Image Matching Using Run-Length Feature

Yung-Kuan Chan and Chin-Chen Chang
Department of Computer Science and Information Engineering
National Chung Cheng University, Chiayi, Taiwan, 621, R.O.C.
E-mail:{chan, ccc}@cs.ccu.edu.tw

Correspondence address:
Professor Chin-Chen Chang
Department of Computer Science and Information Engineering
National Chung Cheng University,
Chiayi, Taiwan 621, R. O. C.
Email:ccc@cs.ccu.edu.tw
FAX:886-5-2720859

Image Matching Using Run-Length Feature

Yung-Kuan Chan and Chin-Chen Chang
Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan, 621, R.O.C.
E-mail:{chan, ccc}@cs.ccu.edu.tw

Abstract

Color histogram is the most commonly used color feature in image retrieval systems. However, this feature cannot effectively characterize an image, since it only captures the global properties. To make the retrieval more accurate, this paper introduces a run-length feature. The feature integrates the information of color and shape of the objects in an image. It can effectively discriminate the directions, areas and geometrical shapes of the objects. Yet, extracting the run-length feature is time-consuming. For that reason, this paper also provides one revised representation of the run-length, called semi-run-length. Based on the semi-run-length feature, this paper develops an image retrieval system, and the experimental results show that the system gives an impressive performance.

Keywords: color-based image retrieval, shape-based image retrieval, color histogram

1. Introduction

Trademarks play an important role in providing unique identity for products and services in the marketing environment. Most of the images contain a group of large regions each with a uniform color if the pixel colors of the images are quantized down to a small subset of representative colors. Many other synthesized images like flags, traffic signs and cartoon images also possess this property. So far no guarantee can be made that new trademarks will not closely resemble existing ones. The size of class and the need for continuous updating have made automatic archiving and retrieval procedures highly desirable. Thus, similar image retrieval is essential for image databases. This paper will provide an effective image retrieval system to take care of this job. In this system, the user inputs the query image through a scanner. The system automatically extracts the visual features of the database and query images, and then it matches the visual features of the query image against those of the database images. The result is a set of images that are similar to the query image rather than an exact match.

Color is a dominant visual feature widely used in many image retrieval systems. The color histogram is the most popular color representation (Sawhney *et al.*, 1994; Stricker *et al.*, 1994) because it is simple and computationally efficient. The color feature is insensitive to the orientation and resolution of images, and noise on images. Apart from that, color matching can be processed automatically without human intervention. Unfortunately, it is only a global attribute without any local information, so color alone is not sufficient to depict an image. Consider the two images in Fig. 1. Even if there is a significant difference in their appearances, we cannot distinguish them with the color feature since they have the same color histograms. To get the retrieval more accurately, this paper proposes a new feature, named run-length. This feature fuses the color information as well as the shape knowledge of objects in an image. It can effectively characterize the direction, area and geometrical shape of an object. However, extracting the run-length feature is time-consuming. Thus, this paper also provides the semi-run-length feature, one alternative vision of the run-length.

The remainder of this paper is organized as follows. A brief introduction to the related works is stated in the next section. Section 3 serves to introduce the new visual features, run-length and semi-run-length. Section 4 describes the proposed image retrieval system. Section 5 shows the experimental results. The conclusions are presented in the last section.

2. Related Works

(Mehtre *et al.*, 1995) proposed a reference color table method for image retrieval. The method defines a set of reference colors. This set of colors is selected such that all the colors in images are approximately covered perceptually. The method then computes a histogram for each image where each pixel in the image is classified against the reference color table and assigned to the nearest color. Thus, the color feature of the image is this reduced color histogram based on the colors of the reference table.

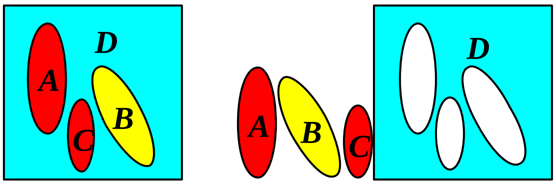
Let f_d = ($?_{d1}$, $?_{d2}$, …, $?_{dk}$) be the histogram of an image I_d, and f_q = ($?_{q1}$, $?_{q2}$, …, $?_{qk}$) be that of the other image I_q. $?_{di}$ and $?_{qi}$ are the relative pixel frequencies (with respect to the total number of pixels) for the i-th color of reference table in the images I_d and I_q, respectively. k is the size of the reference color table. (Mehtre *et al.*, 1995) defines the image matching distance between I_d and I_q as follows

$$D ? \overset{k}{\underset{i?1}{?}} ?_i \sqrt{?\, ?_{di} ? ?_{qi} \,?^2}, \quad \text{where} \quad ?_i ? \begin{cases} ?_{qi} & \textbf{\textit{if}}\ ?_{qi}, ?_{di} ? \textbf{\textit{0,}} \\ \textbf{\textit{1}} & \textbf{\textit{otherwise.}} \end{cases}$$

This paper will compare experimentally the proposed semi-run-length method with this method.

(a) (b)

Fig. 1. Two color images



(a) (b)

Fig. 2. A color image and its objects

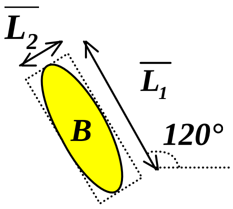


Fig. 3. The direction of Object ***B***

3. The Run-Length Representation

In general, a true color image system takes ***24*** bits to indicate a pixel color. It means that there are a total of approximately sixteen million possible pixel colors; nevertheless, only a small subset of them appears in a particular image. As a result, the cost of the high-speed memory required to support such displays on a high-resolution monitor makes true-color displays impractical and in fact unnecessary for many applications. An available approach is to offer a limited number of bits to specify a pixel color. In this way, a small subset of colors is picked out from the $\boldsymbol{2^{24}}$ colors where there is no noticeable difference between the original and quantized images. Then, each pixel of a color image is mapped to one ***24***-bit value in its related color palette. Two steps must be considered for designing a suitable palette: selecting an appropriate palette and mapping each input pixel to one entry of the palette. Many techniques (e.g. Kolpatzik *et al.* 1995 and Pei *et al.*, 1995) have been proposed to design the palette, so we shall not devote this paper to the part.

Segmentation is very important for shape attributes. However, extracting objects from an image is very difficult because of discretization, occlusions, poor contrasts, viewing conditions, and noises, etc (Adoram *et al.*, 1999). An image with a limited color palette is composed of a set of unicolor regions. In cases where the segmentation is less difficult and possible to overcome, this paper considers each unicolor region to be an object. Fig. 2(a) demonstrates the image consisting of the four objects ***A***, ***B, C*** and ***D*** shown in Fig. 2(b). Here, the object ***D*** is the remains of the original image after removing the objects ***A***, ***B***, and ***C***. An object can be enclosed by a minimum rectangle. Let the two orthogonal line segments $\overline{L_1}$ and $\overline{L_2}$ be two of the four sides of the minimum rectangle, and the length of $\overline{L_1}$ is larger than or equal to that of $\overline{L_2}$. We call the inclination of $\overline{L_1}$ the direction of the object, which can be described as the angle ? drawn from X-axis to $\overline{L_1}$ in the counterclockwise direction, and $0 < ? \leqq 180$ °. For example, according to Fig. 3, the direction of the object B is $120°$.

Unlike retrieval by colors, textures or spatial relationships, the shape attribute does not have a mathematical definition that exactly matches what the users feel as a shape (Bimbo *et al.*, 1996; Bimbo *et al.*, 1997). Therefore, capturing a perceptual shape similarity is in general difficult, as it requires a precise characterization of a class shapes. In this paper, two objects are considered to be similar to each other if their colors, areas, directions and geometric shapes are close. For instance, in Fig. 2(b), the areas and geometrical shapes of the objects A and B are the same, but their colors and directions are dissimilar.

Color histogram is attractive because it is simple and computationally efficient. However, since the color histogram lacks the local information, it cannot distinguish the images with the similar histograms even though they have the difference appearances. To overcome the drawback, this section might introduce the run-length feature. It can efficiently describe the colors, directions, areas and geometrical shapes of the objects in an image. However, extracting the run-length feature from an image is time-consuming. Hence, this paper also presents a more efficient representation called semi-run-length, which is an alternative revision of the run-length.

A_{11}	B_{12}	A_{13}
B_{21}	A_{22}	B_{23}
A_{31}	B_{32}	A_{33}

(a) Image f_1

B_{11}	A_{12}	B_{13}
A_{21}	A_{22}	A_{23}
B_{31}	A_{32}	B_{33}

(b) Image f_2

A_{11}	A_{12}	A_{13}
B_{21}	A_{22}	B_{23}
B_{31}	A_{32}	B_{33}

(c) Image f_3

Fig. 4 Three color images

3.1. The Run-Length Representation

A run-length value is the number of consecutive pixels $\boldsymbol{A_i, A_{i+1}, \ldots, A_j}$ along the scan line $\overline{A_iA_j}$. Here, $\overline{A_iA_j}$ is composed of $\boldsymbol{A_i, A_{i+1}, \ldots, A_j}$ with an identical color, where $\boldsymbol{A_{i-1}}$ differs in color from $\boldsymbol{A_i}$, and the color of $\boldsymbol{A_j}$ is different from that of $\boldsymbol{A_{j+1}}$, either. We define the run-length as a ? -run-length (? -RL) if the inclination of $\overline{A_iA_j}$ is ? . In other words, ? is the angle drawn from the X-axis to $\overline{A_iA_j}$ in the counterclockwise direction. Consider the three $\boldsymbol{3\times3}$ color images shown in Fig. 4, where the color values of the pixels $\boldsymbol{A_{11}}$, $\boldsymbol{A_{12}}$, …, $\boldsymbol{A_{33}}$ are $\boldsymbol{A}$, and those of the pixels $\boldsymbol{B_{11}}$, $\boldsymbol{B_{12}}$, …, $\boldsymbol{B_{33}}$ are $\boldsymbol{B}$. The pixels $\boldsymbol{A_{11}}$, $\boldsymbol{A_{22}}$, and $\boldsymbol{A_{33}}$ in Fig. 4(a) form a scan line $\overline{A_{11}A_{33}}$; the run-length is a $\boldsymbol{135^o}$-RL, and the value is $\boldsymbol{3}$.

Table1. The run-length values of the images in Fig. 4

Color	Color ***A***				Color ***B***			
? -RL	***45°-RL***	***90°-RL***	***135°-RL***	***180°-RL***	***45°-RL***	***90°-RL***	***135°-RL***	***180°-RL***
f_1	***11***	***5***	***11***	***5***	***8***	***4***	***8***	***4***
f_2	***9***	***11***	***9***	***11***	***4***	***4***	***4***	***4***
f_3	***7***	***11***	***7***	***11***	***4***	***8***	***4***	***4***

In the run-length representation, for each pixel $\boldsymbol{P}$, there are only four run-length values to be considered: $\boldsymbol{45^o}$-RL, $\boldsymbol{90^o}$-RL, $\boldsymbol{135^o}$-RL and $\boldsymbol{180^o}$-RL, whose scan lines pass $\boldsymbol{P}$. We call them the ? -RLs of the pixel $\boldsymbol{P}$, where ? = $\boldsymbol{45^o}$, $\boldsymbol{90^o}$, $\boldsymbol{135^o}$ and $\boldsymbol{180^o}$, respectively. Additionally, the ? -RLs of a color $\boldsymbol{C}$ in an image are the summations of the related ? -RLs of all the pixels with the color $\boldsymbol{C}$. For instance, the $\boldsymbol{135^o}$-RL of the pixel $\boldsymbol{A_{22}}$ in Fig. 4(a) is $\boldsymbol{3}$, and the $\boldsymbol{135^o}$-RL of the color $\boldsymbol{A}$, namely the summation of the $\boldsymbol{135^o}$-RLs of $\boldsymbol{A_{11}, A_{13}, A_{22}, A_{31}}$ and $\boldsymbol{A_{33}}$, is $\boldsymbol{11}$. Table 1 shows the ? -RLs of the colors $\boldsymbol{A}$ and $\boldsymbol{B}$ of the images in Fig. 4. The color histograms of each image in Fig. 4 contain $\boldsymbol{5}$ pixels of the color $\boldsymbol{A}$ and $\boldsymbol{4}$ pixels of the color $\boldsymbol{B}$. The color histogram alone cannot discriminate the three images. However, the run-length representation can tell apart the three images f_1, f_2, and f_3 into three different classes. Thus, the discrimination of the run-length representation is more powerful than that of the color histogram.

3.2. The Semi-Run-Length Representation

The run-length representation needs to calculate the run-length value for each pixel. Obviously, it is a time-consuming approach. In general, a computer system scans the pixels of an image row by row from top to bottom, and in each row, the pixels are visited sequentially from left to right. In this way, one can use a table to write down the partial run-length values of the pixels which have been visited. Consequently, the processing time hence can be reduced significantly. Next, this paper will propose a more efficient representation, called semi-run-length. The representation uses the partial run-lengths as the features of images.

A ? -semi-run-length (? -SRL) value of the pixel A_i is the number of successive pixels A_i, A_{i+1} , …, A_j along the scan line $\overline{A_iA_j}$. Here, the $\overline{A_iA_j}$ comprises A_i, A_{i+1} , …, A_j with the same color, so that A_j differs in color from A_{j+1}, and the inclination of $\overline{A_iA_j}$ is ? with respect to A_i. Similarly, the ? -SRL of a color C in an image is also the summation of the related ? -SRLs of all the pixels with the color C. In Fig. 4(a), the line segment $\overline{A_{22}A_{11}}$ consists of A_{11} and A_{22}; hence the 135^o-run-length of A_{22} is **2,** not ***3***. Table 2 illustrates the semi-run-lengths of the colors ***A*** and ***B*** of the three images in Fig 4. The semi-run-length representation can tell the three images apart as well.

The semi-run-length feature can discriminate the differences of directions, areas and geometrical shapes among objects as well. For example, the objects with the color ***A*** in Figures 4(a) and 4(b) have the same areas and geometrical shapes; nevertheless, their ? -semi-run-lengths are quite different since their directions are diverse. Similarly, the objects with the color ***A*** in Figures 4(b) and 4(c) have identical direction and area; nevertheless, their ? -semi-run-lengths are still not alike because of their different geometrical shapes. In Fig. 2(b), the ? -SRLs of the object ***A*** must be obviously larger than those of the object ***C*** in spite of the fact that they have the same direction and geometrical shape, because ***A*** is bigger than ***C***.

Table2. The semi-run-length values of the images in Fig. 4

Color	Color ***A***				Color ***B***			
? -SRL	***45º-SRL***	***90º-SRL***	***135º-SRL***	***180º-SRL***	***45º-SRL***	***90º-SRL***	***135º-SRL***	***180º-SRL***
F_1	***8***	***5***	***8***	***5***	***6***	***4***	***6***	***4***
F_2	***7***	***8***	***7***	***8***	***4***	***4***	***4***	***4***
F_3	***6***	***8***	***6***	***8***	***4***	***6***	***4***	***4***

Consider the input image denoted by a two-dimension array ***Image***. Assume that ***Image*** is an $N_1 \times N_2$ image. Each of the entries on ***Image*** maps to the index of the closest color in a limited color palette. The following algorithm ***Semi-Run-Length*** is designed to compute the ? -SRLs of the pixel colors in ***Image***. In this algorithm, the variable ***SRL*** records the ? -SRLs of the image. Its ***i***-th entry consists of ***SRL*[*i*].*45º***, ***SRL*[*i*].*90º***, ***SRL*[*i*].*135º*** and ***SRL*[*i*].*180º*** which write down the ***45º***-SRL, ***90º***-SRL, ***135º***-SRL and ***180º***-SRL of the ***i***-th color of palette, respectively. Additionally, one-dimension arrays ***PRL-45º***, ***PRL-90º***, ***PRL-135º*** and ***PRL-180º*** are used to record the partial run length values of the pixels which have been visited. ***p*** is the size of common color palette. ***No_Pixel*[*k*]** is the number of pixels on ***Image*** whose colors are mapped to the ***k***-th color of the common palette. The text embedded in a pair of braces is the given comments.

***Algorithm Semi-Run-Length*():**

for $i = 1$ **to** N_1 **and** $j = 1$ **to** N_2

{Compute the ***45º-SRLs***}

if *Image*[*i*][*j*] **is at the first row or the rightmost column then** *PRL-45º* [*j*] = *1*

else if *Image*[*i-1*][*j+1*] = *Image*[*i*][*j*] **then** *PRL-45* [*j*] = *PRL-45º* [*j+1*] + *1*

else *PRL-45º* [*j*] = *1*

SRL[*Image*[*i*][*j*]].*45º* = *SRL*[*Image*[*i*][*j*]].*45º* + *PRL-45º* [*j*]

{Compute the ***90º-SLRs***}

if *Image*[*i*][*j*] **is at the first row then** *PRL-90º* [*j*] = *1*

else if *Image*[*i-1*][*j*] = *Image*[*i*][*j*] **then** *PRL-90º*[*j*] = *PRL-90º SRL*[*j*] + *1*

else *PRL-90º*[*j*] = *1*

SRL[*Image*[*i*][*j*]].*90º* = *SRL*[*Image*[*i*][*j*]].*90º* + *PRL-90º*[*j*]

{Compute the ***135º-SLRs***}

$k = [(N_2 \times i + j - i) \bmod N_2] + 1$

if *Image*[*i*][*j*] **is at the first row or the first column then** *PRL-135º*[*k*] = *1*

else if *Image*[*i-1*][*j-1*] = *Image*[*i*][*j*] **then** *PRL-135º*[*k*] = *PRL-135º*[*k*] + *1*

else *PRL-135º* [*k*] = *1*

SRL[*Image*[*i*][*j*]].*135º* = *SRL*[*Image*[*i*][*j*]].*135º* + *PRL-135º*[*k*]

{Compute the ***180º-SLRs***}

if *Image*[*i*][*j*] **is at the first column then** *PRL-180º*[*j*] = *1*

else if *Image*[*i*][*j-1*] = *Image*[*i*][*j*] **then** *PRL-180º*[*j*] = *PRL-180º*[*j-1*] + *1*

else *PRL-180º*[*j*] = *1*

SRL[*Image*[*i*][*j*]].*180º* = *SRL*[*Image*[*i*][*j*]].*180º* + *PRL-180º*[*j*]

{Compute the average ? -SRLs for each pixel**}**

No_Pixel[*Image*[*i*][*j*]] = *No_Pixel*[*Image*[*i*][*j*]] + *1*

for *k* = *1* **to** *k* = *p*

SRL[*k*].*45º* = *SRL*[*k*].*45º* / *No_Pixel*[*k*]

SRL[*k*].*90º* = *SRL*[*k*].*90º* / *No_Pixel*[*k*]

SRL[*k*].*135º* = *SRL*[*k*].*135º* / *No_Pixel*[*k*]

SRL[*k*].*180º* = *SRL*[*k*].*180º* / *No_Pixel*[*k*]

4. Image Retrieval

Generally speaking, the design of a database system contains two parts: how to efficiently store and retrieve data in and from the database. The implementation of the proposed image retrieval system can be summarized as follows:

In data storage aspect:

The steps to build an image database are:

(1) to transform each image into an $N_1 \times N_2$ contracted image,

(2) to design a common palette for the contracted images,

(3) to map all the pixels on each contracted image to the indices of the closest elements in the common palette, and to compute the histogram of the image,

(4) to extract the ? -SRLs of each contracted image.

In data retrieval aspect:

Given a query image $\boldsymbol{Q}$, the steps to retrieve the similar database images are:

(1) to transform $\boldsymbol{Q}$ into an $N_1 \times N_2$ contracted image,

(2) to map each pixel of the contracted image to the index of the closest element in the common palette, and to compute the histogram of the contracted image,

(3) to extract the ? -SRLs of the contracted image,

(4) to compute the image matching distance between $\boldsymbol{Q}$ and each database image, and

(5) to return the images with the image matching distances less than a given threshold.

A same scene may be characterized to variously sized images owing to some extrinsic factors such as the focal distances of cameras and the resolution of scanners. To eliminate size variances, many techniques in pattern recognition (Grarris *et al.*, 1997) transmute all images to a consistent size. Similarity retrieval works effectively when the users cannot express the queries in a precise way. In this way, an exact match query is not necessary, and a precise description is no need for the similar image retrieval, either. A smaller palette hence can be received by a similar image retrieval system, and each image can be transformed into a consistent contracted image that is regarded as the index of the color image.

After extracting the semi-run-lengths from images, the ? -SRLs $\boldsymbol{SLR_d}$ and $\boldsymbol{SLR_q}$ of a database image $\boldsymbol{I_d}$ and the query image $\boldsymbol{I_q}$ can be obtained. $\boldsymbol{p}$ is the size of the palette. In this system, the image matching distance between $\boldsymbol{I_d}$ and $\boldsymbol{I_q}$ is

$$D = \sum_{k=1}^{p} \Big(\big(SRL_d(k, 45^\circ) - SRL_q(k, 45^\circ)\big)^2 + \big(SRL_d(k, 90^\circ) - SRL_q(k, 90^\circ)\big)^2 + \big(SRL_d(k, 135^\circ) - SRL_q(k, 135^\circ)\big)^2 + \big(SRL_d(k, 180^\circ) - SRL_q(k, 180^\circ)\big)^2 \Big)^{\frac{1}{2}}.$$

5. Experiments

The purpose of the experiments is to compare the performance of the reference color table method (Mehtre *et al.*, 1995) with that of the proposed semi-run-length method. Let $\boldsymbol{D} = \{f_1, f_2, \ldots, f_{400}\}$ and $\boldsymbol{Q} = \{ f_1^{?}, f_2^{?}, \ldots, f_{400}^{?} \}$ be two sets of color images containing ***400*** images. Here, f_i and $f_i^{?}$ are two different images randomly picked out from an identical animation. Most of the animations are downloaded from http://sco25.mi.com.tw/ and http://www.mcsh.kh.edu.tw/. The images in ***D*** are used as the database images, and those in ***Q*** as the query images.

In these experiments, the system returns a short answer list of similar database images for each query image. We say the system precisely answers the query ***i*** if f_i is in the answer list of the query ***i***. ***L*** is the size of the answer list. Table 3 shows the experimental results. In this table, **Ref** and **SRL** denote the reference color table and semi-run-length methods. **Acc** writes down the average accuracy where the system precisely answers a query. **Space** is the total memory space required by the system in the experiments. **Time** is the average executive time for retrieving the images from database that satisfy the requirement of a query. The size of the common palette in these experiments is *27*.

Table 3. The experimental results

	Ref			SRL		
	Acc (%)	Space (bytes)	Time (sec)	Acc (%)	Space (bytes)	Time (sec)
L=1	*85.33*	*43200*	*0.112*	*96.16*	*172800*	*0.140*
L=2	*89.39*	*43200*	*0.115*	*97.52*	*172800*	*0.140*
L=3	*91.65*	*43200*	*0.117*	*98.19*	*172800*	*0.147*
L=4	*92.55*	*43200*	*0.121*	*98.19*	*172800*	*0.148*
L=5	*93.90*	*43200*	*0.122*	*98.19*	*172800*	*0.152*

The experiments demonstrate that the accuracy of semi-run-length method is superior to that of the reference color table method. Since the histograms of the images in Fig. 5 are almost the same, the reference color table method cannot distinguish them, but the semi-run-length method can. However, the semi-run-length needs more memory space and processing time. Furthermore, the semi-run-length can recognize some lightness and rotation variances of images. For example, it can recognize the images in Fig. 6 and in Fig. 8. However, if the rotation and lightness variances between two images are too large, the method cannot recognize them. For instance, it cannot recognize the image pairs in Fig. 7 and in Fig. 9. Moreover, the method is insusceptible to a few noises in images. In these experiments, the proposed method can recognize the pair of images in Fig. 10; nevertheless, it cannot recognize the pair of images in Fig. 11 because of too many noises in the images.

Fig. 5. Three images with the same histogram

Fig. 6. Two lightness variance images

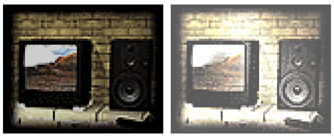


Fig. 7. Two lightness variance images

Fig. 8. Two rotation variance images

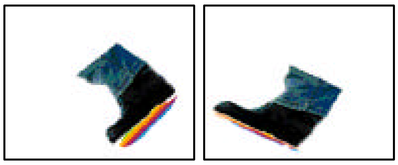


Fig. 9. Two rotation variance images

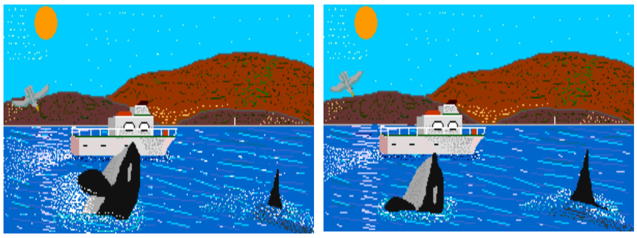


Fig. 10. Two noise variance images

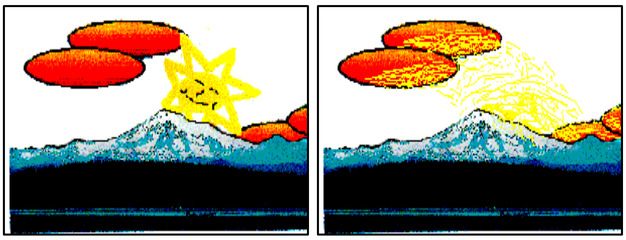


Fig. 11. Two noise variance images

6. Conclusions

Color histogram is insensitive to noises, orientations, and resolution of images, and color matching can be processed automatically without human intervention. However, it is only a global attribute without any local information. Consequently, the color feature cannot effectively distinguish two distinct images even if they have quite different appearances. To make the retrieval more accurate, this paper introduces the concept of run-length feature. The feature can effectively characterize the direction, area and geometrical shape of an object. However, extracting the run-length feature is time-consuming. Therefore, this paper provides a more efficient representation, semi-run-length. On the basis of the semi-run-length feature, the proposed image retrieval system can offer better accuracy of recognition than that based on the color histogram proposed by (Mehtre *et al*., 1995).

References

Adoram, M. and Lew, M. S. (1999). IRUS: Image Retrieval Using Shape. *IEEE International Conference on Multimedia Computing and Systems* 2, 597-602.

Bimbo, A. D. and Pala, P. (1996). Effective Image Retrieval Using Deformable Templates. *Proceedings of the 13th International Conference on Pattern Recognition* 3, 120-124.

Bimbo, A. D. and Pala, P. (1997). Visual Image Retrieval by Elastic Matching of User Sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(2), February, 121-132.

Grarris, M. D., Blue, J. L., Dimmick, G. L., Geist, J. D., Grother, J., Janet, S. A., Omidver, O. M. and Wilson, C. L. (1997). Design of a Handprint Recognition System. *Journal of Electronic Imaging* 6(2), April, 231-243.

Kolpatzik, B. W. and Bouman, C. A. (1995). Optimized Universal Color Palette Design for Error Diffusion. *Journal of Electronic Imaging* 4(2), 131-143.

Mehtre, B. M., Kankanhali, M. S., Narasimhalu, A. D. and Man, G. C. (1995). Color Matching for Image Retrieval. *Pattern Recognition Letter* 16, March, 325-331.

Pei, S. C. and Cheng, C. M. (1995). Dependent Scalar Quantization of Color Images. *IEEE Transactions on Circuits and Systems for Video Technology* 5(2), April 124-139.

Sawhney, H. S., Hafner, J. L. (1994). Efficient Color Histogram Indexing. *IEEE International Conference on Image Processing* 2, 66-70.

Stricker, M. and Swan, M. (1994). The Capacity of Color Histogram Indexing. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 704-708.