An Efficient Parametrization of Character Degradation Model for Semi-synthetic Image Generation

van Cuong Kieu, Muriel Visani, Nicholas Journet, Rémy Mullot, Jean-Philippe Domenger

An Efficient Parametrization of Character Degradation Model for Semi-synthetic Image Generation

V.C Kieu
LaBRI, University of Bordeaux
Bordeaux, France
vkieu@labri.fr

Muriel Visani
L3i, University of La Rochelle
La Rochelle, France
muriel.visani@univ-lr.fr

Nicholas Journet
LaBRI, University of Bordeaux
Bordeaux, France
journet@labri.fr

Rémy Mullot
L3i, University of La Rochelle
La Rochelle, France
remy.mullot@univ-lr.fr

Jean Philippe Domenger
LaBRI, University of Bordeaux
Bordeaux, France
domenger@labri.fr

ABSTRACT

This paper presents an efficient parametrization method for generating synthetic noise on document images. By specifying the desired categories and amount of noise, the method is able to generate synthetic document images with most of degradations observed in real document images (ink splotches, white specks or streaks). Thanks to the ability of simulating different amount and kind of noise, it is possible to evaluate the robustness of many document image analysis methods. It also permits to generate data for algorithms that employ a learning process. The degradation model presented in [7] needs eight parameters for generating randomly noise regions. We propose here an extension of this model which aims to set automatically the eight parameters to generate precisely what a user wants (amount and category of noise). Our proposition consists of three steps. First, N_{sp} seed-points (*i.e.* centres of noise regions) are selected by an adaptive procedure. Then, these seed-points are classified into three categories of noise by using a heuristic rule. Finally, each size of noise region is set using a random process in order to generate degradations as realistic as possible.

Keywords

degradation model, synthetic document image, degradation model validation, performance evaluation

1. INTRODUCTION

In recent years, degradation models are widely used to generate a benchmarking database in order to assess the performance of different document analysis and recognition methods (*e.g.* symbol recognition & spotting system evaluation in [2], handwriting recognition algorithm in [11], OCR, text line detection). The other interest of using degradation model for generating synthetic (or semi-synthetic) images is to enrich training databases. For example, the authors of [13] and [14] have improved their system performance by creating synthetic handwritten characters.

HIP '13 August 24 2013, Washington, DC, USA. Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-2115-0/13/08 ...$15.00. http://dx.doi.org/10.1145/2501115.2501127

Synthetic document image generation is a topic that has been little discussed in the state of the art. Baird presented a state of the art of several methods in [1]. He mentioned several degradations that appear in real documents. He also proposed a general methodology for creating degradation models. In [9], the authors proposed a model for reproducing noise that probably appears during the barcode printing process. J. Zhai *et al.* [3] have proposed a model named "hard pencil noise". This model can synthesize white specks to evaluate their line detection method. A modeling of the bleed-through defect has been proposed by R. F. Moghaddam *et al.* [10]. It can reproduce the apparition of *recto* ink on *verso* document side background. In [12], the author proposed a noise model that reproduces noise appearing when scanner sensor fails. At last, Kanungo model presented in [5] (and validated in [4], [8]) can generate the salt and pepper noise that appears frequently near characters. As mentioned by [8] , the difficulty in creating a degradation model is to validate it (*i.e.* to prove that a model generates realistic defects).

This article presents a new contribution started in [7] and [6] where we proposed two new degradation models. These models are able to reproduce specific defects appearing in old document images. In [6], we presented a 3D model for reproducing geometric distortions such as folds, torns, or convexo-concaves of the paper sheet. In [7], we investigated a model which allows to create gray-level defects such as dark specks near characters or ink discontinuities. We are then collaborating with other researchers in order to estimate initially the impact of defects. Recently, 6000 synthetic degraded images for the music score removal staff line competition of ICDAR 2013[1] were generated by using the two models.

As mentioned earlier in the state of the art, the parametrization of a degradation model is a complex task. First, in order to use a model, it is necessary to understand how to set the different parameters resulting in a lot of possible combina-

[1]http://www.cvc.uab.es/cvcmuscima/competition2013/index.htm

tions. Then, for specific purpose, an user might have to generate specific defects. For example testing the robustness of an OCR might require to generate characters with many disconnections, whereas testing a word spotting algorithm requires the generation of ink spots between words or lines. Hence, we propose a parametrization method for estimating parameters of the model presented in [7] according to the quantity and the specificity of character defects given by users. Real ancient document images suffer from defects that mostly appear in the neighborhood of the characters. Usually, these noise regions can be divided into three types: *independent spot*, *overlapping spot*, and *disconnection spot*. The *independent spots* are the noise regions that appear inside or outside of characters. An *independent black spot* is presented in Fig. 1-a while a white one can be seen in Fig. 1-b). The *overlapping spots* are the noise regions that overlap a character and therefore modify the edge of the character. For example, Fig. 1-c-d have respectively *overlapping black spot* and *overlapping white spot*. At last, the *disconnection spots* are only the white noise regions that break the connectivity of characters (*i.e.* the disconnection white spot in Fig. 1-e).

This paper is organized as follows: first, in Section 2, we explain shortly the principle of our degradation model presented in [7]. Then, in Section 3, an overview of the proposed method for parametrizing automatically our model is described. Several algorithms (for selecting noise regions and type of defect) are introduced in section 4. Finally, experimental results and conclusion are given in section 5 and 6, respectively.

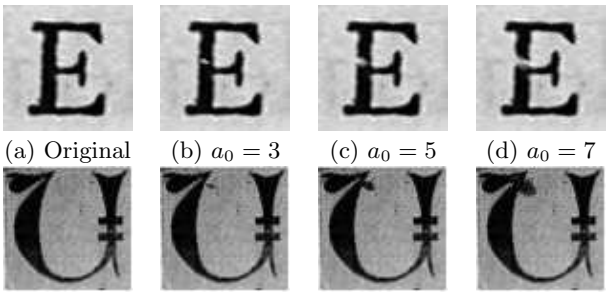
(a) (b) (c) (d) (e)

Figure 1: Examples of three types of noise in real ancient documents: (a), (b) two independent black and white spots; (b), (c) two overlapping black and white spots; (e) disconnection white spot

2. PRESENTATION OF THE CHARACTER DEGRADATION MODEL

As detailed in [7], we proposed a character degradation model based on three steps. Firstly, the centres of noise regions are defined as the flipped pixels resulting from the non-linear local selection process presented by Kanungo *et al.* [5]. This process flips, on a binary image, the value of some foreground and background pixels. These pixels, called "seed-points", are mostly near the edge of characters. They are divided into two sets. The first set (P_{fb}) represents the centre of each future white spot. The second one (P_{bf}) represents the centre of each future black spot. This degradation model has five parameters, $\theta = (\alpha_0, \alpha, \beta_0, \beta, \eta(\eta_0, \eta_1))$ for controlling the amount of generated seed-points.

Secondly, for each seed-point, three properties are computed: shape, direction, and size. In order to create synthetic images as realistic as possible, we propose to generate an elliptic noise region (defined by a minor and a major axis). The major axis depends on the gradient vector at the centre of this noise region and a parameter a_0. The parameter a_0 indicates that the generated noise region becomes large or not. The minor axis is set according to the major axis value and a parameter g (*i.e.* flattening factor of an ellipse). For each noise region, according to the value of the parameters a_0 and g, it will be an *independent spot*, an *overlapping spot* or a *disconnection spot*. For example, Fig. 2 provides six degraded images with the three types of degradations.



(a) Original (b) $a_0 = 3$ (c) $a_0 = 5$ (d) $a_0 = 7$

(e) Original (f) $a_0 = 3$ (g) $a_0 = 4$ (h) $a_0 = 7$

Figure 2: Examples of different noise generation with different a_0 values: (a), (e) original images; (b), (f) two independent white and black spots; (c) overlapping white spot and (g), (h) two overlapping black spots; (d) disconnection spot.

Finally, the gray value of pixels inside each elliptic noise region is changed. To produce a noise as realistic as possible, a new grayscale value is randomly set by using a generation function that satisfies the normal distribution. The mean of this distribution is set according to the distance between each pixel and the centre of the noise region. The variance σ of the function is an input parameter. Thus, our grayscale character degradation model lies on eight parameters: $\Delta(\alpha_0, \alpha, \beta_0, \beta, \eta(\eta_0, \eta_1), a_0, g, \sigma)$ that might lead to difficulties in generating specific synthetic images.

To simplify the noise generation process and to be able to generate specific character defects, we propose a solution for users to parametrize the model with only 4 values: the desired amount of noise regions and the proportion of wished *independent spot*, *overlapping spot*, and *disconnection spot* in the final degraded image.

3. PROPOSED METHOD OVERVIEW

The noise model presented in [7] can generate various kinds of differently visual artifacts (white/dark spots, character disconnections ...). However, with this model, the amount of each type of noise is randomly generated, which makes the visual appearance of the semi-synthetic image result quite unpredictable, and unsuitable to the user's requirement. In this section, we therefore introduce an overview of the proposed method which allows the user to select the amount of each kind of noise injected into the degraded image.

By only using four parameters, the three steps of the degradation model presented in [7] are now simplified as in Fig. 3. First, according to the desired amount of noise regions, we propose an algorithm for setting automatically the Kanungo model parameters [5] and generating the N_{sp} seed-points.

Then, with the given number of *independent spot*, *overlapping spot* and *disconnection spot*, we propose a heuristic rule that assigns each seed-point to one of the three categories. Finally, the size of each noise region is set according to another heuristic rule based on a randomly pseudo process.

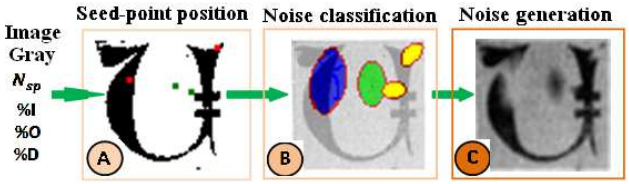
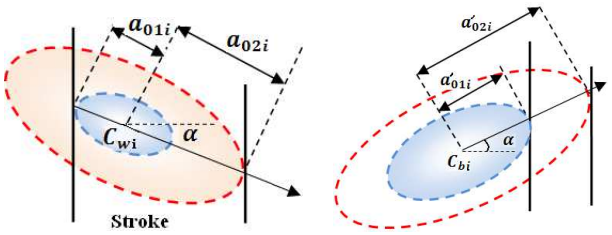


Figure 3: The three main steps of our automatic parameter process for character degradation model

Before describing the three main steps of the model in detail, we would like to focus on what happens to an elliptic noise region when its size increases. Supposed that two seed-points $C_{wi} \in P_{fb}$ and $C_{bi} \in P_{bf}$ are respectively the centres of future white and black spots. Let a_i and a'_i be the sizes of the major axis of these spots. Fig. 4-a shows that the white spot (with its centre inside a character) is connected with the nearest edge of the stroke when its size a_i is equal to a_{01i}. It touches the second one when its size a_i is equal to a_{02i}. Indeed, this white spot is an *independent white spot* until $a_i = a_{01i}$. It becomes an *overlapping white spot* with $a_{01i} \leq a_i \leq a_{02i}$ whereas it will be a *disconnection spot* with $a_i > a_{02i}$. In the same way, the black spot (its centre near a character) C_{bi} will be an *independent black spot* if $a'_i < a'_{01i}$ and an *overlapping black spot* if $a'_{01i} \leq a'_i \leq a'_{02i}$ (see Fig. 4-b) . As mentioned earlier in Section 1, we consider that only white spots can generate *disconnection spots*.



(a) white spot (b) black spot

Figure 4: The evolution of a noise region when its major axis size a_i increases. (a) case of white spot inside a character. (b) case of black spot near a character.

Finally, each seed-point has two thresholds a_{01i} and a_{02i} which will be used to specify the type of each seed-point. Based on this analysis, the noise classification can be generalized by Definition 3.1.

Definition 3.1 *For each seed-point C_i, let a_{01i}, a_{02i} be its thresholds and a_i be its major axis size:*

- *If $a_i < a_{01i}$, C_i is an independent spot.*
- *If $a_{01i} \leq a_i \leq a_{02i}$, C_i is an overlapping spot.*
- *If $a_i > a_{02i}$ and $C_i \in P_{fb}$, C_i is a disconnection spot.*

4. PROPOSED METHOD

4.1 Seed-points position selection

The seed-point generation process detailed in [7] is shortly described in the following procedure (where n is the number of obtained seed-points):

Algorithm 4.1: SEED-POINTSGENERATION$(\alpha_0, \alpha, \beta_0, \beta, \eta_0, \eta_1)$

$n \leftarrow 0$
for each $pixel \in Binarized\_Image$
do $\begin{cases} d \leftarrow distance(pixel \textbf{ to } character\_boundary) \\ \textbf{if } (pixel \textbf{ is } Background) \\ \quad \textbf{then } weight \leftarrow (\beta_0 e^{-\beta d} + \eta_0) \\ \textbf{else } weight \leftarrow (\alpha_0 e^{-\alpha d} + \eta_1) \\ r \leftarrow uniform\_random(0, 1) \\ \textbf{if } (weight \geq r) \textbf{ then } pixel \leftarrow seed\_point; n \leftarrow n + 1; \end{cases}$
return (n)

Instead of using the probability procedure, which takes into account the five parameters of Kanungo noise model $\theta = (\alpha_0, \alpha, \beta_0, \beta, \eta(\eta_0, \eta_1))$, an adaptive procedure is proposed. For this procedure, a user only needs to indicate the desired number of wished seed-points (N_{sp}) and the wished percentage of *disconnection spots* (D) among the N_{sp} seed-points. As mentioned earlier in [7], if we generate too many seed-points very close to each other, the result might look too much synthetic. For that reason, we propose to set the maximum number of seed-points that a user can generate to the number of connected components. This value is usually close to the number of characters appearing in the document image. The proposed procedure is detailed as follows:

Algorithm 4.2: SEED-POINTSIDENTIFICATION(n_{sp}, D)

$\alpha_0 \leftarrow \beta_0 \leftarrow 1, \alpha \leftarrow \beta \leftarrow 0$
$\eta_0 \leftarrow (-D), \eta_1 \leftarrow 0$
$n \leftarrow$ SEED-POINTSGENERATION$(\alpha_0, \alpha, \beta_0, \beta, \eta_0, \eta_1)$
while $(n \neq n_{sp})$
do $\begin{cases} \alpha = random[0, +\infty] \\ \beta = random[0, +\infty] \\ n \leftarrow \text{SEED-POINTSGENERATION}(\alpha_0, \alpha, \beta_0, \beta, \eta_0, \eta_1) \end{cases}$
return $(\alpha, \beta, \alpha_0, \beta_0, \eta_0, \eta_1, n)$

The convergence of Algorithm 4.2 is proved in Appendix A. For the first iteration, α and β are set to 0. It means that the probability for each pixel to be a seed-point is maximum (see (3) and (4) in Appendix A). Then, to optimize the while loop, α and β are selected by using a random number generator whose mean is calculated as in (9) (*i.e.* the expectation E in (9) is equal to the number of input seed-points). The parameter η_0 is equal to (-D) in order to ensure that we always generate D percent of disconnection spots. For example, if D is set at 1 (100%) by the user, according to (3) and (4), the probability for a background pixel to become an ink pixel is equal to ($\beta_0 e^{-\beta d}$ - 1) < 0. It means that only disconnection spots will be generated.

4.2 Noise region classification

The aim of this step is to assign each seed-point to one of the three types. The difficulty is to find an allocation which generates a realistic semi-synthesized document image. Indeed, if too many large noise regions are generated, it might lead to non-realistic results. For example, if a seed-point

far from a character boundary is assigned as an *overlapping spot*, it will produce a non-realistic big black spot. That is the reason why we propose a heuristic rule which assigns, for each seed-point, a type of noise based on its two thresholds.

The output of the previous step is a set N_{sp} of seed-points. Let I, O and D be the number of wished *independent spots*, *overlapping spots*, and *disconnection spots* ($I + O + D = N_{sp}$). Let N_{is}, N_{os}, N_{ds} be the sets of *independent spots*, *overlapping spots*, and *disconnection spots*. Let `min_a01 (setOfSeedPoint S)` and `min_a02 (setOfSeedPoint S)` be the two functions which respectively return the seed-point with the lowest a_{01} and a_{02} value.

Algorithm 4.3: SEED-POINTSAFFECATION(N_{sp}, I, O, D)

$N_{is}, N_{os}, N_{ds} \leftarrow \{\}$
$sp1, sp2 : seedPoints$;
while (N_{sp} *is not empty* **and** $D > 0$)
do $sp1 \leftarrow min\_a02(N_{sp})$;
if ($D > 0$ **and** $sp1 \in P_{fb}$)
then $D--$; $N_{sp} \leftarrow N_{sp} - \{sp1\}$; $N_{ds} \leftarrow N_{ds} + \{sp1\}$;
while (N_{sp} *is not empty* **and** $O > 0$)
do $sp2 \leftarrow min\_a01(N_{sp})$;
if ($O > 0$)
then $O--$; $N_{sp} \leftarrow N_{sp} - \{sp2\}$; $N_{os} \leftarrow N_{os} + \{sp2\}$;
$Move\ all\ the\ rest\ seed-points\ of\ N_{sp}\ to\ N_{is}$;
return (N_{is}, N_{os}, N_{ds})

Let's see an example of five seed-points ($C_1 \rightarrow C_5$) given in Table 1. The percentages of the three types are given by a user as follows: 20% of *independent spots*, 40% of *overlapping spots* and 40% of *disconnection spots*. Indeed, with $I = 1$, $O = 2$ and $D = 2$, N_{is} will have one seed-point, N_{os} and N_{ds} will have two seed-points after the application of Algorithm 4.1. The two values (a_{01} and a_{02}) are computed for each of the five seed-points. First, C_1 and C_4 are set as *disconnection spot*. Then, C_2 and C_3 are set as *Overlaping spot*. At last, C_5 is set as *independent spot*.

Table 1: Example of noise region classification

Seed-point	a_{01}	a_{02}	Final classification
C_1	1.5	2.4	Disconnection spot
C_2	1.3	3.7	Overlapping spot
C_3	2.1	4.6	Overlapping spot
C_4	1.9	2.7	Disconnection spot
C_5	2.8	5.4	Independent spot

4.3 Noise generation process

As a result of the previous step, all the seed-points are classified according to the user's wishes. In this step, the value of every pixel inside each ellipse is degraded to generate noise regions. In order to generate degraded images as realistic as possible, a heuristic rule is used to set randomly the size of the major axis of each ellipse according to the type of degradation while the Definition 3.1 is respected. Let a_i, a_j, and a_k be respectively the sizes of the major axis of an elliptic *independent spot*, *overlapping spot*, and *disconnection spot* regions. Therefore:

$$\begin{cases} a_i = a_{01i} \times \mu_i, 0 < \mu_i < 1 \\ a_j = a_{01j} + \mu_j \times (a_{02j} - a_{01j}), 0 < \mu_j < 1 \\ a_k = a_{02k} + \mu_k \times \delta, 0 < \mu_k < 1 \end{cases} \quad (1)$$

To define these sizes, the values of μ_i, μ_j, μ_k are automatically selected by using a random number generator ranging from 0 to 1. The flattening factor g of the elliptic noise region can be randomly chosen between $[0, 1]$ as well. The parameter δ is set by the average width of all the connected components to avoid generating large disconnection spots. The pixel values within the elliptic noise region are modified by a random number generator following the normal distribution as presented in Section 2.3 of [7]. Last, the Gaussian blur operation is appliqued on the noise region so as to make the values of two adjacent pixels more coherent.

4.4 Degradation level estimation

As a method using a random process for generating degraded document images, a value that indicates the effected proportion of degradation of the image result needs to be computed. In general, this value, also known as the degradation level, is calculated by comparing the image before and after the random degradation process. In [3], the authors calculated the value as the difference of gray value of all pixels before and after the degradation process. Let L be the difference between the image before and after this process. Let I and I' be the gray images before and after the process, respectively. Therefore, L can be calculated as follows:

$$L = I - I' \quad (2)$$

5. EXPERIMENTAL RESULTS

5.1 Generation of images with a fixed number of seed-points

In this test, the number of seed-points is fixed by the number of connected components. Fig. 5 provides three degraded images of the original one in Fig. 5-a, which has 351 connected components. The image in Fig. 5-b is degraded by generating only *independent spots*. It has the effect of generating only small black noise regions near the characters or small white ones inside the characters. In Fig. 5-c, with the same number of seed-points, the degraded image contains only *overlapping spots* in which black overlapping spots connect with the edge of characters whereas white ones modify the ink character regions. In Fig. 5-d, we decided to generate only *disconnection spots*. Indeed, a lot of strokes of characters are completely disconnected. Moreover, the degradation values (L) of the three images are coherent with a visual validation. For the three images, the degradation values are $L = 183$, $L = 1623$ and $L = 4026$, respectively. This shows that the document image in Fig. 5-d is the most degraded image of the three images. Fig. 5.e-h present three zoomed degraded versions of the word "que" at the end of the text line 9 of the three images.

5.2 Generation of images with different amount of seed-points

In this second test, four degraded images are generated from the original image presented in Fig. 6 with the fixed proportion of the three types of degradations ($I = 15\%$, $O = 60\%$, $D = 25\%$) whereas the number of wished seed-points increases progressively.

Fig. 7 provides the degraded letters of the four images which are visually realistic in spite of the high density of seed-points around the characters (*e.g.* in Fig. 7-e, there are nine

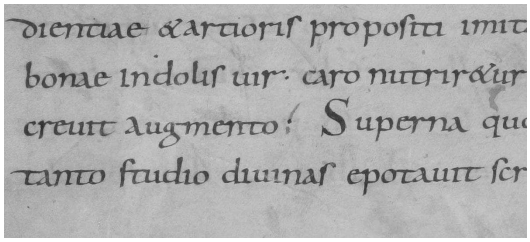


Figure 6: Original document image with 75 connected components

seed-points for seven letters). The calculated degradation values L for the four degraded images are L= 49, L= 91, L= 129, L= 182 corresponding respectively to N_{sp} = 25, N_{sp} = 37, N_{sp} = 56, N_{sp} = 75 (the numbers of seed-points used for degrading the original image in Fig. 6). Once again, the degradation level is coherent with the visual validation results (see more results on [2]).

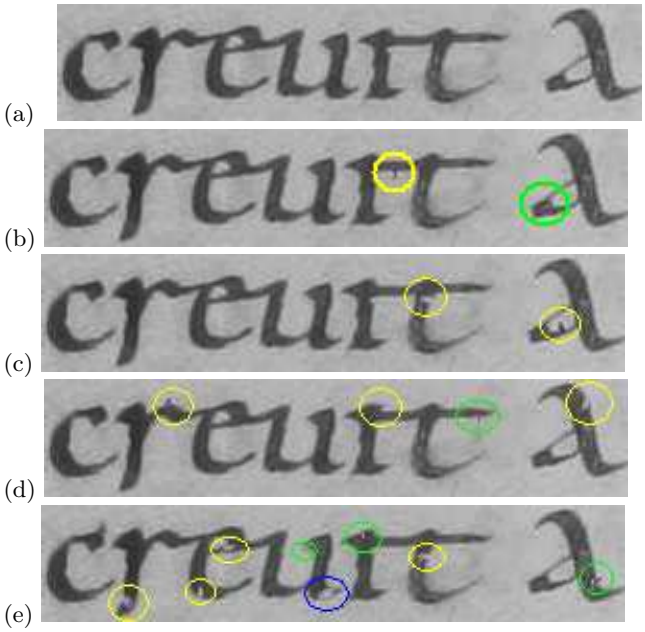


Figure 7: Part of the original image of 7. In yellow an *independent spot*, in blue a *disconnection spot* and in green an *overlapping spot*, (a) the original letters; (b) one independent white spot and one overlapping black spot; (c) two overlapping black and white spots; (d) one independent black spot, one overlapping white spot, and two overlapping black spots; (e) four independent spots, four overlapping spots, and one disconnection spot

6. CONCLUSION

In this paper, we present an efficient parametrization method for a character degradation model. This method aims at simplifying the eight model parameters $\Delta(\alpha_0, \alpha, \beta_0, \beta, \eta, a_0, g, \sigma)$ by using three steps. First, N_{sp} seed-points are selected by adjusting automatically the parameters of Kanungo noise model. The two thresholds a_{01}, a_{02} of each seed-point are also estimated at this step. Then, the seed-points can classified into three types of degradation. Finally, the size of each noise region is randomly chosen in order to generate a specific noise region. Therefore, the model only takes into account four parameters: the number of seed-points and the percentages of the three types of degradation that a user wants to generate. Many images have been generated by using our method. All the images satisfy the user's wishes and look realistic. Since, the calculated degradation levels are coherent with the visual results, that is encouraging us to estimate automatically the four parameters by only the degradation level given by user. The formally validation of the model is also our future work.

[2]http:/www.labri.fr/perso/vkieu/content /DegradationModels/cdm_model.html

Acknowledgement

This research is done within the DIGIDOC project financed by the ANR (Agence Nationale de la Recherche).

7. REFERENCES

[1] H. S. Baird. The State of the Art of Document Image Degradation Modeling. In *In Proc. of 4 th IAPR International Workshop on Document Analysis Systems, Rio de Janeiro*, pages 1–16, Rio de Janeiro, Brazil, 2000.

[2] M. Delalandre, E. Valveny, T. Pridmore, and D. Karatzas. Generation of Synthetic Documents for Performance Evaluation of Symbol Recognition & Spotting Systems. *Int. J. Doc. Anal. Recognit.*, 13(3):187–207, Sept. 2010.

[3] D. D. Jian Zhai, Liu Wenyin and Q. Li. A Line Drawings Degradation Model for Performance Characterization. In *Proc. 7th ICDAR*, pages 1020–1024, Edinburgh, Scotland, August 2003.

[4] T. Kanungo, R. Haralick, H. Baird, W. Stuezle, and D. Madigan. A statistical, Nonparametric Methodology for Document Degradation Model Validation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1209 – 1223, 2000.

[5] T. Kanungo, R. M. Haralick, and I. Phillips. Global and Local Document Degradation Models. In *Proc. of the ICDAR*, pages 730–734, Tsukuba Science City, Japan, Oct. 1993.

[6] V. Kieu, N. Journet, M. Visani, R. Mullot, and J. P. Domenger. Semi-synthetic Document Image Generation Using Texture Mapping on Scanned 3D Document Shapes. In *Accepted for publication in Document Analysis and Recognition (ICDAR), 2013 International Conference on*, 2013.

[7] V. Kieu, M. Visani, N. Journet, J. P. Domenger, and R. Mullot. A Character Degradation Model for Grayscale Ancient Document Images. In *Proc. of the ICPR*, pages 685–688, Tsukuba Science City, Japan, Nov. 2012.

[8] Y. Li, D. Lopresti, G. Nagy, and A. Tomkins. Validation of Image Defect Models for Optical Character Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):99–108, Feb. 1996.

[9] R. Loce and W. Lama. Halftone Banding due to Vibrations in A Xerographic Image Bar Printer. *Journal of Imaging Technology*, 16(1):6–11, 1990.

[10] R. F. Moghaddam and M. Cheriet. Low Quality Document Image Modeling and Enhancement. In *Int. J.Doc. Anal. Recognit*, volume 11, pages 183–201, Berlin, Heidelberg, March 2009. Springer.

[11] M. Mori, A. Suzuki, A. Shio, and S. Ohtsuka. Generating New Samples from Handwritten Numerals Based on Point Correspondence. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 281–290, Amsterdam, Netherlands, 2000.

[12] E. B. Smith. Modeling Image Degradations for Improving OCR. In *16th European Signal Processing Conference (EUSIPCO)*, pages 1–5, Lausanne, Switzerland, August 2008.

[13] T. Varga and H. Bunke. Effects of Training Set Expansion in Handwriting Recognition Using Synthetic Data. In *Proc. 11th Conf. of the Int. Graphonomics Society*, pages 200–203, Scottsdale, AZ, USA, Nov. 2003. Citeseer.

[14] T. Varga and H. Bunke. Generation of Synthetic Training Data for an HMM-based Handwriting Recognition System. In *Proc. 7th ICDAR*, pages 618–622, Edinburgh, Scotland, August 2003.

APPENDIX

A. PROOF OF THE CONVERGE OF ALGORITHM 4.2

Let I_{gray} be the input image. Let n be the number of all pixels in I_{gray} and Z be the maximum distance of a pixel to the nearest edge of characters (the distance transform). The weight w of each pixel $P(x,y)$ is calculated as follows:

$$w = \begin{cases} \alpha_0 \times e^{-\alpha d} + \eta_1, & P : Foreground \\ \beta_0 \times e^{-\beta d} + \eta_0, & P : Background \\ (d \in [1 \rightarrow Z]) \end{cases} \quad (3)$$

where d is the distance transform of the pixel P. As mentioned earlier in Algorithm 4.1, a pixel will be a seed-point when its weight is greater than or equals to the random number r. The random number r is generated by an **uniform distribution** number generator. Let's see a pixel $P(x,y)$ which has the weight w_i. The probability of $P(x,y)$ is calculated hereafter (see Fig. 8) so that this pixel will be a seed-point:

$$p_i = w_i \quad (4)$$

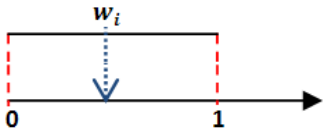


Figure 8: Uniform Distribution

According to (4), the probability with which k pixels will be seed-points is calculated as follows:

$$P_{(n_{sp}=k)} = \sum_{I_m \subset [1 \ldots n]; |I_m| = k; l = 1}^{m=M} \left(\prod_{i \notin I_m} (1 - w_i) \times \prod_{j \in I_m} w_j \right) \quad (5)$$

where M is the number of subsets $I_m \subset [1 \ldots n]$ in which each subset has k elements. Thus, M is calculated as follows:

$$M = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (6)$$

According to (3), $\forall$ pixel P: $0 < w_i < 1$; therefore, $0 < P_{(n_{sp}=k)} < 1$. This means that we can always generate k seed-points with the probability calculated as in (5). As a result, the Algorithm 4.2 always converges. In addition, the expected value is calculated as follows:

$$E = \sum_{l=0}^{n} l \times p_{(k=l)} = \sum_{i=1}^{n} w_i \quad (7)$$

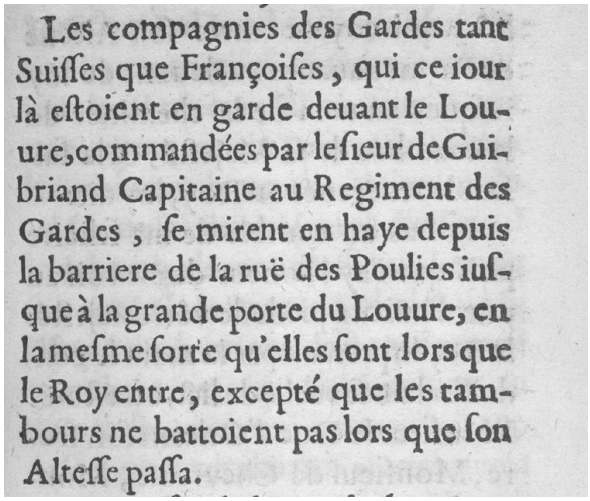
Equation (7) shows that if α and β increase, the weight of each pixel will decrease exponentially according to (3); therefore, the expected number of seed-points will also decrease exponentially.

Let n_{i1} be the number of Foreground pixels which have the same distance d_i. Let n_{i0} be the number of Background pixels which have the same distance d_i. According to (3) and (7), the expectation will be:

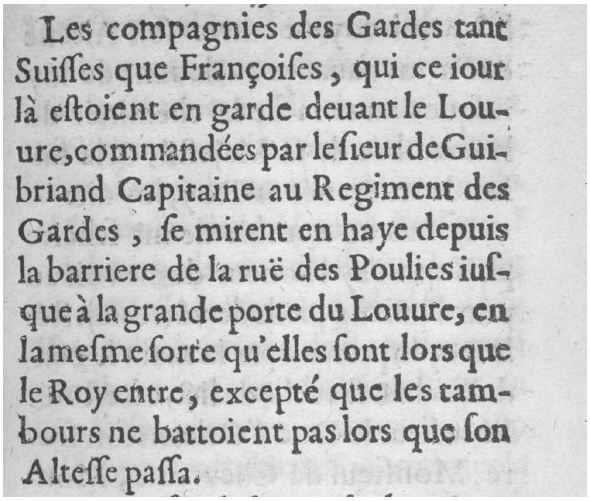
$$\begin{aligned} E &= \textstyle\sum_{i=1}^{Z} (n_{i1} e^{-\alpha d_i} + n_{i0} e^{-\beta d_i} + n_{i1}\eta_1 + n_{i0}\eta_0) \\ &= e^{\alpha} \textstyle\sum_{i=1}^{Z} n_{i1} e^{-d_i} + e^{\beta} \sum_{i=1}^{Z} n_{i0} e^{-d_i} \\ &\qquad + \textstyle\sum_{i=1}^{Z} (n_{i1}\eta_1 + n_{i0}\eta_0) \\ &= e^{\alpha} A + e^{\beta} B + C \end{aligned} \quad (8)$$

where $A = \sum_{i=1}^{Z} n_{i1} e^{-d_i}$, $B = \sum_{i=1}^{Z} n_{i0} e^{-d_i}$, $C = \sum_{i=1}^{Z} (n_{i1}\eta_1 + n_{i0}\eta_0)$. Note that A, B, C are the constants for each input image. We suppose that α and β are generated by using a random number generator. The mean $\mu_{\alpha\beta}$ of this generator can be calculated by using (8):

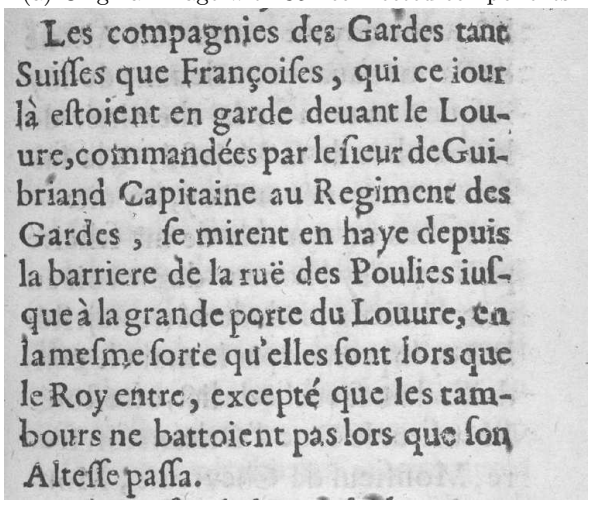
$$\begin{aligned} & E = A e^{\mu_{\alpha\beta}} + B e^{\mu_{\alpha\beta}} + C \\ & \Leftrightarrow \mu_{\alpha\beta} = ln(\tfrac{E-C}{A+B}) \end{aligned} \quad (9)$$



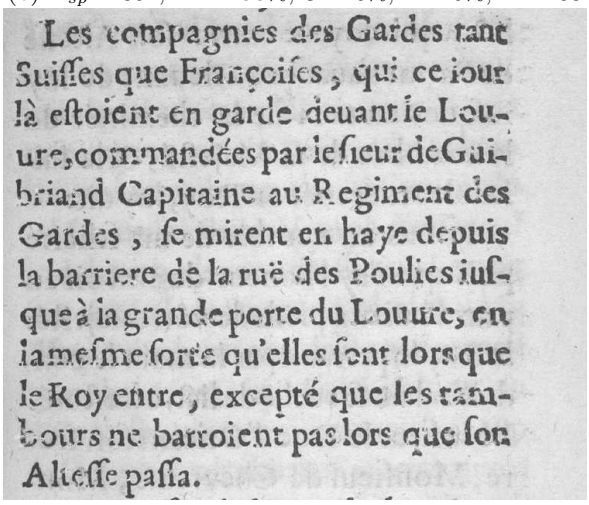
(a) Original image with 351 connected components



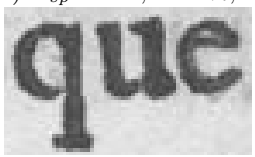
(b) $N_{sp} = 351$, $\mathbf{I = 100}\%$, $O = 0\%$, $D = 0\%$, $L = 183$



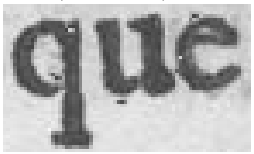
(c) $N_{sp} = 351, I = 0\%$, $\mathbf{O = 100}\%$, $D = 0\%, L = 1623$



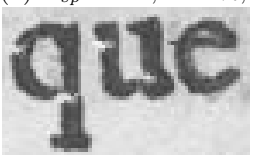
(d) $N_{sp} = 351, I = 0\%, O = 0\%$, $\mathbf{D = 100}\%$, $L = 4026$



(e) Zoomed word original



(f) Zoomed word of (b)



(g) Zoomed word of (c)

(h) Zoomed word of (d)

Figure 5: Degraded images with the maximum number of seed-points $N_{sp} = N_{ccs} = 351$: (a) original image; (b) degraded image with only the independent spot type and $L = 183$; (c) degraded image with only the overlapping spot type and $L = 1623$; (d) degraded image with only the disconnection spot type and $L = 4026$; (e), (f), (g), and (h) are four zoomed words of the images (a), (b), (c), (d), respectively.