Optimization of the image congealing process for handwritten Chinese character recognition

Vincent Giraud, *Student, RIT,*

***Abstract*—One of the various existing ways for an artificial intelligence to recognize a handwritten Chinese character is to compare it with a training set composed of pre-aligned images of each symbol. In order to better take into account all the different writing styles, fuzzy mathematics, which contains notably a generalization of the classic set theory, is exploited here. The goal of this paper is to introduce techniques that can accelerate the algorithm, which is, in its initial state, taking too much time for a use by the general public.**

***Index Terms*—artificial intelligence, Chinese character recognition, fuzzy mathematics, set theory, image alignment, image congealing.**

I. Introduction

Offline handwritten characters recognition is a complex task for a program because of the variety of the symbols, and because of the different writing styles, that are specific to each person. Applying this problem to the simplified Chinese characters set takes the difficulty to a higher level. Indeed, the number of ideograms is significantly higher than the number of letters in the latin alphabet. Handwritten Chinese character recognition (HCCR) is also more complex because the writing style of a person can make the differentiation of two symbols hard even for humans.

As Chinese is used by 25% of the world population, it is one of the most used languages. Considering today's digital needs, there is thus a strong demand for the exploitation of the Chinese sets of characters on electronic devices. There are two different sets of ideograms: the simplified Chinese characters, and the traditional one. While the first contains 3755 symbols, the second contains 6763 of them. The knowledge of thousands of characters is required for the reading or writing of common texts.

The use of embedded devices and processors is extremely high today, as carrying a mobile phone capable of running complex, various programs has become routine for a major part of the population, not only in Asia but in all developed countries. It is expected that any embedded software aimed at the general public should highly take into consideration the constraints that are the power consumption and the limited computational resources. A Chinese character recognition system should thus be suitable for such a context in order to become widely accepted.

Most Chinese character recognition algorithms are based on feature extraction. When considering a symbol, they will try to extract measurable information like the number of lines, or the curvature for example. While this type of method can provide accurate results, one drawback that is worth mentioning is that feature extraction on thousands of characters can end up with a complex and large database. This paper deals with the optimization of a technique that does not require feature extraction. Instead, mean images are generated for each symbol using picture alignment with the help of fuzzy logic, a process which we will call image congealing in this paper. The recognition process also uses fuzzy logic, to align the test image with each stack. As the core of such an algorithm is executed many times, small modifications to it can bring considerable speed-ups.

This paper is organised as follows. Section 2 will present related works and the state of the art in this particular domain. Section 3 deals with the optimization of the image congealing process. In section 4, the results obtained after modifying the core algorithm are given. Section 5 will conclude this paper.

II. Related works

A. The simplified Chinese character set

The simplified Chinese character set can be subdivided in multiple subsets. Five categories can be devised : symbols with a single structure, symbols with a left and right structure, symbols with an upper and lower structure, symbols with a "surrounded by" structure, and symbols with a framework structure [1]. Distinguishing two ideograms from the same category can be tedious, as they rely on the same architecture.

All Chinese symbols are composed of simple lines and, optionally, multiple-lines strokes. Two different characters can look extremely similar. For example, the 备 (*bei*) character's look is close to 奋 (*fen*), bus is not the same. 王 (*wang*) and 玉 (*yu*) are also only different thanks to a small stroke. In the same vein, 辨, 辩, and 辫 all refer to *bian*, but are not exactly the same. Any artificial intelligence trying to identify ideograms should thus be able to precisely acknowledge the presence of small, discreet differences that can even vary depending on the writer's region.

B. Entropy-reducing image alignment

Let a stack be composed of n images of the same ideogram, each made by a different writer. Each picture has a resolution of $m \times m$ pixels. The image stack is thus composed of $m \times m$ pixel stacks. In order to make such a stack useful in recognition of Chinese characters, one

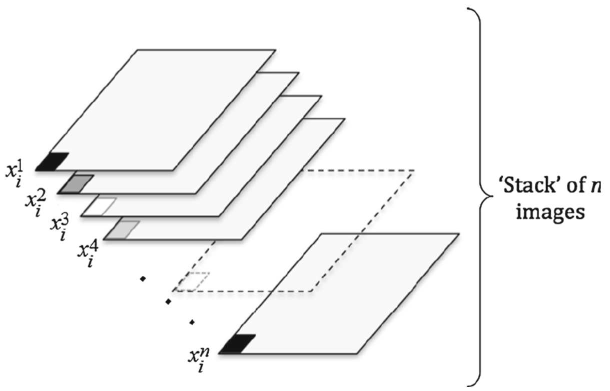

Fig. 1. A stack of images, with a highlighted pixel stack.

should arrange each symbol so that they superpose in the most accurate way possible. Processing the entropy across all pixel stacks then gives a measurement of how aligned are the pictures. The smaller is the entropy, the better is the image congealing. To reduce the entropy, one can apply many transformations the each symbol : translation along *x* (t_x), translation along *y* (t_y), rotation (θ), scaling along *x* (s_x), scaling along *y* (s_y), shearing along *x*(h_x), and shearing along *y* (h_y). These modifications can be applied with multiple transformation matrices, however they can be combined into only one matrix, featured in figure 2.

In order to align one picture with the stack, the following method can be used : create randomly chosen reposition parameters, compute their transformation matrix, and apply it to the image. Then the entropy should be computed. If it is equal or larger than before, the picture as it was before the transformation should be restored. Otherwise, the entropy is better, and the new version of the image should be kept. Exploiting this algorithm iteratively on each symbol helps reducing the differences across the whole stack. Previous experiments showed that applying this routine 3 times to each picture gives satisfying results in artificial intelligence.

When a test character is submitted, it will be aligned with many stacks : one for each potential ideogram. Once all the instances have been congealed with all stacks, a simple k-nearest-neighbors is applied : the result symbol will be the one whose stack's mean picture has the smallest Cartesian difference with the submitted ideogram.

When transforming the images iteratively, one can end up in a situation where the algorithm lowers the entropy by making changes unsuitable to the recognition by an artificial intelligence. For example, the method described above could greatly reduce the scale of all images : the entropy would be reduced, but it would become harder to distinguish different Chinese characters. In order to avoid this, all the transformations have to be counter-balanced by their inverse on all other pictures. When reducing the size of a particular image lowers the entropy across all the pixel stacks, one should reduce its size a little, and make all the other pictures a little bigger. The mean transformation is thus null.

In this domain, the cost function is represented by the sum of all pixel stacks' entropies. When using the classical set theory, the following equation would thus apply :

$$\sum_{i=1}^{n}\left(-\sum_{m\times m}\left(\frac{1}{n}\sum_{j}x_i^{j'}(k)log_2\frac{1}{n}\sum_{j}x_i^{j'}(k)\right)\right)$$

As x_i^j represents the i^{th} pixel of the j^{th} image, $x_i^j(k)$ is the probability of the k^{th} element on the multinomial distribution in x_i^j. However, this computation does not properly take into account the nuances and diversity that can be present in some regions of the image stack. For this reason, fuzzy logic is preferred when exploiting entropy to align pictures.

C. *Fuzzy logic and entropy*

In the regular set theory, an element is contained or not in a set. Fuzzy logic is a generalization of it where this relation is not binary or crisp. Instead of being discrete, it will rather be fuzzy or real. The membership of an element x to a fuzzy set a is determined by its membership function, called $\mu_a(x)$. In the context of Chinese character recognition, this is used to measure how a pixel is contained by the set represented by all the other pixels from the pixel stack.

By using this tool, we can thus compute the fuzzy entropy instead of the regular entropy. This allows for a better consideration of the useful variations during the training. The total fuzzy entropy represents the sum of all the fuzzy entropies :

$$-\frac{1}{n}\sum_{i=1}^{n}log\left(\lambda_i\right)$$

Where λ_i is the fuzzy entropy of the pixel stack i. To compute it, each pixel of the stack needs to be compared to all the other pixels of the same stack.

$$\lambda_i = \sum \mu_i(x,y)$$

As the number of pixel is n, there is thus $\binom{n}{2}$ comparisons to do, which is very expensive computationally.

Such comparisons are thus made with the selected membership function. It is made arbitrarily by the creator of the artificial intelligence. The implementation of the following three functions gave good precision rates in the recognition of Chinese characters :

$$\mu_i(x,y) = 1 - \frac{|i(x)-i(y)|}{|i_{max}-i_{min}|}$$

$$\mu_i(x,y) = exp\left(-\frac{(i(x)-i(y))^2}{2\sigma_i^2}\right)$$

$$\mu_i(x,y) = max\left(min\left(\frac{(i(y)-(i(x)-\sigma_i))}{(i(x)-(i(x)-\sigma_i))},\frac{i(x)+\sigma_i-i(y)}{i(x)+\sigma_i-i(x)}\right),0\right)$$

However their complexity causes very long recognition times.

$$\begin{bmatrix} cos(\theta) \times e^{s_x} + (cos(\theta) \times e^{s_x} \times h_x - sin(\theta) \times e^{s_y}) \times h_y & cos(\theta) \times e^{s_x} \times h_x - sin(\theta) \times e^{s_y} & t_x \\ sin(\theta) \times e^{s_x} + (cos(\theta) \times e^{s_y} + sin(\theta) \times e^{s_x} \times h_x) \times h_y & cos(\theta) \times e^{s_y} + sin(\theta) \times e^{s_x} \times h_x & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. The transformation matrix used to reposition each image.

III. Optimization of the image congealing process

When using the described method to make an artificial intelligence capable of recognizing handwritten Chinese characters, the obtained precision is acceptable. However, it can take up to 15 seconds to identify one ideogram. As explained earlier, this prevents any possibility of releasing such a program to the general public, especially on mobile devices. Several contributions and modifications has been developed here in order to speed-up the recognition process.

When processing the fuzzy entropy across the whole stack, $m \times m$ sub-entropies are computed, because there are $m \times m$ pixels in a picture, and because the total entropy is the sum of their own entropy. This is a heavy task, even for small resolution images. Furthermore, this prevents the exploitation of such an algorithm on large, high resolution pictures because its complexity is at least $O(n^2)$. This is quite problematic in the current context where even embedded devices can deal with high definition resolutions. This implies that a subsampling of the image would be required before proceeding to the identification. To break this constraint, it is proposed to compute only a subset of the pixel stacks. By processing only a fraction of them, the number of fuzzy entropies to calculate is divided by the same fraction. It can be considered to only take into account 1 pixel stack out of 2, 3, or 4, for example.

When mentioning the membership function in section 2, it has been noticed that the three proposed are complex and are composed of many sub-operations. Once a test sample has been submitted, the membership function will be computed $\binom{n}{2} \times m^2 \times \#iterations$ times for each possible ideogram. For example, if a person uses 64×64 pixels images as in the HCL2000 dataset, and if the training is relying on 20 samples for each symbol, the membership function will be executed 2334720 times per possible answer. It can be deduced from this number that minor optimizations in this part can have big consequences on the final time of execution.

Implementing simpler membership functions is proposed to speed-up the recognition of a Chinese character. To be exploitable in this context, a function needs to follow some rules. First, it should return 1 if the pixel is perfectly integrated in the set represented by its pixel stack. On the contrary, it should return 0 if the pixel cannot be assimilated to the same set. If the function contains a division, the algorithm should take care of the case where the denominator is equal to 0. In the same way, if it contains a logarithm, it should specifically deal with any negative parameters.

Two new membership functions are introduced in this paper. The first one is based on a linear model, the second one on an exponential model.

$$\mu_i(x,y) = 1 - |i(x) - i(y)|$$

$$\mu_i(x,y) = exp\left(-\left(i(x) - i(y)\right)^2\right)$$

These two membership functions contain few sub-operations. They also respect the criteria explained above. However they are based on less parameters, and cannot approximate perfectly the classical functions. The precisions obtained using these two formulas thus need to be tested and measured.

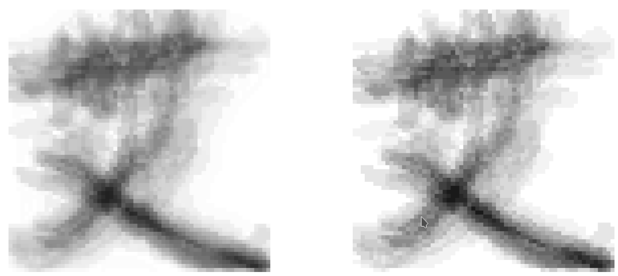

Fig. 3. On the left, the first mean image of a stack during training. On the right, the last mean image, after 15 iterations of picture congealing.

IV. Experimental evaluation

A. Context

To compare with existing results, the same context and tools have to be retrieved. The HCL2000 dataset will be used. It contains the 3755 characters from the simplified set, written by 1000 different writers. Additionally, information about each writer is specified, such as age, gender, education, occupation, etc. The images in this dataset have a resolution of 64×64 pixels.

The algorithm will be implemented using *MATLAB*. During training, pictures congealing will happen on stacks containing 20 samples chosen randomly. 15 iterations will be done during training, 3 only during recognition. Thus, during training, for each possible ideogram, the submitted test sample will be transformed (and eventually reversed if there is not any drop of fuzzy entropy) three times, above a stack of fifteen mean images. The mean images' pixels can contain 255 different shades of grey.

The experiment will take place on subsets of the Chinese simplified character set. One only containing characters with a single structure, another one dedicated

Fraction of stacks processed	1	1/2	1/3	1/4
Speed of execution	1	1.91	2.93	3.93
Precision	1	0.97	0.94	0.83

Fig. 4. The obtained results for various subsampling values, as ratios of the original version.

to left and right structures, a third for upper and lower structures, a fourth with "surrounded by" structures, and finally a fifth containing only characters with a framework structure.

Finally, it should be noted that the original results have been measured by using the first, classical fuzzy membership function presented in section 2.C.

B. Results

1) Fuzzy entropy subsampling: The consequences provoked by the fuzzy entropy subsampling are presented in figure 4.

The fact that the ratios representing the speed of execution are almost identical as the denominators of the fractions of stacks processed is significant. It validates the theory that the gains in speed are proportional to this parameter. The loss in precision was forecasted, as less analyzing the stack's entropies can only result in a poorer knowledge of the images' alignment. It is noticeable that ignoring one stack out of two almost doubles the speed for a precision almost untouched.

2) Reduced membership functions: The precisions with which the algorithm performs depending on the membership function used are presented in figure 5.

Character subset	Original linear	Simplified linear	Simplified exponential
Single	100	97.18	96.94
Left and right	92.31	84.59	82.64
Upper and lower	85.00	78.12	80.06
Surrounded by	83.33	74.33	74.10
Framework	100	98.84	97.83

Fig. 5. The obtained precision for each membership function, as percentages.

As expected, the simplified functions cannot perform as well as the complete version. Differences go from less than two percents to almost ten percents. Ideograms having a single or framework structure are the least affected by the changes of implementation. Indeed, they often are simple and more easily distinguishable from each other. In the end, the obtained speed-ups from the new functions are $\times 3.13$ for the simplified linear version, and $\times 2.48$ for the simplified exponential version.

V. Conclusion

By considering the use of state-of-the-art fuzzy entropy in handwritten Chinese character recognition, and adding on top of it, this paper has proposed optimizations that make this technique more credible for general public applications. The compromise between speed and precision could maybe be avoided by making this algorithm work in collaboration with another kind of artificial intelligence. Indeed, recognition with the help of fuzzy entropy achieves a precision of 100% on some subsets of the Chinese simplified characters. If another program could define the subset before the recognition start, it might be quite efficient.

One important thing to note is that the presented program reuses a large number of times the same, small algorithm for computing entropies. Also, this is the case on both training and recognition. Both of these tasks share most of their source code. This clearly brings hope in a hardware implementation of it. Indeed, it would provide a more adapted treatment. Also, and more importantly, it might be quite cheap, as one single unit for entropy processing could achieve most of the work in this program.

Finally, this work can be extended to all kind of recognition and is not specific to handwritten Chinese characters. Any other alphabet could use it. View-from-sky maps could potentially be aligned using fuzzy entropy as here. Also, this paper deals essentially with two-dimensional data and transformations; but the algorithm could be adapted to work with three dimensions or even more.

Acknowledgment

The author would like to thank M. Ferat Sahin for his help and his teaching during the making of this project, as well as for his proposals about the creation of new membership functions.

References

[1] Zhang H, Jun G, Chen G, Li C, *HCL2000 : a large-scale handwritten Chinese character database for handwritten character recognition*. In: Procedding of the 10th IEEE international conference on document analysis and recognition ICDAR'09. IEEE, New York, pp 286–290, 2009

[2] Fangyi Li, Qiang Shen, Ying Li, Neil Mac Parthaláin, *Handwritten Chinese character recognition using fuzzy image alignment*. 17 November 2015.

[3] W. Y. Liu, J. L. Jiang, *A new Chinese character recognition approach based on the fuzzy clustering analysis*, 12 November 2013.

[4] Cheng Cheng, Xu-Yao Zhang, Xiao-Hu Shao, Xiang-Dong Zhou, *Handwritten Chinese Character Recognition by Joint Classification and Similarity Ranking*, 23-26 Oct. 2016

[5] Qiu-Feng Wang, Fei Yin, Cheng-Lin Liu, *Handwritten Chinese Text Recognition by Integrating Multiple Contexts*, In : IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 34, NO. 8, AUGUST 2012

[6] Cheng-Lin Liu, Fei Yin, Da-Han Wang, Qiu-Feng Wang, *Online and offline handwritten Chinese character recognition: Benchmarking on new databases*. In : Pattern Recognition, Volume 46, Issue 1, January 2013, Pages 155-162

[7] Xu-Yao Zhang, Yoshua Bengio, Cheng-Lin Liu, *Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark*. In : Pattern Recognition, Volume 61, January 2017, Pages 348-360