

# Eigenmosaic: Reconstruction of Images Using PCA Based Mosaic Method

Amirhassan Monadjem<sup>1</sup>, Seyyed Mohammad Saeed Ehsani<sup>2</sup>, Mohammad Hadi Edrisi<sup>3</sup>

1,2- Faculty of Computer Engineering University of Isfahan, Isfahan, Iran

Email: monadjemi,ehsani@eng.ui.ac.ir

3-Faculty of Computer Engineering University of Isfahan, Isfahan, Iran

Email: h.edrisi@eng.ui.ac.ir (Corresponding author)

Received: May 2015

Revised: June 2015

Accepted: June 2015

## ABSTRACT:

In this paper we have introduced new techniques for generating compound images, or mosaics, which include reconstruction of main images using homomorphous square patches. These patches are extracted from another image, called the Key Image, and are placed next to each other in order to produce the mosaic image. To perform that, criterions like color average, variance, and histograms have been used so far due to their important role in describing the facade of an image. PCA is a valuable statistical method in recognition of pattern among the data; that can be used on images as well. These methods basically help us to separate and analyze the image parameters. PCA implementation results will be used as filters, eigen filters, for the main image leading to extraction of constructor lines and different parts of image construction texture. In the proposed approach, the extracted parameters from the main and key image can be a local or global reference to select parts in order to produce the mosaic. The results of experiments have shown that our PCA-based method performance well in building mosaics.

**KEYWORDS:** Mosaic, Image Parameters Extraction and Analysis, Analogy, PCA.

## 1. INTRODUCTION

Digital mosaics are images made of a set of small images or patches. These small patches cover the mosaic image like a cobblestone. Using the source image and according to the available set of patches and restrictions for replacement or shape changes, we may obtain different kinds of digital mosaics. Computer graphic tries to explain mosaics under a wider title called "Unreal Images Surfacing"[1], [2], [3].

Mosaics follow two goals in general: causing similarity with the original image and, visual aesthetics goals. Composition of mosaics includes two main steps; analysis and extraction of main image features, and search on patches database for finding the best matching patch. Search on the database includes analysis of patches features on the basis of image features and selecting a strategy to compare the patches. There are different methods proposed for developing mosaics like artificial mosaics and puzzle mosaics. In this paper, we have emphasized on concept of "causing similarity with the main image" and square patches are arranged in horizontal and vertical order to build up the mosaic. The main goal is to find criterions for image surface or its patches so that we can reconstruct images using other patches. In this way, the value of each criterion is justified according to the

description of an image features and the description of visual similarity [4, 5].

Section(2) is an examination of mosaic techniques. In Section(3) we talk about PCA analytical method and its math basis and implementation in extraction of image features. In Section(4) we introduce new techniques for creating mosaic images. The Proposed method will be used in local content based image retrieval. Three methods are discussed in details and the results of their implementation is proposed.

## 2. MOSAIC TECHNIQUE

In recent decade, efforts have been made to simulate mosaics that are of classic art methods, in a digital way. In this section we review some ordinary mosaic methods. Lots of commercial software such as Adobe Photoshop<sup>f</sup> have provided mosaic filters in order to generate cobblestone images, despite their appearance is far from the classic mosaics. As an example part (a) and (b) in Fig.1 are generated using Adobe Photoshop<sup>f</sup>. There are other techniques which have tried to merge geometric and computational concepts with image processing methods in building mosaics [6], [7].

For example, Hae Berley [1] uses Veronoy diagrams. He places the patches in their place randomly, then colors each region with the color sampled from the

image. This method turns image into a cobblestone, but the patches color varies and the features of image edges are ignored. This method has been used under “Crystallization” in most of implementations [1]. (See Fig.1(c)).

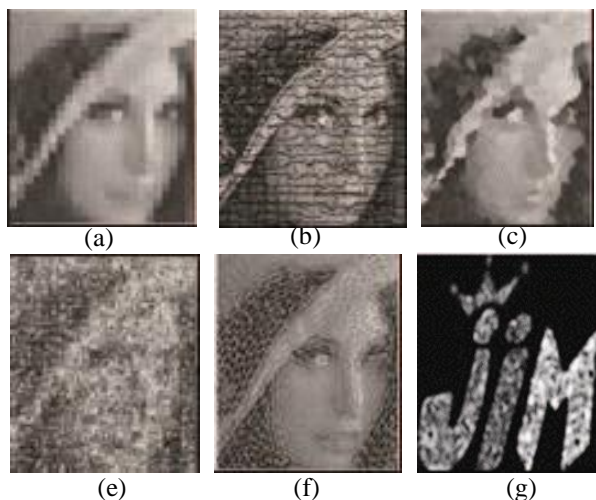
Pictorial mosaic divides the image into a rectangular network of patches. In this method, the algorithm searches a big database of images for nearest block to the main image pixels. This method also has ignored the edges and has a different output with classic mosaics [1, 2, 3]. See (Fig.1(d)).

Hawsner has reached good results using Veronoy center oriented diagrams, edges features,  $L_1$  distance (Manhattan), and accelerating graphics hardware in order to optimize the process time [8]. See Fig.1(e).

Kim *et al.* have invented a technique in which they employ patches with desired shape for constructing the image [9]. The idea behind this is very similar to the image mosaics but the results are very different and interesting. (See Fig.1(f)).

Kim and Pellacini introduce a mosaicing technique, called Jigsaw Image Mosaic (JIM), where image tiles of arbitrary shapes are used to compose the final picture [10]. The idea is quite similar to the photomosaic, but the final effect is very different and interesting. (See Fig.1(f)).

We discuss all three principal techniques of mosaic in the next part of paper. The first technique is called Artificial Mosaic, the second is Puzzle Mosaic developed by Gallo *et al* [11], and the third technique is Pictorial Mosaic which was developed by Di-Blusi and Petralia [1].



**Fig. 1.** Several verity method to produce mosaics [12]: (a) and (b) generating using Adobe Photoshop<sup>®</sup>, (c) create mosaic using Veronoy diagrams, (e) create mosaic using Veronoy center oriented diagrams.

## 2.1. Artificial Mosaics

A closer look at the classic mosaics shows that:

- 1- Every patch has the same color.
- 2- The patches may differ in shape for some logical reason but are usually convex.
- 3- It is not a most that the gaps between the patches are to be minimum.

Instead, we have to use them as a graphical element to show the edges, lines, or borders.

Artificial mosaics method considers the three above features with a description of the way real mosaics work as a source: “To create a mosaic, artists specify the area of the shapes in the image which they want to show, then these shapes will be filled with a sequence of parallel arcs and finally sort the patches in line with these arcs.” Fig. 2 shows this method.



**Fig. 2.** An example of a real manmade mosaic [13].

In general, the patches should follow the main lines and patterns of the image. These patterns are called Vector Guidelines. Hawsner *et al* tried to formulate the problem in a mathematical frame. The mathematical formula presented by Hawsner has presented the mosaic construction from an image as an optimization problem [1].

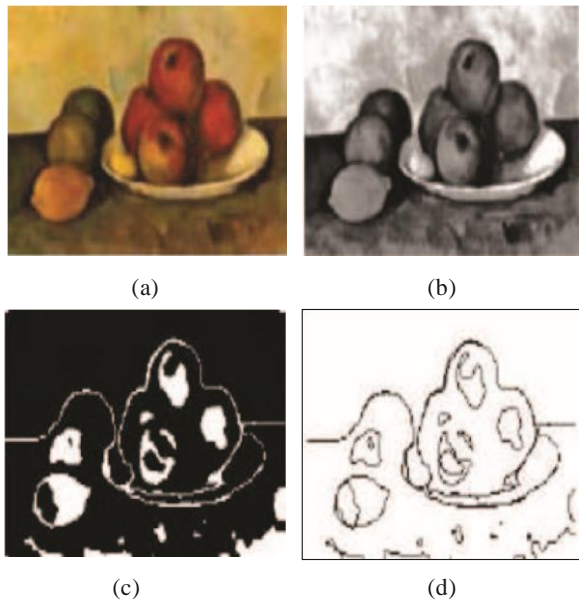
Hawsner results are interesting but in practice they are more complex than his suggestion because of the cost of functions that satisfy art and conceptual needs of mosaics. The optimization process can be even more complex when we model the vector guidelines using splines. Despite that, we can find meaningful mathematical arcs for putting the patches in place, developing a solution to work with sharp and technical points of this algorithm is not an easy job.

According to what said above, in artificial mosaic method we try to find a comprehensive and innovative procedure based on images instead of an abstract mathematical process. The first step of this algorithm starts with finding image guidelines. The guidelines are mostly close to sensual comprehensions. Especially if the source image is a picture, the results differ from the results of algorithms which specify the edges of picture.

Firstly, we perform histogram equalization on the image. Then the image will be convolved with a two dimensional central Gaussian functions. If we call the new image  $I_1$  (a normalized instance of the source image), with the average  $\mu$  and variance  $\sigma^2$  and  $T = \sigma^2/4$ , then we have  $I_2$  as follow:

$$I_2(x, y) = \begin{cases} 1 & \text{if } (|I_1(x, y) - \mu| > T) \\ 0 & \text{in the rest of points} \end{cases} \quad (1)$$

Threshold is specified in an experimental or trial and error way. Then we obtain the guidelines by convolving  $I_2$  with a Laplacian edge detector and eliminating isolated points (Fig.3).



**Fig. 3.** Finding directed guidelines mosaic[11]: (a)Input image[14], (b)Equalization result, (c)Patched image, (d)Directed guidelines.

The inputs of artificial mosaic algorithm are source image and its guidelines now. The next step of the algorithm is to define three following matrices: Distance transformation matrix ( $dtM$ ) which is equal to minimum of distances between each pixel and existing guidelines, and two other matrices called Gradient matrix ( $gM$ ):

$$gM = \arctan \frac{dtM(x, y + 1) - dtM(x, y - 1)}{dtM(x + 1, y) - dtM(x - 1, y)} \quad (2)$$

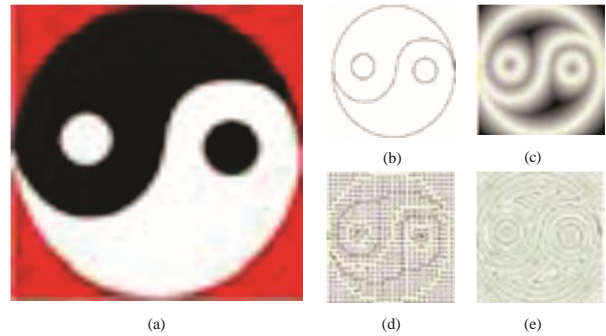
And line level Matrix ( $lM$ ) that can be obtained from  $dtM$  matrix:

$$lM(x,y) = \begin{cases} 2 & \text{if } (M = tSize) \\ 1 & \text{if } (M = 0) \\ 0 & \text{in the rest of points} \end{cases} \quad (3)$$

where  $M$  is:

$$M = \text{module}(dtM(x,y), 2.tSize) \quad (4)$$

In which  $tSize$ (tile size) is an integer selected by the operator that determines the size of patches. These steps are shown in (Fig.4).

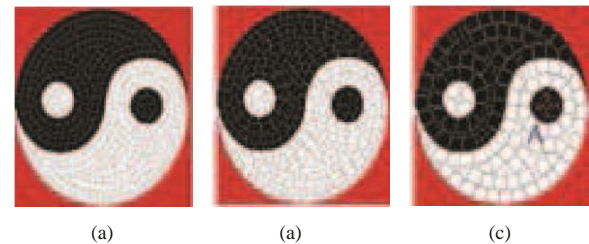


**Fig. 4.** Inputs and matrices used in artificial mosaic algorithm[9]: (a)The Input image. (b)The directional guidelines.(c)The distance transform matrix. (d)The gradient Matrix. (e)The level line matrix

The following algorithm is used for placement of patches, while there is a chain of pixels with value of 2:

- 1- Find a chain
- 2- Start by finding a pixel on it and follow the chain
- 3- Place the new patches on the path with regular distances (the orientation of patches will be obtained using  $gM$  matrix information)

For each patch one unique color equal to the center pixel in source image is selected. Fig.5 shows some of the obtained results using different patch sizes. In general, if patches have a big scale compared to the image details, patches get a nongeometric shape. There is no such problem in real mosaics.



**Fig. 5.** Obtained results using artificial mosaics[15]: (a) Patch size  $5 \times 5$ . (b) Patch size  $10 \times 5$ . (c) Patch size  $10 \times 10$

There are several ideas for improvement of artificial mosaic visual results that we do not mention them here.

## 2.2. Puzzle Mosaics

We formulate the mosaic problem in the following way: If  $I_2$  is a image in  $R^2$  space and we have a chaotic

set of small images and also a chaotic network of  $N$  cells on the image, then find  $N$  patches in data set and place them in existing network where: patches are independent and each cell is covered with one patch [11].

Puzzle mosaic is a method based on the following three assumptions: 1- *Artificial method for finding guidelines.* 2- *Picture random segmentation (using Veronoy diagrams).*

3- *Analogy of patches and shapes.*

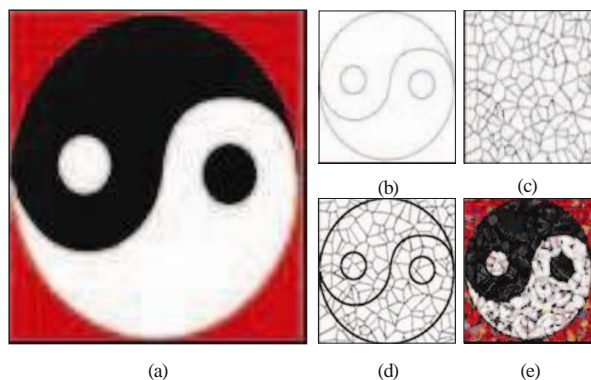
4- *A fast search technique called Antiple tree.*

Puzzle mosaic algorithm steps[16]:

Puzzle mosaic algorithm starts with finding directional guidelines using the artificial mosaic method procedure, however, in order to obtain better results we do not follow all those steps.

Then we produce a Veronoy diagram equal to the size of the source image, Set of points will be selected randomly and cardinality selection is carried out by operator, (see Fig.6(c)). Then guidelines and this diagram will be added in an image called  $R$  as shown in (Fig.6) part (d).

In the final step, for each  $R_i$  region in  $R$  we calculate its features vector using previous part method. Then using Antiple tree technique [17], we find the nearest features vector to each  $R_i$  features vector in data set. After finding the best match, we calculate the appropriate angle for placing the patch in final image. Fig.6 shows the procedure steps.



**Fig. 6.** Puzzle mosaic algorithm steps[9]: (a) Input image. (b) Guidelines. (c) Veronoy diagrams. (d) Final division. (e) Output image

### 2.3. Illustrative Mosaics

Mosaic problem is formulated like below in this method: If  $I_2$  is an image in  $R^2$  space, and we have a dataset of small rectangular images, and also a regular network of  $N$  rectangular cells, then find  $N$  patches in the dataset and sort them in the network in a way that the patches are independent and each cell is covered by a patch [11].

The bottleneck of in each image-mosaic algorithm is to search a big database of images in order to find the best match. This search is usually ordinal and time consuming. Image-mosaic method introduces a new method in order to speed up the search process. This technique is based on Antiple tree and provides us good results. Antiple tree is usually suitable for a search among numerous number of records matched to  $(x,d)$  measurement space[18]. We can divide the proposed algorithm into two step: 1- *Collecting data and creating database, it is really simple.*

Each existing image is divided into a 9 part ordinal network. Then each cell in this network is also divided into another  $3 \times 3$  ordinal network. Then the average of RGB (for color images) or grayscale average (for grayscales) is calculated for each cell of sub network. After these steps a  $X$  vector with 27 (For RGB) of 9 (For Grayscale) value exists for each sub cell.  $X$  is called sub cell vector and can be used for a search in Antiple tree. 2- *The second step is producing Image-mosaic.*

To do so, we process the input image like existing images in the database in order to obtain a specifications vector for each sub cell. The obtained patch will be resized in a way that it covers its relevant sub cell in main image when the best match is found in Antiple tree search. Fig.7 shows the obtained results of Image-mosaic method implementation.



**Fig. 7.** An example of image mosaic algorithm output[10].

### 3. PRINCIPAL COMPONENTS ANALYSIS (PCA) AND ITS APPLICATION IN EXTRACTION OF IMAGE FEATURES

PCA is considered as a useful and capable statistical method in data analysis. In fact, it is a valuable result of using linear algebra and is widely used from nerves physiology to computer graphics. This method is an ordinary technique in extraction of existing patterns in data and is also a way to decrease the dimensions of a complicated dataset to obtain meaningful hidden data in it [19].

PCA is a parameter free method in extraction of meaningful data. This can be both a pro and/or a con. Being parameter free means this method is both easy and widely used. On the other hand, it prevents us from considering specific features of a problem, however, the PCA results are usually used as inputs to other problem specific methods. PCA steps include elimination of dependency among data using statistical concepts like variance and covariance, and also linear algebra. These make some restrictions that lots of ordinary dataset do not have [19].

Directions, Lines (Vectors), or patterns which data show the most variance or distribution along are the main features. Features with a lowest energy are usually considered noise or information error sometimes, but according to the case, they are considered as main features. PCA goal is to extract these features. In case the problem is to decrease the data size in a way, main features (equal to the number of the desired dimensions) are chosen and data set is converted to a new one in which the elements are not correlated and are separated on the basis of mentioned features [19].

When we concern about pattern recognition, these features, are considered as inputs of other functions so that the next processes are implemented on them. For instance, in image patterns recognition, acquired features are considered as the impulse response of a few FIR filters, in image patterns recognition system and in fact the image is filtered in the next steps. The result is generating detail images that each has certain features of the main image that decomposed [20].

In this part we mention the mathematical structures and concepts used in PCA method, then we discuss the image filtering using that method briefly.

#### 3.1. An exact definition of DATA

In order to achieve our goals of using the PCA, we need a more specific definition of data. In each sampling in time or place, we obtain a set of data with different substances (such as information about a certain pixel in different patches of an image). The number of these substances (for instance the number of pixels in each patch of an image) determines the dimensions of dataset.

Generally, each instance of data is a vector in  $m$ -dimensional space in which  $m$  is the number of dataset

dimensions. These  $m$ -dimensional vectors are a linear combination of the base vectors set with unique length. The goal of PCA is to find a base which is a linear composition of the main bases and explain the features of dataset in a more efficient way.

For example if  $X, Y$  are two  $m \times n$  matrices that are connected with  $P$  linear transform vector,  $X$  is the main dataset and  $Y$  is a new presentation of it:

$$PX = Y \quad (5)$$

We define the following variables:

- $p_i$  for rows of  $P$
- $x_i$  columns of  $X$
- $y_i$  columns of  $Y$

Eq.5 Shows a change in bases and can have different interpretations as:

1-  $P$  is a matrix that transforms  $X$  to  $Y$  2- Geometrically  $P$  is a rotation or extension that transforms  $X$  to  $Y$

3.  $P$  rows,  $\{p_1, \dots, p_m\}$  are a new set of vectors with new base to show  $X$  columns

Considering the limitations of being linear, the problem is limited to find the suitable bases vectors. Rows of  $P$  ( $\{p_1, \dots, p_m\}$ ) will be  $X$  components in this transformation.

#### 3.2. Variance criterion and PCA goals

Existence of chaos relates to two potential parameters in a system: noise and data dependency [22]. Noise ratio must be low in any set; otherwise we can not extract any useful information. We have no absolute criterion to measure the noise. Noise is measured in a case oriented method. One ordinary criterion to measure noise is Signal to Noise Ratio (SNR). Signal to noise ratio is defined as:

$$SNR = \frac{P_{signal}}{P_{noise}} \quad (6)$$

Where  $P$  is average power. We generally assume  $SNR \gg 1$ .

Data dependency problem is a bit more ambiguous. Data dependency can have two inferences: firstly, data can follow a specific pattern, secondly data may have redundancy. In other words, some patterns that data follow them, may have no value from the informational point of view, or some data may be transformations of some others. These two points of view are the same in fact, and the difference is in the application, not in the solution method. In fact, eliminating data dependency can result in possible redundancy; and finding patterns existing in data.

#### 3.3. Covariance Matrix

A simple method to find data dependency is some computations among data like variance. The variance is a criterion to find distribution around a variable while we need a criterion to describe distribution among some variables [22].

If  $X_1 = a$  and  $X_2 = b$  and name other row vectors  $[X_3 \dots X_m]$ , we can define  $X$ , as an  $m \times n$  matrix like this: Each row of  $X$  or  $X_i$  refers to all evaluations of a specific type, and each column of  $X$  shows a set of evaluations of a unique sampling among all dimensions. define covariance matrix called  $S_x$  like this:

$$S_x = \frac{1}{n-1} X X^T \quad (7)$$

Obtained  $S_x$  shows correlation between each possible pair of evaluations. A big covariance between a pair shows a high dependency (see Fig.8(c)). Zero covariance shows an absolute independence (see Fig.8(a)).

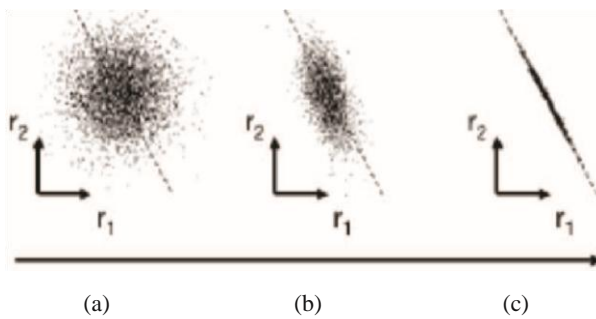


Fig. 8. Data Dependency [23]

So,  $S_x$  matrix determines all type of relations between evaluation pairs. If we want to work on  $S_x$  and obtain  $S_y$ , we need to know which features must be optimized [23].

**3.4. Diagonal Covariance Matrix**

If we aim to reduce dependency, we need to lessen dependency between variables. More specifically, to eliminate dependency, covariance must be zero among separate evaluations. Apparently all non diagonal elements of  $S_y$  are zero. So the redundancy elimination solution is to make the matrix diagonal. There are several ways to make the matrix diagonal. PCA selects the simplest way to keep its generality. This method is similar to a change in bases vectors ( $P$ ). To develop such a method, we have these PCA assumptions:

1- *PCA assumes all bases vectors are orthonormal:*

$$P_i \cdot P_j = \delta_{ij} \quad (8)$$

2- *If  $P$  matrix is the new bases matrix, PCA selects a normalized direction in the  $m$  dimensional space in first step, in a way that variance in that direction in  $X$  matrix is maximum.*

This direction is called  $P_1$ . In the next step, another direction among biggest remained variances is selected. But because of orthonormal assumption, the search is limited to directions which are vertical to the previous directions. The result is an ordered set of  $P_s$  as main

criteria. Second assumption provides us with a method to judge importance of each main direction. For instance, for each  $P_i$  variance can determine that how principal is it. So we can distinct each  $p_i$  bases vector using its relevant variance.

**3.5. A review on assumptions and restrictions**

1- *Linearity:* linearity limits the solution to bases vectors change.

2- *Variance and average criterions:* average and variance describe probability distribution function completely.

Gaussian distribution is the only probability distribution with zero average that can be described with variance completely. So  $X_i$  probability distribution must be Gaussian. Diversion from Gaussian distribution invalidates this assumption. The assumption that guaranties covariance matrix and SNR are able to show dependence and noise correctly.

3- *Big variances have high energies:* this assumption is based on that the main data have big SNRs. So main parts with bigger variances specify main features, and parts with smaller variances are called noise. (As mentioned, this assumption is not satisfied in all PCA applications)

4- *Main parts are orthonormal:* this assumption results in a simplification that PCA is soluble using linear algebra techniques.

5-

**3.6. Finding a solution for PCA (Covariance eigenvectors)**

In conclusion, this is the goal: Find orthonormal  $P$  matrix in which  $Y = PX$  and  $S_y$  matrix be diagonal and equal to  $\frac{1}{n-1} Y Y^T$ .  $P$  rows are main parts of  $X$ .

$$\begin{aligned} \frac{1}{n-1} Y Y^T &= \frac{1}{n-1} (PX)(PX)^T S_y = \\ &= \frac{1}{n-1} P(XX)^T P = \frac{1}{n-1} P A P^T \end{aligned} \quad (9)$$

We defined symmetrical matrix  $A$  equal to  $XX^T$ . Now if we can prove that  $A$  can become diagonal using eigenvectors orthonormal matrix, we have a way to find  $P$ . We use two linear algebra theorems: 1- *A symmetric matrix can be diagonal using an orthonormal matrix.*

2- *Inverse of an orthonormal matrix is equal to its transpose.* For symmetric matrix  $A$  we can use theorem 1 and achieve  $A = EDE^T$ . in which  $D$  is a diagonal matrix and  $E$  is a matrix made of eigenvectors  $A$ , that are arranged in columns order next to each other. Now we select the matrix  $P$  in a way that each row  $P_i$  is a eigenvector of  $XX^T$ . Using this  $P = E^T$ . Using this equation in Eq. 9 and using theorem ( $P^{-1} = P^T$ ), we have  $S_y$  as:

$$\begin{aligned}
 S_{..} &= \frac{1}{n-1} P A^T P^T = \frac{1}{n-1} P (P^T D P) P^T \\
 &= \frac{1}{n-1} (P P^T) D (P P^T) \\
 &= \frac{1}{n-1} (P P^T) D (P P^T) \\
 &= \frac{1}{n-1} (P P^T) D (P P^T) \\
 &= \frac{1}{n-1} D n
 \end{aligned}
 \tag{10}$$

So selecting  $P$ , as said before, can make  $S_y$  diagonal and PCA results are achievable in matrices  $P$  and  $S_y$  as:

- Main parts of  $X$ , are eigenvectors of  $XX^T$  or rows of  $P$ .
- $I_{ih}$  diagonal value of  $S_y$  is  $X$  variance in direction of  $p_i$ .

In case of working with images, if we imagine that rows of  $X$  matrix include grayscale, of a part of image, and rows of  $P$  matrix change in a way that each turn into a square matrix, we can conclude that the image is made of these matrices. So these matrices according to their energy (related eigenvalue) are considered meaningful elements and we can say each one presents a specification of image and their set constructs the image.

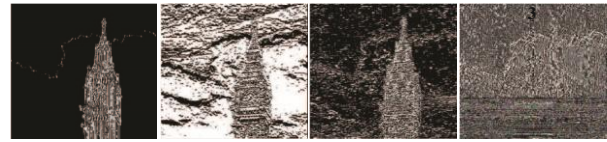
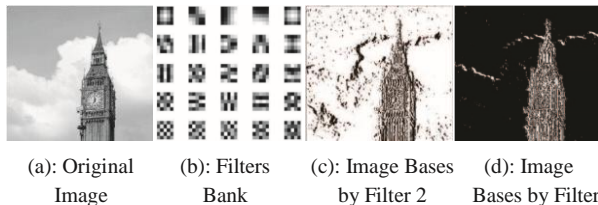
So if we filter the main image using each of these matrices, we obtain new images called detail or eigen images that only include the patterns related to what eigenvector represents.

According to what mentioned so far, eigenvectors obtained using PCA, make a filter bank that can analyze or decompose the image into meaningful components. On the other hand, we can retrieve the main image from its relevant detail/eigen images due to the orthonormality of eigenvectors [3].

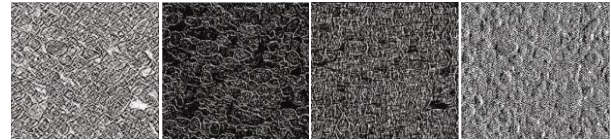
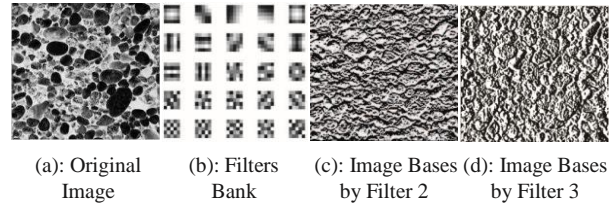
We do the filtering process using a 2 – dimensional convolution that its impulse response is square eigenvector matrices. The input is the main image and the output are eigen images.

$$D_i^G = G \otimes F_i = \{1, \dots, N\}
 \tag{11}$$

$D_i$  and  $F_i$  are eigen image and eigenvector, and  $N$  shows the number of existing pixels in patches of the main image. In Fig.9 and Fig.10 two images are shown with their relevant detail or eigen images and the filter bank. The number of patches in a 512x512 image are 10404, 5x5 patches. To construct source images, among existing 25 filters, filters 2 to 7 are the most energetic filters.



(e): Image Bases by Filter 4 (f): Image Bases by Filter 5 (g): Image Bases by Filter 6 (h): Image Bases by Filter 7  
**Fig. 9.** (a) Original image, (b) Filters bank, (c-h) Images produce by filters 2 to 7 among the best energetic filters



(e): Image Bases by Filter 4 (f): Image Bases by Filter 5 (g): Image Bases by Filter 6 (h): Image Bases by Filter 7

**Fig. 10.** (a) Original image, (b) Filters bank, (c-h) Images produce by filters 2 to 7 among the best energetic filters

#### 4. INTRODUCTION OF A NEW TECHNIQUE FOR IMAGE MOSAIC, EIGENMOSAIC

In this part, we introduce, called eigenmosaic, a new mosaic technique. Algorithms that we introduce differ from previously mentioned algorithms in section(2) in many aspects. However, it is rather similar to illustrative mosaic method.

In this paper, dataset is patches of an image that is divided into an ordered network of  $N$  cells. Each cell constructs dataset elements and we call the whole image, the key image. In other words, we define mosaic method as:

*Selection of patches of key image A and merging them with source image B in order to construct destination image B', in a way that destination image satisfies goals like analogy, aesthetics, etc.*

$$B'(i,j) = \alpha B(i,j) + (1 - \alpha)A(m,n)
 \tag{12}$$

$i$  and  $j$  are patch location in destination and source images.  $\alpha$  shows the merge coefficient and how important are  $B$  elements in construction of destination image.  $\alpha$  is in  $[0,1]$ .

In all techniques that will be introduced in this paper, except in random selection of patches of key image test,  $\alpha$  is assumed zero. This means that we only have considered key image patches to create the destination image.

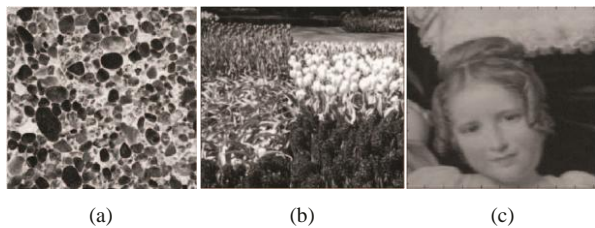
To simplify the implementation, square patches are considered with fixed dimensions and images are black and white (Gray Level). In all techniques smallest comparison part, is grayscale levels and 256 levels is considered.

Firstly, two techniques that include random selection, averaging (using variance factor) and using histograms, our selection base, are patches features solely and we do not mention image features like patterns, edges, and corners. Meanwhile, the results are acceptable and mostly reconstruct main image with its features.

In the last technique, PCA based method or Eigenmosaic, comparison and placement factor is patches. However, in comparison and placement level, these patches are not main image patches but source images patches. These source images are obtained by filtering the main image by eigenvectors. It is obvious that these source images contain information about general features of image like patterns, edges, and corners. So this method can be a good platform for further research in developing mosaics.

In all instances presented in this section, key images are considered constant. Their consistency results in fairly comparison amongst different methods. For faces mosaic, we use Fig.11(c), for landscapes Fig.11(b) and for textures Fig.11(a). Key images size is  $9 \times 9$  or  $5 \times 5$  pixels. The pictures size is  $512 \times 512$ .

We used MATLAB<sup>r</sup> to implement all techniques discussed in this paper.

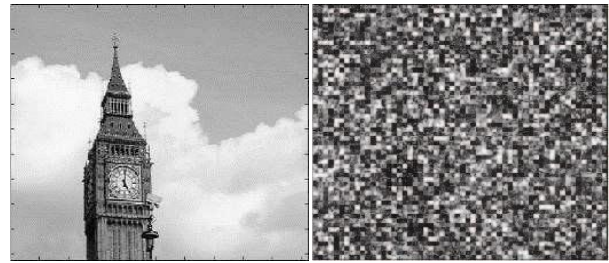


**Fig. 11.** (a) For face mosaic,(b) For landscapes mosaic,(c) For textures mosaic[14].

#### 4.1. Method 1: Random selection of patches

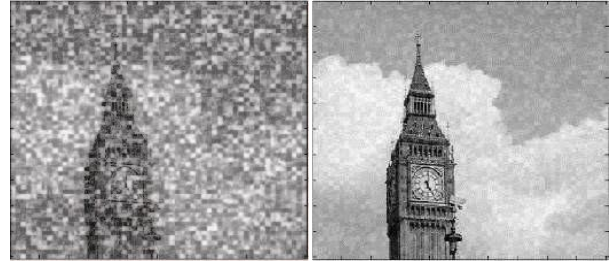
In this method, basic selection is absolutely random. No comparison is done on the selected patch in key image and the selected patch to be replaced it. We merge acquired patch with key image patch with a specified coefficient instead.

In this method, similarity of the destination image with the key image is obtained using merge coefficient or  $\alpha$ . When  $\alpha$  is low, (for example 0.1), the main image is blurred completely. with coefficients around 0.5, we have main image blurred and like a background and if coefficient is 0.9, main image is completely visible and just its color is affected. Fig.11 shows this technique with three mentioned  $\alpha$ .



(a): The Original Key Image

(b):  $\alpha = 0.1$



(d):  $\alpha = 0.5$

(e):  $\alpha = 0.9$

**Fig. 12.** Mosaic using random selection of patches and merging them with key image

Next, that  $\alpha$  merge coefficient is considered zero in other methods, this can prove how important is a patch features in restoration of images. Even if this patches are analyzed without considering the main features of image like edges, corners, and generally patterns.

#### 4.2. Method 2: Averaging and using Variance criterion

In this method we work locally on images. Working locally means considering pixels and a neighborhood around them. Average and variance are two features that are widely used because these two criterions are correlated with images appearance. Average is a criterion to show light reflection or glory average, and variance is a criterion to show contrast [21].

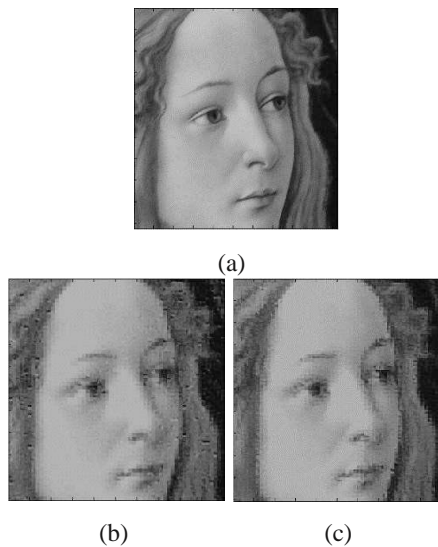
Average filter moves through the image. It replaces the pixel value with the average of grayscale of pixels around it. This diminishes isolated points or noises and may result in blurring of edges.

Average criterion is usually used in combination with variance. In most of applications the averages will be named as an effective threshold to gain the expected results. As using average filter results in blurring of the image, using average criterion based on patches selection, results in some ambiguity in the destination image. Variance criterion is used in combination with average to improve the selection quality. The steps of mentioned algorithm are as follows:

1- *Producing feature vector for each member of dataset*; first we divide key image into an ordered network of  $N$  cells with desired size. Then we produce each pixel's feature vector which includes grayscale average of pixels existing in each cell with their



variance. So each feature vector has two parts and we have feature vector equal to number of patches in the key image. *Sorting dataset in order to improve search time:* feature vector are sorted based on average. This improves the search time in comparison with the sequential search considerably. 2- *Source image analysis and producing destination image:* each source image is divided into an ordered set of N cells. For each cell, we produce a feature vector like last part and then search dataset for nearest average to current average. After finding the desired vector in dataset, we search other vectors around to find the one with nearest variance. This vector represents a patch of key image that we will select.



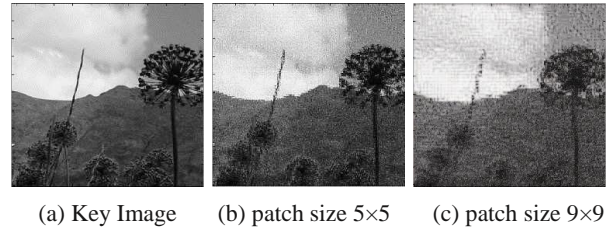
**Fig. 13.** (a)Key image. (b) Mosaic image with average criterion only. (c) Result of using average and variance. As we can see, some improper areas in (b) are refined in (c)

Using variance beside average omits some deficiencies of average criterion as the comparison base, but the main criterion is still average. Research show that variance cannot be a good comparison criterion in order to restore images solely.

If search radius for variance matching be more than needed, average criterion efficiency will decrease, but if this radius be kept limited, it can improve destination image selection quality. Using trial and error method, this radius is considered 2.0 in each side of feature vector. Fig.13 shows some usages of averaging method with and without using the variance criterion. (Patch size is considered  $9 \times 9$ ).

The more cluttered main images are, the lower will be the efficiency of averaging method because of its blurring feature. In such images, Analogically speaking the importance of a method that can detect edges,

corners, and patterns in an image, is clearer. Fig.14 is an example of a cluttered key image with lots of details:



**Fig. 14.** An image with lots of details

### 4.3. Method 3: Using histograms

Using histograms is very common in digital image processing. They provide a general description of image appearance. In fact they are first order probability of grayscales in image. In a discrete form horizontal axis elements show grayscale levels and vertical axis elements show probability of grayscales in image. In other words:

$$p_r(r_k) = \frac{n_k}{k}, \quad k = \{0, 1, \dots, L - 1\} \quad (13)$$

In which  $L$  is the number of grayscale levels,  $p_r(r_k)$  is the probability of  $k^{th}$  grayscale,  $n_k$  is number of  $k^{th}$  level repetition in image and  $n$  is the number of all pixels in image. In general, the more histogram diagram is monotone, the better is image quality. This is the basis for some methods like histogram equalization.

These methods try to equalize histograms based on histogram information. These can provide us with valuable information about a completely ambiguous image sometimes. In some cases, histogram information can be used instead of some parts of image locally. It is clear that in these cases detailed information is mentioned (the application is in textures or images similar to textures). In this case, histogram equalization moves through image and shows its detailed local information more clearly [21].

In method 3, we do not use histograms to improve image quality but as comparison basis in patches selection. We also assume  $0 \leq k \leq 255$ . The steps are as follows:

1- *Producing feature vectors for each member of dataset:* feature vector for each patch in key image here is its histogram diagram. So in first step, we process each patch of key image to evaluate probability of grayscale. We assign information of each patch to a metric space in this way. We can use some clustering techniques to organize the dataset. Clustering can be done using Euclidean distance between two histograms or any other distance measure that can compute the distances in this process. One existing method to cluster is to produce Antiple tree. Clustering using

antipole tree reduces search time from  $O(n)$  to  $O(\log(n))$ .  
 2- *Source image analysis and producing destination image*: like averaging method, we divide the main image into an ordered network of  $N$  cells. For each cell, we obtain its feature vector (Histogram). Next step is to find the nearest histogram in the dataset to current match. If clustering is done in a dataset, this task is like a tree search. Otherwise, search will be sequential. Comparison basis for search in this task was the Euclidean distance. Obtained results show that considering Euclidean patches distance in compared to patches average distance, has a lower efficiency in generating result images analogically.

As we can see in Fig.15, histogram methods have a better impact in maintaining glittery of images and result in a lower ambiguity. In general, if we compare the histograms using a more efficient method, this method could be more powerful in generating similarity.



(a) Path size 5×5

(b) Path size 9×9

Fig. 15. An example of using histogram method

#### 4.4. Method 4: Using PCA analysis and eigenfiltering

We explained basis of the PCA analytical method in Section (3). As mentioned before, PCA is widely used in problems that need to extract patterns, or reduce noise, or both. This method is widely used in computer graphics and machine vision. PCA shows existing patterns in the dataset analytically. These patterns can be used in mosaic images generating as a criterion to compute the similarity that will be discussed later. Although PCA main application is to analyze textures, it provides us good results on general image as well [24].

##### 4.4.1. Proposed Algorithm

1- *Generating appropriate filters*: firstly we divide key image into an ordered network of  $K$  cells. Existing members in each of  $K$  cells will generate rows of a new matrix called  $G$ .  $G$  is a  $K \times n^2$  matrix in which  $K$  is number of patches in key image and  $n^2$  is the number of

existing elements in each patch. By calculating the covariance of  $G$ , eigenvectors and matching eigenvalues can be calculated as:

$$C(X) = E[(X - X^-)(X - X^-)^T] \quad (14)$$

$$(C(X) - \lambda_i I)e = 0 \quad (15)$$

In which  $X^-$  is average of each column of  $G$ ,  $I$  is unit matrix,  $\lambda$  is eigenvalue and  $e$  is eigenvector. Covariance matrix dimensions are  $n \times n$  and covariance matrix eigenvectors produce  $n^2$  vectors that each one turns into a  $n \times n$  rectangular matrix and builds up the filter bank.

Based on the PCA, elements of this filter bank cover all  $n \times n$  existing relations among pixels. As described in Section(3), bases images will be obtained by filtering the main image by this filter bank.

$$B_i^G = G \otimes F_i = \{1, \dots, n^2\} \quad (16)$$

Filters, appropriate to their relevant eigenvalues, are main elements of image and we can have any number of those filters in our filter bank, the more the number of these filters is, more details can be acquired from image and image restoration can be done with a better quality. On the other hand, the smaller size of patches is, the smaller filters will be and they filter smaller and more detailed patterns from image. Generally, size and number of filters should be selected according to the situation and application in order to gain the best results.

(Fig.16)(a) and (b) show Venus pictures filtered using  $17 \times 17$  and  $5 \times 5$  filters. Second to seventh filters have been used among the most energetic filters to produce source images. It seems that bigger filters have worked better in producing general image lines, not considering its details.

##### 2- Generating eigenvector from the source images:

This step includes generating eigenvectors to use in comparison and selection. As mentioned before, comparison basis is selection and placement of patches, with this exception that patches are source image patches here. Source images are also extracted from both main images and key image. Each image has  $N$  base cells and each cell includes  $M$  relevant cells in other source images ( $M$  is the number of source images of an image). Here each cell's eigenvector is equal to sum of relevant cells. In comparison, these eigenvectors from the main image and key image will be compared with each other and the nearest acquired case will specify place of patch from the main image to be replaced in the destination image. Comparison of eigenvectors is carried out by comparison of their values average distance.



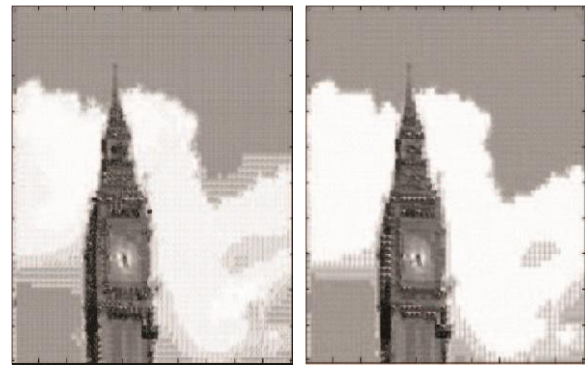
(a) Patches size = filters size =  $17 \times 17$ .



(b) Patches size = filters size =  $5 \times 5$ .

**Fig. 16.** Images are normalized using histogram equalization

Fig.17 shows results of this method. Patches size is  $9 \times 9$  and number of filters used in images generation is different. In Fig.17(a) 9 filters and in Fig.17(b) 25 filters among the most energetic filters have been used in order to generate the source images. *Ad* is the color levels distance.



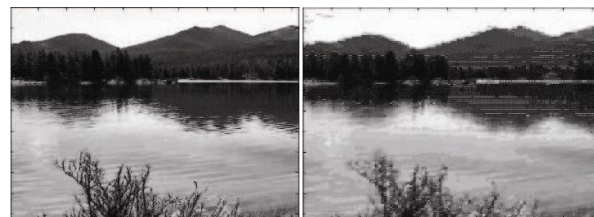
(a) (b)

**Fig. 17.** In part (a) 9 filters and in part (b) 25 filters among most energetic filters have been used in order to generate source images

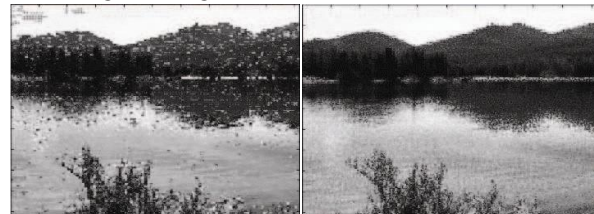
In following pictures, some images have been converted into mosaics using methods 2, 3 ,and 4. Patches size is  $5 \times 5$ . In method 4, all 25 filters have been used in generating the result.

As we can see in Fig.17, method 4 has the least distance in color levels to the main image. Also, obtained image using this method, is less checkered in compared to with other two images that shows this method advantage in extracting image features (even if we use its results locally).

The result of using histograms has a lower quality in compared to with other two images, but in case of restoring real colors, has worked better in comparison with other two methods. We can gain better results from this method using better ways for comparison of diagrams. Method 4 has worked better than other two methods in generating black texture of this material, see Fig. 18, Fig. 19, Fig. 20.



(a)Original Image (b) Method 2:  $ad = 18.53$



(c) Method 3:  $ad = 23.95$  (d) Method 4 :  $ad = 17.70$

**Fig. 18.** An example to compare methods

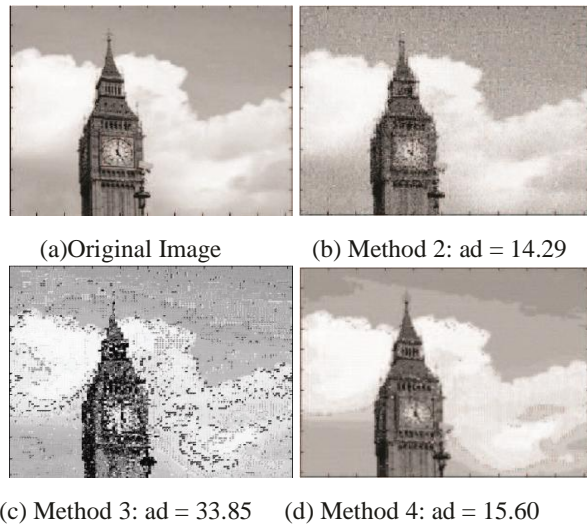


Fig. 19. An example to compare methods

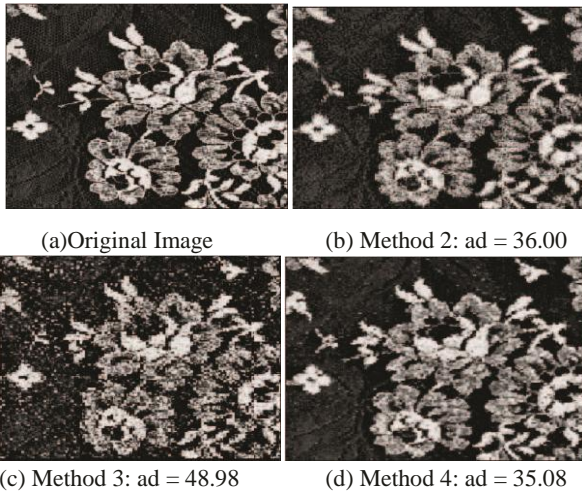


Fig. 20. An example to compare methods

have been presented compared. The results can be seen in table(1). As we can see, from voters point of view, methods 2 and 4 are very similar but method 4 is slightly better than method 4.

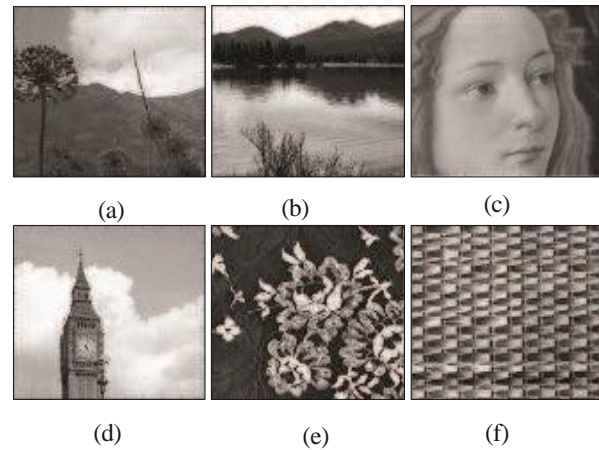


Fig. 21. Main images[14].

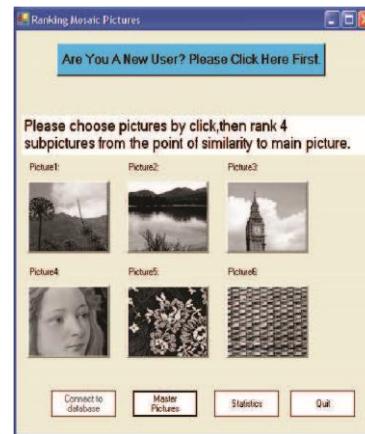


Fig. 22. Polling software

Table 1. similarity with main image percentage, method 1: random selection of patches from key image and  $\alpha = 0.5$ , method 2: averaging using variance, method 3: using histograms, method 4: using PCA and filtering-ranking for similarity is considered from 1 to 5

Mosaicing Method	Method(1)	Method(2)	Method(3)	Method(3)
Image 1	%22.85	%60	%34.28	%60
Image 2	%22.85	%62.85	%37.14	%65.71
Image 3	%25.71	%60	%34.28	%77.17
Image 4	%22.85	%82.85	%54.28	%74.28
Image 5	%25.71	%65.71	%42.85	%65.71
Image 6	%20	%57.14	%37.14	%51.42
Total percentage	%23.33	%64.76	%39.99	%65.71

4.5. Experimental results (polling) from visual similarity point of view

This polling has been run among 10 ordinary people and mosaic images obtained using different methods

5. CONCLUSION

In this paper, we introduced three main methods in order to produce mosaic images. The emphasis was on the different ways of extraction of image features and a

comparison among them in main image restoration from analogical point of view. We came to this conclusion that using the PCA analytical technique and filtering using eigenfilters was the best in restoring main image details and causing visual similarity. Polling results show that using grayscale average in split of its simplicity in concept and implementation has a high performance in visual similarity production.

PCA method has some advantages to grayscale average in:

1- *Restoration of main image details.* 2- *Reduction of checkered look and blurring of restored image.*

using histograms and Euclidean distance as the comparator, has made the highest color level difference with main image, but if we neglect the scratched parts of image, a closer look to obtained results shows that color restoration in this method has been carried out more naturally. We can consider this as a result of histograms, and if we compare them more practically, they can be used as more powerful criterion in this field.

In conclusion, PCA method can be used as a basis for mosaic image producing in mosaics that have random parts with no specific arrangement of meaningful, correlated arcs. the Proposed method could be used in local content based image retrieval, as well as building image structure blocks.

## REFERENCES

- [1] Di Blasi, Gianpiero, "Giovanni Gallo, and Maria Petralia. "Puzzle image mosaic", *In Proc. IASTED/VIIP*, pp. 33-37. 2005.
- [2] Zhang, Lina, Jinhui Yu. "Image mosaics with irregular tiling", *Computer-Aided Design and Computer Graphics (CAD/Graphics)*, 2011 12th International Conference on. IEEE, 2011.
- [3] Monadjemi, A., M. Mirmehdi, and B. Thomas. "Restructured eigenfilter matching for novelty detection in random textures", *learning 5* (2004): 13.
- [4] Abraham, Rintu, Philomina Simon. "Review on mosaicing techniques in image processing", *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on. IEEE*, 2013.
- [5] Battiato, Sebastiano, Alfredo Milone, and Giovanni Puglisi, "Artificial mosaic generation with gradient vector flow and tile cutting", *Journal of Electrical and Computer Engineering 2013* (2013): 8.
- [6] Battiato, Sebastiano, et al., "Digital Mosaic Frameworks?An Overview", *Computer Graphics Forum*. Vol. 26. No. 4. Blackwell Publishing Ltd, 2007.
- [7] Battiato, Sebastiano, et al., "A Survey of Digital Mosaic Techniques", *Eurographics Italian Chapter Conference.*, 2006.
- [8] Hausner, Alejo. "Simulating decorative mosaics", *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.
- [9] Kim, J., Pellacini F., "Jigsaw Image Mosaics", *SIGGRAPH 2002*, str. 657-654
- [10] Kim, Junhwan, Fabio Pellacini, "Jigsaw image mosaics", *ACM Transactions on Graphics 21.3* (2002): 657-664.
- [11] Di Blasi, Gianpiero, Giovanni Gallo, "Artificial mosaics", *The Visual Computer 21.6* (2005), pp. 373-383.
- [12] Battiato, Sebastiano, Gianpiero Di Blasi, Giovanni Maria Farinella, and Giovanni Gallo, "Digital Mosaic Frameworks?An Overview", *In Computer Graphics Forum*, Vol. 26, No. 4, pp. 794-812. Blackwell Publishing Ltd, 2007.
- [13] Roman mosaic of a hyaena found in Tunis, Tunisia. This mosaic was create by an unknown Roman Artist from the 1st century-2nd century A.D. (46.3 x 46.3cm). The Brooklyn Museum. Available : <http://www.hyaenidae.org/Hyaenasin-art-literature.html>
- [14] Available: <http://www.eeng.dcu.ie/hlborgne/icascene.html>
- [15] Dobashi, Yoshinori, et al., "A method for creating mosaic images using Voronoi diagrams", *Proceedings of Eurographics*, Vol. 2, 2002.
- [16] Smith, Lindsay I., "A tutorial on principal components analysis", *Cornell University, USA 51* (2002), 52.
- [17] Cantone, D., Ferro, A., Pulvirenti, A., Recupero, D. R., Shasha, D., "Antipole tree indexing to support range search and k-nearest neighbor search", *in metric spaces. Knowledge and Data Engineering, IEEE Transactions on*, 17(4), 535-550, Chicago, 2005.
- [18] Battiato, Sebastiano, Gianpiero Di Blasi, Giovanni Maria Farinella, and Giovanni Gallo, "A Survey of Digital Mosaic Techniques", *In Eurographics Italian Chapter Conference*, pp. 129-135. 2006.
- [19] Anton, Howard, "Elementary linear algebra", *John Wiley, Sons*, 2010.
- [20] Oppenheim, Alan V., Ronald W. Schafer, and John R. Buck, "Discrete-time signal processing", Vol. 2. *Englewood Cliffs: Prentice-hall*, 1989.
- [21] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing", *Prentice Hall*, 2002.
- [22] Solow, Andrew R., "Detecting change in the composition of a multispecies community", *Biometrics*, pp. 556-565, 1994.
- [23] Comon, Pierre, "Independent component analysis, a new concept", *Signal processing 36.3* pp. 287-314, 1994.
- [24] Bharati, Manish H., J. Jay Liu, and John F. MacGregor, "Image texture analysis: methods and comparisons", *Chemometrics and intelligent laboratory systems 72.1*, pp. 57-71, 2004.