A Study on the effects of noise level, cleaning method, and vectorization software on the quality of vector data

Hasan S. M. Al-Khaffaf[1], Abdullah Zawawi Talib[1], Rosalina Abdul Salam[1]

[1]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM Penang, Malaysia
{hasan, azht, rosalina}@cs.usm.my

Abstract. In this paper we study different factors that affect vector quality. Noise level, cleaning method, and vectorization software are three factors that may influence the resulting vector data. Real scanned images from GREC'03 contest are used in the experiment. Three different levels of salt-and-pepper noise (5%, 10%, and 15%) are used. Noisy images are cleaned by six cleaning algorithms and then three different commercial raster to vector software are used to vectorize the cleaned images. Vector Recovery Index (VRI) is the performance evaluation criteria used in this study to judge the quality of the resulting vectors compared to their ground truth data. Statistical analysis on the VRI values shows that vectorization software have the biggest influence on the quality of the resulting vectors.

Keywords: salt-and-pepper, raster-to-vector, performance evaluation, engineering drawings

1. Background

Raster to vector conversion is a hot topic in the field of graphics recognition [1]. Many factors affect the quality of detected vectors which include: all kind of noise, cleaning method used, vectorization algorithm used. The previous two contests on graphics recognition [2, 3] accompanying GREC'03 and GREC'05 give some insight to the effect of noise on the resulting vector data, but they did not include extensive test on different noise levels or study the effect of different cleaning methods on the quality of the vectors. It also did not reveal the major factor that affects vector quality. Their findings could answer only limited questions regarding the interaction between different factors and treatments.

2. Image Data

The images from GREC'03 are used since ground truth files are readily available for the performance evaluation task. Another reason is that the graphical elements in

those images are relatively thin. Noise will affect these thin elements more than other thick elements which make it more challenging for the cleaning method to retain it and the vectorization software to recognize it correctly.

A random noise (Salt-and-Pepper) is added to each image. The methodology is as follows:

PR = 1 – NL / 100

For each pixel in the image do the following

Create a uniform random number (R) in the range of -1 to +1

If R > PR then add Salt noise to the current pixel

Else if R < -PR then add Pepper noise to the current pixel

NL is the percentage of the noise level to be added to the image and it is between 0 and 100. Mersen Twister random number generator is used to obtain a sequence of uniform random numbers with good randomness and long repetition cycle. Uniform distribution is selected to give all pixels the same chance to be distorted by noise.

Using the above algorithm we create three distorted images with 5%, 10%, and 15% noise levels for each original image.

3. Cleaning Methods

Each distorted image is then cleaned by three Salt-and-Pepper cleaning methods namely: kFill [4, 5], Enhanced kFill [6], Activity Detector [7]; and their enhanced counterparts named as Algorithm A (Alg A), Algorithm B (Alg B), and Algorithm C (Alg C), respectively [8] totaling to six cleaning methods. kFill is a multi-pass two iteration filter capable of removing salt-and-pepper noise. Enhanced kFill cleans the image in a single pass. Activity Detector studies the activity around each connected component (CC) and classify CC's into three categories. The cleaning is performed by removing selected CC's based on specified criteria. A procedure named TAMD is developed to enhance noise cleaning by protecting weak features such as one-pixel-wide graphical element (GE) while removing small spurious limbs attached to the GE's. Alg A and Alg B are created by integrating TAMD into kFill and Enhanced kFill logic. TAMD is performed as a post processing step in Alg C. The parameters for the methods are set as in our previous study [8].

4. Vectorization

Three commercial software (Vectory [9], VPstudio [10], and Scan2CAD [11]) are used to vectorize cleaned images and detected vectors are saved as DXF files. These files are then converted to VEC files which have a simple format and are easier to deal with using the performance evaluation tool. Software selections are based on available features. Having the feature of detecting arcs and circles is the most important. So is the ability to output in DXF format. It would also be advantageous to use software that have been used by other researchers for performance evaluation since they may facilitate comparison and provide us with clue about its performance. The above three software were used in [12, 13].

5. Performance Evaluation

Vector Recovery Index [14] of the detected vectors is the criteria used to judge the quality of the resulting vectors. Performance evaluation tool (ArcEval2005.exe) compares the detected vector file with the ground truth file and output the VRI score. The version of the tool used carries out performance evaluation based on arcs only. All straight lines in the detected vectors file are skipped.

6. Statistical Analysis

SPSS software is used to analyze the resulting VRI values. The VRI values are stored in a form that facilitates the analysis process. Three independent variables are created: noise level (noise), cleaning method (clean), and vectorization software (vectorization). One dependent variable is created (VRI).

Descriptive statistics as well as Analysis of variance (ANOVA) can be used to study the different interactions between factors.

6.1. Setting Up the Experiment

Some parameters for the three vectorization software need to be preset prior to applying vectorization. That is to ensure consistency between different software such as: same measuring units are used and Mechanical Engineering Drawing is used as drawing type. Other parameters and thresholds are left unchanged.

For each vectorization software used, we:

0. Preset software parameters.
1. Load and convert the image into vector form and save the result in DXF form.
2. Convert DXF file into VEC file.
3. Use the performance evaluation tool to get the VRI of the detected vectors.

This is the typical steps for the experiment, but in VPStudio one parameter needs to be preset after loading the image.

6.2. Experimental Results

Six raster images are distorted by three noise levels and then cleaned by the six cleaning methods. The cleaned images are then vectorized by the three commercial raster to vector software. A VRI value is computed from each detected vector and the ground truth vector files. A total of 324 separate VRI values are generated and these values are then analyzed by SPSS. Table 1 shows the frequency table for the VRI.

Table 1. Frequency table for VRI

Measurement	Value
N	324
Mean	.419
Median	.428
Std. Deviation	.184
Skewness	-.052
Kurtosis	-.440
Minimum	.000
Maximum	.772

The minimum value of VRI is 0 which indicates no vector is detected. The mean value of VRI is 0.419 which is below the satisfactory value of 0.8 as suggested by [2]. This is partially due to the weak features of the original image as well as the amount of noise added to the image. The mean value is close to the median, suggesting normal distribution of the data. Small negative value of skewness indicates that the distribution has tiny tail to the left. Negative value of the kurtosis suggests that small proportions of the data are located in the tails of the distribution.

First we need to know which factor has a major impact on the quality of vector data. The general linear model is used to analyze the effects of different independent variables (noise, clean, and vectorization) on the result of the dependent variable (VRI). Table 2 shows that significant value of vectorization variable is less than 0.05 which means that it has the major effect on the VRI. Other factors (clean and noise) does not show significant effect on VRI. As shown in Fig. 1, VPstudio produces better quality of vector data compared to the other software and it also performs better with increased amount of noise.

Table 2. Tests of Between-Subjects Effects

Source	F	Sig.
Corrected Model	1.334	.074
Intercept	1774.551	.000
vectorization	19.562	.000
clean	.727	.604
noise	1.387	.252
vectorization * clean	.775	.653
vectorization * noise	2.630	.035
clean * noise	.180	.998
vectorization * clean * noise	.256	1.00

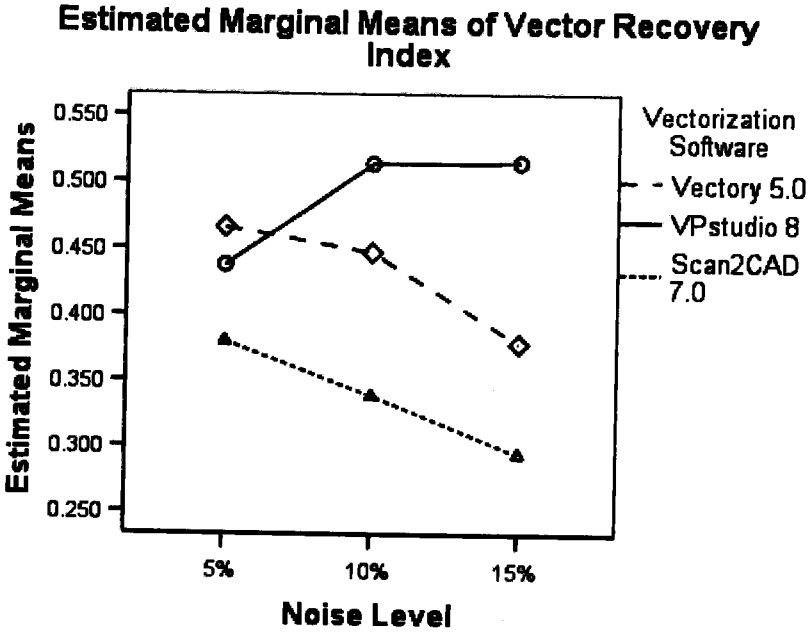

Fig. 1. Software efficiency with different noise levels

For cleaning algorithms [see Fig. 2] the estimated marginal means shows that Alg A and Alg C have better performance within all noise levels compared to their original counterparts. Alg B shows better results compared to Enhanced kFill only with high noise level. More data are needed to prove statistically that our proposed methods are significantly better than the original ones.

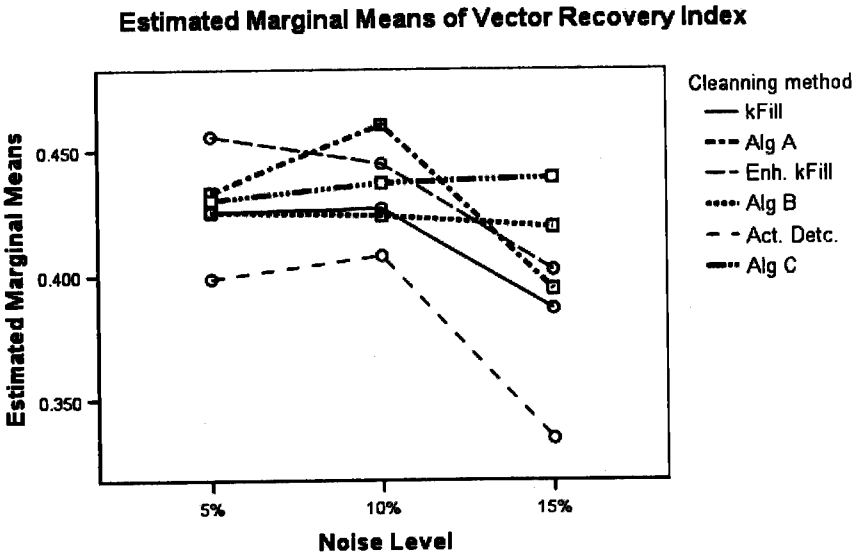

Fig. 2. Cleaning methods efficiency with different noise levels.

Based on the mean values of VRI shown in Fig. 3, we also notice that vectorization software give better performance as when the images are cleaned by specific cleaning method. Vectory gives better results when the images are cleaned by Alg A. Meanwhile VPstudio performs well with Enhanced kFill, and Scan2CAD performs well with Alg C. From Fig. 3, VPstudio also shows higher VRI scores with most cleaning methods.

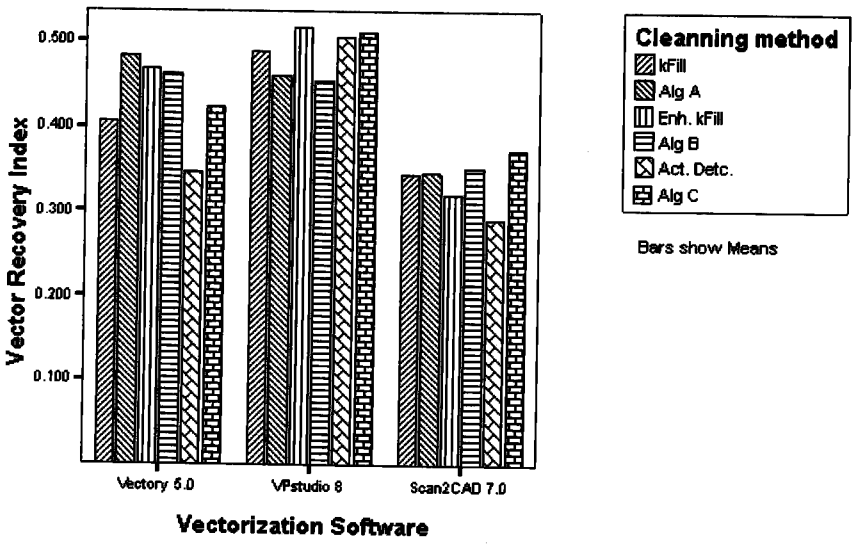

Fig. 3. Vectorization software efficiency with different cleaning methods.

7. Conclusions and Future Work

Many factors that may affect the quality of the vector data are studied in this paper including noise, cleaning methods and vectorization software. An experiment on a scanned drawings shows that vectorization software has the biggest impact on the quality of the vector data. Two of the proposed cleaning methods show better performance with the three noise levels used. Investigation on the interactions between vectorization and cleaning methods is also carried out.

We believe that the experiment in this paper should be extended into different directions in order to make it more general. Using Gaussian noise (more likely to happen in document images) is suggested. In the cleaning methods, some state of the art filters is suggested such as median filter and its variants. The set of test images is to be expanded to include more images. There are many other raster to vector software available hence the need to study their performance.

We also suggest adding more factors to the experiment. For example, if the images are classified into (simple, moderate and complex) using some criteria then we could add image complexity as a factor. The analysis may reveal new information about the interaction of image complexity with other factors. Other factors could further be classified into more specific types such as using Gaussian vs. uniform noise, and single-pass vs. multi-pass filters.

Acknowledgment
We would like to thank Low Heng Chin and Ataharul Islam of the School of Mathematical Sciences – USM for their helps on statistical analysis.

This work is fully supported by a Science Fund grant from the Ministry of Science, Technology, and Innovation (MOSTI), Malaysia under project number 01-01-05-SF0147

References

1. Tombre, K. Graphics recognition: The last ten years and the next ten years. Hong Kong, China: Springer Verlag, Heidelberg, D-69121, Germany (2006)
2. Liu, W., Report of the Arc Segmentation Contest, in Graphics Recognition: Lecture Notes in Computer Science: Recent Advances and Perspectives, Springer pp. 363--366 (2004)
3. Wenyin, L. The third report of the arc segmentation contest. Hong Kong, China: Springer Verlag, Heidelberg, D-69121, Germany (2006)
4. O'Gorman, L. Image and document processing techniques for the RightPages electronic library system. in Proc. 11th IAPR International Conference on Pattern Recognition. Conference B: Pattern Recognition Methodology and Systems. The Hague (1992)
5. Story, G.A., et al., The RightPages image-based electronic library for alerting and browsing. Computer, **25**(9): pp. 17--26 (1992)
6. Chinnasarn, K., Y. Rangsanseri, and P. Thitimajshima. Removing salt-and-pepper noise in text/graphics images. in The 1998 IEEE Asia-Pacific Conference on Circuits and Systems. Chiangmai (1998)
7. Simard, P.Y. and H.S. Malvar. An efficient binary image activity detector based on connected components. in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing. (2004)
8. Al-Khaffaf, H.S.M., A.Z. Talib, and R. Abdul Salam, Internal Report, Artificial Intelligence Research Group, School of Computer Sciences, Universiti Sains Malaysia (2006)
9. Vectory 5.0. Raster to Vector Conversion Software, Graphikon GmbH, Berlin, Germany, http://www.graphikon.de.
10. VPstudio ver 8. Raster to Vector Conversion Software, Softelec, Munich, Germany, http://www.softelec.com and http://www.hybridcad.com.
11. Scan2CAD 7.5d. Raster to Vector Conversion Software, Softcover International Limited, Cambridge, England, http://www.softcover.com.
12. Phillips, I.T. and A.K. Chhabra, Empirical performance evaluation of graphics recognition systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, **21**(9): pp. 849--870 (1999)
13. Chhabra, A.K. and I.T. Phillips. Performance evaluation of line drawing recognition systems. in Proc. 15th International Conference on Pattern Recognition. Barcelona (2000)
14. Liu, W.Y. and D. Dori, A protocol for performance evaluation of line detection algorithms. Machine Vision and Applications, **9**(5-6): pp. 240--250 (1997)