VOL. 9, NO. 7, JULY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.
www.arpnjournals.com

AN EFFECTIVE IMAGE RETRIEVAL METHOD BASED ON FRACTAL DIMENSION USING KERNEL DENSITY ESTIMATION

Zhang Qin, Huang Xiaoqing and Liu Wenbo
College of Automation Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China
E-Mail: Daisyzhangq@126.com

ABSTRACT

Fractal coding has been proved useful for image compression, and it is also proved effective for image retrieval. In the paper, we present a statistical method called variable bandwidth kernel density estimation to analyze fractal coding parameters. Then retrieve images using the retrieval index constructed with this method. Experimental results show that the proposed method with a variable optimized bandwidth performs better than those with a fixed bandwidth and the histogram method both in retrieval rate and retrieval speed. In this paper, the Average Retrieval Rate (ARR) can reach 72.40%, which is more than that obtained by the existing methods.

Keywords: average retrieval rate, image, fractal coding.

INTRODUCTION

Image retrieval has been an active research area for years; there are various kinds of image retrieval technique mainly based on text, content and semantic. The Content-Based Image Retrieval (CBIR) technique [1] is used to retrieve images with image features directly, so we can find the most similar images from the database through the comparison between the image features. Fractal coding parameters can effectively represent essential features of images. Fractal coding, as a new image compression technique, has been applied into image retrieval. Fractal features provide geometric information of an image that is irrelevant to the shape and size of an object in the image, therefore, fractal features are more robust than color and texture features. Meanwhile, retrieving images in fractal domain can be faster and more effective, especially for the compressed images.

Fractal image coding is a block-based scheme that exploits the self-similarity hiding within an image. Fractal features generated by the block-based scheme are quantitative measurements of self-similarity; therefore they can be used to construct image features. Fractal image compression was originally developed by Barnsley and Sloan [2]. Jacquin [3] implemented a block-based fractal compression, which is popularly known as fractal block coding. And fractal block coding has been applied into image retrieval. Pi MH proposed to employ the histogram of range block means and the 2D joint histogram of range block means and contrast scaling parameters as an image index [4, 5], and this technique greatly improved the retrieval rate. Some scholars proposed the histogram of collage error as an image signature and combined fractal parameters with collage error to improve the retrieval rate [6].

The features of an image can be acquired effectively with the statistical characteristics of fractal coding parameters, and the performance in image retrieval has already been confirmed. A statistical method called kernel density estimation is proposed, which can estimate the density of samples more accurately. Compared with the commonly used histogram method, the kernel density estimation can be more accurate and smooth. Therefore, we apply this method into image retrieval. Since the bandwidth of kernel function plays an important role in kernel density estimation, we propose the method with a variable optimized bandwidth in conformity with data [7]. Experimental results show that this method has not only higher retrieval rate but also less retrieval time than the existing methods.

The rest of the paper is organized as follows. The section 2 introduces fractal coding and collage error. Section 3 presents the proposed method. Section 4 presents the performance evaluation. The section 5 presents conclusions and future work.

FRACTAL CODING AND COLLAGE ERROR

Fractal coding

In this paper, the orthogonalization fractal coding method is adopted [8]. An image $(M \times M)$ is first segmented into non-overlapping blocks of size $B \times B$ called range blocks, recorded as $R_1, R_2, \ldots R_m$. A domain block pool Ω is a set of domain blocks of size $2B \times 2B$, generated by dividing the $(M \times M)$ image into overlapped blocks. And the domain blocks are recorded as $D_1, D_2, \ldots D_n$. In general, $B = 2^t$, t is an integer. After the 4-neighborhood pixel average and compression transform, the domain blocks are mapped into the images with size $B \times B$. To improve the quality of the images, eight kinds of isometric transform are applied into the domain blocks. (In Jacquin's scheme, rotation transformation of 0°, 90°, 180°, 270°, vertical midline, horizontal midline and diagonal reflection transformation of 45° 135° are proposed).

According to the Partitioned Iterated Function System (PIFS), we can find out the domain blocks matched with the range blocks using affine transformation iterations.

$$R' = \bar{r}U + s\rho(D - \bar{d}U) \tag{1}$$

ARPN Journal of Engineering and Applied Sciences
©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

For each range block R, orthogonalization fractal block coding is obtained by minimizing the following equation

$$E(R,D) = \left\| R - \bar{r}_i U - s_j \rho(D - \bar{d}U) \right\|^2 \tag{2}$$

The above minimization is performed over $D \in \Omega$ by working with a set of pre-quantized fractal parameters $\{\bar{r}_i\}_{i=1}^{I}$ and $\{s_j\}_{j=1}^{J}$ (I and J are the quantization levels for $\bar{r}_i$ and s_j, respectively). Note that U is a matrix whose elements are all ones, s is a contrast scaling parameter, ρ is the isometric transform, $\|\cdot\|$ is the 2-norm and $\bar{r}$ and $\bar{d}$ are the average of range block and domain block respectively. $\bar{r}_i$ is the average of the i-th range block. Since $\langle U, D - \bar{d}U \rangle = 0$, we define Equation (2) as orthogonalization fractal block coding. Then range block R can be written as $(\bar{r}, s, x_D, y_D) = \arg\min_{D\in\Omega} E(R,D)$. Where (x_D, y_D) is the top-left corner coordinating of the 'best-matching' domain block.

When all the domain blocks matched with the range blocks are found, the fractal coding of the whole image is completed.

Collage error

We define the collage error as follows:

$$e = \hat{E}(R) = \min_{D\in\Omega} \sqrt{\frac{E(R,D)}{B \times B}} \tag{3}$$

Collage error is a quantitative measure of the similarity between range block and "best-matching" domain block. It is relatively robust compared with other fractal parameters which can be quite sensitive to changes in domain block pool. Pi MH [9] has proved that the proposed indices not only reduce computational complexities, but also enhance the retrieval rate, compared with the existing fractal-based retrieval methods.

Program code

```
[imagem imagen]=size(Image1);
Sr=4;Sd=8;
Rnum=(imagem/Sr)*(imagen/Sr);
Dnum=(imagem/Sd)*(imagen/Sd);
Image2=zeros(Dnum,Sr,Sr);
Image2=blkproc(Image1,[Sd/Sr,Sd/Sr],'mean(mean(x))');
RBlocks=zeros(Rnum,Sr,Sr);
DBlocks=zeros(Dnum,Sd,Sd);
DBlocksReduce=zeros(Dnum*8,Sr,Sr);
for i=1:imagem/Sr
for j=1:imagen/Sr
k=(i-1)*imagen/Sr+j;
RBlocks(k,:,:)=Image1((i-1)*Sr+1:i*Sr,(j-1)*Sr+1:j*Sr);
end
end
for i=1:imagem/Sd
for j=1:imagen/Sd
k=(i-1)*imagen/Sd+j;
m=Sr; n=Sr;
DBlocksReduce(k,:,:)=Image2((i-1)*Sr+1:i*Sr,(j-1)*Sr+1:j*Sr);
DBlocksReduce(k+Dnum,:,:)=DBlocksReduce(k,m:-1:1,:);
DBlocksReduce(k+2*Dnum,:,:)=DBlocksReduce(k,:,n:-1:1);
DBlocksReduce(k+3*Dnum,:,:)=DBlocksReduce(k,m:-1:1,n:-1:1);
DBlocksReduce(k+4*Dnum,:,:)=reshape(DBlocksReduce(k,:,:),Sr,Sr)';
A=reshape( DBlocksReduce(k+3*Dnum,:,:),Sr,Sr)';
DBlocksReduce(k+5*Dnum,:,:)=A(:,n:-1:1);
DBlocksReduce(k+6*Dnum,:,:)=
imrotate(reshape(DBlocksReduce(k,:,:),Sr,Sr),90);
DBlocksReduce(k+7*Dnum,:,:)=
imrotate(reshape(DBlocksReduce(k,:,:),Sr,Sr),270);
DBlocks(k,:,:)=Image1((i-1)*Sd+1:i*Sd,(j-1)*Sd+1:j*Sd);
end
end
RandDbest=zeros(Rnum,1)+256^3;
RandDbests=zeros(Rnum,1);
RandDbesto=zeros(Rnum,1);
RandDbestj=zeros(Rnum,1);
for i=1:Rnum
x=reshape(RBlocks(i,:,:),Sr*Sr,1);
meanx=mean(x);
for j=1:Dnum*8
y=reshape(DBlocksReduce(j,:,:),Sr*Sr,1);
meany=mean(y);
s=(x-meanx)'*(y-meany)/((y-meany)'*(y-meany));
o=(meanx-s*meany);
e=(x-s*y-o)'*(x-s*y-o);
if (RandDbest(i)>e)
RandDbest(i)=e;
RandDbests(i)=s;
RandDbesto(i)=o;
RandDbestj(i)=j;
end
end
end
```

Proposed method

Kernel Density Estimation (KDE), as popular nonparametric density estimation, is widely used in the field of pattern recognition, classification and image processing. The histogram is the simplest non-parametric density estimation method which is frequently used. It has been demonstrated that histograms of fractal parameters capture statistical characteristic of texture images effectively [4]. Since the histograms heavily depends on width of bins and end points may result in different histogram distribution, meanwhile, different distributions lead to different results of image retrieval, and thereby affect the retrieval rate.

VOL. 9, NO. 7, JULY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

On the condition that the densities of data are unknown, the kernel density estimation is used, and it has been applied into statistical characteristics of images in massive literatures [10, 11] and has obtained considerable results. The properties of kernel density estimation are, as compared to histograms, smooth, no end points and they depend on bandwidths heavily. This method is more simple and effective than the histogram method, which reduces the computational complexity of data in image retrieval.

Kernel density estimation of fractal parameters

Let $x_1, x_2, \cdots x_n$ be an i.i.d (independent and identically distributed) sample drawn from some distribution with an unknown density $p(x)$. We are interested in estimating the shape of this function $p(x)$. Its kernel density estimation is

$$\hat{p}_n(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{4}$$

where $K(\bullet)$ is the kernel - a symmetric but not necessarily positive function that integrates to one - and h is a smoothing parameter called the bandwidth. Intuitively we want to choose h as small as possible, which will lead to instability. However, we should balance the smoothness and stability of the estimation. However, it's the bandwidth not the function of the kernel that exhibits a strong influence on the estimation results. A bandwidth h of the kernel may alter the density estimation, and it can accordingly affect the goodness-of-fit of the density function $\hat{p}_n(x)$ to the unknown underlying density $p_n(x)$. Generally the most common optimal criterion used to select bandwidth is the Mean Integrated Squared Error (MISE) [12]. This principle is applied to select a fixed optimal bandwidth. The optimal bandwidth is the argument that minimizes the MISE.

$$MISE(\hat{p}_n(x)) = E\int [\hat{p}_n(x) - p_n(x)]^2 dx \tag{5}$$

The integrand of the MISE can be decomposed into three parts: $E\hat{p}_n^2 - 2p_n E\hat{p}_n + p_n^2$. Then we will subtract p_n^2 from the MISE since it is the underlying density and does not depend on the choice of a kernel, thus the cost function as a function of the bandwidth is defined as

$$MISE' = MISE - \int_a^b p_n^2(x)dx = \int_a^b E\hat{p}_n^2 dx - 2\int_a^b p_n E\hat{p}_n dx \tag{6}$$

Here [a, b] is an interval of interest, and the interval length is H. The minimum of the cost function Eq. (6) is an estimate of the fixed optimal bandwidth, which is denoted by h^*.

After obtaining the fixed bandwidth, we will introduce the proposed method to obtain a variable bandwidth. First we define a formula:

$$y = \frac{1}{n}\sum_{i=1}^{N} \delta(x - x_i) \tag{7}$$

Where n is the number of estimated point. $\delta(t)$ is the Dirac delta function. The kernel density estimation is obtained by convoluting a Gaussian kernel $k(s)$ to y.

$$\hat{p}_n = \int y_{x-s} k(s) ds \tag{8}$$

The most commonly used kernel is Gaussian kernel function:

$$k(s) = \frac{1}{\sqrt{2\pi}h}\exp(-\frac{s^2}{2h^2}) \tag{9}$$

As we know, the fixed bandwidth is selected from an entire observation interval [a, b], however, the estimation may be improved by using a kernel bandwidth which is adaptively selected in conformity with data. Thus, the kernel density estimation with the variable bandwidth h_x is expressed as

$$\hat{p}_n = \int y_{x-s} k_{h}(s) ds \tag{10}$$

Here, we provide a method for obtaining the variable bandwidth h_x that minimizes the MISE by optimizing a local interval length among which the variable bandwidth can be regarded as a fixed one. To conduct the local optimization, we introduce the local MISE criterion as

$$localMISE = E\int [\hat{p}_n(x) - p_n(x)]^2 \rho_H dx \tag{11}$$

The weight function ρ_H localizes the integration in the interval H. According to Eq.(6), we introduce the local cost function by subtracting the term irrelevant for the choice of h as

$$C_n(h,H) = localMISE - \int p_n^2 \rho_H dx = \frac{1}{n^2}\sum_{i,j}\psi_{h,H}(x_i,x_j) - \frac{2}{n^2}\sum_{i\neq j} k_h(x_i - x_j)\rho_H^{x_i - x} \tag{12}$$

where

$$\psi_{h,H}(x_i,x_j) = \int k_h(u - x_i)k_h(u - x_j)\rho_H^{u-x} du \tag{13}$$

The optimal bandwidth h^* varies according to different interval length H. We suggest selecting an interval length that scales with the optimal bandwidth as $\gamma^{-1}h^*$, the parameter γ is a smoothing parameter for the variable bandwidth; it regulates the interval length for local optimization. With small γ, the variable bandwidth fluctuates slightly, while with large γ, the variable bandwidth fluctuates significantly.

In order to select a variable kernel bandwidth, firstly, compute the local cost function $\psi_{h,H}(x_i,x_j)$ in Equation (13) and find that minimize the Equation (13)

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

then repeat the procedure above while changing H. Change H to find H^* that satisfies $H=\gamma^{-1}h^*$. We could obtain the variable bandwidth by computing the cost function

$$\hat{C}_n(\gamma)=\int_a^b \hat{p}_n^2 dx-\frac{2}{n^2}\sum_{i\neq j} k_{h_\gamma}(x_i-x_j) \quad (14)$$

where $\hat{p}_n=\sum_i k_{h_\gamma}(x-x_i)$. At last, we should repeat the procedure above to find γ^* that minimizes $\hat{C}_n(\gamma)$, and then apply it to obtain the variable bandwidth h_γ^*. The bandwidth is what we want to calculate kernel density estimation more precisely.

It has been proved that range block mean $\bar{R}$, contrast scaling parameter s and collage error e [4, 6, 9] are effectively used to retrieve images. In this paper, we apply these parameters into kernel density estimation directly. The optimal uniform quantization for $\bar{R}$ is $\{0,1,2,...,63\}$ [13] and to ensure the convergence of the decoding, the scaling factor s is restricted to the interval $(-S_{\max}, S_{\max})$, where $0<S_{\max}<1$, as for collage error e, it is real-valued , hence, before we calculate kernel density estimation of collage errors, they are rounded into the closest integer if collage errors are smaller than $T-1$, or are set as $T-1$ if they exceed $T-1$(T is a user-specified threshold). In this paper, we set $T=20$. Then we calculate kernel density estimation with the processed collage errors and fractal coding parameters.

Figure-1 shows four similar and four different texture images from VisTex texture database [14]. Kernel density estimation of range block mean $\bar{R}$, contrast scaling parameter s and collage error e corresponding to these images are plotted in Figure-2. In most cases, the curves are close for similar texture images, and different for the dissimilar texture images.

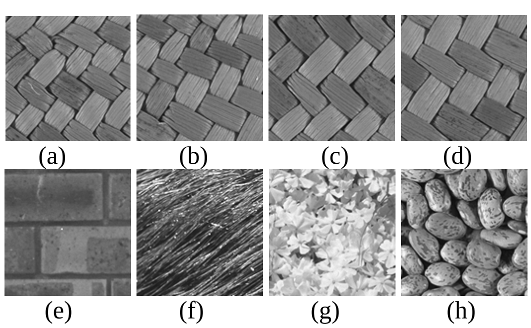

Figure-1. Examples of 128x128 images. (a)- (d) Four similar images; (e)-(h) Four different images.

The left column of Figure-2 shows the kernel density estimation of range block mean $\bar{R}$, contrast scaling parameter s and collage error e respectively according to the first four similar images (a)-(d). Obviously the curves are close for the similar texture images.

The right column of Figure-2 shows the kernel density estimation of range block mean $\bar{R}$, contrast scaling parameter s and collage error e respectively according to the other four different images(e)-(h). We can see that the curves are different for the dissimilar texture images.

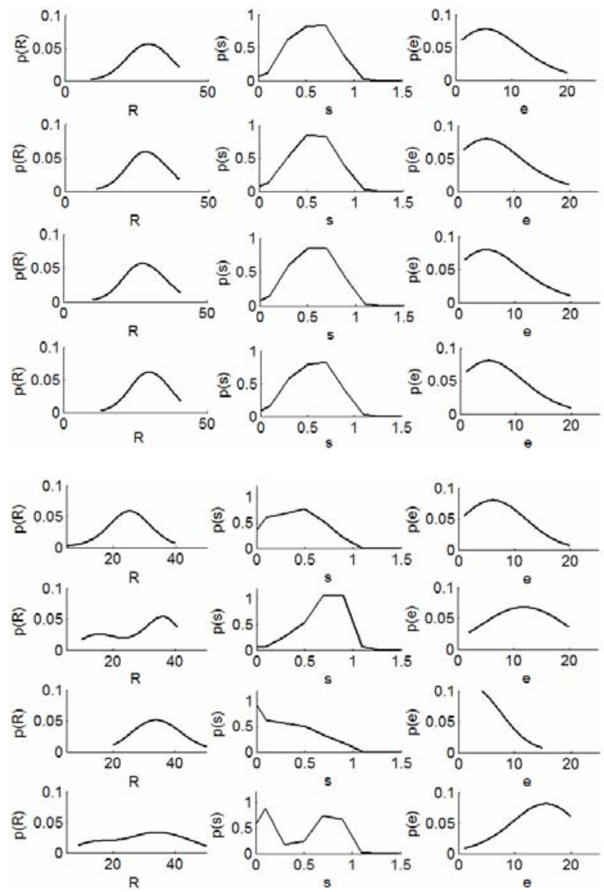

Figure-2. Comparison of the KDE of fractal coding parameters.

Similarity measurement

We define that $V_Q(\bullet)=\{u_1,u_2,...,u_H\}$ and $V_C(\bullet)=\{v_1,v_2,...,v_H\}$ are the features of the query and candidate images respectively. The vectors of range block mean $\bar{R}$, contrast scaling parameter s and collage error e of a query image are expressed as $\{u_{R1},u_{R2},...,u_{RH}\}$, $\{u_{S1},u_{S2},...,u_{SH}\}$ and $\{u_{e1},u_{e2},...,u_{eH}\}$ respectively. The same with vector V_C for candidate images.

In addition, the subscript variable H represents the amount of range block mean $\bar{R}$, contrast scaling parameter s or collage error e of an image. To measure the similarity between two images, we can calculate the deviation between their fractal coding parameters instead since these parameters can express images compactly.

VOL. 9, NO. 7, JULY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

In this paper, we adopt the most commonly used L_2 and *KLD* (Kullback-Leibler divergence) [15] as the distance criterion to measure the similarity between the query and the candidate images, the distance between the two images is calculated as follows:

$$d_{L_2}(Q,C)=\sqrt{\sum_{b=1}^{H}(u_b-v_b)^2} \quad (15)$$

$$KLD(Q,C)=\sum_{b=1}^{H}u_b\log(\frac{u_b}{v_b}) \quad (16)$$

The distances between similar images are much smaller than those between dissimilar images. Experiments show that the similarity measurement Equation (7) performs much better than Equation (8). Thus we only discuss the retrieval rate using L_2 distance. The obtained distances are sorted in an ascending order.

PERFORMANCE EVALUATION

We have performed experiments on VisTex texture database. The set of VisTex is the classical selection of 40 classes of texture images that are used by many literatures for image retrieval [16]. The images are listed as follows: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, and Wood2. These are real world 512×512 images from different natural scenes. Only gray-scale levels of the images are used. As for our experiments each image is divided into sixteen non-overlapping 128×128 sub-images, thus creating a test database of 640 texture images.

In the retrieval experiments, each sub-image in the database is used once as a query image. For comparison purpose, retrieved images are the first 16 most similar images for each query. The relevant images for each query consist of all the sub-images from the same original texture image. All experiments are conducted on a 2GHz PC using Matlab7.8.0 as a programming tool.

The retrieval method using feature vectors of range block mean, contrast scaling and collage error are respectively named as KM kernel density estimations of range block mean , KS kernel density estimations of contrast scaling and KE kernel density estimations of collage error . A large number of texture images in database are used to do testing, from which we could conclude that the KE method works much better than the KM and KS. Therefore, we apply the KE method into images retrieval in this paper.

Average retrieval rate and retrieval speed

We use 40 512×512 VisTex texture images. Each image is divided into16 128×128 non-overlapping sub-images. Finally a test database of $Z=640$ texture images is created. Each sub-image is encoded using full search. Let the number of ideally retrieved images of one class be denoted by F (in this case $F=16$) and m_z be the number of correctly retrieved images of one class from the top 16 images at the z-th test. The performance is measured in Average Retrieval Rate (ARR) that is defined the same with literature [6], which is then calculated as

$$ARR=\frac{\sum_{z=1}^{Z}m_z}{F\times Z} \quad (17)$$

Experiments show that KE (kernel density estimation of collage error) method works better than the others. Table-1 and Table-2 shows that our method performs better than HE (histogram estimation) method, FKE (fixed bandwidth kernel density estimation of collage error) method and other methods.

Table-2 shows that the ARR of the proposed method is 72.40%, which is more that the other listed methods. The runtime of the retrieval, which is completely determined by the performance of the similarity measurement process, is also a key index to indicate the performance. Compared with literature [16], our runtime is largely reduced since the basic arithmetic operations are adopted in our method, while the computationally expensive $\log, e^x$ and x^r operations with more iterations are applied in literature [16], which leads to an increase in computation time.

 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

Table-1. Retrieval rate of three methods (%).

Image	HE	FKE	VKE
Bark0	16.80	21.09	52.34
Bark6	14.84	11.72	37.50
Bark8	22.27	20.31	47.65
Bark9	21.48	22.66	46.87
Brick1	22.27	39.06	64.84
Brick4	34.77	30.86	83.98
Brick5	21.48	24.22	76.56
Buildings9	21.88	26.17	75.39
Fabric0	46.88	41.80	76.56
Fabric4	36.80	41.48	75.00
Fabric7	33.69	38.91	75.78
Fabric9	58.20	59.53	74.21
Fabric11	38.28	42.58	76.56
Fabric14	35.94	40.63	74.60
Fabric15	52.34	46.88	76.56
Fabric17	30.47	38.67	74.60
Fabric18	25.00	46.88	75.39
Flowers5	57.03	58.59	73.82
Food0	67.19	63.44	81.17
Food5	65.40	28.52	73.43
Food8	46.48	51.17	76.17
Grass1	32.03	42.50	73.82
Leaves8	36.72	64.06	76.17
Leaves10	23.83	27.34	75.78
Leaves11	48.44	57.81	75.00
Leaves12	42.58	57.81	76.56
Leaves16	50.47	32.03	71.87
Metal0	38.67	54.53	81.17
Metal2	34.77	48.67	76.09
Misc2	39.45	46.25	77.73
Sand0	37.11	35.16	73.04
Stone1	26.56	33.44	78.12
Stone4	31.25	33.98	75.00
Terrain10	37.97	22.66	77.34
Tile1	72.27	63.28	85.82
Tile4	69.92	74.22	88.04
Tile7	63.28	71.41	86.60
Water5	25.00	22.27	70.31
Wood1	22.66	34.27	75.39
Wood2	38.75	38.13	87.10

Table-2. Average Retrieval Rate (ARR) compared with other literatures (%).

Method	HE	FKE	literature[3] **GGD+MM**	literature[4] **rmm+CT**	**Proposed Method (VKE)**
ARR %	35.42	39.88	67.27	69.52	72.40

CONCLUSIONS

In this paper, we apply orthogonalization fractal coding algorithm into image retrieval, which has been verified that the decoding speed is higher than that of the basic fractal coding. Meanwhile, we propose an image retrieval method based on fractal coding parameter with a variable optimized bandwidth kernel density estimation method. The kernel bandwidth can adjust according to the data distribution. Thus, the statistical characteristics of fractal coding parameters are employed as retrieval indices. Experiments show the superiority in both retrieval rate and retrieval speed when compared with the existing methods. In the future, we will combine some other features of images with fractal parameters to improve performance of image retrieval.

©2006-2014 Asian Research Publishing Network (ARPN). All rights reserved.

REFERENCES

[1] Young Deok Chun, Nam Chul Kim, Ick Hoon Jang.: "Content-Based Image Retrieval Using Multi-resolution Color and Texture Features," IEEE Transaction on Multimedia. vol.10, pp. 1073-1084 (2008).

[2] Barnsley M., Sloan A.D.: "A better way to compress images," BYTE. vol.13, pp.215–233 (1988).

[3] A.E.: "Fractal image coding: a review," Proc. IEEE, vol.81 (10), pp. 1451–1465 (1993).

[4] Pi M, Mandal M, Basu A. "Image retrieval based on histogram of new fractal parameters," Proc. ICASSP', vol. 3, pp. 585-588 (2003).

[5] Pi M., Mandal M., Basu A. "Image retrieval based on histogram of fractal parameters," IEEE Trans. Multimed., vol.7, (4), pp. 597-605 (2005).

[6] M. Pi, H. Li. "Fractal indexing with the joint statistical properties and its application in texture image retrieval," [J]. Image Processing, IET, vol.2 (4), pp.218 -230 (2008).

[7] Hideaki Shimazaki, Shigeru Shinomoto.: "Kernel bandwidth optimization in spike rate estimation," J Comput Neurosci, vol. 29, pp.171-182 (2010).

[8] G.E. Øien and S.LepsØy. "A Class of Fractal Image coders with Fast Decoder Convergence," Signal Processing, vol.40 (1), pp. 105-117 (1994).

[9] Pi M, Tong C.S, Basu A.: "Improving fractal codes based image retrieval using histogram of collage errors," 2nd International Conference on Image and Video Retrieval, vol.2728, pp.121-130 (2003).

[10] Huang Xiaoqing, Zhang Qin, Liu Wenbo.: "A new method for image retrieval based on analyzing fractal coding characters," Journal of Visual Communication and Image Representation. 24(1):42-47 (2013).

[11] Liu Zhi, Shi Ran, Shen Liquan.: "Unsupervised Salient Object Segmentation Based on Kernel Density Estimation and Two-Phase Graph Cut," IEEE Transactions on Multimedia, vol.14 (4), pp. 1275-1289 (2012).

[12] Deng Biao, Yu Chuanqiang, Li Tianshi, Su Wenbin, Pan Yangke. "Dual-bandwidth kernel density estimation algorithm based on estimate points," Chinese Journal of Scientific Instrument, vol.32 (3), pp.615-620 (2011).

[13] Tong C.S., Pi M. "Fast fractal image compression using adaptive search," IEEE Trans. Image Process., vol.10, pp. 1269-1277 (2001).

[14] VisTex texture database, http://vismod.media.mit.edu/vismod/imagery/VisionTexture/.

[15] Do M.N., Vetterli M. "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," IEEE Trans. Image Process., vol. 11(2), pp. 146-158 (2002).

[16] Kwitt. R., Uhl. A. "Lightweight Probabilistic Texture Retrieval." IEEE Transactions on Image Processing. Vol. 19(1), pp. 241-253 (2010).