Fractal-Code Based Image Retrieval System

Arif Rahman[1], Dwi H. Widyantoro[2]
[1]Computer Science Department, Ahmad Dahlan University, Jl. Prof. Soepomo, Yogyakarta 55164, Indonesia
[2]School of Informatics and Electrical Engineering, Institut Teknologi Bandung, Jl. Ganesha 10, Bandung 40132, Indonesia
[1]arif-r@telkom.net, [2]dwi@if.itb.ac.id

***Abstract*—Fractal code is a code generated from image encoding process using Partitioned Iterated Function System (PIFS) method. The characteristic of PIFS is self-similarity searching between image parts based on pixel intensities value. Thus it can be assumed that similarity of fractal code represents similarity of pixel intensities variance pattern, which represents the image texture. In this paper, fractal code-based image retrieval system so-called SPECKTRAL is proposed. Fractal code in this system consists of two components that are invariant to translation, scale and rotation, i.e., contrast and brightness coefficients. Fractal code similarities between two images are measured with the mean value of minimum Euclidean distance between each contrast and brightness coefficients. If the average distance is smaller, then the image considered has more similar texture. The experiment results on Brodatz and Vistex image collections show that the system can retrieve most images that have relevant texture to the query and performs better for textural image collection.**

***Index Terms*—image retrieval, fractal code, content based, texture based.**

I. Introduction

In its early development, Image Retrieval System uses text to annotate the images based on the descriptions of their content. Those texts are stored on database for retrieval process. The annotation is conducted manually, so it is inefficient for huge number of image collection. Another disadvantage is that annotation depends on perception of the annotator [6] .

To solve those problems, Content–Based Image Retrieval (CBIR) system is proposed. A CBIR retrieves images based on their visual content i.e. color, texture or shape. Those visual contents are represented by the system as feature vectors. To retrieve the images, user provides the image query, then the system compares the feature vector of the image query to those in the database. Retrieved images are ranked based on their similarity value [3].

In this paper we proposed texture based image retrieval that uses fractal code as a feature representing image texture. Fractal code is a code from image encoding process using Partitioned Iterated Function System (PIFS) method. The characteristic of PIFS is self-similarity searching between image parts based on pixel intensities value. A texture is defined as variations of pixel intensity and color that form certain repeated patterns. Thus it can be assumed that similarity of fractal code represents similarity of pixel intensities variance pattern that represents the image texture.

Researches on fractal code-based image retrieval system have been conducted. In [9], Zhang explains that if given two identical images I_1 and I_2 (each of them is partitioned into range and domain block), any two identical ranges of blocks of I_1 and I_2 must have the identical domain block that provides the best self-transformation. Thus, their fractal codes must be identical, and, consequently, and so do their entire images. Then, Zhang [9] uses fractal code in clustering image collection with iconic image. Cluster construction is based on similarity between fractal code of image collection and fractal code of iconic image. The similarity is measured based on Joint Fractal Coding scheme. If I is iconic image and M is image in the collection then the more domain is in M, which is selected in fractal coding for range in I, the more similar is M to I. Alternative approach is proposed by Nappi [5]. In *Fractal Indexing and Retrieval SysTem (FIRST)* he uses histogram of fractal code appearance frequency in the image. The histogram is represented as a vector and then similarity degree determined by Euclidean distance value between appropriate vector components.

II. Theoretical Background

A. Fractals and Iterated Function System

Fractal is coined by Benoit Mandelbrot. The term fractal (originated from Latin word *fractus*) means fracture or creating irregular form [4]. With fractals we can describe natural objects that have irregular form like clouds, leaves, rocks, etc. in a small number of parameters. That could not be done with common Euclidean geometric approach.

In fractals, objects can be broken into parts. Each part is a copy of reduced object itself. That is called self-similarity.

One of the methods to construct fractal objects is Iterated Function System (IFS). An IFS is a collection of affine transformation that represents relation between object parts. Affine transformation is a point transformation in n-dimensional axis coordinates as expressed in Equation 1.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (1)$$

In IFS, an object is created by applying transformations to the initial object iteratively [1] as demonstrated in Sierpinski triangle reconstruction. Figure 1 shows this, where the leftmost image is the initial object.

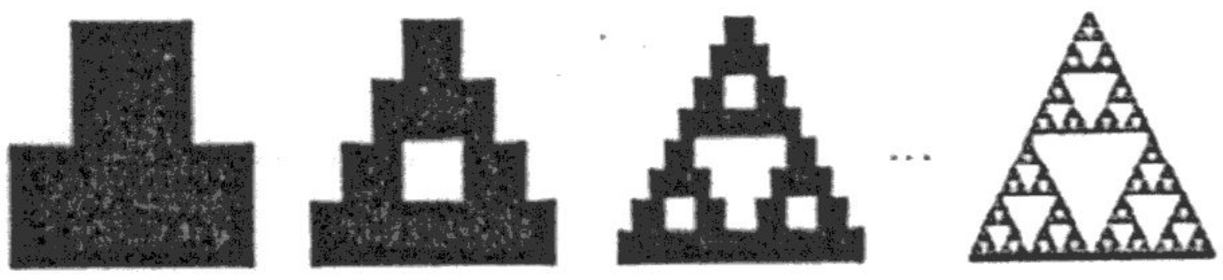

Figure 1. Sierpinski triangle construction

A small number of affine transformation parameter can be used to describe an object in IFS. Given an object, can we find those parameters? This question leads to image compression with IFS, since if it is answered, the image can be expressed in a compact form. But that problem is difficult to solve.

B. Partitioned Iterated Function System

Another solution proposed by Arnaud Jacquin with modification of IFS is called PIFS (Partitioned Iterated Function System). Image encoding with PIFS is performed by building partition in image and finding another part of image similar to that partition. A partitioned image is called Range (R_i) and another part that is similar is called Domain (D_i). Because a part of image is not exactly the same, then before comparing them a w_i transformation is done to D_i as shown in Figure 2.

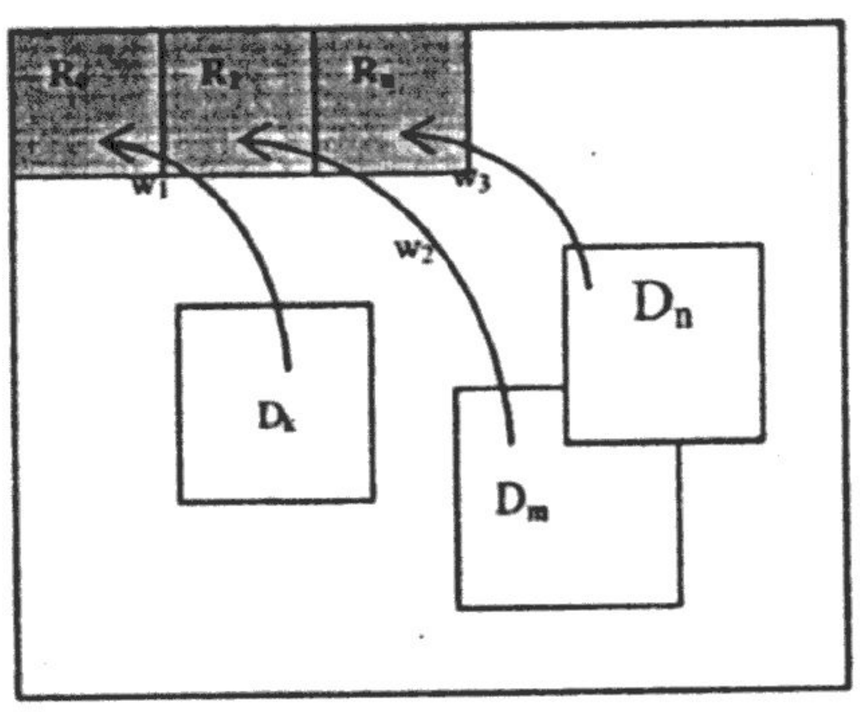

Figure 2. Domain transformation to appropriate range

w_i is transformation of position and pixel value in gray scale image as given by Equation 2.

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (2)$$

z is intensity value of pixel on coordinat (x, y) in the image. The a, b, c dan d coefficients relate to pixel's position transformation , while s and o relates to pixel's contrast and brightness. Those coefficients become the fractal code.

To encode image f is to find a collection of transformation $w_1, w_2, \ldots, w_N$ with $W = \bigcup_{i=1}^{N} w_i$ and fixed point of W in $f = |W|$. So $f = W(f) = w_1(f) \cup w_2(f) \cup \ldots w_N(f)$. Since image is not composed of pieces that can be transformed to fit exactly somewhere else in the image, what we can hope to find is another image $f' = |W|$ with small distance $d(f', f)$ [2]. Therefore, we seek a transformation W whose fixed point is close to f. In that case,

$$f \approx f' = W(f') \approx W(f) = w_1(f) \cup w_2(f) \cup \ldots w_N(f) \quad (3)$$

Thus it is sufficient to approximate the parts of the image with the transformed pieces. It can be done by minimizing the following quantities

$$d(f \cap R_i, w_i(f)); i = 1, 2, \ldots N \quad (4)$$

We seek D_i and transformation w_i in such that if w_i applied to D_i we get the approximation of R_i. Minimizing Equation 4 means two things: (1) finding a good choice for D_i, and (2) finding good contrast and brightness settings s_i and o_i for w_i.

In practice, comparison of domain and range is using Root Mean Square (RMS) metric. Using this metric also allows computation of optimal values for s_i and o_i. Given two blocks containing n pixel intensities, i.e, a_i , ... a_n from D_i and $b_i, .. b_n$ from R_i , we can seek s and o to minimize the quantity in Equation 5.

$$E = \sum_{i=1}^{n} \left[(s \cdot a_i + o) - b_i \right]^2 \quad (5)$$

This will find contrast and brightness settings that make the transformed a_i values have the least squared distance from b_i values. The minimum E occurs when the partial derivatives with respect to s and o are zero, which occurs when

$$s = \frac{\left[n^2 \left(\sum_{i=1}^{n} a_i b_i \right) - \left(\sum_{i=1}^{n} a_i \right) \left(\sum_{i=1}^{n} b_i \right) \right]}{\left[n^2 \sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i \right)^2 \right]} \quad (6)$$

and

$$o = \frac{\left[\sum_{i=1}^{n} b_i - s \sum_{i=1}^{n} a_i \right]}{n^2} \quad (7)$$

In that case,

$$E = \frac{\left[\sum_{i=1}^{n} b_i^2 + s\left(s \sum_{i=1}^{n} a_i^2 - 2 \left(\sum_{i=1}^{n} a_i b_i \right) + 2o \sum_{i=1}^{n} a_i \right) + o\left(on^2 - 2 \sum_{i=1}^{n} b_i \right) \right]}{n^2} \quad (8)$$

If the denominator s is 0 then s = 0 and $o = \sum_{i=1}^{n} \frac{b_i}{n^2}$ D_i with smallest $E^{1/2}$ will be selected from the domain set **D**.

Domain should has larger size than range and $s < 1$ to ensure transformation is contractive and convergen. In practice, domain size is four times larger than range, that is two times larger for each side [8]. Given grayscale image with size 256 x 256 pixels. Let $R_1, R_2, \ldots, R_{1024}$ is 8 x 8 pixels of non overlapped blocks in the image and **D** is a set of all 16 x 16 pixels of overlapped blocks as in Figure 3. **D** contents is (256-16+1) x (256-16+1) = 58.081 blocks since each block overlaps by 1 pixel. For each R_i we seek $D_i \in \mathbf{D}$ that minimizes Equation 3. Since D_i is four times larger than R_i, then D_i should be subsampled by taking the mean value of each 2 x 2 pixels block in D_i

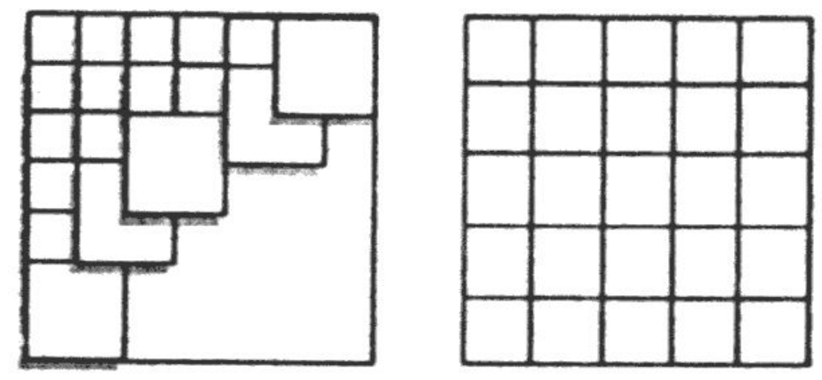

Figure 3. Domain set (left) and range partition (right)

III. System Overview

In this paper we proposed a fractal code-based image retrieval system called SPECKTRAL. This system finds images in the collection that have similar texture to the query image. The similarity comparison is based on their fractal-code. The Retrieval process in this system is based on Query By Example (QBE) method. The query image is encoded and then is compared to image codes in the database. In the end of retrieval process, images with similar code are ranked based on their similarity values. The retrieval process is shown in Figure 4.

Only two of six affine transformation coefficients are employed to build fractal-code in this system, i.e s and o. Those coefficients are selected since they are invariant to translation, scaling and rotation, which are not relevant to the similarity of texture. Fractal code similarity between two images in this system is measured with the mean value of minimum Euclidean distance between each contrast and brightness coefficients. If the average distance is smaller, then the image is considered has more similar texture. To retrieve images efficiently this system employes an index of image collection. The index is constructed using inverted file method, a commonly used method in text retrieval systems [7]. Inverted file consists of fractal-code as the term in text retrieval context and a pointer to the image document in which the code appears.

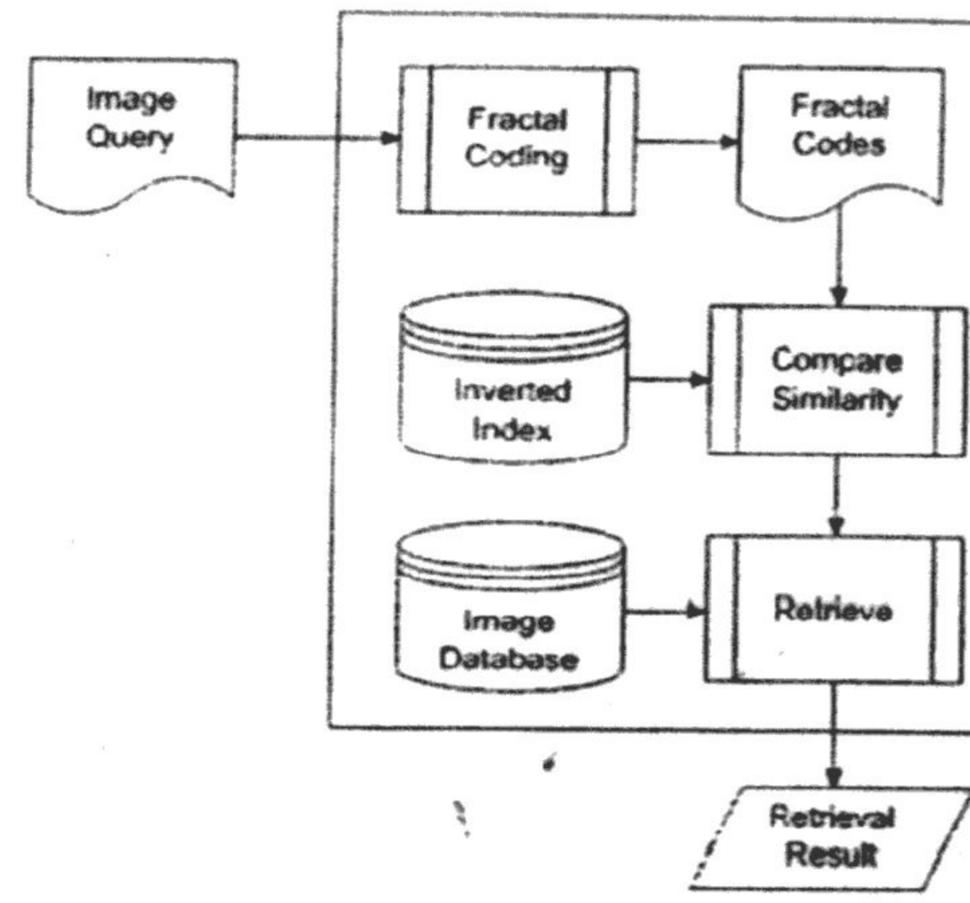

Figure 4. Retrieval Process

A. Fractal Code Construction

In this system, fractal-code is generated from gray-scale image. A color image must be converted to gray-scale first before encoding it. The conversion process is done by firstly converting the color space model of the image to YIQ color model. Then Y component is selected as pixel intensity value of the gray-scaled image [2].

Range and domain block size is 8 x 8 pixels and 16 x 16 pixels respectively. The domain is overlapped by the same size as range, i.e., 8 pixels. This size is selected since it can reduce the number of domains in the encoding process without degrading its fidelity significantly, especially for similarity (not exactly) searching purpose.

B. Similarity Measurement

The difference value between s and o coefficients in the fractal codes of two images is calculated by Euclidean distance function. We seek minimum average distance between that code and each code in the image document (see Equation 9).

$$dist(Q,D) = \frac{\min_{q \in Q, d \in D}\left(\sqrt{(s_q - s_d)^2 + (o_q - o_d)^2}\right)}{|Q|} \quad (9)$$

In Equation 9, Q is query image, D is image document, s_q, o_q are contrast and brightness coefficient of Q. s_d, o_d are contrast and brightness coefficient of D and $|Q|$ expresses the number of codes in Q. $0 \leq s_q, s_d < 1$, while $0 \leq o_q, o_d \leq 255$, thus the difference value between s_q and s_d is not significant compared to that between o_q and o_d. That is s_q and s_d should be multiplied by the maximum values of o_q and o_d, i.e. 255.

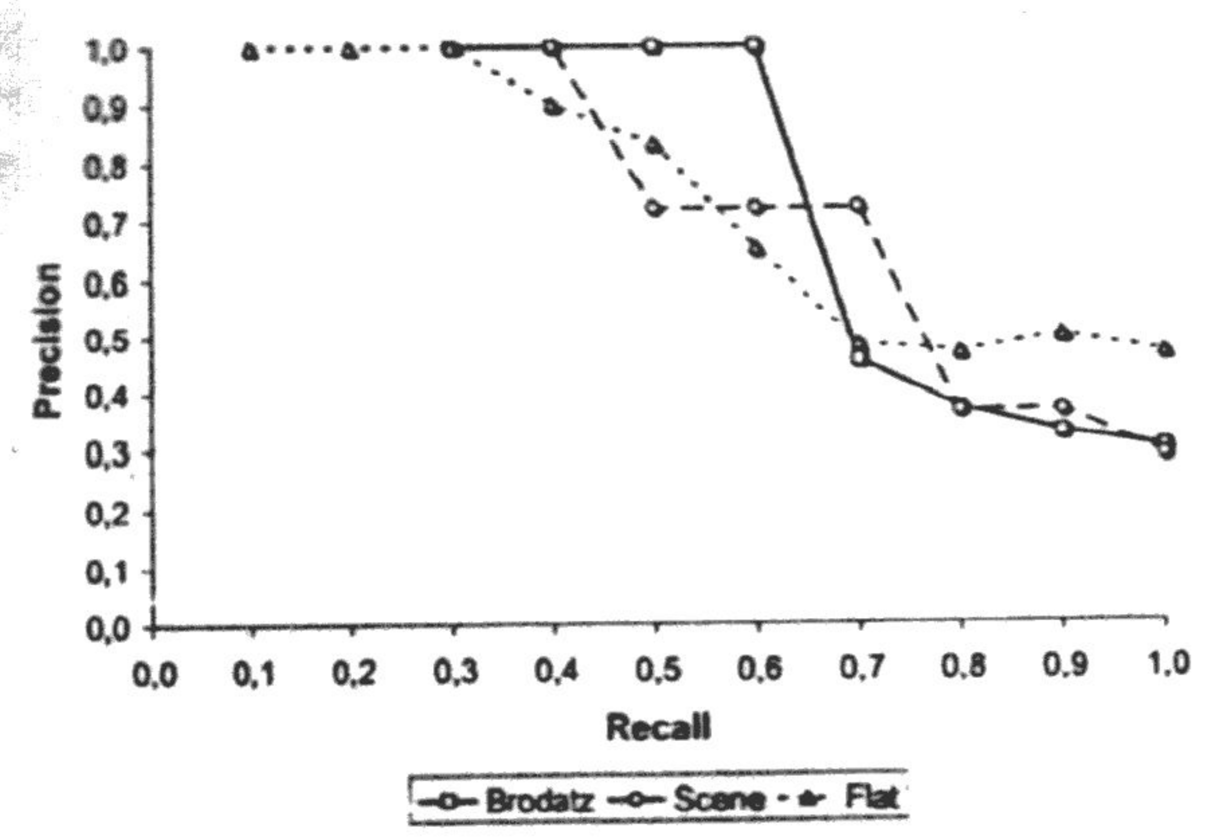

Figure 5. Precision-Recall graph for 25% of collection retrieved

IV. EXPERIMENTAL RESULTS AND EVALUATION

The SPECKTRAL software is developed using Java Servlet running in Apache Tomcat 5.0.28 web server and MySQL 4.1 database server. Experiments are conducted on three different image collections. The first is Brodatz that contains textural images (39 images at 128 x128 pixels). The second collection is Vistex-Scene containing scene images (60 images at 128 x128 pixels). The third collection is Vistex-Flat, which contains images of objects (167 images at 128 x128 pixels). Figure 6 shows an example of retrieval result for image Buildings.0001 on Vistex-Flat collection with top ten ranked images, sorted from left to right.

Precision and recall are used to evaluate the performance of SPECKTRAL. Precision is the portion of retrieved images that are relevant, while recall is the portion of relevant images that are retrieved. Figure 5 shows the results of the experiment. The graph shows, the system particularly on lower recall will produce better precision on Brodatz collection, confirming that the fractal encoding is good at identifying textural image.

V. CONCLUSION AND FURTHER RESEARCH

The experimental results demonstrate that fractal-code describing the contrast and brightness coefficient can be used as feature vector for the image retrieval based on its texture. From the SPECKTRAL system perspective, additional research and experiments need to be done to cluster images automatically and adaptively based on broader image category. In this setting, the retrieval process is conducted only on images in the same cluster in order to get better search result precision.

REFERENCES

[1] Barnsley, M. F., Sloan, A. D., Jan. 1988, *A Better Way to Compress Images*, BYTE Magazine.

[2] Fisher, Y. *Fractal Image Compression: Theory and Application, Springer-Verlag*, NY, 1995.

[3] Long F., Zhang H., Feng D., *Fundamentals of Content-based Image retrieval*, in Multimedia Information Retrieval and Management - Technological Fundamentals and Applications, D. Feng, W.C. Siu, and H.J.Zhang. (ed.), Springer, 2002

[4] Mandelbrot, B. B.. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1982.

[5] Nappi M, G. Polese, G. TortoraDing, W., and Marchionini. *FIRST: Fractal Indexing and Retrieval SysTem for Image Databases*, Image and Vision Computing 16, 1998, 1019–1031.

[6] Rui Y., Huang T.S, Chang S., *Image Retrieval: Past, Present and Future*, Journal of Visual Communication and Image Representation, Vol. 10, pp. 1 - 23, 1999.

[7] Witten, I.H, Moffat, A., Bell, T.C. *Managing Gigabytes : Compressing and Indexing Documents and Images*, Academic Press, 1999.

[8] Wohlberg, B., de Jager , G., 1999, *A Review of the Fractal Image Coding Literature*, IEEE Transactions on Image Processing vol .8, no.12

[9] Zhang, A. Cheng, B. Acharya, R., *A Fractal-Based Clustering Approach in Large Visual Database Systems*, Multimedia Tpols And Applications, vol 3; number 3, pages 225-244, 1996.

[10] Zhang A., Cheng, B; Acharya, R.S., *Texture-Based Image Retrieval Using Fractal Codes*, Dept. of Computer Science, State University of New York at Buffalo,, 199

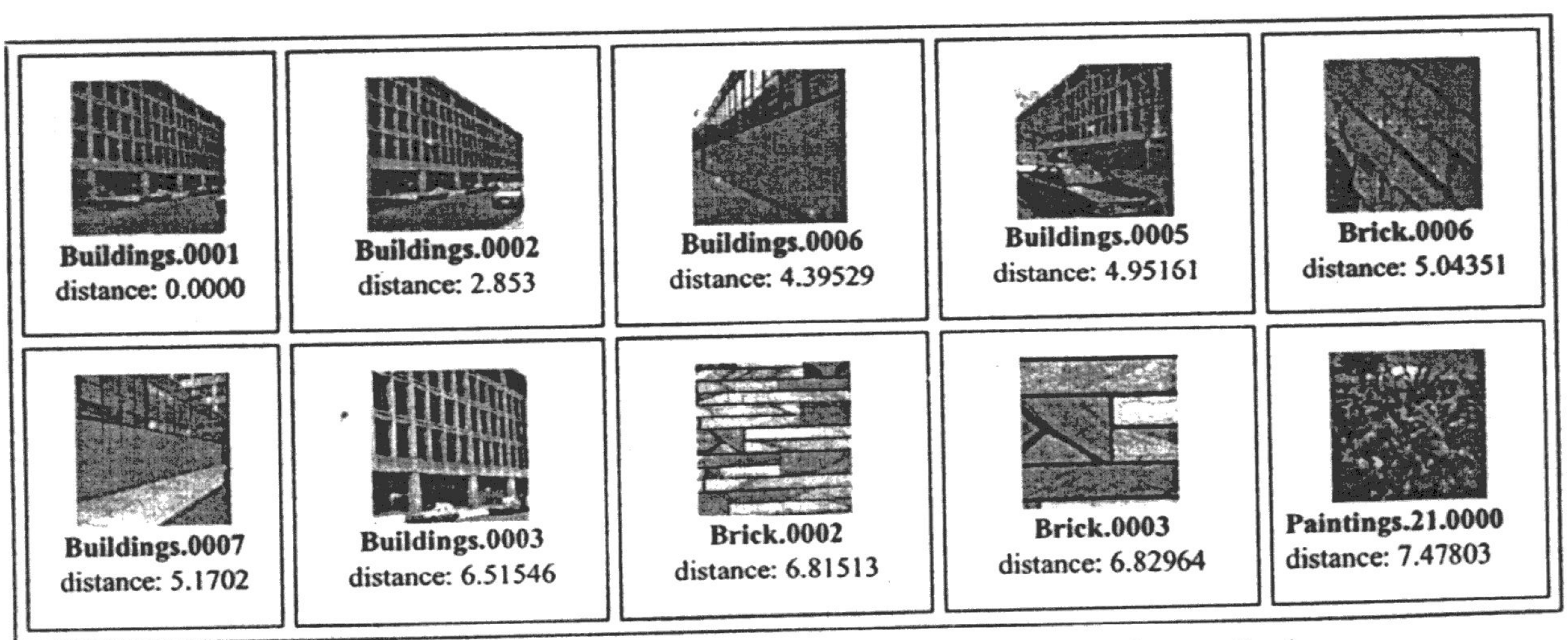

Figure 6. Query result for image Buildings.0001 (upper left) on Vistex-Flat image collection