Retrieval of Rashi Semi-Cursive Handwriting via Fuzzy Logic

Eran Gur
Department of Electronics Engineering,
Jerusalem College of Engineering,
Jerusalem, 91035, Israel
e-mail: gur.eran@gmail.com

Zeev Zalevsky
School of Engineering,
Bar-Ilan University,
Ramat-Gan 52900, Israel
e-mail: zalevsz@biu.ac.il

Abstract **-** Text recognition and retrieval is a well known problem. Automated optical character recognition (OCR) tools do not supply a complete solution and in most cases human inspection is required. In this paper the authors suggest a novel text recognition algorithm based on usage of fuzzy logic rules relying on statistical data of the analyzed font. The new approach combines letter statistics and correlation coefficients in a set of fuzzy based rules, enabling the recognition of distorted letters that may not be retrieved otherwise. The authors focused on Rashi fonts associated with commentaries of the Bible that are actually handwritten calligraphy.

Keywords - fuzzy logic; document analysis systems; character recognition;

I. INTRODUCTION

Automatic recognition and analysis of text and letters has high importance in the scientific community [1-5]. In texts where part of the letters is cropped or are given in low quality, there's a difficulty in recognizing the letters in order to understand the text. An attempt to recognize the letters by means of correlation alone, between the cropped letter and a databank of letters might not be sufficient. In many cases the deterioration in quality leads one letter to look like another and the correlation test will fail. The problem of low quality letters has very high importance when religious scripts are concerned. Errors in text in the Bible and its commentaries are commonly unacceptable. In recent years quite a few research papers were published in the field of character recognition and data retrieval. Two of the most common methods, used in relation to computer vision are Artificial Neural Networks (ANN), and Support Vector Machines (SVM). Following the work of Brown in 1993 [6], many algorithms used ANN for character recognition quite successfully but they all require a large training set and complex network topologies [7-9]. The SVM method, also known as the "maximal margin classifier" method was first introduced in a book by Vapnik [10]. It had proved to yield better results than the ANN approach for many pattern recognition applications. However, it requires quite complex mathematical operations and a large training set. For this reason we were looking for a simpler solution for cases where one lacks the ability to build a large training set or to perform complex calculations, and he has to account on similarity – or correlation.

In this paper we suggest a new approach based on fuzzy reasoning, taking into account three parameters rather than just the correlation coefficient (being a common tool to quantify similarity). The first parameter remains the correlation between the cropped or ill posed letter and a databank of letters in a similar font. The second parameter is the frequency of letters' appearance, e.g., if two letters have a similar correlation coefficient but one of them is much more frequent than the other we will use this data to give the frequent letter a better grade in our fuzzy logic engine. A good example for this is the letter Lamed 'ל' (the Hebrew version of L) and the letter Zadik 'צ' (the Hebrew version of Z), in Rashi font that will be discussed shortly, they share a high correlation coefficient and look similar, however Lamed has a frequency of appearance of 6.29% and Zadik of only 1.28% giving an advantage to Lamed. The third parameter is again a correlation coefficient but without the mid-third of the letter (i.e., only the top and bottom thirds), this parameter represents two important qualities. First, some letters deviate from the central region more than others (like the letters 'b' and 'y' in English letters or Lamed 'ל' or Koof 'ק' in Hebrew) and we wish to use this property. Second, in most cases when ink is used for writing (as in the case of the Bible and its commentaries) the writing starts from top to bottom so that the top third of the letter is solid and clear and may be used to achieve correlation with a large probability for correct recognition.

These three parameters are combined by using a fuzzy connotation rule table such as "if the total correlation is high, the frequency of the letter is medium and the partial correlation is very high then the likelihood (or membership grade) of the current letter is high". This is done using a 3-D matrix where the number of elements in each axis is the same as the number of possible values for each parameter (e.g. very low, low, medium, high, very high – 5 values).

A 3-D Gaussian multiplies the matrix and the algorithm calculates the center of gravity of the new 3-D matrix and

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.262

sets it as a grade for the current letter taken from the databank. The algorithm may be supplemented by hard-decision logic such as: the Hebrew letter Mem-Sofit 'ם' that can appear only at the end of a word and never in the beginning or center of a word, and in the same way the letter Mem 'מ' that can never appear as the last letter of a word.

II. FUZZY LOGIC BASICS

Fuzzy logic was first introduced by Zadeh in 1965 [11,12]. The basic idea in Fuzzy Logic (FL) is displaying data as a part of a fuzzy set rather than a crisp, single value. A fuzzy set is similar to a classical set except that in a classical set, data can either belong to the set or not, whereas in a fuzzy set the data will always belong to the set, but with a different degree. The degree of belonging to a fuzzy set is called a Membership Grade and is often marked as MG. The MG depends upon the structure of the fuzzy set, described by a Membership Function, usually marked as MF. In Fig. 1a, we draw a classical set. If x is between x0 and x2 then it belongs to the set completely (this is denoted by MG=1). However, if x is not included in the interval it does not belong to the set (this is denoted by MG=0). For example, if x is height in centimeters and [x0, x2]=[170, 180] describes an average height of an adult, then x=179 is included in the set but x=181 is not. This example indicates how 'illogical' it is to use classical sets for certain data.

In Fig. 1b, we use a MF (known as a π-function) to describe the fuzzy set. Many times a triangular MF is used, due to the mathematical simplicity of its representation.

When applying fuzzy logic principles to control procedures, one must define fuzzy sets for the entire universe of discourse. An example for such a definition is given in Fig. 2.

In Fig. 2 Xn is the lowest value that can be assigned to x, while Xm is the highest. MF1...5 are the membership functions covering the entire domain. For example, if x is height in centimeters, then we may define [Xn,Xm]=[135,215], this will include (almost) all of the population. This way MF1 represents very short people, MF2 represents short people, MF3 represents average people, etc. One should note that since MF1 and MF5 are at the edges of the domain, they are described by monotone functions.

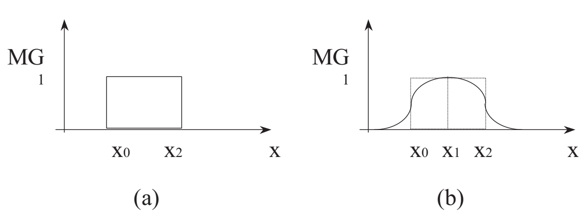

Figure 1. Comparison of (a) a classical set and (b) a fuzzy set.

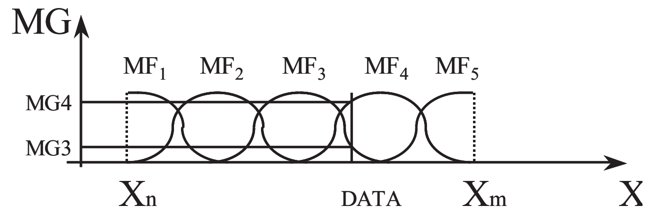

Figure 2. A fuzzy set representation of the region [Xn,Xm] by five membership functions MF1 to MF5, including an illustration of finding two non-zero membership grades for a given data.

The fuzzy control inference engine for a dual-input, single output case is described below. The controller accepts input data x0 and y0 from two independent channels, e.g., temperature and pressure. The outcome z, e.g., torque, will be the outcome of the controller. The first step is matching x0 against all MFs in the first (temperature) domain and y0 against all MFs in the second (pressure) domain. Two sets of MGs are obtained for x0 and y0 and we mark them {μ}, {η} respectively.

At this stage, pre-written rules are applied to the data. An example for such rules is given in Eq. 1.

$$\begin{aligned}&\text{rule 1: if } x \in MF_3^{temp} \text{ and } y \in \mathrm{MF}_3^{\mathrm{pres}} \text{ then } z \in \mathrm{MF}_3^{\mathrm{tork}}\\&\text{rule 2: if } x \in MF_4^{temp} \text{ and } y \in \mathrm{MF}_3^{\mathrm{pres}} \text{ then } z \in \mathrm{MF}_4^{\mathrm{tork}}\end{aligned} \quad (1)$$

According to these rules, the controller establishes MGs for the output. For the given rules we may write the following connections (using the notations listed above):

$$\begin{aligned}&\mu_3 \text{ AND } \eta_3 \Rightarrow \chi_3\\&\mu_4 \text{ AND } \eta_3 \Rightarrow \chi_4\end{aligned} \quad (2)$$

χ being the membership grade for torque (z).

The logical 'AND' is usually interpreted as a minimization (Eq. 3) as in possibility theory [13].

$$\begin{aligned}&\min(\mu_3, \eta_3) = \chi_3 = \mu_3\\&\min(\mu_4, \eta_3) = \chi_4 = \eta_3\end{aligned} \quad (3)$$

This procedure is to be performed on all rules, creating a set of {χ} MGs for the torque output. Next, each χ is matched against the proper MF of the torque. The surface beneath each χ value is marked and finally, the inference engine calculates the center of gravity (COG) of these surfaces as shown in Fig. 3.

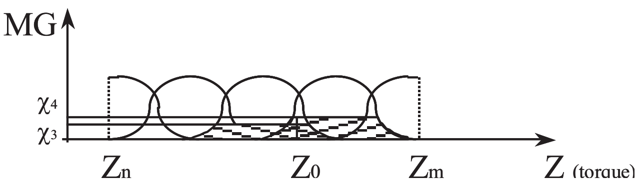

Figure 3. Finding the outcome of a fuzzy inference engine, using a center of gravity method.

In 1995 Gowan [14] published a pioneering work on character recognition via fuzzy logic; however in his work he used only 3 MFs for each parameter, which the authors of this paper find to be insufficient. In the approach suggested in this work 5 MFs are being used for each parameter for better accuracy. Other related work manipulate fuzzy logic [15] or use it as a part of a more intensive approach [16], thus adding to the overall complexity of recognition and retrieval.

III. RASHI FONT RECOGNITION AND RETRIEVAL

Shlomo Yitzhaki (February 22, 1040 – July 13, 1105), better known by the acronym Rashi (RAbbi SHlomo Itzhaki), was a medieval French rabbi famed as the author of the first comprehensive commentary on the Talmud, as well as a comprehensive commentary on the Tanakh (Hebrew Bible). He is considered the "father" of all the Hebrew commentaries that followed on such as the Talmud (i.e., the Baalei Tosafot) and the Tanakh interpretations (i.e., Ramban, Ibn Ezra, Ohr HaChaim, et al.) [17-20]. The Rashi typeface (which was not used by Rashi himself) is based on 15th century Sephardic semi-cursive handwriting, and it was first used in a 1475 version of the Bible, printed in Reggio di Calabria, Italy. Since a Bible fit for religious ceremonies must be handwritten by a Sofer Setam (a copyist of religious material), the Bible and its commentaries were both handwritten.

Fig. 4 shows the Rashi letters representing the Hebrew letters as well as the prime and double-prime symbols, many times used for quotes and abbreviations.

In this work we used a data base of just over 3000 semi-blurred words in Rashi font (employing over 16,000 letters). Unlike most modern techniques we did not attempt to recognize an entire word or phrase but rather single letters since the words and sentences written over 500 years ago might not always have a similar meaning today.

Fig. 5 demonstrates a case in which one of the letters in a word is very weak. In this case the central letter in a 5-letter word is unclear and we wish to resolve it.

To explain the algorithm we start with a 2-D fuzzy logic rule table presenting the rules associated with the total correlation and the frequency of appearance.

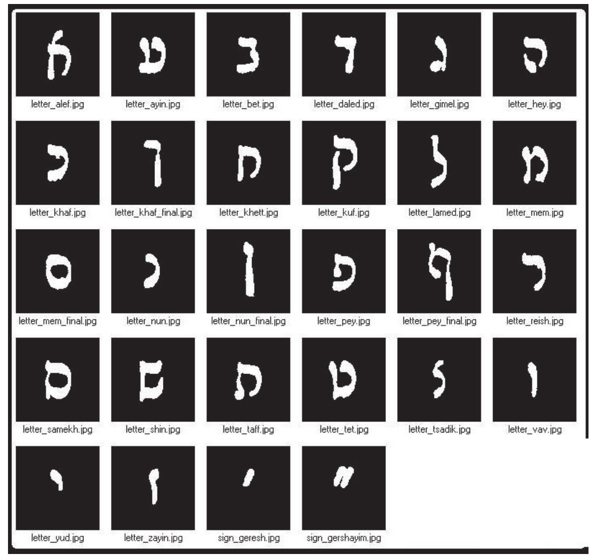

Figure 4. Hebrew letters in the font associated with Rashi.

As shown in Fig. 6, the frequency of appearance of a letter is divided to 5 possible choices from "very rare" to "very frequent", these choices are associated in the field of fuzzy logic to 5 MFs, respectively. The correlation values are also sectioned into 5 choices and for each combination a number is produced, this number is the MG. E.g., in Fig. 6 a medium correlation coefficient and a frequent letter yield a grade of 2. This table of rules was determined empirically after many synthetic tests in which the required letter was known. Note that in this example only extremely high frequency of appearance of a letter can influence the MG.

Since we are dealing with words from the Bible or its commentaries we need the statistics of appearances of each letter in the Bible. This statistics is given in Table I. In order to give the very rare letters a chance to be selected, the algorithm adjusts the frequency range to a linearly distributed range, without changing the order of frequencies.

At this stage we add the third parameter for the fuzzy logic inference engine. This is the correlation coefficient excluding the center of the letter. As stated before, implementing the fuzzy logic based rules is done by using a 3-D matrix, such a matrix is represented by its surfaces in Fig. 7. In addition, a flow chart of the proposed algorithm is given in Fig. 8.

Figure 5. A 5 letter word in which the 3rd letter is unclear.

	Very rare letter (e.g. ף ץ)				Very frequent letter (e.g. י ו ה)
Small correlation (close to 0)	0	0	0	0	0
	1	1	1	1	1
	2	2	2	2	2
	3	3	3	3	4
Large correlation (close to 1)	4	4	4	4	5

Figure 6. Two dimensional rule table for a fuzzy logic inference engine.

At the 1st stage the input image is entered, the specific letter is filtered and re-sampled to 64 by 64 pixels. Then, it is compared to the 1st letter-image in the databank. The frequency of appearance of the letter, according to Table I, is normalized to a small range using Eq. 4.

$$x_0 = (freq - 0.8) \times 495 + 1 \tag{4}$$

TABLE I. HEBREW LETTERS AND THEIR FREQUENCIES OF APPEARANCE IN THE BIBLE.

3.74% ע	11.52% י	7.99% א
1.31% פ	2.79% כ	5.45% ב
0.21% ף	1.17% ך	0.84% ג
0.98% צ	7.38% ל	2.70% ד
0.27% ץ	4.81% מ	8.52% ה
1.36% ק	3.44% ם	10.80% ו
5.69% ר	3.33% נ	0.76% ז
4.86% ש	1.27% ן	2.31% ח
5.28% ת	0.64% ס	0.53% ט

Next, the correlation coefficients are calculated and normalized according to Eqs. 5a and 5b

$$y_0 = full\_corr \times 99 + 1 \tag{5a}$$

$$y_0 = partial\_corr \times 99 + 1 \tag{5b}$$

Letter frequency: Very rare → Very frequent

Correlation: Very small → Very large

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	2	2	1	1	1	2	3
1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2	3	3	2	2	2	3	4
2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	2	2	3	3	4	3	3	4	4	5
3	3	3	3	4	3	3	3	3	4	3	3	3	4	4	4	4	4	5	5	4	4	5	5	5

Two-thirds-Correlation: Very small → Very large

Figure 7. A 3-D table of rules as used by the authors indicating which MG is associated with each combination of frequency of appearance, full correlation and correlation of external two thirds.

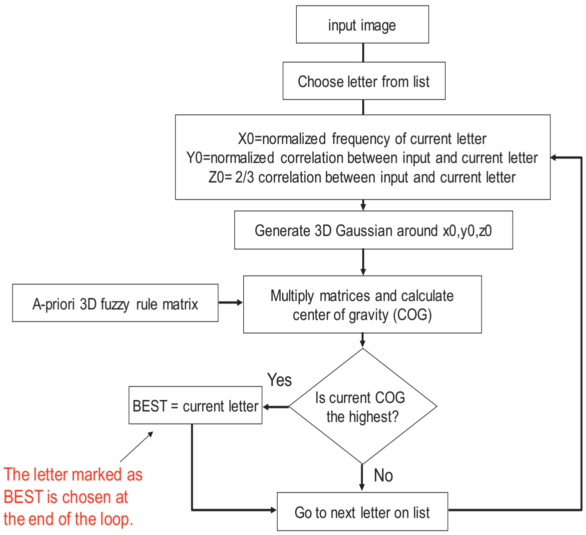

Figure 8. Flow chart of the proposed algorithm for letter recognition.

Then, a 3-D Gaussian of size of 100 X 100 X 100 pixels is generated with the center at (x0, y0, z0). The a-priori generated rule matrix of 5 X 5 X 5 pixels is converted to a 100 X 100 X 100 matrix (simply by stretching each original cell by a factor of 20) and multiplied by the shifted Gaussian. Then, the COG is calculated giving a grade for this choice. The procedure repeats for the 2nd letter in the databank, if it yields a larger COG than before it is being marked as the best match so far, otherwise the previous letter remains the best and the algorithm continues until all the databank is scanned and the letters are ordered according to their COG value. At the in final stage the same procedure is performed for the preceding letter and following letter in the word (if the specific letter is not the initial or final letter in the word) and a COG is determined for the combination (the minimal of the 3 COGs). Then the top 3 combinations are matched against a table of 3-letter-series frequency of appearances and the one most common is chosen (e.g., if 'ARE', 'AFE' and 'ARF' have the largest combined COGs and 'ARE' is a more frequent combination, then it is chosen).

A 2-D example of finding the COG is given in Fig. 9. In this example, a Gaussian centered at 0.35 on the x-axis (normalized frequency of appearance) and 0.7 on the y-axis (correlation) is multiplied by the values of the rule table, where the final grade is given according to the COG of the obtained function. As can be seen, and commonly known in fuzzy logic, the Gaussian covers all of the values in the rule matrix in such a way that every rule contributes something to the final grade and not just the rule containing the center of the Gaussian.

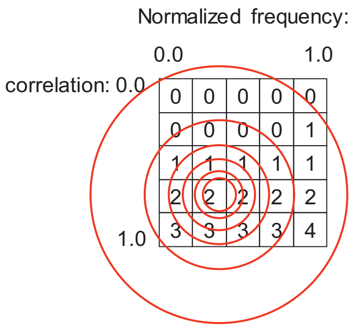

Figure 9. 2-D demonstration of grading a letter in the database using fuzzy logic and a Gaussian MF.

In Fig. 10 one can see an actual text sheet written in Rashi font. The right manuscript is the original and on the left one may see a zoomed version with the details of that sheet. In Fig. 11 we demonstrate the results obtained by the algorithm on two words. In the first word the problematic letter is the 3rd letter (the central one) and the algorithm succeeds in finding the match in such a way that the word has a meaning, in this case the letter is recognized as Lamed 'ל' and the word as HAPLAA (meaning in Hebrew as the name of one of the "Halakha" books written by Rabbi Moses Maimonides- the Rambam). In the second word both the 1st and 2nd letters from the right are unclear but the algorithm resolves them correctly as Alef 'א' and Kaf 'כ' revealing the word KAAMUR (meaning in Hebrew: 'as was said before').

In order to test the strength of the method with comparison to standard OCR software we used a data base of 20 distorted contemporary Hebrew letters, since standard OCR does not recognize Rashi font. In 6 out of 20 cases the suggested algorithm performed better than the OCR algorithm, in one case both algorithms succeeded to indentify the character and in all the rest the neither method recognized the correct letter. Thus the suggested method can be used as a complementary algorithm to identify those letters not recognized by standard optical character recognition.

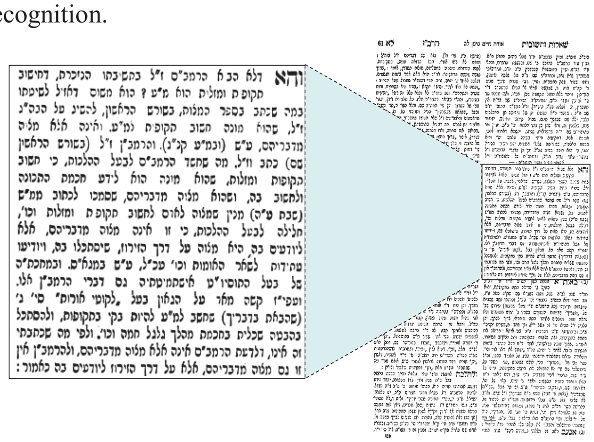

Figure 10. One of several data sheets inspected by the algorithm.

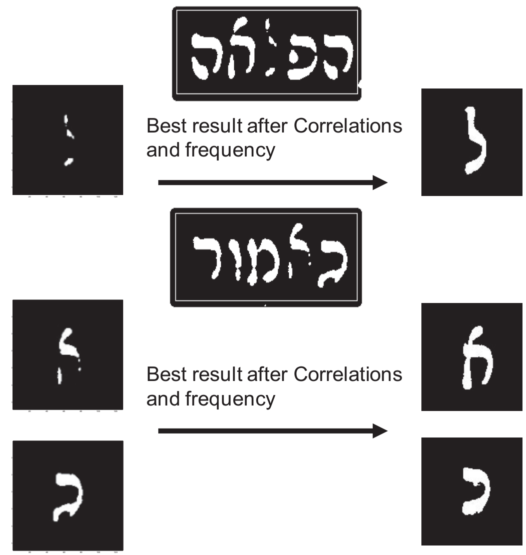

Figure 11. Two examples for using the suggested algorithm on Rashi font.

IV. CONCLUSIONS

In this paper we have demonstrated a new algorithm, based on fuzzy logic for retrieving letters and symbols. The algorithm is especially powerful when using Rashi fonts that standard OCR algorithms are not suited to deal with. However, it can also retrieve letters and symbols from other fonts that the standard OCR algorithms fail to retrieve.

REFERENCES

1. Essoukri N., Amara B., Bouslama F., "Classification of Arabic script using multiple sources of information: State of the art and perspectives," Int. Journal on Document Anal. and Recognition (IJDAR) Vol. 5, pp. 195-212, 2003.
2. Al-Badr B., Haralick R. M., "A segmentation-free approach to text recognition with application to Arabic text," Int. Journal on Document Anal. and Recognition (IJDAR) Vol. 1, pp. 147-166, 1998.
3. Choisy C. and Belaid A., "Cross-learning in analytic word recognition without segmentation," Int. Journal on Document Anal. and Recognition (IJDAR) Vol. 4, pp. 281-289, 2002.
4. Steinherz T., Rivlin E. and Intrator N., "Offline cursive script word recognition - a survey," Int. Journal on Document Anal. and Recognition (IJDAR) Vol. 2, pp. 90-110, 1999.
5. Doermann D. and Jaeger S., Ed. Arabic and Chinese Handwriting Recognition, Lecture Notes in Computer Science 4768/2008, 2008.
6. Brown, E.W., Applying Neural Networks to Character Recognition, Available at: http://www.ccs.neu.edu/home/feneric/charrecnn.html, 1993.
7. Seong-Whan Lee, "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster

neural network," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 6, pp. 648-652, 1996.

8. Senior, A.W. and Robinson, A.J., "An off-line cursive handwriting recognition system," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, pp. 309-321, 1998.
9. Pal A. and Singh D., "Handwritten English Character Recognition Using Neural Network," International Journal of Computer Science & Communication, Vol. 1, No. 2, pp. 141-144, 2010.
10. Vapnik V., “The Nature of Statistical Learning Theory”, Springer, ISBN 0387987800, 1995.
11. Zadeh L. A., “Fuzzy sets,” Information and Control, Vol. 8, pp. 338-353, 1965.
12. Zadeh L. A., “Fuzzy Algorithms,” Information and Control, Vol. 12, pp. 94-102, 1968.
13. Schneider M., Kandel A., Langholz G. And Chew G., Fuzzy expert system tools, Chap. 9, pp. 133-135 and Chap. 10, pp. 159-176, John Wiley and sons, 1996.
14. Gowan W. A., “Optical Character Recognition Using Fuzzy Logic,” Microprocessors and Microsystems, Vol. 19, pp. 423-434, 1995.
15. Hanmandlu M., Murali Mohanb K. R., Chakrabortyc S., Goyald S., and Choudhurye D. R., "Unconstrained handwritten character recognition based on fuzzy logic," Pattern Recognition, Vol. 36, No. 3, pp. 603-623, 2003.
16. Zheru C., Jing W., and Hong Y., "Handwritten numeral recognition using self-organizing maps and fuzzy rules, Pattern Recognition, Vol. 28, No. 1, pp. 59-66, 1995.
17. Miller C., “Rashi's Method of Biblical Commentary,” chabad.org
18. Shulman Y. D., “Rashi: The Story of rabbi Shlomo Yitzchaki,” C.I.S. Publishers, ISBN 1-56062-215-6, 1993.
19. Liber M. “Rashi: Translated from French by Adele Szold,” Jewish Publication Society of America, ISBN 0976654652, 1905. http://www.gutenberg.org/etext/3165.
20. Wiesel E. “Rashi: Translated from French by Catherine Temerson,” Schocken Books, ISBN 0805242546, 2009.