A Robust Skeletonizer for Farsi Image-texts

Vahab Pournaghshband
Computer Science Department
University of California, Berkeley
vahab@cs.berkeley.edu

Abstract

For some document processing applications such as optical character recognition programs, it is crucial to transform an image-text into a skeleton of a unit width as a preprocessing step. In this paper we propose a robust skeletonizer that effectively transforms the image-texts in Farsi into its thinned version. Our proposed algorithm is a combination of various thinning approaches for different languages. We present the algorithm in depth by presenting each stage in details.

1 Introduction

1.1 Skeletonization

Skeletonization or thinning is a procedure which transforms a pattern to a skeleton of a unit width (i.e. one pixel thickness). It has been proven effective in a broad range of image-processing applications including character recognition. Generally, for a skeletonization algorithm to be effective, it should ideally compress data and retain the significant features of the pattern. While there is no formal definition of skeletonization, we define it informally and somewhat intuitively as a method in digital image processing to extract a skeleton of an image that still preserves the images original topology.

Skeletonization is one of the essential parts of any Optical Character Recognition (OCR) program. It is because the OCR engine can work robustly on unit width skeleton of the image as opposed to take different strategies for different visual forms of a text due to the font, size, or boldness.

Although the design of thinning algorithms has been a very active research area, very few researchers addressed the thinning of Farsi characters. Due to the characteristics of Farsi characters, they do not allow direct application of techniques developed for other languages, for they are either too simple (e.g. Latin), or too complex (e.g. Chinese).

1.2 Written Farsi Characteristics

Some of the characteristics of Farsi characters are as follows:

1. Farsi is a cursive language, that is, characters are connected and make a component that is called sub-word. A word consists of multiple sub-words.

2. It has 32 letters each which has 2-4 representations depending on the position. For instance, some letters have three different shapes if they are at the beginning, middle, or end of the sub-word (Figure 2).

3. Although, there are up to four forms for Farsi letters, they share similar and sometime exact topologies (Figure 1).

4. Farsi is a dot-oriented language and some letters even contain three dots. The dots maybe below, above, or even inside the letter.

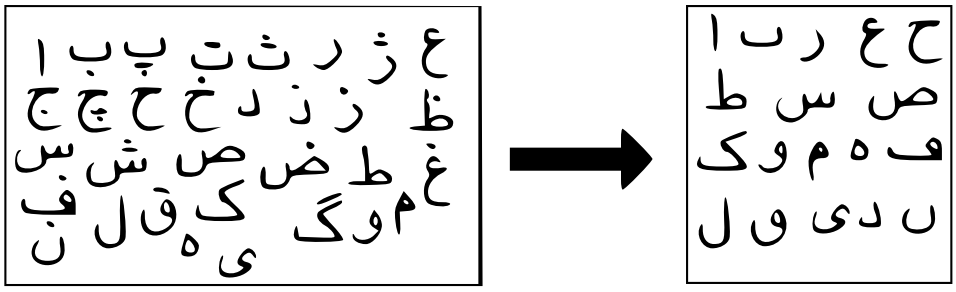


Figure 1: Extracting shared topologies for Farsi isolated letters.

Persian Alphabet / الفبای فارسی

Detached	Initial	Medial	Final	Roman	Name	Detached	Initial	Medial	Final	Roman	Name
ا	ا	ـا	ـا	á	alef	ص	صـ	ـصـ	ـص	ṣ	sád
ب	بـ	ـبـ	ـب	b	be	ض	ضـ	ـضـ	ـض	ḍ	zád
پ	پـ	ـپـ	ـپ	p	pe	ط	طـ	ـطـ	ـط	ṭ	tá
ت	تـ	ـتـ	ـت	t	te	ظ	ظـ	ـظـ	ـظ	ẓ	zá
ث	ثـ	ـثـ	ـث	th	se	ع	عـ	ـعـ	ـع	‘	ayn
ج	جـ	ـجـ	ـج	j	jim	غ	غـ	ـغـ	ـغ	gh	ghayn
چ	چـ	ـچـ	ـچ	ch	che	ف	فـ	ـفـ	ـف	f	fe
ح	حـ	ـحـ	ـح	ḥ	ḥe	ق	قـ	ـقـ	ـق	q	qáf
خ	خـ	ـخـ	ـخ	kh	khe	ک	کـ	ـکـ	ـک	k	káf
د	د	ـد	ـد	d	dál	گ	گـ	ـگـ	ـگ	g	gáf
ذ	ذ	ـذ	ـذ	dh	zál	ل	لـ	ـلـ	ـل	l	lám
ر	ر	ـر	ـر	r	re	م	مـ	ـمـ	ـم	m	mím
ز	ز	ـز	ـز	z	ze	ن	نـ	ـنـ	ـن	n	nún
ژ	ژ	ـژ	ـژ	zh	zhe	و	و	ـو	ـو	v/ú	váv
س	سـ	ـسـ	ـس	s	sin	ه	هـ	ـهـ	ـه	h	he
ش	شـ	ـشـ	ـش	sh	shin	ی	یـ	ـیـ	ـی	y/í	ye

Figure 2: Different shapes of Farsi letters[7].

Knowing the above characteristics, an effective skeletonizer for Farsi must meet the following requirements in order of its significant effect on the recognition process of Farsi OCR:

1. preserve the connectivity and disconnectivity of skeleton
2. converge to skeletons of unit width
3. does not introduce new branches (known as spurious branches)
4. approximate the medial axis or informally preserving the topology of the image (is a method for representing the shape of objects by a set of curves which roughly run along the middle of an object.)
5. achieve a high data-reduction efficiency

2 Approach

The proposed algorithm consists of four stages and follows a waterfall model. The first two, global and local binarization, directly affect the skeletonization process which itself breaks into two stages: core-skeletonization and post-skeletonization. In summary, our proposed thinning system for Farsi image-text is illustrated in figure 3. The details for individual stages are provided in the subsequent sections.

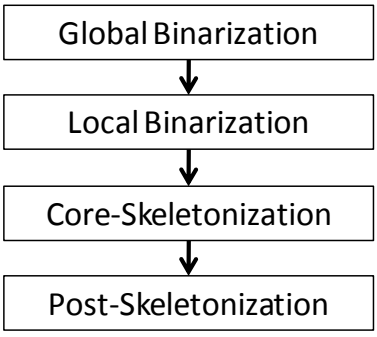


Figure 3: Our proposed thinning algorithm.

2.1 Binarization

In OCR applications, skeletonization is normally considered as the last pre-processing step before the OCR engine comes to play. Having passed through all the pre-processing filters prior to skeletonization, the image is expected to be in the sharpest, least noisy, and monochromatic form. Hence, for correct testing of the proposed skeletonizer, the synthetic input must be well-representative. To assure this, snapshots of text images were used instead of scanned images since scanners are likely to introduce noise to the data. But our input is not yet complete. Skeletonizer assumes a monochromatic image that well-separated the text from the background. Binarization, a very important filter that proceeds skeletonizer, is responsible to provide skeletonizer with such data. Binarization, also referred to as two-tone converter or masking in image processing context, receives a chromatic image, in our case a 256 gray-level text-image, and converts it into 0/1 (data and non-data) monochromatic image. The significant effect of binarization on skeletonization motivates us to seek for an effective binarization for our skeletonizer. is the incentive to seek for a good binarization method for our specific skeletonizer. Figure 4 shows an example of poor binarization that led to a poor result for the skeletonizer.

(a) Original image-text (b) Poor binarization (c) Skeletonization

Figure 4: Effect of poor binarization on skeletonization.

There have been researches in binarization for documents. They mainly apply a global thresholding method and some use dynamic local thresholding methods. The global thresholding is remarkably faster but will introduce critical compromises such as the one in figure 4 (notice how the two letters are connected in (c) and not in (a)). On the other hand, local thresholing would produce a better result but increases the time complexity to an impractical level since OCRs tend to consume a substantial portion of their

time budget on the recognition stage rather than pre-processing steps.

Among all global thresholding algorithms, Otsu's algorithm [6] has been known to be most suitable for text images since it well utilizes the fact that there is often a substantial and consistent gap between text color and the background. It assumes a bimodal histogram for the image and seeks for a cutoff, T, that minimizes the variance of each of the two groups. The histograms itself is produced by the probability (frequency) of each level of grayness.

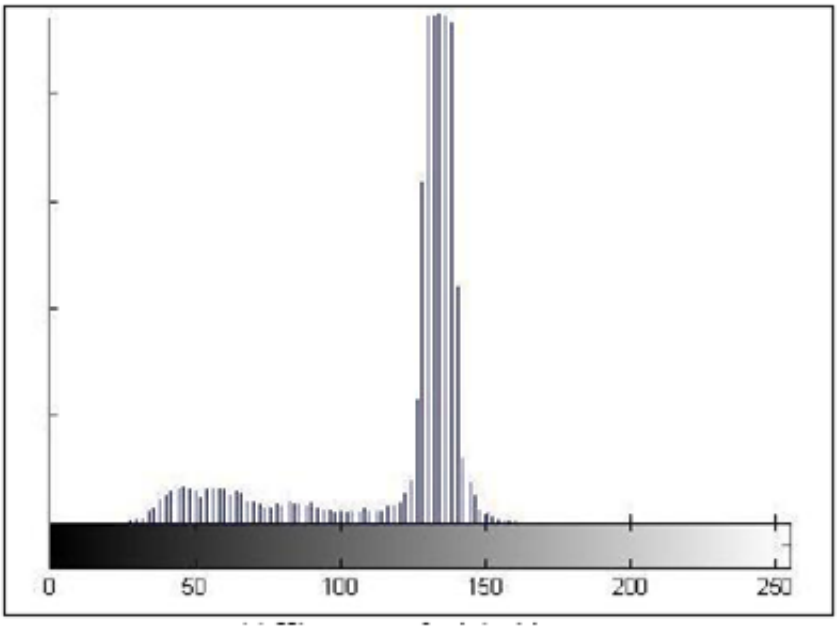


Figure 5: Histogram of an image of a document.

A hybrid algorithm for binarization is as the following presented by Chang et al[1]: First, we resort to the static threshold T from some global binarization method, in this case we use what was obtained at the previous step, for segmenting coarse objects from the document images. This value, T, divides the pixels of the image into two parts. Let M_1 denote the statistical average of the lower part (those whose gray values are less than T). We now consider the set S of all pixels p such that p lies within a 5×5 neighborhood of q with $M_1 \leq g(q) \leq T$, where $g(q)$ denotes the gray value of q. For those

pixels in S, we shall determine their binary values later. For the rest of the pixels, we consider them as either solid black or solid white and assign their binary values as follows. For every pixel p not in S, $b(p) = 1$, if $g(p) < M_1$; or $b(p) = 0$, if $g(p) > T$, where $b(p)$ denotes the binary value of p, 1 stands for black and 0 for white. Now, let us focus on the pixels in S. They are considered as pixels for which there is high probability for a static-threshold method to misjudge their binary values. Thus, we will further examine those pixels with a window-based binarization scheme. The window centered at each of such pixels will be chosen to be sufficiently large to cover representatives from both foreground and background. Thus, a reasonable size of the window has to be related to the scale of the objects. Note that there are some of the pixels in S whose scales have not been defined. They are those pixels whose gray values exceed T but they fall within a 5×5 neighborhood of some pixel whose scales are well-defined. If p is such a pixel and q is the center of the 5×5 neighborhood, then we simply define the scale of p as that of q. Now, if p falls within a region whose scale has been determined as j, we will choose the window $W(p)$ to be a neighborhood centered at p and of size $(2^j) + c$, c being an odd number.

(a) Poor binarization

(b) Chang et al's hybrid binarization

Figure 6: Comparison of binarization methods.

2.2 Skeletonization

2.2.1 Core-skeletonization

Among most popular thinning algorithms such as Stentiford[8] and Zhang-Suen[4], Haung et al's[3] meets the first, second, and forth requirements best for Farsi skeletonizer discussed in section 1.2. With a good post-skeletonization the third requirement would also be met. Huang et al proposed a fully parallel thinning algorithm which involves one iteration in each pass. In parallel thinning algorithms, pixels are examined for deletion based on the results of only the previous iteration. Like most template-based thinning algorithms, it uses the information of 3×3 windows (i.e. the state of 8-neighbors), but in order to preserve connectivity, 3×4, 4×3 and 4×4 masks are also used. Below we present a summary of the Haung et al's algorithm[3]:

All the rules in Table 2 are applied simultaneously to each pixel p. Pixel p will be flagged if it matches any of the rules above, knowing that the center of each template is the pixel p. The rules in Tabel 2 remove two-pixel-width rectangular patterns, resulting in loss of information or pattern connectivity. To obviate this problem, usually refer as excessive erosion, the pixel p is preserved (not flagged) if it matches any of the following templates:

X	0	X
1	**1**	1
1	1	1
X	0	X

(1)

X	0	0
1	1	0
0	**1**	1
0	0	X

(2)

X	0	0
0	**1**	1
0	1	1
0	0	X

(3)

0	0	0	0
0	**1**	1	0
0	1	1	0
0	0	0	0

(4)

X	0	0	0
0	**1**	1	0
0	0	1	X

(5)

X	1	1	X
0	**1**	1	0
X	1	1	X

(6)

0	0	0	X
0	1	**1**	0
X	1	0	0

(7)

Table 1: The preserved template used in Haung et al's algorithm ('x' means do not care).

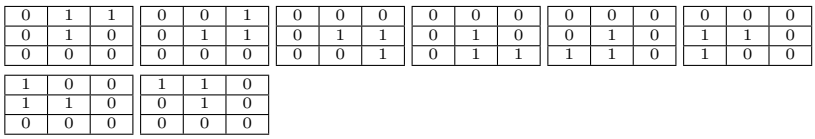
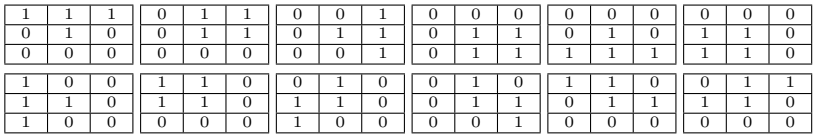
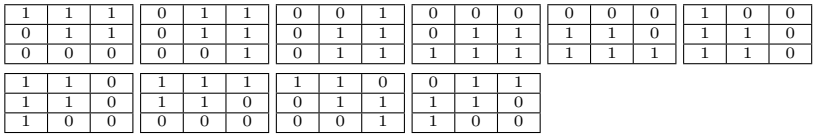
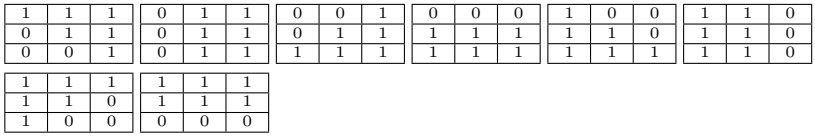
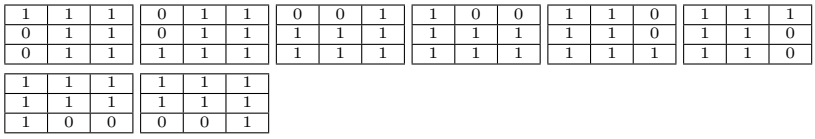
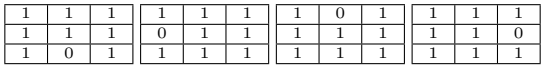
Amount	Elimination Rules
0	Never
1	Never
2	
3	
4	
5	
6	
7	
8	Never

Table 2: Haung at al's elimination rules (0 denotes the white pixel).

At the end of a pass, if there is no flagged pixel the algorithm stops; otherwise the flagged pixels are removed and the next pass starts.

2.2.2 Post-skeletonization

Most OCRs are very sensitive to branches of any size. A major drawback in using Haung et al's algorithm as a preprocessing tool for OCRs is that the created skeleton usually contains spurious branches in presence of large fonts. These branches are considered as noise and must be removed. The proposed algorithm to remove spurious branches is presented after some important definitions.

Definition. The connectivity number of a pixel, C_n, is formally defined as follows:

$$C_n = \sum_{k \in S} (N_k - (N_k \cdot N_{k+1} \cdot N_{k+2}))$$

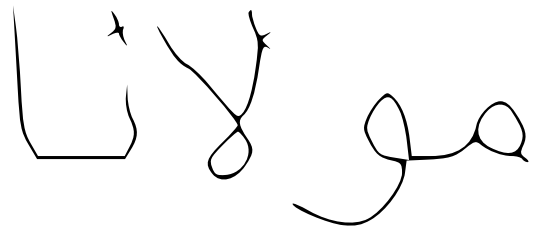
Where N_k is the color of the eight neighbors of the pixel analyzed, N_0 is the center pixel, N_1 is the color value of the pixel to the right of the central pixel and the rest are numbered in counterclockwise order around the center, and $S = 1, 3, 5, 7$ (Table 3).

4	3	2
5	0	1
6	7	8

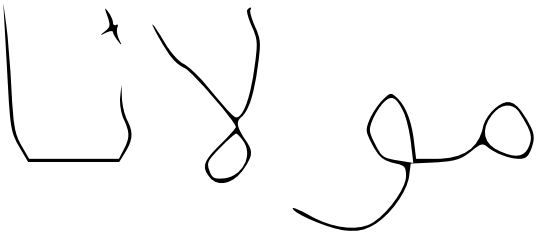
Table 3: Ordering of the pixels for connectivity number calculation..

Definition. An end-point is a black pixel in the skeleton having a connectivity number one, or in other words, the image remains connect upon the removal the pixel.

The algorithm to remove spurious branches is suggested by Liao et al[5??] as follows: first, for each end-point p, the radius R_p of the largest circle of black pixels within the original image that is centered at p is evaluated. Then, the nearest non-end-point q to p is found, and the link between p and q is removed if $dist(p, q) < R_p + R_q$, where $dist$ is the Euclidian distance. A sample result of post-skeletonization using Laio's algorithm is illustrated in Figure 7.



(a) Before



(b) After

Figure 7: The effect of post-skeletonization.

References

[1] Chang, F., Liang, K., Tan, T., and Hwang W. "Binarization of Document Images Using Hadamard Multiresolution Analysis," *Document Analysis and Recognition, Fifth International Conference on Document Analysis and Recognition (ICDAR'99),* p. 157 1999.

[2] Haji, M. (Jan 2005). "Farsi Handwritten Word Recognition Using Continuous Hidden Markov Models and Structural Features," *MS. Thesis,* University of Shiraz, Iran.

[3] Huang, L., Wan, G. and Liu, C. (2003). "An Improved Parallel Thinning Algorithm," *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003),* pp. 780-783.

[4] Lam, L., Lee, S., and Suen C. (1992). "Thinning Methodologies - A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol 14, No 9.

[5] Liao, C., and Haung, J. (1990). "Stroke Segmentation by Bernstein-Bezier Curve Fitting," *Pattern Recognition*, Vol 23, No 5, pp. 475-484.

[6] Otsu, N. "A threshold selection method from gray-scaled histogram," *IEEE Trans. Systems, Man, and Cybernetics,* Vol. 8, pp. 62-66, 1978.