

Feature Extraction using Fractal Codes

Ben A.M. Schouten and Paul M. de Zeeuw

Centre for Mathematics and Computer Sciences (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Phone: 00 31 20 5929333, Fax: 00 31 20 5924199

B.A.M.Schouten@cwi.nl, Paul.de.Zeeuw@cwi.nl

Abstract. Fast and successful searching for an object in a multimedia database is a highly desirable functionality. Several approaches to content based retrieval for multimedia databases can be found in the literature [9, 10, 12, 14, 17]. The approach we consider is feature extraction. A feature can be seen as a way to present simple information like the texture, color and spatial information of an image, or the pitch, frequency of a sound etc.

In this paper we present a method for feature extraction on texture and spatial similarity, using fractal coding techniques. Our method is based upon the observation that the coefficients describing the fractal code of an image, contain very useful information about the structural content of the image. We apply simple statistics on information produced by fractal image coding. The statistics reveal features and require a small amount of storage. Several invariances are a consequence of the used methods: size, global contrast, orientation.

1 Introduction

Automatic indexing and retrieval of images based on content is a challenging research area. What is called content is usually subjective and often depends on the context, domain, etc. This is the reason why content based access is still largely unsolved. High-level content based retrieval requires the use of domain knowledge and is therefore limited to a specific domain. Low-level retrieval techniques are more generic but they can characterize only low-level information such as color, texture, shape, motion etc.

We are interested in low-level retrieval techniques for grey scale images, based on texture and spatial (dis)similarity. We wish to locate a set of images related to a given image ("Query by example"). Fractal coding is effective for images having a degree of self-similarity. Here similar means that a given region in an image may be fitted to another region using some affine transformation. This notion of similarity is particularly useful for textured regions. We will make a distinction between three main aspects of texture: symmetry, contrast and coarseness [15]. The spatial similarity features will be based on spatial relationships, like the distance and the angle between the similar regions.

The power of the method lies in the multiresolution nature: retrieval of high-resolution database images with low-resolution original inexact queries is possible. Retrieval systems for very large databases impose strong demands on the size of the feature vectors, the effectiveness of the indexing techniques, and the efficiency of the searching algorithm. Therefore the features should be simple to compute and be discriminating. It is necessary to develop hierarchical indexing and searching strategies, that is, in subsequent steps one performs an increasingly detailed search on a smaller and smaller subset of the database. By their multiresolution character fractal coding techniques are apt to the construction of hierarchical image indexing and searching schemes.

This paper is a first survey on how effective feature extraction based on fractal codes can be. The need for features with discriminating potential and the possibilities offered by hierarchical schemes in this respect gave reason to write this paper.

2 Background

2.1 Fractal image coding

Fractal coding is a relatively new technique which emerged from fractal geometry. It has been studied thoroughly by several authors, see e.g. [4, 13]. Fractal coding is based on the self-similarity in a picture. This means that small pieces of the picture can be approximated by transformed versions of some other (larger) pieces of the picture. This phenomenon is exploited to extract features that relate to this self-similarity.

We give a brief introduction to fractal image coding, cf. [2, 6, 3]. Without loss of generality, we suppose that the image I measures $2^N \times 2^N$ pixels. We will denote this image area with E . We consider grey scale images and define $G = \{0, \dots, 255\}$. So:

$$I : E \rightarrow G, \quad I \in G^E,$$

$I|_R$ is the restriction of the image to the region R and $I_R := I|_R$.

In fractal coding the image I is partitioned into non-overlapping sub blocks of fixed size, called range blocks. See Figure 1 (courtesy of Dugelay et al. [3]). The fractal encoder searches for every range block R , another block in the image (domain block D) that looks similar under an affine transformation. The range blocks are identified by the coordinates of the lower left corner of the block

$$\mathcal{R} = \{(2^d m, 2^d n) \mid 0 \leq m, n \leq 2^{N-d} - 1, m, n \in \mathbb{Z}\}.$$

The goal of the expression scheme is to approximate, within a certain tolerance ϵ , the range block R by a certain domain block D of double size: $2^{d+1} \times 2^{d+1}$. The chosen domain block is extracted from a domain pool \mathcal{D} . There are several kinds of domain pools; for our survey we use the half overlapping domain pool

$$\mathcal{D} = \left\{ \left(2^{d+1} \frac{m}{2}, 2^{d+1} \frac{n}{2} \right) \mid 0 \leq m, n \leq 2^{N-d} - 1, m, n \in \mathbb{Z} \right\}.$$

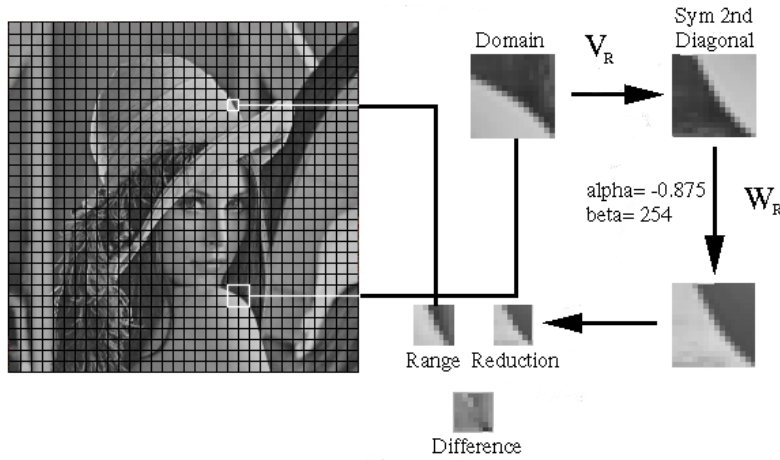


Fig. 1. Fractal coding in steps

The approximation of the range block by the domain block is done in several steps:

- The domain block is brought into position by a symmetry operator V_R .
- The grey values of the domain block are tuned by an operator W_R . W_R consists of a contrast scaling α and a luminance offset β .
- The size of the domain block is reduced with 75 % (averaging, down sampling).

The essential operator in the scheme is W_R . α within W_R is chosen in such a way that W_R is a contraction mapping. The other operators are used to make more fits possible. W_R is an affine mapping of grey values.

$$W_R : G \mapsto \mathbb{R}$$

Given a range block R the coder searches for a domain block $D_R \subset E$ and an affine mapping W_R such that according to the l_1 metric on G :

$$d(W_R(I_D), I_R) \leq \epsilon \quad (1)$$

In the scheme the above procedure is repeated for all range blocks $R \in \mathcal{R}$. Then, the original image I is by approximation a fixed point for the map W :

$$W = \bigcup_{R \in \mathcal{R}} W_R$$

By the Fixed Point Theorem the image can be restored by iterating W in the decoding phase, starting with any picture. This implies that storage of the parameters of the map W is sufficient for the (near) reconstruction of the image.

2.2 Quadrees and multiresolution

Most fractal coding schemes use a quadtree as a further subdivision of the image. In the first stage of the coding, the image is partitioned into range blocks of fixed size. According to the tolerance ϵ (1) there will or will not be a match between a range block and a domain block. This means, there are:

1. *successes* i.e. range blocks for which an approximation by a domain block has been found and,
2. *failures* i.e. range blocks for which no approximation could be found.

The procedure in fractal coding is to subdivide the failures into four sub blocks of $1/4$ size. The search for successes will then start again; now only with range blocks of $1/4$ size.

This "multiresolution" scheme is illustrated in Figure 2. The first level of the

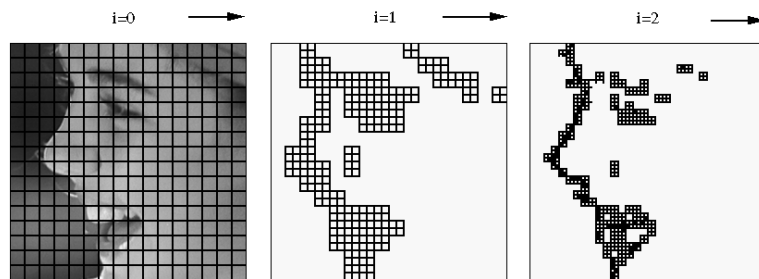


Fig. 2. Subdivision by a quadtree; failures at level i are illustrated at level $i + 1$.

quadtree, $i = 0$, contains range blocks of a fixed size, partitioning the image. The failures at a certain level i are divided into four sub blocks and illustrated at the next level $i + 1$. The number of failures per level i is an important feature, we denote it by f_i . For convenience we also define s_i as the number of successes at level i .

3 Feature extraction

Today there exists several implementations in multimedia database systems such as the QBIC system by IBM [10], Photobook developed by the MIT Media Lab [12], and the Virage system developed by Virage Inc. In these, as well as many other approaches, one defines feature vectors of image properties. It is essential that such feature vectors are much smaller in size than the original images, but represent the image content as accurately as possible. Images are considered to be similar if the distance between their corresponding feature vectors, which are supposed to be elements of a given metric space, is small. For this reason, the

discriminating power of the features has to be strong.

Features often used are color and texture [10, 12]. Furthermore, several authors have suggested to use shape properties [12], or relative position of objects within an image [7, 5], called spatial similarity.

3.1 Textural and spatial similarity

Texture in images has been recognized as an important aspect of human vision. Fractal image coding has good results in coding textures with the exception of statistical texture which is far from ideal [11]. However, we like to stress an important issue: compressing an image and featuring an image are quite different goals. There is no a priori reason why a transform used for indexing multimedia databases has to satisfy the same properties as one for compressing images. It can be argued that the disability of fractal coding to handle statistical texture is an advantage.

Here we are dealing with low-level feature extraction, without any segmentation. We like to show that fractal coding can model spatial info without segmentation and create several features for spatial similarity. In these features we like to express whether similarity between regions is bounded to a certain part of the image. Or whether there is a dominating direction between the blocks that are similar. The extracted information is modeled in a way that is independent of the size of the image; a very desirable item. Smaller thumbnails can then be used to retrieve bigger images.

4 Feature extraction using fractal codes

In our experiments the coding scheme was programmed to use five quadtree levels. At the first step every image is divided into 16 range blocks, regardless the size of the image. At every level of the quadtree several features will be extracted and with this more information about the image is added at every level of the quadtree.

4.1 Texture

We like to distinguish three features for texture: symmetry, contrast and coarseness.

The **symmetry** feature is modeled by the operator V_{R_i} , see Figure 1. V_{R_i} relates a range block to one of the 8 symmetry operators that are used to bring a domain block into position to match this range block at a certain level i . In our experiments we will make histograms of several features. In the symmetry histogram, horizontally the 8 symmetry operators are denoted. The vertical axis shows the fraction by which the various symmetry operations occur at that level of the quadtree, see Figure 4.

Homogeneity of textural **contrast** is modeled by the mean and variance of the grey value scaling α . If a domain block, at a certain level i is matched with a range block, α is the scaling used on the grey values of the domainblock. Again all features are related to i .

The **coarseness** of texture is modeled by the number of successes s_i at a certain level of the quadtree. So we count the number of ranges for which a proper domain block has been found at a certain level of the quadtree. The depth appears to be very important in this respect. If a lot of large domain blocks can be mapped onto range blocks, the scale of the similarity will likely be coarse.

4.2 Spatial similarity.

For the spatial (dis)similarity present in an image, three features are derived from the fractal code depending on the quadtree level: uniformity, direction and dimension.

Uniformity and direction. The first two spatial features are modeled by one vector c_i , depending on the level i of the quadtree. c_i is expressed in terms of its magnitude l_i and angle ϕ_i :

$$c_i = (l_i, \phi_i)$$

l_i measures the distance between a matched range and domain block. The perception is, that l_i is bounded if an image consists of different textures dividing the image into several regions. In our histograms this feature will be divided into 8 classes; length will be calculated as fraction of the distance from lower left corner to upper right corner of the image. In this way the feature is made size invariant.

The spatial **direction** feature (see Figure 3) measures the angle between the horizontal direction and the direction from upper left corner of the domain block to upper left corner of the range block at a certain level i . We choose to represent

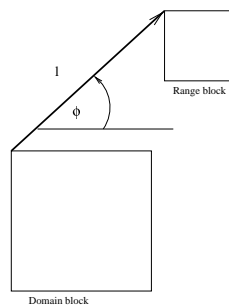


Fig. 3. Spatial uniformity and direction.

these features numerically by vectors $\in \mathbb{R}^8$, and graphically by histograms with eight bars, see Figure 4.

The spatial **dimension** feature relates to the *Box Counting Dimension* [4]. The image I is divided into 16^2 sub blocks. For each sub block we define:

$$d_i = 2 \log \frac{f_{i+1}}{f_i}$$

where f_i is the number of failures at level i . Figure 5 serves as an example. This feature distinguishes between images that have edges scattered all over the image, and images with a few clear-cut lines.

5 Results

In this section we investigate the discriminating power of some features. Here we selected three features: textural symmetry, textural coarseness and spatial uniformity. The example images stem from the Vistex Database of MIT. Figure 4 shows 15 pictures and the corresponding histograms with respect to the selected features.

The first column shows the picture itself; the second column shows histograms related to textural symmetry; the eight values at the horizontal axis correspond to the eight symmetry operators that can be distinguished for mapping domains onto ranges. The first four values denote rotation of a square part over 0 (identity), 90, 180 and 270 degrees respectively. The second quartet denotes the same, but with an additional flip of the plane in which the image lies. The vertical axis shows the fraction by which the various symmetry operations occur at that level of the quadtree. Although all features can be extracted at all levels, we present only the histogram which relates to the level which numbers the most successes, see Section 2.2.

We observe clearly a preference for the 0 and 180 degrees classes in the "Building" images. Other images have a much more even spread over the symmetry operations. Apparently there is a dominating direction in the picture, as could be expected. The feature appears to distinguish between images of man-made and natural environment.

The third column shows histograms with respect to textural coarseness. The horizontal axis of this histogram corresponds to the depth of refinement in the quadtree of the encoding procedure. The vertical axis shows the accumulated success rate of the encoding. It is the fraction of all pixels in the original image that are successfully mapped from domain blocks onto range blocks.

We observe how the "Metal" image is poorly matched even at large depths, due to the statistical texture present in the image. The "Clouds" image mainly seems to consist of similarity at large scale. "Kiss" which has clouds as a background almost shows the same histogram. "Buildings.0008" and "Buildings.0009"

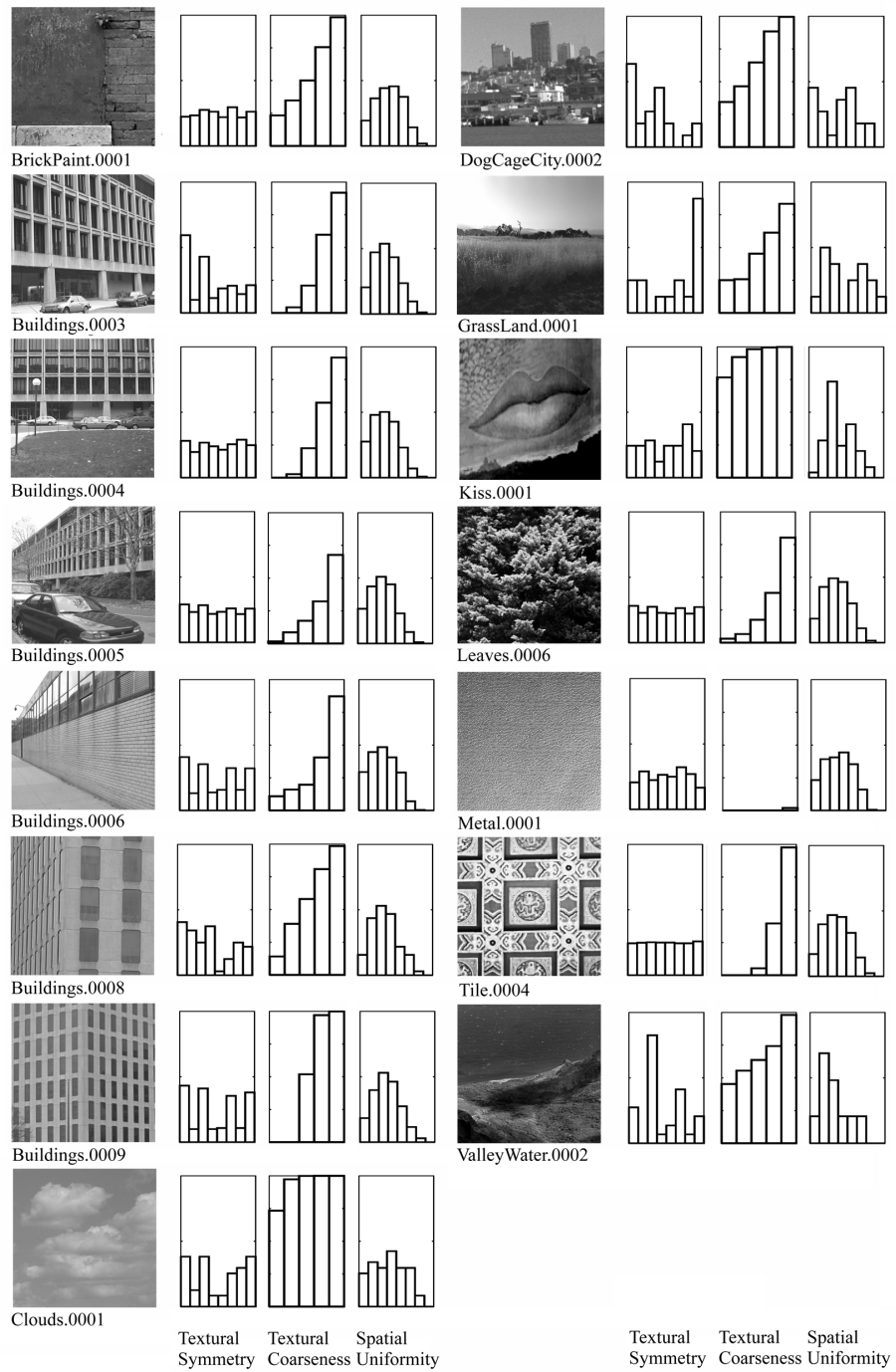


Fig. 4. Pictures from M.I.T. Database and some features.

consist of building structure at a larger scale which is reflected by this feature.

The fourth column shows the histogram with respect to spatial uniformity. The horizontal axes shows 8 classes for the distance between matching domains and ranges.

"Kiss" is a nice example of an image that has texture centered in the image, which is reflected in a biased distribution with a preference for short distances. For "DogCageCity", "GrassLand" and "ValleyWater" the histogram shows two superposed distributions, corresponding to the two main textures.

Finally, we show two examples of the spatial dimension feature. Figure 5, exploits the fractal dimension, times 10, of parts of an image at a certain depth. Typically lines and borders yield $d_i \approx 1$ and areas with lots of inner structure

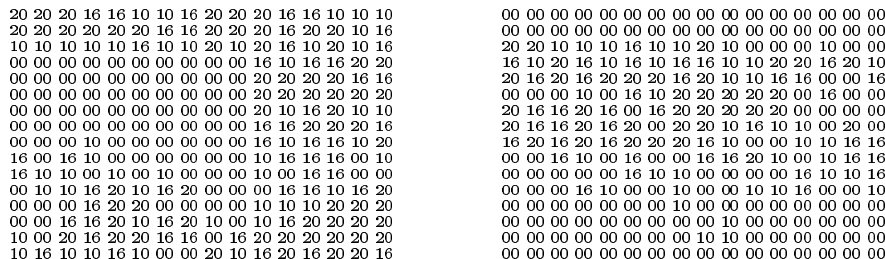


Fig. 5. Spatial dimension (x10); "BrickPaint.0001" and "DogCageCity.0002" images.

yield $d_i \approx 2$. We observe how it is roughly similar to the geometry and topology of the original images and indeed borders and inner areas can be identified by different values of d_i .

6 Conclusions and further research

The features we studied appear to have discriminating power that relates to human vision. Next question is, whether this discriminating power can be used to successfully retrieve an image from a large database. We plan to address this question. The method can be used for a hierarchical search and it combines several desirable options. The first to mention is: feature extraction and compression. The second is that this method has proved to be invariant to size, orientation, contrast scalings and luminance offsets. Therefore our method may improve on previous approaches [9, 14, 17].

7 Acknowledgments

We like to thank Prof. M.S. Keane and Dr. H.J.A.M. Heijmans for their advice.