Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 1308–1315
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Binarization and cleanup of handwritten text from carbon copy medical form images☆

Robert Milewski*, Venu Govindaraju

Center of Excellence for Document Analysis and Recognition, University at Buffalo, UB Commons, 520 Lee Entrance, Suite 202, Amherst NY 14228, USA

Received 20 July 2006; received in revised form 11 June 2007; accepted 25 August 2007

Abstract

This paper presents a methodology for separating handwritten foreground pixels, from background pixels, in carbon copied medical forms. Comparisons between prior and proposed techniques are illustrated. This study involves the analysis of the New York State (NYS) Department of Health (DoH) Pre-Hospital Care Report (PCR) [Western Regional Emergency Medical Services, Bureau of Emergency Medical Services, New York State (NYS) Department of Health (DoH), Prehospital Care Report v4.] which is a standard form used in New York by all Basic and Advanced Life Support pre-hospital health care professionals to document patient status in the emergency environment. The forms suffer from extreme carbon mesh noise, varying handwriting pressure sensitivity issues, and smudging which are further complicated by the writing environment. Extraction of handwriting from these medical forms is a vital step in automating emergency medical health surveillance systems.

1. Introduction

This research evaluates several algorithms that extract handwriting from medical form images (see Fig. 1) to eventually provide the best handwriting recognition performance. This extraction of handwritten stroke pixels from the image is known as binarization. The research copy of the NYS PCR [1] is a carbon mesh document where both the foreground handwriting and the background carbon paper use approximately the same intensity values. While the handwriting on the top form has direct contact between ink and paper, the carbon does not transfer to the paper if there is insufficient pressure. This loss of complete character information in the carbon copy causes character strokes to break after binarization, which leads to recognition failures (the phrase *pressure sensitivity issues* will refer to this situation). Prior binarization algorithms have been reported to better manage noisy and complicated surfaces [2–5]. However, the broken/unnatural handwriting due to ambulance movement and emergency environments, as well as carbon smearing from unintentional pressure to the form, add further complexity to the binarization task. A lexicon-driven word recognizer (LDWR) [6] is used for evaluation of the binarization methods. Analysis of the LDWR, as well as a full view of an actual NYS PCR image, can be found in Ref. [7].

2. Carbon paper

The inconsistent carbon paper, which shows varying grayscale intensities (see Figs. 1 and 2), is referred to as carbon mesh. Fig. 1 shows an example of the "Objective Assessment" region of the NYS PCR form. It provides an overview of the complex nature of the handwriting on the carbon paper. Fig. 2 shows a 400% zoom of one word from Fig. 1. It shows the carbon paper mesh integrated with the carbon handwriting stroke. The displayed word *abd*, in Fig. 2, is a common abbreviation for abdomen. Since the carbon paper causes the paper, the stroke, and any artifacts to have the same intensities, the binarization problem becomes complex. Details of the application of existing algorithms will be discussed in the following sections. This paper describes an algorithm for binarizing the handwriting on carbon paper while preserving the handwriting stroke connectivity better than prior algorithms.

☆ Supported by the National Science Foundation.
* Corresponding author. Tel.: +1 11 716 645 6164.
E-mail addresses: milewski@cedar.buffalo.edu, jyoryoku@gmail.com (R. Milewski), govind@cedar.buffalo.edu (V. Govindaraju).

0031-3203/$30.00 © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.08.018

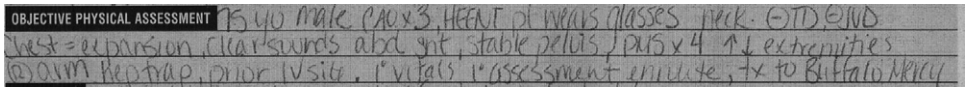


Fig. 1. NYS PCR objective physical assessment.

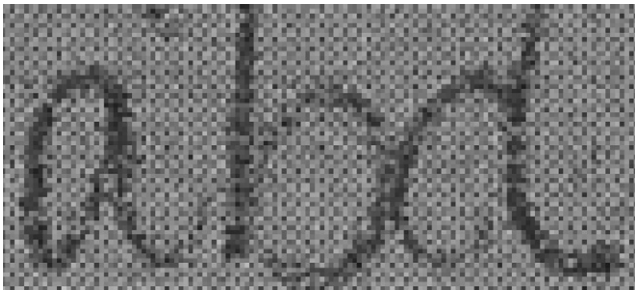


Fig. 2. Grayscale 256 carbon mesh handwriting example (400% zoom).

Pressure sensitivity issues, as a result of light strokes in penmanship, affect the extent of character connectivity after binarization. In order for the carbon copy to receive a reasonable representation of the top copy original, the health care professional needs to press down firmly with the writing instrument. Since the emergency environment is not conducive to good penmanship, the binarization and cleanup algorithms need to compensate.

The carbon paper forms also contain guidelines, which often interfere with the character strokes. These lines can be detected by those pixels with a grayscale value less then a pre-determined threshold; this is consistent across all forms in our data set. To reduce stroke fragmentation, it is sufficient to retain the pixels near the line, thus keeping most character ascenders and descenders reasonably connected. This form drop-out step is performed before binarization.

3. Prior work

In this section, methods described in previous works are compared with our algorithm presented in this research. First we consider the processing of the image in Fig. 3a, using various filters. The histogram of this image, shown in Fig. 4, shows that the foreground (handwriting stroke) and background (carbon paper) use the same intensities in the supplied range. A split at any position in the histogram results in loss of both foreground and background information. The x-axis of the histogram represents the grayscale values 0–255 such that the left most position 0 represents black and the right most position 255 represents white. The y-axis of this histogram is the quantity of pixels for its corresponding grayscale intensity. The mean, median and standard deviation are computations on the grayscale intensities. The standard deviation shows the statistical dispersion of grayscale intensities with respect to the mean. The smaller standard deviation value indicates the grayscale values are clustered around the mean intensity value. The evaluation of pre-processing filters followed by the application of existing binarization algorithms on Fig. 3a is discussed throughout this research.

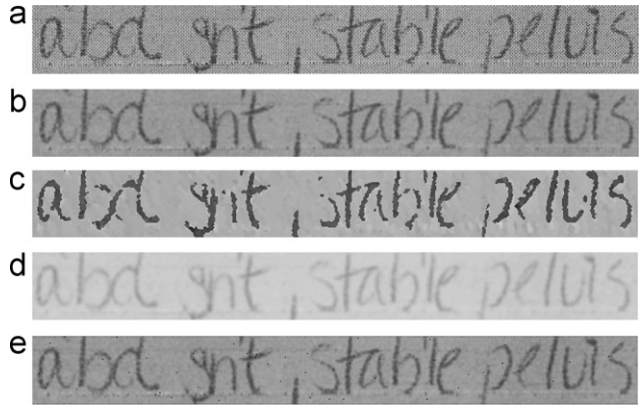


Fig. 3. Smoothing operations (a) Original image + Form drop out (b) Mean filter (c) Median filter (d) Gaussian filter (e) Weiner filter.

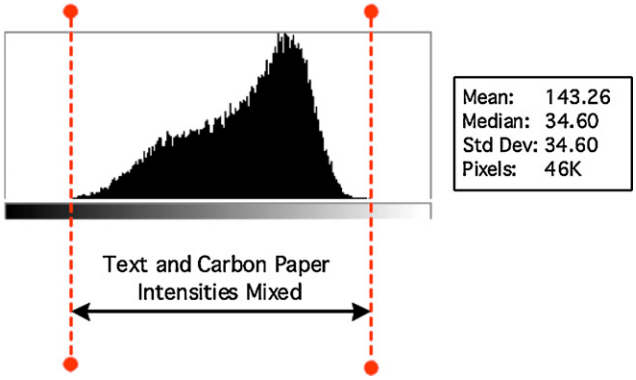


Fig. 4. Histogram for image in Fig. 3a.

Gaussian, median, mean and Weiner filtering/smoothing have been studied in previous works, as a base step, or an integrated step, for noise removal and image enhancement [2,8–11]. Mean filter (Fig. 3b) shows the least damage to strokes in our experiments. Median filter (Fig. 3c) illustrates severe character damage. Gaussian filter (Fig. 3d) demonstrates characters being washed into the background. Weiner filter (Fig. 3e) produces an image very similar to the mean filter, except the background surface is slightly lighter and stroke edges are sharper. Gatos et al. [2] uses the Weiner filter as a pre-processing step to filter image noise.

Global thresholding algorithms determine a single threshold and apply it to the entire image. In the PCR application, the high pressure sensitive areas are binarized well, whereas medium to low pressure areas run the risk of being classified as background.

Other works use algorithms that address some weaknesses of the Otsu [12,13] method, such as with degraded documents.

Any algorithm that computes a global threshold splits the histogram into foreground and background pixels using that threshold. However, since both foreground and background pixels can have the same intensities at different positions on the image, splitting the histogram globally will incorrectly classify foreground pixels and backgrounds. The Wu/Manmatha [4] method expects at least two histogram intensity peaks and globally splits the histogram. This causes large portions of the handwriting to be lost to the background. To compensate, a histogram split is allowed to occur directly before the largest intensity peak in the image (note the highest histogram peak in Fig. 4). This improves the performance of the algorithm, but, still suffers from stroke and background pixels trapped in the largest histogram peak (see Fig. 4).

The Niblack binarization [14] algorithm is an adaptive technique that has been compared to other methods in applications such as image and video text detection and extraction [15], low quality camera images [16], low quality grayscale utility maps (such as cable and hydro maps with various intensity and noise issues [10]), and low quality historical documents [2]. This algorithm results in severe noise, jagged edges and broken character segments. While post-processing improves the algorithm performance, the broken character strokes result in lower performance. This is due to mean-variance computations occurring at lighter stroke regions.

Sauvola binarization [17] modifies the Niblack algorithm [14] and attempts to suppress noisy areas. In the cases of stronger handwriting pressure, Sauvola [17] has positive results. However, Sauvola [17] has fewer positive results than Niblack [14] in our experiments. Sauvola's [17] noise suppression affects the lighter strokes thereby causing incorrect recognizer segmentation.

Gatos et al. [2] introduced an algorithm that incorporates Sauvola [17], but this implies that the performance of Gatos et al. [2] will drop along with that of Sauvola [17]. While Gatos et al. [2] does illustrate a performance improvement over Sauvola [17], this combination still under-performs Niblack [14] after post-processing. This is because Gatos often loses holistic features due to incorrect background estimation of the paper.

Logical binarization uses heuristics for evaluating whether a pixel belongs to the foreground or background. Other adaptive binarization strategies are integrated with such heuristics. The Kamel/Zhao algorithm [18] finds stroke locations and then later removes the noise in the non-stroke areas using an interpolation and thresholding step. Various stroke width combinations from 1–10 pixels were tried. However, the stroke is not adequately traced using this algorithm.

The Yang/Yan [5] algorithm is a variant of the method developed by Kamel/Zhao [18]. The modifications are meant to handle low quality images affected by varying intensity, illumination, and artifacts such as smearing. However, the run analysis step in this algorithm is computed using only black pixels. Neither the foreground (handwritten stroke) or background (carbon paper) of the carbon copy medical forms have black pixels; nor are the foreground pixels the same intensity throughout. Therefore, the stroke-width computation, which is dependent on the run-length computation, cannot be trivially determined in the carbon paper forms.

In addition to the binarization algorithms, various post-processing strategies are commonly used. The despeckle algorithm is a simple noise removal technique using a 3×3 mask to remove a foreground pixel that has no D8 neighbors [9]. The blob removal algorithm is a 9×9 mask that removes small pixel regions that have no neighbors [9]. The amorphous artifact filter removes any connected component whose pixel area is less than a threshold (60 pixels in this research) [9]. The Niblack [14] + Yanowitz and Bruckstein method [11] was found to be the best combination strategy by Trier and Taxt [10]. The Shi and Govindaraju method is an image enhancement strategy that has been used on postal mail-pieces [19].

4. Proposed algorithm

Prior algorithms have relied on techniques such as histogram analysis, edge detection, and local measurements. However, these techniques are less effective on medical forms. Our algorithm uses a larger central $N \times N$ mask, which determines the intensity of one region, and compares it with the intensities of multiple dynamically-moving smaller $P \times P$ masks (see Figs. 5, 6 and 7).

One hypothesis in managing the varying intensities of the carbon mesh and its similarities with the stroke is to use a wave trajectory (see Figs. 6 and 7) for the D8 positioned masks (see Fig. 5), as opposed to a linear trajectory (see Fig. 8). A wave/trajectory is a path, in a Cartesian system, that undulates across an axis in 2D space with an amplitude and frequency that can be adjusted (see Fig. 6). The experiments illustrate that the use of a wave trajectory is beneficial for the following reasons: (i) There is a better chance of the trajectory of the mask to evade a stroke. (ii) The possibility of finding a background region as close as possible to the central mask is enhanced. Note that as one goes further out from the center mask, the more likely it is to find that the carbon mesh of the background can change. (iii) The best background region to compare to a handwriting stroke may or may not be the edge of the stroke.

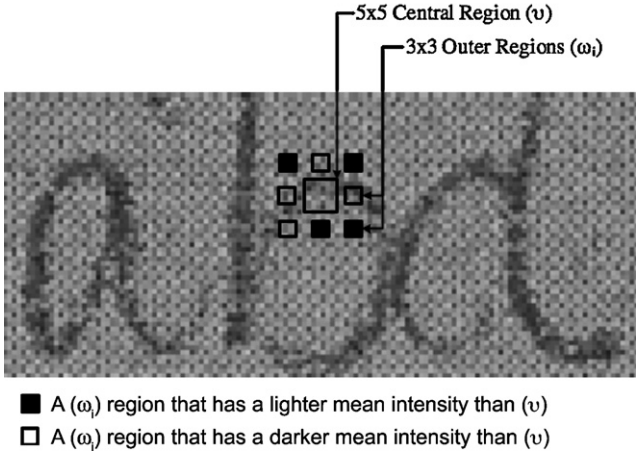


Fig. 5. Initial mask placement example ($N = 5$ and $P = 3$).

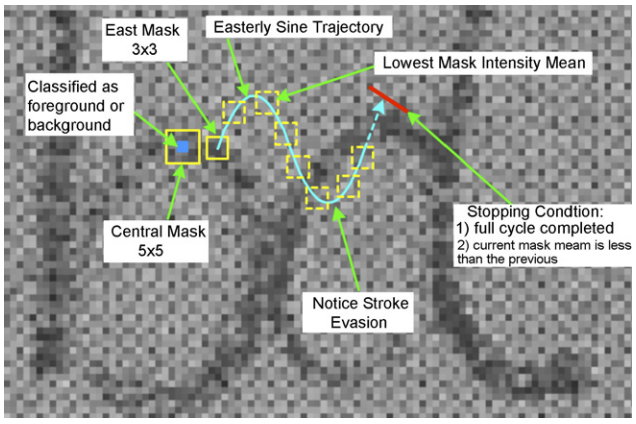


Fig. 6. Sine wave trajectory.

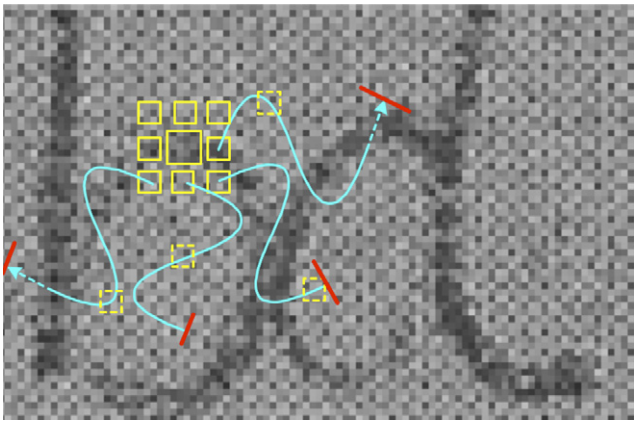


Fig. 7. Sine wave coverage.

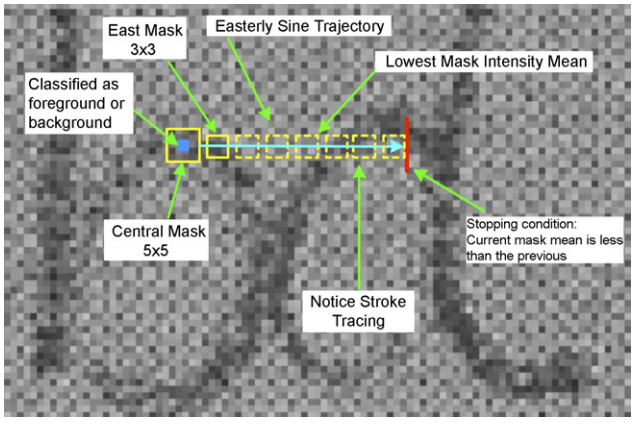


Fig. 8. Linear trajectory.

(iv) Areas surrounding a stroke in the same trajectory can be observed. (v) Eight points of comparison, one for each trajectory, are performed (as opposed to one point used on other algorithms). With the inclusion of a stopping condition operating independently on each trajectory, this approach, as opposed to other global and adaptive approaches, does not get confined to square mask windows which are relative to a central position. In this context, the wave trajectory for scanning can be thought of as searching for lighter pockets in the intensity fluctuation of the carbon mesh (see Figs. 6 and 7).

A sine wave trajectory offers the benefit of beginning at the origin and allowing a continuous trajectory regardless of distance (i.e. the wave will continue until the stopping condition is met as opposed to being confined to an arbitrary box). It allows the control of frequency and amplitude that is necessary to adjust for stroke width. Sinusoidal waves have been used in other contexts for the modeling of human motor function for on-line handwriting recognition, feature extraction and segmentation [20], shape normalization of Chinese characters [21], and signal canceling of pathological tremors while writing [22]. Based on these studies, and the knowledge of the English character set, it was possible to scan out from a character stroke at a certain frequency. This allows a handwriting stroke to be maneuvered, as opposed to traced, in the search for background regions. The sine trajectory can be thought of as a path which has the potential to cross handwritten strokes. This allows the background paper on both sides of the stroke, in all directions, and with a dynamic distance, to be evaluated. Intuitively, more space can be searched and both sides of the stroke can be evaluated in the same computational step at variable distances. It is also presumed that in a moving ambulance, carbon smearing is more likely since the writer will press harder on the document to maintain balance in the vehicle. While strokes in the English language contain both curves and straight lines, at the pixel level they can be considered piecewise linear movements such that a linear scan will trace the stroke and reduce the likelihood of finding the background. Furthermore, holistic features (such as the area in the letter "D") are typically small. Missing the carbon paper inside such character holes may result in missed background analysis. This motivated the use of a higher sine wave frequency so that the trajectory would pass through the center of holistic features as frequently as possible. Additionally, since the thickness of characters fluctuates, it is difficult to precisely calculate the true stroke width.

A grayscale image 0 (black) to 255 (white) is the input, and a binarized image is the output. At a given position on the image, there are 9 masks. A single mask is denoted as Ξ. The mean intensity of all pixels within a single mask is denoted by M(Ξ). The central mask which slides across the image is denoted by (υ) and has a size $N \times N$, such that $N \geqslant 3$ and consists of numerically odd dimensions (e.g. 3×3, 5×5, and 7×7). The size of (υ) is based on the estimated stroke width constant denoted by ϕ. The value of ϕ has been estimated to be 5 pixels (the estimated stroke width which may be dependent on the writing instruments), therefore (υ) is of size 5×5. At each (υ) position over the image, 8 masks are initially stationed in each D8 position (Fig. 5) and are denoted by (ω_i) where $1 \leqslant i \leqslant 8$. The mask size of ($\omega_i$) is $P \times P$ such that $3 \leqslant P \leqslant \lceil N/2 \rceil$. Note that $P \leqslant \lceil N/2 \rceil$ allows a small mask the opportunity of preserving small holistic features when moving on the sine curve, and also making sure that the mask will not overlap (υ). Each (ω_i) is initially stationed as close to (υ) as possible so as to avoid the mask overlapping between (ω_i) and (υ). Each (ω_i) moves in its respective D8 direction, either linearly (see Fig. 8), randomly

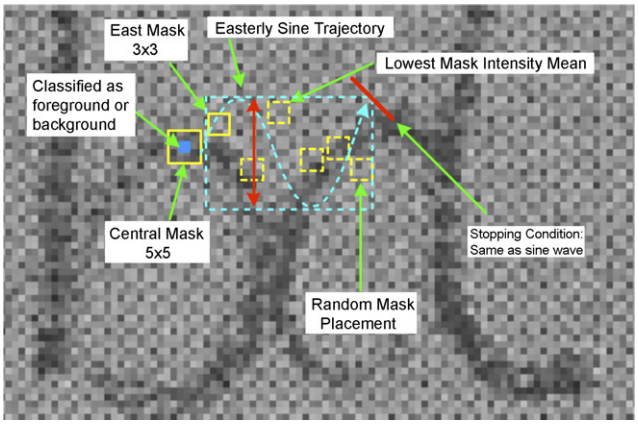


Fig. 9. Random trajectory.

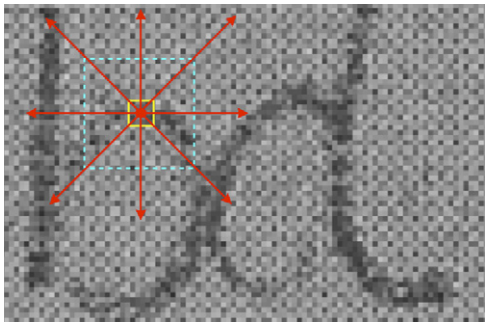


Fig. 10. Sliding square window.

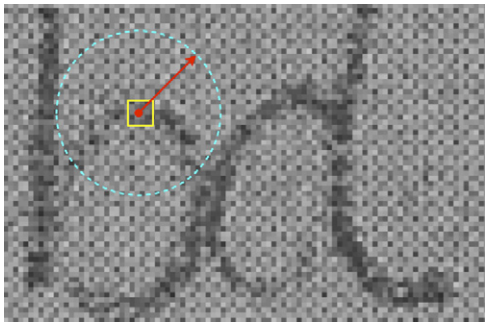


Fig. 11. Sliding circular window.

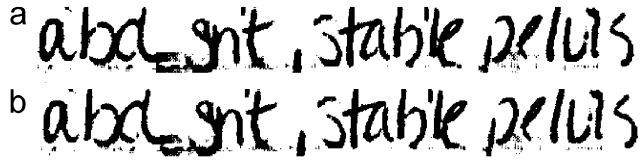


Fig. 12. Window binarization examples: (a) sliding square window (b) sliding circular window.

(see Fig. 9) or via a sinusoidal wave (see Fig. 6). The $M(\omega_i)$ is computed at each position along the trajectory and stops at a position after one cycle and when the current mask average intensity is lighter than the previous one on the sine trajectory. A list of mean values for each position on that trajectory is denoted by $M(\omega_i)_q$ where q is a coordinate on the sine curve. The lowest intensity mask on a single sine wave trajectory is represented by the equation $M(\omega_i)_{\min} = \min(M(\omega_i)_{\forall_q})$. Next, a comparison of all the D8 $M(\omega_i)_{\min}$ positions are made against $M(\upsilon)$. If there are 3–4 (empirically determined using either LDWR performance tests or majority voting) of the 8 $M(\omega_i)_{\min}$ values which satisfy the equation $M(\omega_i)_{\min} - M(\upsilon) \geqslant \kappa$, such that κ is a small constant (using $\kappa = 10$), then the center pixel of (υ) is classified as a foreground pixel. The value κ defines a tolerance with respect to the localized intensity fluctuation of the carbon paper and denotes the carbon intensity similarity rule. Given that a new image has been initialized to white background pixels, it is only necessary to mark the foreground pixels when they are found. A dynamic programming step is used to store each $M(\omega_i)$, corresponding to the appropriate region on the image beforehand, to improve the run-time performance.

The sinusoidal trajectory is defined by Eq. (1).

$$y = 2\phi \sin(\tfrac{1}{2}x). \tag{1}$$

The coordinate (x, y), on a sinusoidal trajectory is relative to its starting location (origin). A nearest neighbor approach is sufficient for conversion of real coordinates to pixel coordinates. Each ω_i is computed on the sine curve trajectory (see Figs. 6 and 7). Note that using ϕ as the amplitude in Eq. (1), without the coefficient, will result in a distance of 2ϕ between the highest and lowest y-axis points (using ϕ as the stroke width). In addition, using 2ϕ as the amplitude yields a distance of 4 times the stroke width, to account for the possibility of 2 touching strokes (i.e. two touching letters). In this way, the curve efficiently exits a stroke while searching for the background. The constant $\frac{1}{2}$ is used in Eq. (1) so that the sine frequency does not trace the handwritten stroke. The objective of the empirically chosen constant is to maximally evade the stroke. Since the stroke pixels are not known in advance, nor easily approximated, choosing a constant is situation dependent. As a general rule, the optimal constant will result in the highest LDWR recognition performance on known sample sets.

One alternative approach to the sinusoidal approach would be to search for the lightest mean intensity mask within a sliding window that is the same distance as one complete cycle of the sinusoidal wave. Figs. 10 and 11, illustrate the window structures and Fig. 12 shows the output. The procedure for computing these window structures is to calculate the intensity average of all 3×3 masks within a window. It is then possible to compare the lightest of these averaged intensity masks with the average intensity of the central 5×5 mask. However, only a single mask value within the window is compared to the central mask which requires κ (as defined above) to be larger ($\kappa = 30$ in Fig. 12) resulting in a completely black image. This is due to the larger area of coverage which increases the likelihood that a lighter area will always be found relative to the center mask, thereby classifying the area as foreground. In addition, there is a degradation of holistic features, close-proximity characters combining and more broken characters than in the sine wave approach (see Section 5 results). This requires κ to be

based on the image. The sine wave approach rarely suffers from this situation since at least 3 of the trajectories authenticate a foreground value instead of one value. Therefore, the carbon intensity similarity rule breaks down, and causes the effectiveness of these sliding window approaches to be as problematic as a global thresholding technique. In this way, κ becomes the global threshold.

Another possible approach computes the Otsu [13] algorithm in small windows rather than over the entire image. However, this results in an output image nearly identical to that obtained by computing Otsu [13] globally. The thresholds chosen are negligibly different between windows and, therefore, the image is still noisy and many strokes are still broken.

An alternate strategy to the sine wave is a randomized mask movement (Fig. 9). Instead of the outer masks moving on the sine wave trajectory, they move on the y-axis randomly within the same rectangular area of the sine wave movement. It may be expected that the randomized version (see Fig. 9) will perform as well as the sine wave. However, since the window involved in the sine wave trajectories is reasonably small, if a stroke is present within that window, and the randomized approach is used, there is no guarantee that the stroke will be evaded. Therefore, if a random position is chosen, and that rests on a stroke as opposed to the background, then the desired background position is missed. The sine wave approach is more likely to cross the stroke rather than tracing it. Furthermore, due to the nature of randomized approaches, the recognition results may not be consistent.

5. Results

All experiments (see Fig. 13) were performed on a set of 62 PCRs consisting of $\sim$ 3000 word images and various size lexicons (see Figs. 16 and 17). The linear, random, square and circular strategies (see Figs. 8–12) were outperformed by Otsu [13]. The sine wave strategy presented here outperformed all prior algorithms, with a 11–31% improvement. After post-processing there was a 4.5–7.25% improvement (see Figs. 16 and 17).

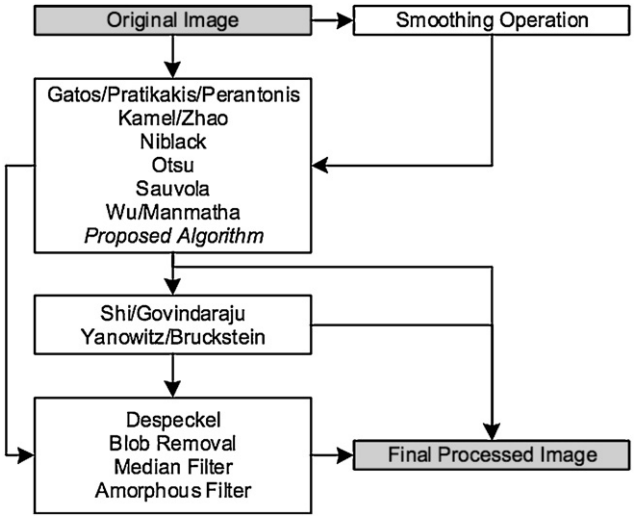


Fig. 13. Image processing combinations.

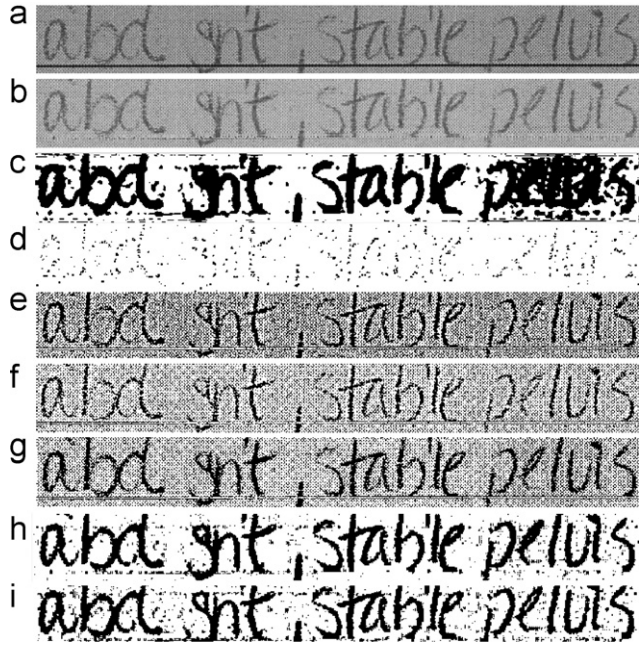


Fig. 14. Comparison of binarization algorithms only: (a) original image (b) original image with form drop out (c) Wu/Manmatha binarization (d) Kamel/Zhao binarization (e) Niblack binarization (f) Sauvola binarization (g) Otsu binarization (h) Gatos/Pratikakis/Perantonis binarization (i) sine wave binarization.

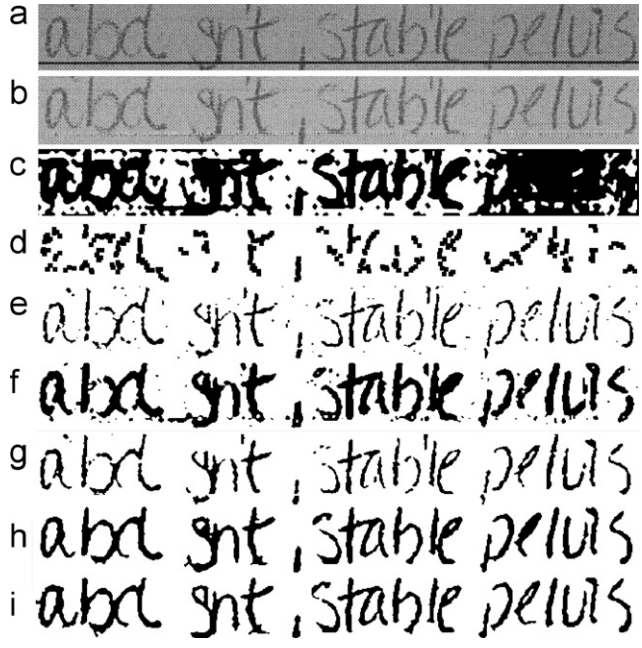
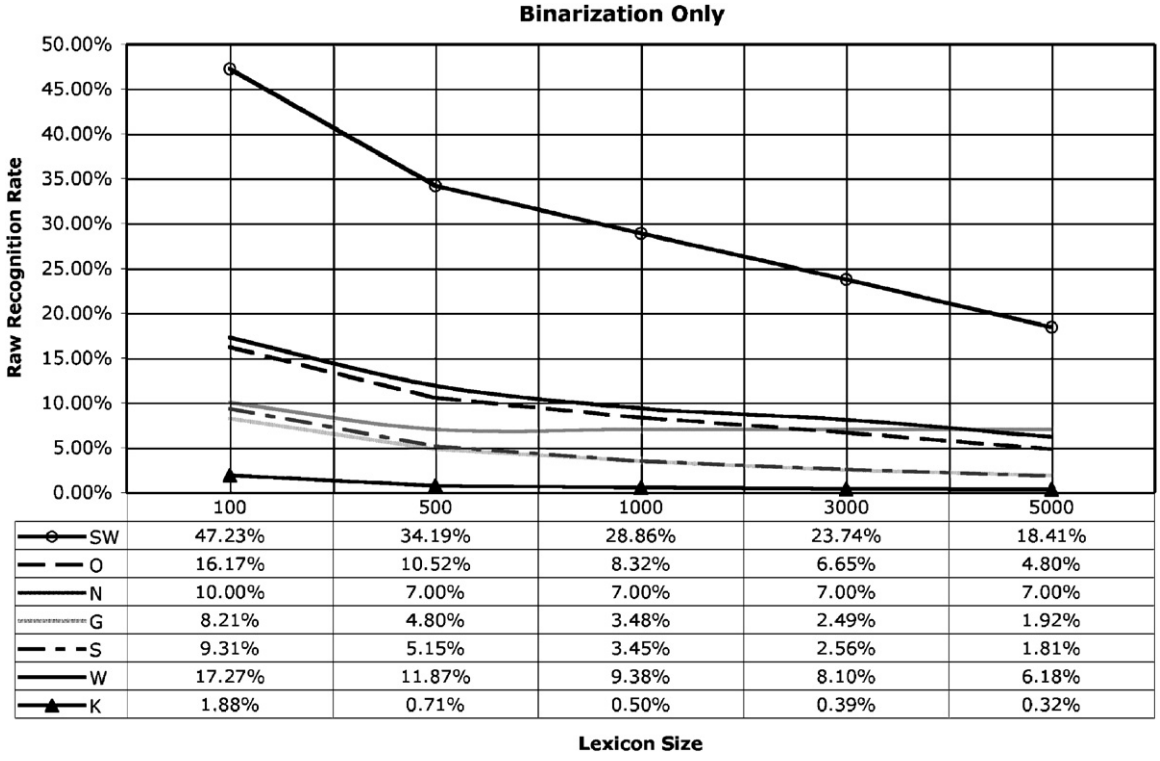
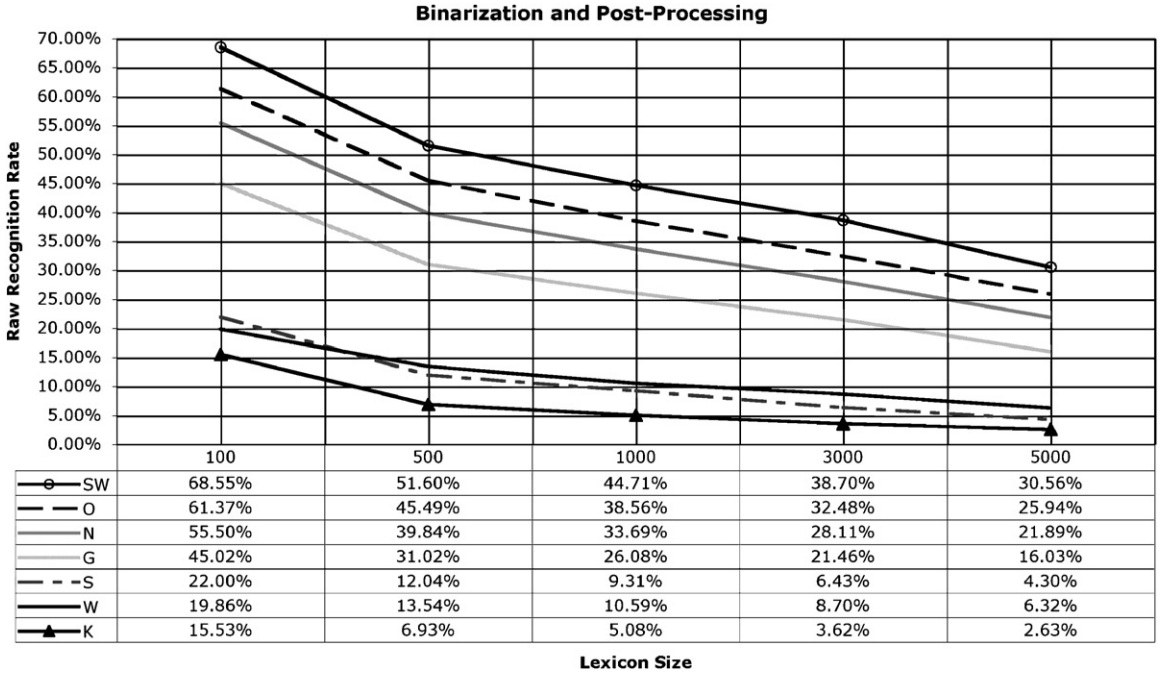


Fig. 15. Comparison of binarization algorithms with their best post-processing strategy: (a) original image (b) original image with form drop out (c) Wu/Manmatha binarization + Shi/Govindaraju + Despeckel (d) Kamel/Zhao binarization + Shi/Govindaraju + Despeckel + amorphous filter (e) Niblack binarization + Despeckel + amorphous filter (f) Sauvola Binarization + Yanowitz/Bruckstein + Despeckel (g) Otsu binarization + Despeckel + blob removal (h) Gatos/Pratikakis/Perantonis binarization + Despeckel + amorphous filter (i) sine wave binarization + amorphous filter.



	100	500	1000	3000	5000
SW	47.23%	34.19%	28.86%	23.74%	18.41%
O	16.17%	10.52%	8.32%	6.65%	4.80%
N	10.00%	7.00%	7.00%	7.00%	7.00%
G	8.21%	4.80%	3.48%	2.49%	1.92%
S	9.31%	5.15%	3.45%	2.56%	1.81%
W	17.27%	11.87%	9.38%	8.10%	6.18%
K	1.88%	0.71%	0.50%	0.39%	0.32%

Fig. 16. Binarization only performance.



	100	500	1000	3000	5000
SW	68.55%	51.60%	44.71%	38.70%	30.56%
O	61.37%	45.49%	38.56%	32.48%	25.94%
N	55.50%	39.84%	33.69%	28.11%	21.89%
G	45.02%	31.02%	26.08%	21.46%	16.03%
S	22.00%	12.04%	9.31%	6.43%	4.30%
W	19.86%	13.54%	10.59%	8.70%	6.32%
K	15.53%	6.93%	5.08%	3.62%	2.63%

Fig. 17. Binarization + post-processing performance.

The handwriting phrase depicted in Figs. 14 and 15, "abd snt, stable pelvis" means "abdominal soft-not-tender, stable pelvis." Figs. 14 and 16 show the performance of the aforementioned binarization strategies with no post-processing support. Figs. 15 and 17 reflect the performance of the best respective post-processing combinations from Fig. 13.

With respect to Figs. 16 (maximum y-axis value is 50% for visual clarity) and 17 (maximum y-axis value is 70% for visual clarity), the y-axis represents the percentage of correctly recognized words by the LDWR [6] versus the lexicon size on the x-axis. It is expected that the performance decreases with an increase in lexicon size since the LDWR [6] is lexicon driven and,

therefore, has more choices from which to select. The definition of correctly recognized words is in the context of raw recognition performance. The error rate is consistently high in this application. Furthermore, words in the form region were manually segmented. The LDWR [6] algorithm uses pre-processing strategies for its own noise removal and smoothing before executing its recognition algorithm [6,23,24]. Therefore, a noisy image submitted to the LDWR algorithm will be internally pre-processed by the handwriting recognizer. The first letter of an author's name is used to refer to the algorithms: (G)atos, (K)amel, (N)iblack, (O)tsu, (S)auvola, (W)u. (SW) designates Sine Wave binarization.

6. Conclusions

In this paper we have described a binarization algorithm for handling carbon paper medical documents. Improvements of approximately 11–31% (using various lexicon sizes) have been obtained over prior binarization algorithms. Approximately 4.5–7.25% improvement is obtained with post-processing.

References

[1] Western Regional Emergency Medical Services, Bureau of Emergency Medical Services, New York State (NYS) Department of Health (DoH), Prehospital Care Report v4.
[2] B. Gatos, I. Pratikakis, S.J., Perantonis, Adaptive degraded document image binarization, J. Pattern Recognition Soc. 2005.
[3] G. Leedham, S. Varma, A. Patankar, V. Govindaraju, Separating text and background in degraded document images a comparison of global thresholding techniques for multi-stage thresholding, in: Proceedings Eighth International Workshop on Frontiers of Handwriting Recognition, 2002.
[4] V. Wu, R. Manmatha, Document image clean-up and binarization, Proceedings of SPIE Symposium on Electronic Imaging, 1998.
[5] Y. Yang, H. Yan, An adaptive logical method for binarization of degraded document images, J. Pattern Recognition Soc. 2000.
[6] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, IEEE Trans. PAMI 19 (4) (1997) 366–379.
[7] R. Milewski, V. Govindaraju, Handwriting analysis of pre-hospital care reports, in: IEEE Proceedings. Seventeenth IEEE Symposium on Computer-Based Medical Systems (CBMS), 2004.
[8] S. Hatami, R. Hosseini, M. Kamarei, H. Ahmadi, Wavelet based fingerprint image enhancement, in: IEEE International Symposium on Circuits and Systems (ISCAS), 2005.
[9] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, second ed., PWS Publishing, 1999.
[10] O.D. Trier, T. Taxt, Evaluation of binarization methods for document images, IEEE Trans. PAMI 17 (3) (1995).
[11] S.D. Yanowitz, A.M. Bruckstein, A new method for image segmentation, Comput. Vision Graphics Image Process. 46 (1) (1989).
[12] P.S. Liao, T.S. Chen, P.C. Chung, A fast algorithm for multilevel thresholding, J. Inf. Sci. Eng. 2001.
[13] N. Otsu, A threshold selection method from gray-level histogram, IEEE Transactions on System Man Cybernetics SMC-9(1) C1979.
[14] W. Niblack, An Introduction to Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[15] C. Wolf, J.M. Jolion, F. Chassaing, Text localization, enhancement and binarization in multimedia documents, in: 16th International Conference on Pattern Recognition (ICPR), 2002.
[16] M. Seeger, C. Dance, Binarising camera images for OCR, Xerox Research Center Europe, in: 6th International Conference on Document Analysis and Recognition, 2001.
[17] J. Sauvola, T. Seppanen, S. Haapakoski, M. Pietikainen, M. Adaptive document binarization, in: International Conference on Document Analysis and Recognition, vol. 1, 1997.
[18] M. Kamel, Zhao, Extraction of binary character/graphics images from grayscale document images, CVGIP: Graphics Models Image Process. 55 (3) (1993).
[19] Z. Shi, V. Govindaraju, Character image enhancement by selective region-growing, Pattern Recognition Lett. 17 (1996).
[20] H. Beigi, Processing, modeling and parameter estimation of the dynamic on-line handwriting signal, in: Proceedings World Congress on Automation, 1996.
[21] C.L. Liu, K. Marukawa, Global shape normalization for handwritten Chinese character recognition: a new method, International Workshop on Frontiers of Handwriting Recognition, 2004.
[22] D.S. Hsu, W.M. Huang, N.V. Thakor, Stylpen: on-line adaptive canceling of pathological tremor for computing pen handwriting, IEEE Trans. Biomed. Eng., 1998.
[23] J. Schurmann, et al., Document analysis-from pixels to contents, Proc. IEEE 80 (7) (1992).
[24] M.K. Brown, S. Ganapathy, Preprocessing techniques for cursive word recognition, Pattern Recognition 16 (5) (1983).

About the Author—Dr. ROBERT MILEWSKI received his Ph.D. (2006), M.S. (2001), and B.S. (1999) in Computer Science/Artificial Intelligence at the University at Buffalo (UB), State University of New York (SUNY).

About the Author—Dr. VENU GOVINDARAJU is a Professor of Computer Science and Engineering at the University at Buffalo (UB), State University of New York. He received his B.Tech (Honors) from the Indian Institute of Technology (IIT), Kharagpur, India in 1986, and his Ph.D. degree in Computer Science from UB in 1992. He is a Fellow of the IEEE and IAPR.