FINDING UNKNOWN REPEATED PATTERNS IN IMAGES

Armando J. Pinho and Paulo J. S. G. Ferreira

Signal Processing Lab, IEETA / DETI
University of Aveiro, 3810–193 Aveiro, Portugal
{ap,pjf}@ua.pt

ABSTRACT

We consider the problem of finding unknown patterns that appear multiple times in a digital image. We want to find the number of repetitions and their positions in the image without constraining the nature and shape of the pattern in any way. We propose a method that is able to pinpoint the possible locations of the repetitions by exploring the connection between image compression and image complexity. The method uses a finite-context model to build a complexity map of the image in which the repeated patterns correspond to areas of low complexity, which mark the locations of the repetitions.

1. INTRODUCTION

Finding repetitions of exact or approximate unknown patterns in images is a difficult problem that has not yet received a satisfactory treatment. This paper addresses this problem and proposes an information-theoretic solution. We estimate the complexity of the images using a class of compression algorithms that is able to approximate the Kolmogorov complexity. The key insight is that the Kolmogorov complexity of a repeated pattern is essentially that of the pattern itself. This shows that repetitions are closely associated with low-complexity areas. To turn this idea into practice we use finite-context models that can capture, in a compact form, the most relevant features of a given image or image region. These models are then used to look for regions with similar characteristics in the same image or in different images.

The relations between complexity theory and data compression are known for some time and have been used in a number of domains (the next section gives some background information). Their potential interest for image analysis is great, but the nature of the commonly available compression algorithms is a major obstacle to further progress. Not every compression method can be used for approximating the Kolmogorov complexity. Only techniques that are able to create internal models of the data are suitable. This requirement excludes most of the popular image compression methods, because they do not create these internal representations. To overcome this limitation, we developed algorithms that rely on finite-context modeling and that work directly on the intensity domain of the image, avoiding the transform or predictive steps that would destroy most of the data dependencies.

A finite-context model provides, on a symbol by symbol basis, an information measure that corresponds in essence to the number of bits required to represent the symbol, conditioned by the accumulated knowledge of all past symbols. We use this information to build complexity surfaces, i.e., images in which the intensity of the pixels indicate how complex is the corresponding region of the original image. Patterns that occur more than once in an image tend to require less bits to encode as repetitions of these patterns are found. Since the repetitions are associated with low complexity regions in the complexity surface, they can be more easily unveiled.

2. MOTIVATION AND BACKGROUND

The works of researchers such as Solomonoff, Kolmogorov, Chaitin and others [19, 20, 8, 3, 23, 18], related to the definition of complexity measures, has found applications in several areas of knowledge. The Kolmogorov complexity of A, denoted by $K(A)$, is defined as the size of the smallest program that produces A and stops. A major drawback of the Kolmogorov complexity (also known as the algorithmic entropy) is that it is not computable. To overcome this limitation, it is usually approximated by a computable measure, such as Lempel-Ziv based complexity measures [9], linguistic complexity measures [7] or compression-based complexity measures [5]. These approximations provide upper bounds on the Kolmogorov complexity.

Compression algorithms provide a natural way of approximating the Kolmogorov complexity, because, together with the appropriate decoder, a bitstream produced by a lossless compression algorithm can be used to reconstruct the original data. The number of bits required for representing these two components (decoder and bitstream) can be viewed as an estimate of the Kolmogorov complexity. Moreover, the search for better compression algorithms is directly related to the problem of improving the complexity bounds.

The evaluation of the similarity between two objects is one of the problems that can be attacked using Kolmogorov theory. Following this line, Li et al. proposed a similarity metric [10] based on an information distance [2], defined as the length of the shortest binary program that is needed to transform A and B into each other. This distance depends not only on the Kolmogorov complexity of A and B, respectively $K(A)$ and $K(B)$, but also on conditional complexities, for example $K(A|B)$, that indicates how complex A is when B is known. Because this distance is based on the Kolmogorov complexity (which is not computable), they proposed a practical analog based on standard compressors, which they call the normalized compression distance [10].

Successful applications of these principles have been reported in areas such as genomics, virology, languages, literature, music, handwritten digits and astronomy [4]. However, applications of the normalized compressing distance to the imaging area are scarce. This might look surprising, but it is justifiable due to the following reasons. According to Li et al. [10], a compression method needs to be "normal" in order to be used as a normalized compression distance. One of the conditions for a compression method to be normal is that compressing AA (the concatenation of A with A) should generate essentially the same number of bits as compressing A alone [4].

This implies that, in order to be suitable to approximate the Kolmogorov complexity, a compression algorithm needs to accumulate knowledge of the data while the compression is performed. It needs to be able to find dependencies, to collect statistics, i.e., it has to create an internal model of the data.

The Lempel-Ziv compression algorithms belong to the class of methods that create internal data models. They are also the most often used compression algorithms in compression-based complexity applications, including those reported in the imaging field [22, 12, 6]. Unfortunately, although the Lempel-Ziv compression techniques are quite effective for uni-dimensional data, they do not perform as well in the case of multi-dimensional data and hence in images.

On the other hand, state-of-the-art image compressors, such as those of the JPEG2000 [21] or JPEG-LS [24] standards, are not normal. They start by decorrelating the data using a transformation (for example, the DCT or DWT as in JPEG or JPEG2000) or a predictive method (as in JPEG-LS). Therefore, they assume an a priori data model that remains essentially static during compression. Moreover, this decorrelating step destroys most of the data dependencies, leaving to the entropy coding stage the mere task of encoding symbols from an (assumed) independent source. Because they are not normal, they cannot be used for the purpose of conditional complexity estimation, i.e., for the estimation of $K(A|B)$.

These drawbacks lead us to seek compression algorithms that are both normal and adequate to images. We found that algorithms based on finite-context models are good candidates: they build an image model and are well suited to images. As an example, a method based on binary tree decomposition and arithmetic coding driven by finite-context models is capable of an average lossless compression performance 7.9% better than JPEG2000 [15], on the eighteen 8-bit ISO images. An L-infinity-constrained version of the technique also exists [16, 17]. A method based on arithmetic coding driven by a multi-bitplane finite-context model lead to a state-of-the-art microarray image coding technique [14]. Since finite-context models show good performance with images and are normal, in the sense previously explained, they are natural candidates to build effective compression-based image complexity measures. In the next section we discuss how we did this and what results we obtained.

3. THE PROPOSED METHOD

The main idea is to explore finite-context models because they can capture, in a compact form, the most relevant features of a given image or image region. They can then be used to identify image regions with similar characteristics. These models provide, on a symbol by symbol basis, an information measure that corresponds to the number of bits that are required to represent the current symbol, taking into consideration the accumulated knowledge of all past symbols. Therefore, they can be used to build complexity surfaces, i.e., images in which the intensity of the pixels indicate how complex is the corresponding region of the original image. As far as we know, the idea of constructing these complexity surfaces and using them for image analysis is completely novel.

3.1 Finite-context models

Formally, a finite-context model collects statistics of an information source and, for every outcome of the source, assigns probability estimates to the symbols of the alphabet $\mathscr{A} = \{s_1, s_2, \ldots, s_{|\mathscr{A}|}\}$, where $|\mathscr{A}|$ denotes the size of the alphabet. These estimates are calculated taking into account a conditioning context computed over a finite and fixed number, $k > 0$, of past outcomes (usually, the most recent) $x_{n-k+1..n} = x_{n-k+1} \ldots x_{n-1}x_n$ (order-k finite-context model) [1]. The number of conditioning states of the model is $|\mathscr{A}|^k$. In the case of multi-dimensional data, and particularly in the case of images, the notion of recent past usually refers to spatial proximity. This means that $x_{n-k+1..n}$ may refer to the set of the k spatially closest samples, and not necessarily to the k most recently processed samples. Nevertheless, causality is always preserved.

The probabilities, $P(X_{n+1} = s|x_{n-k+1..n}), \forall_{s\in\mathscr{A}}$, are calculated using symbol counts that are accumulated while the image is processed. Therefore, they are dependent not only of the k context symbols, but also of n. We estimate the probabilities using

$$P(X_{n+1} = s|x_{n-k+1..n}) = \frac{C(s|x_{n-k+1..n}) + \alpha}{C(x_{n-k+1..n}) + \alpha|\mathscr{A}|}, \qquad (1)$$

where $C(s|x_{n-k+1..n})$ represents the number of times that, in the past, the information source generated symbol s having $x_{n-k+1..n}$ as the conditioning context and where

$$C(x_{n-k+1..n}) = \sum_{a\in\mathscr{A}} C(a|x_{n-k+1..n}) \qquad (2)$$

is the total number of events that has occurred so far in association with context $x_{n-k+1..n}$. Parameter α allows balancing between the maximum likelihood estimator and an uniform distribution, preventing the estimator from generating zero probabilities (recall that encoding an event with an estimated probability equal to zero requires an infinite number of bits). Also note that if we define

$$\mu = \frac{C(x_{n-k+1..n})}{C(x_{n-k+1..n}) + \alpha|\mathscr{A}|}, \qquad (3)$$

we can rewrite (1) as

$$P(X_{n+1} = s|x_{n-k+1..n}) = \mu\frac{C(s|x_{n-k+1..n})}{C(x_{n-k+1..n})} + (1-\mu)\frac{1}{|\mathscr{A}|}, \qquad (4)$$

revealing a linear combination between the maximum likelihood estimator and the uniform distribution. Moreover, it is also easy to see that when the total number of events, n, is large, this estimator behaves essentially as a maximum likelihood estimator. When $\alpha = 1$, (1) is the well-known Laplace estimator.

3.2 Complexity surfaces

A complexity surface is an image ϕ, with the same geometry as the original image f, where each pixel $\phi_{i,j}$ contains the code length required to encode $f_{i,j}$ estimated by the finite-context model, i.e.,

$$\phi_{i,j} = -\log_2 P(F_{i,j} = f_{i,j}|c_{k,i,j}), \qquad (5)$$

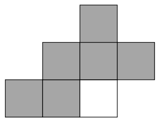

Figure 1: Context template of order six used by the finite-context model.

where $c_{k,i,j}$ denotes the (usually) two-dimensional order-k context and i, j the pixel coordinates. In the experiments reported in this paper we used the context configuration depicted in Fig. 1.

When using a method based on finite-context models, the data are scanned symbol by symbol, in a certain order. If the pattern A is found for the first time, it will be assigned a certain complexity, depending on the number of bits needed to represent it. When the pattern is seen again, the number of bits needed will be smaller. This reasoning shows that the complexity assigned to the pattern A depends on the order by which the data are scanned. In practice, the first seen occurrence of the pattern could be masked. This dependency is easy to remove, if desired: it is enough to scan the data in several directions and always assign the minimum complexity. In the examples presented in this paper, we have scanned the image in four different orders, that can be easily obtained by raster scanning the four different versions obtained by consecutive rotations of ninety degrees.

One of the problems associated with finite-context modeling is the exponential growth of the memory resources as a function of the size of the alphabet. Images, even gray-level, use alphabets that render these models almost useless. Therefore, before computing the complexity surfaces, we perform a reduction in the number of intensities to a maximum of twenty, using Lloyd-Max quantization [11, 13].

4. EXPERIMENTAL RESULTS

We now illustrate the potential of this idea by means of two examples. In the first one, we inserted four copies of a square textured region into different locations of the well-known "Lena" image. For a human, it is a matter of a fraction of a second for detecting all the occurrences of the alien pattern. However, not knowing the pattern in advance, it is not obvious how to build an efficient algorithm for finding all its occurrences. In Fig. 2, we show the modified "Lena" image and the corresponding complexity surface obtained using the method previously described (in fact, for facilitating the observation of the details, we display the logarithm of the complexity surface scaled to the $[0, 255]$ range). It is easy to find, in the complexity surface, the dark squares revealing the zones of low complexity that have been originated by the repetition of the pattern that we have inserted.

In Fig. 3, we show in greater detail the first 70 rows of the "Lena" image and the corresponding complexity surface showing a curious horizontal dark strip. It is curious that we only have noticed the existence of a repeated pattern in the first few rows of the image after visualizing its complexity surface, demonstrating one of the potential uses of the technique.

For a second, less obvious, example, we tampered a landscape image by replicating parts of a rock at several locations. Figure 4 shows the resulting image and its corresponding complexity surface. Contrarily to the example using the "Lena" image, where the modifications were evident, in this case it would be much harder to find the changes, even for a human (this clearly contrasts with the fact that for an algorithm both cases are equally challenging). Once more, it can be easily observed that in the labeled locations (marking the positions of the copies) the complexity is significantly lower (the image is darker) than in most of the rest of the image, unveiling the repetitions. If desired, a pixel-based method could be used to fine-tune the results.

5. CONCLUSIONS

The problem of finding unknown patterns that appear multiple times in a digital image is interesting but also challenging. The absence of *a priori* knowledge about the pattern makes the task difficult. The difficulty is visually obvious: examination of the upper image of Fig. 4 does not readily reveal the nature and location of the repetitions (or at least all of them). This is due to the absence of hints concerning texture, shape, intensity or color that could be easily explored.

We took an information-theoretic approach instead of a pixel-based approach. A normal compression method, in the sense explained in [10], that finds a pattern A that has already been found multiple times will be able to represent it with comparatively few bits. The number of bits that the method uses to locally represent a set of pixels is seen to provide an indication of its information-theoretic complexity. But without adequate compression methods this idea would remain useless.

We approached the problem using finite-context models, which have lead to good results. The finite-context model retains information about the image as it scans it, and produces a complexity map. The low-complexity regions of the map contain the repeated patterns. A pixel-based method can then be used to check or fine-tune the regions. No *a priori* knowledge about the shape, texture or color of the pattern is required and the method can find partial repeats, even in areas where it is visually difficult to do so.

The method proposed shows good performance in a number of situations but it has shortcomings. It is unable to deal with noisy copies, and cannot deal with operations such as rotation. However, it represents one first step in the direction of blind repetition finding and we hope to address some of its limitations in the future.

6. ACKNOWLEDGMENT

This work was supported in part by the FCT, Fundação para a Ciência e Tecnologia.

REFERENCES

[1] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text compression*. Prentice Hall, 1990.

[2] C. H. Bennett, P. Gács, M. L. P. M. B. Vitányi, and W. H. Zurek. Information distance. *IEEE Trans. on Information Theory*, 44(4):1407–1423, July 1998.

[3] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[4] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Trans. on Information Theory*, 51(4):1523–1545, Apr. 2005.

Figure 2: On the left, the "Lena" image changed by inserting the same small textured region into four different places. On the right, the logarithm of the complexity surface obtained with the method proposed in this paper, where it can be seen four darker regions corresponding to low complexity occurrences motivated by the repeating pattern.

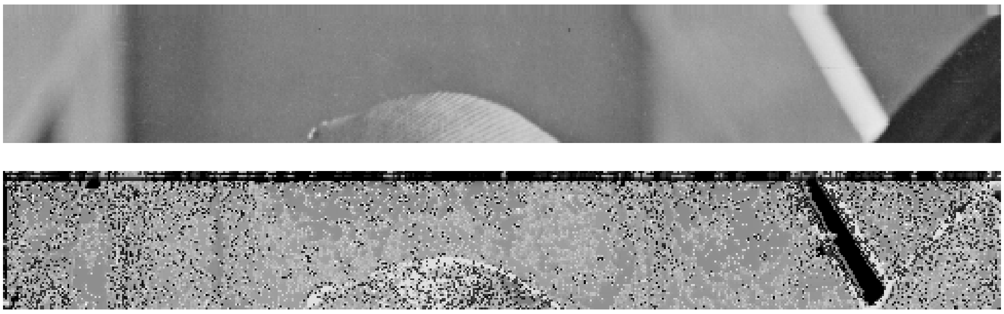

Figure 3: At the top, the first 70 rows of the "Lena" image. Below, the corresponding image of the logarithm of the complexity surface, showing a curious horizontal dark strip, indicating a repetitive pattern that might has been created during some previous manipulation of the image. The oblique strip is also marked as a low complexity region.

[5] T. I. Dix, D. R. Powell, L. Allison, J. Bernal, S. Jaeger, and L. Stern. Comparative analysis of long DNA sequences by per element information content using different contexts. *BMC Bioinformatics*, 8(Suppl. 2):S10, 2007.

[6] I. Gondra and D. R. Heisterkamp. Content-based image retrieval with the normalized information distance. *Computer Vision and Image Understanding*, 111:219–228, 2008.

[7] G. Gordon. Multi-dimensional linguistic complexity. *Journal of Biomolecular Structure & Dynamics*, 20(6):747–750, 2003.

[8] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[9] A. Lempel and J. Ziv. On the complexity of finite sequences. *IEEE Trans. on Information Theory*, 22(1):75–81, Jan. 1976.

[10] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitányi. The similarity metric. *IEEE Trans. on Information Theory*, 50(12):3250–3264, Dec. 2004.

[11] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, 28(2):129–137, Mar. 1982.

[12] A. Mallet, L. Gueguen, and M. Datcu. Complexity based image artifact detection. In *Proc. of the Data Compression Conf., DCC-2008*, page 534, Snowbird, Utah, 2008.

[13] J. Max. Quantizing for minimum distortion. *IRE Trans. on Information Theory*, 6:7–12, Jan. 1960.

[14] A. J. R. Neves and A. J. Pinho. Lossless compression of microarray images using image-dependent finite-context models. *IEEE Trans. on Medical Imaging*, 28(2):194–201, Feb. 2009.

[15] A. J. Pinho and A. J. R. Neves. Lossy-to-lossless compression of images based on binary tree decomposition. In *Proc. of the IEEE Int. Conf. on Image Processing, ICIP-2006*, pages 2257–2260, Atlanta, GA, Oct. 2006.

[16] A. J. Pinho and A. J. R. Neves. L-infinity progressive image compression. In *Proc. of the Picture Coding Symposium, PCS-07*, Lisbon, Portugal, Nov. 2007.

[17] A. J. Pinho and A. J. R. Neves. Progressive lossless compression of medical images. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,*

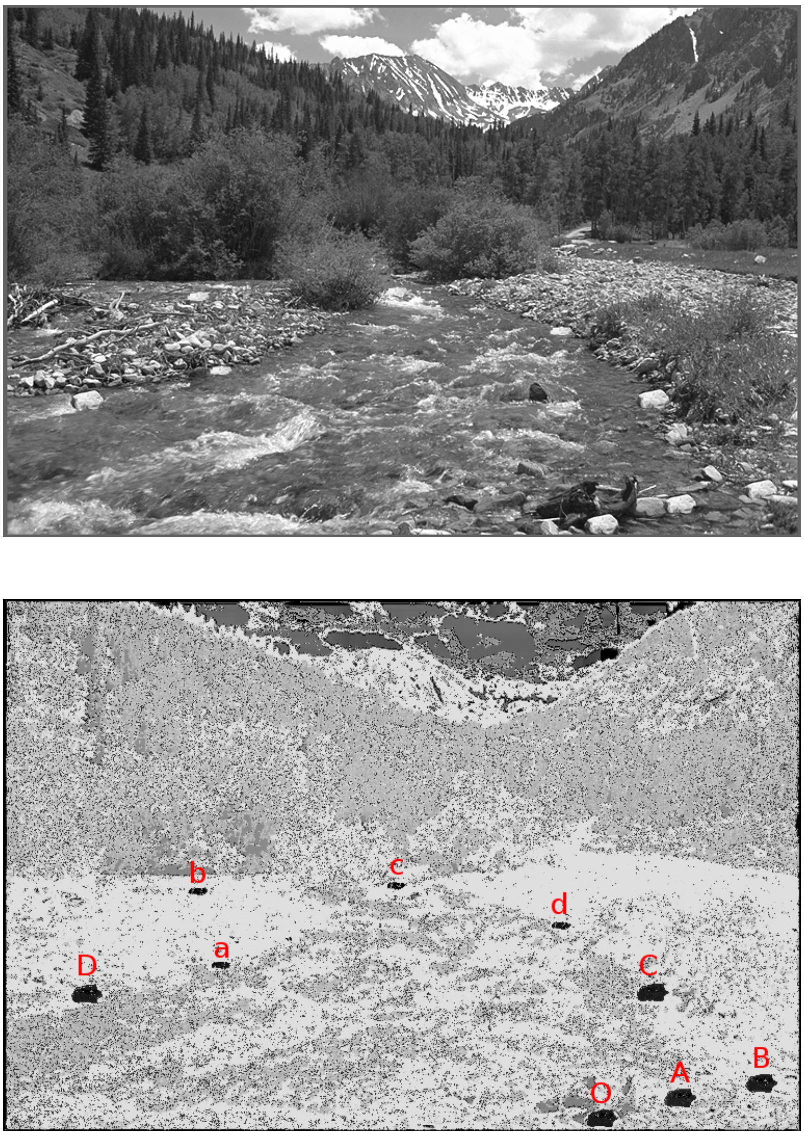

Figure 4: At the top, an image (image "13" from the Kodak set) obtained from the original one by copying a small region, corresponding to one of the stones, into several other places. At the bottom, a representation of the logarithm of the complexity of the image. The label "O" marks the original region that has been copied. The other labels indicate a copy of region O (A, B, C and D) or a partial copy of region O (a, b, c and d).

ICASSP-2009, Taipei, Taiwan, Apr. 2009.

[18] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[19] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7(1):1–22, Mar. 1964.

[20] R. J. Solomonoff. A formal theory of inductive inference. Part II. *Information and Control*, 7(2):224–254, June 1964.

[21] D. S. Taubman and M. W. Marcellin. *JPEG2000: image compression fundamentals, standards and practice*. Kluwer Academic Publishers, 2002.

[22] N. Tran. The normalized compression distance and image distinguishability. In *Human Vision and Electronic Imaging XII — Proc. of the SPIE*, page 64921D, Jan. 2007.

[23] C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, Aug. 1968.

[24] M. J. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Trans. on Image Processing*, 9(8):1309–1324, Aug. 2000.