Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Dictionary based color image retrieval

A. Macedonas, D. Besiris, G. Economou, S. Fotopoulos *

Electronics Laboratory, Department of Physics, University of Patras, Rio 26500, Greece

ARTICLE INFO

Article history:
Received 28 December 2007
Accepted 24 June 2008
Available online 2 July 2008

Keywords:
Color image retrieval
Data compression
Sequence analysis
Dictionary
Similarity metric
Kolmogorov complexity
Information theory
Histogram Intersection
Earth Movers Distance
Wald–Wolfowitz Test

ABSTRACT

In this work the normalized dictionary distance (NDD) is presented and investigated. NDD is a similarity metric based on the dictionary of a sequence acquired from a data compressor. A dictionary gives significant information about the structure of the sequence it has been extracted from. We examine the performance of this new distance measure for color image retrieval tasks, by focusing on three parameters: the transformation of the 2D image to a 1D string, the color to character correspondence, and the image size. We demonstrate that NDD can outperform standard (dis)similarity measures based on color histograms or color distributions.

1. Introduction

Over the last years the tremendous growth of the available information, stored in digital form raises many demanding issues. The task of retrieving relevant information clearly indicates the need for effective tools in data mining. When applied to 2D images these tools, fall into four different categories. There are schemes based on (i) the histogram, (ii) first and/or second order statistics, (iii) templates or (iv) compression. It is obvious that the operation and performance of each system depends critically on the feature extraction method used, while each feature reveals or hides some of the image characteristics.

The need of a universal similarity metric that minimizes every computable distance lead the authors in [1] to introduce the normalized information distance, based on the noncomputable notion of Kolmogorov complexity. On the same spirit, a parameter free, universal similarity distance, the normalized compression distance (NCD), computed from the lengths of compressed data files (singly and in pair wise concatenation) is introduced in [2], while a parameter free data mining algorithm based on the same theory [3] calculates the (dis)similarity of two sequences just by compressing them. Apart from these compression based similarity measures that have been proposed and applied, other researchers who worked in this area have included entropy based measures [4] or discriminant functions [5].

It is shown in [2] that for genomic sequences, music and literature, it is possible to find the (dis)similarities using the distance calculated from the compression of an object. Those sequences can be thought as 1D, because they have specific starting point as well as a distinct ending point. Informally one can think of the dimension of an object as the number of directions possible when describing it fully from a starting point. For a text, we say it is 1D, as there is a beginning, an end, and the words in between are in a particular order in one direction. Unfortunately, retrieval using compression does not trivially extend to higher dimensions. All available compressors deal with 1D sequences, so one has to translate the high dimensionality data in linear fashions. For some high dimensional data is difficult to express reasonably in fewer dimensions. For a 2D image the first problem occurs in the determination of the starting point. The most common starting point is the upper left pixel of the image. After this we have to choose which nearest pixel we should take next; to the right or below the first pixel. Furthermore, the third pixel should be chosen among pixels that are closer to the first pixel and those that are closer to the second one, and so on. Either choice we make, we have misrepresented the closeness of all the pixels in question.

While one of the most desirable properties of the compression based (dis)similarity measures is the ability to reveal (dis)similarities without any prior knowledge of the sample space, it is necessary to relax this condition when it comes to image processing. In [7] the authors limited their investigation

* Corresponding author. Fax: +30 2610 997456.
E-mail address: spiros@physics.upatras.gr (S. Fotopoulos).

1047-3203/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2008.06.006

to black and white images. In [8,9], the computation of the compression distance is calculated over gray scaled images, while in [10] their proposed method is evaluated over gray scaled textures. Moreover thorough examination in [11] shows that a choice of compression algorithm implies a specific, definable representation of the data within a feature space. Compression algorithms may be categorized according to the type of data (text, sound, and image) they are designed to compress. Most compression algorithms use two different approaches: one which generates a statistical model for the input data (bzip, LZ77, and LZW), and another which maps the input data to bit (Huffman encoding, and arithmetic encoding).

Another crucial problem is the size of the sequences to be compressed. In [12] experimental results obtained by the use of various compressors, reveal that the NCD is skewed by the size of the objects, independently of their type. For sequence sizes smaller than certain values (related to the block and window sizes in the compressors), the distance between two identical sequences is usually quite small, which proves that the NCD is a good tool for this purpose. However, for larger sizes, when the inner limitations of the compressors are violated, obviously the distance between two identical sequences grows to very high values, making the NCD practically unusable.

In this work we focus on retrieving color images using compression. We introduce a new similarity metric that is based on the dictionary of each image sequence. The words of each dictionary are subsequences of the initial sequence, while each word consists of a different succession of color characters. So the dictionary of an image captures in each character the color information at pixel level, while the succession order of those colors on the image plane is encoded in each word. Thus the problem of similarity between two color images can be translated to that of the similarity between the corresponding dictionaries. In order to produce the dictionary of each image we have to transform the 2D array to a 1D string using row by row or column by column scanning. The color information is embedded in the produced sequence by quantizing the RGB plane into cubes, where each cube is associated with a different color character.

The rest of the paper is organized as follows. In Section 2, we survey background and related work, while in Section 3 we present our framework for image retrieval. Experimental results as long as conclusions are given in Sections 4 and 5, respectively.

2. Background and related work

In this section we give some background information on the Kolmogorov complexity. We review some compression based similarity metrics which will help us to deploy our proposed method.

2.1. Kolmogorov complexity

Kolmogorov complexity is a measure of randomness of strings based on their information content. It was proposed to quantify the randomness of strings and other objects in an objective and absolute manner. One can think of the Kolmogorov complexity of an object as its shortest description. Considering this measure, an object is complex if its shortest description is very long. Essentially, the Kolmogorov complexity of a file is the length of the ultimate compressed version of the file.

We can define the Kolmogorov complexity $K(x)$ of a string x as the length of the shortest program capable of reproducing x on a universal computer, such as a Turing machine. The conditional Kolmogorov complexity $K(x|y)$ of x relative to y is defined similarly as the length of the shortest program to compute x if y is given as an auxiliary input to the computation.

2.2. Normalized information distance

The normalized information distance is a similarity metric based on the concept of Kolmogorov complexity. This metric does not need any prior knowledge about the object to be tested. It is based on the *information distance* $E(x,y)$, introduced in [6].

The *information distance* defined as the length of the shortest binary program for the reference universal prefix Turing machine that, with input x computes y, and vies versa. It is shown that, up to an additive logarithmic $O(\log(\max\{K(x,y),K(y,x)\}))$ term

$$E(x,y) = \max\{K(x,y),K(y,x)\} \tag{1}$$

It is proved in [6] that the *information distance* $E(x,y)$ is a metric.

The normalized version of $E(x,y)$, called the *normalized information distance*, is defined as

$$\mathrm{NID}(x,y) = \frac{\max\{K(x,y),K(y,x)\}}{\max\{K(x),K(y)\}} \tag{2}$$

It too is a metric, and it is universal in the sense that this single metric minimizes up to a minor additive $O(1/\max\{K(x),K(y)\})$ term all normalized admissible distances in the class considered in [1].

2.3. Normalized compression distance

Unfortunately, the Kolmogorov complexity is non computable in the Turing sense. Thus the NID is computed by some approximations. The result of approximating the *normalized information distance* by the use of a real compressor C is called the *normalized compression distance*

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}} \tag{3}$$

where $C(x,y)$ denotes the compressed size of the concatenation of x and y, $C(x)$ denotes the compressed size of x, and $C(y)$ corresponds to the compressed size of y. The NCD is a nonnegative number representing how different the two sequences are. Smaller numbers represent more similar sequences.

The calculation of this distance does not require a specific compressor, while for lossless compressors, $C(\cdot) = K(\cdot) + k$, where k is unknown and depends on the data and the compressor.

2.4. The dictionary of a sequence

Dictionary based data compressors identify patterns, called *words*, of the data and store them in a *dictionary*. In particular, it segments a sequence into several distinct subsequences called *words*, in a way that each word is the shortest subsequence that is not previously parsed as *word*. It is clear that the same word can appear several times in the *dictionary*. The number of the words belonging to a *dictionary* depends not only on the length of the initial sequence, but also on the number of the characters used and the way they are combined into words. Some dictionary coders use a “static dictionary”; one whose full set of strings is determined before coding begins and does not change during the coding process. More common are methods where the dictionary starts in some predetermined state but the contents change during the encoding process, based on the data that has already been encoded.

In order to understand the *dictionary* creation process, it is useful to recall how the LZW compression algorithm [13] works. The *dictionary* is formed directly for the incoming sequence. As the compressor serially examines the sequence, it stores every unique two-character string into the dictionary table as a code/character concatenation, with the code mapping to the corre-

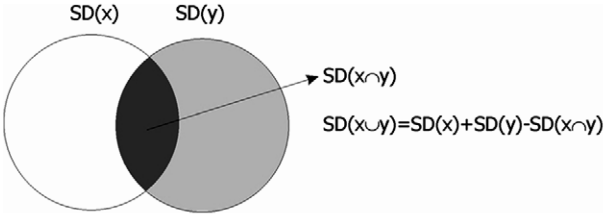

Fig. 1. Illustration of the sequence dictionaries where circle $\mathrm{SD}(x)$ represents the dictionary of sequence x, circle $\mathrm{SD}(y)$ represents the dictionary of sequence y and the total area of the two circles $\mathrm{SD}(x \cup y)$ is the joint dictionary of the two sequences x and y.

sponding first character. As each two-character string is stored, the first character is exported. Whenever a previously encountered string is read from the input, the longest such previously encountered string is determined, and then the code for this string concatenated with the extension character (the next character in the input) is stored in the table. The code for this longest previously encountered string is outputted and the extension character is used as the beginning of the next string.

For a sequence with length n and words with length w, the computational complexity of the dictionary formation is proved to be $O(K \log K)$, where $K = w + n$[14].

Fig. 1 illustrates some principal relations of two sequence dictionaries. We will show that sequence dictionary satisfies the following properties:

(1) Idempotency: $\mathrm{SD}(x \cup x) = \mathrm{SD}(x)$, and $\mathrm{SD}(\lambda) = \emptyset$, where λ is the empty string.
(2) Monotonicity: $\mathrm{SD}(x \cup y) \geqslant \mathrm{SD}(x)$.
(3) Symmetry: $\mathrm{SD}(x \cup y) = \mathrm{SD}(y \cup x)$.
(4) Distributivity: $\mathrm{SD}(x \cup y) + \mathrm{SD}(z) \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z)$.

- *Idempotency*. The joint dictionary created by two identical sequences will be the same dictionary extracted from that sequence. Also the dictionary of an empty string will be empty.
- *Monotonicity*. From Fig. 1 we see that $\mathrm{SD}(x \cap y) \geqslant \emptyset$ (in case that the two dictionaries are completely unrelated the intersection of these two disjoint dictionaries will produce an empty set). So $\mathrm{SD}(x \cup y) \geqslant \mathrm{SD}(x) + \mathrm{SD}(y) \geqslant \mathrm{SD}(x)$ as $\mathrm{SD}(y) \geqslant \emptyset$ (the equality stands for the empty string).
- *Symmetry*. One of the basic properties of set theory is $\mathrm{SD}(x \cup y) = \mathrm{SD}(y \cup x)$.
- *Distributivity*. The proof of this property can be found in Appendix A.

3. Dictionary based similarity metric

3.1. The normalized dictionary distance (NDD)

Given two sequences x and y, we define the *normalized dictionary distance* as follows

$$\mathrm{NDD}(x,y) = \frac{\mathrm{SD}(x \cup y) - \min\{\mathrm{SD}(x), \mathrm{SD}(y)\}}{\max\{\mathrm{SD}(x), \mathrm{SD}(y)\}} \tag{4}$$

where $\mathrm{SD}(x)$ is the sequence dictionary of x, $\mathrm{SD}(y)$ is the sequence dictionary of y, and $\mathrm{SD}(x \cup y)$ is the joint dictionary of the two sequences x and y.

For two identical sequences we have $\mathrm{NDD}(x,x) = 0$ $(\mathrm{SD}(x \cup x) = \mathrm{SD}(x))$, whilst for two sequences with disjoint dictionaries we have $\mathrm{NDD}(x,y) = 1$ $(\mathrm{SD}(x \cup y) = S(x) + S(y))$.

The NDD is unchanged by interchanging x and y in Eq. (4). It is obvious, as it can been seen in Fig. 1, that $\mathrm{SD}(x \cup y) = \mathrm{SD}(y \cup x)$, which leads to $\mathrm{NDD}(x,y) = \mathrm{NDD}(y,x)$.

If we deploy farther the Eq. (4) (with the help of set properties), we have

$$\mathrm{NDD}(x,y) = \frac{\mathrm{SD}(x) + \mathrm{SD}(y) - \mathrm{SD}(x \cap y) - \min\{\mathrm{SD}(x), \mathrm{SD}(y)\}}{\max\{\mathrm{SD}(x), \mathrm{SD}(y)\}}$$

While $0 \leqslant \mathrm{SD}(x \cap y) \leqslant \min(\mathrm{SD}(x), \mathrm{SD}(y))$, the NDD is a nonnegative number $0 \leqslant \mathrm{NDD} \leqslant 1$ representing how different the two sequences are. Smaller numbers represent more similar dictionaries.

The NDD is a normalized admissible distance satisfying the metric (in) equalities, that is, a similarity metric (see Appendix B).

3.2. Transforming a 2D image to 1D sequence

The first step for the generation of an image *dictionary* is to convert the image in a 1D sequence. In order to accomplish this we consider the pixels row by row, and column by column as it is shown in Fig. 2.

It is obvious that different scanning procedures generate different 1D sequence, which generates a different *dictionary*. Moreover by reshaping an image in this way we preserve part of the spatial image information, information that cannot be preserved by histogram based techniques.

3.3. Embedding the color information in the 1D sequence

One of the most challenging tasks in this work was how to embed the color information in the 1D sequence. Most of the previews compression based approaches [8–10], discard color

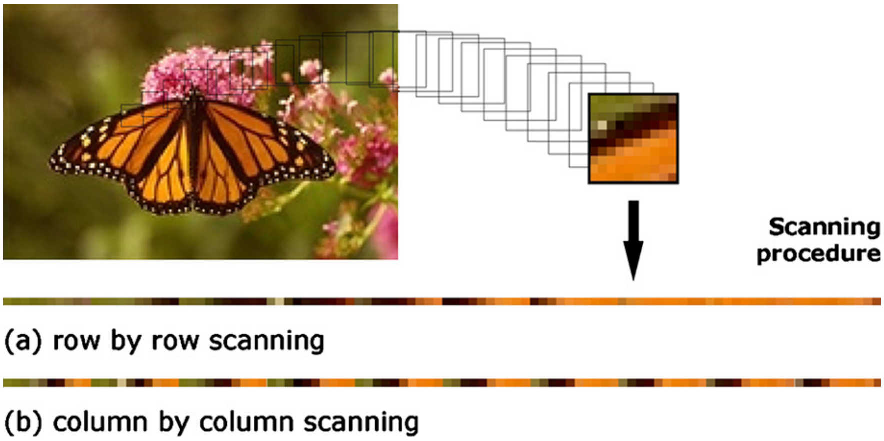

Fig. 2. Conversion for 2D to 1D sequence.

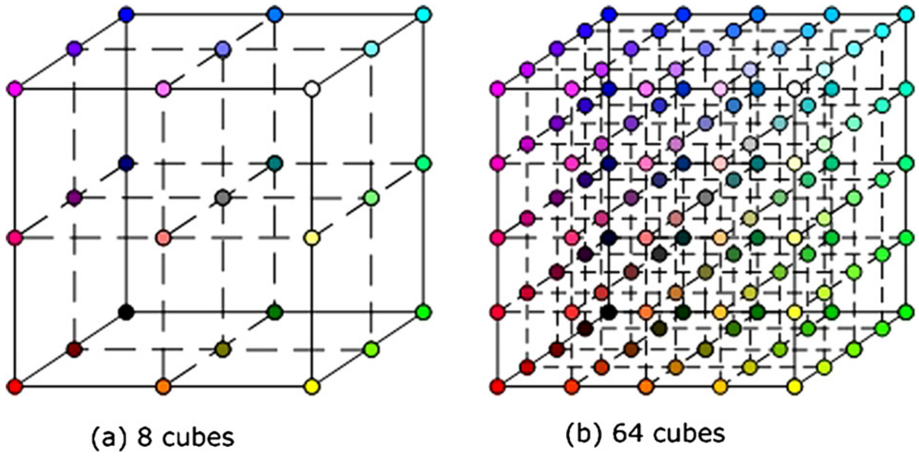

Fig. 3. Division of RGB space into (a) 8 and (b) 64 cubes.

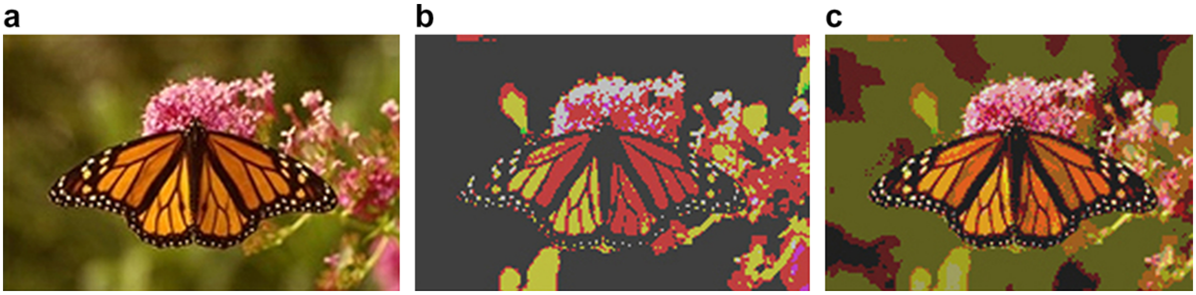

Fig. 4. Illustration of quantizing the initial image (a) into 8 levels (b) and 64 levels (c).

and compute the (dis)similarity of two images, based on notions of information distance, computed using intensity values.

Motivated by the 3D structure of the RGB color space, we divide the RGB plane into 8 and 64 cubes as shown in Fig. 3. In each cube we assign a *character*, thus the resulting image sequence will consist of 8 *characters* in case of 8 cube division, or 64 *characters* in case of a 64 cube division. The quantization results are depicted in Fig. 4.

4. Experimental study

4.1. Data set performance evaluation

The collection of images used to produce the experimental results presented in this work is part of the Corel Gallery [15]. This collection consists of $D = 1000$ still color images of 24 bpp each, given in portable pixel map format of sizes $[192 \times 128]$ or $[128 \times 192]$ pixels that are pre-assigning into 20 distinct classes (e.g. cars, views, flowers, airplanes, etc.). Furthermore another 60 images, three for each one of the above classes, were also selected from the Corel Gallery to construct the Query image data base $Q = 60$. The evaluation procedure was run over each Query, and finally the performance was averaged across all Queries.

In order to evaluate the performance of our proposed methodology, we used the *precision (Pr)* and *recall (Re)* quantities [16]. These quantities were used for comparing our approach, with several other (dis)similarity measures that are based on color histograms and color distributions [17–21]. In all cases, the reported results are averaged scores, referring to the same set of Query images.

In the following lines we will give a short description of those measures used for comparison purposes, assuming that $Q = \{q_i\}$ and $D = \{d_i\}$ are the histograms from a query image Q and a database image D, respectively, each one containing n bins and $k_i = \frac{q_i + d_i}{2}$.

Kullback–Leibler Divergence (KLD): It measures how inefficient, on average, it would be to code one histogram using the other as the code-book [17,18]:

$$d_{\mathrm{KLD}}(Q, D) = \sum_{i=1}^{n} q_i \log \frac{q_i}{d_i}$$

Jeffrey Divergence (JD): It is a modification of KLD that is symmetric, numerical stable and robust with respect to noise and number of bins taken into count [17], given by

$$d_{\mathrm{JD}}(Q, D) = \sum_{i=1}^{n} \left(q_i \log \frac{q_i}{k_i} + d_i \log \frac{d_i}{k_i} \right)$$

Histogram Intersection (HI): This measure is used for color image retrieval in the spatial domain [19] and it is found to be attractive due to its ability to handle partial matches [18]. For two equal histograms, the HI is equivalent to the L_1 distance. The HI distance is given by

$$d_{\mathrm{HI}}(Q, D) = 1 - \frac{\sum_{i=1}^{n} \min(q_i, d_i)}{\sum_{i=1}^{n} d_i}$$

We have applied HI in 8 bin and 64 bin histograms derived from the RGB quantization scheme.

χ^2 statistics (χ^2): It is a statistical index showing how likely is for one distribution to get drawn from the population represented by the other [17,18], and is given by

$$d_{\chi^2}(Q, D) = \sum_{i=1}^{n} \frac{q_i - k_i}{k_i}$$

Earth Movers Distance (EMD): This distance is based on the solution of the transportation problem [18]. It is calculated over the minimum amount of work needed in order to transform one distribution into the other, normalized by the sum of the costs needed to move the individual features:

$$d_{\mathrm{EMD}}(Q,D)=\frac{\sum_{i,j}f_{ij}g_{ij}}{\sum_{i,j}f_{ij}}$$

where g_{ij} is the ground distance between bins q_i and d_j, respectively, f_{ij} is the optimal flow between the two distributions such that the total cost $\sum_{i,j}f_{ij}$ is minimized, subject to some constrains [18].

Multivariate Wald–Wolfowitz Test (ww-test): This measure can be used to test whether any two given multi-dimensional point samples are coming from the same multivariate distribution [20]. A minimal spanning tree (MST) is built over all points in $\mathbf{R}^p$. Then, based on the sample identities of the points, a test statistic R is computed, representing the total number of runs (a consecutive sequence of identical sample identities). It is shown that the quantity:

$$W=\frac{R-E[R]}{\sqrt{\mathrm{Var}[R]}}$$

approaches asymptotically the standard normal distribution. The $E[R]$ and $\mathrm{Var}[R]$ quantities of R depend on the sizes m and n of the two point samples and can be computed as

$$E[R]=\frac{2mn}{N}+1$$

and

$$\mathrm{Var}[R]=\frac{2mn}{N(N-1)}\times\left\{\frac{2mn-N}{N}+\frac{C-N+2}{(N-2)(N-3)}[N(N-1)-4mn+2]\right\}$$

where $N=m+n$ and C is the number of edge pairs sharing a common node.

Color Coherence Vectors (CCV): This histogram based method [21] is used for comparing images that incorporate spatial information. Each pixel is classified in a given color bucket as either coherent or incoherent, based on weather or not is part of a large similarly colored region. Considering two images Q and D, together with their CCV's G_Q and G_D let the number of coherent pixels in color bucket i be α_{Qi} and α_{Di}, while denote the number of incoherent pixels as b_{Qi} and b_{Di}, respectively. The distance between the two CCV's is given by the quantity:

$$D_{\mathrm{CCV}}(Q,D)=\sum_{i=1}^{n}|(\alpha_{Qi}-\alpha_{Di})|+|(b_{Qi}-b_{Di})|$$

where n is the number of the color buckets used. We have applied CCV using 8 and 64 color buckets derived from the RGB quantization scheme.

4.2. Experimental results

For the introduced methodology, the results are coming from different settings of the involved parameters, which are the scanning procedure, the color to character mapping, and the sequence length (image size). In order to construct the image dictionary, each one of the three RGB vectors is quantized into two levels, producing eight distinct cubes everyone associated with a distinct character. The dictionary of each image sequence is created from different combination of these characters.

In the case of the color histogram, the only free parameter is the number of bins used, which was set to 20 for each RGB-channel. Experimentation with a wide range of binning values gave no significant changes in recall performance. Finally for the development of the *Multivariate Wald–Wolfowitz Test* [20], 60 pixels were randomly selected from each image.

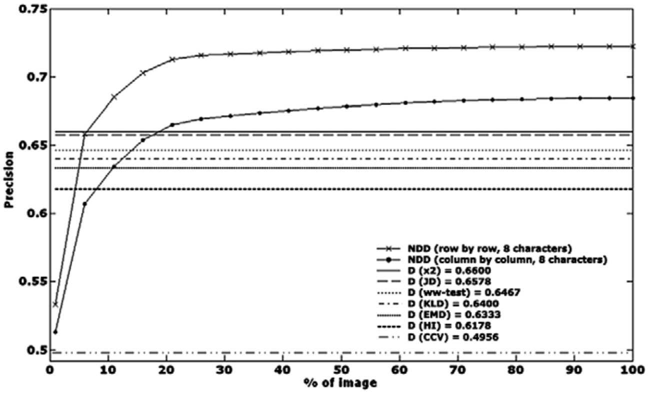

Fig. 5. Precision retrieval results, as a function of image scaling factor for the 10 most similar images. The two top curves correspond to the new dictionary based technique, for row by row ($*$) and column by column ($\cdot$) scanning procedure, respectively. The dictionary of each image sequence is created using eight characters. Highest precision achieved by the other dissimilarity measures are depicted as horizontal lines.

In Fig. 5 we illustrate retrieval results for the 10 top retrieved images, along with the corresponding level of precision for each dissimilarity measure. The NDD was implemented, using an eight character dictionary derived from a row by row and a column by column scanning procedure. In order to change the sequence length we had to resize the original image by nearest neighbor interpolation. The horizontal lines shown in the same figure indicate the highest precision index that was achieved by the other dissimilarity measures. It is clear that both NDD procedures perform better than the other dissimilarity measures. It is also observed in Fig. 5, that the precision of the proposed system increases along with image size. The histogram based techniques are empirical estimates of the image distribution; a feature that is not significantly affected by image size and gave no significant deviation with this change, demonstrating their scale invariant nature. On the contrary in the proposed NDD methodology, image content is described more accurately as the length of the sequence increases and it reaches its maxima for image size beyond the 25% of the original.

Another interesting point is the fact that the row by row scanning scheme gives better results than the column by column one. The row by row scanning procedure appears to give a better representation of the spatial information contained in an image. In order to investigate this finding and test if it is related to image aspect ratio (portrait or landscape), we cropped every image and kept a centered square window of size [128 × 128], assuming the main image information is positioned in its center part. Yet, this did not lead to any significant deviations from our initial observations.

Next, the effect of the character set that was used to produce the dictionaries is examined. Each one of the three RGB vectors was divided into four levels to produce 64 distinct cubes. Results indicate that the quantization of the RGB plane plays a significant role in the NDD. A more detailed representation of each image, leads to more accurate description of the image and better retrieval results. In Fig. 6 it can be seen that the behavior of our proposed technique boosts up with the use of a 64 character dictionary. At 10% of the image size, the system reached a plateau, and gave no significant change in its performance by increasing further the image size. The horizontal lines shown in Fig. 6 correspond to the highest precision index that was achieved by the other measures considered in this work.

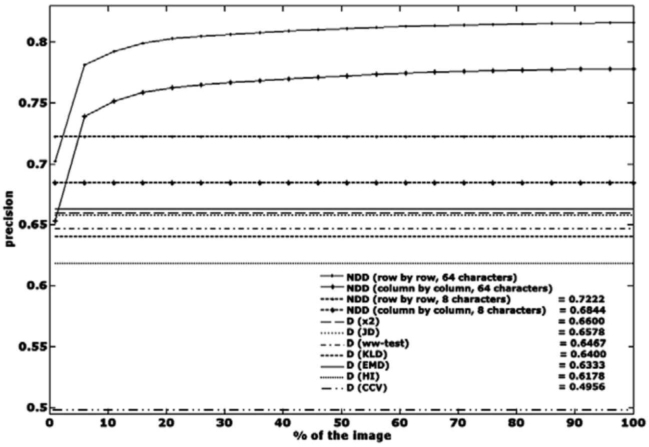

Fig. 6. Precision as a function of image size, for the 10 most similar images. Measurements of the two curves correspond to row by row and column by column pixel scanning, respectively. The dictionary of each image sequence is created using 64 characters. The maximum precisions values of the other examined dissimilarity measures are depicted as horizontal lines.

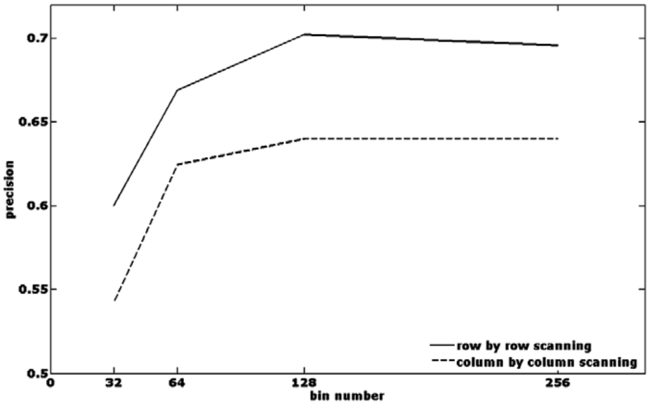

Fig. 7. Precision as a function of bin number, for the 10 most similar images. The color embedding took place in the HMMD color space. Measurements of the two curves correspond to row by row and column by column pixel scanning, respectively.

We also run some tests by first transforming the RGB color space to the HMMD color space introduced in MPEG-7 [22], and then quantizing the resulting color space in 32, 64, 128, and 256 levels. Fig. 7 illustrates the results of those tests. It is clear that the embedding of the color information in the sequence using the RGB cubes accomplishes better results. Furthermore as we increase the number of color quantization levels we observe that no significant changes occur in the precision of the system.

Finally, in Fig. 8 the operation of all the different methods under comparison was evaluated by using the standard *Precision* versus *Recall* diagram. It is observed that the NDD methodology employing the row by row scanning procedure gives superior results indicating that the representation of image information content is better by the horizontal scanning procedure.

5. Conclusions

In this work we introduce a novel compression based similarity metric based on the sequence dictionary. The dictionary can be easily extracted by available data compressors and contains valuable information about the structure of a data sequence. Thus for any pair of sequences dictionary similarity directly reflects the sequence similarity.

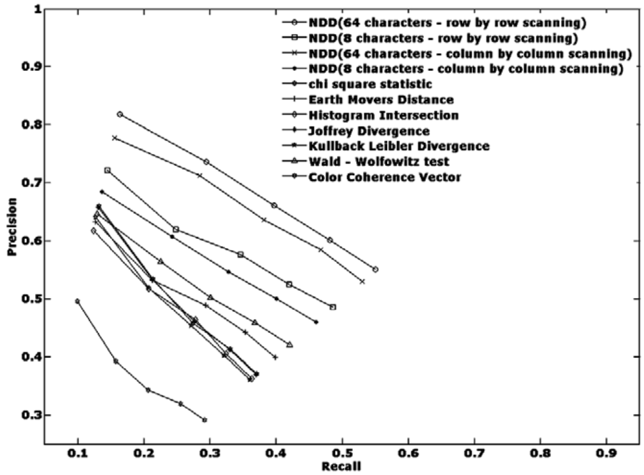

Fig. 8. Comparison of retrieval performance via the Precision versus Recall diagram for the different approaches.

The new similarity metric is tested successfully in the color image retrieval problem producing better results than traditional techniques. The new approach examines deeper the image plane by exploiting not only color values and the resulting distribution like histograms do, but also utilizing spatial relationship between pixels giving a very good description of the local color variation.

Experimental results indicate that when applied in color image retrieval, the NDD, is affected by the size of the image sequence, the color to character embedding and the transformation of the 2D image to a 1D sequence. It is observed that as the set of color characters used for the description of an image as well as image size increases, the performance of the system reaches a plateau. Additionally it is also noticed that the row by row scanning procedure captures more efficiently the image spatial information although we could not provide a viable explanation about this.

Finally as far as computational complexity is concerned, the most time consuming process of the method is the dictionary formation which is completed in $O(K \log K)$ time. The transformation of the 2D image in a 1D sequence, the embedding of the color information in that sequence, and the computation of the NDD is done in linear time.

Acknowledgments

This work was financially supported by the General Secretary of Research and Technology of the Greek Ministry under project PENED 2003.

Appendix A. Proof of the distributivity property

The distributivity property is not immediately intuitive. We are going to saw that the stronger distributivity property

$$\mathrm{SD}(x \cup y \cup z) + \mathrm{SD}(z) \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z)$$

holds.

For set theory we have for three sequences $x,\ y,\ z$ that

$$\begin{aligned} \mathrm{SD}(x \cup y \cup z) = {} & \mathrm{SD}(x) + \mathrm{SD}(y) + \mathrm{SD}(z) - \mathrm{SD}(x \cap y) - \mathrm{SD}(x \cap z) \\ & - \mathrm{SD}(y \cap z) + \mathrm{SD}(x \cap y \cap z) \Longleftrightarrow \mathrm{SD}(x \cup y \cup z) \\ & + \mathrm{SD}(z) = \mathrm{SD}(x) + \mathrm{SD}(y) + 2\mathrm{SD}(z) - \mathrm{SD}(x \cap y) \\ & - \mathrm{SD}(x \cap z) - \mathrm{SD}(y \cap z) + \mathrm{SD}(x \cap y \cap z) \end{aligned}$$

But $\mathrm{SD}(x \cup z) = \mathrm{SD}(x) + \mathrm{SD}(z) - \mathrm{SD}(x \cap z)$ and $\mathrm{SD}(y \cup z) = \mathrm{SD}(y) + \mathrm{SD}(z) - \mathrm{SD}(y \cap z)$. So we have

$$
\begin{aligned}
&\mathrm{SD}(x \cup y \cup z) + \mathrm{SD}(z)\\
&\quad = \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z) - \mathrm{SD}(x \cap y) + \mathrm{SD}(x \cap y \cap z)\\
&\quad = \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z) - [\mathrm{SD}(x \cap y) - \mathrm{SD}(x \cap y \cap z)]\\
&\quad = \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z) - \mathrm{SD}(x \cap y \cap [1 - z])\\
&\quad = \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z) - \mathrm{SD}(x \cap y \cap \bar{z})\\
&\quad \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z)
\end{aligned}
$$

as $\mathrm{SD}(x \cap y \cap \bar{z}) \geqslant \emptyset$ ($\bar{z}$ is the complementary set of z).

Farther more we have $\mathrm{SD}(x \cup y \cup z) \geqslant \mathrm{SD}(x \cup y) + \mathrm{SD}(z)$, which lead us to

$$\mathrm{SD}(x \cup y) + \mathrm{SD}(z) \leqslant \mathrm{SD}(x \cup y \cup z) + \mathrm{SD}(z) \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z).$$

Appendix B. Proof of the triangle inequalities of NDD

Without loss of generality we assume that $\mathrm{SD}(x) \leqslant \mathrm{SD}(y) \leqslant \mathrm{SD}(z)$ and $x, y, z \neq \lambda$. This assumption provides

$$
\begin{aligned}
&\mathrm{SD}(x) = \min(\mathrm{SD}(x), \mathrm{SD}(y)), \quad \mathrm{SD}(x) = \min(\mathrm{SD}(x), \mathrm{SD}(z)),\\
&\mathrm{SD}(y) = \min(\mathrm{SD}(y), \mathrm{SD}(z)), \quad \mathrm{SD}(y) = \max(\mathrm{SD}(x), \mathrm{SD}(y)),\\
&\mathrm{SD}(z) = \max(\mathrm{SD}(x), \mathrm{SD}(z)), \quad \mathrm{SD}(z) = \max(\mathrm{SD}(y), \mathrm{SD}(z)).
\end{aligned}
$$

Since the NDD is symmetrical, there are only three triangle inequalities that can be expressed by $\mathrm{NDD}(x, y), \mathrm{NDD}(x, z), \mathrm{NDD}(y, z)$. We will verify them one by one.

(1) $\mathrm{NDD}(x, y) \leqslant \mathrm{NDD}(x, z) + \mathrm{NDD}(y, z)$: By distributivity the sequence dictionary itself satisfies $\mathrm{SD}(x \cup y) + \mathrm{SD}(z) \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z)$. Subtracting $[\mathrm{SD}(x) + \mathrm{SD}(y)]$ from both sides and rewriting, $\mathrm{SD}(x \cup y) + \mathrm{SD}(z) - [\mathrm{SD}(x) + \mathrm{SD}(y)] \leqslant \mathrm{SD}(x \cup z) + \mathrm{SD}(y \cup z) - [\mathrm{SD}(x) + \mathrm{SD}(y)] \Rightarrow [\mathrm{SD}(x \cup y) - \mathrm{SD}(x)] + \mathrm{SD}(z) - \mathrm{SD}(y) \leqslant [\mathrm{SD}(x \cup z) - \mathrm{SD}(x)] + [\mathrm{SD}(y \cup z) - \mathrm{SD}(y)]$. Dividing both sides by $\mathrm{SD}(z)$, we find $\frac{\mathrm{SD}(x \cup y) - \mathrm{SD}(x)}{\mathrm{SD}(z)} + 1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \leqslant \frac{\mathrm{SD}(x \cup z) - \mathrm{SD}(x)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(y \cup z) - \mathrm{SD}(y)}{\mathrm{SD}(z)} \Rightarrow \frac{\mathrm{SD}(x \cap y) - \min(\mathrm{SD}(x), \mathrm{SD}(y))}{\max(\mathrm{SD}(x), \mathrm{SD}(y))} \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + 1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \leqslant \frac{\mathrm{SD}(x \cup z) - \min(\mathrm{SD}(x), \mathrm{SD}(z))}{\max(\mathrm{SD}(x), \mathrm{SD}(z))} + \frac{\mathrm{SD}(y \cup z) - \min(\mathrm{SD}(y), \mathrm{SD}(z))}{\max(\mathrm{SD}(y), \mathrm{SD}(z))} \Rightarrow \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + 1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \leqslant \mathrm{NDD}(x, z) + \mathrm{NDD}(y, z)$. We have to prove that $\mathrm{NDD}(x, y) \leqslant \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + 1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)}$. Moving all variables to the left side we have $\mathrm{NDD}(x, y) - \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - 1 + \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \leqslant 0 \Rightarrow \mathrm{NDD}(x, y) \left[1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)}\right] - \left[1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)}\right] \leqslant 0 \Rightarrow [\mathrm{NDD}(x, y) - 1] \left[1 - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)}\right] \leqslant 0$. The first bracket is always less than or equal to zero, while the second one is always greater than or equal to zero, thus their product will always be less than or equal to zero.

(2) $\mathrm{NDD}(x, z) \leqslant \mathrm{NDD}(x, y) + \mathrm{NDD}(z, y)$: By distributivity we have $\mathrm{SD}(x \cup z) + \mathrm{SD}(y) \leqslant \mathrm{SD}(x \cup y) + \mathrm{SD}(z \cup y)$. Subtracting $[\mathrm{SD}(x) + \mathrm{SD}(y)]$ from both sides, rearranging, and dividing both sides by $\mathrm{SD}(z)$ we obtain $\frac{\mathrm{SD}(x \cup z) - \mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant \frac{\mathrm{SD}(x \cup y) - \mathrm{SD}(x)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(z \cup y) - \mathrm{SD}(y)}{\mathrm{SD}(z)} \Rightarrow \frac{\mathrm{SD}(x \cup z) - \min(\mathrm{SD}(x), \mathrm{SD}(z))}{\max(\mathrm{SD}(x), \mathrm{SD}(z))} \leqslant \frac{\mathrm{SD}(x \cup y) - \min(\mathrm{SD}(x), \mathrm{SD}(y))}{\max(\mathrm{SD}(x), \mathrm{SD}(y))} \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(z \cup y) - \min(\mathrm{SD}(z), \mathrm{SD}(y))}{\max(\mathrm{SD}(z), \mathrm{SD}(y))} \Rightarrow \mathrm{NDD}(x, z) \leqslant \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \mathrm{NDD}(z, y)$. We have to prove that $\mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \mathrm{NDD}(z, y) \leqslant \mathrm{NDD}(x, y) + \mathrm{NDD}(z, y)$. Moving all variables to the left side $\mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - \mathrm{NDD}(x, y) \leqslant 0 \Rightarrow \mathrm{NDD}(x, y) \left[\frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - 1\right] \leqslant 0$. But $\mathrm{NDD}(x, y)$ is greater than or equal to zero, while $\frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - 1 \leqslant 0$, thus their product will always be less than or equal to zero.

(3) $\mathrm{NDD}(y, z) \leqslant \mathrm{NDD}(x, y) + \mathrm{NDD}(x, z)$: By distributivity we have $\mathrm{SD}(y \cup z) + \mathrm{SD}(x) \leqslant \mathrm{SD}(y \cup x) + \mathrm{SD}(z \cup x)$. Subtracting $[2\mathrm{SD}(x) + \mathrm{SD}(y)]$ from both sides, and dividing both sides by $\mathrm{SD}(z)$ we obtain $\frac{\mathrm{SD}(y \cup z) - \mathrm{SD}(y)}{\mathrm{SD}(z)} - \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant \frac{\mathrm{SD}(y \cup x) - \mathrm{SD}(x)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(z \cup x) - \mathrm{SD}(x)}{\mathrm{SD}(z)} - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \Rightarrow \frac{\mathrm{SD}(y \cup z) - \min(\mathrm{SD}(y), \mathrm{SD}(z))}{\max(\mathrm{SD}(y), \mathrm{SD}(z))} - \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant \frac{\mathrm{SD}(x \cup y) - \min(\mathrm{SD}(x), \mathrm{SD}(y))}{\max(\mathrm{SD}(x), \mathrm{SD}(y))} \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(z \cup x) - \min(\mathrm{SD}(z), \mathrm{SD}(x))}{\max(\mathrm{SD}(z), \mathrm{SD}(x))} - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \Rightarrow \mathrm{NDD}(x, z) - \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \mathrm{NDD}(z, y) - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} \Rightarrow \mathrm{NDD}(x, z) \leqslant \mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \mathrm{NDD}(z, y) + \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)}$. We have to prove that $\mathrm{NDD}(x, y) \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \mathrm{NDD}(z, y) + \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant \mathrm{NDD}(x, y) + \mathrm{NDD}(z, y) \Rightarrow \mathrm{NDD}(x, y) \left[\frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} - 1\right] - \frac{\mathrm{SD}(y)}{\mathrm{SD}(z)} + \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant 0 \Rightarrow \mathrm{NDD}(x, y) \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(z)}{\mathrm{SD}(z)}\right] \leqslant \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(z)}\right]$.

Case 1: $\mathrm{SD}(y) = \mathrm{SD}(z)$. $\mathrm{NDD}(x, y) \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(z)}{\mathrm{SD}(z)}\right] \leqslant \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(z)}\right] \Rightarrow 0 \leqslant 1 - \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \Rightarrow \frac{\mathrm{SD}(x)}{\mathrm{SD}(z)} \leqslant 1$ which is valid.

Case 2: $\mathrm{SD}(y) < \mathrm{SD}(z)$.The fraction $\left[\frac{\mathrm{SD}(y) - \mathrm{SD}(z)}{\mathrm{SD}(z)}\right]$ is less than zero, so $\mathrm{NDD}(x, y) \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(z)}{\mathrm{SD}(z)}\right] \leqslant \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(z)}\right] \Rightarrow \mathrm{NDD}(x, y) \geqslant \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(y) - \mathrm{SD}(z)}\right]$. The fraction $\left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(y) - \mathrm{SD}(z)}\right]$ is less than or equal to zero, because the numerator is greater than or equal to zero and the denominator is less than zero. Finally $\mathrm{NDD}(x, y) \geqslant 0 \geqslant \left[\frac{\mathrm{SD}(y) - \mathrm{SD}(x)}{\mathrm{SD}(y) - \mathrm{SD}(z)}\right]$.

References

[1] M. Li, X. Chen, X. Li, B. Ma, P. Vitànyi, The similarity metric, IEEE Trans. Inf. Theory 50 (2004) 3250–3264.

[2] R. Cilibrasi, P. Vitànyi, Clustering by compression, IEEE Trans. Inf. Theory 51 (2005) 1523–1545.

[3] E. Keogh, S. Lonardi, C.A. Ratanamahatana, Towards parameter-free data mining, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, 2004, pp. 206–215.

[4] D. Benedetto, E. Caglioti, V. Loreto, Language trees and zipping, Phys. Rev. Lett. 88 (4) (2002).

[5] E. Frank, C. Chu, I.H. Witten, Text categorization using compression models, IEEE Data Compression Conf. (2000) 200–209.

[6] C.H. Bennett, P. Gàcs, M. Li, P. Vitànyi, W. Zurek, Information distance, IEEE Trans. Inf. Theory (1998) 1407–1423.

[7] B. Hescott, D. Koulomzin, On Clustering Images Using Compression, Technical Report, CS Department, Boston University, 2007.

[8] M. Li, Y. Zhu, Image classification via LZ78 based string kernel: a comparative study, Lect. Notes Comput. Sci. (2006) 704–712.

[9] Y. Lan, R. Harvey, Image classification using compression distance, in: Proceedings of the Second International Conference on Vision, Video and Graphics, Edinburgh, 2005.

[10] L.V. Batista, M.M. Meira, N.L. Canalcanti Jr., Texture classification using local and global histogram equalization and the Lempel–Ziv–Welch algorithm, in: Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, 2005.

[11] D. Sculley, C.E. Brodeley, Compression and machine learning: a new perspective on feature space vectors, IEEE Proc. Data Compression Conf. (2006).

[12] M. Cebriàn, M. Alfonseca, A. Ortega, Common pitfalls using the normalized compression distance: what to watch out for in a compressor, Commun. Inf. Syst. 5 (4) (2005) 367–384.

[13] T.A. Welch, A technique for high performance data compression, IEEE Comput. 17 (1984) 8–19.

[14] L. Gasieniec, W. Rytter, Almost optimal fully LZW-compressed pattern matching, Data Compression Conf. Proc. (1999).

[15] Corel Stock Photo Library, Corel Corp., Ontario, Canada.

[16] V. Castelli, L.D. Bergman, Image Databases: Search and Retrieval of Digital Imagery, John Wiley and Sons, New York, 2002.

[17] Y. Rubner, J. Puzicha, C. Tomasi, J.M. Buhmann, Empirical evaluation of dissimilarity measures for color and texture, Comput. Vis. Image Understand. 84 (2001) 25–43.

[18] Y. Rubner, C. Tomasi, Perceptual Metrics for Image Database Navigation, Kluwer Academic Publishers (2001).

[19] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.

[20] C. Theoharatos, N.A. Laskaris, G. Economou, S. Fotopoulos, A generic scheme for color image retrieval based on the multivariate Wald–Wolfowitz test, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 808–819.

[21] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of the ACM International Multimedia Conference and Exhibition, 1996.

[22] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley and Sons, 2002.