via the homography transformation as computed by the selected points. As can be observed, the comparison is far from desirable. This problem is due to the fact that the homography transformation is highly sensitive to the exact correspondence of the selected points in figure 4b). Presumably, one can pick more correspondence points in a quest to reduce the overlapping errors by SVD technique, e.g., n=9 reported in [14]. However, it is all the more exhausting to pick so many points carefully in the midge. As a result, manual selection is extremely difficult to produce the overlapping performance to guarantee useful corrective drawing actions.

In what follows, Genetic Algorithms (GA) is introduced to arrive homography transformation at a highly performance upon the initialization of 4 roughly selected correspondences in the image. Formulation of the GA process is as given.

A. GA Representation

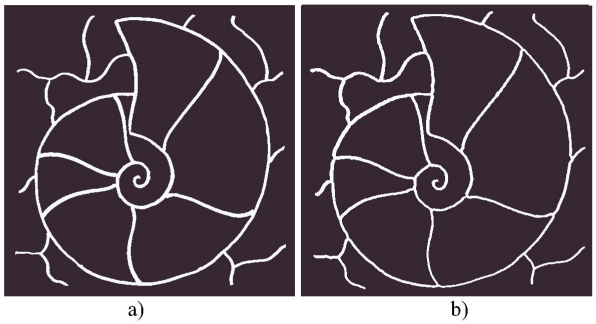
In this work, we pick parametric genes ($\Delta P_{i=1,\ldots,4}$) as the fine tuning in the displacement of the manually selected correspondence points. The chromosome comprising of the parametric genes is as shown in figure 6.

ΔP_{1x}	ΔP_{1y}	ΔP_{2x}	ΔP_{2y}	ΔP_{3x}	ΔP_{3y}	ΔP_{4x}	ΔP_{4y}

Figure 6. Chromosome structure of the correspondence displacement

B. Objective and Fitness Evaluation

The objective function serves to provide a measure of how individuals have performed in the problem domain [15]. In the case of a minimization problem, the most fitted individuals will have the lowest numerical value of the associated objective function. This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a GA [16]. Referring to figure 7, which shows the original image and rectified image converted to black and white, with the foreground pixels as 1s and the background pixels as 0s.



a) b)

Figure 7a). The original image in black and white. b). The rectified executed image in Black and White.

The objective function is then defined as:

$$\textbf{Objective: } \text{minimize } o(H) = \sum_{j}^{m}\sum_{i}^{n} p_{ij}^{r} \cdot \overline{p_{ij}^{o}}, \qquad (4),$$

where *i* is the row number, *j* is the column number. p_{ij}^{o} is the pixel value of $n \times m$ original image, p_{ij}^{r} pixel value in $n \times m$ rectified executed image, $\overline{p_{ij}}$ denotes the pixel value conjugate, $(0 \rightarrow 1$ or $1 \rightarrow 0)$. $o(H)$ is the objective cost function for the homography *H,* as where the 3×3 matrix *H* is calculated based on the displaced correspondences as according to the chromosome given.

The fitness function serves to transform the objective function value into a measure of relative fitness. This mapping is always necessary when the objective function is to be minimized as the lower objective cost correspond to fitter individual can expect to produce in the next generation. In this paper, a non-linear fitness assignment is useful in preventing premature convergence. Individuals are assigned a fitness value according to their rank in the population rather than their raw performance. The fitness of a individual in the population is calculated as,

$$\textbf{Fitness: } F(x_i) = 2 - PRS + 2(PRS - 1)\frac{x_i - 1}{N_{ind} - 1} \qquad (5),$$

where $x_i = ranking(o(H_i))$ is the position in the ordered population of individual *i*, *i*=1,…, N_{ind}, and N_{ind} is a population size in each generation. As suggested in [17], such the variable, *PRS*, is typically chosen in the interval [1.1, 2.0].

This fitness assignment ensures that each individual has a probability of reproducing according to its relative fitness which is related to the overlapping quality under its transformation H_i.

C. *Evolutionary Computing*

The following genetic parameters and operations are adopted aiming at the projective rectification problem at hand.

1) Displacement resolution: d_r pixel unit.
2) Displacement range: $d_{\max}$ and $d_{\min}$ pixel unit.
3) Chromosome length: 4 correspondences points; 8 degree of freedoms. Chromosome length in binary coded, $l_{bit} = 8\left\lceil \log_2\left(\frac{d_{\max} - d_{\min}}{d_r} + 1\right)\right\rceil$
4) Crossover method:
   One-point crossover with probability of crossover p_c
5) Mutation method:
   New individuals take the current population and mutate each element with probability of mutation $p_m = 0.7 / l_{bit}$.
6) Selection method: Stochastic Universal Sampling with population selection p_s=90%.
7) Replacement: Fitness-base reinsertion to current population.
8) Population size N_{ind}: between 40-100, based on the length of the chromosome.
9) Number of generation past: N_{gen}
10) Initial population: randomly generated within the range of the displacement range according to the displacement resolution.

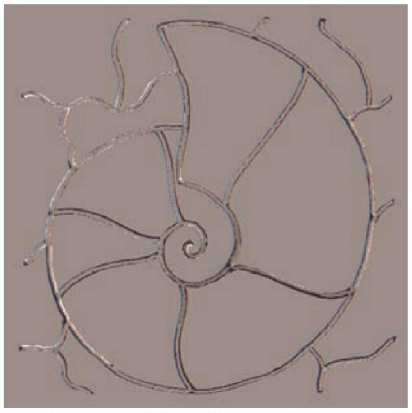


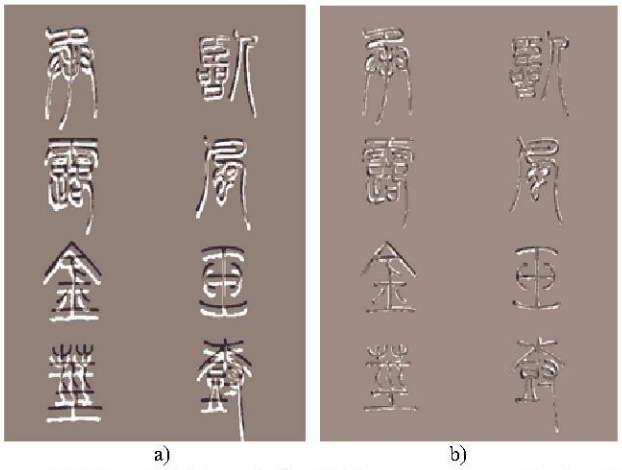
Figure 8. Exact overlapping qualities of the rectified executed image in white and the original image in black

D. *Results of GA homography transformation*

The GA homography estimation result is performed on a PC with Pentium 4 (2.66GHz) CPU, 1GB RAM, using Windows XP. The computation time for finding the optimal homography transformation is roughly 45-60mins. This computation time (average by multiple runs) is for reference only since the program is running in the debug mode of MatLab environment. The actual speed should be much faster. Besides, as further corrective action due to be generated off-line, this computational cost is not too important here. Moreover, calibration is needed to perform only once except for further adjustment to camera.

The overlapping result as produced by the evolutionary computing is shown in figure 8, which indicates marked improvement over that of figure 5. Here, the pixel size is 580×580 image. The parameters are set to be: d_r=0.5 pixel

units, $d_{\max}$ =20 pixel units, $d_{\min}$ =-20 pixel units, l_{bit} =56, p_c =0.7, p_m =0.0125, N_{ind} =60 and N_{gen} =80. The objective cost $o(H)$ of the executed image upon rectification by the manually selected correspondance points, depicted as figure 5, is =12091. After GA homography estimation, the objective cost decreases to $o(\tilde{H})$ =854 in figure 8. The corresponding offset displacement is ΔP =[19, 4.5, -13, 9.5, -4, 0, -3, -5.5] pixel units.



a) b)

Figure 9a). The rectified image before GA homography correction in white. b). The one after correction shows the executed stroke is thicker than the original image one.

a) b)

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

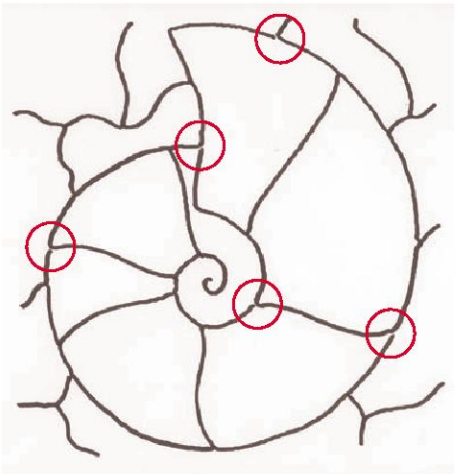


Figure 11. Wrong drawing sequence causes improperly executed branch points which are encircled.

Figure 9 shows another result in the overlapping 580×400 image. The parameters are d_r =0.2 pixel units, $d_{\max}$ =15 pixel units, $d_{\min}$ =-15 pixel units, l_{bit} =64, p_c =0.7, p_m =0.0109, N_{ind} =70 and N_{gen} =60. Figure 9a) shows the case of homography transformation based on manually selected points. The corresponding objective cost is $o(H)$ =10546. Figure 9b) shows the case of GA homography transformation, and the objective cost is $o(\tilde{H})$ =4176. In this case, the cost is not reduced as much as in the previous case because the executed stroke is thicker than the original image. The offset displacement ΔP =[1.2, -5, -8.6, 4, -1, -8.8, -0.2, -11.2] pixel units.

VII. Application To Robot Drawing

Figure 8 comprising of the overlapping results using GA homography transformation is now applied to generate correction actions in the execution of branch points. This is possible because brush tip deformation during the drawing process usually yields rather significant offset in the brush lines should improper drawing sequence be executed.

Focusing on the branch point regions, figure 10a) shows a circular disk centered at the stroke skeleton branch point of the original image, and figure 10b) shows the same disk formed with same coordinates and size 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 five encircled branch points on the executed drawing image on which drawing sequences have been identified to be improperly executed. This thus provides the needed information for correct branch point drawing in the second robot execution. The result is depicted in Figure 12. All branch points are now properly executed. More iteration may be conducted if the branch points are of more complicated nature than that of the present case.

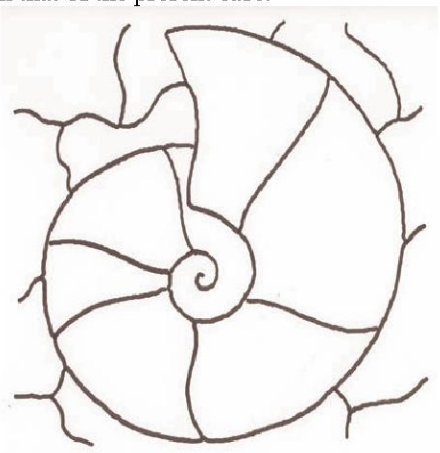
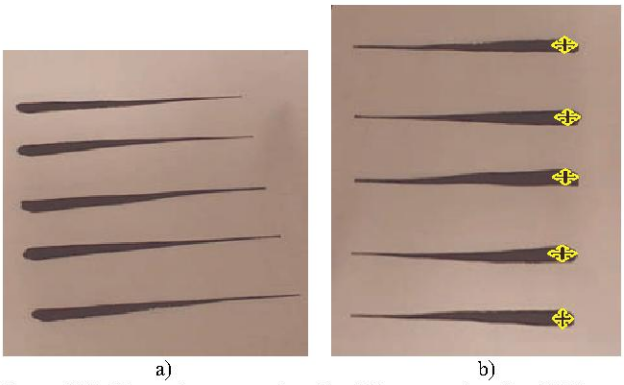


Figure 12. Second time executed image with corrective drawing sequence.

Besides the branch point decision, other drawing and painting performance may also be measured using the vision

information. Figure 13a) depicts the stroke lines executed under difference z-axis (vertical to the drawing plane) values as captured by the camera at an angle. Analysis of its projective rectified image in figure 13b) actually allows the calibration of executed stroke width as function of the z-axis command. The calibration curve is shown in figure 14.



a) b)

Figure 13a). The strokes were painted in different z-axis value. b) The projective rectified image for stroke thickness measurement.

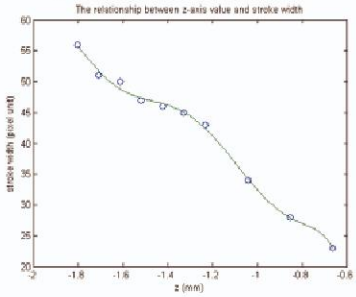


Figure 14. The graph of stroke thickness against the z-axis value from vision information obtained.

Application of the overlapping information to generate corrective action in branch point decision of a line sketch is attempted. This information is very useful to analyze and execute the calligraphy painting, especially for one kind of ancient Chinese font called Seal Character "篆書".

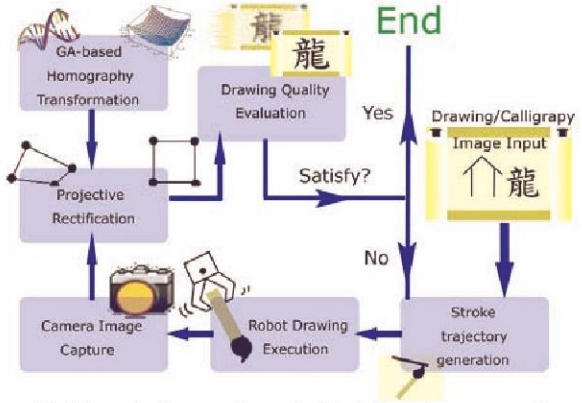


Figure 15. Schematic diagram shows the ideal about the process of visual-based correction action for iterative drawing.

VIII. Conclusions

A robot drawing platform aimed at studying Chinese painting and calligraphy has been developed in our laboratory. This paper describes the addition of a camera system to the platform and the ability to overlap the executed drawing with the original. For close comparison that is desired, we specifically develop a GA-based homography transformation which automatically generates a high performance overlapping upon the coarse manual selection of 4 correspondence points. The result shows that the process developed is successful in pinpointing the branch points that were incorrectly executed, which were then rectified in the next execution of the drawing. It is envisioned that visual-based capabilities would be expanded in the future to conduct iterative drawing quality enhancement in thickness and trajectory control of albeit simple painting and calligraphy with full brush strokes, as depicted in figure 15.

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] P. Monaghan, "An art professor uses artificial intelligence to create a computer that draws and paints," *The Chronicle of Higher Education*, pp. 27-28, May 1997.

[7] 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.

[8] 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.

[9] 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

[10] Peters, R.A., II., "A new algorithm for image noise reduction using mathematical morphology", *Image Processing*, IEEE Transactions on , Volume: 4 , Issue: 5 , May 1995, Pages:554 - 568

[11] Zhang, Y.Y.; Wang, P.S.P., "A parallel thinning algorithm with two-subiteration that generates one-pixel-wide skeletons", *Pattern Recognition, 1996., Proceedings of the 13th International Conference on* , Volume: 4 , 25-29 Aug. 1996, Pages:457 - 461 vol.4

[12] Liebowitz, D. and Zisserman, A., "Metric Rectification for Perspective Images of Planes", In Proceedings of the *Conference on Computer Vision and Pattern Recognition*, pages 482-488, June, 1998.

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

[14] K.W. Kwok, Y. Yam and K.W. Lo, "Vision System and Projective Rectification For A Robot Drawing Platform", Proceeding of *2005 International Conference on Control and Automation (ICCA'05)*, Budapest, Hungary, June 27-29, 2005, pp.691-695.

[15] Chin-Teng Lin , C. S. George Lee, *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1996

[16] A. J. Chipperfield, P. J. Fleming, H. Pohlheim and C. M. Fonseca, *"Genetic Algorithm Toolbox User's Guide"*, ACSE Research Report No. 512, University of Sheffield, 1994.

[17] J. E. Baker, "Adaptive Selection Methods for Genetic Algorithm", *Proc. ICGA 1*, pp. 101-111, 1985.