An Algorithm for Word Counting

Roger Silvestre Anglada

Abstract—An algorithm is presented, designed to count word from a user submited photo from a book page. In this paper, the diferent modules forming this algorithm are discused and explained along with the order in which they are applied. A final conclusion on how effective it is is presented, along with some examples.

Index Terms—Introduction, Binarization, Image Blurring, Search Change, Checking Connections, Contourn Counting, Algorithm, Conclusion

1 INTRODUCTION

COUNTING words from a physical source can be an arduous task, specially having in mind that humans are prone to committing errors, like skipping a line or losing track. Therefor a computer operated algorithm can be designed to try and solve this problem.

Normally these methods consist of scanning said text, and then applying a character recognition algorithm to make a digitalized copy of it. Having this copy, counting the words becomes a trivial problem.

In this project, the main objective is to solve this counting problem from a different approach. By directly transforming the original image the aim is to provide an alternative method where a character recognition algorithm isn't needed.

The final algorithm is a composition of three different types of modules, each type having different variations. Those three main types are binarization, merging and counting. Due to the binarization and merging, an image full of spots, each one corresponding to a word, is obtained. Then a contour detection algorithm is used to count the number of isolated spots, obtaining this way the total number of words in the text.

Throughout this article the mentioned modules, along with some minor ones, are explained in detail.

2 BINARIZATION

As its implied in its name, binarization is the process of converting a pixel image to a binary image. In this particular case, the objective is always to transform a pixel filled array to a binary array, separating this way the information that we want (with value 0), from the background (value 1).

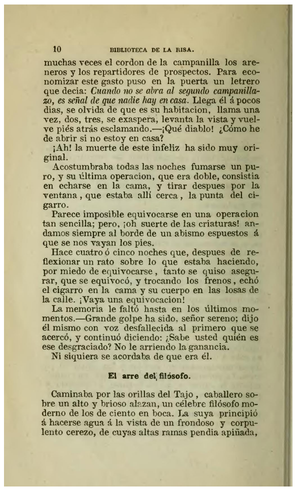
Binarization is used in two different yet similar ways. First to extract information of the original photography (obtaining the text and deleting as much background sound as possible), and second as a mean to get back a binary image after any blurring procedure.

This procedure is implemented in two form: by using a simple cut-off method, or using an adaptative one.

Using the simple method, a global threshold (T_G) is calculated and simply applied (1) to the image (P_{val} is the value of the pixel in said image), calculating this way the final value of each pixel ($P(x,y)$). The value of this threshold is the mean value of the entire image plus a constant (manually determined).

$$P(x,y) = \begin{cases} 1 & \text{if} P_{\text{val}}(x,y) > T_G \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Using the adaptative one, a particular threshold (T_P) is calculated pixel by pixel. In this project, the threshold value is a mean of the neighbourhood of each pixel plus a constant (also manually determined). Although the shape is that of an X by X box, the size of said box is also a variable (being always odd number due to symmetry).



10 BIBLIOTECA DE LA RISA.

muchas veces el cordon de la campanilla los areneros y los repartidores de prospectos. Para economizar este gasto puso en la puerta un letrero que decia: *Cuando no se abra al segundo campanillazo, es señal de que nadie hay en casa.* Llega él á pocos dias, se olvida de que es su habitacion, llama una vez, dos, tres, se exaspera, levanta la vista y vuelve piés atrás esclamando.—¡Qué diablo! ¿Cómo he de abrir si no estoy en casa?

¡Ah! la muerte de este infeliz ha sido muy original.

Acostumbraba todas las noches fumarse un puro, y su última operacion, que era doble, consistia en echarse en la cama, y tirar despues por la ventana, que estaba allí cerca, la punta del cigarro.

Parece imposible equivocarse en una operacion tan sencilla; pero, ¡oh suerte de las criaturas! andamos siempre al borde de un abismo espuestos á que se nos vayan los pies.

Hace cuatro ó cinco noches que, despues de reflexionar un rato sobre lo que estaba haciendo, por miedo de equivocarse, tanto se quiso asegurar, que se equivocó, y trocando los frenos, echó el cigarro en la cama y su cuerpo en las losas de la calle. ¡Vaya una equivocacion!

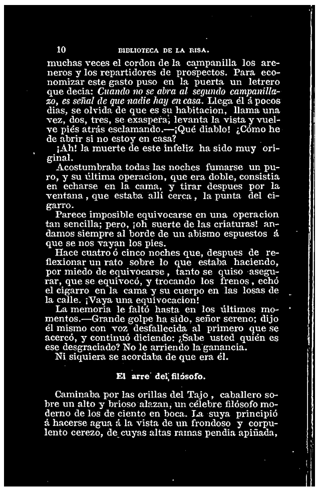
La memoria le faltó hasta en los últimos momentos.—Grande golpe ha sido, señor sereno; dijo él mismo con voz desfallecida al primero que se acercó, y continuó diciendo: ¿Sabe usted quién es ese desgraciado? No le arriendo la ganancia.

Ni siquiera se acordaba de que era él.

El arre del filósofo.

Caminaba por las orillas del Tajo, caballero sobre un alto y brioso alazan, un célebre filósofo moderno de los de ciento en boca. La suya principió á hacerse agua á la vista de un frondoso y corpulento cerezo, de cuyas altas ramas pendia apiñada,

(a)



(b)

Figure 1: *Adaptative Threshold* Binarization display, where the original text 1a and its binarization 1b can be seen.

3 IMAGE BLURRING

As explained, our method of word counting is based on a contour detection algorithm. This algorithm requires each word to be isolated from the rest but at the same time, all of its letters must be connected.

By using this module, the objective is to take a binarized image and blur it in a way that the letters of each word are always in contact with each other (merged together) but at the same time, individual words must not be in contact with others. If successful, a new image composed only of dots, each dot corresponding to a word, is obtained.

It must be noted that for the counting algorithm to give accurate results, there shouldn't be any noise to the final image, as those noise dots would be interpreted as word dots. Therefor a side objective of the blurring process is to get rid of said noise.

Those objectives are accomplished by the application of two different blur methods.

3.1 Gaussian Blur

This filter is applied by convolving the binary image (I) with a gaussian function (3). Said function (2) depends of the variable σ, which in this case is introduced manually. The result is inversely proportional to sigma, meaning that the bigger σ gets, the less noticeable is the blur.

$$G(\rho, \sigma) = e^{\frac{-\rho^2}{2\sigma^2}} \tag{2}$$

$$I * G = FT^{-1}[FT(I)G(\rho, \sigma)] \tag{3}$$

FT and FT^{-1} stands for Fourier Transform and Inverse Fourier Transform, and ρ is the polar coordinate.

This particular filter is especially efficient eliminating little noise-like gaps or dots.

3.2 Point Spread Function OR "PSF Blur"

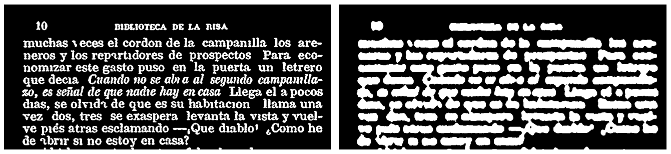
As its name implies this uses a Point Spread Function (PSF) to blur the original image. By again convolving the image with the specific function (4) we can obtain a defocused version of said image (5). In (4) $circ(Z)$ is a function with a value of 0 if $Z < 1$ and 1 if $Z > 1$.

$$PSF(\rho) = TF(circ(\frac{r_0}{\rho})) \tag{4}$$

$$I_{Defoc}(x, y) = I(x, y) * |PSF(x, y)|^2 \tag{5}$$

Even though a PSF allows to model not just a basic defocusing but also an out-of-focus system with spherical aberration, in this particular case by modelling a focus-set system with zero spherical aberration, we obtain the desired results. Therefor, 4 just depends of one manually set variable r_0 (radius of the exit pupil).

Note that this method allows for a better merging of the letters and, at the same time, allows different words to remain unconnected (part of the reason of why an out-of-focus with spherical aberration PSF isn't used).



10 BIBLIOTECA DE LA RISA

muchas veces el cordon de la campanılla los areneros y los repartıdores de prospectos Para economızar este gasto puso en la puerta un letrero que decıa *Cuando no se abra al segundo campanillazo, es señal de que nadie hay en casa* Llega el a pocos dıas, se olvıda de que es su habıtacıon llama una vez dos, tres se exaspera levanta la vısta y vuelve pıés atras esclamando —¡Que dıablo! ¿Como he de abrır sı no estoy en casa?

(a) (b)

Figure 2: Merging display, where the already binarized text can be seen before 2a and after being convolved with the *PSF* 2b

4 SEARCH CHANGE

Even though words are horizontally separated in an orderly manner (leaving approximately the same space between them), vertically this situation is much more chaotic. Due to abnormally large letters like "*q*", "*l*", or capital letters, the distance between words in different rows is irregular and pretty much aleatory (a pattern can't be found). This means that is way easier for words to wrongly merge vertically than otherwise.

Though with carefully chosen blur constants (σ and r_0) this problem can be "ignored", an algorithm can be developed to find the space between rows, record it (forming erase lines), and once the image is fully blurred with some words vertically fused, we can use those lines to separate them.

This algorithm consists on taking an already blurred image (still having empty space between rows) and finding its top left extreme. Once found, a vertical sweep starts, recording the points where the value of a pixel changes from 0 to 1 or 1 to 0 (from word to space and vice versa). At the same time the algorithm keeps track of the mean length of both empty space and words. When it detects that the length of the current word or space is greater than 1.12 times its mean value, it starts sweeping to the right (maintaining its height), looking for said 1 or 0. In case of not finding it, it jumps to the next row, and starts the search again.

Due to the nature of this module, it is only possible to be applied when the text in the image is ideal enough (not curved nor with too much perspective)

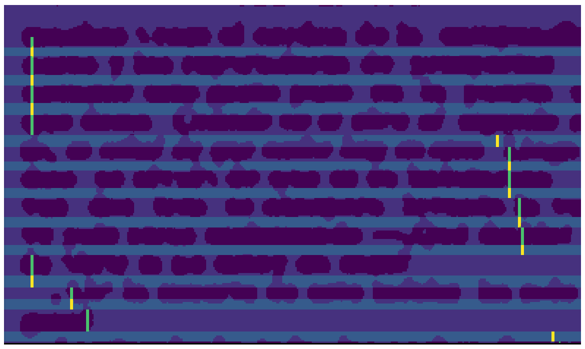
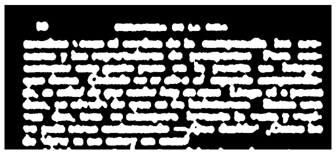


Figure 3: Example of the algorithm at work. After detecting that an excessive distance has been travelled, it sweeps right until it finds a valid point. The green lines indicates at which width the vertical sweeping is taking place.


(a)

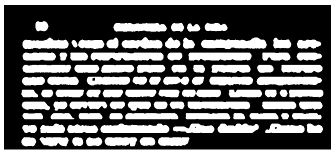
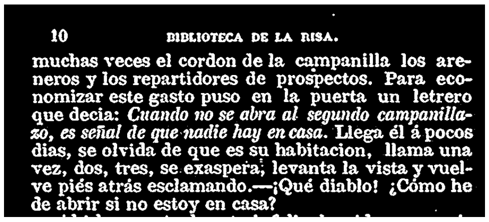

(b)

Figure 4: After applying the *Search Change* algorithm to the Blurred Image 4a, a second image is obtained 4b

5 CHECKING CONNECTIONS

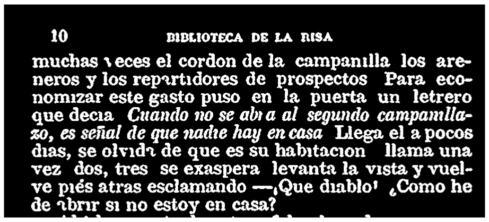
One major flaw in the previous blurring algorithms is that if there is some dots not big enough to be words nor small enough for neither *PSF* nor *Gaussian Blur* to delete them, they will remain untouched and therefor will be counted by the *Contour Counting*.

This module checks for the connections of each pixel with its neighbourhood and registers it, obtaining this way a list with the groups of connected pixels, its correspondent sizes, and its coordinates. With this, it calculates the mean size of a group, divides this value by a constant manually set (*DBCConst*, bigger than 1), and proceeds to delete any group smaller than this threshold.


10 BIBLIOTECA DE LA RISA.

muchas veces el cordon de la campanilla los areneros y los repartidores de prospectos. Para economizar este gasto puso en la puerta un letrero que decia: *Cuando no se abra al segundo campanillazo, es señal de que nadie hay en casa.* Llega él á pocos dias, se olvida de que es su habitacion, llama una vez, dos, tres, se exaspera; levanta la vista y vuelve piés atrás esclamando.—¡Qué diablo! ¿Cómo he de abrir si no estoy en casa?

(a)


10 BIBLIOTECA DE LA RISA

muchas veces el cordon de la campanılla los areneros y los repartıdores de prospectos Para economızar este gasto puso en la puerta un letrero que decıa *Cuando no se abra al segundo campanılla-zo, es señal de que nadıe hay en casa* Llega el a pocos dıas, se olvıda de que es su habıtacıon llama una vez dos, tres se exaspera levanta la vısta y vuelve pıés atras esclamando —Que dıablo ¿Como he de abrır sı no estoy en casa?

(b)

Figure 5: Taking the already blurred image 5a, the algorithm is applyied, obtaining a new image without noise or isolated dots/commas 5b

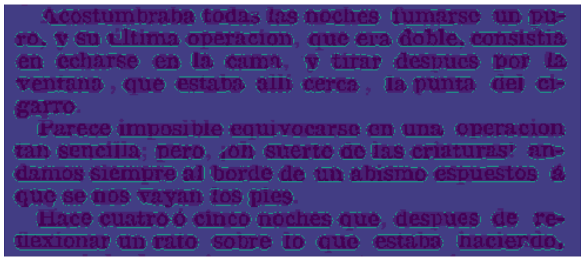
By using a binarized image (with 1-empty,0-word) or its inverse, it can eliminate big dots corresponding to noise and isolated characters (like commas) or empty spaces inside a word, which are originated from bad merging or capital letters like "*O*" or "*P*".

6 CONTOUR COUNTING

With this module, the contours of a binarized image are recorded and posteriorly drawn in said image. With the transformations applied, the image contains multiple dots, each corresponding to a word and isolated from the rest. Therefor, when calculating the number of contours presents in said image, the number of words is also obtained.

The algorithm Suzuki85[1] is used to retrieve the contours from a binary image and therefor, obtain the total number of words.

After being recorded, the contours are also drawn in a parallel image containing the sum of *Text+Blurred Dots+Contours*. This image is displayed to check on any major flaw in the algorithm (as it depends on some manually entered constants).


Acostumbraba todas las noches fumarse un puro, y su última operacion, que era doble, consistia en echarse en la cama, y tirar despues por la ventana, que estaba allí cerca, la punta del cigarro.

Parece imposible equivocarse en una operacion tan sencilla; pero, ¡oh suerte de las criaturas! andamos siempre al borde de un abismo espuestos á que se nos vayan los pies.

Hace cuatro ó cinco noches que, despues de reflexionar un rato sobre lo que estaba haciendo,

Figure 6: After finding and recording the contours, the number is displayed, and the contours are drawn in a new image. This new image 6 displays the original text, the blurred words and the contours found.

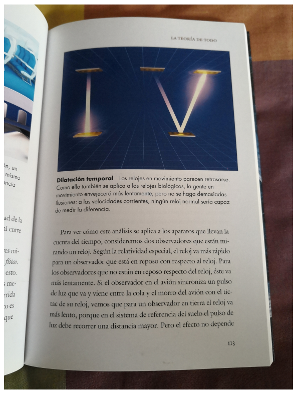
7 ALGORITHM

0) The user is asked to submit the photo of a text, by placing said photo inside the correspondent folder (/Fotos) and tipping its name and extension.
1) The user is asked to give a value to different constants. In case of using one of the 3 example texts, it hints its optimal value.
2) The text is binarized with an *Adaptative Threshold*, and a copy of the result is made for later use.
3) In case of being needed, a horizontal *Search Change* is applied (to delete possible page borders). The decision of making or doesn't making it is left to the user.
4) A weak *Gaussian Blur* is applied to eliminate little noise dots, merge some letters together and make them bigger, followed by a *Check Connections* to delete the bigger noise dots. The image is binarized (*Global Threshold*)
5) A *PSF* is applied to fuse the letters inside words together, enough to merge most characters but leaving some words with empty space inside. The image is binarized (*Global Threshold*)
6) A vertical *Search Change* is applied if possible, obtaining the horizontal separation lines. The application of this step is also left to the user. It should only be applied if the text is ideal enough.
7) A new *Gaussian Blur* is applied. Stronger than 4) (meaning than σ will be normally smaller). This is specially true if a vertical Search Change has been successfully applied. The image is binarized (*Global Threshold*)
8) *Check Connections* is applied once again, to eliminate any separated tops of tall letters (like "l", "t",etc.).
9) A final *Check Connections* is applied, but this time inversing the image before applying it, deleting this way any empty space inside words.
10) Finally, *Count Contours* is applied, obtaining the number of contours (and therefor words) and its coordinates.
11) A Final image is created combining the blurred dots, the original text (previously saved) and the contours.

This is displayed, helping the user to check for possible errors.

As seen, to obtain the best possible results, this algorithm realies on the user to introduce the ideal constant values. Those values can be left in its default state, which will mean that the result obtained probably won't be as acurate as i could be.

One important thing to note is that the photo submited should have just text in it. Meaning the picture should be taken in a way that only text appears in it, as shown in 7.

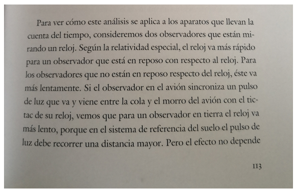

LA TEORÍA DE TODO

Dilatación temporal Los relojes en movimiento parecen retrasarse. Como ello también se aplica a los relojes biológicos, la gente en movimiento envejecerá más lentamente, pero no se haga demasiadas ilusiones: a las velocidades corrientes, ningún reloj normal sería capaz de medir la diferencia.

Para ver cómo este análisis se aplica a los aparatos que llevan la cuenta del tiempo, consideremos dos observadores que están mirando un reloj. Según la relatividad especial, el reloj va más rápido para un observador que está en reposo con respecto al reloj. Para los observadores que no están en reposo respecto del reloj, éste va más lentamente. Si el observador en el avión sincroniza un pulso de luz que va y viene entre la cola y el morro del avión con el tictac de su reloj, vemos que para un observador en tierra el reloj va más lento, porque en el sistema de referencia del suelo el pulso de luz debe recorrer una distancia mayor. Pero el efecto no depende

113

(a)


Para ver cómo este análisis se aplica a los aparatos que llevan la cuenta del tiempo, consideremos dos observadores que están mirando un reloj. Según la relatividad especial, el reloj va más rápido para un observador que está en reposo con respecto al reloj. Para los observadores que no están en reposo respecto del reloj, éste va más lentamente. Si el observador en el avión sincroniza un pulso de luz que va y viene entre la cola y el morro del avión con el tictac de su reloj, vemos que para un observador en tierra el reloj va más lento, porque en el sistema de referencia del suelo el pulso de luz debe recorrer una distancia mayor. Pero el efecto no depende

113

(b)

Figure 7: Example of a picture which will cause problems 7a, and how it should be submited 7b (avoiding pictures if possible, and framing just the text).

8 CONCLUSION

Even though each photo is different (the angle of the camera, torsion of the page, illumination, type of camera, etc.), the developed algorithm provides an accurate approximation on the number of words in the text when said photo is correctly submited.

It works by just having a minimal resolution, therefore it doesn't matter the size of the text to analyse.

Its only requirement is to set the constants previously mentioned to a correct value (specific to each text). As seen in 1, in the end, even though multiple constants exist, we just need to change one or two to truly adapt the algorithm to an specific text.

As a possible improvement, it should be possible to reduce the computing time by optimizing the code, and at the same time, an automatic way to set the constants to its correct value could be ideated. A way to automatically clip wrongly sumbited photos (as in **??**) should also be possible to made.

REFERENCES

[1] Suzuki, S. and Abe, K., *Topological Structural Analysis of Digitized Binary Imagesby Border Following.*, CVGIP 30 1, pp 32-46 (1985)

[2] Y. Linde, A. Buzo and R. Gray, *An Algorithm for Vector Quantizer Design*, IEEE Transactions on Communications, vol. 28, no. 1, pp. 84-95, January 1980.
(Document used as a journal example)

[3] Steven Hawking, "El Universo en una cascara de nuez (Spanish)", Editorial Planeta S. A., 2016
(Text2)

[4] Rafael Boira, "EL LIBRO DE LOS CUENTOS (Spanish)", Imprenta de D. Miguel Arcas y Sanchez, 1862
(Text1)

APPENDIX A

	Threshold		**PSF**	**Gauss Blur**		**Check Connect.**
	NeighBoSize1	CttBin1	r_0	σ_0	σ_1	DBCConst3
Text 1	11	10	7.5	0.73	0.4	10
Text 2	23	10	7.5	0.8	0.8	6
Text 3	23	10	7.5	1.2	0.8	10

Table 1: Ideal constant values set for the three diferent example texts. As seen, the main values which require a change are the neighbourhood size, the sigmas, and the constant corresponding to the final *Check Connections*.

	Words	**Words Counted**
Text 1	315	315.5
Text 2	331	331
Text 3	242	241

Table 2: Final results for three different texts. As seen, the results are nearly perfect when establishing the constant values correctly. In *Text 1* the *Search Change* module is used, meaning that the results can slightly deviate as a result of the method used to find the top left corner (which includes randomly generated numbers).