A Study on the Effects of Noise Level, Cleaning Method, and Vectorization Software on the Quality of Vector Data

Hasan S.M. Al-Khaffaf, Abdullah Zawawi Talib, and Rosalina Abdul Salam

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

Abstract. Correct detection of line attributes by line detection algorithms is important and leads to good quality vectors. Line attributes includes: end points, width, line style, line shape, and center (for arcs). In this paper we study different factors that affect detected vector attributes. Noise level, cleaning method, and vectorization software are three factors that may influence the resulting vector data attributes. Real scanned images from GREC'03 and GREC'07 contests 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 has the biggest influence on the quality of the resulting vectors.

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

1 Introduction

Raster to vector conversion is a hot topic in the field of graphics recognition [1]. Good line detection method could be judged by its ability to recognize line features correctly and thoroughly. Line features include: end points, width, line style, line shape, and center (for arcs). Since line detection usually follows other image analysis stages, its action upon the image would be affected by prior stages that change image content. Among the many factors affecting the quality of detected vector are: all kind of noise, cleaning method used, and 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.

In the noise factor three treatments (levels) are studied which is 5%, 10%, and 15% noise levels. Uniform salt-and-pepper noise is used in all three treatments. A study on

W. Liu, J. Lladós, and J.-M. Ogier (Eds.): GREC 2007, LNCS 5046, pp. 299–309, 2008.

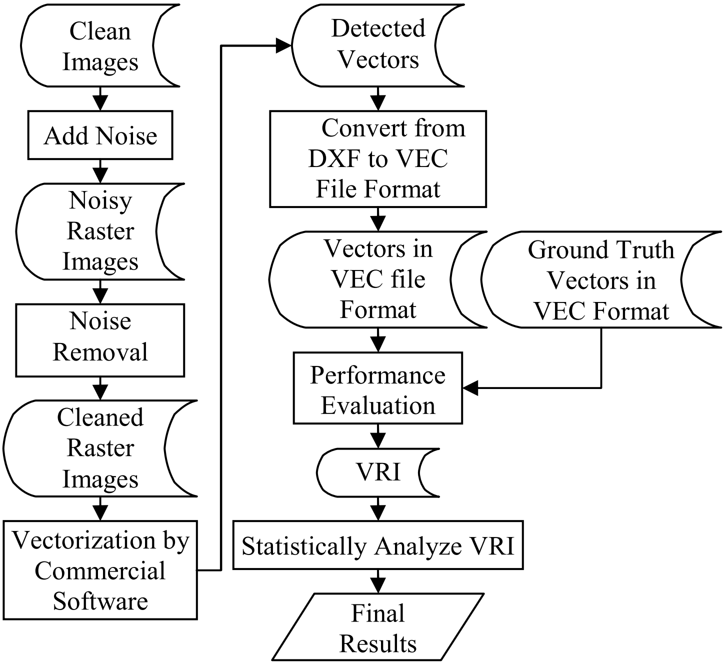
Clean Images
Add Noise
Noisy Raster Images
Noise Removal
Cleaned Raster Images
Vectorization by Commercial Software
Detected Vectors
Convert from DXF to VEC File Format
Vectors in VEC file Format
Ground Truth Vectors in VEC Format
Performance Evaluation
VRI
Statistically Analyze VRI
Final Results

Fig. 1. Flowchart showing the steps of our experiment

the effect of different noise levels on the quality of vector data is carried out. We also studied vectorization performance within different noise levels. Six algorithms are studied for the cleaning factor. The performance of these algorithms within vectorization software is also described. Finally, three commercial vectorization software are used to study vectorization factor. This factor proved to be major player on the quality of vector data. The Vector Recovery Index (VRI) is the performance evaluation criteria used to judge the quality of vector data. Statistical analysis is used to further analyze the data. Analysis of Variance (ANOVA) as well as Estimated Marginal Means (EMM) of the VRI are used. Fig. 1 shows the steps of our experiment.

The rest of the paper is organized as follows: The image data for the experiment is discussed in Section 2. This includes the method used to add the noise. Cleaning methods and vectorization software used in the study are explained in Sections 3 and 4, respectively. Performance evaluation method is described in Section 5. Statistical analysis is explained in Section 6. Finaly, conclusion and future work are presented in Section 7.

2 Image Data

The images from GREC'03 and GREC'07 contests are used since ground truth files are readily available for the performance evaluation task [4]. Another reason is that the graphical elements in GREC’03 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 algorithm 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 [5, 6], Enhanced kFill [7], Activity Detector [8]; and their enhanced counterparts named as Algorithm A (Alg A), Algorithm B (Alg B), and Algorithm C (Alg C), respectively [9] totaling to six cleaning methods. kFill is a multi-pass two iteration filter capable of removing salt-and-pepper noise. Enhanced kFill (Enh. kFill) cleans the image in a single pass. Activity Detector (Act. Detec.) studies the activity around each connected component (CC) and classifies 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 [9].

4 Vectorization

Three commercial software (Vectory [10], VPstudio [11], and Scan2CAD [12]) 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 provide us with some information and clue about its performance. The above three software were used in [13, 14].

Note that vectorization software include many features that could be utilized to enhance the detection of graphical elements thus enhancing the vector quality. Our interest is in the automatic conversion process, thus most of these features are not used.

5 Performance Evaluation

Vector Recovery Index [15] 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. For real scanned images the ground truth data may be prepared manually.

VRI is an objective performance evaluation of line detection algorithms (vectorization software in our case) that works at vector level. The VRI index is a combination of two matrices which are vector detection rate (D_v) and vector false alarm rate (F_v). The VRI is calculated as follows:

$$VRI = \beta D_v + (1 - \beta)(1 - F_v) \cdot \tag{1}$$

where β is taken as 0.5 in this work to give similar weight to vector detection rate and vector false alarm rate.

Vector detection rate is defined by two terms which is line basic quality and fragmentation quality. Line basic quality represents the accuracy of the detection of line attributes which include end points, width, line style, line shape, and center (for arcs) compared with the attributes of ground truth data. Fragmentation quality measures the fragmentation of the detected line compared to the ground truth line. The False alarm rate measures the degree of a detected line being a false alarm. VRI value is in the range of 0 to 1, the higher the better in detection.

6 Statistical Analysis

SPSS software is used to analyze the resulting VRI values. We have three factors: noise level, cleaning method, and vectorization. Hence three independent variables (IV) are created in SPSS: noise [three levels: 5%, 10%, 15%], clean [six levels: kFill, Enh. kFill, Act. Detec., Alg A, Alg B, Alg C], and vectorization [three levels: VPstudio, Vectory, Scan2CAD]. One dependent variable (DV) is created (VRI).

Since we have three different factors to study, Three-Way ANOVA is used in our analysis. The analysis are used to show the main effects and interaction (combination) effects of the IV's on the DV. The interaction effects show combination effects of two or more IV's on the DV. The description of the Three-Way ANOVA is complicated because of the three factors involved. So, an explanation of One-Way ANOVA which has only one IV (vectorization) and one DV (VRI) is illustrated below.

We start our analysis by formulating a hypothesis on our data. Our hypothesis (called null hypothesis) assume that the means of the VRI for the different levels of vectorization are equal as shown below:

$$H_0 : \mu_{VPstudio} = \mu_{Vectory} = \mu_{Scan2CAD} \tag{2}$$

Our alternative hypothesis is mutually exclusive compared to the null hypothesis and it should be exhaustive. The alternative hypothesis is shown below:

$$H_1 : \text{Not all the means are equal} \tag{3}$$

It is the significance of the F-test that shows if the group means differ. The F-test insures that any difference in group means does not happen by chance. If the change in the group means is not significant then we will assume that the IV (vectorization in this example) has no effect on DV (VRI in our case). If the significance of F-test is equal or less than 0.05, then the change in mean is considered as significant and we will reject the null hypothesis formulated above and accept the alternative hypothesis. The value 0.05 is called α and it represents the probability of rejecting the null hypothesis when it is true.

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 cleaned image into vector form and save the result as a DXF file.
2. Convert DXF file into VEC file.
3. Use the performance evaluation tool to get the VRI of the detected vectors.

These are the typical steps for the experiment, but in VPstudio one parameter needs to be preset after loading the image.

6.2 Experimental Results and Discussions

Eleven raster images are distorted with the three different levels of noise and then cleaned by the six cleaning methods. The cleaned images are then vectorized by the three commercial raster to vector software. One VRI value is computed from each detected vector and the ground truth vector files. A total of 594 separate VRI values are to be generated, but some values could not be generated and thus reducing the number of VRI values to 588. The VRI values are then analyzed by SPSS. The values that could not be generated are related to Act. Detec. and Alg C when the noise level

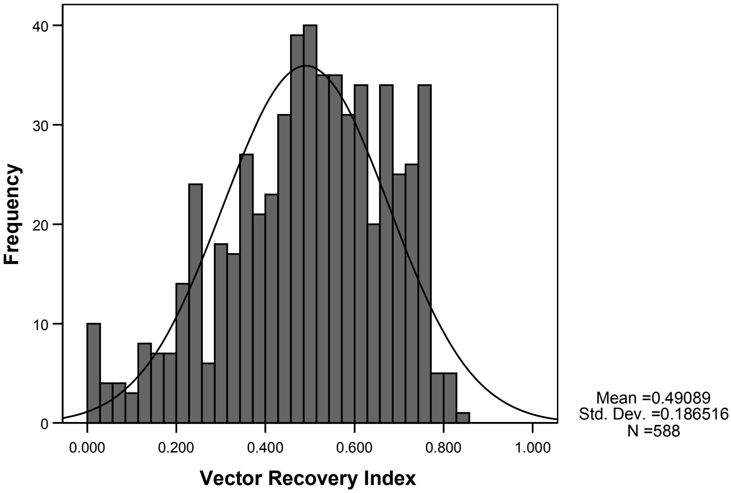

Fig. 2. Frequency table for VRI

is set to 15%. This is due to the number of CC's generated become larger than the space allocated to it in the implementation. Fig. 2 shows the frequency histogram for the VRI.

The minimum value of VRI is 0 which indicates no vector is detected. The mean value of VRI is 0.491 which is below the satisfactory value of 0.8 as suggested by [2]. The low value of VRI is due to the setting of the detected vectors width to 1 as we are not able to obtain the actual width of the detected vectors. The low value of VRI is also due to the weak features of some of the original images as well as the amount of noise added to the image. Another reason for the low value of VRI is that vectorization parameters for the three software are not modified to give better quality since we are focusing on the automatic conversion capabilities of the vectorization software. The mean value (.491) is close to the median (.508), suggesting normal distribution of the data. Small negative value of skewness (-.516) indicates that the distribution has tiny tail to the left. Negative value of the kurtosis (-.273) suggests that small proportions of the data are located in the tails of the distribution.

First we need to know factors that have major impact on the quality of vector data. Three-Way ANOVA is used to analyze the effects of different independent variables (noise, clean, and vectorization) on the dependent variable (VRI). Table 1 shows the significance of each separate factor and the combinational effect of different factors on VRI.

As shown in Table 1, the significant value (Sig.) of vectorization factor (.000) and the interaction between the two factors vectorization*noise (.012) is less than the threshold value 0.05 leading to the conclusion that vectorization and the combination of vectorization and noise do affect VRI values.

Other factors (clean and noise) and combination of factors (vectorization * clean, clean * noise, and vectorization * clean * noise) have significant values of more than 0.05 which lead to the conclusion that it does not effect VRI.

Table 1. Tests of Between-Subjects Effects

Effect	Source	F	Sig.
Main effect	vectorization	33.413	.000
	clean	1.433	.211
	noise	1.981	.139
Two-way interaction	vectorization * clean	1.341	.205
	vectorization * noise	3.227	.012
	clean * noise	.215	.995
Three-way interaction	vectorization * clean * noise	.296	.999

6.2.1 Vectorization

As shown in Fig. 3, VPstudio produces better quality of vector data compared to the other software. It also performs better with increased amount of noise when the noise level is moderate and the performance drops with increase amount of noise when the noise level is high. In fact, we have also carried out further investigation regarding performance of VPstudio by running an experiment with 20% noise for images of GREC'03 only. The result as shown in Fig. 4 confirms further that performance will drop as for other software when the noise increases. The other two software show a drop in performance with an increase amount of noise regardless of noise levels. Contrast test also shows that VPstudio has significant difference over the other two vectorization software.

VPstudio which has the best performance in VRI has the least sensitivity with any cleaning method as shown in Fig. 5. However, its best performance is when it works with Enh. kFill (Estimated Marginal Means of VRI = 0.589) and lowest result when it works with Alg B (0.546).

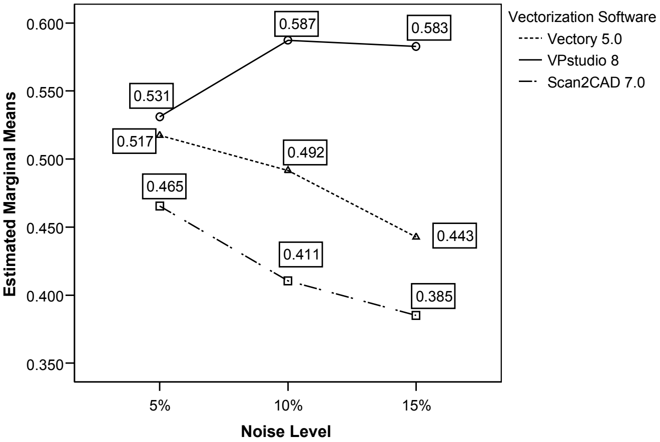

Fig. 3. Software efficiency with different noise levels

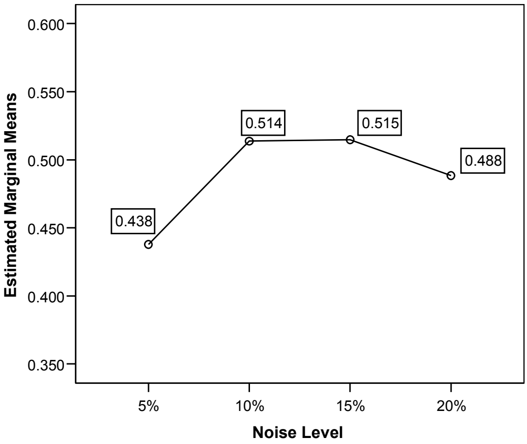

Fig. 4. VPstudio efficiency with different noise levels

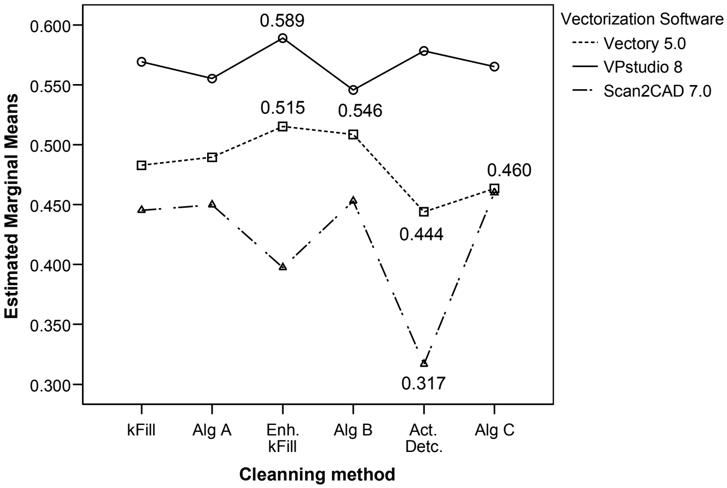

Fig. 5. Efficiency of vectorization software with many cleaning methods

Fig. 5 also shows that Vectory exhibit moderate sensitivity to cleaning methods and has better quality with images cleaned by Enh. kFill (0.515) and lowest VRI with images cleaned by Act. Detec. (0.444).

Based on Fig. 5, Scan2CAD shows highest sensitivity to cleaning methods. It has best performance when working with Alg C (0.460) and global lowest value of 0.317 with Act. Detec.

6.2.2 Noise Levels

The EMM of VRI show a slight drop in performance with increase amount of noise as shown in Table 2. The three levels of noise used in this study show little impact on

Table 2. Estimated Marginal Means of VRI with different noise levels

Noise Level	EMM of VRI
5%	0.505
10%	0.496
15%	0.471

the result of VRI. More levels of noise are required in order to show the real impact of noise levels on VRI.

6.2.3 Cleaning Methods

All cleaning methods (except Act. Detec.) show close performance (see Table 3). Act. Detec. has the lowest performance compared to others because it could not remove noise that touches GE and may lead to difficulties during the recognition process. Alg C which is an enhanced version of Act. Detec. performs better than Act. Detec. since it did not suffer the aforementioned drawback, but its performance is close to the other four algorithms.

Table 3. Estimated Marginal Means of VRI for cleaning methods

Cleaning method	EMM of VRI
kFill	.499
Alg A	.498
Enh. kFill	.500
Alg B	.502
Act. Detec.	.448
Alg C	.496

We have observed that even if some noise still exist in most image area (especially in 15% noise level) such as in Enh. kFill and Alg B due to its single pass nature these methods perform close to multi pass filters, with respect to EMM of VRI.

For cleaning algorithms, EMM of VRI shows that Alg A and Alg B have similar performance compared to their original counterparts.

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. 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. Ongoing investigations include using Gaussian noise (more common in document images), and for the cleaning methods some state of the art filters are being used such as median filter and its variants.

Morphological operators should also be investigated. 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. Thanks also go to Faisal Shafait of University of Kaiserslautern who provides the ground truth data for GREC'07 contest's test images.

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. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 422–426. Springer, Heidelberg (2006)
2. Liu, W.: Report of the Arc Segmentation Contest. In: Graphics Recognition: Lecture Notes in Computer Science: Recent Advances and Perspectives, pp. 363–366. Springer, Heidelberg (2004)
3. Wenyin, L.: The third report of the arc segmentation contest. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 358–361. Springer, Heidelberg (2006)
4. Shafait, F., Keysers, D., Breuel, T.M.: GREC 2007 Arc Segmentation Contest: Evaluation of Four Participating Algorithms, vol. 5046. Springer, Heidelberg (2007)
5. 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, pp. 260–263 (1992)
6. Story, G.A., O'Gorman, L., Fox, D., Schaper, L.L., Jagadish, H.V.: The RightPages image-based electronic library for alerting and browsing. Computer 25(9), 17–26 (1992)
7. Chinnasarn, K., Rangsanseri, Y., Thitimajshima, P.: Removing salt-and-pepper noise in text/graphics images. In: The 1998 IEEE Asia-Pacific Conference on Circuits and Systems, Chiangmai, pp. 459–462 (1998)
8. Simard, P.Y., Malvar, H.S.: An efficient binary image activity detector based on connected components. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 229–233 (2004)
9. Al-Khaffaf, H.S.M., Talib, A.Z., Abdul Salam, R.: Internal Report, Artificial Intelligence Research Group, School of Computer Sciences, Universiti Sains Malaysia (2006)
10. Vectory 5.0. Raster to Vector Conversion Software. Graphikon GmbH, Berlin, Germany, `http://www.graphikon.de`

11. VPstudio ver 8.02 C6. Raster to Vector Conversion Software, Softelec, Munich, Germany, http://www.softelec.com, http://www.hybridcad.com
12. Scan2CAD 7.5d. Raster to Vector Conversion Software, Softcover International Limited, Cambridge, England, http://www.softcover.com
13. Phillips, I.T., Chhabra, A.K.: Empirical performance evaluation of graphics recognition systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(9), 849–870 (1999)
14. Chhabra, A.K., Phillips, I.T.: Performance evaluation of line drawing recognition systems. In: Proc. 15th International Conference on Pattern Recognition, Barcelona, pp. 864–869 (2000)
15. Liu, W.Y., Dori, D.: A protocol for performance evaluation of line detection algorithms. Machine Vision and Applications 9(5-6), 240–250 (1997)