Available online at sjuoz.uoz.edu.krd

journals.uoz.edu.krd

Science Journal of University of Zakho

Vol. 8, No. 3, pp. 118– 122, September-2020

p-ISSN: 2410-7549
e-ISSN: 2414-6943

A Modified Vector Recovery Index

Hasan S. M. Al-Khaffaf [a]

[a] Dept. of Computer Science, College of Science, University of Duhok, Duhok, Kurdistan Region, Iraq – (hasan.salim@uod.ac)

Received: Aug., 2020 / Accepted: Sep., 2020 / Published: Sep.,2020 ***https://doi.org/10.25271/sjuoz.2020.8.3.755***

ABSTRACT:
In this paper, we show that averaging of the Vector Recovery Index (VRI) score for a test involving many images is not accurate and leads to bias. We demonstrate that the higher the difference in primitive count between the data files in an experiment, the higher the bias in calculating the VRI. Normalizing VRI scores is proposed to remove the bias and to get VRI scores that precisely reflects the performance based on images under scrutiny. Empirical performance evaluation on three datasets from the arc segmentation contests attached to International Workshops on Graphics Recognition 2005, 2009, and 2011 shows that the proposed normalization score provides accurate and realistic performance results than the unweighted average of VRI scores. The results based on the modified VRI score show that the vectorisation methods have lower performance than was usually thought.

KEYWORDS: Vector Recovery Index, Performance Evaluation, Raster to Vector Conversion, Document Images.

1. INTRODUCTION

Many performance evaluation criteria in the area of graphics recognition had been proposed in the literature (Hori and Doermann, 1995; Liu and Dori, 1997; Phillips and Chhabra, 1999; Chhabra and Phillips, 2000; Shafait et al., 2006, 2008). These methods are used to measure the quality of the recognition of line-like shapes as well as text strings. The raster to vector conversion is still a hot topic with many papers published recently (Inoue and Yamasaki, 2019, Popov et al., 2020, Al-Khaffaf and Talib, 2020). Hori and Doermann (1995) presented a quantitative measure for straight line recognition. Liu and Dori (1997) presented the Vector Recovery Index (VRI) as an objective performance evaluation metric for comparing a set of detected vectors with their corresponding set of ground truth vectors and each set from a separate physical file. Many types of graphics primitives are included in the performance matrice including straight lines, arcs, circles. Solid and dashed line primitives are also included. The VRI value combines two matrices, the detection rate, and the false alarm rate. The VRI score is between 0 and 1, where higher is better recognition.

EditCost Index (Phillips and Chhabra, 1999; Chhabra and Phillips, 2000) is designed for graphics recognition systems. It operates on raster images containing different graphical primitives such as straight lines, circles, circular arcs, and text. The EditCost Index value is between 0 and 1. The value represents the amount of editing that the user needs to perform to rectify the inaccuracy in recognition. The lower the value, the less the required editing.

Vectorial score (Shafait et al., 2006, 2008) detect segmentation errors such as over-, under-, and mis-segmentation in page segmentation algorithms. The performance of detecting lines and text components can be measured.

VRI score is popular in graphics recognition and it was the criterion of choice in most of the Arc Segmentation Contests attached to the International Workshops on Graphics Recognition (Liu et al., 2002; Liu, 2004; Wenyin, 2006; Al-Khaffaf et al., 2010, 2013; Bukhari et al., 2014). One reason is attributed to the availability of the software tool by its creator. The aforementioned tool works with vector files in the VEC text format, a very simple file format defined by Chhabra and Phillips (1998). In practice, many test images with their corresponding ground truth data files were used in a bench-marking session. Hence, researchers end up calculating one VRI score for each ground truth and detected file pair. To find the overall performance for a vectorisation method, researchers used the unweighted mean of the VRI scores. To the best of our knowledge, this seems to be the case in all of the research publications (Wang et al., 2010; Bonnici and Camilleri, 2013; Wu et al., 2013; Zhang et al., 2015; Kasimov et al., 2017; Bonnici et al., 2019; Alwan et al., 2019) that use VRI as the preferred performance criterion as well as papers published by the authors of VRI (Liu et al., 2001). However, using the unweighted mean is only accurate if all the images (ground truth images) have the same number of primitives (graphical entities) which is rarely the case. By having images with a different number of primitives we risk to allow the images with a small number of primitives to have a big influence on the overall score. This influence could be in either direction of the performance i.e. either pushing the results towards high performance or dragging the performance down. In this paper, the issue of using VRI is demonstrated through three scenarios. The disadvantage of averaging VRI scores is shown. Then a modified VRI is presented to reduce the bias incurred when working with images of a different number of graphical elements. An experiment is performed to show that modified VRI scores provide more accurate and stable performance scores.

2. DEFINITIONS

Assume $I = \{x_0, x_1, .., x_n\}$ is an ordered set of raster images used in an experiment and $G = \{g_0, g_1, .., g_n\}$ is the corresponding ordered set of ground truth images. Image pair (x_i, g_i) is used to calculate one VRI score.

Within-image primitives: Two or more primitives that belong to one image file x_i.

Between-image primitives: Two or more primitives where any of these primitives belongs to a different image file $\{g_i, g_j \parallel g_i \in x_k \wedge g_j \in x_l \wedge k \neq l\}$, where $\wedge$ means logical AND operator.

Vectorisation method v: A virtual raster to vector conversion method that converts raster images into vector form. The method detects the primitives in the image and saves the attributes of the primitive into a vector file. For the purpose of generality, this method is considered as a black box in the rest of this paper. This assumption helps in studying different vector detection hypotheses while keeping the study general.

Vectorisation pool V: A set of all virtual vectorisation methods $V = \{v_1, v_2, v_3, \ldots\}$

3. WEAKNESS IN THE CURRENT USE OF VRI

Consider a scenario where a researcher studying the performance of method v acquired two image files (x_i and x_j). Image x_i (let's call it A) contains one primitive while image x_j (let's call it B) contains nine primitives. Let's further assume, that these two images are acquired through scanning a page of a paper document. The issue of the current method of calculating the overall performance of raster to vector conversion will be illustrated by relying on three different scenarios. In the three scenarios, we will use image A, image B. Liu and Dori (1997) defined VRI as

$$VRI = \beta D_v + (1-\beta)(1-F_v) \quad (1)$$

where D_v is the detection rate, F_v is the false alarm rate, and β is the trade-off weight between detection rate and false alarm. In order to simplify the presentation of the issue, only the detection rate is assumed and presented. The trade-off parameter β is set to 1 to give full weight to the detection rate, hence canceling the false alarm (F_v) of Eq. 1. However, what we will present regarding the detection rate is also correct for the case of false alarm. Since no false alarms are assumed in the following examples, hence the *VRI* will be reduced to only the first term of the *VRI* equation, i.e. D_v. In the next paragraphs, we are going to present three possible scenarios of performance evaluation on a virtual vectorisation method v. These scenarios are synthetic but likely to happen in real evaluations.

Scenario 1: In the first scenario, a vectorisation method v_1 detects all primitives in both of the test images. The *VRI* score for each image is shown in Table 1.

Table 1: The unweighted mean of image A and B. All primitives of image A and B are detected (Scenario 1).

	VRI	image contr. to overall score	V_n†	V_d‡	v/i* cont. to VRI	v/i* cont. to overall score
Image A	1.000	50%	1	1	100%	50%
Image B	1.000	50%	9	9	11.11%	5.556%
		100%	10	10		100%
$\overline{VRI}$	1.000					

†V_n= number of primitives in the ground truth image
‡V_d= number of detected vectors in detected image
*v/i= vector per image

Both images got a *VRI* score of 1 since all the ten ground truth vectors are detected perfectly. However, a closer look at the contribution of the primitives of both images shows that the single primitive in image A participated by 50% of the overall score while each vector in image B participated with only 5.556% of the overall score. This difference in v/i contribution in between-images is casual and not scientifically justified. In other words, the single primitive in image A participated by 50% of the overall score just because it happens to be the only primitive in image A. Again, the single primitive in image A participated by larger percentage to the overall performance than the primitives of image B just because it's the sole primitive in the first image while the other nine primitives are contained in the second image. The reason for getting this result is because of the use of unweighted mean. By using unweighted mean we are giving similar weights to each image of the experiment and at the same time ignoring the between-images differences in terms of the number of primitives. In other words, using an unweighted mean will give the primitives of the images with a small number of entities higher weight compared with those with a larger number of entities, hence giving them a high participation level in the calculation of overall performance score.

Scenario 2: In this scenario, a vectorisation method v_2 detects all primitives of image B while the single primitive in image A is not detected. The *VRI* score for each image is shown in Table 2.

Table 2: Overall performance based on image A and B relying on unweighted mean (Scenario 2).

	VRI	image contr. to overall score	V_n†	V_d‡	v/i* cont. to VRI	v/i* cont. to overall score
Image A	0	50%	1	0	100%	50%
Image B	1.000	50%	9	9	11.11%	5.556%
		100%	10	9		100%
$\overline{VRI}$	0.500					

In this scenario, the mis-detection of the sole primitive of image A causes the image's *VRI* score to drop from 1 to 0. As with scenario 1 above, this single primitive is still contribute to 50% of the overall score of the VRI ($\overline{VRI}$). When this primitive is not detected, the overall score ($\overline{VRI}$) drops from 1 to .500. This is a flaw in calculating the overall score. In this scenario one single primitive causes the overall score to drop from 1 in the case of full detection of the 10 primitives to only .500 in a case where 9 out of the total 10 primitives are fully detected.

Scenario 3: In the third and last scenario, a vectorisatin method v_3 detects 8 out of 9 primitives of image B. The single primitive in image A is also detected. The *VRI* score for each image is shown in Table 3.

Table 3: Overall performance based on image A and B relying on unweighted mean (Scenario 3).

	VRI	image contr. to overall score	V_n†	V_d‡	v/i* cont. to VRI	v/i* cont. to overall score
Image A	1.000	50%	1	1	100%	50%
Image B	0.889	50%	9	8	11.11%	5.556%
		100%	10	9		100%
$\overline{VRI}$	0.945					

In this scenario, the mis-detection of one primitive of image B causes the image's *VRI* score to drop from 1 to .889. As opposed to scenario 2 above, this single primitive contributes by only 5.555% of the overall score of the VRI. When this primitive is not detected, the overall score dropped from 1 to 0.945. In scenario 2 and scenario 3, only one primitive is not detected. However, the impact of missing one primitive in image A is much higher than the impact of missing one primitive in image B on the overall performance. Again, there is no justified reason for this difference because in each of the two scenarios only one primitive is missed.

4. VRI COMPUTATION AND THE PERFORMANCE EVALUATION PROCESS

The bias occurs because the mathematical mean (unweighted mean) is being used to get the overall *VRI* score ($\overline{VRI}$). Thus, each of the physical images contributes equally to the overall *VRI* score. In other words, a primitive within the physical image with low primitive-count have higher contribution than a primitive within the physical image with high primitive-count. This issue is not desirable because low primitives count is not correlated with higher difficulty in recognizing these primitives. To solve this issue, the weighted mean for each *VRI* score is used to account for the difference in primitive-count.

One way to describe the current usage of VRI is that of each physical image is considered as a separate group. Currently, we are trying to give similar weights to each group (i.e. physical image). However, in practice, the attributes of the physical image are not valuable in calculating VRI. The image dimension, size in bytes, and the number of pixels are not important. The content of the physical image in terms of graphical elements are the most valuable. Different images usually have a different number of graphical elements. Hence, the focus shall be on the graphical elements when calculating VRI. At a minimum, the graphical elements rather than the physical images have to have equal weights. Unfortunately, these two facts are mutually exclusive i.e. we either give the same weights to all physical images or the same weight to primitives.

5. PROPOSED SOLUTION: NORMALIZED VRI

One way to approach an unbiased solution is to rethink the logic of empirical performance evaluation. The presentation of VRI, by its authors, refers to only two images, detected vector image and ground truth image. If we assume that only one logical image of detected primitives needs to be compared against one logical image of ground truth data, then a solution can be realized. Within this logical image of detected vectors, all the physical images under study can be grouped.

The three different scenarios in Section 3 show that using the unweighted mean for calculating the overall performance of a raster to vector method is not accurate when the test images contain a varying number of primitives. The overall performance of a method should be proportional to the total number of primitives of the test images. In other words, the number of images to be used in the experiment should not be considered as a factor or should not contribute to calculating the overall performance. This issue can be reduced by normalizing images' VRI scores. The following formula (Eq. 2) is proposed in this paper to normalize the VRI score for each image i.

$$VRI_i' = \frac{VRI_i \times V_n}{V_N} \quad (2)$$

where i is ground-raster image pair number, V_n is the number of vectors (primitives) in a ground-truth image i, V_N is the total number of vectors (primitives) in all images of the experiment (i.e number of primitives in the logical ground-truth image).

The following formula is then used to find the weighted mean of all the VRI scores.

$$VRI'' = \sum_{i=0}^{Z-1} VRI_i' \quad (3)$$

where Z is the total number of ground-raster image pairs in the experiment.

Using the above equations, we get the normalized scores of the three scenarios as shown in Tables 4, 5, and 6. The overall performance (VRI$''$) of Eq. 3 is not biased to any of the images and all primitives have the same weight in the calculation of the overall performance.

In the scenario of using the normalized VRI scores (VRI''), the difference in primitives factor is ruled out and it will not be an issue when ground-truth images have a different number of primitives.

Table 4: All vectors detected correctly (Scenario 1). The $\overline{VRI}$ and VRI$''$ score values are both 1 in this ideal case.

	VRI	VRI'	V_n	V_d	v/i cont. to VRI"
Image A	1	0.1	1	1	0.10
Image B	1	0.9	9	9	0.10
$\overline{VRI}$	1		10	10	
VRI"		1			

Table 5: The lonely vector in image A is not detected (Scenario 2). The $\overline{VRI}$ score is dropped to 0.5 while VRI$''$ score is deoped from 1 to 0.9.

	VRI	VRI'	V_n	V_d	v/i cont. to VRI"
Image A	0	0	1	0	0.10
Image B	1	0.9	9	9	0.10
$\overline{VRI}$	0.5		10	9	
VRI"		0.9			

Table 6: One vector in image B is not detected (Scenario 3). The miss-detection of one vector causes $\overline{VRI}$ score to drop by 0.056 while VRI$''$ score is droped from 1 to 0.9.

	VRI	VRI'	V_n	V_d	v/i cont. to VRI"
Image A	1	0.1	1	1	0.10
Image B	0.889	0.8	9	8	0.10
$\overline{VRI}$	0.944		10	9	
VRI"		0.9			

In the ideal case of perfect detection of all primitives, it is shown in Table 4 that $\overline{VRI}$ and the VRI$''$ have the value of 1. When detecting 9 out of 10 total primitives it is shown in Table 5 and 6 that VRI$''$ value is not biased to any of the images but gives a uniform result of 0.9 while VRI value is dropped from 1 to 0.5 in scenario 2 (Table 5) due to miss detection of only one primitive and $\overline{VRI}$ is dropped by a reasonable ratio of 0.056 in scenario 3 (Table 6) due to miss detection of only one vector of image B. This illustrates the bias of $\overline{VRI}$ to the images with a smaller number of primitives. This also indicates that the proposed method is more stable and produce the same result no matter the number of primitives in the physical image.

It is shown in Table 5 and 6 that VRI$''$ does not suffer from bias towards images with less graphical elements and all vectors (within- and between-image) will have the same weight. The number of files factor is removed from affecting the overall resutls.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed normalized VRI formula is tested on the results of Arc Segmentation Contests' datasets attached to GREC'05, GREC'09, and GREC'11 (Wenyin, 2006; Al-Khaffaf et al., 2010, 2013). All images in these datasets are binary (mono tone) scanned images of mechanical engineering drawings containing straight lines, circles, and arcs. Only circles and arcs are considered in the experiments.

Table 7: The VRI" and $\overline{VRI}$on GREC'05 dataset.

		Elliman		Keysers		Hilaire	
Image	**V_n**	**VRI**	**VRI'**	**VRI**	**VRI'**	**VRI**	**VRI'**
5	**19**	**0.119**	**0.0306**	**0.591**	**0.1517**	**0.904**	**0.2321**
6	**7**	**0.896**	**0.0848**	**0.796**	**0.0753**	**0.939**	**0.0888**
7	**22**	**0.092**	**0.0274**	**0.268**	**0.0797**	**0.404**	**0.1201**
8	**7**	**0.76**	**0.0719**	**0.729**	**0.0690**	**0.736**	**0.0696**
9	**4**	**0.855**	**0.0462**	**0.611**	**0.0330**	**0.97**	**0.0524**
10	**15**	**0.458**	**0.0928**	**0.614**	**0.1245**	**0.862**	**0.1747**
V_N	**74**						
$\overline{VRI}$		**0.530**		**0.602**		**0.803**	
VRI"			**0.354**		**0.533**		**0.738**

Tables 7, 8, and 9 shows VRI and VRI$''$ scores of the methods under consideration. The VRI scores are taken from the aforementioned studies while VRI$''$ is calculated using the proposed formula.

It is shown from Tables 7, 8, and 9 that VRI" values are different than $\overline{VRI}$ values due to the normalization effect. The VRI” is not biased to images of lower primitives count. It is also shown that VRI" scores are usually smaller than $\overline{VRI}$ scores indicating vectorisation mehtods produce a lower vector quality than is actually thought of VRI scores. In general, the VRI$''$ scores are not necessarily smaller than $\overline{VRI}$, but it depends on the empirical results of the methods and the dataset.

Table 8: The VRI” and $\overline{VRI}$on GREC’09 dataset.

		Scan2CAD		Vectory		VPStudio		VrLiu		Qgar-Lamiroy	
Image	V_n	VRI	VRI'	VRI	VRI'	VRI	VRI'	VRI	VRI'	VRI	VRI'
P0036	11	0.322	0.0224	0.162	0.011	0.593	0.041	0.726	0.051	0.634	0.044
P0093	26	0.444	0.0731	0.54	0.089	0.519	0.085	0.371	0.061	0.322	0.053
P0096	49	0.411	0.1275	0.004	0.001	0.641	0.199	0.608	0.189	0.487	0.151
P0168	23	0.391	0.0569	0.526	0.077	0.534	0.078	0.419	0.061	0.405	0.059
P0169	23	0.488	0.0710	0.384	0.056	0.563	0.082	0.772	0.112	0.49	0.071
P0537a	16	0.403	0.0408	0.063	0.006	0.381	0.039	0.803	0.081	0.539	0.055
P1099	10	0.545	0.0345	0.211	0.013	0.795	0.050	0.736	0.047	0.657	0.042
V_N	158										
$\overline{VRI}$		0.429		0.270		0.575		0.634		0.505	
VRI"			0.426		0.254		0.574		0.601		0.475

Table 9: The VRI” and $\overline{VRI}$on GREC’11 dataset.

		Liu’ s		ArcFind		EAS		Qgar-Lamiroy	
Image	Pn	VRI	VRI'	VRI	VRI'	VRI	VRI'	VRI	VRI'
P061	22	0.798	0.087	0.149	0.016	0.31	0.034	0.49	0.054
P168	18	0.738	0.066	0.237	0.021	0.081	0.007	0.127	0.011
P229	13	0.565	0.037	0.314	0.020	0.245	0.016	0.524	0.034
P234	21	0.386	0.040	0.085	0.009	0.343	0.036	0.235	0.025
P238	14	0.24	0.017	0.105	0.007	0.119	0.008	0.311	0.022
P253	13	0.846	0.055	0.065	0.004	0.157	0.010	0.572	0.037
P254	33	0.501	0.082	0.174	0.029	0.123	0.020	0.26	0.043
P260A	24	0.507	0.061	0.078	0.009	0.083	0.010	0.207	0.025
P260B	43	0.424	0.091	0.038	0.008	0.198	0.042	0.193	0.041
PN	201								
$\overline{VRI}$		0.556		0.138		0.184		0.324	
VRI"			0.535		0.124		0.184		0.291

7. CONCLUSIONS

It is shown in this paper that averaging of VRI scores leads to bias. Normalizing the VRI scores is proposed in this paper to remove such bias. Experimental results showed that normalized VRI is not affected by differences in primitive-count between images and that averaging the VRI scores of an experiment involving many test images produces a biased score. It is also found that the quality of vectorisation methods relying on averaging of VRI scores is worse than it usually is due to the bias in averaging many VRI scores. Hence, the proposed normalized VRI score is superior and more accurate in presenting the quality of raster to vector conversion systems.

ACKNOWLEDGEMENTS

The author would like to thank the two anonymous reviewers for their valuable comments that helped in improving the quality of this manuscript.

REFERENCES

Al-Khaffaf, H. S. M., Talib, A. Z. (2020). "Three-stage Junction Detection in Document Images," 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2020, pp. 142-145, doi: 10.1109/CSASE48920.2020.9142083.

Al-Khaffaf, H. S. M., Talib, A. Z., and Osman, M. A. (2013). Final report of GREC’11 arc segmentation contest: Performance evaluation on multi-resolution scanned documents. In Kwon, Y.-B. and Ogier, J.-M., editors, Graphics Recognition. New Trends and Challenges, pages 187–197, Berlin, Heidelberg. Springer Berlin Heidelberg.

Al-Khaffaf, H. S. M., Talib A. Z., Osman M. A., Wong P. L. (2010) GREC’09 Arc Segmentation Contest: Performance Evaluation on Old Documents. In: Ogier JM., Liu W., Lladós J. (eds) Graphics Recognition. Achievements, Challenges, and Evolution. GREC 2009. Lecture Notes in Computer Science, vol 6020. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13728-0_23

Alwan, S., Le Caillec, J.-M., and Le Meur, G. (2019). Detection of Primitives in Engineering Drawing Using Genetic Algorithm. In ICPRAM 2019 : 8th International Conference on Pattern Recognition .Applications and Methods, Prague, Czech Republic.

Bonnici, A., Akman, A., Calleja, G., Camilleri, K. P., Fehling, P., Ferreira, A., Hermuth, F., Israel, J. H., Landwehr, T., Liu, J., and et al. (2019). Sketch-based interaction and modeling:

where do we stand? Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 33(4):370-388.

Bonnici, A. and Camilleri, K. (2013). A circle-based vectorization algorithm for drawings with shadows. In Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling, SBIM:13, page 69:77, New York, NY, USA. Association for Computing Machinery.

Bukhari, S. S., Al-Khaffaf, H. S. M., Shafait, F., Osman, M. A., Talib, A. Z., and Breuel, T. M. (2014). Final report of grec'13 arc and line segmentation contest. In Lamiroy, B. and Ogier, J.-M., editors, Graphics Recognition. Current Trends and Challenges, pages 234–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chhabra, A. K. and Phillips, I. T. (1998). The second international graphics recognition contest - raster to vector conversion: A report. Graphics Recognition, 1389:390–410.

Chhabra, A. K. and Phillips, I. T. (2000). Performance evaluation of line drawing recognition systems. In Proceedings of the 15th International Conference on Pattern Recognition, volume 4, pages 864–869, Barcelona, Spain.

Hori, O. and Doermann, D. (1995). Quantitative measurement of the performance of raster-to-vector conversion algorithms. volume 1072 of LNCS, pages 57–68.

Inoue, N. and Yamasaki, T. (2019). "Fast Instance Segmentation for Line Drawing Vectorization," 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, Singapore, pp. 262-265, doi: 10.1109/BigMM.2019.00-14.

Kasimov, D., Kuchuganov, A., and Kuchuganov, V. (2017). Vectorization of raster mechanical drawings on the base of ternary segmentation and soft computing. Program Comput Soft, 43:337-344.

Liu, W. (2004). Report of the arc segmentation contest. In Graphics Recognition: Lecture Notes in Computer Science: Recent Advances and Perspectives, volume 3088, pages 363–366. Springer.

Liu, W., Zhai, J., Dori, D., and Tang, L. (2001). A system for performance evaluation of arc segmentation algorithms. In Proc. Third CVPR Workshop Empirical Evaluation Methods in Computer Vision.

Liu, W. Y. and Dori, D. (1997). A protocol for performance evaluation of line detection algorithms. Machine Vision and Applications, 9(5-6):240–250.

Liu, W. Y., Zhai, J., and Dori, D. (2002). Extended summary of the arc segmentation contest. In Graphics Recognition: Algorithms and Applications, volume 2390 of Lecture Notes in Computer Science, pages 343–349.

Phillips, I. T. and Chhabra, A. K. (1999). Empirical performance evaluation of graphics recognition systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(9):849–870.

Popov, S., Glazunov, V., Chuvatov, M., and Purii, A. (2020). "Raster to Vector Map Convertion by Irregular Grid of Heights," 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, pp. 386-391, doi: 10.23919/FRUCT48808.2020.9087552.

Shafait, F., Keysers, D., and Breuel, T. M. (2006). Pixel-accurate representation and evaluation of page segmentation in document images. In Proceedings of the 18th International Conference on Pattern Recognition, volume 1, pages 872–875, Hong Kong, China.

Shafait, F., Keysers, D., and Breuel, T. M. (2008). Performance evaluation and benchmarking of six-page segmentation algorithms. Ieee Transactions on Pattern Analysis and Machine Intelligence, 30(6):941–954. 286UW.

Wang, Y., Song, X., and Wang, S. (2010). Algorithm of arcs recognition based on bar tracking. In 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, volume 6, pages 2535–2540.

Wenyin, L. (2006). The third report of the arc segmentation contest. In Lecture Notes in Computer Science, volume 3926 NCS of Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 358–361, Hong Kong, China. Springer Verlag, Heidelberg.

Wu, J., Chen, K., and Gao, X. (2013). Fast and accurate circle detection using gradient-direction-based segmentation. J. Opt. Soc. Am. A, 30(6):1184–1192.

Zhang, Z., Wang, X., Han, K., and Jiang, Z. L. (2015). A novel arc segmentation approach for document image processing. International Journal of Pattern Recognition and Artificial Intelligence, 29(01):1553001.