Optimization of Random Sampling for Character Recognition Using Larges Binaries Strings

H. Jiménez-Hernández, J. Figueroa-Nazuno

Centro de Investigación en Computación,
Instituto Politécnico Nacional
México DF, Col Lindavista, C.P. 07738.
hugo@correo.cic.ipn.mx, jfn@cic.ipn.mx

Abstract. Extract the most significant information for discriminate a phenomenon is a problem without general solution. In pattern recognition, the search of a model to represent the information and methods for coding in those models helps us to get different way and manipulate a phenomenon. One model for represent the information is using binary strings and one method for coding the information is a random sampling; together allows propose a method for solve character recognition problem. Therefore, we must warrant that the probabilistic random sampling is significant and search an adequate sampling over the information is the principal task for construct one method for character recognition. This work presents a method to optimize a random sampling, and shows experimentally how it has influence in the efficiency of recognition, with a set of different characters.

1 Introduction

We need a model to represent the information and set of valid operators, which depend of the way the information is coded, this establishes the possibilities to analyze the phenomenon. In the subject of pattern recognition, the method for coding and distinguish the most representative information of some phenomenon, does not have a general solution. Using distinct representation model helps us to characterize different forms of a phenomenon. One model to characterize is using non-arithmetical binary representation and a set of operations are the boolean operators; both, allows us the possibility of establish concepts like similarity and pattern.

A particular case of pattern recognition is the character recognition, where using binary representation and a method to coding, offers a way to identify the different letters. One method to get the relevant information is using a random sampling. Good random sampling warrant gets the most representative information. Use random sampling method has the advantage do not use the totally of information, only a minimum part.

Using a random sampling to extract extract the information, get us relevant information, always it has the same probability in the phenomena [16]. The problem of character recognition will be seen as select a set of dots, such that, with these dots we can identify and classify the different letters. The random sampling has the subject of the extract dots in the images of the different letters. A first approximation, these dots are selected using random position with a uniform distribution; however, the distribution of the dots in the different letters does no have this distribution; so on, for select the best dots we need to define some criterions which warrant use the best dots for accomplish this task. In this work present one method for optimize the random sampling and it will be more significant for coding each letter in binary strings, and propose basic character recognition.

2 Sampling image

One mechanism for extract information is the sampling probabilistic method. If the random sampling is significance to population, it pick up of compact way the information of each character and is not necessary use the total information from the symbols images. The random sampling consist in search one list of positions dots such identify adequate each alphabet letter.

The representative dots are stored in one list L of dots, from L build the binary strings. These strings represent the different variants of letters over similarity function that will discus.

3 Representation

We consider images with binary color representation, if the image has more than two colors, we define a transformation function, where usually is defining as follow:

$$f : \mathrm{N} \rightarrow \{'0','1'\} \text{ where the most usual is define an interval such } f \text{ will be defined as follows : if } x, y \in \mathrm{N} \Rightarrow \mathrm{f(x)} = \begin{cases} x < y & '0' \\ x \geq y & '1' \end{cases} \quad (1)$$

The binary string is the result of the concatenation of binary symbols of the set {'0','1'}, each representation of the sampling set will be re writing as follows:

$$s_1 \cdot s_2 \cdot \ldots \cdot s_n \quad (2)$$

such that each $s_i \in \{'0','1'\}$, where n is the number of sampling dots.

4 Coding

Each image of letter represented as a matrix M of size $s \times t$ such that for all element $M_{i,j} \in \mathrm{N}$ (where N represents the natural number set), the value *j, i th* mean the color of the dot *j, i,* it correspond a natural number which represent the color intensity.

Let n be the number of dots that need for sampling from the matrix M, then it will be necessary to generate $2n$ random dots, to make the n positions for sampling over M, it has the form (i, j) between the interval $(0,0) - (s, t)$, these dots are storage in the list L over an order $<$ in agreement are generated. By notation $L^{<}_{k}$ represent the k-th element of L, where the first element is $L<1$ and by consequence the last will be $L<n$.

For each tuple (i, j) contained in $L^{<}$, we get of its value in the matrix M, and we apply the transformation function, concatenating each element, generating the binary string as follow.

$$f(M_{L_1}) \cdot f(M_{L_2}) \cdot \ldots \cdot f(M_{L_n}) \tag{3}$$

where n is the number of dots cotains in L.

From this set of dots, we can construct a set of binary strings, which represent each letter from the alphabet. If the letters used for construct the binary strings are representative from the set of variants of letters, then these strings could use for identify variant of these letters.

5 The space $\{0, 1\}^n$

The representation of the information has made taking storage structures and transformation mechanism of information. The data has correlation, compared with the traditional model. The first idea such has this model as follows:

1. *The representation is uniform.* All things are shown by dots over the same space.
2. *The meaning is internal as follow.* The nearest dots by the representation over the space have the same meaning, in other words, the semantics do not separate from the syntax.

A vector in the space is one mathematic model particularly, which could be satisfying, two conditions: the space N has high dimensionality, and is enough for coding the problem [16]. The high dimensionality is more important than the natural dimension for make models with binary dimension. For example, the capacity of the information, which is, content in one binary vector, could correspond to one page of text, and all operations over a Turing machine could be done has a one page of Turing.

A computer word is a register of a database in traditional computation, and it will be divided in fields, which represent at the same time the part, which we want to represent,

making a high conceptual level. The facts are modeled using patterns with N dimensionality; each attribute could be coded as binary strings, which represent facts too.

5.1 Space $\{0, 1\}^n$ concepts

Let n be the number of dimensions of the space. Then number of possible dots will be $N=2^n$ (13). The dots in N are representing by n *th* tuples of zeros and ones, and will be rewritten as integer numbers of n bits in binary representation. We establish some concepts:

Definition. The norm of some dot expressed as x, is denoted as $|x|$, and is defined as the number of ones contained in the dot x. Formally its expressed as follow:

$$|\ |: \{0,1\}^n \to N \text{ such that } |x| = \sum_{i=1}^{n} x_i \text{ if } x_i = 1 \tag{4}$$

Definition. The difference between two dots x and y, is denoted like $x - y$, the difference is another dot such that has ones where x and y are different and zeros in other case. The difference is commutative, formally we have:

$$-: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \text{ such that } -(x,y) = \forall_{i \in \{1\ldots n\}} \begin{cases} 0 \text{ if } x_i = y_i \\ 1 \text{ in another way} \end{cases} \tag{5}$$

Definition. The distance or Hamming distance between two dots x and y are denoted like $d(x,y)$, and is the number of components which x and y differ. The distance and the norm are scalar. Formally, we have:

$$d: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \text{ such that } \quad d(x,y) = |x - y| \tag{6}$$

Definition. A circle with radio r and center x, is denoted by $O(r,x)$, and represent the set of dots which has to much a distance of r from x, formally is expressed as follows:

$$O: \mathrm{N} \times \{0,1\}^n \to 2^{\{0,1\}^n} \text{ such that} O(r,x) = \{y \in \mathrm{N} / d(x,y) <= r\} \tag{7}$$

5.2 Pattern and Similarity definition

The definition of *pattern* is restricted to the model used to represent the information, and to formulate a definition is by the construct a similarity function. If one element has more significance to set, then the element represent a set and it is a pattern of this set. In binary coding system, the pattern definition has taken as follow:

Definition. A pattern is a binary string $x \in \{0,1\}^n$, of length n, such that a set $C \subset \{0,1\}^n$, and over similarity function $F: \{0,1\}^n \times \{0,1\}^n \to N$, x is a representative string to set C over a given δ.

The similarity between elements of $\{0, 1\}^n$ has taken by the Hamming distance.

Definition. Be the similarity function $F: \{0,1\}^n \times \{0,1\}^n \rightarrow N$, *is defined as* $F(x, y)=|x-y|$, *where* $|x|$ *is the norm and* $x-y$ *is the difference, we will say* x *is like* y *iff* $F(x, y) < \delta$, *where* δ *is the nearest radius of circle* $O(x, \delta)$ *where* $\delta << n$.

Then we have that $\forall_{y \in C} . F(x, y) \leq \delta$ for all elements in C.

Given two image M and M', which represent the same letter with few differences, calculate the binary strings for M and M' and the associated dots c and c' respectively, must have a very small hamming distance in $\{0,1\}^n$. Near distances are considered similar, far distances denoted distinct elements.

5.3 Minimum distance criterion

Let P denote the set of representative patterns, which represent the binaries strings that represent each letter. From this basic set, (using the similarity metrics described in the last sub chapter), we can decide when an image, that keeps a letter is similar to any pattern element in $p_i \in P$. The minimum criterion consists in provide an image M that keeps a letter, build the binary string m, and calculate the Hamming distance with all elements in P.

The element, which has the minimum distance to m in P, has the most similar element. Therefore, we can say m is similar to p_i or in other words, m is a possible variant of the pattern that is representing by p_i, and it corresponds to the letter *i-th.* Formally, we say:

$$Mostsimilar = \text{Min } (\{x / m \in \{0,1\}^n : \forall_{p_i \in P} d(p_i, m)\}) \quad (8)$$

(Note: For proposal of this paper, we use this criterion, but we can define other methods, by the combination of different criterions)

6. Sampling Optimization

To use different methods for search a string pattern or average string from a set C, generate different string patterns, which has few differences [15], these string will be considerate to select the best set of dots for sampling in the images. The difficulty is the number of letters for recognize, in a complete alphabet there are 26 letters upper case and 26 lower case, for our propose we only use a set of letter variants in lower case.

6.1 Distributions dots analysis

The random sampling has done over the images of the letters; consist in build a list L, of length n, where n represent the number of random dots for sampling over the images.

However, to use random dots not always guarantees searching the optimum dots for gets a good sampling over the images.

Some times we may select dots, or the sampled areas over the image do not get us the enough information for characterize it; to define a method for optimize the random sampling warrant we occupy the dots which have more significance for the character recognition. One first approximation to optimize consists in searching the correct distribution of dots, only analyze the relevant areas for discriminate the letter; discard squares and some areas which never use the letter. In the figure 1, we can see the pattern p_a, which is result of the set C_a, analyze patterns help us to manage only few part of the information.

The average pattern is calculated by some methods, (like Majority Rule, Majority Rule Modified or Random Reads [13]), for the set of alphabets $\{\Sigma_1,\ldots, \Sigma_n\}$, generating the sets $C_a,\ldots, C_z$. These represent the set of possible letters variant of each letter; from each set C_i, the representative pattern is denoting by p_i. The set of $P= \{p_a,\ldots,p_z\}$ is the set of pattern distinctive of each letter.

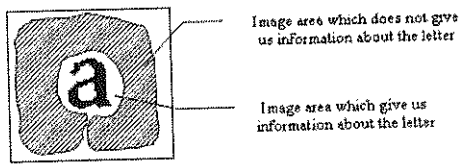

Fig. 1. Show us the areas where could get relevant information over the letter

If each pattern p_i represents a particular letter, the dots with ones in p_i, are the significant dots for *i-th* letter. Then when we make the superposition over (using the "*or*" operator) two letter, we get the dots used by both letters, there are some dots which are common; these dots are not significant because they do not get information which use for discriminate a particular letter. The common dots are locating by logical operator *and* between the images, the difference between the superposition and the common dots is the information non-duplicate in both letters. The figure 2 shows an example with the patterns p_a and p_b and the result c) is an image which contains the difference between both images ignoring these dots which are common in two images. Dot analysis must take this in consideration to conform the set of dots for sampling over the images.

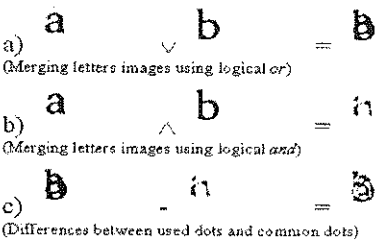

Fig. 2. Operations for a) join b) search common dots and c) make the difference between two images

Generalizing for all letters of alphabet (26 different letters in lower case), we can express as follow:

Be $p_a, p_b, \ldots, p_z$ the patterns calculates from the set $C_a, C_b, \ldots, C_z$, which contains the letter variants of *i-th* letter, the used dots are denoted by the follow expression:

$$p_a \vee p_b \vee \ldots \vee p_z \tag{9}$$

Where the result must be a dot (graphically as image), which contains ones where are located dots by almost one letter.

For locate the set of common dots we must denote as follow:

$$p_a \wedge p_b \wedge \ldots \wedge p_z \tag{10}$$

the set of dots which have only the dots used by the pattern p_i's without considering duplicates dots, are defined as the difference between Eq. (9) and Eq. (10). This is expressed as:

$$p' = (p_a \vee p_b \vee \ldots \vee p_z) - (p_a \wedge p_b \wedge \ldots \wedge p_z) \tag{11}$$

then p' has the significant dots for define L; we may consider to p' as a meta pattern, over the set of patterns p_i's of each letter. This pattern shows the information about the dots, which are significant, and are use without duplicate for discriminate each letter and each variant. Now we take as parameter the number of dots, which we wish to use for the sampling denoted by n and represent the work dimensions in $\{0, 1\}^n$ in p'. Let n be the number of dots for sampling over the matrix M which represent the image keept in p', we need to generate $2n$ random dots, for built the *n-th* position for sampling over M, which has the form (i, j) over the interval $(0, 0) - (p, q)$. L denotes the set of dots over an order $<$ accord-

ing these be generated. By notation $L^{<}_{k}$ represent the *k-th* element of the list L, where the first element is $L^{<}_{1}$ and the last be $L^{<}_{n}$.

Using L we can generate a binary string which represent a dot l in the space $\{0, 1\}^{n}$. The dot l must have a norm like $l \approx n/2$ to guarantee the dots be distributed best over the dot used and no used. The pseudo cede will be express as follow:

1. Be l_i such that $l_i \in L$ random taking which $f(M_{li})$ be 0 if $|cl|<n/2$ or 1 if $|cl|>n/2$.
2. Generate random dots has the form (i,j) over the interval $(0,0) - (m,n)$.
3. Calculate p'.
4. if $|p'|\approx n/2$ finish
5. if $|p'|< n/2$ search two values i, j non duplicates which $f(M_{(i,j)})=1$ and go to 3.
6. search two values i, j non duplicates which $f(M_{(i,j)})=0$ and go to 3.

6.2 Superposition Dots Analysis

Sampling over the image p', help us to distribute efficiently the dots to select a good sampling, balancing in p' the dots for sampling. However, we need make an exhaustive analysis, because there are situations where a high number of letters uses a dot and these dots are no significant for choice for the sampling. They can generate sub patterns of the others or the majority part of one pattern be overlapping. As a direct consequence the patterns has no significance and we won't characterize and define a nearest radius $O(p_i, r)$ of each letters variant making false recognitions for set of given dots.

The superposition problem has no exclusive for a select cases, there are big number of dots of the p' image that exhibits a high level of superposition (each component represent a dot of an image, in particular the image which contains the join of all p's with n defined as $n = p \times q$, coding the image as a binary string). The Graph 1 shows the superposition frequency to join the p_i's patterns, see the existence of high superposition by the areas that has almost 16 letters that use, then these areas contains dots that are less adequate for the information extraction than the dots which has few level superposition frequency.

The figure 3 is built from the natural sum of all p_i's dots, i. e. the frequencies of each particular dot from each image p_i, as a level curves. Then dots which are used by many patterns p_i, tend to maximum frequency (26 because is the totally of letters in the lower case alphabet), on the other hand, dots which are use by few number of letter trend to 0. A good sampling does not select dots with high frequency levels and low frequency levels; so we want for each dot in L does not have these frequencies, then we define a superposition matrix Mp (graph 1 and graph 2 will be generate from Mp), which keep the sum of all patterns images p_i.

The verify of L, is done by analyze each element in L such that each dot in Mp, do not correspond to prohibit areas, in other word, if $Mp(i,j) < Umbral$ where $(i, j) \in L$ and *Umbral* is a integer value such that define the significance of the dots.

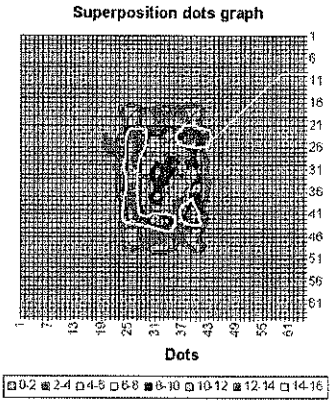

Fig. 3. Level curves of different levels of superposition of dot from the images of patterns p_i

The value of *umbral* set as 13, which is the maximum number of elements that will be in superposition (26 divided by 2). We guarantee that the maximum number of letters, which a particular dot participate be 13, more over if we consider a letters with high similarity, this threshold should take few values. The lower limit must be defined, because if exists areas which the dots has a frequency near to zero, they do not offer enough information, making an opposite analysis, using zeros, and represent areas which has a high level of superposition of zeros. The minimum threshold ideally need be 13, but the area used by the letters do not be exactly ½. We consider a near limit to area use by the superposition of all letters, it will be ¼, which is the area use by the ones compared with the total image area; an approximation to lower limit is defined like ¼ of maximum frequency, of elements that be in superposition. So all elements in L will be in the interval, in other case we need select only dots which has these conditions. The algorithm is express as follow:

1. While $\forall l \in L \wedge Mp_l \leq 13$ go to 2 else finish.
2. Search $l \in L$ such that $\max(Mp_l)$
3. Generate a couple of random number *(i, j)*
4. if $(Mp_{i,j} \leq Mp_l \vee Mp_{i,j} \leq MaxFrequency) \wedge Mp_{i,j} \neq 0$ make a substitution in L, for the element l, by (i, j) and continue, else go to 3.
5. End While

When we finish the algorithm execution, L has not elements with high or lower level of superposition, moreover we do not change the balance with the conditions present in the step 4, do not use dots that $Mp_{i,j} = 0$, which represent zeros in the image p'.

To conclude we need to calculate the dots p_i's of length n, using the dots in L, where each element will be the patterns p_i's of length n, that identify each set of C_i, which has the possible letter variants.

7 Experimental Model and Results

To probe the algorithm we make an experimental model for evaluate the efficiency taken a set of distinct alphabets denoted by $\{\sum_1,\ldots,\sum_m\}$, where $m = 80$, each alphabet has the 26 different letters in lower case, and the dimensions of the images are 64 by 64 pixels in monochromatic mode.

Two lists were generating L_1 y L_2, (L_1 is not optimized and L_2 is optimized), the length for each list was taken from the set {200, 150, 100, 50}. For each alphabet we built the set $C_a,\ldots,C_z$, computing each p_i, which is calculated by two methods: Majority Rule Modified and Random Reads. We consider 160 elements over each C_i for the method of random reads. By each list and each method we build the p_i's, and show the average distance and the standard deviation. The obtained statistical result are shown in table 1, they denoted an increment in the average distance and standard deviation before to a) and after to b) applying the optimization method.

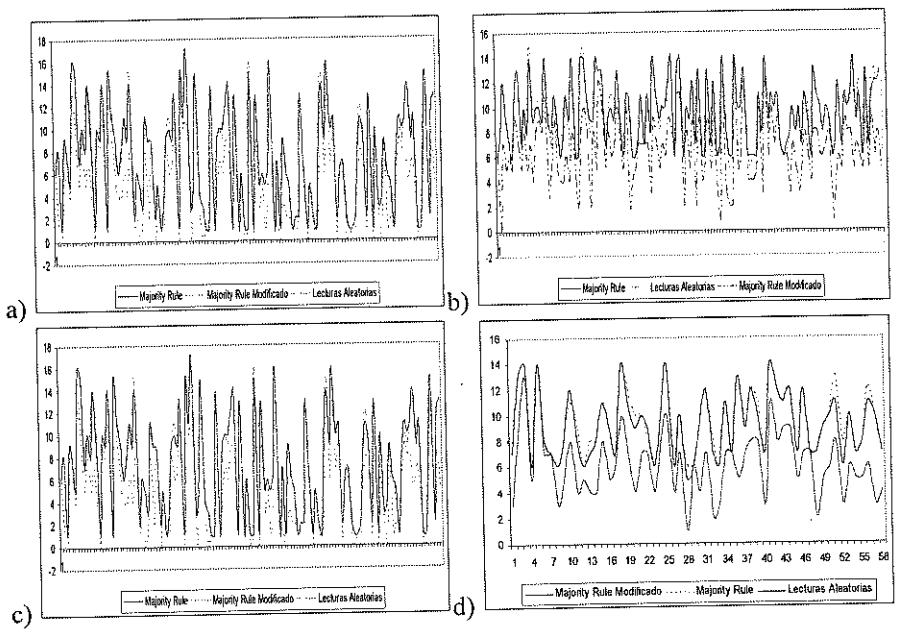

Fig. 4. . Frequency of superposition graph for a) L1 with n=200 b) L2 with n=200 c) L1 with n=100 y d) L2 with n=100

The figure 4, show the frequency superposition for L_2, shown a distribution over the dos in the defined interval, and L_1 do not exhibits this behavior. The result only exhibits for the selected case by $n= 200$ *and* $n=100$. In figure 1, we present the dots distributions.

Table 1. Statistical Measures of the distance and standard deviation of a) L_1 b) L_2

a)

	Mean		Standard Deviation		
n^1	RR[2]	MRM[3]	*n*	RR	MRM
200	17.275	17.239	**200**	9.650	9.661
150	8.822	8.796	**150**	5.361	5.339
100	7.611	7.602	**100**	4.312	4.250
50	4.060	4.049	**50**	2.371	2.403

b)

n	RR	MRM	*n*	RR	MRM
200	20.251	20.209	**200**	11.142	11.240
150	13.349	13.304	**150**	7.124	7.162
100	8.830	8.802	**100**	5.023	4.926
50	4.455	4.441	**50**	2.708	2.707

1 String Length
2 Random Reads
3 Majority Rule Modified

a) b)

Fig. 5. Distributions of dots for a) L_1 with n=200 b) L_2 with n=200

Table 2. Efficiency table for identify if a generated binary string is similar to any p_i using the criterion of minimum hamming distance

	L1		L2	
n^1	MRM[2]	RR[3]	MRM[2]	RR[3]
200	81.60	81.10	89.65	88.62
150	78.60	79.10	87.65	86.70
100	67.90	67.15	83.25	81.00
50	56.95	58.65	78.85	75.40

1 String Length
2 Random Reads
3 Majority Rule Modified

The tables show when we have a uniform distribution we can increases the identify of letters using the criterion of the minimum hamming distance for the similarity of any pattern p_i

8 Conclusions

The use of random methods gets us an alternative to characterize and code some ph nomena.

These results show experimentally that representing and coding the information by no arithmetical binary strings is a good model for retrieve the information, in the charac recognition. In the table 2 we can see how the efficiency of character recognition increas considerably with the random optimization sampling.

The coding do not use all information, only use a few part making easier to handle t information and the model offers a simple algorithm with low complexity and easy i plementation.

This is a non-classical model of computation, but we can do emulation in classic model, which give us an approximation of the model and offers a different way to ma character recognition.

The method provides a way to optimize a random sampling making an analysis of s perposed dots, which can get us information over a explicit letter.

By experimental process, we showed how the sampling encodes the representative i formation of each letter variant, building a list of dots, which identify the best informati over the letters. The optimization over the sampling provides a best criterion to sel when one binary string is similar to some pattern p_i.

The information provided by the random sampling could be used to build of basic cha acter recognition.

References

1. Stern, A., Matrix Logic, North Holland, Elsevier Scientific Publishers, 1988.
2. Chaitin, G. J., Information, Randomness & Incompleteness: Papers on Algorithmic Inform tion Theory, Second Edition, IBM, P O Box 704, Yorktown Heights, NY 10598.
3. Díaz de León, J. L., Cornelio, Y., Modelo SDM, colección de reportes técnicos acerca estado del arte de Memorias Asociativas, No. 52 Serie: VERDE Fecha: Junio 2001.
4. Dz Mou J., George N., N Tuple Feature for OCR revisited, IEEE Transactions on patte analysis and machine intelligence, vol. 18 no 7, July 1996.
5. González R. C.,Woods R. E, Digital Image Processing, Adison Wesley Publishing Compa Reprinted September 1995.
6. Fonseca, F., Rubio, J., Figueroa J. Análisis Experimental de la Superposición de Informaci en Espacios de Memoria Aleatoria. A.T. IEEE ROC&C 2001, 12ª. Reunión de Otoño de C municaciones, Computación, Electrónica y Exposición Industrial. Octubre 2001
7. Espinosa, A., Villanueva, E., Figueroa, J., Análisis Cuantitativo y Experimental de Superpo ción de Información en Memorias Aleatorias. MICAI/TAINA/TIARP 2000 Acapulco, Méxi 11-14 Abril, 2000.
8. Jiménez H., Figueroa J., Reconocimiento de caracteres por cadenas binarias aleatorias, mem rias ANIEI 2003.

9. José Ruiz Shulcloper, Formación Integral del especialista en Reconocimiento de Patrones, CIC-IPN, Reconocimiento de patrones avances y perspectivas, Edit Boad, 2001.
10. Castleman, K., Digital Image Processing, New Yersey, edit. Prentice Hall 1996.
11. Motwani, E., Raghavay, P. Randomized Algorithms, Cambridge University Press(1995)
12. Kanerva, P., Encoding Structures in Boolean Space, RWCP, Theorical Foundations SICS.
13. Kanerva, P., Sparse Distributed Memory A Bradford Book, The MIT Press, 1988, Cambridge Massachusetts.
14. Ash, R., Information Theory, Published in Canada by General Publishing Company, 1990.
15. Walpole, Myers, Myers, Probabilidad y estadística para ingeniería, Personal Education 1998.
16. Useaka, Y., Kanerva, P., Asoh, H., Foundations of real-world intelligence, CSL publications, 2001.