2013 12th International Conference on Document Analysis and Recognition

An objective method to evaluate stroke-width measures for binarized documents

Marte A. Ramírez-Ortegón
and Volker Märgner
Institüt für Nachrichtentechnik
Technische Universität Braunschweig
Schleinitzstrae 22, 38106,
Braunschweig, Germany.
Email: mars.sasha@gmail.com,
maergner@ifn.ing.tu-bs.de

Raúl Rojas
Institüt für Informatik
Freie Universität Berlin.
Takustr. 7, 14195
Berlin, Germany.
Email: rojas@inf.fu-berlin.de

Erik Cuevas
Departamento de Ciencias Computacionales
Universidad de Guadalajara.
Av. Revolución 1500,
Guadalajara, Jalisco, Mexico.
Email: erik.cuevas@cucei.udg.mx

***Abstract*—In this article, we propose an objective method to evaluate stroke-width measures. With this aim, we discuss the relevance of features based on the stroke width for document analysis. Then, we point out that most of the consulted references have a vague definition of stroke width. Because of this, we propose a formal definition of the stroke-width and remark the linearity of the stroke-width as an important property. Inspired by these ideas, we propose a measure together with a dataset to evaluate the linearity of the measurements of the stroke width and conduct an evaluation for seven well-known stroke-width methods. Our experiments have interesting results, like the fact that the most popular method is the one with the worst performance and that the best method is the easiest to implement. We hope that our objective evaluation assists further authors to choose suitable stroke-width methods for their applications.**

I. Introduction

The efforts on document analysis to recognize and classify documents have lead to develop complex systems that are capable of dealing with problems such as irregular illumination [1], background patterns [2], and handwritten documents [3] to mention some. These systems exploit diverse pixel features extracted not only from the color/gray space, but also from the structural relation between pixels. In particular, the stroke width is one of these structural features.

As a feature, the stroke width is useful to tune parameters, to set constraint criteria, and to characterize objects of interest. Some examples: As a parameter tuner, the mode and maximum of the stroke width have been used to set the size of the pixel neighborhood in binarization algorithms [4], [5], [6]; as a constraint criteria, the mode of the stroke width has been used to identify touching text characters [7], [8]; and as an object attribute, the empirical distribution of the stroke width has been used to segment and classify strokes [3], [9].

Intuitively, in handwritten letters, the stroke width is the nib diameter of the quill (ballpoint pen). Under this definition, Brink et al. [9] identify three factors that influence the width variations of a stroke: the writing instrument, the habitual tip angle, and the individual movement style. These factors cause that the stroke traces are not constant even for the same character stroke. In similar manner, printed characters are more likely to have no constant stroke width due to stylish design or stroke overlapping. So, this arises a question: What does the stroke width mean for strokes with irregular traces?

To answer our previous question, the stroke width should be well-defined at any instant of the trace. However, we noticed that most of our consulted references have an ambiguous definition of the stroke width, such as in [2], [6], [8]. This lack of formalism happen in some applications because the stroke width is considered as an auxiliary information rather than the primordial goal. Moreover, to the best of our knowledge, no publication has been devoted to describe and compare the methods to estimate the stroke width. This motivates us to describe and propose a technique to evaluate different methods that estimate the stroke width.

In Section II, we address the stroke-width definition describing the different interpretations of stroke width and how the digitization affect the theoretical values. Subsequently, in Section III, we explain a desirable property for stroke-width methods and, then, we propose a method to evaluate the stroke-width methods. Later on, we describe our dataset, the evaluated stroke-width methods, and the experiments in Section III-A, Section III-B, and Section III-C, respectively. Our conclusions are given in Section IV.

Our article is based on 63 selected articles from more than 1000 articles that pop up during our searching queries. Unfortunately, due to the page limitations, we have cited only the most important articles in the context of this article. Nevertheless, readers can found our citation database in [10].

II. Definition of stroke width

In our introduction we mentioned that, intuitively, the stroke width is the nib diameter of the quill (ballpoint pen), but the stroke trace may fluctuate due to several factors. Hence, the stroke width has a distribution rather than being a constant value. For example, in Fig. 1, the trace is wide when it is moved perpendicularly and thin when it is moved sideways.

Among all our consulted references, only Nefedov [11] and Brink et al. [9] proposed models on how the stroke width is generated. Nefedov assumed that the nib diameter is constant

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.42

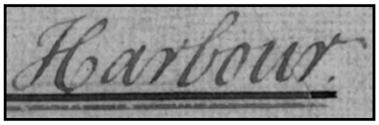
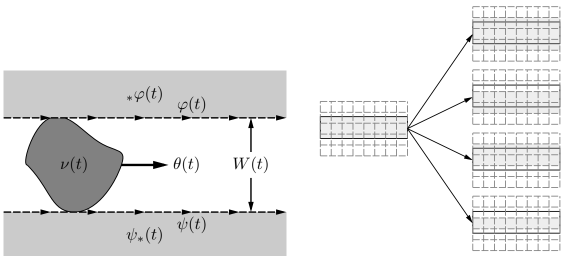


Fig. 1. Handwritten example from a historical document.



(a) (b)

Fig. 2. a) Definition of stroke width: The nib area at the moment t is given by $\nu(t)$, the trace orientation is given by $\theta(t)$, and the stroke width is given by the distance between the oriented lines $\varphi(t)$ and $\psi(t)$. b) The digitization of a horizontal line causes that the observed stroke width differs of the actual stroke width.

throughout the writing and that wider segments are result of more than one pass of the pen (merged strokes or junctions). However, Nefetov's model is unsuitable for documents images as in Fig. 1 which exhibits stroke-width variations in a single trace pass. On the other hand, Brink et al. [9] proposed two stroke-width models for writer identification where the stroke width depends on the nib shape (rectangular and ellipsoidal), the nib orientation, and the local trace direction. In this manner, Brink's models deal with irregular-width traces. However, in these models the nib shape holds constant (different orientations may generate different stroke widths).

Inspired on Brink's models we propose a definition for the stroke width, which is independent of the nib shape.

Definition 1: Let $\nu(t)$ be the set of points contained by the nib at the time $t \in [0, l]$ where $\nu(0)$ and $\nu(l)$ correspond to the initial and final position of the nib, and let $\theta(t)$ be the direction vector of the nib trajectory at the time t. Then the stroke width at the time t is defined as

$$W(t) = ||(\varphi(t), \psi(t))||, \tag{1}$$

where $||\cdot,\cdot||$ denotes the Euclidian distance between two lines, $\varphi(t)$ and $\psi(t)$ are two oriented lines that satisfy:
1) $\varphi(t)$ and $\psi(t)$ are parallel to $\theta(t)$ and oriented as $\theta(t)$.
2) $\varphi(t) \cap \nu(t) \neq \emptyset$ and $\psi(t) \cap \nu(t) \neq \emptyset$.
3) The left semi-plane $_*\varphi(t)$ defined by $\varphi(t)$ satisfies $_*\varphi(t) \cap \nu(t) = \emptyset$.
4) The right semi-plane $\psi_*(t)$ defined by $\psi(t)$ satisfies $\psi_*(t) \cap \nu(t) = \emptyset$.
Figure 2(a) illustrates our definition.

The digitization process arises another problem to measure the stroke width. For instance, Fig. 2(b) shows a horizontal line whose stroke width is around two pixels. However, the binary representation may be a line whose stroke width is one, or two, or three pixels depending on how the binarization is done. Conversely, if we observed a line with one pixel width, then this line may come from a stroke whose stroke width range between zero and three pixels width. This example illustrates that the construction of groundtruth for the evaluation of stroke-width methods is not straightforward even when the groundtruth's construction model is exact.

III. A method to evaluate stroke-width methods

As we discussed in the last section, a measurement of the stroke width from a binary image has more than one solution so that two different methods may estimate two different stroke widths, and both methods may be potentially right. Hence, to overcome this ambiguity, we propose an evaluation method that avoids the direct comparison of the stroke width accuracy, but evaluates a desirable property.

Most of the consulted references assume that the stroke width is linear at scaled resolutions: If w is the stroke width of a stroke $\mathcal{S}$ at resolution x, then $k \cdot w$ must be the stroke width of $\mathcal{S}$ at resolution $k \cdot x$. In this context, resolution refers to the number of pixels per distance unit.

The linearity of the stroke width is implicitly used to set other parameters or constraint criteria. For example, in [1], a binarization method based on water flow model is proposed. A crucial parameter of this method is the rate of rainfall $\lambda = k \cdot w$, where k is a constant manually tuned. Notice that if a stroke-width method attains the half value of the actual stroke-width, then k could be replaced by $k' = 2k$ so that λ obtains the corresponding λ's at each resolution. Other similar examples can be found for binary noise removal rules in [2] and for pixel neighborhood size in [6], [7].

To define our measure, let us assume that $\mathcal{S}_i$ for $i = 1, \ldots, n$ is a binarized stroke at resolution x_i such that $x_i = i \cdot x_1$. This is, $\mathcal{S}_i$ is i-times bigger than $\mathcal{S}_1$. Let $\widehat{w}_i$ denotes the stroke width estimated by a method M; see Fig. 3(a). Then, the absolute residual errors of a linear regression are defined as

$$r_i = |\widehat{w}_i - (x_i \cdot m + b)| \tag{2}$$

where m and b are the estimated slope and intersect terms of a linear regression by the least square method. Notice that the $r_i's$ magnitudes depend on the magnitude of m: the larger m, the larger the expected r_i. Then, comparing the residuals of different methods with different slopes may lead to wrong conclusions. For example, Fig. 3(b) shows two linear regressions of two stroke-width methods M and M' from $\widehat{w}_i$ and $\widehat{w}'_i$ for $i = 1, 2, \ldots$, respectively. Observe the residuals

$$r_2 \;=\; |\widehat{w}_2 - (x_2 \cdot m + b)| = 0.8 \text{ (dotted line) and} \tag{3}$$
$$r'_2 \;=\; |\widehat{w}'_2 - (x_2 \cdot m' + b')| = 0.6 \text{ (dashed line).} \tag{4}$$

Thus, we could interpreted that M' is better than M since $r_2 > r'_2$. However, notice that $\widehat{w}'_2 > \widehat{w}'_3$, which implies that M' confuses $\mathcal{S}_2$ with $\mathcal{S}_3$. On the contrary, M is consistent for $\mathcal{S}_2$ since $\widehat{w}_2$ is closer to $x_2 \cdot m + b$ (predicted resolution) than to $x_1 \cdot m + b$ (predecessor resolution) or to $x_3 \cdot m + b$ (successor resolution). We think that the relative position between predecessor or successor resolutions is more important

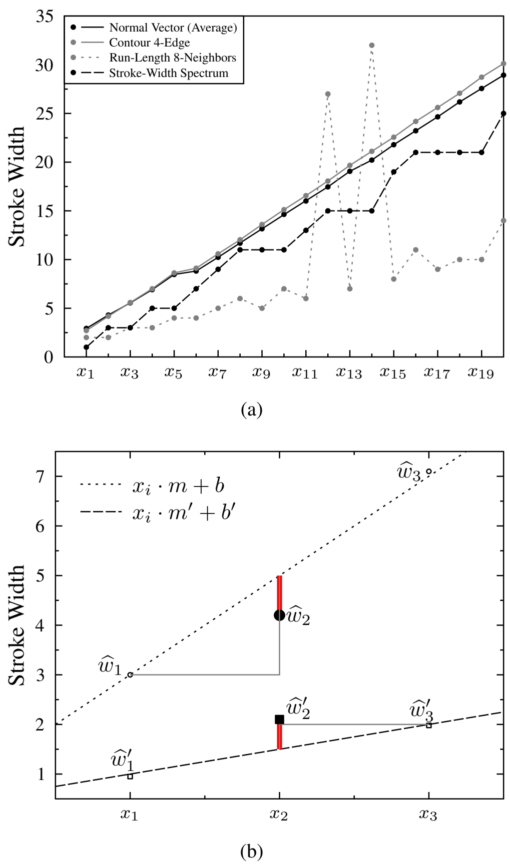


Fig. 3. a) Stroke-width measurements ($\widehat{w}_i$) from 20 images of α at different resolutions. Each line represents the measurements of a stroke-width methods. b) Linear regression of two stroke-width methods.

than the absolute error magnitude because the magnitudes can be scaled, as we mentioned for the λ's example. Hence, we measure the linearity as the ratio of the residuals to the corresponding slope:

$$e_i = \frac{r_i}{m}. \tag{5}$$

The smaller the e_i, the better; when e_i is close to one, then the stroke-width method is likely to confuse predecessor or successor resolutions.

A. Our dataset

The dataset is formed by a total of 760 binary images from 38 Greek letters, where black pixels are foreground and white pixels are background. The selected letters are listed in Table I.

For each Greek letter G_i for $i = 1, \ldots 38$, we generated 20 images $B_{i,j}$ for $j = 1, \ldots 20$ at different resolutions so that the image $B_{i,j}$ is j-times bigger than $B_{i,1}$.

All images were generated by GLE[1] software ©. GLE is a software that generates vectorial images capable of exporting to different formats like png and jpg; the parameter resolution in its export options specifies the bitmap fallback resolution. We adjusted this parameter so that we obtained successive incremented sizes of the same letter. This is equivalent to scanning the same document at different resolutions. However scanning is time-consuming task and it introduces the problem of binarizing the document. Our dataset is available in [10]. .

[1] http://glx.sourceforge.net/

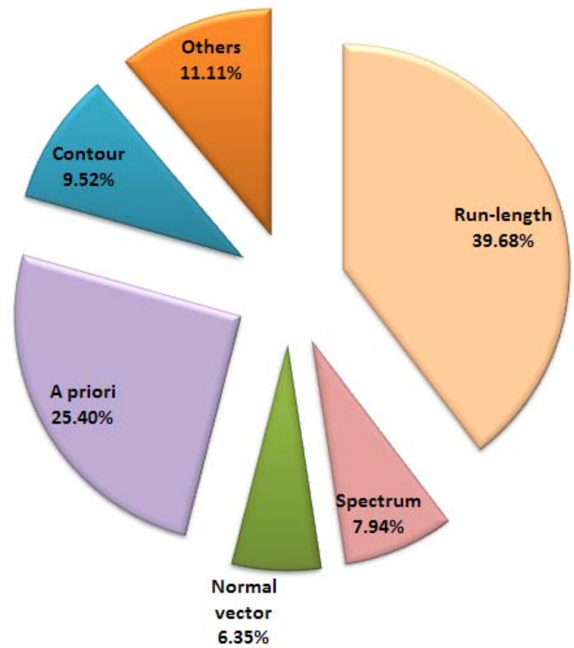


Fig. 4. Distribution on how 63 consulted articles estimate the stroke width. We found that 25.4% of them assume that the stroke-width is known a priori.

B. Methods to estimate the stroke width

After an extensive review of 63 publications (mainly article from journals) that consider the stroke width as a feature, we identified four popular techniques to estimate the stroke width; see Fig. 4.

1) **Run-length**: These methods construct the histogram of run-length sizes for some certain directions. A run-length is a path of consecutive pixels $\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n, \boldsymbol{p}_{n+1}$ in a certain direction so that $\boldsymbol{p}_0$ and $\boldsymbol{p}_{n+1}$ are background pixels and $\boldsymbol{p}_i$ for $i = 1, \ldots n$ are foreground pixels. The size of a run-length path is the number of foreground pixels (n) in its path. Usually, horizontal and vertical directions (4-neighbors) are computed and, in some cases, diagonals (8-neighbors) are also computed. Then, the stroke width is computed as the mode [4], [7], [8], or as the average [1], or as the median [2] of the histogram. We implemented the mode of the histograms from 4-neighbors and 8-neighbors.

2) **Normal vectors**: These methods construct the histogram of the minimum distance between edge pixels that are located on opposite contour sides of a stroke. Some implementations are based on tangential vector angle [9], tensor voting [12], and Delaunay triangulation [13]. To compute the normal vector, we implemented Brink's method [9] whose parameter r was set

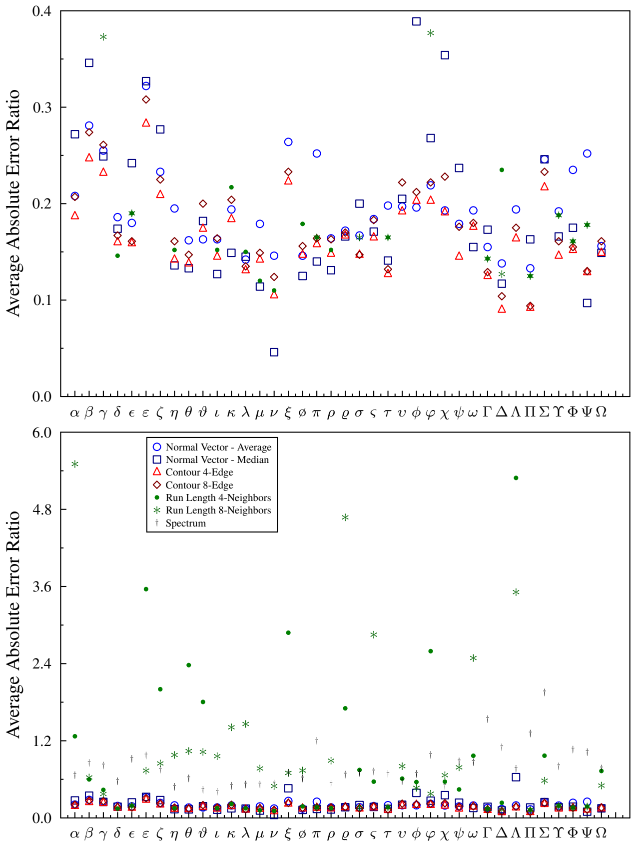


Fig. 5. Details of average of the absolute error ratio ($\bar{e}_{k,i}$) for all 38 Greek letters.

to 2. Then, the stroke width is estimated in two manners: as the average and as the median.

3) **Contour proportion**: These methods compute a ratio between the total of foreground pixels in a stroke (connected component) and the contour pixels. Depending on the method, the definition of contour pixel may change. For example, in [3], the contour pixels are background edge pixels that are externally surrounding the stroke. On the contrary, in [1], the contour pixels are defined as the foreground edge pixels. We followed [1] implementation but we evaluated the method with two different neighborhoods: 4-neighbors and 8-neighbors.

4) **Stroke-width spectrum**: Unlike the previous approaches, we did not find variants of this method but several publications use the same method; [6], [14], [15]. It computes the stroke-width spectrum levels s_i for $i = 1, 3, 5 \ldots, n$. The stroke width is the index i that maximizes $i \cdot s_i / s_1$, where s_i is the number of squares of size $i \times i$ such that p percent of its pixels are foreground. We set $p = 0.9$.

C. Experiments

For our experiment, we evaluated seven methods: 1) Run-lenght (4-neighbors), 2) Run-length (8-neighbors), 3) Normal vector (average), 4) Normal vector (median), 5) Contour proportion (4-neighbors), 6) Contour proportion (8-neighbors), and 7) stroke-width spectrum.

Let $w_{k,i,j}$ denotes the stroke width computed by the k^{th} stroke-width method for the Greek letter G_i at the resolution x_j. Let $e_{k,i,j}$ be the ratio of the residuals to the corresponding slope:

$$e_{k,i,j} = |w_{k,i,j} - x_j * m_{k,i} + b_{k,i}| / m_{k,i}, \qquad (6)$$

where $m_{k,i}$ and $b_{k,i}$ are the slope and intercept terms of the linear regression for the points $(x_j, w_{k,i,j})$ for $j = 1, \ldots, 20$; see Fig. 3(a). Then, we computed the average of the absolute error ratio as:

$$\bar{e}_{k,i} = \frac{1}{20} \sum_{j=1}^{20} e_{k,i,j}, \qquad (7)$$

The values $\bar{e}_{k,i}$ are shown in Fig. 5 and in Table I.

According our results, the stroke-width methods based on the contour proportion performed the best, but closely followed by the methods based on normal vectors. In fact, the averages of both approaches are small enough to consider that such methods do not confuse predecessor or successor resolutions. On the other hand, the worst average performances are observed on the run-length methods, which are the most populars among the consulted references.

Note that our measure is empirically consistent with some expected results. For example, methods based on run-length tend to perform better for letters where straight lines are dominant and tend to perform the worst for letters where curves are dominant; see Δ and ω in Table I.

IV. Conclusions

We addressed the problem of estimating the stroke-width from binary images. We briefly reviewed the applications of features based on the stroke width for document analysis. We pointed out that the stroke-width definition is frequently vague. Because of this, we proposed a formal definition for stroke width. We also discussed the ambiguity to infer the actual stroke width from binary images.

Later on, we proposed an objective measure together with a dataset to test the linearity of stroke width measurements. It allowed us to compare different stroke-width methods head-to-head. Our results showed that the contour proportion and normal vectors approaches are robust to resolution changes, while run-length and stroke-width spectrum are not.

Although the methods based on contour proportion performed the best, we should notice that such methods compute a single value for each connected component and, consequently, the distribution of stroke width is not retrieved. On the contrary, methods based on the normal vector and run-length computes a distribution from which, potentially, more information can be extracted.

Our evaluation is limited to test the linearity of the stroke-width methods. So, more measures should be developed in order to test the accuracy of the stroke-width methods compared with the actual stroke width. This implies the construction of groundtruth, in which case, our Definition 1 and our dataset (Section III-A) provide a starting point.

TABLE I

DETAILS OF THE AVERAGE OF THE ABSOLUTE ERROR RATIO ($\bar{e}_{k,i}$) FOR EACH LETTER THE VALUES ARE GIVEN IN PERCENT AND THE BEST RESULTS ARE PRINTED IN BOLD.

	Contour Prop.		Normal Vector		Run-Length		S.W.
Letter	4-N.	8-N.	Ave.	Med.	4-N.	8-N.	Spec.
α	**18.69**	20.75	20.83	22.54	127.00	550.60	66.38
β	**24.69**	27.44	28.08	38.52	59.95	62.95	85.45
γ	**23.23**	26.11	25.47	27.05	43.63	37.29	81.46
δ	15.97	16.74	18.57	17.95	**14.61**	863.27	57.00
ϵ	**15.91**	16.12	18.02	32.45	19.01	19.01	91.79
ε	**28.31**	30.85	32.16	43.70	355.91	73.64	97.09
ζ	**20.89**	22.50	23.30	28.53	200.20	84.82	75.01
η	14.24	16.07	19.50	**12.14**	15.23	97.95	48.26
θ	**13.81**	14.74	16.25	16.96	237.72	104.07	61.54
ϑ	17.44	19.96	**16.25**	18.27	180.43	102.69	43.62
ι	14.46	16.41	16.30	**12.05**	15.23	96.08	40.50
κ	**18.43**	20.40	19.43	19.08	21.68	141.02	49.76
λ	13.09	13.52	14.19	**11.45**	15.02	146.22	51.92
μ	14.17	14.91	17.95	**10.68**	12.00	77.03	51.90
ν	10.53	12.35	14.56	**8.07**	11.04	49.57	53.55
ξ	**22.30**	23.29	26.41	43.24	288.13	69.98	70.38
o	14.66	15.57	14.59	**12.49**	17.87	73.77	61.31
π	15.85	16.43	25.19	**14.29**	16.47	16.47	119.80
ρ	14.82	16.32	16.44	**10.55**	15.23	89.06	52.84
ϱ	16.67	17.03	17.18	**15.53**	170.51	467.59	67.43
σ	14.73	**14.70**	16.68	17.79	74.61	16.47	70.00
ς	**16.49**	18.34	18.40	22.51	56.37	285.06	72.16
τ	**12.65**	13.23	19.83	13.55	16.47	16.47	68.57
υ	**19.22**	22.17	19.66	27.34	61.11	80.49	57.10
ϕ	20.33	21.15	**19.58**	32.53	55.82	46.83	68.30
φ	**20.30**	22.21	21.93	27.62	259.36	37.73	98.00
χ	**19.15**	22.85	19.31	48.83	56.44	66.51	49.66
ψ	**14.52**	17.59	17.85	23.31	44.31	78.69	87.88
ω	**17.56**	18.01	19.26	21.46	96.67	248.80	86.60
Γ	**12.46**	12.95	15.51	13.86	14.35	14.35	153.61
Δ	**9.03**	10.41	13.80	11.58	23.52	12.71	109.80
Λ	**16.40**	17.49	19.36	31.89	528.93	351.28	76.37
Π	**9.23**	9.43	13.25	14.09	12.47	12.47	131.27
Σ	**21.70**	23.31	24.60	23.08	96.77	58.01	195.19
Υ	**14.57**	16.10	19.23	15.20	18.83	18.83	80.23
Φ	**15.24**	15.55	23.47	18.96	16.08	16.08	106.27
Ψ	12.88	13.01	25.21	**10.82**	17.84	17.84	102.25
Ω	**14.87**	16.13	15.65	15.38	72.95	50.44	76.92
Mean	**16.57**	17.95	19.56	21.19	88.42	122.43	79.40
Median	**15.88**	16.58	19.24	18.11	43.97	71.81	71.27
Std. Dev.	**4.13**	4.67	4.31	10.28	114.99	173.94	31.73

REFERENCES

[1] M. Valizadeh and E. Kabir, "An adaptive water flow model for binarization of degraded document images," *International Journal on Document Analysis and Recognition*, vol. —, pp. 1–12, 2012.

[2] M. Haji, T. D. Bui, and C. Y. Suen, "Removal of noise patterns in handwritten images using expectation maximization and fuzzy inference systems," *Pattern Recognition*, vol. 45, no. 12, pp. 4237 – 4249, 2012.

[3] X. Peng, S. Setlur, V. Govindaraju, and S. Ramachandrula, "Using a boosted tree classifier for text segmentation in hand-annotated documents," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 943 – 950, 2012, special Issue on Awards from ICPR 2010.

[4] S. Lu, B. Su, and C. L. Tan, "Document image binarization using background estimation and stroke edges," *International Journal on Document Analysis and Recognition*, vol. 13, pp. 303–314, December 2010.

[5] V. Vonikakis, I. Andreadis, and N. Papamarkos, "Robust document binarization with off center-surround cells," *Pattern Analysis & Applications*, vol. 14, pp. 219–234, 2011.

[6] R. F. Moghaddam and M. Cheriet, "AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization," *Pattern Recognition*, vol. 46, no. 6, p. 24192431, June 2012.

[7] P. P. Roy, U. Pal, J. Llads, and M. Delalandre, "Multi-oriented touching text character segmentation in graphical documents using dynamic programming," *Pattern Recognition*, vol. 45, no. 5, pp. 1972 – 1983, 2012.

[8] H. Lee and B. Verma, "Binary segmentation algorithm for english cursive handwriting recognition," *Pattern Recognition*, vol. 45, no. 4, pp. 1306 – 1317, 2012.

[9] A. Brink, J. Smit, M. Bulacu, and L. Schomaker, "Writer identification using directional ink-trace width measurements," *Pattern Recognition*, vol. 45, no. 1, pp. 162 – 171, 2012.

[10] [Online]. Available: https://sites.google.com/site/martehomepage/projects/stroke-width

[11] A. Nefedov, "Classification of multiple-stroke junctions in handwriting recognition," *Pattern Recognition and Image Analysis*, vol. 16, pp. 1–4, 2006.

[12] H. P. Le, T. N. Dinh, S. H. Kim, and G. S. Lee, "Text detection in binarized text images of korean signboard by stroke width feature," in *2010 10th IEEE International Conference on Computer and Information Technology*, 2010, pp. 1588 – 1592.

[13] Y. Xiao and H. Yan, "Location of title and author regions in document images based on the delaunay triangulation," *Image and Vision Computing*, vol. 22, no. 4, pp. 319 – 329, 2004.

[14] D. Rivest-Hnault, R. Farrahi Moghaddam, and M. Cheriet, "A local linear level set method for the binarization of degraded historical document images," *International Journal on Document Analysis and Recognition*, vol. 15, pp. 101–124, 2012.

[15] R. Hedjam, R. F. Moghaddam, and M. Cheriet, "A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images," *Pattern Recognition*, vol. 44, no. 9, pp. 2184 – 2196, September 2011.