Using Mathematical Morphology for Document Skew Estimation

L. Najman
Laboratoire A2SI, Groupe ESIEE Cité Descartes, BP99
93162 Noisy-le-Grand Cedex France
l.najman@esiee.fr

ABSTRACT

We propose a concise definition of the skew angle of document, based on mathematical morphology. This definition has the advantages to be applicable both for binary and grey-scale images. We then discuss various possible implementations of this definition, and show that results we obtain are comparable to those of other existing algorithms.

1. INTRODUCTION

1.1. Foreword

We are going to apply the mathematical morphology formalism to document skew estimation. The use of mathematical morphology in the context of document image analysis allows us to give a clean definition of the skew angle of a document, through the definition of a new operator based on the closing by a segment and on the rose of directions.[1] Our definition, relying on a well-established formalism, is straightforwardly applicable both to binary and grey-scale images. We can then propose several different possible efficient implementations of this operator in different contexts.

1.2. Skew estimation and document

The automatic processing of document images is now widespread and is performed for a variety of reasons including, for example optical character recognition. Often there are problems in the automatic processing because the document image is skewed, and it is advisable to detect or estimate the skew angle, and correct the skew, before applying the further image processing.

Various techniques have been proposed for automatic skew detection in document images.[2] These are usually methods based on clustering of nearest neighbors, methods based on Hough transforms, or method involving projection profiles. This latest method is of interest for the comprehension of the one presented in this paper, and is thus described in the next section. Let us note that it is not straightforward to extend existing methods to grey-scale images. Numerous reviews can be found in the abundant existing litterature[3, 4] on skew estimation.

2. A WELL-KNOWN SKEW ESTIMATION ALGORITHM

A popular method for skew detection employs projection profiles. A projection profile is a histogram of the number of ON-pixel accumulated along parallel sample lines taken through the document. The profile may be at any angle, but often is taken horizontally along rows or vertically along columns, and these are called the horizontal and vertical projection profiles, respectively. For a document whose text lines span horizontally, the horizontal projection profile will have peaks whose widths are equal to the character height and mimimu height valleys whose widths are equal to the between-line spacing. For multicolumn documents, the vertical projection profile will have a plateau for each column, separated by valleys for the between-column and margin spacing.

The most straightforward use of the projection profile for skew detection is to compute it at a number of angles close to the expected orientation.[5] For each angle, a measure is made of the variation in the bin heights along the profile, and the one with the maximum variation gives the skew angle. At the correct skew angle, since scan lines are aligned to text lines, the projection profile has maximum height peaks for text and valleys for between-line spacing. Modifications and improvements can be made to this general technique to more quickly iterate to correct the skew angle and more accurately determine it.[6, 7]

All implementations of this kind of techniques rely on the maximization of a non-linear cost function. For instance, the cost function can be the sum of the squares of the successive differences of the projection profile bin heights in a given direction. No linear cost function can be chosen: if it were the case, we would compute a result proportional to the total number of all the black pixels, whatever the direction.

3. MATHEMATICAL MORPHOLOGY AND DOCUMENT

Founded in the sixties by G. Matheron and J. Serra, mathematical morphology is a theory for the analysis of spatial structures. It is called morphology because it aims at analysing the shape of unknown objects through the use of known shapes called structuring elements. A modern reference on this subject is the book of P. Soille.[8]

A structuring element is a small set used to probe the image under study. The shape and the size of the structuring element must be adapted to the geometric properties of the image objects to be processed. For instance, linear structuring elements are suited to the extraction of linear objects.

3.1. Basic morphological operators

The two basic morphological operators are the *dilation* and the *erosion*.

The *dilation* answers the question: *does the structuring element hit the set?* For a binary image X and a structuring element B, it is given by

$$X \oplus B = \{x | B_x \cap X \neq \emptyset\} \tag{1}$$

where $B_x = \{b + x | b \in B\}$ is the translated of B at point x, and for a grey-scale image f, by

$$(f \oplus B)(x) = \max_{b \in B} f(x + b) \tag{2}$$

The *erosion* is the dual of the dilation, and answers the question: *does the structuring element fit the set?* For a binary image X and a structuring element B, it is given by

$$X \ominus B = \{x | B_x \subseteq X\} \tag{3}$$

and for a grey-scale image f, by

$$(f \ominus B)(x) = \min_{b \in B} f(x + b) \tag{4}$$

Advanced operators can be derived by combining the dilation and the erosion. For instance, a closing is a dilation followed by an erosion. Under the name of RLSA, and for a linear structuring element, the closing enjoys a wide popularity in the document analysis community.

3.2. Run-Length Smoothing closing algorithm

Let L_0 be a horizontal segment of unit length. The Run-Length Smoothing closing algorithm $RLSA$[9] is a closing of the binary image X by γL_0:

$$RLSA(X) = X \oplus \gamma L_0 \ominus \gamma L_0 \tag{5}$$

where γ is a size parameter.

The $RLSA$ is a popular method for smoothing characters into smal, unrecognizable blobs. Assuming that the skew is corrected, the method merges characters into words, and words into text lines.

Using the morphological description of the RLSA, we can compute $RLSA$ on lines oriented at an angle α different from 0, or we can apply it to a grey-scale image f. For an image A (binary or grey-scale), we thus define the operator

$$RLSA_\alpha(A) = A \oplus \gamma L_\alpha \ominus \gamma L_\alpha \tag{6}$$

Note that computing $RLSA_\alpha$ is an idea quite similar to computing a projection profile at an angle α: we are looking for runs of pixels oriented in the α direction.

3.3. Rose of directions

Another useful operator is a kind of granulometry, in the sense that we are filtering our image according to a given family of structuring elements, and that we are computing a measure (namely the surface) on the resulting filtered image. In our case, we are not going to filter according to the *size* of the structuring element, but according to its *direction*.

Let L_α be a segment of unit length and of angle α with respect to the horizontal direction, and let $\beta \in \Re$ a size parameter. We define the following operator $\rho(\alpha)$:

$$\rho(\alpha) = Area(A \ominus \beta L_\alpha) \tag{7}$$

Ideally, the function ρ should be maximum at the predominant angle for the image. For a document image, this function should be maximum for the skew angle.

In the real world, however, it is difficult to extract from the function ρ applied directly on the original image useful information: except for very small β, the erosion will remove nearly all characters present on the document.

An operation related to equation 7 is the erosion not with L_α but with a pair of points P_α where

$$P_\alpha = \{(0,0)(\cos\alpha, \sin\alpha)\}\,. \tag{8}$$

Area of erosions by pair of points separated by a fixed distance but with varying orientations are sometimes represented in a polar diagram which is called a *rose of directions.*[1] This is the curve $(\rho(\alpha), \alpha)_{\alpha \in [0,360]}$.

Let us note that the morphological covariance K of an image F consists in measuring the volume (or the surface) of the image eroded by a pair of points P_α.

$$K(f; P_\alpha) = Vol(f \ominus P_\alpha(f)) \tag{9}$$

For binary images, it reduced to

$$K(A; P_\alpha) = Area(A \cap A_{(\cos(\alpha),\sin(\alpha))}) \tag{10}$$

which is the same as the rose of directions.

See[10, 11] for a recent industrial application on metal sheet. For binary images, the morphological covariance is the autocorrelation function. The correlation has been used for skew estimation, for instance using autocorrelation of projections. Correlation methods[12, 13] require large text regions on a page and takes more time to compute than methods like profile projection.

3.4. Morphological skew estimator

Equation 7 cannot be applied directly on the original document image, and we have to filter the image before applying the operator. According to what has been presented in section 3.2, the logical idea is to filter the image by $RLSA_\alpha$. Thus, we are going to compute

$$\nu(\alpha) = Area(RLSA_\alpha(A) \ominus \beta L_\alpha) \tag{11}$$

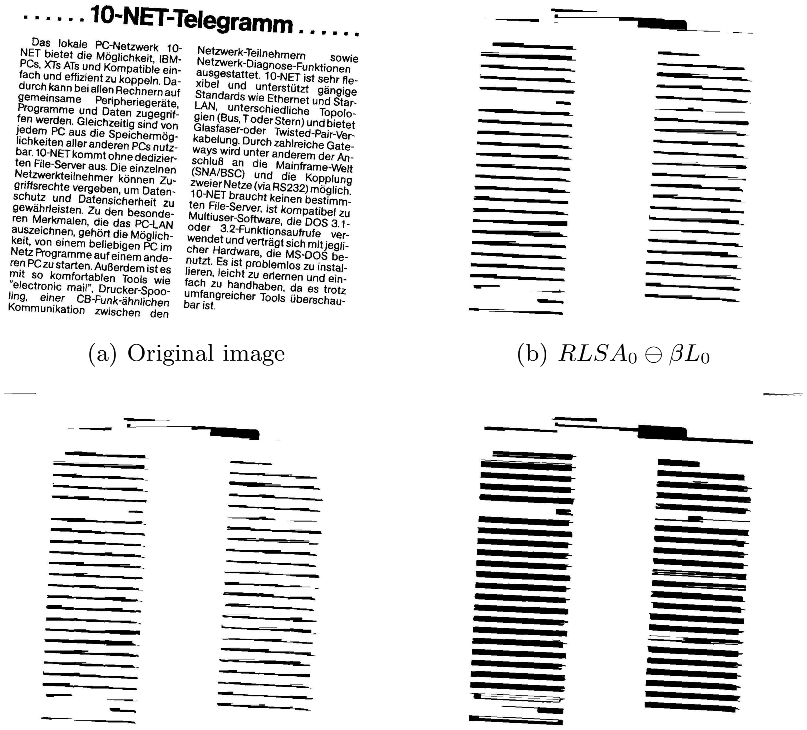
and the skew angle of image A is provided by

$$\begin{aligned} skew(A) &= \arg\max_\alpha \nu(\alpha) && (12)\\ &= \arg\max_\alpha Area(A \oplus \gamma L_\alpha \ominus (\gamma+\beta) L_\alpha) && (13) \end{aligned}$$

Let us note that for a precision of one degree, due to the grid structure of the image, the length $\gamma + \beta$ of the line structuring element should be greater than $\frac{1}{\tan 1} = 58$ pixels. Typically, the lenght $\gamma + \beta$ can be of the order of the length of a line of the document. The value of γ should be of the order of the size of a letter of the document or the size of the space between words, whichever is the greater.

The comparison of this formula with the profile projection method is enlightening. Indeed, the projection profile is replaced by a closing and the non-linear cost function necessary for the projection profile method is replaced by the erosion morphological operator, which is non-linear. So, what have we gained? We now have an elegant definition for the skew angle of document, relying on a well-founded theory; moreover, operators of this theory have been intensively studied from the point of view of algorithmics. Indeed a straightforward implementation of equation 11 is prohibitive. In the sequel of the paper we are going to discuss various possible efficient implementations of this formula. Another benefit is that this formula is also directly applicable on grey-scale images (replacing the area by the sum), which is not the case for most skew estimation algorithms.

We illustrate equation 11 on the image 1. Figure 1 shows the erosion of $RLSA_\alpha$ computed for various angles. We can see that the closer we are to the correct angle, the darker the image is.



...... 10-NET-Telegramm ......

Das lokale PC-Netzwerk 10-NET bietet die Möglichkeit, IBM-PCs, XTs ATs und Kompatible einfach und effizient zu koppeln. Dadurch kann bei allen Rechnern auf gemeinsame Peripheriegeräte, Programme und Daten zugegriffen werden. Gleichzeitig sind von jedem PC aus die Speichermöglichkeiten aller anderen PCs nutzbar. 10-NET kommt ohne dedizierten File-Server aus. Die einzelnen Netzwerkteilnehmer können Zugriffsrechte vergeben, um Datenschutz und Datensicherheit zu gewährleisten. Zu den besonderen Merkmalen, die das PC-LAN auszeichnen, gehört die Möglichkeit, von einem beliebigen PC im Netz Programme auf einem anderen PC zu starten. Außerdem ist es mit so komfortablen Tools wie "electronic mail", Drucker-Spooling, einer CB-Funk-ähnlichen Kommunikation zwischen den Netzwerk-Teilnehmern sowie Netzwerk-Diagnose-Funktionen ausgestattet. 10-NET ist sehr flexibel und unterstützt gängige Standards wie Ethernet und Star-LAN, unterschiedliche Topologien (Bus, T oder Stern) und bietet Glasfaser-oder Twisted-Pair-Verkabelung. Durch zahlreiche Gateways wird unter anderem der Anschluß an die Mainframe-Welt (SNA/BSC) und die Kopplung zweier Netze (via RS232) möglich. 10-NET braucht keinen bestimmten File-Server, ist kompatibel zu Multiuser-Software, die DOS 3.1-oder 3.2-Funktionsaufrufe verwendet und verträgt sich mit jeglicher Hardware, die MS-DOS benutzt. Es ist problemlos zu installieren, leicht zu erlernen und einfach zu handhaben, da es trotz umfangreicher Tools überschaubar ist.

(a) Original image

(b) $RLSA_0 \ominus \beta L_0$

(c) $RLSA_1 \ominus \beta L_1$

(d) $RLSA_{-3} \ominus \beta L_{-3}$

Figure 1. Run-Length Smoothing Closing eroded at several angles for a document with skew of -3 degree

Figure 2 displays an example of the curve we maximize to obtain the correct skew. The maximum of this curve is at the correct skew angle.

3.5. Morphological algorithm for skew estimation

The naive method is to evaluate $\nu(\alpha)$ at many α in a global search. This is expensive to compute for typical document images, containing more than 10^6 pixels. Thus, a traditional optimization algorithm that requires fewer function evaluations is applied. Given a initial bracketing interval, Brent's method[14] returns the position of the maximum to within a specified tolerance. It is advantageous to subsample the image prior to the optimization search. Downsampling and other implementation issues are discussed in section 5.

Brent's method is a kind of parabolic interpolation in which the values of six parameters a, b, u, v, w and x, are monitored. The parameters a and b are the limits of a bounding interval in which the minimum is located,

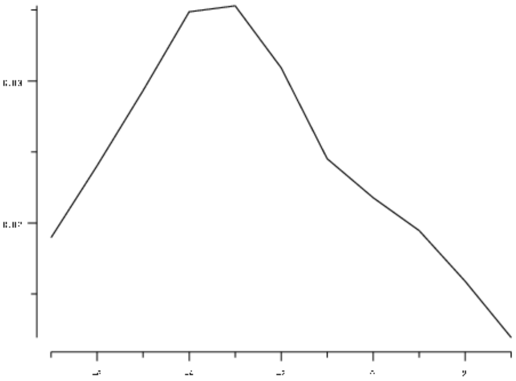


Figure 2. Curve of $\nu(\alpha)$ for $\alpha \in [-7, 3]$ for a document with skew of -3 degree. The maximum of the curve is at -3 degree.

x is the point with the lowest function value found so far, w is the point with the second lowest function value found so far, u is the point at which the function was evaluated most recently, and v is the previous value of w. The method is iterative.

According to Brent's method, parabolic interpolation is attempted fitting through the points x, v and w. In order to be considered acceptable, the current parabolic-interpolation step must (i) produce a new minimum which falls within the bounding interval (a, b), and (ii) imply a movement (amount of change) from the best current value, x, that is less than half the movement of the step before last. This second criterion ensures that the successive steps of the method will lead to convergence. In the worst case, where successive steps approximately alternate between parabolic steps and golden sections, there will ultimately be convergence thanks to the golden sections.

4. IMPLEMENTATIONS

The raw algorithm for computing erosion or dilation consists in calculating the minimum or maximum of pixels in the structuring element for each pixel of the image. There are $(n-1)$ min/max comparisons per image pixel for a structuring element of n pixels. This number can be drastically reduced by appropriate algorithms and data structures. This section describes the best algorithms adapted to our problem.

4.1. Binary implementations

4.1.1. Constant time implementation

There exists several possibilities for binary implementation of dilation and erosion. For instance, one can use the method described by Nadadur and Haralick,[15] which uses bresenham lines and requires only one maximum operation for erosion and one minimum operation for dilation transform, and one thresholding step and one translation step per result pixel. Such algorithms run at constant time with all possible length line structuring elements. However, we can find implementations requiring less than one operation per result pixel. This is the object of the next section.

4.1.2. Logarithmic decomposition

A logarithmic decomposition of convex sets has been proposed by Boomgaard and van Balen,[16] based on a definition of extreme sets of a convex set given by Pecht.[17] Moreover, since w pixels of a binary image can be stored in a w-bit data type word, a logical operator can be simultaneously applied to w pixels using a bitwise operator. For the completeness of this paper, we are going to provide the main idea.

It is a well known fact in mathematical morphology that convex sets are divisible with respect to dilation. Let n be an integer, and S a convex set. We define

$$nS \quad = \quad \underbrace{S \oplus \ldots \oplus S}_{n \text{ terms}}$$

nS has the same shape as S but n times larger. Using the associativity of the dilation, $X \oplus nS$ can be computed by s succesive dilation by S. This is called *linear decomposition.* We can go further if we suppose that S is a two-dimensional convex symmetric set. In this case the linear decomposition can be replaced[17] with

$$nS \quad = S\oplus \quad \underbrace{E(S) \oplus \ldots \oplus E(S)}_{(n-1) \text{ terms}}$$

where $E(S)$ is the extreme set of S. For discrete sets, the extreme set $E(S)$ contains just the vertices of the continuous convex hull of S. The linear decomposition can furthermore be replaced by the more efficient *logarithmic decomposition.* Using the extreme set, we have $2S = S \oplus E(S)$, and by associativity $4S = 2S \oplus E(2S) = S \oplus E(S) \oplus E(2S)$, and so on. In the general case for arbitrary n, we have

$$nS = S \oplus E(S) \oplus E(2S) \oplus \ldots \oplus E(2^{m-1}S) \oplus E((n - 2^{m-1})S) \tag{14}$$

where $m = [\log_2(n)]$.

If we apply that formula to our case, as an example we have

$$8L_0 = \{\bullet \bullet \bullet\} \oplus \{\bullet \circ \bullet\} \oplus \{\bullet \circ \circ \circ \bullet\} \oplus \{\bullet \circ \circ \circ \circ \circ \circ \circ \bullet\} \tag{15}$$

which has a total of 9 pixels instead of the 19 pixels of $8L_0$.

For an efficient algorithmic implementation, we are using a bit-mapped representation of binary images. Each addressable word of memory/CPU contains 32 (or 64) horizontally consecutive pixels. Furthermore, morphological operations such as dilation or closing can be computed in parallel, leading to fast algorithms. Indeed, a dilation is the logical OR operation, and an erosion is the logical AND operation. The precise description of this implementation can be found in the paper by Boomgaard and van Balen.[16]

Thus, knowing that the surface can be efficiently computed by a hash table in the last step, one step of the skew estimation is computed in $O((\log(\gamma) + \log(\gamma)\log(\beta))\frac{nm}{w})$ where γ and β are the parameters of the rose of directions, nm is the number of pixels of the image, and w is the size of a machine word.

4.1.3. Fourier implementation

On dedicated hardware, it is possible to implement morphological dilation and erosion using a Fourier transform.[18, 19] Let us give the basic idea. The *indicator function* k_A of a set A is given by $k_A(x) = 1$ if $x \in A$ and $k_A(x) = 0$ otherwise. The *convolution* of a function f by a function c is defined by $c \star f(x) = \sum_u c(u)f(x-u)$. The relationship between mathematical morphology and convolution lies in the following equation

$$k_A \star k_{\check{K}}(x) = Area(A \cap K_x) \tag{16}$$

where $\check{K}$ is the reflected set of K with respect to the origin. We have

$$A \oplus K \quad = \quad \{x | k_A \star k_{\check{K}}(x) > 0\} \tag{17}$$

$$A \ominus K \quad = \quad \{x | k_A \star k_{\check{K}}(x) = Area(K)\} \tag{18}$$

As computing a convolution is easier using Fourier transform, using special processors which are currently commercially available based on the Fast Fourier Transform, the skew estimation algorithm described in this paper can be straightforwardly implemented on such processors.

4.2. Grey-scale implementation

Our formula for skew estimation can directly be used on grey-scale images, prior to binarization. We thus have to look for implementations of min/max filter for grey-scale images, and several of them can been found that run in constant time. For linear structuring elements, van Herk[20] proposed a recursive algorithm requiring 3 min/max comparisons per image pixel whatever the length of the line segment. This algorithm has been extended[21] to bresenham lines at arbitrary angles. However, this algorithm does not give the exact result, and Soille and Talbot[22] showed that an exact translation invariant implementation of min/max filters along a line segment of slope in the form of an irreducible fraction dy/dx can be achieved at the cost of $2 + k$ min/max comparisons per image pixel, where $k = \max(|dx|, |dy|)$.

5. TESTING THE ALGORITHM AND COMPARISON WITH OTHERS SKEW DETECTION METHODS

The tests we made (using the logarithmic decomposition implementation) validate the skew estimation, for a global computation time less than $0, 75$ seconds on a Pentium III 733 Mhz for a 1214x1151 pixels image, including call to the wrapping code necessary to our development framework. This timing is comparable to timing of commercial packages or existing algorithms, albeit slower than some. Let us note that the test implementation is quite generic and portable, and thus does not take advantage of any processor-dedicated optimization.

We have to estimate the quality of the result of the morphological estimator, and to compare its accuracy with respect to other estimators. As document database, we have chosen the *UW English Document Image Database - I.*[23] This database contains 979 scanned images. For each of the scanned images, an estimate of the actual image skew is provided in the database, with an expected error of from 50 to 100 milli-degrees.

5.1. Choice of estimator for comparison

To our knowledge, there exist two methods using morphological tools as an important brick of the estimation.

- Chen and Haralick[24] used a more involved method that started with threshold reduction,[25] applied recursive morphological closings and openings to close up textlines and remove ascenders and descenders, determined connected components, fit the best line to the points in each set of connected components, and estimated a global skew by discarding outlier lines and averaging the remaining ones. The skew error reported was greater than 0.3 degrees on about 10 percent of the images in the UW English Document Image Database.[23]
- Das and Chanda[26] propose to use $RLSA_0$, *i.e.* a closing of the image using line structuring element. That forms solid black bands corresponding to text lines. They open the closed image with small square structuring element to remove bumps. Then they register all 0 to 1 transitions, which provide lines. The skew is computed from the biggest lines. The method is reported to work for small skew angles.

One can note that contrary to our method, none of these methods is fully morphological We will not compare with those methods.

D. Bloomberg, both in a paper[7] and on his web site,[27] claims that notwithstanding the size of skew estimation bibliography, there is only one method that can be recommended for both speed and accuracy. It is based on the maximization of the *differential skew function*, and is a variant of the projection profile method.

The sum for the i^{th} sloped line at angle α is denoted by $s(i, \alpha)$, with

$$s(i, \alpha) = \sum_{\text{pixels in}(i,\alpha)\text{scan},j\in w} p(i, j) \tag{19}$$

$$S(\alpha) = \sum_{\text{scanlines}} s^2(i, \alpha) \tag{20}$$

The differential skew function $S_\delta(\alpha)$ is defined as the sum of squares of differential counts on adjacent lines:

$$S_\delta(\alpha) = \sum_{scanlines} (s(i,\alpha) - s(i-1,\alpha))^2 \tag{21}$$

As noted by Bloomberg,[7] in contrast to the sum skew function $S(\alpha)$, the width of the differential function can be much less than the textline asperity ratio. D. Bloomberg provides an excellent open source C implementation of this method.[27]

The differential skew estimator is reported to work well on the *UW English Document Image Database - I* and we are going to compare this method with the morphological skew estimator.

5.2. Experiment and measurement on scanned images

It is well known that the assessment of skew measurement performance on scanned images is problematic. The first problem is that the actual image skew is not known. The approach taken with the UW database was to measure the skew of each image manually. Unfortunately, as noted in,[7] *observation of many of the scanned images indicates that there is no single global skew. The most common scan artifact is that the skew at the top and bottom of the image differ, often by two hundred milli-degrees or more. There are also a few "outliers" where the measured angle is far from correct. Significant measured bias is occasionally induced by text fragments with non-conforming skew imaged from adjacent book pages.*

For each of the 979 images of the UW database, we compute both the result of the morphological skew estimator, and the result of the differential skew estimator, and we compare those results with the "ground-truth" from the UW database. For the morphological skew estimator, we obtain the best results with parameters 64 for the dilation and 512 for the erosion, which corresponds respectively to the space between word and to the size of a line of text. The morphological skew estimator has a mean absolute error of 0.2 and a root-mean-square of 0.25 with respect to the ground-truth, while the differential skew estimator has a mean absolute error of 0.15 and a root-mean-square of 0.25. By downsizing the images by a factor 2, we obtain for the morphological skew estimator a mean absolute error of 0.25 and a root-mean-square of 0.31.

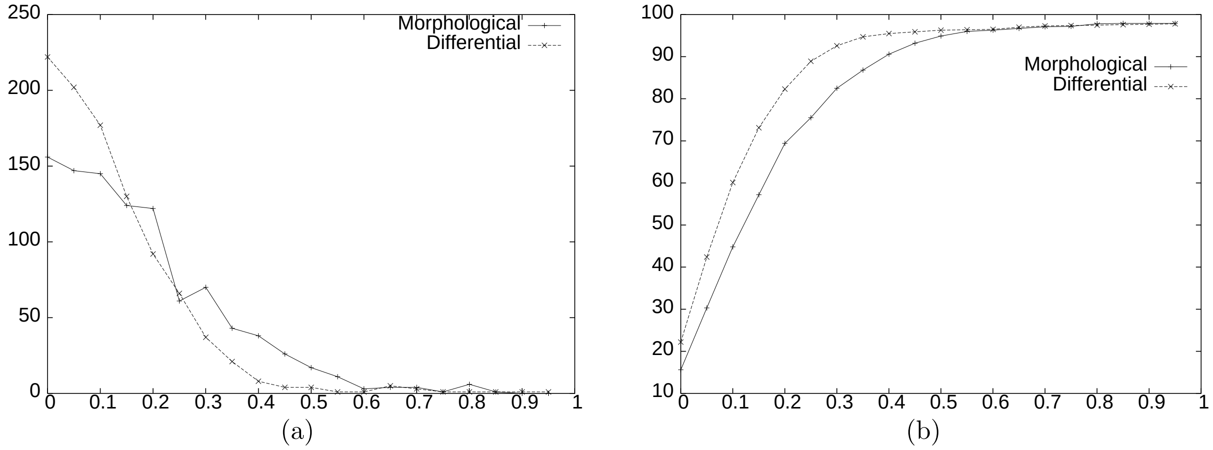


Figure 3. Histogram (a) and cumulative histogram (b) of absolute errors with respect to the estimated "ground-truth" on the UW database both for the differential skew estimator and for the morphological estimator. 20 bins were used for the histogram and 50 bins for the cumulative histogram.

Figure 3 gives the graph of the histogram and the cumulative histogram of absolute errors with respect to the estimated ground-truth on the UW database, both for the differential skew estimator and for the morphological skew estimator. One can see that the differential skew estimator behaves slightly better than the morphological skew estimator. We believe that part of the difference between the two methods lies in the fact that the two implementations have different optimization methods.

6. CONCLUSION

This paper demonstrates how to use mathematical morphology to accurately and rapidly estimate the skew angle of a document image. Testing shows that this method is competitive with the ones existing on the market.

The benefits of this paper are the following:

- We now have a clean mathematical definition of the skew angle of document as a function of morphological operators, that works both for binary and grey-scale images.
- We can find algorithmic implementations that are competitive, both in term of speed and precision.

Further testing is needed to assess if the accuracy of the skew estimation is enhanced if we apply our formula directly on grey-scale images, without any binarization of the document image.

Part of this work has been done while the author was with Océ Print Logic Technologies, 1, rue Jean Lemoine 94105 Créteil France. Patent Pending.

REFERENCES

1. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
2. L. O'Gorman and R. Kasturi, *Executive briefing: document image analysis*, IEEE Computer Society Press, 1997. ISBN 0-8186-7802-X.
3. A. Amin, S. Fisher, A. Parkinson, and R. Shiu, "Comparative study of skew detection algorithms," *Journal of Electronic Imaging* **5**, pp. 443–451, October 1996.
4. O. Okun, M. Pietikäinen, and J. Sauvola, "Document skew estimation without angle restriction," *International Journal on Document Analysis and Recognition* **2**, pp. 132–144, 1999.
5. W. Postl, "Detection of linear oblique structures and skew scan in digitized documents," in *Proc. 8th Int'l Conf. Document Analysis and Recognition (ICPR)*, pp. 687–689, IEEE CS Press, (Los Alamitos, Calif), 1986.
6. H. Baird, "The skew angle of printed documents," in *Proc. Conf. Society of Photographic Scientists and Engineers*, pp. 14–21, 1987.
7. D. Bloomberg, G. Kopec, and L. Dasari, "Measuring document image skew and orientation," in *IS&T/SPIE EI'95, Conference 2422: Document Recognition II*, pp. 302–316, (San Jose, CA), Feb 1995.
8. P. Soille, *Morphological Image Analysis*, Springer-Verlag, 1999.
9. K. Wong, R. Casey, and F. Wahl, "Document analysis system," *IBM J. Res. Develop.* **26**(2), pp. 647–656, 1982.
10. A. Tuzikov, P. Soille, D. Jeulin, H. Bruneel, and M. Vermeulen, "Extraction of grid lines on stamped metal pieces using mathematical morphology," in *Proc. 11th IAPR International Conference on Pattern Recognition, COnference A: Computer Vision and Applications*, **1**, pp. 425–428, (The Hague), 1992.
11. R. Peyard, P. Soille, J.-C. Klein, and A. Tuzikov, "A dedicated hardware system for the extraction of grid patterns on stamped metal sheets," in *Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, I. Pitas, ed., pp. 867–870, (Neis Marmaras), 1995. URL: http://poseidon.csd.auth.gr/Workshop/-papers/p_34_3.html.
12. H. Yan, "Skew correction of document images using interline cross-correlation," *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing* **55**(6), pp. 538–543, 1993.
13. A. Chaudhuri and S. Chaudhuri, "Robust detection of skew in document images," *IEEE Trans. on Image Processing* **6**(2), pp. 344–349, 1997.
14. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*, ch. Brent, pp. 283–286. Cambridge University Press, 1989.
15. D. Nadadur and R. Haralick, "Recursive binary dilation and erosion using digital line structuring elements in aritrary orientations," *IEEE Transactions on Image Processing* **9**(5), pp. 749–759, 2000.

16. R. van den Boomgaard and R. van Balen, "Methods for fast morphological image transform using bitmapped binary images," *CVGIP: Graphical Models and Image Processing* **54**(3), pp. 252–258, 1992.
17. J. Pecht, "Speeding up successive minkowski operations," *Pattern Recognition Letters* **3**(2), pp. 113–117, 1985.
18. J. Mazille, "Mathematical morphology and convolution," *Journal of Microscopy* **156**, pp. 3–13, October 1989.
19. M. Killinger, J. de Bougrenet de la Tocnaye, P. Cambon, and C. L. Moing, "Morphological filtering using a fourier transform hologram," *Optics Communications* **73**, pp. 434–438, November 1989.
20. M. van Herk, "A fast algorithm for local minimum and maximum filters on rectangular and orthogonal kernels," *Pattern Recognition Letters* **13**, pp. 517–521, 1992.
21. P. Soille, E. Breen, and R. Jones, "Recursive implementation of erosions and dilations along discrete lines at arbitrary angles," *IEEE Trans. on PAMI* **18**(5), pp. 562–566, 1996.
22. P. Soille and H. Talbot, "Directional morphological filtering," *IEEE Trans. on PAMI* **23**(11), pp. 1313–1329, 2001.
23. S. Chen, M. Jaismha, J. Ha, I. Phillips, and R. Haralick, "Uw english document image database - (i)," tech. rep., University of Washington, 1993.
24. S. Chen and R. M. Haralick, "An automatic algorithm for text skew estimation in document images using recursive morphological transforms," in *ICIP-94*, pp. 139–143, (Austin, TX), November 1994.
25. D. Bloomberg, "Image analysis using threshold reduction," in *SPIE Conf. on Image Algebra and Morphological Image Processing II*, **1568**, pp. 38–52, (San Diego, CA), July 1991.
26. A. Das and B. Chanda, "A fast algorithm for skew detection of document images using morphology," *International Journal on Document Analysis and Recognition* **4**, pp. 109–114, 2001.
27. D. Bloomberg, "Leptonica web pages." http://www.leptonica.com/, 2003. Offers an image processing software with a highly unrestricted form of copyright license.