An Algorithm for Fast Adaptive Image Binarization With Applications in Radiotherapy Imaging

Torbjørn Sund* and Karsten Eilertsen

***Abstract*—Locally adaptive image binarization with a sliding-window threshold can be an effective tool for various image processing tasks. We have used the method for the detection of bone ridges in radiotherapy portal images. However, a straightforward implementation of sliding-window processing is too time consuming for routine use. Therefore, we have developed a new thresholding criterion suitable for incremental update within the sliding window, and we show that our algorithm gives better results on difficult portal images than various publicly available adaptive thresholding routines. For small windows, the routine is also faster than an adaptive implementation of the Otsu algorithm that uses interpolation between fixed tiles, and the resulting images are equally good.**

***Index Terms*—Adaptive binarization, pixel histogram, portal imaging, second moments adaptive binarization (SMAB), thresholding.**

NOMENCLATURE

M	Side length of image.
N	Side length of sliding window.
$i,\ j$	$= 1 \cdots M$. Center of sliding window.
$x_{i,\,j}$	Gray value of source pixel at position $i,\ j$.
$y_{i,\,j}$	Gray value of destination pixel at position $i,\ j$.
g	Maximum gray value.
$h(p)$	Histogram of source image.
$H_0(x)$	$= \sum_{p=0}^{x} h(p)$. Cumulative histogram.
$H_1(x)$	$= \sum_{p=0}^{x} p \times h(p)$. First-order moment of h with respect to x.

I. INTRODUCTION

MODERN radiotherapy is performed with millimeter precision. To maintain this accuracy throughout the treatment, it is necessary to compare treatment planning reference images with X-ray images taken during the actual treatment (portal images). Due to the high energy of the X-rays (typically, 4–25 MV), the portal images have very low contrast, making the image comparison error prone and time consuming.

Several authors have addressed the issue of coregistration of reference and portal images; see [4] for an overview. We have implemented both semimanual and fully automated methods, based on enhancing anatomical structures that are present in both images, in particular, the high intensity ridges formed behind the crests of bones. The techniques require that the images are binarized. Until recently, we used a gradient operator followed by global thresholding and thinning to extract the ridges from the images. We have now introduced adaptive binarization as a simpler, faster, and more robust procedure.

We first investigated algorithms for adaptive binarization that are available in the public domain image processing library XITE [2], [15], but found that none were suitable for our application. We then tried global thresholding methods within a sliding window, and the well-known method described by Otsu [9] gave excellent results. Unfortunately, it was very time consuming. This is understandable when one considers how sliding-window thresholding works: A window with a suitable size is stepped pixel by pixel over the source image. A threshold valid for the window as a whole is found, but only the *single* target pixel at the center is classified.

To reduce the number of pixels that must be processed in sliding-window image processing, several authors (e.g., [5] and [8]) have suggested an incremental update of the functions that depend on the pixels within the window. This is straightforward when the functions are based on the histogram alone, but is not obvious for more complicated functions. For median filtering, Huang *et al.* [6] developed a method for pointwise incremental update of the cumulative histogram within a sliding window. In this paper, we extend their idea to image binarization, using a new threshold selection method suitable for incremental update.

II. METHODS

A. Assumptions

We assume temporarily that the image extends indefinitely in all directions, so that we can ignore image boundaries and focus on the binarization method. In Section II-G, we show our actual treatment of image boundaries. To further simplify our presentation, both the image and the sliding window are assumed to be square. Generalizing the method to rectangular cases is a trivial matter, and our implementation is written for arbitrary image and window sizes.

B. Binarization by the Method of Otsu

We selected the method described by Otsu [9] as our reference binarization method. It is well known and widely used, and was ranked highest in a comparative study of four global binarization methods [15]. It consists of selecting the threshold t which maximizes the variance v between the means of the histogram

Manuscript received November 27, 2000; revised July 1, 2002. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was M. Unser. *Asterisk indicates corresponding author.*

*T. Sund is with Telenor Research and Development, B2F, N-1331 Fornebu, Norway (e-mail: torbjorn.sund@telenor.com).

K. Eilertsen is with the Department of Medical Physics and Technology, The Norwegian Radium Hospital, N-0310 Oslo, Norway.

Digital Object Identifier 10.1109/TMI.2002.806431

classes on each side of the threshold. The condition can be expressed as a function of the cumulative histogram H_0 and the cumulative first moment H_1

$$v(x) = \frac{(H_1(x) - H_1(g) \times H_0(x)/H_0(g))^2}{H_0(x) \cdot (1 - H_0(x)/H_0(g))} \quad (1)$$

$$t = \mathrm{Argmax}(v(x)) \quad (2)$$

$$y(i, j) = \begin{cases} 0, & x(i, j) < t \\ 1, & x(i, j) \geq t. \end{cases} \quad (3)$$

It is not possible to solve (2) analytically, so the threshold must be found by a search method.

C. Incremental Update in a Sliding Window

A straightforward brute-force computation of the histogram within a sliding window entails first setting the histogram to zero, and then updating the count for each of the N^2 pixels contained in the window. The number of operations per image pixel is thus $O(g+N^2)$ (ignoring edge effects). Instead of recounting completely the histogram when the window moves to cover a new pixel row, the histogram can be incremented for the new pixels that come into the window, and decremented for those that are excluded from the window. This reduces the operations per pixel to $O(2N)$.

The pixel histogram alone is, however, sufficient only for a limited set of image transformations. To use Otsu's method for example, the cumulative histogram needs to be known, and for medical images with ten to twelve bits (1024 to 4096 gray levels), an incremental update of the cumulative histogram is more costly than its recomputation for each window position. This is because every pixel added to or removed from the window affects *all* cumulative histogram points above that pixel value.

The cumulative histogram H_0 for the *single gray value* at the center of the sliding window can nevertheless be updated efficiently, as follows. Let us write out H_0 as $H_0(i, j, x_{i,j})$, to demonstrate its dependency both on the window position i, j and on the center pixel value $x_{i,j}$. Knowing the value $H_0(i, j-1, x_{i,j-1})$ at the preceding position of the window, the value $H_0(i, j, x_{i,j-1})$ at the *current* window position and for the *previous* center pixel value is found by counting up respectively down for all pixels $\leq x_{i,j-1}$ that are added to or removed from the window when it is moved. The histogram is also updated. Second, $H_0(i, j, x_{i,j})$ for the current center pixel value is determined from $H_0(i, j, x_{i,j-1})$ by adding the updated histogram values for p in the interval $x_{i,j-1} \cdots x_{i,j}$ if $x_{i,j} > x_{i,j-1}$, and subtracting otherwise.

The difference $|x_{i,j-1} - x_{i,j}|$ between neighbor pixels is usually only a small fraction of the full grayscale, and the number of steps to find the cumulative histogram at the current pixel value is correspondingly small. In the extreme (and irrelevant) case of no or very small similarities between neighboring pixels, the incremental algorithm "blows up" with an execution time proportional to g. In a computer implementation, this should be avoided, e.g., by checking the mean difference between neighbor pixels after each row traversal.

D. Adaptive Binarization by Second Moments of the Histogram

To exploit the possibility of incremental update, we devised a new binarization criterion based on the second moment of the histogram, using the value of the center pixel as the pivot point. In the following, this method will be called *second moments adaptive binarization* (SMAB). If the second moment to the "left" of the pivot (toward dark pixels) is smaller than the second moment to the right, the center pixel is considered to belong to the class of dark pixels, and *vice versa*. Expressing this in the mathematical framework used above, and denoting the left and right moments by M_L and M_R, respectively, we have

$$M_L(x) = \sum_{p=0}^{x} (x-p)^2 \cdot h(p)$$

$$M_R(x) = \sum_{p=x}^{g} (x-p)^2 \cdot h(p)$$

$$y_{i,j} = \begin{cases} 0, & M_L < M_R \\ 1, & M_L \geq M_R. \end{cases}$$

It is a trivial matter to incrementally update $M_L(x)$ and $M_R(x)$ (for constant x) when the window is moved to its new position and pixels are added and removed. Iterative formulas to extend M_L and M_R to the new value of x are found by substituting $x-1$ and $x+1$ for x in the above expressions. Subtracting and rearranging, we arrive at the following expressions:

$$\begin{aligned} M_L(x+1) &= M_L(x) - 2H_{1L}(x) + (2x+1)H_{0L}(x) \\ M_L(x-1) &= M_L(x) + 2H_{1L}(x) - (2x-1)H_{0L}(x) - h(x) \\ M_R(x+1) &= M_R(x) - 2H_{1R}(x) + (2x+1)H_{0R}(x) - h(x) \\ M_R(x-1) &= M_R(x) + 2H_{1R}(x) - (2x-1)H_{0R}(x) \end{aligned}$$

where the following definitions have been used:

$$H_{0L}(x) = \sum_{p=0}^{x} h(p)$$

$$H_{0R}(x) = \sum_{p=x}^{g} h(p)$$

$$H_{1L}(x) = \sum_{p=0}^{x} p \cdot h(p)$$

$$H_{1R}(x) = \sum_{p=x}^{g} p \cdot h(p).$$

Formulas for extending H_{0L}, H_{0R}, H_{1L}, and H_{1R} to $x+1$ or $x-1$ are obvious. Initial values (at the start of each pixel row) are found by setting $x = 0$ in the above equations.

SMAB can be further speeded up by a technique of *pivot search optimization*. As formulated above, the pivot point for the second moment should be iterated from the previous gray value $x_{i,j-1}$ to the new value $x_{i,j}$. However, when the previous pivot point is on the same side of $x_{i,j}$ as the larger of the two moments, moving the pivot point toward $x_{i,j}$ can only result in an even larger difference between the two moments. For neighbor pixels that belong to the same class, the class of the new pixel

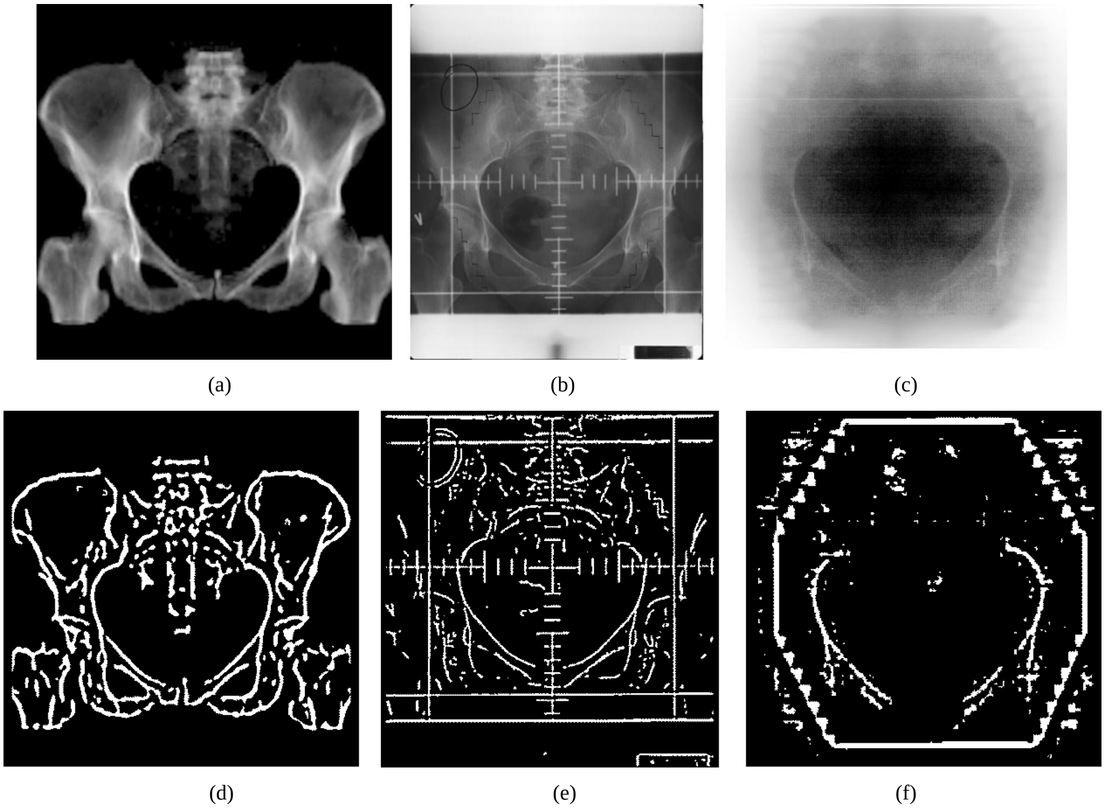


Fig. 1. Application of SMAB to extract the contour of the pelvis for various image modalities. (a) Digitally reconstructed radiograph (DRR). (b) Digitized X-ray film from the radiotherapy treatment simulator. (c) Image acquired from an electronic portal imager during treatment. (d)–(f) Binarized images.

may thus be found even without moving the pivot point. Similarly, for neighbor pixels of different class, the iteration of the pivot point toward $x_{i,\,j}$ can be terminated when the difference between the two second moments changes sign.

E. Adaptive Binarization Using Fixed Tiles

A common alternative to sliding-window image processing, suggested by several textbook authors (e.g., [8]) consists of finding separate thresholds for adjoining, fixed windows, and using bilinear interpolation between the four nearest window centers to determine the threshold for each pixel. The cost to determine the thresholds increases with the number of tiles, which is proportional to $1/N^2$, whereas the time needed for the interpolation is independent of N. The execution time of the tiled version should, therefore, be nearly constant for large windows; for small windows, it is expected to rise steeply with decreasing N.

It should be noted that the interpolation may mask abrupt changes in the image on a sub-tile scale [12], possibly counteracting the intended effect of the adaptive image thresholding. Alternatively, one can employ partly overlapping tiles and no interpolation (as in, e.g., [3]), but in our experience, this tends to introduce stripes in the image corresponding to the overlap.

F. Classification of Uniform Regions

Adaptive binarization in a sliding window will always produce a threshold value, even if the window covers a low-contrast background area which should not be thresholded. An adaptive binarization algorithm must, therefore, comprise a method to differentiate between bilevel and uniform areas. We use the sum of the two second moments of the histogram $H_{0L} + H_{0R}$ as a measure of the contrast within the window, normalized so that a contrast level of 100 corresponds to the second moment that would result if all the pixels were evenly divided into two histogram peaks, separated by one tenth of the full grayscale.

The bilevel threshold cannot be used to classify pixels in uniform regions. If the class of the uniform regions is known *a priori*, the center pixel can be set accordingly; otherwise, an adaptive algorithm must be used. In their implementation of the Otsu algorithm, Eikvil *et al.* [3] compared the overall pixel mean within uniform areas to a running mean of foreground and background pixels, obtained from already classified bilevel areas. The center pixel was classified as belonging to the closest of the two. We use this method for adaptive classification of uniform areas and refer to Eikvil's article [3, sec. 2.2] for further details.

G. Boundary Pixels

When the size of the whole image is large compared to the sliding window, the boundary region can be satisfactorily handled by extending the image with artificial pixels consisting of zeros, or by repetition or reflection of the image. This will, however, result in artifacts in the boundary region. In our application, this is not acceptable, since the size of the sliding window can be comparable to the image itself. Our treatment of the image borders consists of simply excluding from the computations pixels within the sliding window that are outside of the image. This slightly complicates the algorithm, since the total number of

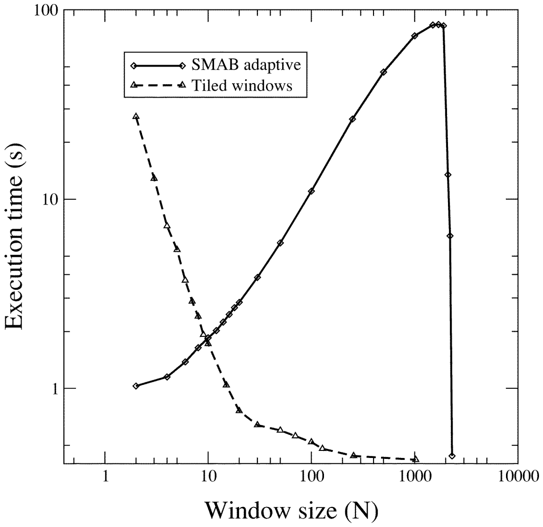


Fig. 2. Execution time of SMAB compared to that of an approximate adaptive binarization that uses interpolation between fixed tiles. Note logarithmic axes. Image size 1024 × 1024.

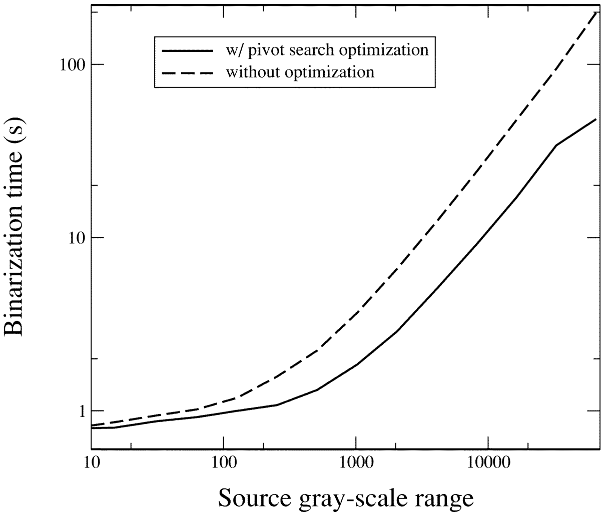


Fig. 3. Execution time for SMAB as a function of grayscale. The upper curve shows execution time for the algorithm without the pivot search optimization; the lower curve was obtained with the optimized algorithm. Note logarithmic scales. Refer to Section II for explanation of the optimization.

pixels in the histogram diminishes when the center pixel approaches the borders, reaching a minimum of $N/2 \times N/2$ for the pixels at the image corners. For $N = 2M$, the sliding-window method degenerates into a global transform.

III. Results

A. Binarization of Radiotherapy Images

The binarization of radiotherapy images is illustrated in Fig. 1. All images were taken from the same patient. The top row shows unprocessed images, from left to right: the digitally reconstructed radiograph (DRR), the digitized X-ray film from the radiotherapy treatment simulator, and the image acquired from an electronic portal imager during treatment. Bony structures yield ridge-like features corresponding to X-ray traversals near the periphery of compact bone. These ridges are well suited for the coregistration of the actual and planned images. As can be seen from the figure, the ridges stand out very prominently in the binarized images (bottom row).

B. Execution Time of SMAB

To be useful in a clinical setting, the binarization procedure must be fast. We have, therefore, investigated the execution time of SMAB. As predicted in Section II-G, the execution time of SMAB increases almost proportionally with window size N for $N/M \leq 0.8$, as shown in Fig. 2 (unbroken curve). When $N = 2M$, the sliding-window routine degenerates into global thresholding, and the execution time is at a minimum.

Also shown in Fig. 2 (broken curve) is the execution time for a tiled-windows adaptive binarization, using Otsu's method to determine the threshold within each tile. The execution time of the tiled-window binarization is proportional to $1/N^2$ for small N. As can be verified from the figure, there is a crossover point in execution time for the two methods. Tiled Otsu is most useful for large windows, and SMAB for small ones.

Using a progressively downscaled version of an original 16-bit image, we have also investigated the execution time of SMAB as a function of grayscale g. The results are shown in Fig. 3. The execution time increased first slowly by only a factor of 1.4 when the grayscale increased from 2 to 8 bits. Above 8 bits, the execution time rose more steeply and approached proportionality with grayscale range. We also found that the inclusion of the pivot search optimization resulted in approximately 30% reduction in total execution time.

C. SMAB Compared to Other Binarization Algorithms

For binarization of portal images, SMAB compares favorably with the adaptive binarization algorithms that are included in the extensive image processing library XITE [2], [7]. An example is shown in Fig. 4, where the result of SMAB of the image in Fig. 1(c) is shown in upper left, together with the results obtained with seven of the ten different adaptive methods in XITE. (The three XITE methods not shown either produced a totally black image, or did not terminate.) Detailed description of each method is found in the references cited in Table I. Only the method labeled Be came close to extracting the bone contours with contours comparable to that of SMAB. Table I also shows the parameter values used for each of the methods. Fine tuning of the parameters is not feasible for an automated procedure. In our experience, SMAB can be expected to yield acceptable results with a fixed parameter set, for a wide range of different images. For the XITE methods, extensive experimentation was required to find a combination of parameter values that resulted in a meaningful target image, and changes in the image (e.g., bit depth or window size) required readjusting the whole parameter set.

As explained in Section II, we consider the Otsu algorithm as a reference method for binarization. We find only very slight visual differences between medical images binarized using the Otsu algorithm within a sliding window and the corresponding images resulting from SMAB. To quantify the

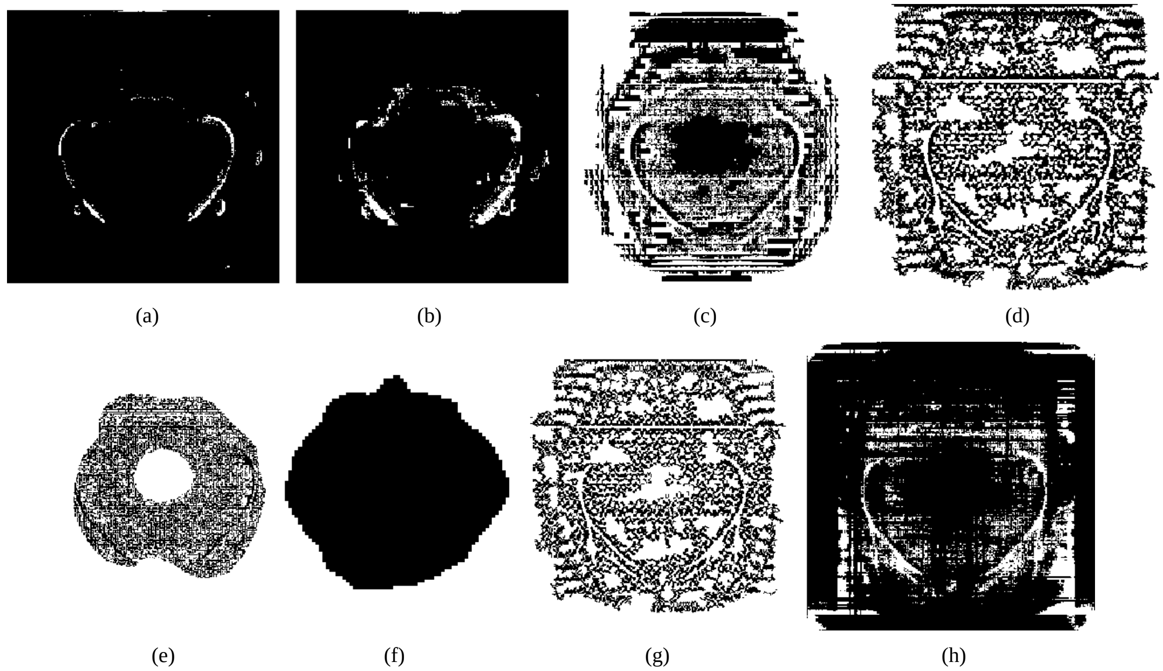


Fig. 4. Adaptive binarization of the example portal vision image. SMAB is upper left, followed by seven algorithms from the image-processing library XITE.

TABLE I
PARAMETER VALUES FOR THE EIGHT DIFFERENT ADAPTIVE BINARIZATION ROUTINES USED IN FIG. 4. IMAGE 256 × 256, 8 BITS. WINDOW SIZE WAS 12 PIXELS WHERE APPLICABLE. THE METHODS' ABBREVIATED NAMES ARE AS USED IN XITE

Method	SMAB	Be	ETM	IWR	Pa	TFJ	WR2	YB
Ref.		[1]	[3]	[16]	[10]	[14]	[16]	[17]
Params	-w 12 -c 100	-low - r12 -c 68	-r 12 -s 5 -l 13	-a 20 -ma 20 -m 3 -sl 100	-r 12 -k 0.2 -s 6	-r 12 -s 3 -t 10	-a 20 -ma 20 -m 3 -s 10 -sl 100 -l 1 -single	-b 0.1 -m 5 -t 50

difference between the methods, we tested binarization of a synthetic checkerboard image corrupted with additive uniform noise $U(0, W)$. The results are demonstrated in Fig. 5. The figure displays the fraction of misclassified pixels as a function of noise level W measured relative to the difference between black and white pixels in the uncorrupted image. Both methods fail markedly approximately when the noise level exceeds the black-and-white (b/w) separation, but the Otsu algorithm has a steeper breakdown curve than SMAB.

D. Other Applications of SMAB

SMAB is a general-purpose algorithm not particularly tied to radiotherapy imaging, and is suitable for binarization of a variety of images. As an example, scanned text documents can be binarized with excellent results. Fig. 6 shows an X-ray referral document scanned under suboptimal conditions, obtained in conjunction with a teleradiology service developed by one of the authors [13]. Creases, stains, and variations in illumination and handwriting density are virtually eliminated, and the resulting b/w image has uniform print density and excellent readability.

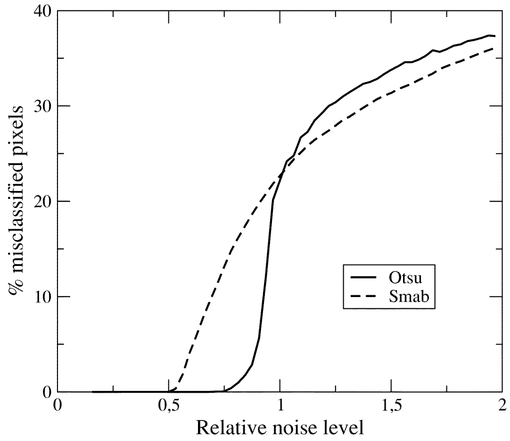


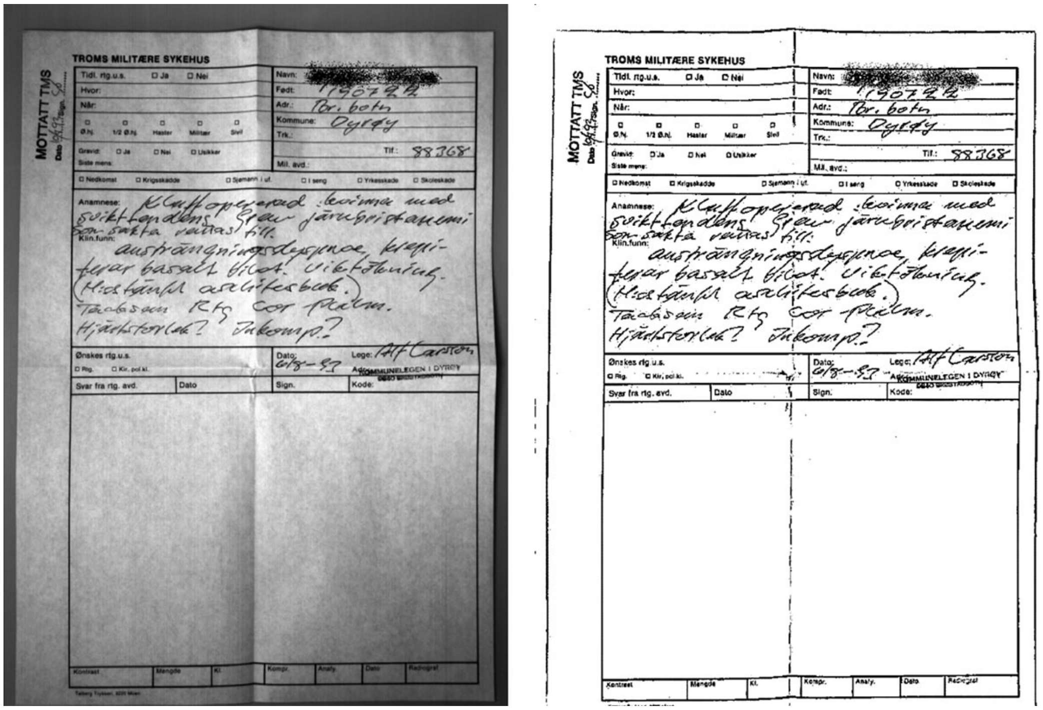
Fig. 5. Percentage of misclassified pixels using sliding-window Otsu and SMAB on a checkerboard b/w image corrupted with uniform additive noise.

IV. DISCUSSION

A. Automatic Coregistration of Medical Images

In our hospital department, every day, potentially 70–80 portal images are to be processed and coregistered with the corresponding simulator or DRR images. Visual inspection of the outcome of the automated registration is mandatory, and if the result can be accepted by the operator without need for further adjustment, large time savings are gained. Successful extraction of a sufficient amount of anatomical fiducial structures is a prerequisite in order to achieve this goal.

Our results show that sliding-window SMAB is robust to changes in the image, so that default parameters give acceptable results in most cases. SMAB may either be used as the only processing step, preferably for high-contrast images such as the simulator or DRR images, or as the final step in a processing sequence to enhance anatomical structures in the portal images.



MOTTATT TMS
Dato 10/8-97 Sign. Sø

TROMS MILITÆRE SYKEHUS

Tidl. rtg.u.s. □ Ja □ Nei
Hvor:
Når:
□ Ø.hj. □ 1/2 Ø.hj. □ Haster □ Militær □ Sivil
Gravid: □ Ja □ Nei □ Usikker
Siste mens:

Navn: [redacted]
Født: 190722
Adr.: Por. botn
Kommune: Dyrøy
Trk.:
Tlf.: 88368
Mil. avd.:

□ Nedkomst □ Krigsskadde □ Sjømann i ut. □ I seng □ Yrkesskade □ Skoleskade

Anamnese: Klaffopererad kvinna med sviktandlens. Grav järnbristanemi som sakta vaxas till.
Klin.funn: ansträngningsdyspnoe, krepiterar basalt bilat. Viktökning. (Hjärtsvikt avslutes bok.) Tacksam Rtg cor pulm. Hjärtstorlek? Inkomp.?

Ønskes rtg.u.s.
□ Rtg. □ Kir. pol.kl.
Dato: 6/8-97
Lege: Alf Carsson
Adr.: KOMMUNELEGEN I DYRØY

Svar fra rtg. avd. | Dato | Sign. | Kode:

Kontrast | Mengde | Kl. | Kompr. | Analy. | Dato | Radiograf

(a) (b)

Fig. 6. (a) Image of a handwritten X-ray referral document scanned under suboptimal conditions. Image size 1024 × 768. (b) Binarized with SMAB, window size 4 × 4.

B. Other Sliding-Window Applications

To our knowledge, Huang *et al.* [6] were the first to publish the idea of updating the median within a sliding window by stepwise iteration of the cumulative histogram, and they obtained execution times for the median filter that were much lower than for methods based on pixel sorting. Their ideas can readily be extended to other image processing functions based on the cumulative histogram, resulting in a fast adaptive version. As an example, we have implemented a sliding-window version of contrast limited adaptive histogram equalization (CLAHE) [11], [18], which is also in daily use in our department.

Availability

A software library written in C that includes SMAB and CLAHE is available on request from the authors.

Acknowledgment

Most of this work was carried out in the Department of Medical Physics and Technology at the Norwegian Radium Hospital, where Prof. A. Skretting gave invaluable assistance in reviews and discussions, both of content and of form. The authors appreciate the contribution of A. Kjølstad and P. Malm during early phases of this work. They also thank the anonymous referees for thorough reviews and pertinent comments.

References

[1] J. Bernsen, “Dynamic thresholding of grey-level images,” in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, Oct. 1986, pp. 1251–1255.

[2] S. Bøe, T. Lønnestad, and O. Milvang, “XITE user’s manual,” Image Processing Lab., Dept. Informatics, Univ. Oslo, Oslo, Norway, Rep. 56, Sept. 1998.

[3] L. Eikvil, T. Taxt, and K. Moen, “A fast adaptive method for binarization of grey level images,” in *Proc. Int. Conf. Document Analysis and Recognition*, vol. 1, 1991, pp. 435–443.

[4] K. Eilertsen and F. Van den Heuvel, “Image processing and image registration,” in *Proc. 6th Int. Workshop Electronic Portal Imaging*, vol. 6, June 2000, pp. 48-1–48-7.

[5] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison Wesley, 1993.

[6] T. S. Huang, G. J. Yang, and G. Y. Tang, “A fast two-dimensional median filtering algorithm,” *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. ASSP-27, pp. 13–18, Feb. 1979.

[7] T. Lønnestad and O. Milvang, “XITE—X-based image processing tools and environment,” in *Experimental Environments for Computer Vision and Image Processing*, H. I. Christensen and J. L. Crowley, Eds. Singapore: World Scientific, 1994, pp. 63–88.

[8] W. Niblack, *An Introduction to Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[9] N. Otsu, “A threshold selection method from grey-level histograms,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-1, pp. 62–66, Jan. 1979.

[10] J. R. Parker, “Gray level thresholding in badly illuminated images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 813–819, Aug. 1991.

[11] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, B. ter Haar Romeny, J. B. Zimmermann, and K. Zuiderveld, “Adaptive histogram equalization and its variations,” *Comput. Vis. Graph. Image Process.*, vol. 39, pp. 355–368, 1987.

[12] B. C. Stoel, A. M. Vossepoel, F. P. Ottes, P. L. Hofland, H. M. Kroon, and L. J. Schultze Kool, “Interactive histogram equalization,” *Pattern Recognit. Lett.*, vol. 11, pp. 247–254, 1990.

[13] T. Sund, "Mira—Teleradiology and digital radiology," *Telektronikk, Telenor Res. J.*, vol. 1993, no. 1, pp. 33–36, 1993.

[14] T. Taxt, P. J. Flynn, and A. K. Jain, "Segmentation of document images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1322–1329, Dec. 1989.

[15] O. D. Trier and T. M. Taxt, "Evaluation of binarization methods for document images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 312–315, Mar. 1995.

[16] J. M. White and G. D. Rohrer, "Image thresholding for optical character recognition and other applications requiring character image extraction," *IBM J. Res. Develop.*, vol. 27, no. 4, pp. 400–411, July 1983.

[17] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Comput. Vis. Graph. Image Process.*, vol. 46, no. 1, pp. 82–95, Apr. 1989.

[18] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems*, P. S. Heckbert, Ed. New York: Academic, 1994, vol. 4, pp. 474–485.