ON THE CONTRIBUTION OF COMPRESSION TO VISUAL PATTERN RECOGNITION

Gunther Heidemann
Intelligent Systems Group, University of Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, Germany

Helge Ritter
Neuroinformatics Group, Bielefeld University, Universitätsstr. 25, D-33615 Bielefeld, Germany

Keywords: Compression, mutual information, Lempel-Ziv, gzip, bzip2, object recognition, texture, image retrieval.

Abstract: Most pattern recognition problems are solved by highly task specific algorithms. However, all recognition and classification architectures are related in at least one aspect: They rely on compressed representations of the input. It is therefore an interesting question how much compression itself contributes to the pattern recognition process. The question has been answered by Benedetto et al. (2002) for the domain of text, where a common compression program (gzip) is capable of language recognition and authorship attribution. The underlying principle is estimating the mutual information from the obtained compression factor. Here we show that compression achieves astonishingly high recognition rates even for far more complex tasks: Visual object recognition, texture classification, and image retrieval. Though, naturally, specialized recognition algorithms still outperform compressors, our results are remarkable, since none of the applied compression programs (gzip , bzip2) was ever designed to solve this type of tasks. Compression is the only known method that solves such a wide variety of tasks without any modification, data preprocessing, feature extraction, even without parametrization. We conclude that compression can be seen as the "core" of a yet to develop theory of unified pattern recognition.

1 INTRODUCTION

Pattern recognition is a task that has to be solved by many biological organisms and technical systems alike. Though applicable solutions have been found in branches like speech recognition or computer vision, neither a unifying theory of pattern recognition exists nor even a broadly usable algorithmic method. To date, almost any pattern classification system is tailored to a specific task — not only by its particular processing design, but also by a concomitant usually very careful parameterization. Almost all approaches in pattern recognition, however, have one property in common, regardless of the particular domain: They use some kind of *compressed* representation of the original input data, partly for redundancy reduction, partly for filtering out only the most discriminative constituents.

In a vigorously discussed paper (Benedetto et al., 2002), Benedetto et al. relate compression to pattern recognition in an amazingly straightforward way: They used the common gzip compressor for language recognition (see also (Benedetto et al., 2003), for comments see (Cho, 2002; Ball, 2002; Khmelev and Teahan, 2003)). The method is surprisingly simple: There are n versions $T_1 \ldots T_n$ of a text in different languages. Each text file T_i is compressed using gzip , the resulting bit length $S(T_i)$ of the compressed file has to be memorized. Now the language i^* of a new text T^* (written in one of the n languages) can be recognized in the following way: T^* is appended to each of the uncompressed texts $T_1 \ldots T_n$ to obtain the enlarged files $T_1^* \ldots T_n^*$. Then the enlarged files are compressed, their bit lengths being $S(T_1^*) \ldots S(T_n^*)$.

The language i^* of T^* is then obtained by $i^* = \arg\min_i (S(T_i^*) - S(T_i))$, i.e., given by the language of the text T_{i^*} which exhibits the *smallest* increase $S(T_i^*) - S(T_i)$ of its compressed length after appending T^*.

The principle of this method is straightforward. The gzip program is based on the Lempel-Ziv algorithm (LZ77) (Lempel and Ziv, 1977), which de-

Heidemann G. and Ritter H. (2008).
ON THE CONTRIBUTION OF COMPRESSION TO VISUAL PATTERN RECOGNITION.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 83-89
DOI: 10.5220/0001078000830089

tects repeatedly occurring symbol sequences within the data, such that a dictionary can be established. A repeated symbol sequence can then be replaced by the symbol defined in the dictionary. Thus, it is not surprising that compression of a text appended to another one in the same language profits from the availability of shared constituents caused by the similarity of text strings. However, the accuracy reported in (Benedetto et al., 2003) is astonishing, it is possible not only to recognize language and to attribute authorship, but to reconstruct entire language trees.

The question arises if the good performance compression achieves for recognition is just the result of a judicious combination of a particular algorithm (LZ77) with a particular recognition task (text). In this paper, we investigate the two crucial questions that must be answered to allow generalization of the results reported by Benedetto et al.:

1. Text is a linear sequence of symbols, which makes recognition by compression easy. It can thus be objected that the method would fail for a more difficult pattern recognition task, in particular, when sensor data are to be evaluated. We will therefore apply the method to three real world vision tasks.

2. Using compression for recognition might depend on the particular way LZ77 performs compression, which explicitly searches for repeated symbol sequences. We will therefore apply also an alternative compression algorithm (`bzip2`) relying on different principles (Burrows and Wheeler, 1994; Hirschberg and Lelewer, 1990).

In this paper we will show that, surprisingly, both objections do not hold: Recognition by compression is neither limited to text, nor is it bound to a particular compressor. As a consequence, compression appears to be a very essential aspect of pattern recognition and may be a first step towards a unifying theory. The approach can be connected with the concept of *mutual information* (cf. Section 2.2), which has already been used by several authors to gain a unified perspective on various important operations in pattern recognition systems (Sinkkonen and Kaski, 2002; Hulle, 2002; Imaoka and Okajima, 2004; Erdogmus et al., 2004). Section 2 describes the method itself, its theoretical background and the applied compression algorithms. In Section 3, experiments are carried out for three different problems: Object recognition, texture classification, and image retrieval. The concluding Section 4 discusses the results and implications.

2 COMPRESSION FOR RECOGNITION

2.1 Method

Following the approach of Benedetto et al. (Benedetto et al., 2002), we compare two images I_1, I_2 by considering the similarity measure

$$D_{Comp}(I_1,I_2) \;=\; S(I_1)+S(I_2)-S(I_{12}), \qquad (1)$$

where $S(.)$ denotes the bit size of a compressed image. I_{12} is the “joint” image obtained as juxtaposition of pixel arrays I_1 and I_2. In the experiments described in Section 3, both images I_1 and I_2 are first compressed in isolation to obtain the bit size $S(I_1)$ and $S(I_2)$ of their compressed representation, respectively. The original images I_1 and I_2 are then merged to become a single image I_{12}. Compressing the merged image I_{12} yields the bit size $S(I_{12})$.

Note that the method is applied to the raw images, without any preprocessing or adaptation.

2.2 Theoretical Background

The idea of Eq. (1) lies in information theory, which relates the size of the shortest possible message length for an information source I to its entropy $H(I)$. The compression factor, i.e. the bit length of the output of a compressor algorithm divided by the bit length of the uncompressed message can, therefore, be seen as an approximation of its entropy $H(I)$ — as long as the compressor works close to optimal. For LZ77, the compression factor indeed tends to $H(I)$ when the length of the message tends to infinity (Lempel and Ziv, 1977; Wyner, 1994).

Thus, $D_{Comp}(I_1,I_2)$ can be seen to approximate the *mutual information* $H(I_1,I_2)$ of I_1 and I_2, which in information theory measures the amount of information that I_1 can predict about I_2 and vice versa (Cover and Thomas, 1991). A large positive value of D_{Comp} thus indicates that I_1 and I_2 are very similar, while the smallest possible value of zero is obtained when the two images are completely unrelated. $D_{Comp}(I_1,I_2)$ measures similarity, but can not be used as a distance measure in a strict sense.

The idea to judge pattern similarity by compression properties is not new per se and is related to the Minimum-Description-Length (MDL) principle for model selection (Rissanen, 1978; Vitanyi and Li, 1996). In image processing, MDL has been applied as a global criterion to optimize segmentation (Leclerc, 1989; Keeler, 1990; Kanungo et al., 1994). However, such approaches operate on the level of regions, which are defined by task dependent similarity criteria. So, the uses of compression-based and related methods were restricted to low-dimensional data. Application on raw data (of much higher dimension) helps not only avoid the problematic stage of feature extraction, but opens up the possibility to treat signals stemming from different sources in a unified way.

2.3 Compression Algorithms

We used two standard lossless compression tools which rely on different principles: The `gzip` program[1], which is based on LZ77 (Lempel and Ziv, 1977), and the `bzip2` program[2]. `bzip2` was chosen for comparison with `gzip` because it relies on a completely different compression technique: the Burrows-Wheeler block sorting text compression algorithm (Burrows and Wheeler, 1994) with subsequent Huffman coding (Hirschberg and Lelewer, 1990). Thus, it can be shown that the ability to judge pattern similarity is not a result of the particular way LZ77 builds up a compressed representation.

2.4 Discussion of the Similarity Measure

There are obvious methods that would improve the performance of the approach. In particular, specialization to the domain of images would probably increase recognition rates:

- `gzip` and `bzip2` perform *lossless* compression. But to judge image similarity, minor variation of gray values should be tolerated. So, tuning the compressors to a certain level of data loss would probably improve performance.
- The compression algorithms are neither optimal to approximate the mutual information, nor are they the best compression techniques on the market. Compression can be improved when algorithms specialized to a particular data type are used.
- The measure is applied only for judging similarity of *complete* images. An additional segmentation would help to identify objects or patterns in the presence of varying background, however, segmentation is a specialized computer vision technique.

But we did not make use of any of these measures, since this would destroy the simplicity of the method and its universal applicability to entirely different data domains. It is not the aim of this paper to build an actually usable recognition system but to demonstrate the capability of the approach in a way easy to reproduce and transferable to other domains. For this demonstration, `gzip` and `bzip2` were used in their original Unix-implementation without any modification.

3 EXPERIMENTS

3.1 Object Recognition

The first experiment was an object recognition task. Images were taken from the COIL-100 library (Nene et al., 1996), which is a standard benchmark data set for object recognition[3]. It comprises for each of 100 different objects 72 rotational views (128×128 pixel resolution, 5^0 rotation angle separation) of the object centered on a dark background. Since color facilitates recognition, we discarded color information and used only the gray level version of each image.

For each test, the COIL-100 was partitioned into a set M_α of "memorized" object views and the complementary set U_α of "unknown" object views. Several test sets were created, whose partitions differed by the chosen view angle spacing α of successive views selected for the "memorized" set M_α (e.g., M_{15} contains poses of $0, 15, 30\dots$ degrees and U_{15} poses of $5, 10, 20, 25\dots$ degrees).

For recognition of an unknown image $I_i^U \in U_\alpha$ we computed its similarities $D_{Comp}(I_i^U, I_j^M)$ with each of the memorized images $I_j^M \in M_\alpha$, using Eq. (1) and either `gzip` or `bzip2` to obtain the size values $S(.)$. The memorized image I_j^M leading to maximal D_{Comp} was then taken to identify the "unknown" image I_i^U.

Figure 1 shows the results of the object classification task. Naturally, object representations including more memorized views (smaller angular spacing α) lead to better recognition. For $\alpha = 10^0$, `gzip` reaches 99.4% correct classifications (chance level is 1%). Remarkably, even for very sparse sampling recognition is considerable: $\alpha = 120^0$ (3 views) still leads to 82.0% correct classifications. Different levels of compression by which the tradeoff between computational efficiency and compression factor can be influenced lead only to minor performance changes.

To give an estimate of the difficulty of the task, two complementary basic methods were used: *Correlation* based matching shows how much of the recog-

[1] In version `gzip 1.2.4` available from `http://www.gzip.org`.

[2] In version `bzip2 1.0.1`, available from `http://sources.redhat.com/bzip2`.

[3] Available from `http://www.cs.columbia.edu/CAVE/research/softlib/coil-100.html`.

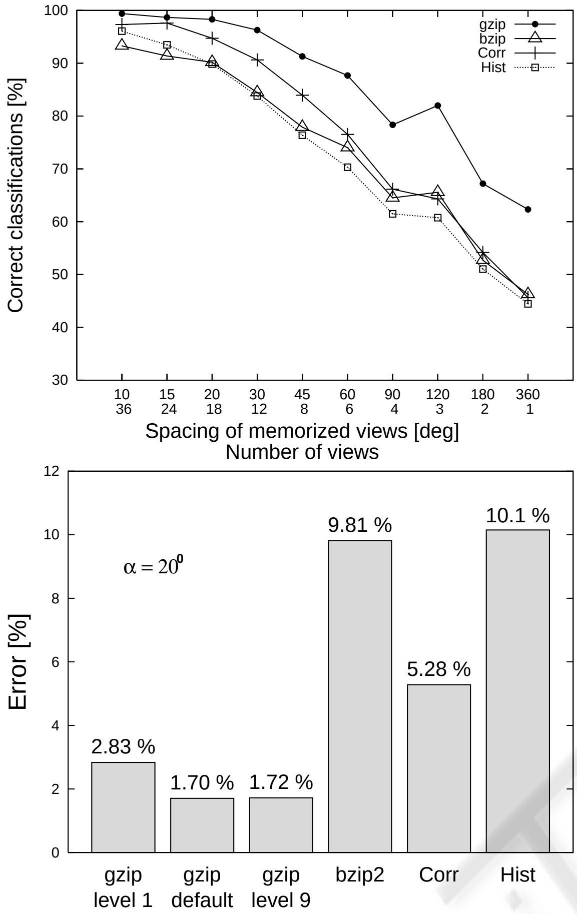

Figure 1: Object recognition results for a gray value version of COIL-100. Compression based on gzip performs better than correlation based matching, histogram matching, and compression using bzip2 .
Above: The percentage of correct classifications decreases with the number of memorized object views. The advantage of gzip becomes clear especially for large angular spacing α of memorized views.
Below: Error rates for a fixed angular spacing $\alpha = 20^0$ of memorized views. gzip is superior for all compression levels. Compression level 1 biases the tradeoff between compression and speed for best speed, level 9 for compression. The default setting selects level 6 as a compromise between these two.

nition results can be explained by a simple comparison of the spatial gray value distribution, whereas gray value *histogram* matching is independent of the spatial structure. For classification from correlation, instead of D_{Comp} the similarity measure D_{Corr} based on the pixel correlation of the normalized images $\hat{I}_1$ and $\hat{I}_2$ was used:

$$D_{Corr}(I_1,I_2) \;=\; \sum_{xy} \hat{I}_1(x,y) \cdot \hat{I}_2(x,y) \quad \text{with} \quad \|\hat{I}_{1,2}\| = 1. \qquad (2)$$

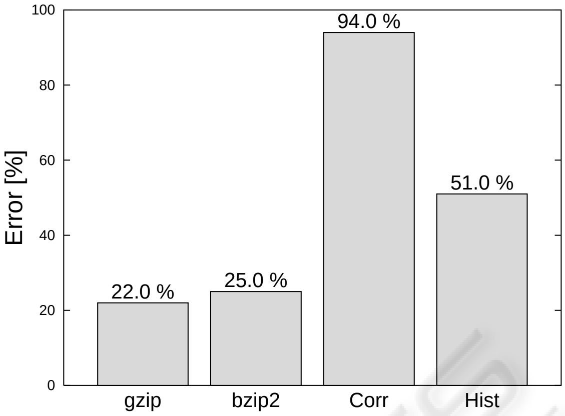

Figure 2: Recognition results for 50 gray level VisTex texture image pairs. gzip and bzip2 outperform correlation based matching and histogram matching.

Histogram based similarity is calculated by

$$D_{Hist}(I_1,I_2) \;=\; -\frac{1}{2}\sum_{i=1}^{q}\left(C_i(I_1) - C_i(I_2)\right)^2, \qquad (3)$$

where $C_i(I_j)$ denotes the count of pixels of image I_j with gray values in histogram bin $i \in \{1\ldots q\}$. We use bin boundaries equidistant in the range $[0\ldots 255]$. Throughout the paper a value $q = 6$ was used because on average it yields the best results.

D_{Comp} based on gzip performs clearly much better than both D_{Corr} and D_{Hist} (Figure 1), which indicates that gzip implicitly uses a combination of the spatial structure and the gray value frequencies for recognition. bzip2 performs between D_{Corr} and D_{Hist}, but as correct classification is still over 90%, results are good also for bzip2 .

3.2 Texture Classification

In the second experiment, the capability of D_{Comp} to discriminate textures was tested. From the VisTex database (Picard et al., 1995) 50 image pairs were selected, each showing two different views of the same texture at resolution 512×512 pixels and transformed to gray scale. The database comprises both natural and artificial textures and includes difficulties like different perspectives, scaling and distortions. This time, the sets M and U were formed of all “first” and “second” views of the 50 pairs, respectively. Figure 2 depicts the percentage of errors in assigning views of U to their partner views in M. Both gzip and bzip2 reach much better results than correlation and histogram matching. Naturally, techniques specialized on texture classification would yield still better results, but it has to be kept in mind that D_{Comp} is treated here as a general purpose recognition method.

3.3 Image Retrieval

With the upcoming of large image collections on the Internet or in databases, a major challenge is image retrieval and indexing. The inherent diversity of this domain makes the extraction of good general features particularly difficult. Consequently, retrieval systems still mostly rely on color and texture information, while potentially more powerful structural features are only rarely exploited (for an overview see e.g. (Smeulders et al., 2000)).

Therefore, in the third experiment it was tested if LZ77 can discriminate image categories. As a database, 20 categories were formed from the Corel database (Corel, 1997), each consisting of 80 images. Since for some categories color is known to provide an exceptional feature that facilitates discrimination, again all images were transformed to gray level. Figure 4 shows some example images.

A typical retrieval task is to find similar images by specification of a query image. We calculated for each of the 1599 non-query images the k images that were most similar to the query image in terms of D_{Comp}. A query was counted a success if among the k query results there was at least one of the correct category. Figure 3 shows the results: Even for $k = 1$, about three out of four query results yield the correct category, for $k = 4$, the success rate rises to over 90% for both `gzip` and `bzip2` . This last experiment appears particularly impressive, because most visual categories are much broader than single object classes (Tarr and Bülthoff, 1998).

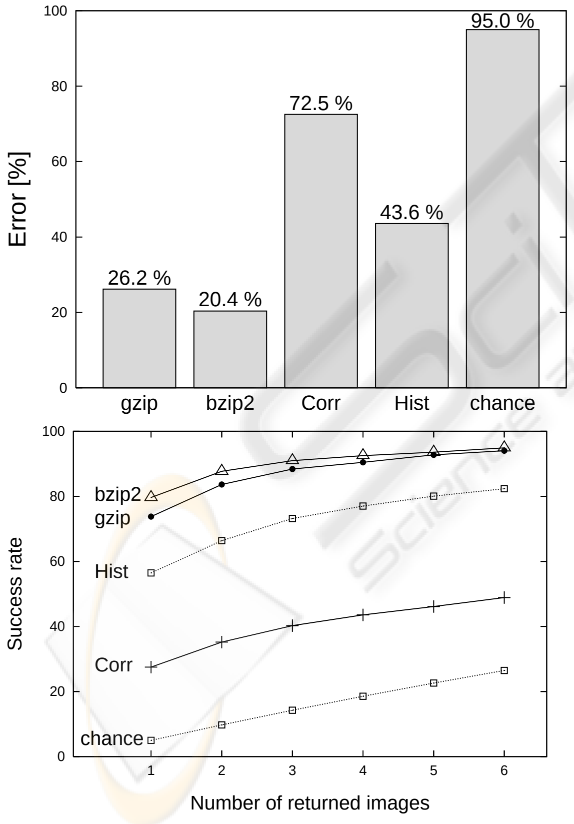

Figure 3: Recognition results for the retrieval task. Above: Error rates if only one image is returned for a query ($k = 1$), this case resembles a mere classification-type task. Below: In an actual retrieval system, usually more than one image is returned ($k > 1$). A query is counted a "success" if at least one image of the correct category is returned.

4 CONCLUSIONS

For each of the three types of tasks presented here carefully specialized recognition architectures exist (e.g. (Murase and Nayar, 1995; Paulus et al., 2000; Rui et al., 1999; Laaksonen et al., 2000)), which can achieve better results, at least in the case of object and texture recognition. What makes the compression based approach unique is its simplicity and applicability to entirely different types of domains: Without any designed feature extraction and without tuning of parameters, even "of the shelf" compression programs achieve remarkable results for vision tasks as well as text. Most astonishing appears the performance for image retrieval based on gray values — most such systems heavily depend on color (Smeulders et al., 2000).

Naturally, the unmodified `gzip` and `bzip2` programs are not yet the optimal or fastest solution for entropy approximation — they were chosen to illustrate the principle in a "pure" and reproducible form. As pointed out in (Khmelev and Teahan, 2003), *n*th order Markov chain models outperform `gzip` on certain text recognition tasks. But the fact that the results do not depend on the particular choice of LZ77 raises hope that compression is a fundamental mechanism that will open up a new perspective on pattern recognition. Though we do not yet have a universal theory of pattern recognition, compression is very likely to be one of its key components.

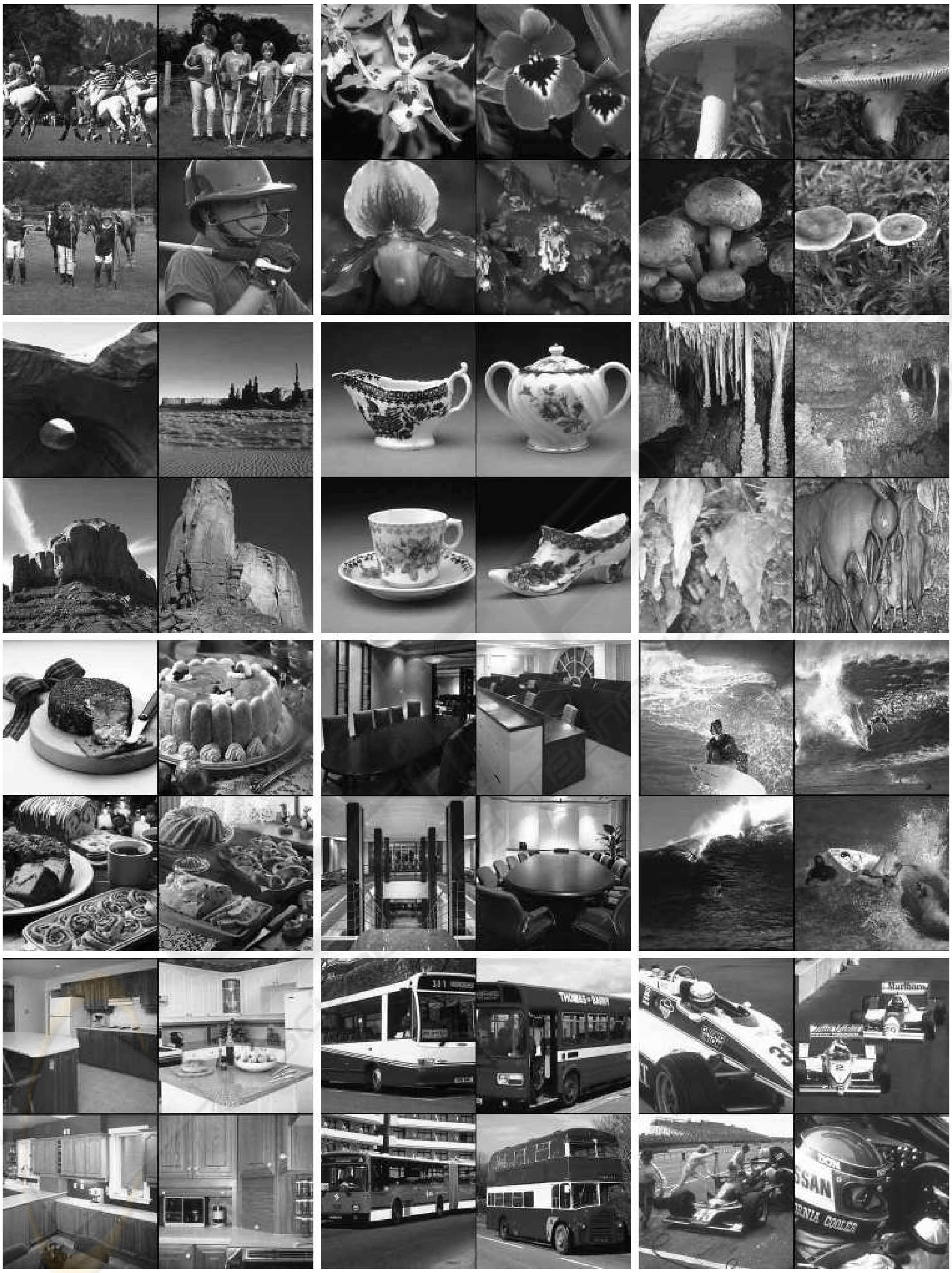

Figure 4: As a retrieval task, query images had to be found in 20 categories, each comprising 80 gray level images (color was discarded). Here, 12 categories are shown by four images each. From left to right and top to bottom: “Polo”, “blossoms”, “mushrooms”, “desert”, “porcelain”, “stalactite caves”, “food on the table”, “interiors: hotels”, “surfing”, “interiors: kitchens”, “busses”, “car racing”. The complete dataset can be made available on demand.

REFERENCES

Ball, P. (2002). Algorithm makes tongue tree. *Nature Science Update*.

Benedetto, D., Caglioti, E., and Loreto, V. (2002). Language Trees and Zipping. *Phys. Rev. Lett.*, 88(4).

Benedetto, D., Caglioti, E., and Loreto, V. (2003). Zipping out relevant information. *Computing in Science and Engineering*, 5:80–85.

Burrows, M. and Wheeler, D. J. (1994). A Block-sorting Lossless Data Compression Algorithm. Research Report 124, Digital Systems Research Center.

Cho, A. (2002). Reading the Bits of Shakespeare. *ScienceNOW*.

Corel (1997). *Corel GALLERY*^{TM} *Magic 65000*. Corel Corp., 1600 Carling Ave., Ottawa, Ontario, Canada K1Z 8R7.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley, New York.

Erdogmus, D., Hild, K. E., Rao, Y. N., and Príncipe, J. C. (2004). Minimax Mutual Information Approach for Independent Component Analysis. *Neural Computation*, 16(6):1235–1252.

Hirschberg, D. S. and Lelewer, D. A. (1990). Efficient Decoding of Prefix Codes. *Communications of the ACM*, 33(4):449–459.

Hulle, M. M. V. (2002). Joint Entropy Maximization in Kernel-Based Topographic Maps. *Neural Computation*, 14(8):1887–1906.

Imaoka, H. and Okajima, K. (2004). An Algorithm for the Detection of Faces on the Basis of Gabor Features and Information Maximization. *Neural Computation*, 16(6):1163–1191.

Kanungo, T., Dom, B., Niblack, W., and Steele, D. (1994). A fast algorithm for MDL-based multi-band image segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition CVPR*.

Keeler, A. (1990). Minimal length encoding of planar subdivision topologies with application to image segmentation. In *AAAI 1990 Spring Symposium of the Theory and Application of Minimal Length Encoding*.

Khmelev, D. V. and Teahan, W. J. (2003). Comment on ”Language Trees and Zipping”. *Physical Review Letters*, 90(8):089803–1.

Laaksonen, J. T., Koskela, J. M., Laakso, S. P., and Oja, E. (2000). PicSOM – Content-Based Image Retrieval with Self-Organizing Maps. *Pattern Recognition Letters*, 21(13-14):1199–1207.

Leclerc, Y. G. (1989). Constructing simple stable descriptions for image partitioning. *Int'l J. of Computer Vision*, 3:73–102.

Lempel, A. and Ziv, J. (1977). A Universal Algorithm for Sequential Data Compression. *IEEE Trans. Inf. Th.*, 23(3):337–343.

Murase, H. and Nayar, S. K. (1995). Visual Learning and Recognition of 3-D Objects from Appearance. *Int'l J. of Computer Vision*, 14:5–24.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library: COIL-100. Technical Report CUCS-006-96, Dept. Computer Science, Columbia Univ.

Paulus, D., Ahrlichs, U., Heigl, B., Denzler, J., Hornegger, J., Zobel, M., and Niemann, H. (2000). Active Knowledge-Based Scene Analysis. *Videre*, 1(4).

Picard, R., Graczyk, C., Mann, S., Wachman, J., Picard, L., and Campbell, L. (1995). Vision Texture Database (VisTex). Copyright 1995 by the Massachusetts Institute of Technology.

Rissanen, J. (1978). Modeling by Shortest Data Description. *Automatica*, 14:465–471.

Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image Retrieval: Current Techniques, Promising Directions and Open Issues. *J. of Visual Communications and Image Representation*, 10:1–23.

Sinkkonen, J. and Kaski, S. (2002). Clustering Based on Conditional Distributions in an Auxiliary Space. *Neural Computation*, 14(1):217–239.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.

Tarr, M. J. and Bülthoff, H. H. (1998). Image-Based Object Recognition in Man, Monkey and Machine. *Cognition*, 67:1–20.

Vitanyi, P. M. B. and Li, M. (1996). Ideal MDL and its Relation to Bayesianism. In *Proc. ISIS: Information, Statistics and Induction in Science World Scientific*, pages 282–291, Singapore.

Wyner, A. D. (1994). 1994 Shannon Lecture. Typical Sequences and All That: Entropy, Pattern Matching, and Data Compression. AT & T Bell Laboratories, Murray Hill, New Jersey, USA.