IEICE TRANS. INF. & SYST., VOL.E94–D, NO.6 JUNE 2011

PAPER

Empirical Performance Evaluation of Raster-to-Vector Conversion Methods: A Study on Multi-Level Interactions between Different Factors

Hasan S.M. AL-KHAFFAF[†a)], ***Member***, **Abdullah Z. TALIB**[†b)], ***and*** **Rosalina ABDUL SALAM**[††c)], ***Nonmembers***

SUMMARY Many factors, such as noise level in the original image and the noise-removal methods that clean the image prior to performing a vectorization, may play an important role in affecting the line detection of raster-to-vector conversion methods. In this paper, we propose an empirical performance evaluation methodology that is coupled with a robust statistical analysis method to study many factors that may affect the quality of line detection. Three factors are studied: noise level, noise-removal method, and the raster-to-vector conversion method. Eleven mechanical engineering drawings, three salt-and-pepper noise levels, six noise-removal methods, and three commercial vectorization methods were used in the experiment. The Vector Recovery Index (VRI) of the detected vectors was the criterion used for the quality of line detection. A repeated measure ANOVA analyzed the VRI scores. The statistical analysis shows that all the studied factors affected the quality of line detection. It also shows that two-way interactions between the studied factors affected line detection.

key words: *raster-to-vector conversion, performance evaluation, salt/pepper noise, engineering drawing, binary image, statistical analysis, repeated measure ANOVA, document analysis and recognition*

1. Introduction and Literature Review

The empirical performance evaluation of raster-to-vector conversion methods is an important topic in the area of graphics recognition [1], [2]. Comparing the performance of a vectorization method with third party methods will not only prove that the quality of one's own method is better than the others but also will gauge the maturity of the raster-to-vector methods being studied. The outputs of the systems (methods) are compared with that of others using test images and a selected performance evaluation criterion, which may include time and storage [3]–[5]. However, it is highly recommended to perform such a test (or more precisely a large systematic test) among many methods (research prototypes and/or commercial software) using a unified platform with suitable test images and a proper performance evaluation method. It is also desirable to study the effect of other factors (processes) on the quality of vectorization systems. These factors may include noise type, quantity of noise, and noise-removal methods. Performing such an experiment is not only complex and labor intensive, but it also may require a proper statistical test to analyze the output of the experiment.

Phillips and Chhabra [6] performed an empirical test on three automatic raster-to-vector converters (two commercial and one research prototype). *EditCostIndex* was the performance evaluation criterion. The experiment tested the detection of straight lines (dashed and solid), arcs, circles, and text. The test data was synthetic images with their corresponding ground truth images in VEC format. The VEC file format is also introduced in this work. Three vectorization software applications (two commercial and one research prototype) were used: I/Vector (Vectory), VPstudio, and MDUS. However, no noise was incorporated into the images used in the experiment. To reveal the weaknesses and the strengths of the systems (methods), the authors suggested increasing the degradation of the test images and using more complex drawings in future studies.

Chhabra and Phillips [7] performed another empirical test to evaluate complete vectorization systems. The major advantage of this test compared with previous works is that real scanned images with their ground truth data were generated manually. The work also describes how to set up such experiments. Because of the time-intensive process of converting real paper images into usable scanned images, only a small subset (ten images) of the available paper images were used. Generating the ground truth data and aligning them with the real scanned one was another labor-intensive part of the experiment. The ground truth data was saved as VEC files. The empirical test involved four vectorization systems (three commercial and one research prototype): Scan2CAD, TracTrix, Vectory, and VrLiu. Again, *EditCostIndex* was used as the performance evaluation criteria.

The focus of Wenyin et al. [8] was on solid arc detection in raster images. Seven images (four synthesized and three scanned) were used in the test. The ground truth data were saved as VEC files. The overall average of VRI scores was used as the ultimate measure of performance. The test involved two research prototypes (the methods of Dave Elliman [9] and Xavier Hilaire [10]) with their parameters fixed during the test. No commercial software were tested. Four types of noise were introduced into the images: Gaussian, high-frequency, hard pencil, and geometry distortion.

Manuscript received October 5, 2010.
Manuscript revised February 23, 2011.

[†]The authors are with School of Computer Sciences, Universiti Sains Malaysia, USM 11800 Penang, Malaysia.
[††]The author is with the Faculty of Science & Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai 71800, Nilai, Negeri Sembilan, Malaysia.
a) E-mail: hasan@m.ieice.org
b) E-mail: azht@cs.usm.my
c) E-mail: rosalina@usim.edu.my
DOI: 10.1587/transinf.E94.D.1278

Copyright © 2011 The Institute of Electronics, Information and Communication Engineers

Further research by Wenyin [11] was also on solid arc detection from engineering drawings. Twelve real scanned and synthetically generated images were used (some images were corrupted by artificial noise). The ground truth data were saved as VEC files. Again, the systems were run as a black box with fixed parameters, and no user intervention was allowed during the test. The overall average of VRI was the unique measure of performance. Two research prototypes (the methods of Song JiQiang and Dave Elliman [9]) were evaluated. The recognition of arcs was more challenging because the test images were complex and contained tangent arcs that were hard to precisely locate.

The main theme in Wenyin's study [12] was arc segmentation in engineering drawings. The new element in this experiment included the use of eighteen new images (six real scanned and twelve noisy versions). The ground truth data were saved as VEC files. The images were distorted by a moderate amount of salt-and-pepper noise. The average of the VRI was the performance evaluation criterion. However, an updated version of the VRI formula was used. The new formula ($VRI = \sqrt{D_v * (1 - F_v)}$) used the geometrical mean rather than the mathematical mean. Three research raster-to-vector software (the methods of Dave Elliman [9], Daniel Keysers & Thomas Breuel [13], and Xavier Hilaire [14]) were used in this test. However, the noise effect was not large, and some noisy images obtained high scores when compared with their original clean ones [12].

Shafait et al. [15] shifted the way of representing the ground truth data and the performance evaluation. Five real scanned images were used. Ground truth data were generated manually from the scanned drawings and stored in TIF image files. Vectorial score was the criterion rather than VRI. The experiment used four vectorization software, out of which three were commercial: VPstudio (VP), Vectory (Vec), and Scan2CAD (S2CAD); one was a research method (VrLiu). No noise was used in this experiment.

Al-khaffaf et al. [16] proposed a methodology to study the effect of many factors that may affect line detection. The three factors were noise-removal method, noise level, and raster-to-vector method. The study was performed using six noise-removal methods: kFill [17], [18], Enhanced kFill (EkFill) [19], Activity Detector (AD) [20], and their respective enhanced counterparts Algorithm A (AlgA) [21], Algorithm B (AlgB) [22], and Algorithm C (AlgC) [23]. Three noise levels were studied: 5%, 10% and 15%. Three commercial raster-to-vector conversion methods were tested: VP, S2CAD, and Vec. The experiment used eleven images from GREC'03 and GREC'07. VRI was the performance evaluation measure. Although this paper studied many factors, the statistical method that analyzed VRI scores could only answer limited questions in the context of revealing the important factors that affect line detection. The interactions between the studied factors were also not shown.

Al-khaffaf et al. [24] studied the performance of two research prototypes (VrLiu [25] and Qgar-Lamiroy [26]) and three commercial software (VP, S2CAD, and Vec). The work also included studying the performance of many versions of one commercial software. The study created new test images, and VRI was the performance index. No artificial noise was used.

From the review above, we conclude that empirical performance evaluation tests are already becoming a trend within the graphics recognition community. These studies have usually been performed during contests attached to the International Workshop on Graphics Recognition (GREC). One advantage of such contests is the adoption of the contest data and evaluation methods by other researchers in their work and publications [1], [2].

With all the advantages brought by the previous studies, there are still some issues and shortcomings that need tackling by researchers, such as the following:

1. There is insufficient research on the effect of noise on raster-to-vector conversion process and the use of an unspecified small amount of noise (such as the study of Wenyin [12]) or performing the test on clean images (such as the study of Shafait et al. [15]).
2. The interaction between noise level and noise removal is not studied in the context of raster-to-vector conversion. The interaction could be considered obvious when we look at it as an image processing problem in which more noise in the image make it more difficult to remove the noise. However, the interaction between these factors in the context of document image analysis and recognition is not trivial, and it still needs to be studied. Here, we can cite the work of Wenyin [12], in which the author sheds some light on the effect of salt-and-pepper noise on the quality of line detection.
3. In the case of using noise, it has not been shown which method (if any) removes the noise before performing vectorization. Hence, the effect of the interaction between noise-removal methods and raster-to-vector methods is not clear. Here, we recall the question of Karl Tombre [2]: "Do we actually test the quality of the de-noising method or the recognition capabilities of the method?" The interaction between the noise-removal method and the vectorization algorithm is not obvious. As is demonstrated in this paper, vectorization methods have varying sensitivity to noise-removal methods. A vectorization method would have better performed line detection if the noise was removed using one noise-removal method rather than the other (Sect. 4.2.1 of this paper).
4. When noise is used, it is not stated how much noise is added to the image. Also, the effect of using many levels of noise is not studied.
5. The interaction between many factors (noise-removal method and vectorization method, for example) that may affect line detection has not been studied yet.
6. The effect of the resolution factor also has not been studied.

This paper tackles the first five issues stated above. Many types of noise may appear in scanned images. However, in this study, the focus was on salt-and-pepper noise

only. Other types of noise were not covered in this study. The resolution factor was not studied because of the lack of availability of images with different resolutions within the graphics recognition community. The available image datasets usually have one resolution. Most of the available datasets were created during Arc Segmentation Contests attached to the GREC workshop, such as the following four editions: GREC'03, GREC'05, GREC'07, and GREC'09. The creation of a new dataset is time consuming and labor intense for the time being (it is necessary to search for many paper drawings, scan the drawings with different resolutions, and perform ground truthing for all of them). However, it may be a good subject for another study.

The rest of this paper is organized as follows. Section 2 shows the limitation of a related study and how it is avoided in the proposed methodology. Section 3 presents the steps of the proposed methodology. The steps include selecting/corrupting the images dataset, removing the noise, vectorizing the images, measuring the performance index, and statistically analyzing the performance index scores. The details of the statistical test are presented in Sect. 4. This includes the preliminary statistical test on data, analysis of each variable, and the analysis of the interaction between variables. The conclusions of this work are presented in Sect. 5.

2. Background and the Proposed Method

The suitability of a vectorization method in terms of line detection can be judged by its ability to recognize line features correctly and thoroughly. Line features include end points, line width, line style, line shape, and center (for arcs). Because 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 the detected vector are the amount of noise in the original raster image, the noise-removal method, and the vectorization algorithm, which detects lines and their features. A recent study [12] gives some insight on how noise affects the resulting vector data, but only one unspecified level of salt-and-pepper noise was added to the images separately. The authors also did not study the effect of different noise-removal methods on the quality of line detection. The interactions between different factors (vectorization and noise removal, for example) were also not studied. The total number of VRI values analyzed was only 54 (#images $*$ 3 $*$ #vectorization methods = $6 * 3 * 3$), which is not enough for a more stringent analysis of the results. Because the study only uses one noise level, the effect of noise can only be sensed by looking at the VRI values and/or by looking at the mathematical mean. Another limitation of this methodology is that it prevents the researcher from performing a rigorous study on the effect of each factor and the effects of interaction between the different factors.

This paper proposes a new methodology that studies many factors that may affect the quality of line detection. As with many previous studies, VRI is the performance index. Noise factor, used in other studies, is also to be inspected here. However, our study uses it in a more systematic way, and we will also study many levels of noise. The third factor is noise removal (newly studied). As opposed to other methodologies that can only evaluate raster-to-vector methods, the proposed methodology could be used to study the interactions between many different factors affecting the quality of line detection and enables a proper analysis based on a statistical test. Other methodologies focus on studying only one factor (vectorization), and their focus on other factors is limited. Because the proposed methodology relies on statistical analysis, it can detect significant improvements in performance for any studied algorithm in the context of line detection. Other methodologies can only show the performance of the tested vectorization methods, which may not be enough when more stringent criteria are required to select a method for specific industrial problems.

Figure 1 shows the detail of the proposed methodology. The next section presents the details of each step of the proposed methodology.

3. Steps of the Proposed Methodology

3.1 Selecting/Corrupting the Images

Real scanned images of mechanical engineering drawings were selected. This type of image contains straight lines as well as circular arcs. The images from the GREC'03 and GREC'07 contests [11], [15] were used because ground truth files were readily available for the performance evaluation task.

Uniform salt-and-pepper noise was added to each image in an independent manner (i.e., the original images were always used to generate noisy images). In this way, bias was avoided, and the data is suitable for the statistical analysis. We have used the same amount of noise as in our previous study [16]. Because the image data in this area of research (Graphics Recognition) is mostly binary images (black and white), the highest noise level used (15%) was not small. High noise values will distort the fine lines in such drawings, which may render the vectorized image useless and make the process of vectorization meaningless. In this research, the focus is not on the noise-removal factor alone but on many factors that may affect line detection; hence, it is reasonable to use three noise levels taken arbitrarily in the range from 5% to 15%. This will ensure that the noisy images are still usable (it is practical for them to be vectorized) and that the interaction between the factors can be studied. The total number of images created by corrupting images with three levels of noise was #images $*$ #noise levels = $11 * 3 = 33$.

3.2 Removing the Noise

All noisy images were then cleaned by six salt-and-pepper noise-removal methods. We used the same methods as in our previous study [16]. The parameters for the noise-removal methods were set as follows: the window size

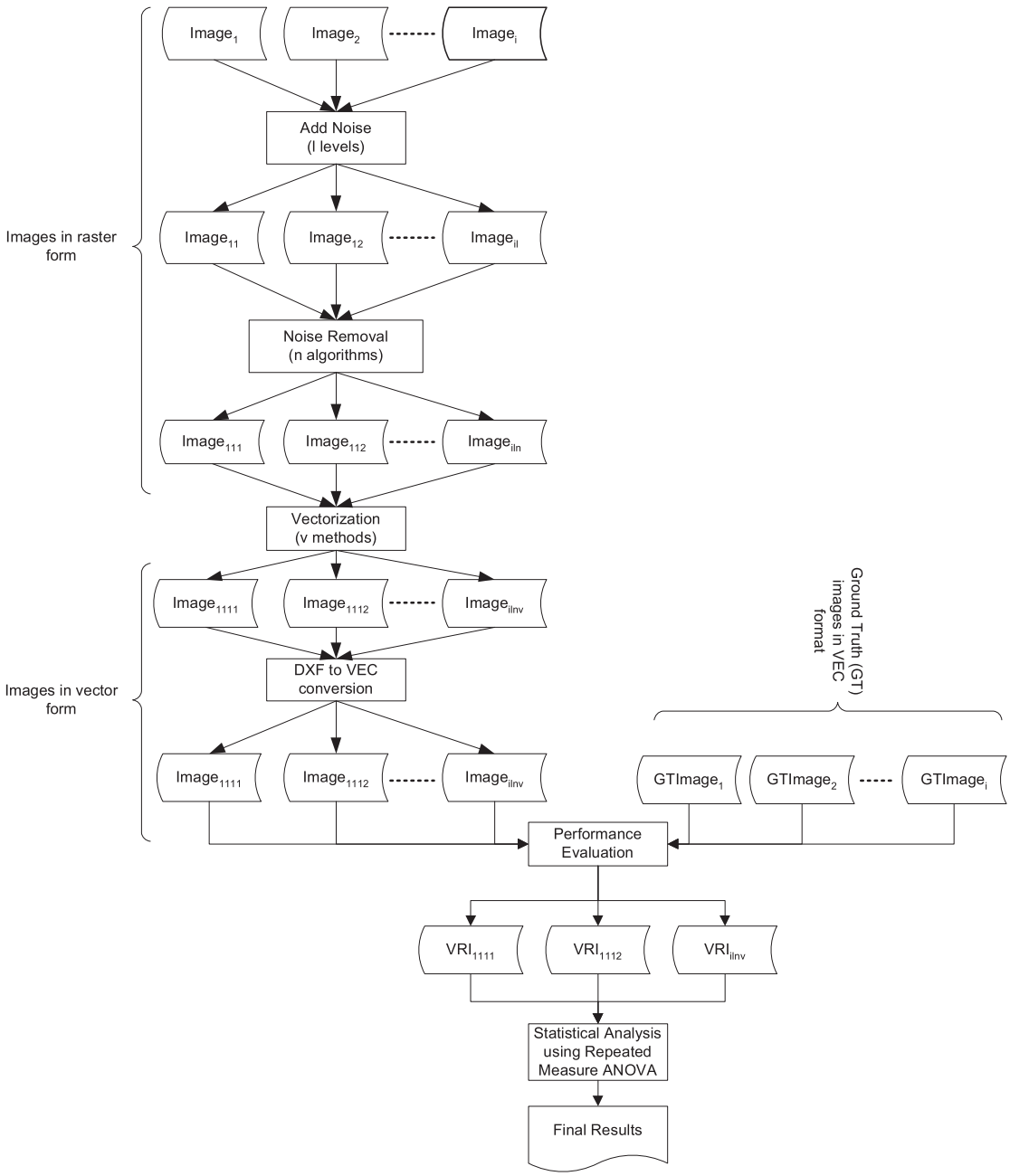

Fig. 1 Detailed steps of the proposed methodology.

was set to $3 * 3$ pixels for all algorithms. The parameters for AD were set according to the values suggested by Simard and Malvar [20] to perform strong noise removal. For AlgA, AlgB, and AlgC, *LT* (Length Threshold) was set to 4, 5, and 6 for 5%, 10%, and 15% noise, respectively. The total number of images created by cleaning all the noisy images using the six noise-removal algorithms was #noisy images $*$ #noise-removal methods $= 33 * 6 = 198$.

3.3 Vectorizing the Images

The cleaned images were then vectorized by several commercial software. The three vectorization software that were used in our previous study [16] were also used here. The software applications vectorized cleaned images and saved the detected vectors as DXF files. These files were then converted to VEC files, which have a simple format and are easier to deal with using the performance evaluation tool. Our interest was in the automatic conversion process; thus, most software features that could be manually used to enhance the detection were not used.

Some parameters that the three vectorization software needed were pre-set prior to applying vectorization. This ensured consistency between different software. The measuring units were unified, and the drawing type was set to Mechanical Engineering. Other parameters and thresholds

were unchanged.

The total number of vector images created by vectorizing the cleaned images using the three vectorization software was #cleaned images $*$ #vectorization software $= 198 * 3 = 594$.

3.4 Measuring the Performance Index Scores

The VRI of the detected vectors was the criterion used to judge the quality of vector detection. The performance evaluation method[†] compared the detected vector file with the ground truth file and outputted the VRI score. VRI is an objective performance evaluation of line detection algorithms (vectorization software in our case) that works at the vector level. The VRI index is a combination of two matrices: vector detection rate D_v and vector false alarm rate F_v. The VRI is calculated as in Eq. (1) below.

$$VRI = \sqrt{D_v * (1 - F_v)} \quad (1)$$

The vector detection rate (D_v) is defined by two terms: 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 with the ground truth line. The vector false alarm rate (F_v) measures the probability of a detected line being a false alarm. VRI value is in the range of 0 to 1, with higher values indicating better vector recovery.

VRI is a well accepted criterion for performing empirical performance evaluation of raster-to-vector methods. It has been used in several editions of the Arc Segmentation Contests held in conjunction with GREC. The total number of VRI scores obtained by running the performance evaluation tool is equal to the total number of vector images generated by the vectorization.

3.5 Statistically Analyzing the Performance Index Scores

Our experiment included three factors to be studied, with many levels for each factor. Hence, hundreds of VRI scores generated by applying the three factors on the images needed analysis. Simple statistics (such as the mathematical mean) were not sufficient to show the significance of a specific factor, nor can they directly explain the interactions between the different factors. A repeated measure ANOVA is a suitable statistical analysis method used in our experiment, considering that many factors were involved in the study and each subject (image) participated in more than one score (measurement). ANOVA is a well-known statistical method. However, the use of ANOVA helps extract more information on the interaction between two or more studied factors. That is the reason why ANOVA is used in this study. The methodology coupled with ANOVA could be used to find the best match of the off-the-shelf noise-removal methods (for example) with other raster-to-vector conversion methods. The methodology, however, is not limited to the three different factors already studied in the paper, but it is also applicable to other factors stemming from research preferences.

4. Experimental Results and Discussions

The work flow of the experiment can be summarized as follows. The eleven raster images were distorted with the three noise levels and then cleaned by the six noise-removal methods. The cleaned images were then vectorized by the three commercial raster-to-vector software. One VRI value (score) was computed from each detected vector file and its corresponding ground truth vector file. A total of 594 separate VRI values were generated out of the performance evaluation stage shown in Fig. 1, but some values could not be generated and thus reduced the number of VRI values to 588. The VRI values were then analyzed by a repeated measure ANOVA. The values that could not be generated were related to AD and AlgC when the noise level was set to 15%. This is due to the number of connected components generated that turned out to be larger than the allocated space in the implementation. Because some data are missing, we had an unequal n design and reporting EMM rather than the observed mean, avoiding the bias incurred in calculating the mathematical mean when some data are missing.

For clarity purposes, the three variables (*Vectorization*, *Cleaning*, *Noise*) are shown in italics when referenced in the text.

Because we have many factors to study and each image was used in measuring more than one VRI score, a repeated measure ANOVA was used to analyze the VRI scores.

There are three requirements to use this statistical test [27]: (i) order effects should be avoided. This condition was guaranteed because we used separate copies of the image before applying any treatment (one level of a factor); (ii) the data in each cell is normally distributed (some abnormality is accepted); and (iii) the sphericity condition should not be violated.

The second and third requirements above are explained below:

1. Before proceeding with a repeated measure ANOVA, the data needs validation for the analysis. The Shapiro-Wilk test checks the normality of the data in each cell of the design. Equations (2) and (3) show the null hypothesis and the alternative hypothesis for the Shapiro-Wilk test, respectively.

$$H_0 : \text{There is no difference between the distribution of the data and the normal.} \quad (2)$$

$$H_1 : \text{There is a difference between the distribution of the data and the normal.} \quad (3)$$

[†]http://www.cs.cityu.edu.hk/~liuwy/ArcContest

Table 1 Tests of within-subjects contrasts.

Measure:VRI						
Source	Vectorization	Cleaning	Noise	df	F	Sig.
Vectorization	Vec vs. VP			1	468.955	**.000**
	VP vs. S2CAD			1	829.028	**.000**
Cleaning		kFill vs. AlgA		1	.030	.867
		AlgA vs. EkFill		1	.141	.716
		EkFill vs. AlgB		1	.172	.688
		AlgB vs. AD		1	114.338	**.000**
		AD vs. AlgC		1	225.912	**.000**
Noise			5% vs. 10%	1	1.549	.245
			10% vs. 15%	1	74.618	**.000**
Vectorization * Cleaning	Vec vs. VP	kFill vs. AlgA		1	.234	.640
		AlgA vs. EkFill		1	4.959	.053
		EkFill vs. AlgB		1	4.768	.057
		AlgB vs. AD		1	10.444	**.010**
		AD vs. AlgC		1	4.916	.054
	VP vs. S2CAD	kFill vs. AlgA		1	7.133	**.026**
		AlgA vs. EkFill		1	17.265	**.002**
		EkFill vs. AlgB		1	17.050	**.003**
		AlgB vs. AD		1	266.484	**.000**
		AD vs. AlgC		1	177.488	**.000**
Vectorization * Noise	Vec vs. VP		5% vs. 10%	1	22.109	**.001**
			10% vs. 15%	1	10.412	**.010**
	VP vs. S2CAD		5% vs. 10%	1	92.973	**.000**
			10% vs. 15%	1	5.220	**.048**
Cleaning * Noise		kFill vs. AlgA	5% vs. 10%	1	2.066	.184
			10% vs. 15%	1	2.228	.170
		AlgA vs. EkFill	5% vs. 10%	1	.525	.487
			10% vs. 15%	1	.926	.361
		EkFill vs. AlgB	5% vs. 10%	1	.108	.750
			10% vs. 15%	1	6.044	**.036**
		AlgB vs. AD	5% vs. 10%	1	.600	.458
			10% vs. 15%	1	68.143	**.000**
		AD vs. AlgC	5% vs. 10%	1	.281	.609
			10% vs. 15%	1	6.546	**.031**

If the significance (ρ) of the Shapiro-Wilk test is less than or equal to .05, then the null hypothesis will be rejected and the alternative hypothesis will be accepted. A normality test shows that 50 out of 54 cells are normally distributed (i.e., we fail to reject the null hypothesis). Four cells are not normally distributed. Their data are skewed to the left (skewness is negative). ANOVA is not very sensitive to the normality (unless the abnormality is severe). From the abnormality test, we note that the abnormality of the four cases is not severe and it is caused by some outliers. Hence, the normality condition is not violated.

2. The Mauchly's Test is required to ensure that the sphericity condition is not violated. If so, our data will be suitable for interpretation using the Tests of Within-Subjects Effects.
   For *Vectorization*, the Mauchly's Test is significant $F(2, 18) = 327.189$, $\rho < .05$, which infers that the sphericity assumption is violated. However, we will choose to interpret the Tests of Within-Subjects Effects by using the Sig. of Huynh-Feldt in the Tests of Within-Subjects Effects.
   The Mauchly's Test of Sphericity for *Cleaning* is not significant $F(14, 45) = 54.816$, $\rho > .05$ which infers that the sphericity assumption is not violated. This means that we fail to reject the null hypothesis of Mauchly's Test. Hence, we will interpret the Tests of Within-Subjects Effects.
   The Mauchly's Test of Sphericity for *Noise* is not significant $F(2, 18) = 29.077$, $\rho > .05$. This means that we fail to reject the null hypothesis of Mauchly's Test. Hence, we will interpret the Tests of Within-Subjects Effects.

At this stage, the data were ready to be analyzed. GLM repeated measure[†] was used to analyze the resulting VRI values. We had three factors: noise level, noise-removal method, and vectorization. Hence, three independent variables (IV) were created: *Noise* [three levels: 5%, 10%, and 15%], *Cleaning* [six methods: kFill, EkFill, AD, AlgA, AlgB, and AlgC], and *Vectorization* [three software: VP, Vec, and S2CAD]. One dependent variable (DV) was created (*VRI*).

The analysis of VRI scores using a repeated measure ANOVA will be explained in the following sections. All studied factors and two-way interactions were shown to be significant (Sig. $\leq$ .05). Hence, the Within-Subjects Contrasts (Table 1) can be further interpreted.

[†]SPSS menu item: Analyze-> General Linear Model-> Repeated Measures

Table 2 Pair-wise comparison of vectorization software.

Measure:VRI

(I) Vectorization	(J) Vectorization	Mean Difference (I-J)	Sig.[a]
Vec	VP	−.080*	.000
	S2CAD	.066*	.000
VP	Vec	.080*	.000
	S2CAD	.147*	.000
S2CAD	Vec	−.066*	.000
	VP	−.147*	.000

Based on estimated marginal means

*. The mean difference is significant at the .05 level.

[a]. Adjustment for multiple comparisons: Bonferroni.

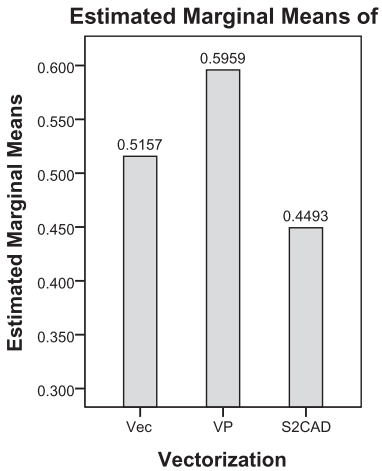

Fig. 2 Performance comparison of vectorization methods.

Table 3 Pair-wise comparison of cleaning methods.

Measure:VRI

(I) Cleaning	(J) Cleaning	Mean Difference (I-J)	Sig.[a]
kFill	AlgA	.001	1.000
	EkFill	.003	1.000
	AlgB	.001	1.000
	AD	.067*	.000
	AlgC	.010	1.000
AlgA	kFill	.000	1.000
	EkFill	.002	1.000
	AlgB	−6.667E-5	1.000
	AD	.067*	.000
	AlgC	.010	1.000
EkFill	kFill	−.003	1.000
	AlgA	−.002	1.000
	AlgB	−.002	1.000
	AD	.065*	.000
	AlgC	.008	1.000
AlgB	kFill	.000	1.000
	AlgA	6.667E-5	1.000
	EkFill	.002	1.000
	AD	.067*	.000
	AlgC	.010	1.000
AD	kFill	−.067*	.000
	AlgA	−.067*	.000
	EkFill	−.065*	.000
	AlgB	−.067*	.000
	AlgC	−.057*	.000
AlgC	kFill	−.010	1.000
	AlgA	−.010	1.000
	EkFill	−.008	1.000
	AlgB	−.010	1.000
	AD	.057*	.000

Based on estimated marginal means

[a]. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

4.1 Single-Factor Effects

The next three sections present the effect of the three factors independently. For each factor, the effect of each level on the VRI index is shown.

4.1.1 Vectorization Effect

The *Vectorization* factor was significant, $F(1.268, 11.411) = 327.189$, $\rho < .001$. The differences in means between Vec vs. VP and VP vs. S2CAD were significant, $F(1, 9) = 468.955$, $\rho < .001$ and $F(1, 9) = 468.955$, $\rho < .001$, respectively (Table 1). This difference can be seen by investigating Fig. 2. The pair-wise comparison in Table 2 shows that the difference between software performances was significant ($\rho < .001$). We conclude that the three vectorization software had significant differences between them in the context of VRI scores. VP was the best performer (EMM = .596) followed by Vec (EMM = .516). S2CAD was the lowest performer (EMM = .449).

4.1.2 Noise-Removal Effect

The *Cleaning* factor was significant, $F(5, 45) = 54.816$, $\rho < .001$. The differences in means (Table 3) between AD and the other five algorithms were significant[†]. This difference can be seen by investigating Fig. 3. It indicates a low performance for AD compared to the other noise-removal methods. However, the mean differences between each of the other five algorithms were not significant, indicating that their performances were similar.

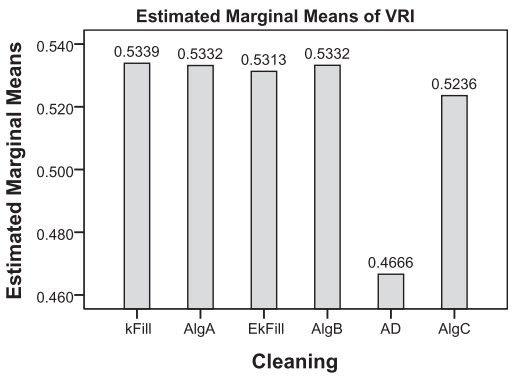

Fig. 3 Performance comparison of noise-removal methods.

[†]An asterisk symbol (*) is shown in the mean difference column if the corresponding mean difference is significant.

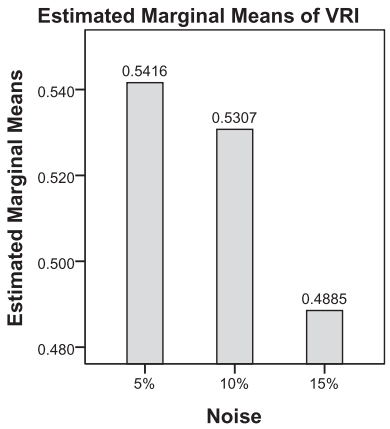

Fig. 4 Performance comparison of noise levels.

Table 4 Pair-wise comparison of noise levels.

Measure:VRI

(I) Noise	(J) Noise	Mean Difference (I-J)	Sig.[a]
5%	10%	.011	.734
	15%	.053*	.000
10%	5%	−.011	.734
	15%	.042*	.000
15%	5%	−.053*	.000
	10%	−.042*	.000

Based on estimated marginal means

[a]. Adjustment for multiple comparisons: Bonferroni.

*. The mean difference is significant at the .05 level.

4.1.3 Noise Level Effect

The *Noise* factor was significant, $F(2, 18) = 29.077$, $\rho < .001$. The difference in means (Table 1) between 5% and 10% noise was not significant, $F(1, 9) = 1.549$, $\rho = .245$. However, the difference between 10% and 15% noise was significant, $F(1, 9) = 74.618$, $\rho < .001$. This difference can be observed by investigating Fig. 4. The pair-wise comparisons in Table 4 show that the difference between the EMM of VRI for 5% and 10% noise was not significant. However, there was a significant difference between 10% and 15% noise and between 5% and 15% noise. This indicates that the VRI performance index did not drop much when the amount of noise in the image was increased from 5% to 10%, whereas the performance drops significantly when the noise level was increased from 10% to 15%.

4.2 Multi-Factor Interaction Effects

As opposed to the previous three sections, in which each factor was studied independently, the next three sections present the two-way interactions between the three factors. For each pair of factors, we studied the interaction effect of their different levels on the VRI index.

4.2.1 Two-Way Interaction: *Vectorization* * *Cleaning* Effect

The effect of the combination of the two factors *Vectorization***Cleaning* was significant, $F(2.95, 26.546) = 14.791$, $\rho < .001$. To know at which levels the difference of the means were significant, we refer to Table 1. The *Vectorization***Cleaning* contrast tests the hypothesis that the mean of the specified *Cleaning* contrast is the same across the three *Vectorization* levels. Considering the first two levels of *Vectorization* (Vec vs. VP), the fourth contrast (AlgB vs. AD) of *Cleaning* was significant, $F(1, 9) = 10.444$, $\rho < .05$. This indicates that the mean difference in the VRI made between AlgB and AD was not the same across the two levels of *Vectorization* (Vec and VP). The EMM of AD within Vec was considerably lower than the first four levels (Fig. 5), indicating low performance of AD within the Vec software. The performance of AlgC was also low compared to the first four methods; this indicates that the performance of Vec dropped when used with the AD and AlgC noise-removal methods.

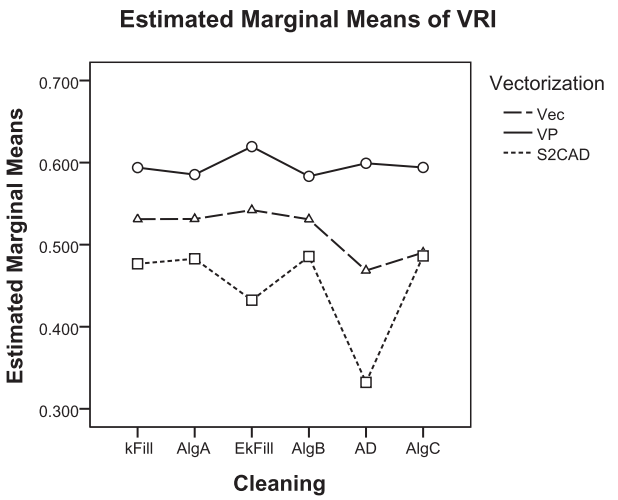

Fig. 5 Two-way interaction between *Vectorization* and *Cleaning*.

Considering the second two levels of *Vectorization* (VP vs. S2CAD), the five contrasts of *Cleaning* were significant, which indicates that the mean differences in the VRI made between kFill vs. AlgA, AlgA vs. EkFill, EkFill vs. AlgB, AlgB vs. AD, and AD vs. AlgC are not the same across the two levels of *Vectorization* (VP and S2CAD). S2CAD had more sensitivity to cleaning methods, which is shown as a sharp oscillation of performance (Fig. 5). The results of this section can be summarized as follows: S2CAD shows poor performance when used with the AD noise-removal method, Vec shows significant poor performance when working with AD and AlgC, and VP shows little sensitivity when working with different noise-removal algorithms. It is thus considered stable and may be used with any of the six noise-removal methods.

4.2.2 Two-Way Interaction: *Vectorization* * *Noise* Effect

The effect of the combination of the two factors *Vectorization***Noise* was significant, $F(3.937, 35.434) = 35.155$, $\rho < .001$. To know at which levels the difference of the means was significant, we refer to Table 1. The *Vectoriza-*

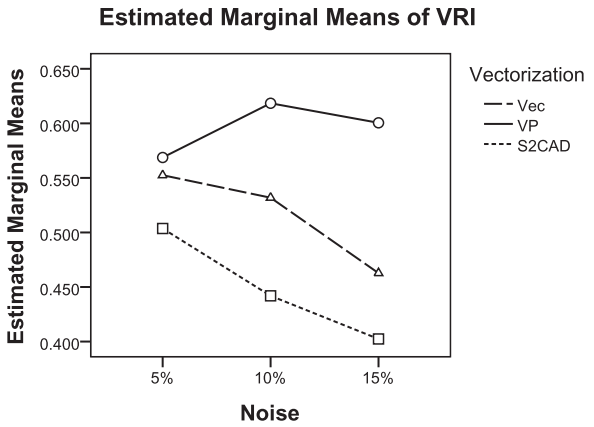

Fig. 6 Two-way interaction between *Vectorization* and *Noise*.

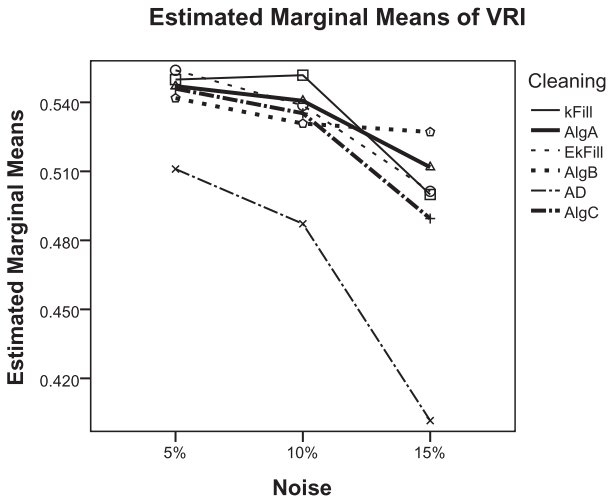

Fig. 7 Two-way interaction between *Cleaning* and *Noise*.

*tion*Noise* contrast tested the hypothesis that the mean of the specified *Noise* contrast is the same across the three *Vectorization* levels. All the four contrasts were significant, indicating that the mean difference of VRI values was not the same in the three software that considered the three noise levels. Intuitively, increasing the noise level caused a drop in VRI values for all tested vectorization software in general. This confirms the results of Sect. 4.1.1, which demonstrated that the three different software have a large difference among them in affecting VRI scores. The performance of the S2CAD dropped sharply when the noise level was increased (Fig. 6). Vec had a moderate drop in performance when the noise level was increased. VP had the best resistance to noise; hence, its performance slightly dropped when the noise level was increased. VP performance at the 15% noise level was better than Vec and S2CAD at the three noise levels. However, VP had unexpected behavior with 5% noise, at which it scored lower than the other two cases (10% and 15%). This unusual case is difficult to interpret because we are dealing with the software as a black box in which the content (such as the raster-to-vector algorithm) is not usually known. A similar unusual behavior has been reported in past studies, such as that performed by Wenyin [12], in which some noisy images obtained higher VRI compared to their original clean images.

4.2.3 Two-Way Interaction: *Cleaning * Noise* Effect

The effect of the combination of the two factors *Cleaning*Noise* was significant, $F(3.471, 31.243) = 5.108$, $\rho < .01$. To know at which levels the differences of the means were significant, we refer to Table 1.

The *Cleaning*Noise* contrast tested the hypothesis that the mean of the specified *Noise* contrast is the same across the six levels of *Cleaning*. Only three contrasts were significant, indicating that the VRI values for the significant contrast were not the same in the two noise levels (10% and 15%) considering the four methods (EkFill, AlgA, AD, and AlgC) of *Cleaning*.

The sixth contrast (10% vs. 15%) was significant, $F(1, 9) = 6.044$, $\rho < .05$, which indicates that the mean difference in VRI between 10% and 15% noise levels was not the same across the two levels of the noise-removal algorithms (EkFill and AlgB). The enhanced counterpart of the noise-removal algorithms performed better with a higher level of noise than its original counterpart (Fig. 7). EkFill suffered a mean difference of $(.539 - .501 = .038)$ when noise level was increased from 10% to 15%, whereas our algorithm (AlgB) suffered a mean difference of only $(.531 - .527 = .004)$ under the same condition.

The eighth contrast (10% vs. 15% noise) was significant, $F(1, 9) = 68.143$, $\rho < .001$, which indicates that the mean difference in VRI between the 10% and 15% noise levels was not the same across the two levels of noise-removal algorithms (AlgB and AD). AlgB performed better with a higher level of noise than AD did (Fig. 7). AlgB suffered a mean difference of only $(.531 - .527 = .004)$ when the noise level was increased from 10% to 15%, whereas AD suffered a mean difference of $(.487 - .402 = .085)$ under the same condition.

The tenth contrast (10% vs. 15%) was significant, $F(1, 9) = 6.546$, $\rho < .05$, which indicates that the mean difference in VRI between the 10% and 15% noise levels was not the same across the two levels of noise-removal algorithms (AD and AlgC). AlgC performed better with a higher level of noise than AD did (Fig. 7). AlgC suffered a mean difference of only $(.535 - .489 = .046)$ when noise level was increased from 10% to 15%, whereas AD suffered a mean difference of $(.487 - .402 = .085)$ under the same condition.

5. Conclusions

Many of the reviewed studies look at the raster-to-vector conversion process as the sole major factor when performing empirical performance evaluation, or they do not study other factors (such as noise removal and noise level) rigorously. In this paper, we have proposed a methodology to study many factors that may have a role in affecting line detection in the raster-to-vector conversion process. The proposed method-

ology, which can study two or more factors simultaneously, was also coupled with a robust statistical analysis method. It is not the aim of this proposed methodology to directly improve existing raster-to-vector method(s). However, the methodology provides a means to utilize the large number of existing methods, whether for noise removal or raster-to-vector conversion. For example, the method could be used to find algorithms from noise removal and from raster-to-vector conversion that can fit with each other in a way that allows the production of high-quality vector data. An experiment was performed to study three independent factors: noise-removal algorithm, noise level, and vectorization software. The interpretation of the output of the statistical analysis shows that the three studied factors affected line detection and that there is an interaction between the studied factors.

In the vectorization factor, the three studied methods showed significant inter-differences. The best performer was VP, followed by Vec. S2CAD was the lowest performer. Concerning the noise-removal factor, AD showed a significant difference in performance when compared to the other five methods, which showed comparable performance among themselves. AD was the lowest performer. Concerning the noise level factor, 15% noise showed a significant difference compared to the other lower noise levels (5% and 10%). The VRI index dropped significantly at 15%.

For the two-way interactions, the vectorization-cleaning interaction was significant, which shows that the quality of line detection for raster-to-vector methods is related to the noise-removal method used in image enhancement. VP was the most stable and had no sensitivity when used with any of the six noise-removal methods. It performed best when used with EkFill. Vec had low performance when used with AD and AlgC but good performance when used with EkFill. S2CAD was more sensitive to noise-removal methods and showed poor performance when used with AD but good performance when used with AlgB/AlgC.

The vectorization and noise level interaction was also significant. The VP method showed stable performance, and its VRI index had a moderate drop when noise was increased. VP performance at 15% noise was better than the performance of Vec and S2CAD at the three noise levels. The Vec and S2CAD methods had sharper drops in performance when the noise was increased.

The significance of the interaction between noise level and noise-removal method is intuitive. All algorithms showed no significant drop in performance when noise level was increased from 5% to 10%. However, increasing the noise from 10% to 15% affected some algorithms as follows: AlgB showed significantly better performance compared to its original counterpart (EkFill). AlgB also performed significantly better than AD. AlgC performed significantly better than its original counterpart (AD).

The proposed methodology of this paper is not limited to studying the factors affecting raster-to-vector conversion. The methodology can also be used in other areas of computer vision, such as OCR. The stages in Fig. 1 could be replaced by other stages of the computer vision process, such as OCR stages. The key point in using the methodology is to run the experiment systematically and in alignment with the requirements of the statistical analysis method.

The future direction of this research includes studying the scanning resolution factor.

Acknowledgements

The first author is a Post-Doctoral Fellow in the School of Computer Sciences/USM. The authors would like to thank Low Heng Chin of the School of Mathematical Sciences/USM for her help with statistical analysis and the two anonymous reviewers of this article for their valuable comments and guidance to enhance the quality of this paper.

References

[1] K. Tombre, "Is graphics recognition an unidentified scientific object?," Graphics Recognition: Recent Advances and New Opportunities, vol.5046, pp.329–334, 2008.

[2] K. Tombre, "Graphics recognition: The last ten years and the next ten years," 6th International Workshop on Graphics Recognition, GREC 2005, Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol.3926 NCS, Hong Kong, China, pp.422–426, Springer Verlag, Heidelberg, D-69121, Germany, 2006.

[3] X. Hilaire and K. Tombre, "Robust and accurate vectorization of line drawings," IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.6, pp.890–904, 2006.

[4] S. Jiqiang, M.R. Lyu, and C. Shijie, "Effective multiresolution arc segmentation: Algorithms and performance evaluation," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.11, p.1491, 2004.

[5] D. Dori, "Orthogonal zig-zag: An algorithm for vectorizing engineering drawings compared with hough transform," Advances in Engineering Software, vol.28, no.1, pp.11–24, 1997.

[6] I.T. Phillips and A.K. Chhabra, "Empirical performance evaluation of graphics recognition systems," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.9, pp.849–870, 1999.

[7] A.K. Chhabra and I.T. Phillips, "Performance evaluation of line drawing recognition systems," Proc. 15th International Conference on Pattern Recognition, pp.864–869, Barcelona, 2000.

[8] W.Y. Liu, J. Zhai, and D. Dori, "Extended summary of the arc segmentation contest," Graphics Recognition: Algorithms and Applications, vol.2390, pp.343–349, 2002.

[9] D. Elliman, "Tif2vec, an algorithm for arc segmentation in engineering drawings," Graphics Recognition Algorithms and Applications, Lect. Notes Comput. Sci., vol.2390, pp.350–358, 2002.

[10] X. Hilaire, "Ranvec and the arc segmentation contest," Graphics Recognition Algorithms and Applications, Lect. Notes Comput. Sci., vol.2390, pp.359–364, Springer Berlin/Heidelberg, 2002.

[11] W. Liu, "Report of the arc segmentation contest," Graphics Recognition, Lect. Notes Comput. Sci., Recent Advances and Perspectives, vol.3088, pp.363–366, Springer, 2004.

[12] L. Wenyin, "The third report of the arc segmentation contest," Lect. Notes Comput. Sci. (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol.3926 NCS, Hong Kong, China, pp.358–361, Springer Verlag, Heidelberg, D-69121, Germany, 2006.

[13] D. Keysers and T.M. Breuel, "Optimal line and arc detection on run-length representations," Graphics Recognition, vol.3926, pp.369–380, 2006.

[14] X. Hilaire, "Ranvec and the arc segmentation contest: Second evaluation," Graphics Recognition, Ten Years Review and Future Per-

spectives, Lect. Notes Comput. Sci., vol.3926, pp.362–368, 2006.
[15] F. Shafait, D. Keysers, and T.M. Breucl, "Grec 2007 arc segmentation contest: Evaluation of four participating algorithms," Graphics Recognition: Recent Advances and New Opportunities, vol.5046, pp.310–320, 2008.
[16] H.S.M. Al-Khaffaf, A.Z. Talib, and R.A. Salam, "A study on the effects of noise level, cleaning method, and vectorization software on the quality of vector data," Graphics Recognition, Recent Advances and New Opportunities, ed. W. Liu, J. Llados, and J.M. Ogier, Lect. Notes Comput. Sci., vol.5046, pp.299–309, Springer Berlin/Heidelberg, 2008.
[17] G.A. Story, L. O'Gorman, D. Fox, L.L. Schaper, and H.V. Jagadish, "The rightpages image-based electronic library for alerting and browsing," Computer, vol.25, no.9, pp.17–26, 1992.
[18] L. O'Gorman, "Image and document processing techniques for the rightpages electronic library system," Proc. 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems, pp.260–263, The Hague, 1992.
[19] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima, "Removing salt-and-pepper noise in text/graphics images," 1998 IEEE Asia-Pacific Conference on Circuits and Systems, pp.459–462, Chiangmai, 1998.
[20] P.Y. Simard and H.S. Malvar, "An efficient binary image activity detector based on connected components," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.229–233, 2004.
[21] H.S.M. Al-Khaffaf, A.Z. Talib, and R.A. Salam, "Removing salt-and-pepper noise from binary images of engineering drawings," 19th International Conference on Pattern Recognition, vol.1-6, pp.1271–1274, 2008. (ICPR 2008) DEC 08-11, Tampa, Florida, USA, 2008.
[22] H.S.M. Al-Khaffaf, A.Z. Talib, and R.A. Salam, "Enhancing salt-and-pepper noise removal in binary images of engineering drawing," IEICE Trans. Inf. & Syst., vol.E92-D, no.4, pp.689–704, April 2009.
[23] H.S.M. Al-Khaffaf, A.Z. Talib, and R.A. Salam, "Salt and pepper noise removal from document images," Visual Informatics: Bridging Research and Practice, LNCS, vol.5857, pp.607–618, 2009.
[24] H.S.M. Al-Khaffaf, A.Z. Talib, M.A. Osman, and P.L. Wong, "Grec'09 arc segmentation contest: Performance evaluation on old documents," Graphics Recognition, Achievements, Challenges, and Evolution, ed. J.M. Ogier, W. Liu, and J. Llados, Lect. Notes Comput. Sci., vol.6020, pp.251–259, Springer Berlin/Heidelberg, 2010.
[25] W.Y. Liu and D. Dori, "Incremental arc segmentation algorithm and its evaluation," IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.4, pp.424–431, 1998.
[26] B. Lamiroy and Y. Guebbas, "Robust and precise circular arc detection," Graphics Recognition. Achievements, Challenges, and Evolution, Lect. Notes Comput. Sci., vol.6020, pp.49–60, Springer Berlin/Heidelberg, 2009.
[27] M.J. Roberts and R. Russo, A Student's Guide to Analysis of Variance, Routledge, 1999.

Hasan S.M. Al-Khaffaf gets his BSc. and MSc. in computer sciences from University of Mosul, Iraq in 1997 and 2000 respectively. In 2002 he started his career as a full time lecturer in the Department of Computer Sciences, Al-Hussein Bin Talal University/Jordan. He get his Ph.D. in Computer Sciences from Universiti Sains Malaysia in 2010. He is a Member of Australian Computer Society (ACS) with the status of Certified Technologist. His research interests include analysis of engineering drawings, raster-to-vector conversion process and its performance evaluation.

Abdullah Z. Talib obtained B.Sc. (Hons.) in Mathematical Sciences from the University of Bradford, Britain in 1983, M.Sc. in Computing Science from the University of Newcastle upon Tyne, Britain in 1985 and Ph.D. in Computer Science from the University of Wales in Swansea, Britain in 1995. He started his career as a university lecturer at the University of Science Malaysia (USM) in 1986 and promoted to the current position as an Associate Professor in 2003. He has also served as chairperson for computer science, information systems and computing science program at the School of Computer Sciences, USM. Currently he is the deputy dean for industry and community network at the same school. His research interests include graphics and visualization, geometric computing, computational modeling and intelligent systems. He has published over 50 papers in conferences and journals, and served as conference organizing chair, member of program committees for many international/national conferences. He has also reviewed several conference and journal papers.

Rosalina Abdul Salam is professor at the Faculty of Science & Technology, Universiti Sains Islam Malaysia. She received her Bachelors degree in Computer Science in 1992 from Leeds Metropolitan University, United Kingdom. She was a system analyst in Intel Penang, from 1992 to 1995. She returned to United Kingdom to further her studies. She received her Masters degree in Software Engineering from Sheffield University, United Kingdom in 1997. She completed her Ph.D. in 2001 from Hull University in the area of computer vision. She has published more than 60 papers in journals and conferences. She is a member of International Computational Intelligence Society and World Enformatika Society. Recently she joined the editorial board of the International Journal of Computational Intelligence and the International Journal of Signal Processing. Presently, she is continuing her teaching, graduate supervisions and her research. Her current research area is in the area of artificial intelligence, image processing and bioinformatics applications. The most recent project that she is working is on underwater images and cellular images.