Image Retrieval using Compression-based Techniques

Daniele Cerra and Mihai Datcu

***Abstract*— A new method for semantic retrieval of color images employing data compression is presented. While typical content-based image retrieval systems operate in some parameter space, the introduced data-driven technique uses as features the very image data, represented as sets of recurring patterns collected in dictionaries. In a first offline step, the images are quantized in the Hue Saturation Value space and converted into strings, after being modified to preserve some vertical information in the process, and representative dictionaries are extracted from each string with a data compression algorithm; subsequently, the dictionaries are matched in pairs and the distance between each couple of them is estimated. On the basis of the computed distances, the system enables the user to retrieve images with similar content to a given query image. Compression-based classification techniques, being data-driven, have the drawback of being computationally intensive and have been applied, in the general case, to restricted datasets; instead, the low-complexity solutions employed in this work allow applying this similarity measure on larger datasets, keeping at the same time the desirable parameter-free approach which is characteristic of these methods. Experiments show that the proposed technique outperforms similar recent concepts based on Vector Quantization.**

***Index Terms*—Compression, similarity measure, information retrieval, vector quantization, pattern matching.**

I. Introduction

In the digital era existing database technology, usually dependent on structured texts and metadata, face difficult challenges when handling multimedia data: the lack of natural language descriptors for images, video and audio datasets has generated a great interest for alternative solutions in information retrieval. In the case of images, systems enabling queries based on the actual images content have been described: usually, they focus on a lower level of descriptors, in the form of parameters representing the direct data content (typically color histograms, layouts, and/or shapes) [1-3]; the user is able to present to the system a query image, and retrieve images which are similar, according to given criteria. In recent years, new techniques employing Vector Quantization (VQ) have been defined. Of particular interest is the Minimum Distortion Image Retrieval (MDIR) by Jeong and Gray, which fits to the training data Gaussian Mixture Models later used to encode the query features and to compute the overall distortion, outperforming previous techniques based on histogram matching [4] [5]. Daptardar and Storer introduced then a similar approach using VQ codebooks and mean squared error (MSE) distortion [6], refined later by decoupling to some degree spectral and spatial information, training separate codebooks in different regions of the images, outperforming in turn MDIR [7]: we refer to their methodology as Jointly Trained Codebooks (JTC).

This work takes a further step down in selecting the right descriptors for an image, choosing the full image data: the similarities between individual images are computed through data-compression by considering the information shared by each couple of objects. This new content-based image retrieval methodology comes from the use, introduced in recent years, of general data compression algorithms at the core of similarity measures, with its most widely known notion being the Normalized Compression Distance (NCD) [8]; the main advantage of compression-based applications is their applicability to general data with a basically parameter-free approach [9].

The proposed method's workflow is the following. In a first offline step, the images are quantized in the Hue Saturation Value (HSV) space and converted into strings, after being modified to preserve some vertical information in the process; subsequently, representative dictionaries are extracted from each object and the similarities between individual images are computed by comparing each couple of dictionaries. Compression-based methods are reinterpreted to avoid their most problematic drawback for applications on medium-to-large datasets: the computational complexity, coming from their data-driven nature. Experiments performed on the COREL image dataset [10] [5] show that the proposed method outperforms its predecessors based on VQ.

The paper is organized as follows. Section II introduces the compression concepts to the base of the proposed methods; section III describes the encoding of images in strings and the system's content-based retrieval system; section IV presents the final results; we conclude in section V.

II. Preliminaries

A. Compression-based Similarity Measures

The most widely known and used compression based

D. Cerra is with the German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: daniele.cerra@dlr.de).
M. Datcu is with the German Aerospace Agency (DLR), 82234 Wessling, Germany, and also with Télécom Paris Tech, 75013 Paris, France (e-mail: mihai.datcu@dlr.de).

similarity measure for general data is the Normalized Compression Distance (NCD), proposed by Li et al. [8]. The NCD derives from the notion of the Kolmogorov complexity $K(x)$ of an object x, which quantifies how difficult it is to compute or describe x [11]: the quantity $K(x)$ is incomputable in se, but can be approximated by compression algorithms and on its basis the *NCD* is defined for any two objects x and y as:

$$NCD(x,y) = \frac{C(x,y) - \max\{C(x), C(y)\}}{\min\{C(x), C(y)\}} , \quad (1)$$

where $C(x)$ represents the size of the (lossless) compressed version of x, while $C(x,y)$ is the compressed version of x appended to y. In plain English, the idea is that if x and y share common information, they will compress better together than separately, since the compressor will be able to reuse the recurring patterns found in one of them to more efficiently compress the other. The *NCD* and its variants can be explicitly computed between any two strings or files x and y, facilitating the use of this quantity in applications to diverse data types with a basically parameter-free approach [9] [12].

In general, there is an aspect of compression-based methodologies which has been seldom properly addressed: the difficulty in applying these techniques to large datasets. Usually, the data-driven approach of these methods requires iterated processing of the full data, not allowing compact representations in any explicit parameter space; therefore, all experiments presented so far using these techniques have been performed, whenever the computation of a full distance matrix is required, on restricted datasets seldom containing more than 100 objects (see e.g. [9] [12] [13]).

In [14] a solution is brought forward by suggesting a Support Vector Machine (SVM) based classification, where distances with representative objects of each class, which are chosen as anchors, form in a feature vector which is then used as an input for the SVM. Nevertheless, this solution introduces undesired subjective choices, such as choosing the right anchors, and its results would then be based on a partial analysis of the dataset: this would be a drawback especially for the problem of image retrieval in large databases, where a decision has to be taken for each object in the set.

Another compression-based technique seems more apt to be exploited for these means: the Pattern Representation based on Data Compression (*PRDC*), a classification methodology introduced by Watanabe et al. [15] independently from the *NCD*; a direct link between these two concepts is being established [16]. The idea to the basis of *PRDC* is to extract offline typical dictionaries, obtained with a compressor belonging to the LZ family [17], directly from the data previously encoded into strings; these dictionaries are later used to compress other files in order to discover similarities with a specific object on the basis of the dictionaries compression power. For two strings x and y *PRDC* is usually faster than *NCD*, since the joint compression of x and y which is the most computationally intensive step is avoided [16]; furthermore, the dictionaries can be extracted offline, saving additional online processing time. Nevertheless, results obtained by the former technique are not as accurate as the ones obtained by applying the latter [18].

B. Vector Quantization

Vector Quantization is used to compress data in a lossy way, usually dividing a set of points (or vectors) in clusters having roughly the same dimensions [19]. The simplest quantization method is the Uniform Quantization (UQ), a kind of scalar quantization in which (for the case of images) the RGB values are divided in levels having the same space. In this work simple UQ has been used.

III. Color Image Retrieval

A. 1 dimensional encoding

Before extracting the dictionaries and computing the distance between the images, it is needed to assign a single value to each pixel and convert the 2D image in a 1D string. An UQ of the color space is performed to avoid a full representation of the RGB color space, since 256 values are available for each color channel and the size of the alphabet would have a size of 256^3, clearly not practical for our purposes.

Since the RGB channels are correlated, the Hue Saturation Value (HSV) is chosen as color space, in order to have a more meaningful and less redundant representation. In the HSV color space a finer quantization of *hue* is recommended with respect to *saturation* and *intensity*, since the human visual perception is more sensitive to changes in the former [20]: in our experiment we used 16 levels of quantization for *hue*, and 4 for both the *saturation* and *value* components, as in [4]. Therefore, the HSV color space is quantized in 8 bits, which allow a representation with $16 \times 4 \times 4 = 256$ values.

The images are going to be converted into strings before being compressed, and traversing the image in raster order would mean a total loss of its vertical information content. Therefore we choose to represent a pixel with 9 bits, adding an extra bit for the basic vertical information, assigning 0 to smooth and 1 to rough transitions of a pixel with respect to its adjacent vertical neighbours: this information may be regarded as a basic texture information, and is needed only for the vertical direction, being implicit in the horizontal one. For a pixel p at row i and column j, the value of the bit related to the vertical information $v_{i,j}$ is given by the following equation:

$$v(p_{i,j}) = \begin{cases} 1, & if \quad (d(p_{i,j}, p_{i+1,j}) > t) \,||\, (d(p_{i,j}, p_{i-1,j}) > t) \\ 0, & otherwise \end{cases} , \quad (2)$$

where

$$d(p1, p2) = \sqrt{\left\|h_{p1} - h_{p2}\right\|^2 + \left\|s_{p1} - s_{p2}\right\|^2 + \left\|i_{p1} - i_{p2}\right\|^2} , \quad (3)$$

t is a threshold comprised between 0 and 1, and h_p, s_p

and i_p are respectively the hue, saturation and intensity values of p. In other words, it is simply checked whether the $L2$-norm of the differences in the HSV space between a pixel and its neighbors in the same column and in the two adjacent rows is above a given threshold. In the experiments in section IV a threshold of 0.4, which splits the data in two sets of comparable cardinality, has been manually chosen. Each image x goes through the above steps of data preparation, and is then converted into a string sx by recurring the image in raster order.

B. Dictionary Distance

Taking as a starting point the considerations made in section II, a step forward is taken by combining the speed of *PRDC* without skipping the joint compression step which yields better performance with *NCD*. The idea, inspired by the experiments of Cucu-Dumitrescu [21], is the following: a dictionary $D(sx)$ is extracted in linear time with the LZW algorithm [22] from each image converted to string sx, and sorted in ascending order: the sorting is performed to enable the binary search of each pattern within $D(sx)$ in time $O(log\mathrm{N})$, where N is the size of $D(sx)$. The dictionary is then stored for future use: this procedure may be carried out offline and has to be performed only once for each data instance.

Whenever a query image q is then checked against a database containing n dictionaries, a dictionary $D(q)$ is extracted from q, and then only $D(q)$ is matched against each of the n dictionaries. We define the Dictionary Distance (DD) between q and an object i represented by $D(i)$ as:

$$DD(q,i) = \frac{|D(q)| - \bigcap(D(q), D(i))}{|D(q)|} \qquad (4)$$

where $|D(q)|$ and $|D(i)|$ are the sizes of the relative dictionaries, and $\bigcap(D(q), D(i))$ is the number of patterns which are found in both dictionaries. The $DD(x,y)$ ranges for every x and y from 0 to 1, representing minimum and maximum distance, respectively, and if $x = y$, then $DD(x,y)=0$. Every matched pattern counts as 1 regardless of its length: the difference in size between the matched dictionary entries is balanced by LZW's prefix-closure property which applies to the patterns contained in the dictionary: so, a long pattern p common to $D(x)$ and $D(y)$ will naturally be counted $|p|$-1 times, where $|p|$ is the size of p.

The intersection between dictionaries represents the joint compression step performed in *NCD*, since the patterns in both the objects are taken into account (instead of just applying those contained in one for the compression of the other), and requires less computational resources: to compute (4) a number of operations proportional to $N1\log N2$ is required, where $N1$ and $N2$ are the number of entries in the two dictionaries, typically well below the total number of pixels in the image. Furthermore, since the entries in the dictionary are ordered and have the prefix-closure property, patterns extending some other pattern which was not been found in both dictionaries may be directly skipped in the process. Therefore, the computational complexity decreases with respect to techniques which need to process the full data of the two objects to compute each distance, such as typical solutions relying on *NCD*; this is done to expense of the generality of the latter, directly applicable to general data without a previous step of encoding into strings.

If it is desired to retrieve images in the database which are similar to the query image q, one may apply a simple threshold to the DD between q and any object in the dataset and retrieve all the images within the chosen degree of similarity. A sketch of the workflow is depicted in fig. 1

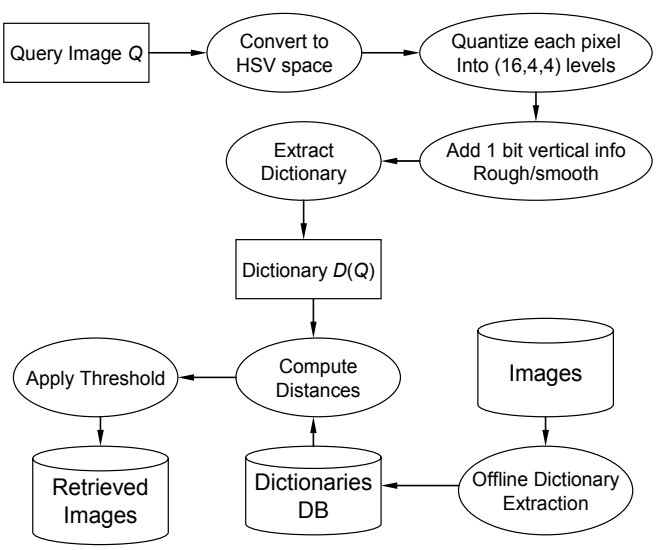

Fig. 1. Workflow for the dictionary-based retrieval system. After preprocessing, a query image Q generates a dictionary which is then compared to other dictionaries previously extracted from all the data instances.

Fig. 2. Dataset sample of each of the 15 classes in raster order (2 images per class): *Africans, Beach, Architecture, Elephants, Flowers, Horses, Caves, Postcards, Sunsets, Buses, Dinosaurs, Tigers, Mountains, Foods,* and *Women.*

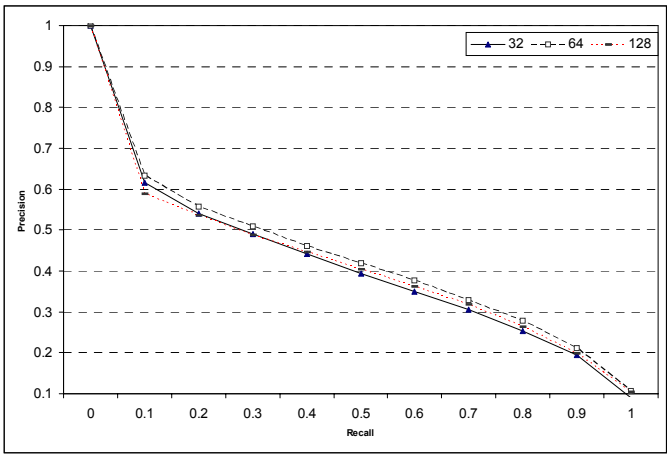

Fig. 3. Precision vs. Recall for different sizes of the images. The best performance is given for an image size of 64x64 pixels.

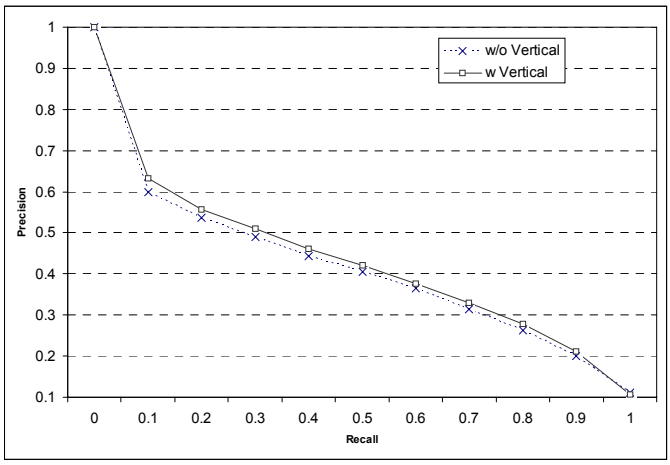

Fig. 5. Precision vs. Recall with and without addition of the bit representing vertical information. Results are improved in spite of the fact that the representation space for the pixels results doubled.

IV. Results

In the following experiments we used a subset of the COREL dataset, for a total of 1500 images equally divided in 15 classes, of which a sample is reported in Fig. 2.

A. Image retrieval

The same set of 210 query images used previously in [7] and [5] has been used in the following experiments to compute the Precision vs. Recall curves, where Precision is the number of relevant documents retrieved by a query divided by the total number of documents retrieved, and Recall is the number of relevant documents retrieved divided by the total number of relevant documents [23]. All images of original size 256x256 have been resampled to different resolutions, from 128x128 to 32x32: this has been done considering works like [24], where it is empirically shown that for a typical 256x256 image representing a full scene (so of the same size of the data contained in the COREL dataset) is usually enough for a human to analyze its 32x32 subsampled version to understand the images semantics and distinguish almost every object within; also in [25] is hinted that an image at lower resolution does not lose information as much as one would expect. In our experiments we then compared the results for the same images with sizes of 128x128, 64x64 and 32x32 pixels. The best results have been obtained with the 64x64 images, with fig. 3 showing the difference in performance when adopting a different image size.

Fig. 4 reports the main result of this work, which is a comparison of the DD with the previous VQ-based methods GMM-MDIR and JTC: for values of recall bigger than 0.2, the DD outperforms the previous techniques. As in the case of JTC, where a codebook taking into account positions within the images is adopted, also for DD the inclusion of vertical spatial information (where the horizontal is implicit) improves

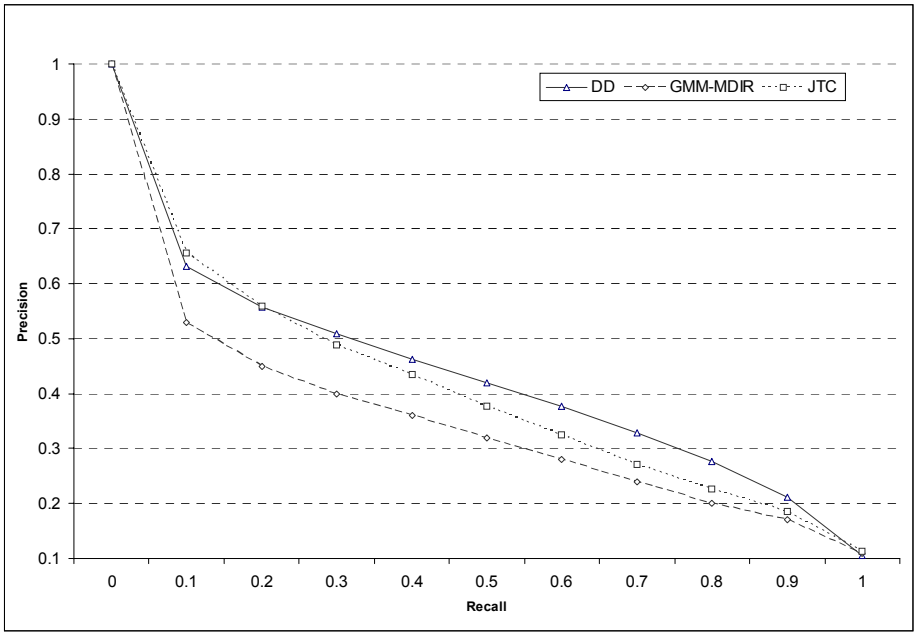

Fig. 4. Precision vs. Recall comparing MDIR and JTC with the proposed method, where the Dictionary Distance DD is used after representing the pixel values in the HSV space, adding an extra bit for the vertical information and performing scalar quantization.

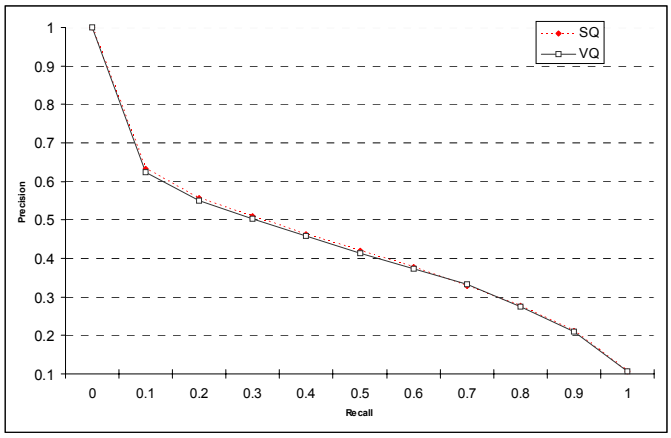

Fig. 6. Precision vs. Recall for scalar quantization and vector quantization. The VQ, performed on the basis of 24 training images, does not improve the results. In addition, it would require a new computation of the vector quantizer if new semantic classes are added to the dataset.

the results obtained. The difference in performance when the vertical information is considered is reported in fig. 5: the improvement is not dramatic but constant, and the computational simplicity of the algorithm employed justifies the use of this extra information. In addition to the simple UQ, more refined VQ has been also tested: the training vectors have been computed on the basis of 24 training images, but this representation did not improve the results (see fig. 6). In addition, adopting a non uniform VQ would require a new computation of the vector quantizer whenever new semantic classes are added to the dataset.

B. *Classification*

Two simple classification experiments have been performed, where each image q has been used as query against all the others. In the first experiment, q has been assigned to the class minimizing the average distance; in the second, to the class of the top-ranked object retrieved, that is the most similar to q. Results obtained are reported in tables I and II, and show an accuracy of 71.3% for the former method and 76.3% for the latter. It has to be remarked that intraclass variability in the COREL dataset is sometimes very high: for example most of the 10 images not recognized for the African class reported in Table I may be in fact considered as outliers since just landscapes with no human presence are contained within (see fig. 7); this shows the existence of limits imposed by the subjective choice of the training datasets.

On a computer with a double 2 GHz processor and 2GB of RAM the total running time for extracting the dictionaries and compute the distance matrix for the 1500 64x64 images was around 20 minutes, while it takes more than 150 with *NCD* (estimated through the tool Complearn [26]) : considering that the java code used is not yet optimized for speed, this makes the *DD* a good candidate for applications to larger databasets and image information mining, and a good compromise between execution speed and quality of the results obtained.

V. Conclusions

A new approach to image retrieval based on data compression has been presented. The main idea is to extract directly from the data typical dictionaries representing the

TABLE I
CONFUSION MATRIX FOR CLASSIFICATION ACCORDING TO THE MINIMUM AVERAGE DISTANCE FROM A CLASS

	Afr.	Beach	Archit.	Bus.	Dinos.	Eleph.	Flow.	Hors.	Mount.	Food	Caves	Post.	Suns.	Tig.	Wom.
Africans	**90**	0	0	0	1	0	0	0	0	1	0	0	0	8	0
Beach	12	**43**	8	14	0	1	0	0	1	3	0	0	0	18	0
Architecture	7	0	**72**	3	0	0	0	0	0	1	0	0	1	16	0
Buses	6	0	0	**93**	0	0	0	0	0	1	0	0	0	0	0
Dinosaurs	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0
Elephants	16	0	2	2	0	**46**	0	4	0	3	0	1	0	26	0
Flowers	6	0	3	1	0	0	**83**	1	0	3	0	0	0	3	0
Horses	0	0	0	0	0	0	0	**97**	0	0	0	0	0	3	0
Mountains	7	1	11	23	0	2	0	0	**39**	0	0	0	0	17	0
Food	6	0	0	1	0	0	0	0	0	**92**	0	0	0	1	0
Caves	17	0	9	1	0	1	0	0	0	5	**60**	0	0	7	0
Postcards	0	0	0	0	1	0	0	0	0	1	0	**98**	0	0	0
Sunsets	18	0	1	6	0	0	2	0	0	16	3	1	**39**	14	0
Tigers	1	0	0	1	0	0	0	5	0	0	0	0	0	**93**	0
Women	35	0	0	6	2	0	0	0	0	20	4	0	0	5	**28**
Avg Accuracy								71.3%							

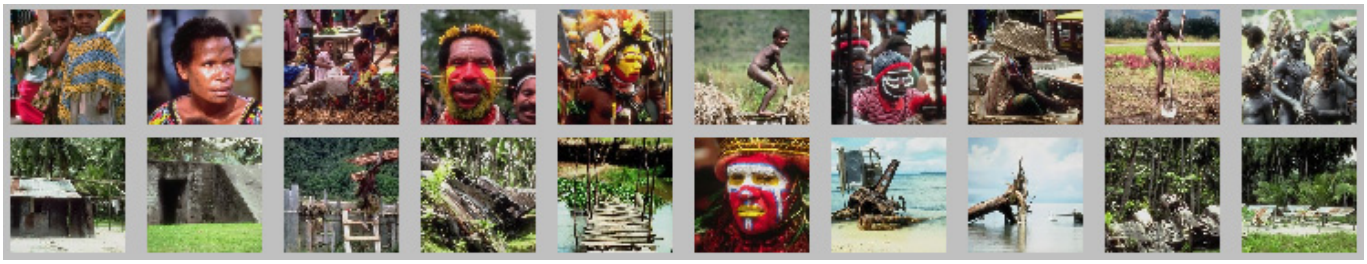

Fig. 7. Typical images for the class "Africans" (top row) and all misclassified images (bottom row), ref. Table I. The false alarms may be considered as outliers, and the confusion with the class "tigers" is justified by the landscapes dominating the images with no human presence, with the exception of the 6th one in the bottom row (incorrectly assigned to the class "food").

TABLE II
CONFUSION MATRIX FOR NEAREST NEIGHBOUR CLASSIFICATION

	Afr.	Beach	Archit.	Bus.	Dinos.	Eleph.	Flow.	Hors.	Mount.	Food	Caves	Post.	Suns.	Tig.	Wom.
Africans	**91**	0	0	0	0	0	0	0	0	1	1	0	0	7	0
Beach	8	**31**	9	6	0	8	0	0	15	0	5	1	0	16	1
Architecture	3	1	**59**	0	0	1	1	0	3	1	10	0	0	21	0
Buses	3	1	3	**86**	0	0	0	0	2	3	0	0	0	2	0
Dinosaurs	1	0	0	0	**98**	0	0	0	1	0	0	0	0	0	0
Elephants	0	0	1	0	0	**89**	0	2	0	1	1	0	0	6	0
Flowers	0	0	0	0	0	0	**96**	0	0	0	0	1	0	2	1
Horses	0	0	0	0	0	0	0	**95**	0	0	0	0	0	5	0
Mountains	2	11	7	9	1	9	0	0	**52**	1	3	0	2	3	0
Food	4	0	1	1	0	1	0	0	0	**91**	0	2	0	0	0
Caves	3	0	6	1	0	3	0	1	0	0	**82**	0	1	3	0
Postcards	4	0	0	0	1	0	0	0	0	10	0	**82**	0	3	0
Sunsets	3	0	1	3	0	2	3	0	0	3	9	0	**67**	9	0
Tigers	1	1	1	0	0	1	0	1	0	0	0	0	0	**95**	0
Women	25	0	0	1	1	4	3	0	4	8	13	0	0	10	**31**
Avg Accuracy								76.3%							

recurring patterns, trying to keep as much information as possible by employing quantization and by addition of the essential vertical information; in a subsequent step, similarities between two objects are computed on the basis of the size of the intersection set between the relative dictionaries.

The precision-recall curves show that the proposed method performs better than previous similar techniques; furthermore, it avoids the long processing times usually required by compression-based techniques, which generally process redundantly the full data. Finally, the scalar quantization adopted facilitates the addition of new images to the database, since no parameters need to be recomputed afterwards. To further reduce the processing time, a DataBase System could be employed, representing each dictionary with a table in the database, thus enabling quick queries on the joint table sets.

ACKNOWLEDGMENT

This work was carried out in the frame of the joint German Aerospace Center–Centre National d'Etudes Spatiales-Télécom Paris Tech Centre of Competence for Information Extraction and Image Understanding for Earth Observation. The authors would like to thank A. H. Daptardar for providing the data and for many useful comments, and Dr. C. Cucu-Dumitrescu for the original idea of dictionary intersection.

REFERENCES

[1] N. Vasconcelos, "Minimum probability of error image retrieval" *IEEE Trans. Signal Proc.*, vol. 52, no. 8, pp. 2322–2336, 2004.

[2] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval: the end of the early years," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 22, pp. 1349–1380, Dec. 2000.

[3] M. Flickner et al., "Query by Image and Video Content: the QBIC System", *ACGM SIGMOD Record*, vol. 24, no. 2, pp. 475-484, 1995.

[4] S. Jeong and R. M. Gray, "Image Retrieval using Color Histograms Generated by Gauss Mixture Vector Quantization", *Comput. Vision Image Understand.*, no. 94, pp.44-66, 2004.

[5] S. Jeong and R. M. Gray, "Minimum Distortion Color Image Retrieval based on Lloyd-clustered Gauss Mixtures", *DCC 2005*, pp. 279-288, 2005.

[6] A. J. Daptardar and J. A. Storer, "Content-based Image Retrieval via Vector Quantization", *ISVC*, *Lecture Notes in Computer Science*, vol. 3804, pp. 502-509, 2005.

[7] A. H. Daptardar and J. A. Storer, "VQ based Image Retrieval using Color and Position Features", *DCC 2008*, pp. 432-441, 2008.

[8] M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitányi, "The Similarity Metric", *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3250-3264, 2004.

[9] E. J. Keogh, S. Lonardi, C. Ratanamahatana, "Towards Parameter-Free Data Mining", *SIGKDD 2004*.

[10] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-insensitive Integrated Matching for Picture Libraries", *IEEE Trans. Pattern Anal. Mac. Intel.*, vol. 23, no. 9, pp. 947-963, 2002.

[11] A. N. Kolmogorov, "Three Approaches to the Quantitative Definition of Information," *Problems of Information Transmission*, vol. 1, no. 1, pp. 1–7, 1965.

[12] R. Cilibrasi and P.M.B. Vitányi, "Clustering by Compression", *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1523-1545, 2005.

[13] R. Cilibrasi, P. Vitányi, and R. de Wolf, "Algorithmic Clustering of Music based on String Compression", *Computer Music Journal*, vol. 28, no. 4, pp.49-67, 2004.

[14] R. Cilibrasi, "Statistical Inference Through Data Compression", *Lulu.com Press* , 2006.

[15] T. Watanabe, K. Sugawara, and H. Sugihara, "A New Pattern Representation Scheme Using Data Compression", *IEEE Trans. Pattern Anal. Mac. Intel.*, vol. 24, pp. 579-590, 2002.

[16] D. Cerra and M. Datcu, "Compression-based Data Clustering: Similarity Measures using Dictionaries and Grammars", submitted to *IEEE Tr. Pattern Anal. Mac. Intel.*

[17] J. Ziv and A. Lempel, "Compression of Individual Sequences Via Variable-Rate Coding", *IEEE Trans. Inf. Theory*, 1978.

[18] D. Cerra and M. Datcu, "Image Classification using Data Compression Based Techniques", *Proceedings IGARSS'08*, vol.1, pp. 237-240, 2008.

[19] R. M. Gray, "Vector Quantization", *IEEE ASSP Mag.*, pp. 4--29, 1984.

[20] Y. Androutsos, K.N. Plataniotis, and A.N. Venetsanopoulos, "A Novel Vector-based Approach to Color Image Retrieval using a Vector Angular-based Distance Measure", *Comput. Vision Image Understand.*, no. 75, pp. 46-58, 1999.

[21] C. Cucu-Dumitrescu, M. Datcu, F. Serban, and M. Buican, "Data Mining in Satellite Images using the PRDC Technique", *Romanian Astronomy Journal,* vol. 19, no. 1, pp. 63-79, 2009.

[22] Welch, T. A., "A technique for high-performance data compression", *Computer*, vol. 17, no. 6, pp. 8-19, 1984.

[23] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval", *ACM Press*, 1999.

[24] A. Torralba, "How Many Pixels Make an Image?", *Visual Neuroscience*, vol. 2, no. 1, pp. 123-131, 2009.

[25] X. Zhang and X. Wu, "Can Lower Resolution be Better?", *DCC 2008*, pp.302-311, 2008.

[26] R. Cilibrasi, A. Cruz, S. de Rooij, and M. Keijzer, Complearn tool, [Online] available at http://www.complearn.org/index.html.