Implemented a Facial Recognition Based on Fractal Coding and Quadtree Techniques

Raheem Abdul-Al Sahib Ogla[1] and Yasser Dhiaa Jawad[2]

[1,2]University of Technology, Computer Science Department, Baghdad. Iraq

[1] 110137@Uotechnology.edu.iq
[2] yasserdheaa87@gmail.com

Received 2 March 2017 Accepted 4 May 2017

Abstract

The research aims to design and implement a hybrid algorithm through a combination of fractal coding and quad tree algorithms to build revealed the identity of persons through the recognition of the human characteristics for the destination system. The work is implemented through two phases the first phase is detection phase (Training Phase), to discover information of skin complexion and stored them in the database, some critical coefficient parameters are extracted and stored in coding file like **Peak Signal-to-Noise Ratio** (PSNR), offset bits, Scale bits, mean absolute error (MAE), width and height of the cutting face these coefficient parameters are computed based on coding fractal scheme algorithm.The second phase is the stage of recognition of persons. It is carried out through matching extracted information in the discrimination phase with the information stored in the database in the detection phase. At the stage of discrimination, quad tree algorithm is used as an algorithm searching and matching at the same time in order to accelerate the matching process in both phases as a solution to the problem (wasted time) faced by the fractal algorithm. The work is implemented on color images, with various directions images (forward, 10^o, 15^o, 20^o, 25^o, 30^o, 35^o) the database has been trained on a standard database (MIT) as well as through the images in real time. Experimental results proved that the speedup matching between image faces stored and their information in the database and the image faces want to distinguish them was very high (0.0 1sec), and the accuracy ranged matching between (89% -92%) .

Key Words: Facial detection, Facial recognition, Image Processing, Quadtree

تمثيل تميز الوجة البشري باعتماد تقنيتي ترميزالكسوريات والشجرة الرباعية

رحيم عبد الصاحب عكلة[1] و ياسر ضياء جواد[2]

[1،2]الجامعة التكنولوجية - قسم علوم الحاسبات - بغداد - العراق

الخلاصة

يهدف البحث الى تصميم وتمثيل خوارزمية هجينة من خلال المزج بين خوارزمتي الكسوريات والشجرة الرباعية لبناء منظومة كشف هوية الأشخاص من خلال تمييز خصائص الوجه البشري .تم تنفيذ العمل من خلال مرحلتين المرحلة الأولى هي مرحلة الكشف، لاكتشاف المعلومات الخاصة في بشرة الجلد وتخزينها في قاعدة البيانات ويتم استخراج بعض المتغيرات المهمة وتخزينها في فايل الترميز مثل PSNR، Offset bits، Scale bits ، ومعدل الخطأ المطلق (MAE)، بالاضافة الى عرض وطول الوجه المقطوع. هذه المتغيرات يتم حسابها من خلال خوارزمية ترميز الكسوريات.المرحلة الثانية هي مرحلة تمييزالأشخاص تم تنفيذها من خلال مطابقة المعلومات المستخرجة في مرحلة التمييز مع المعلومات المخزنة في قاعدة البيانات في مرحلة الكشف. في مرحلة التمييز، تم استخدام خوارزمية الكواد تري كخوارزمية للبحث والتقسيم في نفس الوقت من أجل تسريع عملية المطابقة بين المرحلتين كحل للمشكلة التي تعاني منها خوارزمية الكسوريات. العمل تم تنفيذه على الصور الملونة وبمختلفة الاتجاهات (الجانب الأمامي، 35^o, 30^o,25^o, 20^o, 15^o, 10^o درجة) ، قاعدة البيانات تم تدريبها على قاعدة بيانات قياسية (MIT) وكذلك من خلال الصور في الزمن الحقيقي.النتائج التجريبية اثبت ان سرعة التطابق بين صور الأشخاص المخزونة معلوماتهم في قاعدة البيانات (في مرحلة الكشف) وبين صور الأشخاص المراد تميزها في (مرحلة التمييز) كانت عالية جدا (0.01sec)، ودقة التطابق بين المرحلتين تراوحت بين (89% -92%).

الكلمات المفتاحية : الكشف عن الوجه البشري ، تمييز الوجه البشري ، معالجة الصور ، الشجرة الرباعية

Introduction

Face recognition system is one of biometric techniques that uses images of a person's face for determining person identity. This system is needed in many security vital applications like: in surveillance field, in database investigations and control access. figure (1) shows the stage of image face recognition.

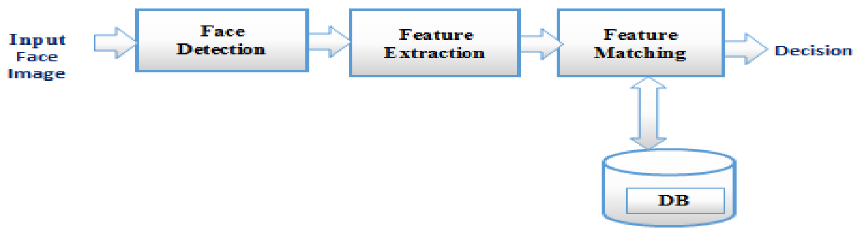


Figure (1) Face Recognition System

Recently, thousands of individuals are suffering from identity theft, and biometrics is mostly considered as possible solution. On this basis, the face recognition process has no need for human intrusiveness within identification and verification procedures in the sense that no physical contact is required to acquire the face image. Contrary to other biometric applications which require a corporation and interaction from human. In the biometric recognition systems, the gallery is constructed from the raw biometric images, which are retained in the database of recognition system. The query (probe) images are often referred to as those images that are obtained during authentication phase [1].
Biometric systems are a pattern recognition systems depending on fingerprint, voice, iris and the most recent gene structure (DNA fingerprint), while face recognition has a key benefit because it can be taken at a declared way and in a hidden way. There are a lot of biometric systems but among the six common biometric features studied by Hietmeyer, with a Machine Readable Travel Documents (MRTD) model facial features scored the greatest compatibility, such as enrolment, renewal, machine requirements, and public perception [2][3].

Related Work

Face recognition and detection has been widely studied for several decades. A lot of work has been done to handle the problem under different conditions, including lighting, pose, expression, etc. A thorough survey can be found in [4]. The recognition process and its effect on face analysis, which we are interested in, has recently attracted research effort. Most work has focused on modelling the recognition process [5][6], face estimation [7, 8, 9, 10], and simulation [11, 12]. In comparison, face verification across ages is far less studied [13].

Concept of Encoding Fractal Images

Fractal coding is concerned with studying and analyzing the local similar blocks within the image as a whole or the local self-similarity between some parts of the image itself, to get the Iterative Function System (IFS) to perform encoding and decoding processes. In the encoding process there was a need for developing the IFS by taking advantage of the state of self-similarity within the image[14]. The development of the (IFS) of the image is done by dividing the image into fixed sized sub-blocks in order to perform self-affine contraction to estimate another sub blocks in the same image. Decoding process used the collage theory to perform the IFS to iterate the base image multi times to establish an estimated image similarity, the image is divided into non-overlapped blocks called range pool, and overlapped blocks called domain blocks, where for each block in the range will matched with the most similar one in the domain blocks in order to get the IFS encoding parameters [15][16][17].

Face Recognition Approaches

There are many approaches have been used by researchers in the field to detect and recognize faces which were used several methods for the purpose of revealing the identity of individuals and get to know their identities. The study and analysis of these methodologies are improved and developed based on the vision and requirements for disclosure of the identity of persons, the type of systems and user application. Some of these approaches are described below briefly: [18][19].

a. **Feature approach**

   Every face has many of the characteristics and unique feature that distinguish it from others. So there are many of the criteria and methods that are used to characterize and measurement

b. **Holistic approach**

   Descriptions face image based on the entire image rather than on local features of the face.

c. **Hybrid approach**

   This approach is the mixing of holistic and feature approaches.

Work Flow of the Proposed System

The main block diagram of the proposed hybrid algorithm consists of two stages training phase and testing phase, many internal processes are implemented and executed, SQL database created to stores matching information between training database and coding file that store related coefficient parameters in the testing phase as shown in figure (2), algorithm (1) shows the main steps of training phase.

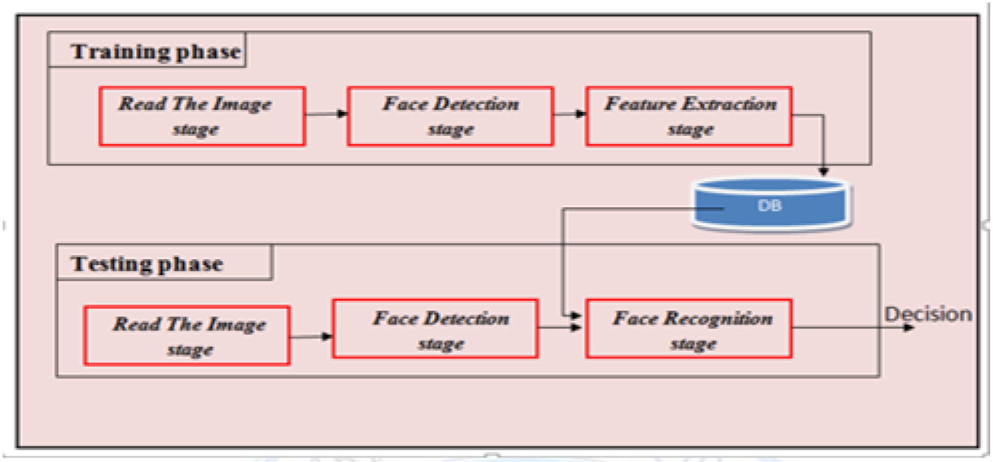


Figure (2) Main Block Diagram of the Proposed System

Algorithm (1) Main Steps of Phase Detection(Training Phase)
Face Detection *stage* **Input:** Set of facial Images/ image file name **Output:** **Store matching Parameters in SQL DataBase**, Build Encoder (ID_Person, ID_Image ,ImageName,ImagePath,PSNR, MAE, MSE, CR, BR, OffSetbits, ScaleBits, Elapsed Time, and other)
Step1: Perform Color Transformation From (R,G,B) To (YCbCr) ,(YIQ) and (HSV) **Step2:** Locate area of the skin in three color space separately and Collect Extracted Skin Areas. **Step3:** Perform Noise Removing using median method. **Step4:** Filling holes by using Dilate edge of (3x3) structure element to remove small holes in edge and Erode edge using (3x3) structure element to disjoint connection among edges. **Step5:** Face localization and labeling objects by connect component labeling method. **Step 6:** Extraction features of objects. **Step 7:** Detected the Accurate Faces and Clipping the largest one. **Step 8:** Get the Coefficient coding parameter(Width, Height, Scale Bits, Offset Bits, MimErr, MaxErr, Symmetry Cases) **Step 9**: Go to Fractal to implement IFS and PIFS techniques. **Step10:** built the Encoder **Step 11:** Store the necessary Information in The SQL database

Encoding Unit

This unit consists of three modules which are all together responsible for reducing the data size of the desired face image and generate the Partitioned Iterated Function System (PIFS) of range and domain pools stream of data to represent the face image.

1. Range Poll Generation Module

The way used to partition images in this research is a fixed size partitioning scheme because it requires less computational time. This kind of partitioning is done by choosing the size of the block once time in the program, then the image (range) will be divided into non overlapping blocks called "range blocks". Choosing the block size must be done accurately, because a small block size leads to good matching between range and domain blocks but the time of the searching process increases (because there will be a lot of range blocks that should be matched with blocks listed in the domain pool). While, if the block size is chosen large, then the encoding time is reduced and the quality of the reconstructed image, is reduced. The numbers (n_r) of range blocks, whose size is ($k \times k$) pixels, are generated from an image whose size is ($W \times H$) by using fixed blocks partitioning which is determined by the following equation(1) :

$$n_r = \left\lfloor \frac{W}{k} \right\rfloor x \left\lfloor \frac{H}{k} \right\rfloor \text{---}(1)$$

2. Domain Pool Generation Module

This module is responsible for generating another two-dimensional array, called the domain, with size different from the size of the range array. As shown in Figure (3), this module consists of the following two sub modules:

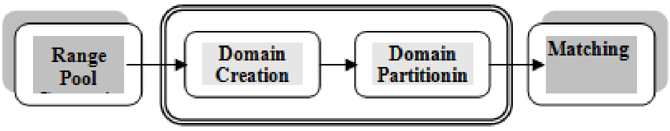


Figure (3) Domain pool generation module

The goal of the domain pool generation is to prepare the templates that are used as suitable approximates for the range block. The chance of getting good approximations increases when there are many domain blocks (templates). The domain array is a two dimensional array with size (H_d xW_d), and this size is quarter the size of the range array, and computing using equation (2)

$$H_d = \frac{H}{2} \text{ , and } \quad W_d = \frac{W}{2} \text{ ,} \ldots\ldots\ldots\ldots\ldots\ldots \text{ (2)}$$

The data of this array are produced from the range array. There are many ways to select the data from range to fill the domain, one of these rules that are used in this research is taking the average of every four (2x2) adjacent elements in range and put it in its corresponding position in domain, Selecting a large step size will decrease the encoding time and the reconstructed image be less quality. Taking into consideration the step size must be less than or equal to the block size. The numbers (n_d) of domain blocks (kxk), generated from an image (WxH) by using fixed blocks partitioning method with jump step (j), and are determined by using the following equation(3) .

$$n_d = \left(\left\lfloor \frac{W-2k}{j} \right\rfloor + 1 \right) * \left(\left\lfloor \frac{H-2k}{j} \right\rfloor + 1 \right) \text{,} \ldots\ldots\ldots\ldots \text{ (3)}$$

the total number of domain blocks (n_m) in the overall range/domain matching processes will be:

$$n_m = 8 n_r n_d \text{ ,} \ldots\ldots\ldots\ldots\ldots\ldots \text{ (4)}$$

3. Matching Module

Suppose there are a range block having pixel intensities ($r_i, \ldots, r_n$) and domain block having pixel intensities ($d_i , \ldots, d_n$). For each range block (r_i) the optimal affine approximation can be represented as :

$$r_i' \approx s d_i + o \text{ ,} \ldots\ldots\ldots\ldots\ldots\ldots \text{ (5)}$$

where, r_i' is the optimal approximated ith pixel value in the range block,d_i is the corresponding pixel value in the domain block,s, o are the scaling and offset coefficients, respectively.The search process implies that all the domain blocks (d_i) listed in the domain pool should be matched with the considered range block, to list out the optimal approximation (r_i'). Figure (4) shows that the matching module consists of the following three sub modules(domain pool generation, Processing, coding) :

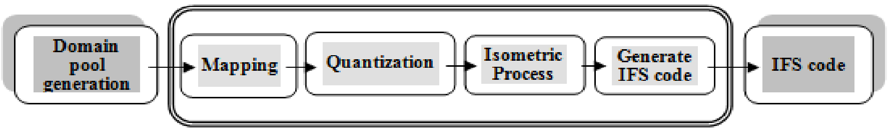


Figure (4) Matching sub modules

To determine the designation parameters at each one mapping process i.e., scale (s) and offset (o), which are named the IFS parameters. These parameters (s) and (o) are computing by applying the amount of sum χ^2 of square errors between r_i' and r_i according to equation (6) :

$$\chi^2 = \sum_{i=0}^{n-1} \left(r_i' - r_i \right)^2 \quad \text{,…………………………..…………………….... (6)}$$

The minimum of χ^2 occurs when: $\dfrac{\partial \chi^2}{\partial s} = 0$ and $\dfrac{\partial \chi^2}{\partial o} = 0$,....……...... (7)

We can obtain the value of (s) by substituting equation (5) in (6) and utilized equation (7):

$$s = \frac{n\sum_{i=0}^{n-1} r_i d_i - \sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} d_i}{n\sum_{i=0}^{n-1} d_i^2 - \left(\sum_{i=0}^{n-1} d_i \right)^2} \quad \text{,……….…..…………...……...…...... (8)}$$

$$o = \frac{\sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} d_i^2 - \sum_{i=0}^{n-1} r_i d_i \sum_{i=0}^{n-1} d_i}{n\sum_{i=0}^{n-1} d_i^2 - \left(\sum_{i=0}^{n-1} d_i \right)^2} \quad \text{,……………………………………......(9)}$$

$$\chi^2 = \frac{1}{n}\left[\sum_{i=0}^{n-1} r_i^2 + s\left(s\sum_{i=0}^{n-1} d_i^2 - 2\sum_{i=0}^{n-1} r_i d_i + 2o\sum_{i=0}^{n-1} d_i \right) + o\left(no - 2\sum_{i=0}^{n-1} r_i \right) \right], \text{(10)}$$

where, d_i is the ith pixel value of the matched domain block, r_i is the ith pixel value of the range block., n is the number of pixels in each block (i.e. the block size).

But if the denominator $\left[n\sum_{i=0}^{n-1} d_i^{\,2} - \left(\sum_{i=0}^{n-1} d_i\right)^2\right]$ equals zero then the above equations become

$$s = 0 \text{ ,……………………………………………………… (11)}$$

$$o = \frac{1}{n}\left(\sum_{i=0}^{n-1} r\right) \text{ ,……………………………………………………… (12)}$$

$$\chi^2 = \left[\sum_{i=0}^{n-1} r_i^{\,2} + o\left(no - 2\sum_{i=0}^{n-1} r_i\right)\right] \text{ ,…………………………………… (13)}$$

Each time the process of conducting matching occurs between the (Range-Domain) and before determine the value χ^2 there are some conditions must be performed to limit the values of (s) and (o) some of these conditions are:

1. The condition that achieves the value of (s) within limited range ($s_{\min} \le s \le s_{\max}$) , minimum scale value (s_{min}) should be not less than (-1), maximum scale value (s_{max}) must be not more than (1) and the following condition must be applied:

   If s < MinScale then s=MinScale Else if s > MaxScale then s=MaxScale

2. The condition that achieves the value of (o) within limited range ($o_{\min} \le o \le o_{\max}$), minimum offset value (o_{min}) should be not less than (-256), the value of maximum offset (o_{max}) should be not more than (511). and the following condition must be applied:

   If o <MinOffset then o=MinOffset Else if o > MaxOffset then o=MaxOffset

4. Generating IFS Coding

After finding out the near optimal IFS code for each range block listed in the range pool, the output will be a set of IFS parameters. The scale and offset quantization indices (i_s,i_o) are stored rather than the actual scale and offset value (s,o) because these indicies need specific numbers of bits (i.e.; n_s, n_o, respectively). Also, the position index (*posI*) of the best matched domain blocks is registered instead of its position coordinates (x_d,y_d). This will decrease the number of bits needed to be saved. The number of the IFS code sets is equal to number of range blocks in the range pool. The IFS mapping parameters are listed in Table (1).

Table (1) The set of IFS mapping parameters registered in the encoding file

Parameters	Description
posI	*Position of the best matched domain block*
i_s	*The scale index value of the best matched domain block*
i_o	*The offset index value of the best matched domain block*
Sym	*The isometric transformation index for the best matched domain block*

Decoding Unit

This unit consists of two modules, as shown in Figure (5); it starts with loading the IFS code and ends with the attractor as output.

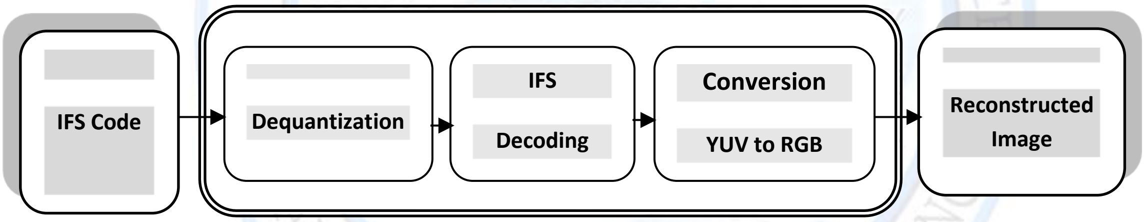


Figure (5) The unit of the decoding process

The decoding process can be summarized by the following steps:

1. Generating arbitrary domain pool. The domain pool could be initialized as a blank image or a piece of image extracted from any available image.
2. The values of the indices of (i_s) and (i_o) for each range block should be mapped to reconstruct the quantized values of the scale (s_q) and offset (o_q) coefficients..
3. Choosing the number of possible iterations, and the threshold value of the mean square error (MSE). At each iteration the following steps are performed:

i. To find the best matching domain block one determine the coordinates of (xd ,yd) for each matched range block of the IFS coefficients (posI) to extract of domain block (d) from temporary domain face image .

ii. For each range block ,its estimated value r_i' is produced by multiplying the identical best matched domain block(d) by the value of scale(sq) ,then the result will added to offset value(oq) depending on equation (5).

iii. The created r_i' block is transformed (translation, reflection, or both) depending on its identical IFS symmetry case value (Sym).

iv. The decoded image is created and filled with the corresponding r_i' block position.

v. Testing if there is any range block, if yes, repeated steps (i, ii, iii).

vi. Down sampling process will perform to generate domain blocks by utilizing the averaging sampling.

4. Evaluating the mean absolute error (MAE) between the regenerated range and previous regenerated image range. If the value of MAE is greater than the threshold value (ε) then the iteration go on and the previous steps (i-vi) must repeated and the iteration will continue until reaching the reconstruct attractor image.
5. The steps (i-vi) must be performed on the three bands(Y,U,V) to generate the attractor of each band.
6. Transforming the reconstructed (YUV) color bands to its original bands (RGB) utilizing inverse transform, then Calculating the fidelity criteria (PSNR, MSE, Elapsed Time, offset bits, scale bits).

Encoding of Fractal Image Algorithm

The fractal image encoding algorithm implemented as follows:

To perform fractal image encoding algorithm some steps can be implemented:

1. The original facial image is divided into non-overlapping blocks (B) called range blocks of size k×k.
2. The domain pool image is generated by overlapping the blocks of range by using down sampling (averaging each four pixels) which is called domain pool blocks (Di) of size $D_{max} \times D_{max}$, usually, $D_{max} = 2B_{max}$ (maximum No. of range blocks).
3. To perform matching process, each block in the domain (Di) is matched with the best block rang block (Ri), through the matching process some of coding parameters are

computed like Mean Absolute Error (MAE) , Standard deviation (Sd) and variant (Var) to obtain the best matching between Ri and Di Blocks, according to specific threshold and smaller MAE value between Rang(Ri) and Domain(Di), those parameters will recorded in coding file and will used in decoding process to reconstructed face image.

4. If there is no matching, block Ri is divided using a quad tree technique, then continue to step 2.
5. End.

In the figure (6) , three face image are selected (P1,P2,P3), Image P1 is encoded image (image recorded), image (P2,P3) stored in database through training phase (face image P2 is the same person as face image P1 with different side view , face P3 is a different person from face image P1 .

1. To retrieve face image (P1) empty Background will initialize and encoding word is used to generate fractal coding parameters.
   a. Decoding face image (P1-1) is iterates fractal coding only once to be reconstructed.
   b. Decoding face image (P1-2) is iterates fractal coding twice to be reconstructed.
   c. Decoding face image (P1-3) is iterates fractal coding three times to be reconstructed.
   d. Decoding face image (P1-4) is iterates fractal coding four times to be reconstructed.
2. When image P1 Utilized face image (P2) as word code to generate fractal codes.
   a. Decoding face image (P2-1) is iterates fractal coding only once to be reconstructed.
   b. Decoding face image (P2-2) is iterates fractal coding twice to be reconstructed.
   c. Decoding face image (P2-3) is iterates fractal coding three times to be reconstructed.
   d. Decoding face image (P2-4) is iterates fractal coding four times to be reconstructed.
3. When image P1 Utilized face image (P3) as codewords to generate fractal codes .
   a. Decoding face image (P3-1) is iterates fractal coding only once to be reconstructed
   b. Decoding face image (P3-2) is iterates fractal coding twice to be reconstructed.
   c. Decoding face image (P3-3) is iterates fractal coding three times to be reconstructed.

d. Decoding face image (P3-4) is iterates fractal coding four times to be reconstructed.

After a few iterations of the same face image but at different face image expressions and postures are experimented as details in (section 6 step 3), it turns out, when face image P1 used face image P2 as (CodeWords) the similar face image to (P1) was (P1-4), which is the face image of record. But when two different face images (P1, P3) are experimented, the reconstructed face image was not clear (P3-4), where there was a number of small blocks unclear in this face image. Based on compute PSNR (Peak Signal to Noise Ratio) we can recognize between face images of individuals with different expressions and directions, where the PSNR between face image (P1-4) and P1 is 29.86, while the PSNR between face P2 and image P1 is 24.65,at the same time the PSNR between face image (P3-4) and face P1 is 18.32, face image P1 is the face of stored in the database, whilst face image P2 is clearly the (recognized) face image.

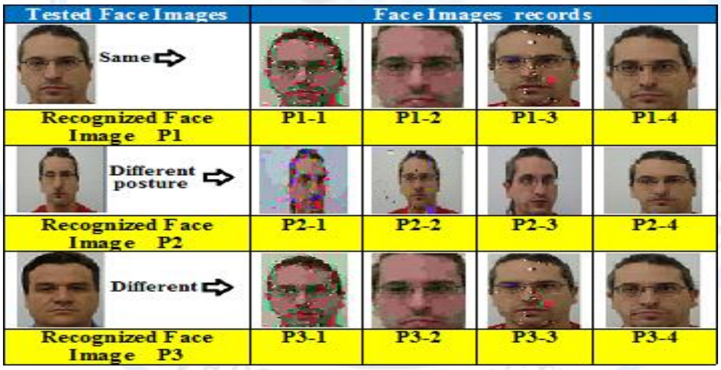


Figure (6) Facial Face Image Recognition

Experiment Steps: Recognition of face image

In the detection phase, the data set of Cambridge University database is based in the practical experimental, about 400 facial images of 40 peoples with various expressions and posture are tested and stored in the database , their fractal encoding parameters is stored in encoding files ".cmw" those database records and encoding file will used in recognition phase . the process person verification can described as :The input is set of Cambridge University database recorded images as training database, then , the face images encoding (some parameters are computed) and stored in database as code words (assume n face images in the database), for each face image that is tested encoding file will be created (.cmw) as a encoding dictionary , so, there are n ".cmw" encoding files. The n ".cmw". In verification (recognition) process encoding files are iterated until nearest greater PSNR are obtained by using face image recognize decoding program, which gives n decoding face images. The PSNR between face images of recorded face image and decoding face image is based, if the face images have PSNR is higher than PSNR of decoding face image ,then swapping process of values of PSNR (the near greatest value is accepted) , the swapping process continue until all files ".cmw" are examined. when the appropriate value of PSNR is obtained, the database will opened and all PSNR in the database will examined, the record that have near highest PSNR will recognized and considered the recognized face image. If the value PSNR less than all of the PSNRs in the encoding file(.cmw), then there is not face image of the same person that are stored in the database, so, there is two options , rejected or registered in the database and added to set of face images to become new person in the database. The equation (14) is used to compute PSNR for verification face image.

$$PSNR = 10\log_{10}\frac{H \times W \times 255^2}{\sum\left(I_{i,j} - \bar{I}_{i,j}\right)^2} \qquad (14)$$

Where H, W are the height and width of the face image, $I_{i,j}$ is the recorded face image and $\bar{I}_{i,j}$ is the decoding face image in the database.

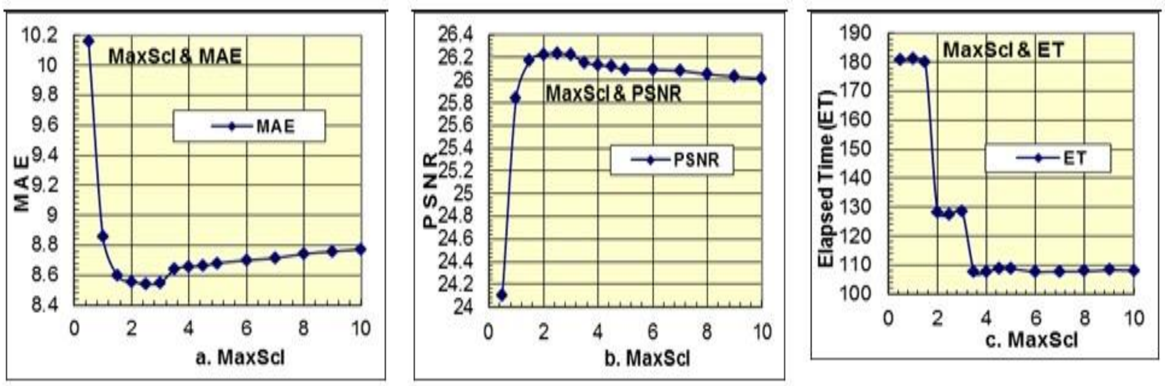
Experimental Results

Work is devoted to present the results of the conducted tests to study the fractal performance on the face image. Some of the famous fidelity measures (MSE,MAE, PSNR, Elapsed time) have been used to assess the quality of the recognized face image. To estimate the efficiency of the generated colored FIC , several tests have been performed and their impact of results are shown figures(7-13). These tests to discover the impact of coding coefficients on performance detection and recognition of face images: different values of parameters are experimented (Block size, MinScale Bits and MaxScale Bits , Jump Step, offset bits, Scale bits, Step size, Minimum block Error) to study their impacts on important measurement fidelity (MAE, MSE, , PSNR, Elapsed time) respectively at different values of ScaleBits parameter. Table (5) shows the effects of TotalBits on the coding performance parameters. All the tests have been conducted on face images of the data set of Cambridge University database that are resized to (200x200, 24 bits per pixel). The value of the parameters MaxOffset and MinOffset were fixed in all these tests, at (255) and (-255), respectively. To test the effect of any parameter, the values of other coding parameters have been fixed to specific values. This testing mechanism was repeated for all the considered coding parameters as follow.

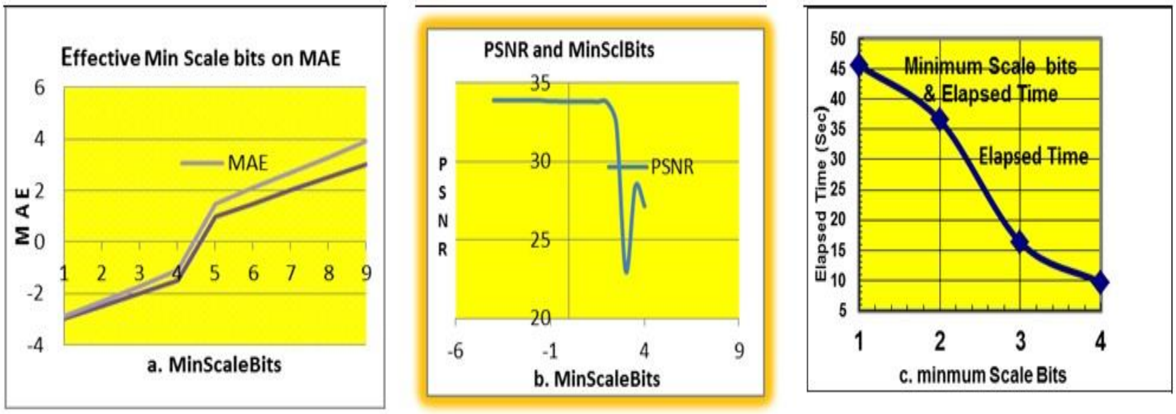
a. Testing Maximum scale and Minimum Scale : This band of tests was dedicated to study the impact of MinScale, and MaxScale on the efficiency parameters of the recognized face image. In these tests the value of ScaleBits and OffsetBits were set 6 bits, BlockSize=(4x4), StepSize=2. Pwrmissible Error value ε =0.3, MAE=0.2 as a default values.

Figure (6) shows a sample of the recognized face image. Figures (7،8) show the effects of MaxScale bits behave on performance parameters MAE, PSNR, and Elapsed time respectively. The effects of MinScale and MaxScale parameters could be summarized by the following remarks:

1. At MinScale= -1.5 and MaxScale=3 led to highest value of PSNR.
2. MAE is directly proportional with MinScale and MaxScale.
3. Encoding Time is nearly constant with MaxScale and MinScale variation.



Figure(7) Effective of Maximum Scale Bits Tests



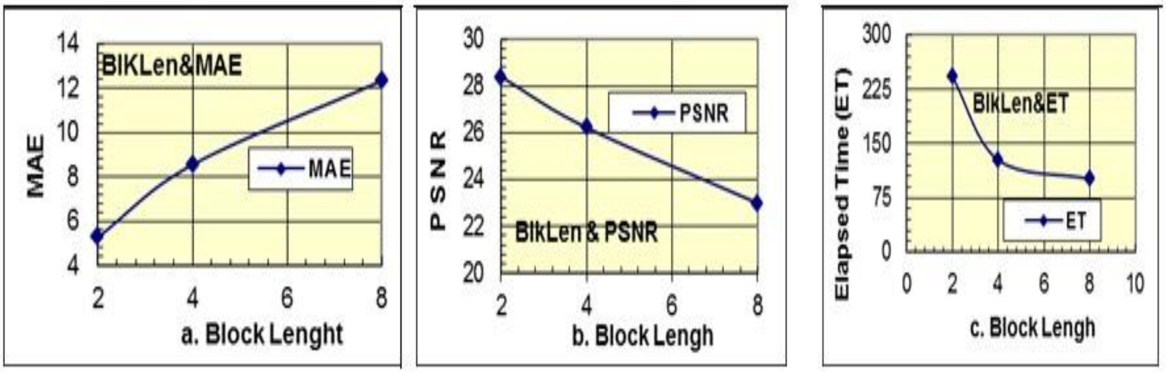
Figure(8) Effective of Minimum Scale Bits Tests

b. Block Size Tests: This band of tests aimed to study the impacts of the block size parameter on the recognized face image efficiency parameters. In these tests the value of parameters are assigned to as fixed values: ScaleBits=6, OffsetBits=6, MinScale=-1.5, MaxScale=3, StepSize=2, ε_o =0.3, MAE=0.2.

The above listed results show in in the figure(9).

1. When the BlockSize is taken as (4x4) the value of PSNR is more appropriate than other sizes.
2. The increase in BlockSize causes an increase in MAE, MSE, and MaxDiff.
3. PSNR and elapsed time are inversely proportional with BlockSize.
4. BlockSize (2x2) shows highest value of PSNR with more time than the case (4x4) and(8x8).

Vol: 13 No:4 , October 2017
DOI : http://dx.doi.org/10.24237/djps.1304.324B
P-ISSN: 2222-8373
E-ISSN: 2518-9255

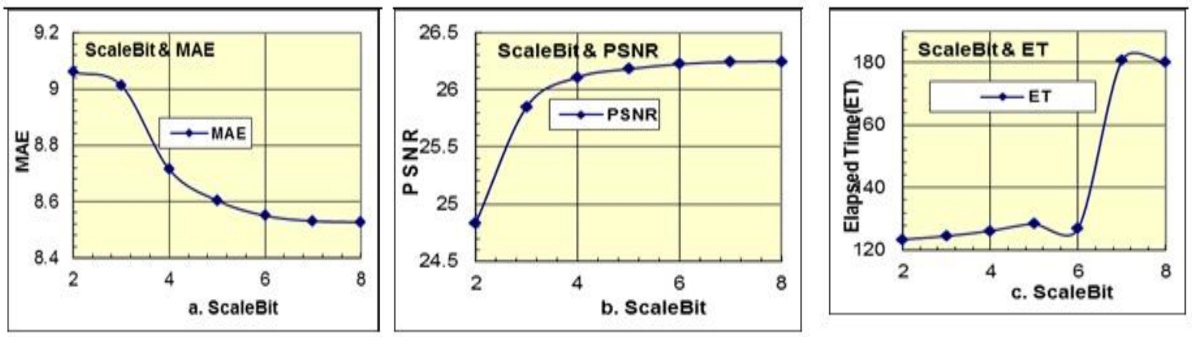


Figure(9) Effective Block Lenght Tests

c. Scale Bits and Offset BitsTests

The effective of scale bits are discussed as shows in figure (10) and explained in the following steps:

1. Increasing scale bits values will decreasing the MAE.
2. Increasing scale bits values will increase the quality of Image face (PSNR).
3. Elapsed time (ET) will Increasing smoothly at specific value Scale bits and suddenly be a big jump when the value of scale bits increased.

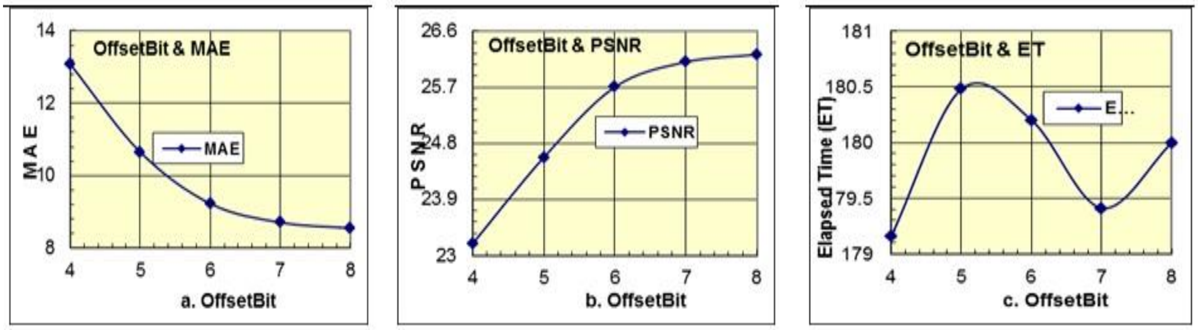


Figure(10) Effective of Scale bits

The effective of offset bits are discussed as shows in figure (11) and explained in the following steps

1. Increasing scale bits values will decreasing the MAE.

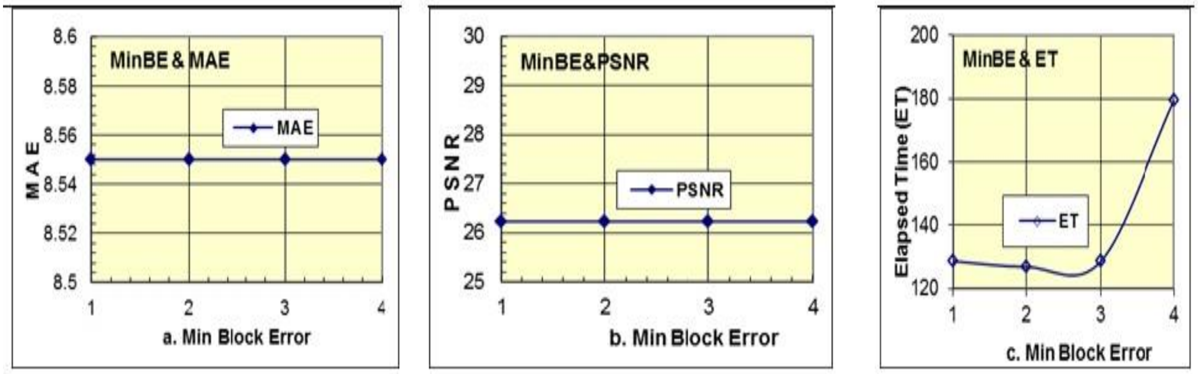
2. Increasing offset bits values will increase the quality of Image face (PSNR).
3. Effective of offset bits on the Elapsed time (ET) be like sine wave.



Figure(11) Effective of Offset Bits

d. Minimum Block Error

The values of Minimum Block Error (MBE) have an effect on the time of matching, quality of facial image and error ratio is simple (few).as shows in figure(12).



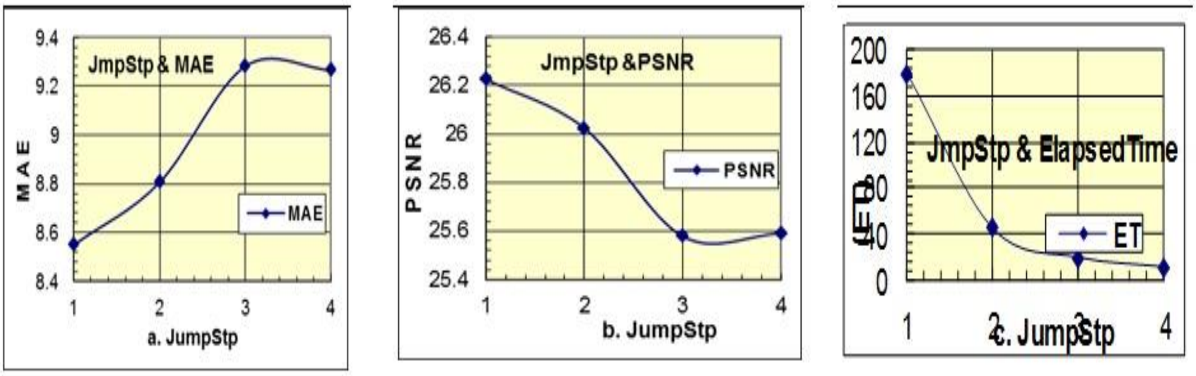
Figure(12) Effective Minimum Block Error

e. Jump Step

The effective of Jump step can be described as in figure (13) and the following steps.

1. Increasing the values of jump steps will increase the MAE.
2. Increasing value of jump steps values will decreasing the quality of face image (PSNR).

3. Increasing values of jump steps will reduce elapsed time (ET).



Figure(13) Effective of jump Steps

The table(2) show the no. of attempts to test some of testing face images (Identified persons) with those stored In the data base set in different face images size, the last column shows the effects of minimum size(2*2) of block will produce highest recognition ration(92.3%). The Recognition Ratio (RR) and False Attempts Ratio (FAR) the following equations:

1. Recognition Ratio (RR): is defined as a ratio between the numbers of correct recognition decision to total numbers of attempts:

$$RR = \frac{Correct \quad Attempts \quad Numbers}{Attempts \quad Total \quad Numbers} \times 100\% \quad \ldots\ldots\ldots\ldots\ldots(15)$$

2. False Attempts Ratio (FAR):is defined as a ratio between the numbers of false recognition decision to total numbers of attempts:

$$\text{FAR}== \frac{False \quad Recognition \quad Attempts \quad Number}{Attempts \quad Total \quad Numbers} \times 100\% \quad ..(16)$$

Example: if the attempts total number is 150 and correct attempts number is 130 then

RR= (130/150) = 0.86 $\times$ 100=86.6% Recognition Ratio

150-130=20 false

FAR= (20/150) = 0.13 $\times$100=13.4% False Attempts Ratio

Table(2) Recognition Face images Using Evaluation Criteria

Recognition Rate				
Control parameter	Evaluation Criteria			
Block Size	Total Attempts	False Attempts	False attempts Ratio (FAR)	Recognition Rate (RR)
2*2	155	12	7.7%	92.3%
4*4	155	16	10.3%	89.7%
8*8	155	25	16.1%	83.9%
16*16	155	30	19.3%	80.7%

The first column in the table(3) shows the effects of jump steps value (1) will produce highest recognition ration (92.3%). when the jump steps is increased gradually the false attempts ratio (FAR) will increased and recognition rate (RR) will decreased.

Table (3) Recognition Face images Using Jump Steps

Recognition Rate				
Control parameter	Evaluation Criteria			
Jump Step	Total Attempts	False Attempts	False attempts Ratio (FAR)	Recognition Rate (RR)
1	155	12	7.7%	92.3%
2	155	19	12.2%	87.8%
3	155	26	16.7%	83.3%
4	155	33	21.2%	78.8%

Table(4) shows bands of tests was dedicated to study the impact of permissible error(ε0) on the effectiveness parameters of the recognized face image. the range tested values of permissible error(ε_0) between (0.1-6), the values of other parameters is fixed to specific default values , where the values of ScaleBits and OffsetBits is set 6 bits, BlockSize=(4x4), StepSize=2, MAE=0.2.when no. of attempts be (155) and value of permissible error(ε0) is (0.1) the recognition ration is (92.3%) and false attempts ratio is(7.7%),when the values of (ε0) increased the values of FAR and (RR) are relatively related decreasing.

Table (4) Recognition Face images Using Different values of permissible error

Recognition Rate				
Control parameter	Evaluation Criteria			
ε0	Total Attempts	False Attempts	FAR	RR
0.1	155	12	7.7%	92.3%
0.2	155	13	8.4%	91.6%
0.3	155	15	9.7%	90.3%
0.4	155	16	10.3%	89.7%

0.5	155	17	11%	89%
0.6	155	18	11.6%	88.4%
0.7	155	20	12.9%	87.1%
0.8	155	22	14.2%	85.8%
0.9	155	23	14.8%	85.2%
1	155	25	16%	84%
2	155	27	17.4%	82.6%
4	155	30	19.4%	80.6%
6	155	34	21.9%	78.1%
8	155	37	23.9%	76.1%
10	155	40	25.8%	74.2%

The final recognition rates

Coding parameters that give the best results implemented in the table (6) when the values in table (5) are examined.

Table (5) Testing Different Values of Coding Parameters

Maximum Scale	Minimum Scale	Block Size	Jump Step	Down sampling	Permissible Error Value(ε)	Iterations no
3	-1.5	2*2	1	Divided by 4	0.1	7

Table (6) Final best results

Recognition Rate (Evaluation Criteria)			
Total Attempts	False Attempts	False attempts Ratio (FAR)	Recognition Rate (RR)
155	20	12.9%	87.1%

References

1. Al-Hilo, Eman A., and Rusul Zehwar. "Fractal Image Compression by YIQ Color Space", 2014 International Conference on Computational Science and Computational Intelligence, 2014.
2. Al-Hilo, Eman A., and Loay E. George. "Study of Fractal Color Image Compression Using YUV Components", 2012 IEEE 36th Annual Computer Software and Applications Conference, 2012.
3. Abed, Fadhil Salman. "A Proposed Encoding and Hiding Text in an Image by using Fractal Image Compression", International Journal on Computer Science & Engineering, 2012.
4. W. Zhao, R. Chellappa, P. Phillips, A. Rosenfeld. “Face recognition: A literature survey”. ACM Comp. Surv., 35(4):399-458, 2003.

5. N. Ramanathan and R. Chellappa, “Modeling Age Progression in Young Faces”, CVPR, I:387-394, 2006.

6. Wang, Xing-Yuan, Xing Guo, and Dan-Dan Zhang. "An effective fractal image compression algorithm based on plane fitting", Chinese Physics B, 2012.

7. Y. H. Kwon and N. da Vitoria Lobo. “Age classification from facial images”. CVIU, 74:1-21, 1999.

8. A. Lanitis, C. Draganova, and C. Christodoulou, “Comparing Different Classifiers for Automatic Age Estimation”, IEEE Trans. SMC B, 34(1):621-8, 2004.

9. N. Ramanathan and R. Chellappa, “Face Verification across Age Progression”, IEEE Trans. on Image Processing, 15(11):3349-3361, 2006.

10. S. K. Zhou, B. Georgescu, X. Zhou and D. Comaniciu, “Image Based Regression Using Boosting Method,” ICCV, I:541-548, 2005.

11. A. Lanitis, C. J. Taylor, and T. F. Cootes. “Toward Automatic Simulation of Aging Effects on Face Images”. IEEE Trans. PAMI, 24(4):442-455, 2002.

12. J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen, “A Multi-Resolution Dynamic Model for Face Aging Simulation”, CVPR, 2007.

13. E. Nakamura. "Fractal image coding combined with subband decomposition", ICECS 2001 8th IEEE International Conference on Electronics Circuits and Systems (Cat No 01EX483) ICECS-01, 2001.

14. Muhammad Kamran. "A novel domain optimization technique in fractal image compression", 2010 8th World Congress on Intelligent Control and Automation, 07/2010.

15. Jack Kozik. "Facial image recognition based on fractal image encoding", Bell Labs Technical Journal, 06/2010 .

16. A., Eman, and Rusul Zehwar. "Comparison Fractal Color Image Compression using YIQ and YUV Color Model", International Journal of Advanced Computer Science and Applications, 2015.

17. Loay E. George. "Isometric process behavior in fractal color image compression by zeromean method", 2010 2nd International Conference on Computer Engineering and Technology, 04/2010.

18. Gaoping Li. "Accelerating Fractal Image Encoding Based on Gray Value Moment Features of Normalized Block", 2008 International Symposium on Computer Science and Computational Technology,12/2008.

19. Jianji Wang, , and Nanning Zheng. "A Novel Fractal Image Compression Scheme With Block Classification and Sorting Based on Pearson's Correlation Coefficient", IEEE Transactions on Image Processing, 2013.