Chaos-based Associative Retrieval and Identification of Audio Information

A.M. Didovyk and K.V. Zakharchenko

Institute of Radio Engineering and Electronics of the Russian Academy of Science
Mokhovaya st. 11, Moscow, 103907 Russia. tel. +7 (095) 202-7830
E-mail: chaos@mail.cplire.ru

In this work the approach to recording, storing and associative retrieval of multimedia information is proposed. For this purpose the method for recording and storing the information on the trajectories of the nonlinear dynamic systems is used. Principles of recording, allowing to retrieve the desired record from the database of audio records by an arbitrary piece of the record we are looking for, are proposed. Methods for optimization of the proposed principles by the CPU time and amount of memory used are developed. Described principles are implemented in Matlab software environment. The results of investigation of the proposed algorithms are given and analyzed.

Key words: nonlinear dynamical systems, nonlinear discrete dynamics, maps construction, audio data, redundancy elimination, trajectories of dynamic systems, attractor, data storing and retrieval, numerical simulation, Internet, applications of chaos.

PACS numbers: 89.20.Ff; 89.20.Hh; 05.45.-a; 05.45.Pq; 47.52.+j; 95.10.Fh; 07.05.Tp; 01.50.Fr; 07.05.Tp; 05.45.Gg;

1 Introduction

The possibilities of transmission, storage and processing digital information are growing constantly. It is because of rapid development of data storage and communication technologies, e.g. Internet.

A large part of digital information is multimedia information (MMI) which includes text, graphical, video and audio kinds of information. Effective use of MMI demands appropriate algorithms for processing large amounts of data.

Unlike texts and discrete data, graphical and video information cannot be reproduced by a human without special technical means. That is why methods, designed for text or discrete information processing, cannot be used effectively for major kinds of MMI processing, including storage and retrieval.

As a result at present associative retrieval of information is attracting scientists' attention. According to explanatory dictionary association is the bond, which appears in certain conditions between two or more mental formations (feelings, motive acts, perceptions, ideas, etc.). Associations are distinguished by contiguity (in space or time), similarity and contrast. The term was introduced by J. Lokk (1698). By the term "the associative retrieval of information" retrieval of sought information using semantic bonds, not exact coincidence of the retrieval request with the results of the retrieval, i.e. the retrieval of information similar in some specific way to information containing in the request.

Here we will briefly review some results obtained in the realm of associative information retrieval systems.

In [1] the system for the associative retrieval of images called "Blobworld" is presented. This system divides images into specific regions called "blobs", which are mostly coincident with the objects on the image. Each image's partitioning is represented as a composition of several regions, which are approximately homogenous with respect to texture and colour. A region is described by the colour distribution and texture. Algorithms of segmentation are completely automatic and do not require any tuning of the parameters or any manual selection of the regions. To carry out retrieval one provides an image, the system divides it into the regions and after it one should point at regions he or she wants to use as information for carrying out the retrieval. As a consequence images containing regions similar to ones pointed at earlier will be found.

QBIC system [2] (Query By Image Content): the system for storage of and retrieval in large arrays of images. It exploits images' contents to make up retrieval requests. This system affords an opportunity to make up retrieval requests based on outline, shape, texture, colour distribution and image elements arrangement. The system carries out retrieval as an approximate correspondence to provided criteria. The set of parameters characterizing provided criteria is represented by multy-dimensional vector in non-Euclidian parameters space. Thus similar images are characterized by proximity of corresponding parameter vectors in the space. A metric of the described space which considers peculiarities of the parameters is proposed. It is shown that its general form is quadratic form. It is asserted that images which do not correspond to provided retrieval criteria may appear on the "output" of the system, nevertheless all sought images will be present on the "output".

The one is afforded an opportunity to make up two kinds of requests:

1. "direct request", when in "input" of the system desired combination of colour, shape and texture is provided directly, i.e. by choosing colour from a palette, drawing outline by hand etc.

2. "request by example", when a set of parameters is taken from an image provided by a user of the system.

In the work general retrieval principles as well as problems of optimization of the system with respect to processor time used and number of disk operations made are considered.

"Forget-me-not" system [3]: the associative retrieval system for storage, recording and retrieval of the text information. Software complex "Forget-me-not" is designed for processing (recording, retrieval and visualization) unstructured facsimile copies of documents and text documents in 32-bit Windows family operating systems. While dealing with facsimile copies optical character recognition must be done beforehand i.e. the text of the document which will be used for retrieval must be created. The aim of the system is to compile fascimile archives of unstructured documents and to provide easy and convenient access to them, supplied with an ability of searching documents using natural language requests (including specific position within the document). In contrast to majority of search systems which use keyword requests (keywords with logical relations), here the search system is oriented to use rather large document fragments or even the whole document as a request. Therefore, the result of the search is not documents containing the request word (perhaps, in all its forms), but the documents that are most close by content to the document of the request. The search technology is based on the ideas of complex dynamics of nonlinear systems. Here the information image is unambiguously related to a periodic motion of the dynamic system which is the information "storehouse". Incoming text documents are converted by means of the Box Manager, which is a part of the "Forget-Me-Not" information system, into a dynamic archive – storage of the text information images. During analyzing incoming documents, the system creates an artificial language related to the contents of the stored documents. Combination of this language and the dynamic system provides content-sensitive (associative) search for information.

Besides systems described above, there is a number of other associative search systems (e.g. [4–7]).

In this work approach to recording, storing and associative retrieval of multimedia information is proposed. For this purpose a method for recording and storing information on trajectories of nonlinear dynamical systems is used.

Principles of recording, allowing retrieval of desired record from a database of audio records by an arbitrary piece of the record, are proposed. Methods for optimization of the proposed principles by CPU time and amount of memory used are developed.

Described principles are implemented in Matlab software environment. The results of investigation of the proposed algorithms are given and analyzed.

The structure of the article is following: in a part 2 a method for recording and storing information on trajectories of nonlinear dynamic systems is considered; in a part 3 a problem of preparing data for storing it on the trajectories of dynamical system is considered; in a part 4 principles of audio data recording and retrieval are described; eventually in a part 5 the software complex realized in Matlab environment is presented.

2 Dynamical systems based associative memory

The procedure for storing and retrieval information on the basis of limit cycles in one-dimensional dynamical systems was introduced in [8] and [9] and further developed in [3,10–12]. Here is a brief description of the method. Let

$$a_1 a_2 ... a_n \quad (1)$$

be a given sequence of symbols (called an information block), each element a_i of which belongs to an alphabet composed of N symbols (in our case 256). A one-dimensional map of an interval into itself is designed for this sequence. The map is chosen such that it possesses a stable limit cycle of period n, the iterates of which are in mutual unambiguous correspondence with the elements of the sequence (**??**). In the simplest case, each element of the alphabet is related to its own value of the mapping variable and to an interval of the mapping variable of the length $1/N$ [8], [9].

Similarly, a map with a few stable limit cycles can be formed, whose elements are unambiguously related to the elements of the corresponding stored sequences (fig. 1).

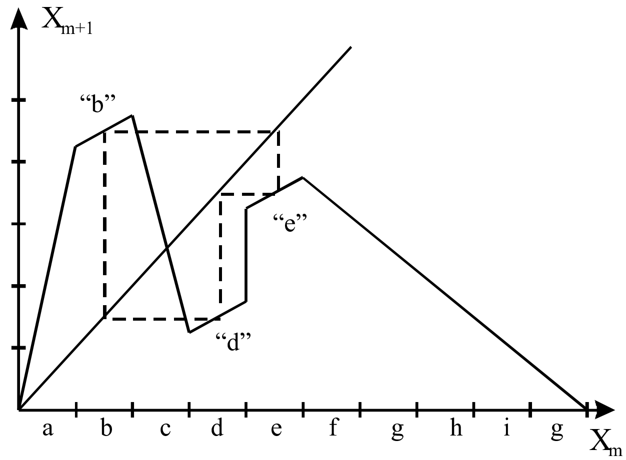

FIG. 1. Word "bed", stored on 1D map

The information storage capacity of this system is quite limited. For example, two strings which contain at most two equal symbols cannot be stored simultaneously. To overcome this limitation the concept of storage level is introduced, and a map is designed in which each value of the mapping variable corresponds not to a single element of the sequence (**??**) but to q consecutive elements [8], [9] (fig. 2). Retrieval of information in this case is performed by setting the initial condition within one of the segments on the unit interval $[0, 1]$ corresponding to a fragment of an information block which is q symbols in length, iterating the map from this initial condition, and by then transforming the sequence of numbers back into a sequence of symbols.

Such a generalization of the map design procedure enables one to store arbitrary symbol sequences in which there are no coinciding fragments containing q or more common elements. Otherwise, storage at this level is impossible. A natural way to overcome this limitation is to use a higher storage level, but this way leads to other restrictions. Actually, as the storage level increases, the length of the information intervals in the map decreases, thus forcing us to proceed from ordinary precision calculations to special high-precision calculations with all the corresponding problems (deceleration of calculations, increased memory requirements, etc.).

The problem of storing arbitrary sequences can be solved by means of a special coding procedure [5,10–12] which we describe briefly below. Let the sequences be stored at a q-th level, and let them contain a few identical fragments of length q. A new generalized element representing this fragment is added to the alphabet and all inclusions of this fragment in the sequences are replaced by the new element of the alphabet. If, upon the next presentation of the sequences (using the newly added element of the alphabet), identical fragments of the length q are once again found, they are replaced by another new element of the alphabet. Alphabet extension is continued until the stored information sequences contain no repeating fragments of the length greater than or equal to q.

This encoding procedure allows arbitrary information blocks to be stored at any level beginning with level 2.

For example let us consider in details the procedure for storing two sequences "*abc...dxyw...*" and "*cde...kxyz...*" made up of symbols from the Latin alphabet. Let the storage level be $q = 2$. Let us take the first fragment of the first information block "*ab*" and look through the sequences. If there is no other inclusion of this fragment, the next fragment "*b*" is taken and the search is repeated. Let a fragment "*xy*" which occurs at i-th step be encountered again in the sequences. Then a new element $\beta =$ "*xy*" is added to the initial alphabet, and every "*xy*" in the sequences is replaced by β. The sequences now look like: "*abc...dβw...*" and "*cde...kβz...*". Then the search process for identical fragments is repeated begining with the fragment "*βw*". If other fragments are found to be repeated in the sequences then a new element is added to the alphabet for each such occurrence, and corresponding substitutions are made. The process finishes when the last fragment of the second sequence is reached. As a result, a new extended alphabet is formed. The original sequences expressed in this alphabet now contain no repeating fragments of the length greater than or equal to q. The number of computer operations necessary to form the new alphabet and to represent (to encode) the original sequences with it, is proportional to the square of amount of stored information, i.e., for an amount L of stored information, the number of operations is $\sim L^2$.

Now let us consider the problem of retrieval of information recorded using described above scheme. The problem of information retrieval using a fragment represented in initial alphabet may be solved in the following way. When the fragment of the sequence is represented using initial alphabet it is necessary to convert it into extended alphabet. Then

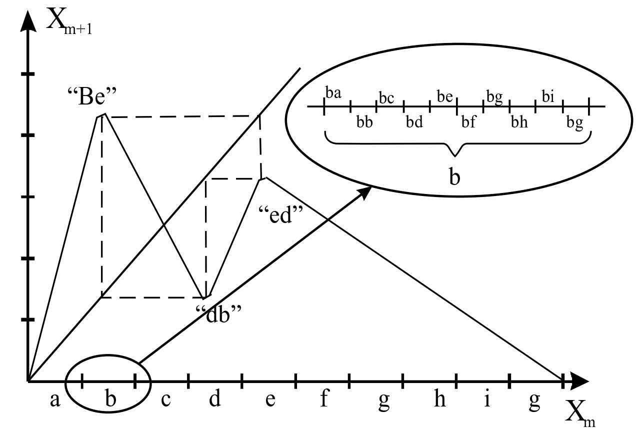

FIG. 2. Word "bed" stored on 1D map, $q = 2$

the new, with respect to initial, elements of the extended alphabet are sequences of initial alphabet. It may seem that to convert a produced fragment it is necessary to compare all its information blocks of the length q with the extended alphabet's elements and to carry out appropriate substitution in the case of their coincidence. However it is not quite so: the typical situation is that the beginning of the produced (sought) fragment, which is represented using initial alphabet, being encoded sufficiently is not coincident with the beginning of the first extended alphabet's encoding element. Therefore the problem of retrieval of entire information sequence by its fragment is non-trivial. It may be solved in the following way. Let

$$b_1 b_2 ... b_k \tag{2}$$

be a fragment of the sequence of the length k being retrieved, which is specified using initial alphabet. For specificity, further it is assumed that recording is done using level 2, i.e. $q = 2$.

Let us consider first two elements of the produced fragment $b_1 b_2$. If the element $c_1 = b_1 b_2$ exists in the initial alphabet, then substitution in the fragment (**??**) is made.

$$c_1 b_3 ... b_k \tag{3}$$

In the presence of another such pairs of elements in (**??**) they are substituted by c_1 again. If there is no element $c_1 = b_1 b_2$ in the extended alphabet, then the pair b_2b_3 is examined repeating the procedure and so forth. In the first case when element $c_1 = b_1b_2$ was found presence of the element $c_2 = c_1b_3$ in the extended alphabet is examined and so on. The result of such procedure is a certain representation of the produced fragment using extended alphabet. Important quality of such representation is that even if the beginning of produced fragment is not coincident with the beginning of one of additional elements of extended alphabet, provided sufficient encoding, (it is a typical case), the first information block of the produced fragment, which corresponds to a full extended alphabet's element, provided sufficient encoding, will be encoded correctly.

In fact, let $b_1...b_l$ is the informayion string which contains only a piece of an element of the extended alphabet provided sufficient encoding. The representation of this string in extended alphabet is equivalent to "garbage" – a sequence of extended alphabet's elements $d_1...d_i$, which is absent in corresponding recorded information block being represented in its extended alphabet. The string $b_{l+1}...b_{l+g}$ is the first corresponding to the extended alphabet's element h_1. The elements h_2h_3 immediately following h_1 are thus correctly encoded elements. However in the end of the produced fragment some elements of the initial alphabet constituting only a part of an element of the extended alphabet may appear being encoded sufficiently. Being encoded they result in "garbage".

It is important to note that once-only fragment encoding is sufficient to realize associative retrieval algorithm. It is due to uniqueness of additional extended alphabet's elements and therefore vanishingly small probability of the incidental appearance of their combinations in the produced fragment.

Representation of the produced fragment using extended alphabet is the first step of the algorithm for information retrieval under described above method of recording. The second step of the algorithm consists in examination of the fragment, represented in extended alphabet, for existence of informative sections corresponding to elements of this representation on the map. For that it is verified if some informative section corresponds to a first pair of elements of the fragment. In the case of positive answer, a verification of yet 4 next elements coincidence is performed. And if all of them are present on the map one after another a conclusion that the sought fragment is stored on the map is drawn.

In a general case the fragment begins with elements $d_1...d_i$ which correspond to "garbage" and do not correspond to any informative segments. Therefore the verification yields a negative answer and the procedure is repeated again starting with d_2, and so forth until finally h_1h_2 sequence will be considered and the coincidence of the next 4 elements will take place.

General properties of dynamical systems as information storage are following:

- high capacity;
- capability of entire information block retrieval by its arbitrary small fragment;
- the number of separate information blocks which may be stored on the dynamical system corresponds to the number of limit cycles of this dynamical system;
- quadratic dependence of recording time on volume of information being recorded;
- logarithmic dependence of the retrieval time on volume of information recorded.

3 Audio data representation

The human's ear perceives analog signal, but to process it using a computer or to record it on CD it must be digitalized. At present there are some standards of sound digitalizing, which are distinguished by a sampling frequency and by the number of bits per sample.

In widely present audio processing systems sampling frequencies of 44.1 and 48 kHz are used as a rule. The most widely used number of bits per sample is 16 which results in a limit signal to noise ratio of 96dB.

In professional equipment higher resolutions are used: 18-, 20-, 24- and 32 bits per sample quantizations with the 56, 96 192 kHz sampling frequencies. It is done to preserve high harmonics in audio signals, which are not perceived by an ear directly, but have an influence on an overall sound image.

To digitalize more narrow-band signals and less qualitative signals a sampling frequency and the number of bits per sample may be reduced; for instance in telephone lines 7- or 8- bit quantization with a 8..12 kHz frequency is used.

At present the most widely used standard is 16 bit 44.1 kHz, e.g. musical compact discs use this standard. In this form 1 minute audio record will occupy 9Mb in digital form – one 74 minutes CD contains 650Mb of digital data.

To lower volume of data stored compression with a partial loss of information is used. At present a lot of standards of such compression are present, but the most common is MPEG Layer-3.

This compression method is based on the fact that human's ear is not ideal and in initial sound (e.g. CD-audio record) a lot of extra information is contained. The compression algorithm removes those signals which are inaudible or mostly inaudible for human. For example, if in addition to a strong audio signal a weak signal with a close frequency is present it will not be audible for human's ear. In a same way a human ear's perception is weakened for the certain time interval after and before an appearance of the strong audio impulse. In the psychoacoustic model used in the MPEG compression method total frequency spectra of initial audio is divided into parts in which signal level is considered to be constant; the signals inaudible by our ear are removed taking into account principles described above.

The MPEG method is quite flexible. It may be used to compress sound with a wide range of compression rates (approximately from 1:4 to 1:40) and wide range of sampling frequencies.

At present the most widely used compression standard is 16bit 44.1kHz 128kbps (i.e. the compression is 1:11), thus 1 minute is about 1Mb of data.

The input data for MPEG coders is uncompressed digital audio in PCM form (e.g. PCM WAV-file format). The compressed data will appear as a file on the output of a coder. This file has a linear structure, i.e. each its arbitrary fragment (with a lower limit by duration – a fragment cannot be shorter than one frame, a quant of encoded audio in MPEG format) will be a valid MPEG file.

To play compressed file appropriate decompressor, which uses MPEG file as input and sends uncompressed PCM audio data to audio card must be used.

4 Audio data recording and retrieval

Designed by present software complex Forget-me-not [3] based on the method for recording and storing the information on the trajectories of the nonlinear dynamic systems works in the following way:

• entire information contained in n files, is recorded onto one map;

• search request is a text string which is presented to the search program;

• an answer is a file containing sought string or a negative answer.

Such scheme of encoding of audio signals is difficult to use because of large volume of multimedia information compared to the volumes of text information which is usually used.

Therefore the following scheme will be used for processing audio information:

1. For each audio record to be put into the database information pattern, which volume is much smaller than that of initial audio record, is created. (The procedure of creation of such a pattern is described below);

2. Each information pattern created is being written into a separate map;

3. On a basis of the retrieval request, which is a fragment of audio record, information pattern, is created in the same way it was created in item 1; it is presented to the search program;

4. The search program will retrieve presented fragment through all maps in series;

5. The result of the search is information pattern from those written in the database or a negative answer of the system;

6. Information pattern is associated unambiguously with the audio record for which it was created.

Using this bond the full name of the file containing found audio record is restored;

7. Audio record is now reproduced.

5 Software implementation

For the practical check-up of the proposed ideas the software complex in Matlab environment was developed and created. It consists of two main blocks: a recording block and a search block.

Creation of information pattern is performed in the following way: from the file containing audio record the sequence consisting of each n-th byte is selected. This sequence is considered to be information pattern. The volume of such pattern will be n times and the recording time will be n^2 times smaller than those of the initial file.

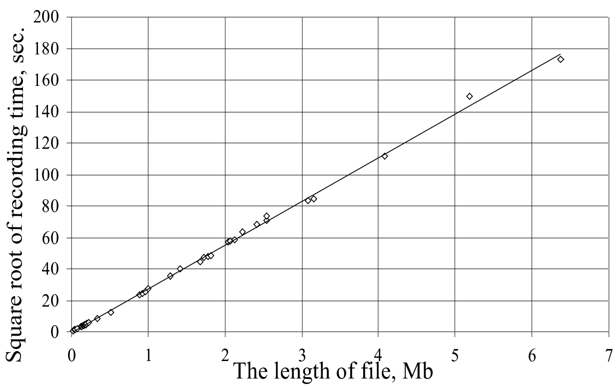

FIG. 3. Square root of the time of recording

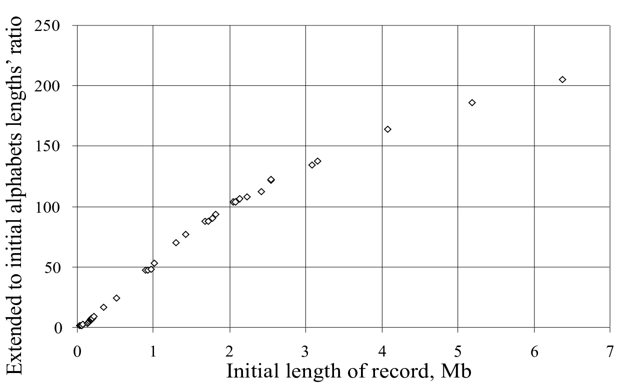

FIG. 4. Extended to initial alphabets lengths' ratio

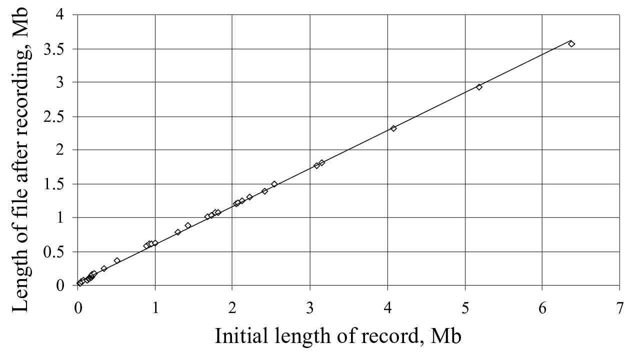

FIG. 5. Length of file after recording

Created information pattern is the one recorded onto a map. A map is represented by a matrix and recorded into a separate file along with its extended alphabet.

The second module is a search program. A search request is an arbitrarily small fragment of an audio record. Retrieval is performed for all information patterns in series.

On a basis of retrieval request information pattern is created and represented in extended alphabet. The presence of substrings of the information pattern on each map is verified. If at least one such substring is present on at least one map, retrieval is considered to be successful.

Performance of the complex was examined in this work .

As it was expected (part 2), the recording time depends on the length of the record, i.e. on volume of information being recorded, quadratically (fig. 3). An average length of audio record is 2 – 2.5 Mb. An average recording time of such audio record is 40 – 80 minutes. Taking into account nonoptimality of Matlab software environment, using which the complex was realized the time of recording one audio record in the same complex realized using C/C++ will be 2 – 3 minutes. As a part of the check-up more than 40 records of various lengths with total duration about 80 minutes were written.

An average retrieval time in the described above database was 2.5 – 3.5 minutes, 3 – 7 seconds for C/C++ realization. Retrieval time increases linearly with respect to the length of the database. An unambiguous identification of an audio record is reached even if its length is about 0.1 second.

The length of the map increases practically linearly with respect to the length of the file being recorded (Fig. 4). Compression ratio, i.e. map length to the initial information pattern ratio, for the records with lengths about megabyte is 1.5 – 2 (about 1.75 for 2 – 3Mb file, fig.5). The length of the extended alphabet increases with an increase of a file being recorded, and for 2 – 3Mb file it gives 25 – 30 thousand symbols.

6 Conclusions

Principles of recording and storing allowing retrieving the desired record from the database of audio records by an arbitrary piece of the record we are looking for are proposed.

For experimental check-up of the proposed principles in Matlab software environment software the complex for associative retrieval of audio information, allowing audio record retrieval by its arbitrary small fragment, was developed and realized.

The methods of optimization of the used algorithms, which allowed substantial reduction of demanded system resources, were proposed.

Developed complex may be considered as a prototype of a system for storage, fast retrieval and processing of audio information in the corporate networks and Internet.

References

[1] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. VISUAL 1999, Amsterdam, The Netherlands, 509–516

[2] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems. **3**, 231–262 (1994).

[3] Y.V. Andreyev, A.S. Dmitriev, and S.O. Starkov. "Pattern processing using one-dimensional dynamic systems. Software Package "Inform-Chaos". Moscow Institute of Radioengineering and Electronics. Russian Academy of Sciences. Preprint no. **2**, 584, 1993 (in Russian)

[4] Kyoji Hirata, Sougata Mukherjea, Yusaku Okamura, Wen-Syan Li, Yoshinori Hara. Object-based Navigation: An Intuitive Navigation Style for Content-oriented Integration Environment. In ACM Hypertext 97 Proceedings, 75–86 (1997).

[5] Charles Frankel, Michael J. Swain, and Vassilis Athitsos. WebSeer: An Image Search Engine for the World Wide Web. University of Chicago Department of Computer Science Technical Report **TR-96-14**, August 1996.

[6] X. Wan and C.-C. J. Kuo. Image retrieval with an octree-based color indexing scheme. 1997 IEEE International Symposium on Circuits and Systems. Hong Kong, June 9-12, 1997.

[7] S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. WWW / Computer Networks. **30** (1-7), 107-117 (1998)

[8] A.S. Dmitriev. Storing and recognision information in one-dimensional dynamical systems. Radiotekhnika i Electronica, vol. **36**, no. 1, 101–108, 1991 (in Russian).

[9] A.S. Dmitriev, A.I. Panas, S.O. Starkov. Storing and recognition information based on stable cycles of one dimensional maps. Phys. Lett. A. **155**, no. 8–9, 494–499 (1991).

[10] Y.V. Andreev, Y.L. Belsky, A.S. Dmitriev. Information processing in nonlinear systems with dynamic chaos. Doklady mezhdunarodnogo seminara "Nelineyniye tsepi i sistemy", Moskva, 51–60 (1992).

[11] Y.V. Andreev, A.S. Dmitriev, L.O. Chua, C.W. Wu. Associative and random access memory using one-dimensional maps. International Journal of Bifurcation and Chaos. vol. **3**, no. 2, 483–504 (1992).

[12] A.S. Dmitriev. Chaos and information processing in dynamical systems. Radiotekhnica i Elektronika. vol. **38**, no.1, 1–24, 1993 (in Russian).