Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 111 (2008) 219–228
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Content-based image retrieval with the normalized information distance

Iker Gondra *, Douglas R. Heisterkamp

Department of Mathematics, Statistics, and Computer Science, St. Francis Xavier University, P.O. Box 5000, Antigonish, NS B2G 2W5, Canada
Department of Computer Science, Oklahoma State University, Stillwater, OK 74078, USA

Received 11 July 2006; accepted 6 November 2007
Available online 21 November 2007

Abstract

The main idea of content-based image retrieval (CBIR) is to search on an image's visual content directly. Typically, features (e.g., color, shape, texture) are extracted from each image and organized into a feature vector. Retrieval is performed by image example where a query image is given as input by the user and an appropriate metric is used to find the best matches in the corresponding feature space. We attempt to bypass the feature selection step (and the metric in the corresponding feature space) by following what we believe is the logical continuation of the CBIR idea of searching visual content directly. It is based on the observation that, since ultimately, the entire visual content of an image is encoded into its raw data (i.e., the raw pixel values), in theory, it should be possible to determine image similarity based on the raw data alone. The main advantage of this approach is its simplicity in that explicit selection, extraction, and weighting of features is not needed. This work is an investigation into an image dissimilarity measure following from the theoretical foundation of the recently proposed normalized information distance (NID) [M. Li, X. Chen, X. Li, B. Ma, P. Vitányi, The similarity metric, in: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 863–872]. Approximations of the Kolmogorov complexity of an image are created by using different compression methods. Using those approximations, the NID between images is calculated and used as a metric for CBIR. The compression-based approximations to Kolmogorov complexity are shown to be valid by proving that they create statistically significant dissimilarity measures by testing them against a null hypothesis of random retrieval. Furthermore, when compared against several feature-based methods, the NID approach performed surprisingly well.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Content-based image retrieval; Normalized information distance; Kolmogorov complexity; Compression; Raw pixel data; Visual content; Similarity measure

1. Introduction

In recent years, the rapid development of information technologies and the advent of the Web have accelerated the growth of digital media and, in particular, image collections. As a result and in order to realize the full potential of these technologies, the need for effective mechanisms to search large image collections becomes evident. The traditional keyword-matching approach to image retrieval uses a textual representation based on the manual annotation of images with descriptive keywords. This is not only subjective and error-prone but also very time-consuming and cumbersome for large image collections. Recently, some approaches for automatic image labelling [41,47,52] have been proposed as an attempt to improve this manual annotation process. In [41], image recognition techniques are used for automatically assigning descriptive keywords to images. Their approach uses only a limited number of keywords. Furthermore, because image recognition techniques are not completely reliable, automatically assigned keywords still have to be verified by a human. In [47], the textual context of images in a web page is used to automatically extract descriptive keywords. The collateral text that usually accompanies an image (e.g., captions) is exploited in [52]. The performance of those approaches is not as high as that obtained with manual annotation and their applicability is limited in situations where there is no textual context (e.g., a photo album). Also, in the case

* Corresponding author. Fax: +1 902 867 3302.
E-mail address: igondra@stfx.ca (I. Gondra).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2007.11.001

of web image retrieval and unlike text-based image retrieval that uses well-organized captions [48], not all words in an HTML file are related to an image. More importantly, textual descriptions can only begin to capture the richness and complexity of an image's visual content.

To overcome these problems, content-based image retrieval (CBIR) [49] was proposed in the early 90's. The idea is to search on an image's visual content directly. Retrieval is performed by image example where a query image is given as input by the user and an appropriate metric is used to find the best matches in the corresponding feature space. In traditional approaches [11,13,24,35,37,46,51,55], each image is represented by a set of global features that are calculated by means of uniform processing over the entire image and describe its visual content (e.g., color, texture). Users usually look for particular objects when describing the semantic interpretation of an image. Thus, due to global image properties affecting the recognition of certain objects depicted in an image, low retrieval performance is often attained when using global features.

In region-based image representations [3,5,33,54], the use of local features that describe each of a set of segmented regions in an image provides a more meaningful characterization that is closer to a user's perception of an image's content. Many image segmentation algorithms have been proposed. Object (or strong) segmentation is defined as a grouping of the image pixels into regions such that each region contains all the pixels of a single physical object and nothing else. It is an extremely difficult image processing task mainly due to the fact that most segmentation algorithms use low-level data-driven properties to generate regions that are homogeneous according to some criterion. Unfortunately, it is very often the case that such regions do not correspond to meaningful units (i.e., physical objects). Some approaches (e.g., [12]) have been proposed that can learn object categories. However, due to the great difficulty of accurately segmenting an image into regions that correspond to a human's perception of an object, several approaches have been proposed [5,33,50,54] that consider all regions in an image for determining similarity. As a result, the problems of inaccurate segmentation are reduced.

Integrated region matching (IRM) [33] is proposed as a measure that allows a many-to-many region mapping relationship between two images by matching a region of one image to several regions of another image. Thus, by having a similarity measure that is a weighted sum of distances between all regions from different images, IRM is more robust to inaccurate segmentation. The image segmentation algorithm that is used in IRM first partitions an image into blocks of 4×4 pixels. Then, a feature vector $\mathbf{f} = [f_1, f_2, f_3, f_4, f_5, f_6]^{\mathrm{T}}$ representing color and texture properties is extracted for each block. The first three features are the average color components and the other three represent energy in high frequency bands of the wavelet transforms [8,39]. The k-means algorithm is then used to cluster the feature vectors into several regions. The number of regions is adaptively chosen according to a stopping criteria. A feature vector $\mathbf{h} = [h_1, h_2, h_3]^{\mathrm{T}}$ is then extracted for each region to describe its shape characteristics. The shape features are normalized inertia [16] of order 1 to 3. A region is described by $\mathcal{R} = \{\hat{\mathbf{f}}, \mathbf{h}\}$, where $\hat{\mathbf{f}}$ is the average of the feature vectors of all blocks assigned to the region. Recently, a fuzzy logic approach, unified feature matching (UFM) [5] was proposed as an improved alternative to IRM. UFM uses the same segmentation algorithm as IRM. In UFM, an image is characterized by a fuzzy feature denoting color, texture, and shape characteristics. Because fuzzy features can characterize the gradual transition between regions in an image, segmentation-related inaccuracies are implicitly considered by viewing them as blurring boundaries between segmented regions. As a result, a feature vector can belong to multiple regions with different degrees of membership as opposed to classical region representations in which a feature vector belongs to only one region. The similarity between two images is then defined as the overall similarity between two sets of fuzzy features. A fuzzy feature is defined by a membership function that measures the degree of membership of a feature vector $\mathbf{x}$ to the fuzzy feature.

A method of measuring similarity of images in a database to a query is needed when completing an image retrieval request. If the images in the database are annotated with text, then standard text-based information retrieval methods may be used. In the case of CBIR, the feature vectors are viewed as points in a space and a distance metric is used to select the points closest to the query and retrieve the corresponding images. This approach suffers from the fact that there is a large discrepancy between the low-level visual features that one can extract from an image and the semantic interpretation of the image's content that a particular user may have in a given situation. That is, users seek semantic similarity but we can only provide similarity based on low-level visual features extracted from the raw pixel data. This situation, known as the semantic gap, is exacerbated when the retrieval task is to be performed in broad image domains (e.g., the Web) where images with similar semantic interpretations may have unpredictable and large variability in their low-level visual content. In contrast, when the retrieval task is performed in narrow domains (e.g., medical images, frontal views of faces) usually there are specific assumptions particular to the application that, for a given semantic interpretation, limit the variability of its corresponding low-level visual content. As a result, it is easier to find links between low-level visual content and semantic interpretations (i.e., the semantic gap is smaller).

Relevance feedback (RF) learning has been proposed as a technique aimed at reducing the semantic gap. It works by gathering semantic information from user interaction. The simplest form of RF is to indicate which images in the retrieval set are relevant. Based on this RF, the retrieval scheme is adjusted. This process iterates until the user is satisfied with the retrieved images or stops searching. Thus,

by providing an image similarity measure under human perception, RF learning can be seen as a form of supervised learning that finds relations between high-level semantic interpretations and low-level visual properties. Hence, it attempts to reduce the semantic gap by tailoring the retrieval strategy to the narrow image domain the user has in mind. Two main RF learning strategies have been proposed: query modification and distance re-weighting. Query modification changes the representation of the query in a form that is closer to the semantic intent of the user. In particular, query shifting involves moving the query feature vector towards the region of the feature space containing relevant images. This is based on the assumption that relevant images have similar feature vectors and cluster together in feature space. Distance re-weighting changes the calculation of image-to-image similarity to strengthen the contribution of relevant image components in regard to the current query.

In [42], a probabilistic feature relevance learning algorithm that automatically captures feature relevance based on RF is presented. It computes flexible retrieval metrics for producing neighborhoods that are elongated along less relevant feature dimensions and constricted along most influential ones. In [20], we propose a probabilistic region relevance learning algorithm that can automatically (i.e., without asking the user) make informed estimates for the importance of each region in an image. Instead of updating individual weights, it is also possible to select from a predefined set of similarity measures. In [29], a Bayesian framework is used to associate each image with a probability that it corresponds to the user's semantic intent. The probability is updated based on the RF at each iteration. A re-ranking method to improve web image retrieval by reordering the images retrieved from an image search engine is proposed in [32]. The re-ranking process is based on a relevance model, which is a probabilistic model that evaluates the relevance of the HTML document linking to the image, and assigns a probability of relevance.

Recently, support vector machine (SVM) learning has been applied to CBIR systems to significantly improve retrieval performance. Basically, the probability density of relevant images can be estimated by using one-class SVMs. For instance, in [6], a one-class SVM is used to estimate the distribution of target images by fitting a tight hypersphere in a non-linearly transformed feature space. In [59], the problem is regarded as a two-class classification task and a maximum margin hyperplane in a non-linearly transformed feature space is used to separate relevant and non-relevant images. In [19], we present a short-term learning approach based on generalized SVMs that can be used with region-based (i.e., variable-length) image representations. Because a generalized SVM does not place any restrictions on the kernel, any region-based similarity measure (i.e., not necessarily an inner product one) can be used.

Most current retrieval systems that exploit RF are based on a short-term-learning-only approach. That is, the system refines the retrieval strategy by using RF supplied by the current user only and the learning process starts from ground up for each new query. Some approaches attempt long-term learning (i.e., RF from past queries are used to improve the retrieval performance of the current query). The results from those approaches show a tremendous benefit in the initial and first iteration of retrieval. Long-term learning thus offers a great potential for reducing the amount of user interaction by decreasing the number of iterations needed to satisfy a query. The method proposed in [31] was one of the first attempts to explicitly memorize learned knowledge to improve retrieval performance. A correlation network is used to accumulate semantic relevance between image clusters learned from RF. In [27,26] latent semantic analysis is used to provide a generalization of past experience. Both [7] and [58] take the approach of complete memorization of prior history. We have developed several techniques [18,21,22,17] for performing long-term learning. Those techniques use SVMs in combination with RF for learning the class distributions of users' semantic intents from retrieval experience. The geometric view of one-class SVMs allows a straightforward interpretation of the density of past interaction in a local area of the feature space and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

Many of these techniques can also be applied to video retrieval. A general framework for video retrieval consists of building a supervised classifier from sample training shots and using the classifier to find more relevant shots whose features match those of the training shots. Several important issues, such as the fusion of multimodality information [57], how to use active learning for iteratively building better semantic classifiers by selecting the most informative shots from user feedbacks [4], or how to use the temporal information of video data [10] have been explored.

Both selecting the features and adapting the distance metric continue to be active areas of research. The purpose of this research is to investigate what we believe is the logical continuation of the CBIR idea of searching visual content directly. Because ultimately, the entire visual content of an image is encoded into its raw data (i.e., the raw pixel values), in theory, it should be possible to determine image similarity based on the raw data alone. That is, everything that we need to know regarding the visual content of the image is in the raw data itself. Humans are very good at looking at an image (i.e., the raw data) and extracting all the important features from it. Thus, all the important features are "hidden" in the raw data somewhere. The problem of feature extraction is just that we do not entirely know yet what they are and how (we, humans) "find" them. Thus, instead of attempting to determine image similarity based on a small set of (probably incomplete) set of features, why not have a similarity measure that is based on the raw data itself (since everything is in the raw data).

We attempt to bypass the feature selection step (and the distance metric in the corresponding feature space) by taking the normalized information distance (NID) [34] approach. The NID approach is based on the notion of Kolmogorov complexity [30,36]. The information distance between a and b is the complexity of the transformations of a into b and b into a. The information distance is normalized by the individual complexities of a and b. In theory, the complexity of a is measured by the length of the shortest program that can compute a from scratch. The complexity of the transformation of a into b is the length of the shortest program that can compute b given a as an auxiliary input. Kolmogorov complexity is not computable, but it has been used as the foundation for the minimum description length (MDL) principle [9,45] and the minimum message length (MML) principle [53]. We investigate the application of NID to image dissimilarity measurement by approximating the complexity of an image by the size of the compressed image. In [34], NID was successfully applied to the problems of determining whole mitochondrial genome phylogenies and classifying natural languages when using a compression-based approximation of complexity. It has also been shown to be applicable to chain letters [2]. This article is a revised and expanded version of our initial investigation into this idea, put forth in [23]. The main advantage of the NID approach is its simplicity. The objective of this research is to obtain some preliminary evidence as to whether a compression-based approximation to the NID can actually create a statistically significant image similarity measure and to compare its performance against that of more traditional feature-based methods.

The rest of this article is organized as follows. Section 2 gives a brief introduction to the NID as presented in [34]. In Section 3, we explain how we approximate Kolmogorov complexities of images and apply the NID to CBIR. Experimental results with real data sets are presented in Section 4. Finally, concluding remarks are given in Section 5.

2. The normalized information distance

The NID presented in [34] is based on the incomputable notion of Kolmogorov complexity. The Kolmogorov complexity of a string x, $K(x)$, is defined as the length of the shortest effective binary description of x. Broadly speaking, $K(x)$ may be thought of as the length of the shortest program that, when run with no input, outputs x. It has been shown that, although there are many universal Turing machines (and thus many possible shortest programs), the corresponding complexities differ by at most an additive constant [15]. Thus, $K(x)$ is the smallest amount of information that is needed by an algorithm to generate x. Let x^* be the smallest program that generates x. Then, $K(x) = |x^*|$. Similarly, the conditional Kolmogorov complexity of x relative to another string y, $K(x|y)$, is the length of the shortest program that, when run with input y, outputs x. Also, $K(x,y)$ is the length of the smallest program that generates x and y along with a description of how to tell them apart. The theory and development of the notion of Kolmogorov complexity are described in detail in [36]. The *information in y about x* is defined as [30,34]

$$I(x:y) = K(x) - K(x|y^*)$$

A result from [14] shows that, up to additive constants, $I(x:y) = I(y:x)$. Thus [34],

$$K(x) + K(y|x^*) = K(y) + K(x|y^*) \tag{1}$$

The *information distance* $E(x,y)$ is defined as the length of a smallest program that generates x from y and y from x [34]. A result from [1] indicates that, up to an additive logarithmic term,

$$E(x,y) = \max\{K(y|x), K(x|y)\} \tag{2}$$

Because it is not normalized, Eq. (2) may not be an appropriate distance measure. For instance, according to Eq. (2), the distance between two very long strings that differ only in a few positions would be the same as the distance between two short strings that differ by the same amount. In [34], the NID $d(x,y)$ is proposed

$$d(x,y) = \frac{\max\{K(x|y^*), K(y|x^*)\}}{\max\{K(x), K(y)\}} \tag{3}$$

The function $d(x,y)$ is a normalized information distance (i.e., it is a distance metric, takes values in [0,1], and satisfies the normalization condition). It is also universal because it includes every computable type of similarity in the sense that, whenever two objects are similar in normalized information in some computable sense, then they are at least that similar in $d(x,y)$ sense [34]. For proofs and more details, refer to [34].

3. Applying the NID to CBIR

Computational complexity is related to the length of the shortest program that is able to perform some computation. For example, as previously discussed, the Kolmogorov complexity of a string x, $K(x)$, may be thought of as the length of the shortest program that, when run with no input, outputs x. For example, if x consists of 5 billion 1's, although very long, can be generated by a very short program (e.g., a single loop that iterates 5 billion times and outputs a 1 on each iteration). On the other hand, if x is the text of Hamlet, although much shorter than 5 billion characters, cannot be generated by any simple program and hence is much more complex. Thus, according to the Kolmogorov definition of complexity, intuitively, random strings of symbols have the greatest complexity (Hamlet can be put through a text compression algorithm that takes advantage of repeated words and grammatical structure). A raw image is a string containing byte streams describing color information. Thus, similarly and according to the Kolmogorov definition of complexity, the fewer the number of regularities found in the image, the more complex the image is. Thus, images with small color variation and/or with large areas of one color are far less com-

plex than those with large color variations and/or no such homogeneous regions (see Fig. 1).

Let x and y be two raw images (i.e., strings containing byte streams describing color information). In order to be able to use Eq. (3) for determining distance between x and y, we need to estimate $K(x)$, $K(y)$ and their conditional complexities $K(x|y)$, $K(y|x)$. For the conditional complexities, by (1), $K(x|y) = K(x,y) - K(y)$ (up to an additive constant) [34]. Also, $K(x,y) = K(xy)$ (up to additive logarithmic precision) [34].

Kolmogorov complexity is not computable but we can try to approximate it by using compression. The idea behind image compression is generally the same regardless of what compression method is used. That is, whenever there is a group of neighboring pixels of the same color, it is more efficient to use a single description for the entire region. Furthermore, with lossy compression, a group of pixels that have very similar colors can be replaced with their average color, which results in a much more compact description at the expense of slight image distortions. What is different among compression algorithms is how the regions that contain pixels of similar colors are found and described. For example, some methods use wavelets to represent the image. Thus, since compression algorithms take advantage of redundancy (i.e., spatial, color coherence) in an image to shrink the representation, they approximate the spirit of Kolmogorov complexity. That is, if x is a more complex image than y, the size of the compressed x will generally be larger than that of y. Thus, this corresponds to the intuition that $K(y)$ should be smaller than $K(x)$. Thus, the size of the compressed x is used to approximate $K(x)$, similarly for $K(y)$. We also use compression to approximate the conditional Kolmogorox complexities. The compressed size of concatenation of x with y is used to estimate $K(xy)$, similarly for $K(yx)$. We justify this by noting that, similarly, what a compressor does in order to code the concatenated xy sequence is to search for information that is shared by x and y in order to reduce the redundancy of the entire sequence. Thus, if the result is much smaller than the sum of the individual complexities, it means that a lot of information contained in x can be used to code y. If this happens, we could describe x by making references to (parts of) y and a "shorter program" would be needed to describe x. Thus, using Eq. (3), the distance between two raw images x and y can be defined as

$$d'(x,y) = \frac{\max\{(|c(xy)| - |c(y)|), (|c(yx)| - |c(x)|)\}}{\max\{|c(x)|, |c(y)|\}} \tag{4}$$

where $c(i)$ is the compressed version of input i and $|c(i)|$ is its corresponding size. Note that $|c(x)|$, $|c(xy)|$ are approximations to $K(x)$ and $K(x,y)$, respectively.

It is common for compression algorithms to exploit similarities among neighboring points. Thus, if two images contain an object of interest but in different spatial locations, simple concatenation methods (e.g., sequential or interleaving of the bytes of the two images) may not be able to exploit this (see Fig. 2). Intuitively, if the objects of interest were in the same spatial location in the two images (e.g., if the apple in the second image in Fig. 2 appeared also on the upper left hand corner), we would expect the byte-interleaving method to have better performance. Thus, for some compression algorithms, it may be advantageous to have an interleaving of the regions in the two images such that the objects of interest (e.g., the apples) appear together. We propose the following additional method. First, each image is partitioned into n blocks of equal size (see Fig. 3(a)). Note that this "fixed segmentation" avoids the need for feature extraction, which would defeat the whole purpose of the NID approach to image retrieval. Next, n interleavings of the resulting blocks are generated such that every pair of blocks from the two images appear together once (see Fig. 3(b)). Note that setting $n = 1$ results in a simple sequential concatenation and, the larger the value of n, the more likely that the objects of interest will appear next to each other (i.e., without any "background" separating them) (see Fig. 4). Ideally, all possible combinations of the image blocks (i.e., $n \times n$) and thus all possible interleaving orderings would be used instead of only n. However, for every pair of images, this would require n^2 compressions (i.e., each of the n^2 interleavings has to be compressed). As a result, due to its computational complexity, the proposed approach would not be practical. Thus, this required us to compromise accuracy to achieve a reasonable computational complexity (i.e., linear instead of quadratic) for the similarity operation between each pair of images. Lastly, the distance between the two images x and y is redefined as

$$d''(x,y) = \frac{\min\limits_{j=1,\ldots,n} \{\max\{(|c(xy_j)| - |c(y)|), (|c(yx_j)| - |c(x)|)\}\}}{\max\{|c(x)|, |c(y)|\}} \tag{5}$$

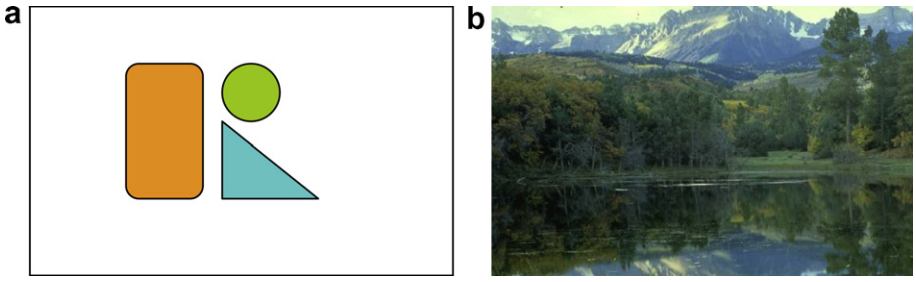

Fig. 1. The (Kolmogorov) complexity of an image is proportional to the number of regularities found in the image; (a) a simple image of a logo containing homogeneous areas in one color; (b) a more complex photographic image.

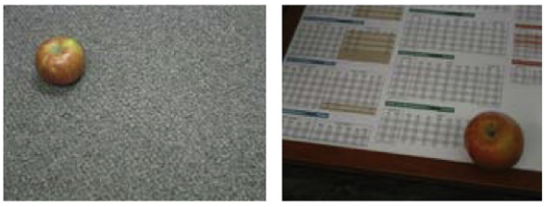

Fig. 2. In the concatenation (either sequential or byte-interleaving) of the two images, the objects of interest (i.e., the apples) appear far from each other. Sample images taken from the SIVAL benchmark (http://www.cs.wustl.edu/~sg/accio/SIVAL.html).

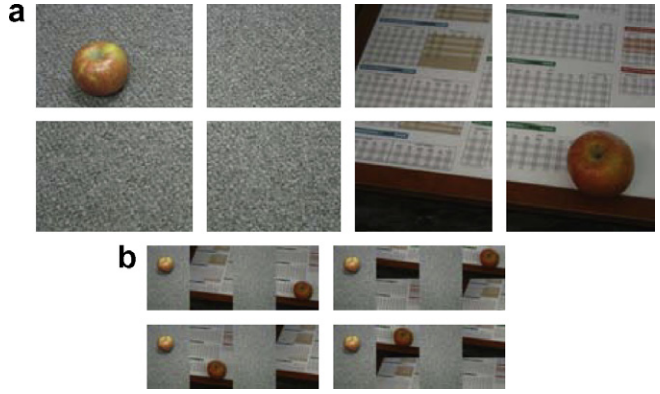

Fig. 3. Concatenation method with $n = 4$; (a) partitioning into n blocks; (b) resulting n concatenations.

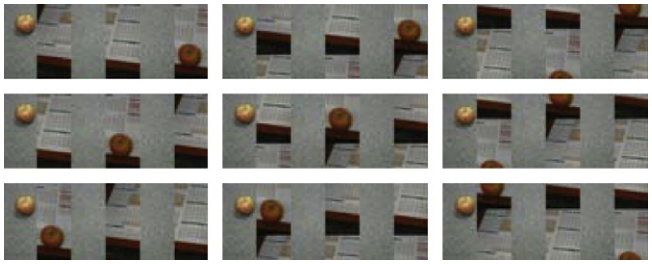

Fig. 4. Resulting concatenations with $n = 9$. Note that in one of the concatenations, the objects of interest (i.e., the apples) appear almost next to each other.

where $c(i)$ and $|c(i)|$ are as before and xy_j, yx_j refer to the jth xy, yx concatenations respectively.

4. Experimental results

Consider a database consisting of a set of images $\mathcal{D}$. Let x be a query image and $\mathcal{A} \subset \mathcal{D}$ be the subset of images in $\mathcal{D}$ that are relevant to x. After processing x, the image retrieval method generates $\mathcal{R} \subset \mathcal{D}$ as the retrieval set. Then, $\mathcal{R}^+ = \mathcal{R} \cap \mathcal{A}$ is the set of relevant images to x that appear in $\mathcal{R}$. Users want the database images to be ranked according to their relevance to x and then be presented with only the k most relevant images so that $|\mathcal{R}| = k < |\mathcal{D}|$. Thus, images are ranked by their distance to the query image and, in order to account for the quality of image rankings, precision at a cut-off point (e.g., k) is commonly used. Thus, the performance of the image retrieval method is commonly measured by *precision*, which quantifies the ability to retrieve only relevant images and is defined as $\textit{precision} := \frac{|\mathcal{R}^+|}{|\mathcal{R}|}$. For example, if $k = 20$ and the top 20 ranked images are all relevant to x, then $\mathcal{R}$ contains only relevant images and thus precision is 1. On the other hand, if $k = 40$ and only the first top 20 images are all relevant to x, then half of the images in $\mathcal{R}$ are non-relevant to x and thus precision is only 0.5.

The objective of our experiments was to obtain evidence as to whether a compression-based approximation to the NID actually creates a statistically significant image similarity measurement. Therefore, we tested its performance against an uninformed method that used uniform random retrieval to select the images in $\mathcal{R}$. We also compared the performance of the NID approach against that of several feature-based methods. The following four real-world data sets were used for evaluation:

(1) *Texture*. The Texture data set, obtained from MIT Media Lab [44]. There are 40 different texture images that are manually classified into 15 classes. Each of those images is then cut into 16 non-overlapping images of size 128×128. Thus, there are 640 images in the database. Sample images are shown in Fig. 5.
(2) *Letter*. The Letter data set, obtained from the UCI repository of machine learning databases [38]. It consists of 20,000 rectangular pixel displays of the 26 capital letters in the English alphabet. The character images are based on 20 different fonts and each letter within these 20 fonts is randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus is converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 to 15. Sample images are shown in Fig. 6.
(3) *GroundTruth*. The University of Washington GroundTruth image database [61]. The images are photographs of different regions and topics. Sample images are shown in Fig. 7. We use the set of 675 annotated images. Each image contains multiple annotations (i.e., keywords).
(4) *IAPR-12*. The benchmark database and standard queries from technical committee 12 of IAPR [28]. The data consists of 1000 images and 30 standard queries. Sample images from the queries can be found in Fig. 8.
(5) *Corel*. A subset of 2000 labelled images from the general purpose COREL image database. There are 20 image categories, each containing 100 pictures. Sample images are shown in Fig. 9.

The *Texture* and *Letter* data sets were used first. For this experiment, we used $d'(x, y)$ Eq. (4) and compression algorithms from the UCL library [40], a portable lossless data compression library written in ANSI C. The compression algorithms included in the UCL library are block compres-

Fig. 5. Sample images from *Texture* data set.

Fig. 6. Sample images from *Letter* data set. First row contains images of the letter "O"; images on the second row are of the letter "Q".

Fig. 7. Sample images from *GroundTruth* data set.

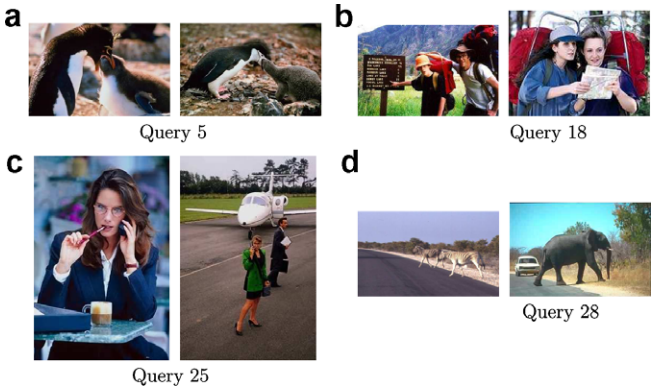

Fig. 8. Sample query images from *IAPR-12* data set; (a) query 5, (b) query 18, (c) query 25, (d) query 28.

sors (i.e., each memory block passed to the compressor gets compressed independently). Each block of data is compressed into matches (a sliding dictionary) and runs of non-matching literals. The algorithm exploits long matches and long literal runs so that it produces good compression results on highly redundant data. The image concatenation was a sequential placement of the raw bytes of the second image at the end of the first image. Each image was used as a query and the precision of a retrieval set of the 20 nearest images was measured. The results are presented in Tables 1 and 2, which show the average precision over the 640 and 20,000 queries respectively. The NID approach performed surprisingly well and is obviously statistically different than uniform random retrieval. It performs almost as well as extracting a 16-dimensional feature vector (Gabor filters in the case of *Texture*, statistical moments and edge counts in the case of *Letter*) from each image and using Euclidean distance to select the points closest to the query. Since the texture images contain the repeating patterns of the texture and the letter images are very simple black-and-white images, they are probably the best case situation for approximation based on compression.

The *GroundTruth* data set was used next. For this experiment, we used $d'(x,y)$ Eq. (4), $d''(x,y)$ Eq. (5), and gzip [25] as the compressor. The deflation algorithm used by gzip is a variation of LZ77 [60]. It finds repeated strings in the input data. The second occurrence of a string is replaced by a pointer to the previous string, in the form of a (distance, length) pair. When a string does not occur anywhere

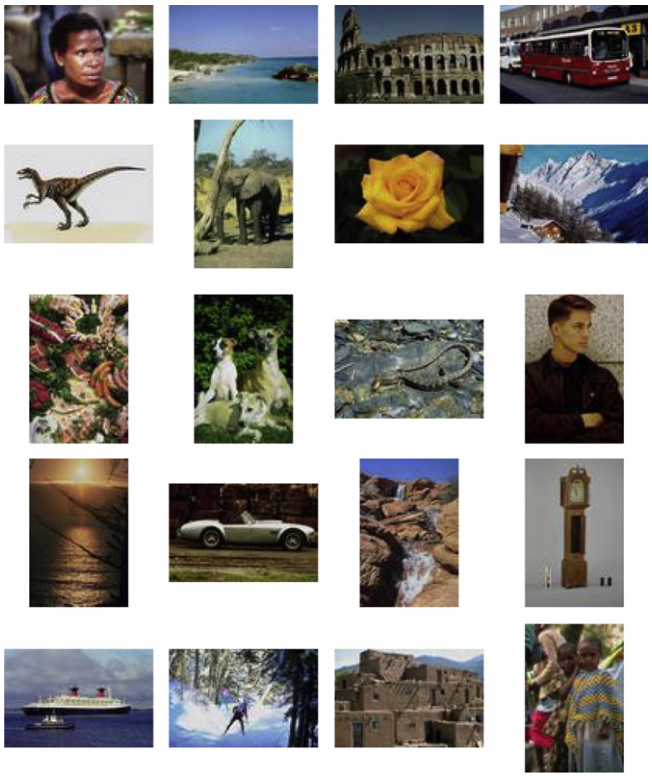

Fig. 9. Sample images from *Corel* data set.

in the previous 32K bytes, it is emitted as a sequence of literal bytes. Literals or match lengths are compressed with one Huffman tree, and match distances are compressed with another tree. The trees are stored in a compact form at the start of each block [25]. In the case of $d'(x,y)$ Eq. (4), the image concatenation was a sequential placement of the raw bytes of the second image at the end of the first image and, for $d''(x,y)$ Eq. (5), $n = 9$. Each image was used as a query and the precision of a retrieval set of the 20 nearest images was measured. We define y as being relevant to x when x and y share at least one common annotation. The results are presented in Table 3, which shows the average precision over the 675 queries. The NID approach with $d'(x,y)$ Eq. (4) had a precision of 0.578 and a precision of 0.602 with $d''(x,y)$ Eq. (5). The random method has a precision of 0.414. To determine if the NID approach with $d'(x,y)$ Eq. (4) is statistically different from the random method, McNemar's test [56] was used. In McNemar's test for two classifiers, A and B, the z statistic is

$$z = \frac{|n_{01} - n_{10}| - 1}{\sqrt{n_{10} + n_{01}}}$$

where n_{01} is the number of samples misclassified by A but not by B and n_{10} is the number of samples misclassified by B but not by A. In this case, $n_{01} = 2358$ and $n_{10} = 4572$ out of a total of 13,500 classified samples (20 for each of the 675 images) and $z = 26.58$. The quantity z^2 is distributed approximately as χ^2 with one degree of freedom. Thus we can reject the null hypothesis that the classifiers have the same error rate and assert that the NID is expressing a statistically significant similarity measure.

Table 1
Texture data set performance

	Random	NID	Gabor filters
Precision at 20 images	0.079	0.80	0.81

Table 2
Letter data set performance

	Random	NID	Moments, edge counts
Precision at 20 images	0.038	0.82	0.849

Table 3
GroundTruth data set performance

	Random	NID Eq. (4)	NID Eq. (5)
Precision at 20 images	0.414	0.578	0.602

The *IAPR-12* data set was used next. We used the queries that contained two images (queries 5, 18, 20, 21, 25, 26, and 28). Each image was used as a query image and the rank of the other image was determined by sorting the images based on distance from the query. For this experiment, we used $d'(x,y)$ Eq. (4) and compression algorithms from the UCL library [40]. Two methods of image concatenation were tried. In addition to the previous sequential concatenation, an interleaving of the two images was done by alternating the bytes from the two images. The sequential concatenation performed well on query 18 (Fig. 8(b)) with the desired retrieval image ranking first, but on query 25 (Fig. 8(c)) the desired image had rank 926. Over all of the queries, the average rank of the desired image was 501 and not different than random retrieval (which would average 499.5). Switching the concatenation to an interleaving approach improved the average rank to 395 but actually pushed the worst result from query 25 out to rank 981. Though the approach worked very well on some of the individual queries, further investigation of the IAPR data set is needed due to the difficulty of some the queries.

The *Corel* data set was used next. For this experiment, we used $d'(x,y)$ Eq. (4) and JPEG compression [43]. JPEG is a lossy compression algorithm that uses transform coding. First, the image is subdivided into blocks of 8×8 pixels. Then, a conversion to the frequency domain is performed by applying a two-dimensional discrete cosine transform (DCT) to each block. The results of psychophysical experiments suggest that the human eye is not so sensitive to high frequency brightness variation. Thus, the amount of information contained in the high frequency components can be greatly reduced without humans being able to perceive any significant difference in the image. Therefore, the next step is a quantization step in which each component in the frequency domain is divided by a constant for that component and then rounded to the nearest integer. This is the main lossy step in the algorithm. The results of this quantization are then encoded by using a

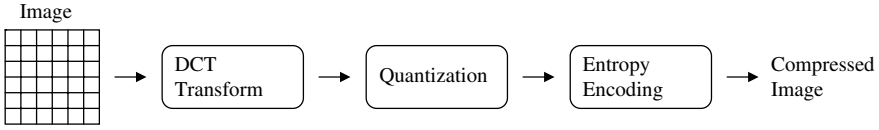

Fig. 10. JPEG compression.

Table 4
Corel data set performance

	Random	NID	UFM	IRM
Precision at 20 images	0.05	0.331	0.466	0.275

special form of lossless data compression known as entropy encoding. This involves arranging the quantized coefficients in a zig-zag order that groups similar frequencies together and then using Huffman coding [43]. Fig. 10 shows the main steps of JPEG compression.

The image concatenation was a sequential placement of the quantized coefficients (resulting from the quantization step) of the second image at the end of the quantized coefficients of the first image. Then, the entropy encoding step was performed on the concatenated coefficients. Note that, in the quantization step, frequency components from both images that are close enough will be rounded to the same nearest integer (i.e., to the same quantized coefficient). Thus, the entropy encoder step will exploit not only redundancies between the two images but also implicitly, similarities between them. Each image was used as a query and the precision of a retrieval set of twenty nearest images was measured. The results are presented in Table 4, which shows the average precision over the 2000 queries. Once again, the NID performed surprisingly well and is obviously statistically different than the random approach. It performs better than IRM [33] and not much worse than UFM [5], both described in 1.

5. Summary and future research

Based on the observation that the entire visual content of an image is encoded into its raw data, we investigated the possibility of using the NID [34] to bypass the feature selection step (and the distance metric in the corresponding feature space). The NID is a universal dissimilarity measure and, although the measure is not computable and not even effectively approximable, it does provide insight into what we would want to do in the ideal case. This insight can be used to guide our attempts at simulating the NID measure at various levels of precision. In this article, we determined that even simple compression-based approximations to Kolmogorov complexity resulted in statistically significant dissimilarity measures for images when the NID approach was followed. Furthermore, this method performed surprisingly well when compared against some feature-based approaches. This is an encouraging result that indicates that other attempts at simulating NID may yield good results. Another area where it may be useful to try the NID approach is in the matching of variable-length feature vectors. The NID approach may create a very practical method that goes beyond the individual region matching but does not require the expense of determining the higher level relationships among the regions. Another area of future research is the exploration of the NID approach as a feature-independent method of structuring an image data set.

References

[1] C. Bennett, P. Gács, M. Li, P. Vitányi, W. Zurek, Information distance, IEEE Transactions on Information Theory 44 (4) (1998) 1407–1423.
[2] C. Bennett, M. Li, B. Ma, Linking chain letters, Scientific American (2003) 76–81.
[3] C. Carson, Blobworld: Image segmentation using expectation-maximization and its applications to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1026–1038.
[4] M. Chen, M. Christel, A. Hauptmann, H. Wactlar, Putting active learning into multimedia applications: dynamic definition and refinement of concept classifiers, in: Proceedings of ACM International Conference on Multimedia, 2005, pp. 902–911.
[5] Y. Chen, J.Z. Wang, A region-based fuzzy feature matching approach to content-based image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (9) (2002) 1252–1267.
[6] Y. Chen, X. Zhou, T. Huang, One-class SVM for learning in image retrieval, in: Proceedings of IEEE International Conference on Image Processing, 2001, pp. 34–37.
[7] J. Cox, M.L. Miller, T.P. Minka, P.N. Yianilos, An optimized interaction strategy for Bayesian relevance feedback, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 553–558.
[8] I. Daubechies, Ten Lectures on Wavelets, Capital City Press, 1992.
[9] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley and Sons, New York, NY, 2001.
[10] S. Ebadollahi, L. Xie, S.-F. Chang, J. Smith, Visual event detection using multi-dimensional concept dynamics, in: Proceedings of IEEE International Conference on Multimedia and Expo, 2006, pp. 881–884.
[11] C. Faloutsos, M. Flicker, W. Niblack, D. Petkovic, W. Equitz, R. Barber, Efficient and effective querying by image content, Technical report, IBM, 1993.
[12] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from google's image search, in: Proceedings of IEEE International Conference on Computer Vision, 2005, pp. 1816–1823.
[13] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, The QBIC project: querying images by content using color, texture, and shape, in: Proceedings of SPIE Storage and Retrieval for Image and Video Databases, 1993, pp. 173–181.
[14] P. Gács, On the symmetry of algorithmic information, Soviet Mathematics Doklady 15 (1974) 1477–1480 .
[15] A. Gammerman, V. Vovk, Kolmogorov complexity: Sources, theory, and applications, The Computer Journal 42 (4) (1999) 252–255.

[16] A. Gersho, Asymptotically optimum block quantization, IEEE Transactions on Information Theory IT-25 (4) (1979) 231–262.
[17] I. Gondra, D.R. Heisterkamp, J. Peng, Improving the initial image retrieval set by inter-query learning with one-class support vector machines, in: Proceedings of International Conference on Intelligent Systems Design and Applications, 2003, pp. 393–402.
[18] I. Gondra, D.R. Heisterkamp, Improving image retrieval performance by inter-query learning with one-class support vector machines, Neural Computing and Applications 13 (2) (2004) 130–139.
[19] I. Gondra, D.R. Heisterkamp, Learning in region-based image retrieval with generalized support vector machines, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2004.
[20] I. Gondra, D.R. Heisterkamp, Probabilistic region relevance learning for content-based image retrieval, in: Proceedings of International Conference on Imaging Science, Systems, and Technology, 2004, pp. 434–440.
[21] I. Gondra, D.R. Heisterkamp, Semantic similarity for adaptive exploitation of inter-query learning, in: Proceedings of International Conference on Computing, Communications, and Control Technologies, vol. 1, 2004, pp. 142–147.
[22] I. Gondra, D.R. Heisterkamp, Summarizing inter-query knowledge in content-based image retrieval via incremental semantic clustering, in: Proceedings of IEEE International Conference on Information Technology, vol. 2, 2004, pp. 18–22.
[23] I. Gondra, D.R. Heisterkamp, A Kolmogorov complexity-based normalized information distance for image retrieval, in: Proceedings of International Conference on Imaging Science, Systems, and Technology: Computer Graphics, 2005, pp. 3–7.
[24] A. Gupta, R. Jain, Visual information retrieval, Communications of the ACM 40 (5) (1997) 70–79.
[25] gzip. GNU zip compression utility. http://www.gzip.org.
[26] D.R. Heisterkamp, Building a latent semantic index of an image database from patterns of relevance feedback, in: Proceedings of the 16th International Conference on Pattern Recognition, August 2002, pp. 132–135.
[27] X. He, O. King, W. Ma, M. Li, H. Zhang, Learning a semantic space from user's relevance feedback for image retrieval, IEEE Transactions on Circuits and Systems for Video Technology 13 (1) (2003) 39–48.
[28] IAPR. IAPR's technical committee 12: Multimedia and visual information systems, benchmarking for visual information retrieval. http://sci.vu.edu.au/clement/tc-12/benchmark.
[29] J. Ingemar, J. Cox, The Bayesian image retrieval system, PicHunter, theory, implementation, and psychological experiments, IEEE Transactions on Image Processing 9 (2000) 20–37.
[30] A. Kolmogorov, Logical basis for information theory and probability theory, IEEE Transactions on Information Theory 14 (1968) 662–664.
[31] C. Lee, W.Y. Ma, H.J. Zhang, Information embedding based on user's relevance feedback for image retrieval, in: Proceedings of SPIE International Conference on Multimedia Storage and Archiving Systems, vol. 4, 1999, pp. 19–22.
[32] W. Lin, R. Jin, A. Hauptmann, Web image retrieval re-ranking with relevance model, in: Proceedings of the IEEE/WIC International Conference on Web Intelligence, 2003, pp. 242–248.
[33] J. Li, J. Wang, G. Wiederhold, IRM: Integrated region matching for image retrieval, in: Proceedings of the 8th ACM International Conference on Multimedia, 2000, pp. 147–156.
[34] M. Li, X. Chen, X. Li, B. Ma, P. Vitányi, The similarity metric, in: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 863–872.
[35] M. Li, Z. Chen, H. Zhang, Statistical correlation analysis in image retrieval, Pattern Recognition 35 (12) (2002) 2687–2693.
[36] M. Li, P. Vitányi, An introduction to Kolmogorov complexity and its applications, Springer, New York, NY, 1997.
[37] S. Mehrotra, Y. Rui, M. Ortega, T. Huang, Supporting content-based queries over images in MARS, in: Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1997, pp. 632–633.
[38] C. Merz and P. Murphy, UCI repository of machine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html.
[39] Y. Meyer, Wavelets Algorithms and Applications, SIAM, Philadelphia, 1993.
[40] M. Oberhumer, UCL compression library, version 1.02. http://www.oberhumer.com/opensource/ucl.
[41] A. Ono, M. Amano, M. Hakaridani, T. Satoh, M. Sakauchi, A flexible content-based image retrieval system with combined scene description keywords, in: Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1996, pp. 201–208.
[42] J. Peng, B. Bhanu, S. Qing, Probabilistic feature relevance learning for content-based image retrieval, Computer Vision and Image Understanding 75 (1/2) (1999) 150–164.
[43] W.B. Pennebaker, J.L. Mitchell, The JPEG still image data compression standard, Van Nostrand Reinhold, New York, 1993.
[44] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, L. Campbell, MIT media lab: Vision texture database. http://vismod.media.mit.edu/vismod/imagery/VisionTexture/.
[45] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[46] S. Sclaroff, L. Taycher, M.L. Cascia, ImageRover: a content-based image browser for the world-wide web, Technical Report 97-005, CS Dept., Boston University, 1997.
[47] H.T. Shen, B.C. Ooi, K.L. Tan. Giving meanings to WWW images, in: Proceedings of ACM Multimedia, 2000, pp. 39–48.
[48] A.F. Smeaton, I. Quigley, Experiments on using semantic distances between words in image caption retrieval, in: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 174–180.
[49] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.
[50] J.R. Smith, C.S. Li, Image classification and querying using composite region templates, Computer Vision and Image Understanding 75 (12) (1999) 165–174.
[51] J. Smith, S. Chang, VisualSEEk: a fully automated content-based image query system, in: Proceedings of ACM Multimedia, 1996, pp. 87–98, 1996.
[52] R.K. Srihari, Z. Zhang, A. Rao, Intelligent indexing and semantic retrieval of multimodal documents, Information Retrieval (2) (2000) 245–275.
[53] C. Wallace, D. Boulton, An information measure for classification, Computer Journal 11 (1968) 185–195.
[54] J. Wang, G. Li, G. Wiederhold, Simplicity: Semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 947–963.
[55] J. Wang, G. Wiederhold, O. Firschein, X. Sha, Content-based image indexing and searching using Daubechies' wavelets, International Journal of Digital Libraries 1 (4) (1998) 311–328.
[56] A. Webb, Statistical Pattern Recognition, second ed., John Wiley and Sons, Ltd., Hoboken, NJ, 2002.
[57] R. Yan, J. Yang, A.G. Hauptmann, Learning query-class dependent weights in automatic video retrieval, in: Proceedings of ACM International Conference on Multimedia, 2004, pp. 548–555.
[58] P. Yin, B. Bhanu, K. Chang, Improving retrieval performance by long-term relevance information, in: Proceedings of the 16th International Conference on Pattern Recognition, 2002, pp. 533–536.
[59] C. Zhang, T. Chen, An active learning framework for content-based information retrieval, IEEE Transactions on Multimedia 4 (2) (2002) 260–268.
[60] J. Ziv, A. Lempel, A universal algorithm for sequential data compression, IEEE Transactions on Information Theory 23 (3) (1997) 337–343.
[61] University of Washington, groundtruth image database. http://www.cs.washington.edu/research/imagedatabase/groundtruth.