Open access • Proceedings Article • DOI:10.1109/ICFHR.2012.236

Moment-Based Image Normalization for Handwritten Text Recognition

— Source link

Michal Kozielski, Jens Forster, Hermann Ney

Institutions: RWTH Aachen University

Published on: 18 Sep 2012 - International Conference on Frontiers in Handwriting Recognition

Topics: Intelligent character recognition, Normalization (image processing), Handwriting recognition, Intelligent word recognition and Image moment

Moment-based Image Normalization for Handwritten Text Recognition

Michał Kozielski, Jens Forster, Hermann Ney
Human Language Technology and Pattern Recognition Group
Chair of Computer Science 6
RWTH Aachen University, D-52056 Aachen, Germany

{*kozielski,forster,ney*}@*i6.informatik.rwth-aachen.de*

Abstract

In this paper, we extend the concept of moment-based normalization of images from digit recognition to the recognition of handwritten text. Image moments provide robust estimates for text characteristics such as size and position of words within an image. For handwriting recognition the normalization procedure is applied to image slices independently. Additionally, a novel moment-based algorithm for line-thickness normalization is presented. The proposed normalization methods are evaluated on the RIMES database of French handwriting and the IAM database of English handwriting. For RIMES we achieve an improvement from 16.7% *word error rate to* 13.4% *and for IAM from* 46.6% *to* 37.3%*.*

1. Introduction

Text in handwritten images typically shows strong variability in appearance due to different writing styles. Appearance differs in the size of the words, slant, skew and stroke thickness. Such variability calls for the development of normalization and preprocessing techniques suitable for recognition of handwritten text. Among the most common preprocessing steps applied in current state-of-the art systems are noise removal, binarization, skew and slant correction, thinning, and baseline normalization [3]. For slant correction, Pastor et al. [17] proposed to use the maximum variance of the pixels in the vertical projection and Vinciarelli et al. [21] observed that non-slanted words show long, continuous strokes. Juan et al. [20] showed that normalizing ascenders and descenders of the text reduces significantly the vertical variability of handwritten images. A linear scaling method applied to whole images has been used in various systems to reduce the overall size variability of images of handwritten text [6, 3, 8]. A drawback of all those approaches is that they rely on assumptions that may or may not hold for a given database. A second drawback is that all those methods are applied to whole images making it difficult to address local changes. Furthermore, the methods for slant correction rely on binarization which is a non-trivial problem in itself and should be avoided if possible, as Liu et al. [13] found in their benchmark paper. Recently España-Boquera et al. [7] proposed using trained Multi-Layer-Perceptrons for image cleaning and normalization. While they report competitive results on standard databases, the training and labeling process is time consuming. In contrast to the methods mentioned until now, methods based on image statistics and moments do not suffer from heuristical assumptions and have been extensively studied in the area of isolated digit recognition. Casey [4] proposed that all linear pattern variations can be normalized using second-order moments. Liu et al. [14] used Bi-moment normalization based on quadratic curve fitting and introduced a method to put a constrain on the aspect ratio when the x and y axis are normalized independently [12]. Miyoshi et al. [16] reported that computing the moments from the contour of a pattern, and not from the pattern itself, improves the overall recognition results.

We propose a moment-based normalization scheme for handwritten images. We use the image gradient and zero-th order moments to globally normalize the stroke thickness of a pattern. The algorithm operates directly on grey-scale images and is not susceptible to local distortions. The image is segmented into slices using a sliding window and size and shift of the sliding window are estimated using moments. Finally, local variability in size and position is modelled independently in separate slices using second-order moments.

2. Normalization scheme

Consider a grey-scale image $f(x,y) : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ of width W and height H and pixels values in the range $0 - 255$.

Geometric moments of a p+qth order of f are given by:

$$m_{pq}[f] = \sum_x \sum_y x^p y^q f(x,y) \tag{1}$$

From now on we omit the bracket $[f]$ when its clear to which function we refer. The central moments are given by:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{2}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$ are the coordinates of the centre of gravity of an object contained in this image. The second-order moments μ_{20} and μ_{02} reflect how much pixels deviate from the center of gravity. We interpret them as the size of the object in x and y direction independently.

Image moments give us important information about the structure and density of the object and form a basis for normalization algorithms described in this section.

2.1. Stroke thickness normalization

Images of handwritten text usually vary in the thickness of strokes, which correspond to a different pressure applied to a pen. Therefore a stroke thickness normalization procedure that reduces this variability would be of our high interest. We denote the normalized grey-scale image as $f'(x,y) : \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$.

Let us consider a shape that resembles a long, thin, straight stroke. We assume that this shape has some dimension τ, to which we refer as a stroke thickness of that shape. We further assume that τ is constant throughout the whole shape and we make τ a subject of a normalization procedure. We define the thickening as an operation that linearly increases the value τ and express it by means of morphological dilation with a structuring element of a radius r.

$$f'(x,y) = \max_{r_x, r_y : d(r_x, r_y) < r} f(x + r_x, y + r_y) \qquad for \quad r \geq 0 \tag{3}$$

with $d(r_x, r_y)$ being the Manhattan distance from the center of the structuring element. For negative values of r we express this operation by means of morphological erosion.

$$f'(x,y) = \min_{r_x, r_y : d(r_x, r_y) < -r} f(x + r_x, y + r_y) \qquad for \quad r < 0 \tag{4}$$

Rivest [19] defined the image gradient $g(f)$ as the difference between the dilation and erosion of that image with a structuring element of a radius ρ. We observe that a sum over all values of f' is proportional to the area of a thickened shape. We refer to it as $m_{00}[f']$, recalling the geometric moment definition. Furthermore we observe that a sum over all values of the image gradient $g(f)$ is proportional to the change of that area. We refer to it as $m_{00}[g(f)]$. If we apply the thickening operation to the shape with some radius r, the value of $m_{00}[f']$ will increase linearly with respect to r and the increase will be proportional to τ. Following this observations we compute the stroke thickness of the shape as:

$$\tau = 2\rho \frac{m_{00}[f']}{m_{00}[g(f)]} \tag{5}$$

Note that in case of images it is not possible to compute the gradient for $\rho \to 0$. Therefore we use the smallest value that does not require interpolation, which is 1.

Figure 1 shows the plot of the values $m_{00}[f']$ and $m_{00}[g(f')]$ computed on an image thickened with a structuring element of radius r. If we now treat the moment m_{00} as a function of r, we can observe that its characteristic deviates from the one of a linear function for real-world images, as our assumption about the geometrical structure of the object is only a rough estimate of handwriting shapes. The dilation or erosion operation creates an effect similar to the median filter and therefore reduces the overall gradient and the increase rate of $m_{00}[f']$. Therefore we use the moment of the original gradient $m_{00}[g(f)]$ in computation of the stroke thickness, because it gives the best estimate of the change in the area under the thickening operation.

If we now denote the stroke thickness of the normalized image f' as T, which is a parameter to be optimized, the normalization procedure is equivalent to dilating (eroding) the image f with a radius $r = T - \tau$. The value of r is real therefore we have to interpolate the image appropriately.

To further overcome the deviation of $m_{00}[f'](r)$ from the characteristic of a linear function we apply an iterative algorithm in which we recompute the value τ after dilating (eroding) the image. If the condition $T - \tau < \epsilon$ is not met, with ϵ being a certain, small threshold, the value of r is reestimated and the normalization step is repeated.

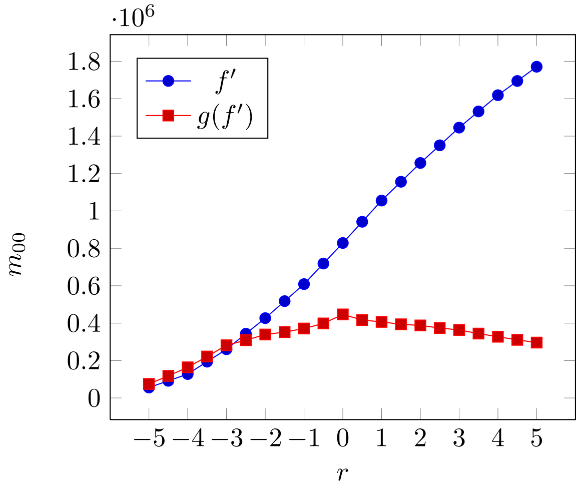
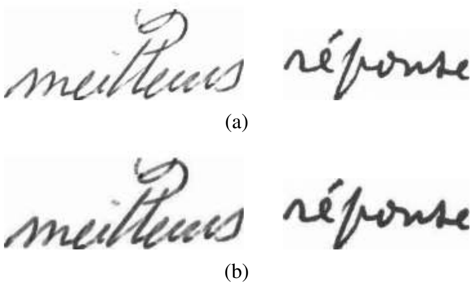


Figure 1: Plot of the $m_{00}[f']$ and $m_{00}[g(f')]$ with respect to r for a sample image

Note that erosion is not the inverse of dilation, therefore special considerations have to be made during implementation so that the final oscillation over T does not degrade the image.

Figure 2 shows sample images with different stroke thickness and their normalized versions. The dilation (erosion) operation will also degrade the quality of the image if the parameter T is too small or too big, which is a natural effect. Therefore it is crucial to find a correct value for T during optimization.



(a)

(b)

Figure 2: a) Sample images and b) their thickness-normalized versions

2.2. Segmentation

Consider a grey-scale image f of height H and width W which contains one line of text. The image is usually cropped from a page of multiple lines of handwritten text. The crop points can be affected by many factors such as the size of ascenders and descenders, artefacts, and segmentation errors. It is then possible as seen on the figure 3, that the image contains too much or too little whitespace and that the image height does not reflect the actual size of the text baseline. We then need to reestimate the height of the image from the actual text characteristics. We could use the original second-order moments, but the their definition implies that the distance computed between pixels is squared and therefore highly influenced by outliers. So we alter the computation formula of the vertical moment μ_{02} and define the moment ν, which uses the absolute distance instead.

$$\nu = \sum_x \sum_y |y - \bar{y}| f(x, y) \tag{6}$$

The height of the image is then recomputed by:

$$H' = \beta \frac{\nu}{m_{00}} \tag{7}$$

The parameter β is merely for convenience and is chosen in such a way that the average H is equal to average H' across a given corpus. The value of H' is a better estimate of the vertical dimension of the image as it depends on the density of the image. Figure 4 shows an illustration of reestimated image height. Note that we do not crop the image in any way, but we only use the value of H' for the estimation of segmentation parameters.

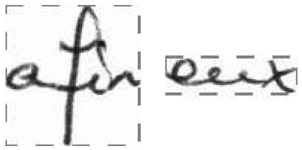
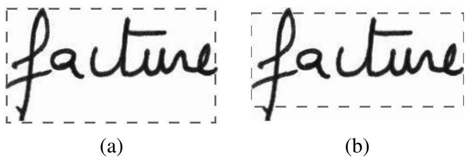


Figure 3: Illustration of different bounding boxes of images with the same size of the baseline text

We segment the image with a sliding window of height H, width $\gamma_1 H'$, and shift $\gamma_2 H'$. By relating the size and shift of the sliding window to the image height we ensure that different scaling of the original image does not influence the aspect ratio and the quantity of single slices. The crop points of a given slice are real values and have to be rounded to natural values. We then apply a horizontal cosine window to the slice in order to smooth the borders. All slices segmented from a single image are of the same size.



(a)

(b)

Figure 4: a) Original and b) reestimated bounding box of a sample image

2.3. Size and translation normalization

Let us use h to refer to a single slice of width W_1 and height H_1. We are now interested in a normalization procedure that will allow us to normalize every slice with respect to size and translation independently.

We reestimate the area subject to scaling using moments:

$$\begin{cases} \delta_x = \alpha\sqrt{\frac{\mu_{20}[h]}{m_{00}[h]}} \\ \delta_y = \alpha\sqrt{\frac{\mu_{02}[h]}{m_{00}[h]}} \end{cases} \tag{8}$$

where δ_x and δ_y are the new horizontal and vertical dimensions of the slice. We use $\alpha = 4$ for our experiments.

Let us denote the normalized grey-scale image of width W_2 and height H_2 as $h'(x', y')$. The normalization procedure that maps the normalized image to the original image is implemented by the following backward mapping:

$$\begin{cases} x = (\frac{x'}{W_2} - \frac{1}{2})\delta_x + \bar{x} \\ y = (\frac{y'}{H_2} - \frac{1}{2})\delta_y + \bar{y} \end{cases} \tag{9}$$

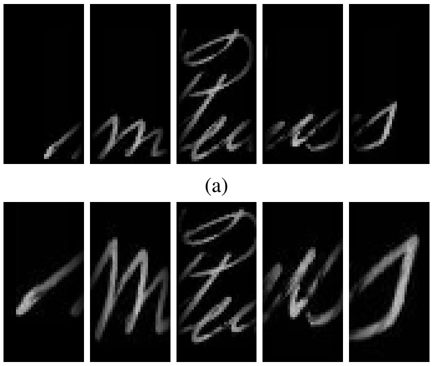
This procedure not only resizes the image but also shifts the center of gravity to the center of the image $[\frac{W_2}{2}, \frac{H_2}{2}]$. We use 32 for H_2 and W_2 is computed by $\gamma_1 H_2$.

Note that scaling x and y axis independently has a negative effect of changing the original aspect ratio. This can lead to serious pattern degradation and affect inter-class distances. We will alleviate this problem by incorporating additional information about the original object characteristics into the feature vector as described later. The figure 5 shows a few slices extracted from one sample image and their normalized versions. The objects in slices are shifted to the center and stretched according to the normalization procedure described earlier.

2.4. Feature extraction

We extract feature vectors from separate slices. One slice is transformed into one feature vector. We use simple pixel values (appearance based features) normalized to the range $[0, 1]$ as features. Note that the size normalization procedure affects the original aspect ratio as described in the previous subsection. Therefore we provide the classifier with an additional information about the original characteristics of the object by adding the following complementary features to the feature vector.

$$\left[\frac{\frac{\mu_{10}}{m_{00}}}{W_1}, \frac{\frac{\mu_{01}}{m_{00}}}{H_1}, 2\frac{\sqrt{\frac{\mu_{20}}{m_{00}}}}{W_1}, 2\frac{\sqrt{\frac{\mu_{02}}{m_{00}}}}{H_1}\right] \tag{10}$$



(a)

(b)

Figure 5: a) Sample image slices and b) their size-normalized versions

The final feature vector is subject to PCA transformation and number of components is reduced from $\gamma_1(H_2)^2 + 4$ to 30.

3. Experiments

We applied the moment normalization scheme to the RIMES [1] and IAM [15] corpora and compared it to the results obtained using standard preprocessing steps [18] [11] and with results reported by other groups.

We use the RIMES corpus from the ICDAR 2011 competition. The corpus consist of 59,202 images with French handwriting: 51,738 for training and 7464 for validation. The validation set has been used before as the test set for the ICDAR 2009 competition, therefore we compare our results with official results from this competition [10]. The problem is defined as an isolated word recognition in a closed-vocabulary scenario with the size of vocabulary of 5335 words. We use an unigram language model with perplexity 45.2.

The IAM database consist of handwritten English text sentences, which have been built upon the LOB corpus. There are 6161 images for training, 920 for validation, and 2781 for testing. We apply a trigram language model that has been built upon the LOB, Brown, and Wellington corpora. The sentences appearing in IAM validation and test sets have been excluded for the purpose of language model training. For the lexicon we extract the 50k most frequent words therefore producing an open-vocabulary scenario. The perplexity of the language model is equal to 258.7. The OOV rate is equal to 4.01% for validation set and 3.47% for test set.

The baseline system have been optimized in the work by Pesch [18] and Jonas [11]. For classification we use a HMM model with 12 states for RIMES and 10 states for IAM with every two subsequent states sharing the

Table 1: Comparison with the results reported by other groups on RIMES

Systems	WER [%]	CER [%]
preprocessing baseline	16.7	8.3
moment normalization	13.4	5.5
TUM (RNN) [10]	9.0	-
UPV (MLP, HMM) [10]	16.8	-
ParisTech (HMM) [10]	23.7	-
IRISA (HMM) [10]	25.3	-
SIEMENS (HMM) [10]	26.8	-

Table 2: Results for moment normalization on RIMES

Systems	WER [%]	CER [%]
size norm. w/o comp. features	18.2	8.1
size normalization	15.6	6.8
+ height reestimation	14.4	6.1
+ thickness normalization	13.4	5.5
BIM size normalization	16.4	8.7

same output probabilities. The model is trained with the Viterbi algorithm using maximum likelihood (ML) as training criterion. The output probabilities are trained with Gaussian mixtures with 10 splits for RIMES and 7 splits for IAM. We use the language model scaling of 20 for both corpora. The parameters of the sliding window γ_1, γ_2 have been experimentally optimized. We take 0.03 for γ_2 and for γ_1 $32/16$ for RIMES and $32/14$ for IAM.

The table 1 shows the comparison of the results on the RIMES database. We achieve an excellent word error rate of 13.4%, which is comparable with today's state of the art systems. This result is obtained just using moment normalization scheme and HMM, we do not use neural network or other preprocessing steps, that are commonly applied by other groups. The preprocessing baseline result has been obtained by using the same HMM and language model, but optimized with different parameters. We have used the following preprocessing steps as described by Pesch [18]: median blurring, contrast normalization, deslanting, baseline normalization. The results from other groups are from the ICDAR 2009 competition [10].

The table 2 summarizes the development of the normalization scheme. Simple second-order moments revisited in this paper perform better than the BIM method proposed by Liu [14]. The introduction of the hight reestimation method described in section 2.2 improves the error rate by 1% absolute. The stroke thickness normalization method improves the result by further 1%. We noticed a high influence of the complementary features described in section 2.4 on the recognition performance. The word error rate on the Rimes corpus increased by 2.6% when we excluded those features from the feature vector.

Table 3: Comparison with the results reported by other groups on IAM

Systems	WER [%]		CER [%]	
	Devel	Eval	Devel	Eval
preprocessing baseline	35.0	46.6	16.9	16.6
moment normalization	26.6	37.3	10.6	18.1
Espana. et al. [7] (HMM)	32.8	38.8	-	18.6
Bertol. et al. [2] (HMM)	30.9	35.5	-	-
Dreuw et al. [6] (HMM)	31.9	38.9	8.4	11.7
D. et al. [5] (MLP/HMM)	22.7	32.9	7.7	12.4
Bertol. et al. [2] (HMMs)	26.8	32.8	-	-
Graves et al. [9] (RNN)	-	25.9	-	18.2
E. et al. [7] (MLP/HMM)	19.0	22.4	-	9.8

Table 4: Results for moment normalization on IAM

Systems	WER [%]	CER [%]
size normalization	28.7	11.7
+ height reestimation	27.9	11.1
+ thickness normalization	26.6	10.6

The table 3 shows the comparison of the results on the IAM database. We achieve a word error rate of 37.3% on the test set. The baseline has been obtained using similar preprocessing steps to those applied on RIMES as described by Jonas [11]. The results from other groups in the middle part of the table are reported for HMM models with Gaussian Mixtures and preprocessing. The results in the lower part include system combinations or neural networks and are some of the best results reported so far for IAM. Dreuw [6] applied a discriminative-trained HMM model to features preprocessed with feed-forward neural networks. España-Boquera [7] preprocessed the images with several neural networks, one network for each preprocessing step. Graves [9] used recurrent neural network. Bertolami [2] applied a voting strategy to several HMM models.

The table 4 summarizes the performance of different normalization steps on IAM. The height reestimation method and the thickness normalization method give similar improvements to the ones seen on the RIMES database. In all our experiments using the moment normalization on original images outperformed the preprocessing schemes.

4. Conclusions

We showed that the use of moments improves significantly the recognition performance in handwriting recognition and outperforms other preprocessing approaches. On the RIMES database our moment and HMM based system is the best pure HMM system and achieves a performance of 13.4 word error rate. Additionally, moment-based normalization of slice height and line thickness improves the result over the baseline moment method. For the IAM database we observe similar results as on the RIMES database and a total improvement of 9% over the baseline from 46.6 to 37.3. Finally, the second-order moment normalization technique described in this paper requires no training, is not based on heuristics but on image statistics, is fast to compute and easy to integrate into existing systems.

5. Acknowledgements

This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

[1] E. Augustin, M. Carré, G. E., J. M. Brodin, E. Geoffrois, and F. Preteux. Rimes evaluation campaign for handwritten mail processing. In *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, pages 231–235, 2006.

[2] R. Bertolami and H. Bunke. Hidden markov model-based ensemble methods for offline handwritten text line recognition. 41(11):3452–3460, Nov. 2008.

[3] H. Bunke. Recognition of cursive roman handwriting: past, present and future. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 448 – 459 vol.1, aug. 2003.

[4] R. G. Casey. Moment normalization of handprinted characters. *IBM Journal of Research and Development*, 14(5):548 –557, sep. 1970.

[5] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney. Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: a comparison for offline handwriting recognition. In *IEEE International Conference on Image Processing*, Brussels, Belgium, Sept. 2011.

[6] P. Dreuw, G. Heigold, and H. Ney. Confidence- and margin-based mmi/mpe discriminative training for off-line handwriting recognition. *Int. J. Doc. Anal. Recognit.*, 14(3):273–288, Sept. 2011.

[7] S. España-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):767 –779, april 2011.

[8] A. Giménez, I. Khoury, and A. Juan. Windowed bernoulli mixture hmms for arabic handwritten word recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on*, pages 533 –538, nov. 2010.

[9] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. 31(5):855–868, May 2009.

[10] E. Grosicki and H. El Abed. Icdar 2009 handwriting recognition competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1398 –1402, july 2009.

[11] S. Jonas. Improved modeling in handwriting recognition. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, Jun 2009.

[12] C.-L. Liu, M. Koga, H. Sako, and H. Fujisawa. Aspect ratio adaptive normalization for handwritten character recognition. In T. Tan, Y. Shi, and W. Gao, editors, *Advances in Multimodal Interfaces — ICMI 2000*, volume 1948 of *Lecture Notes in Computer Science*, pages 418–425. Springer Berlin / Heidelberg, 2000. 10.1007/3-540-40063-X_55.

[13] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[14] C.-L. Liu, H. Sako, and H. Fujisawa. Handwritten chinese character recognition: alternatives to nonlinear normalization. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 524 – 528 vol.1, aug. 2003.

[15] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. 5(1):39–46, Nov. 2002.

[16] T. Miyoshi, T. Nagasaki, and H. Shinjo. Character normalization methods using moments of gradient features and normalization cooperated feature extraction. In *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, pages 1 –5, nov. 2009.

[17] M. Pastor, A. Toselli, and E. Vidal. Projection profile based algorithm for slant removal. In A. Campilho and M. Kamel, editors, *Image Analysis and Recognition*, volume 3212 of *Lecture Notes in Computer Science*, pages 183–190. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-30126-4_23.

[18] H. Pesch. Advancements in latin script recognition. Master's thesis, RWTH Aachen University, Aachen, Germany, Nov. 2011.

[19] J.-F. Rivest, P. Soille, and S. Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336, 1993.

[20] A. H. Toselli, A. Juan, J. González, I. Salvador, E. Vidal, F. Casacuberta, D. Keysers, and H. Ney. Integrated handwriting recognition and interpretation using finite-state models. 2004.

[21] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2001.