Novel image clustering based on image features for robust reversible data hiding

Mohamed Tahar Ben Othman, Senior Member, IEEE

Abstract—For better data hiding (watermarking) robustness we introduced a new clustering method based on the image features. The image is divided into clusters using the Content Addressable Method (CAM) introduced in our previous works. This is considered as the first level of image segmentation where a clustering function specifies the feature used to divide the image by allocating an Id for each group of pixels. In the second level, each cluster is divided into sub-clusters where each contains the same embedded watermark portion. This duplication is necessary to improving the robustness. To get a higher robustness, in one side, we use the clusters that have more distribution rate of pixels over the image, and in the other side, we build the sub-clusters with higher dispersal rate of pixels over the cluster. The results show that the higher dispersal rate of sub-clusters and more uniform distribution of pixels over the clusters are the better watermark robustness can be obtained.

Keywords— watermarking, clustering, distribution rate, cluster dispersal rate, color images, Content Addressable Method.

I. Introduction

During the last decade, image clustering gained interest in the domain of image processing. It is used for images classification [1], [2], [6], [8], [17] among a set of classifiers like Bayesian, Decision Tree, Markov Model, Neural Network, and Linear Classifiers. The main idea focusses on what is referred to Content-Based Image Retrieval (CBIR) [2], [8]. These types of classification are used mainly for image retrieval in a set or database of images. Most of these works used color image clustering [1], [4], [5], [7].

Clustering is also used in the domain of pattern recognition, for image segmentation, and in many other domains. Image segmentation is used to easily retrieve information in the image itself [3], [9], [10].

This paper presents a new image clustering technique based on the Content Addressable Method (CAM). Unlike most of studied image clustering techniques, the aim of this research is to stress watermarking robustness. Each cluster contains all pixels coordinates of the image that have the same content address produced by the chosen clustering function. A cluster is divided into sub-clusters that are used to hold the same watermark data portion.

The remainder of this paper is organized as follows: A review of related works is presented in Section 2, Image clustering is described in Section 3. The proposed watermark scheme is given in Section 4. Section 5 illustrates experimental results. The conclusion and future directions are presented in Section 6.

II. Related works

In this section we give a brief study of some researches that used clustering in their watermark scheme. Lingling *et al.* used the Statistical Quantity Histogram (SQH) shifting and clustering to construct a new watermark scheme for good robustness and low run-time complexity [11]. They obtain comprehensive performance in terms of reversibility and robustness. Their work mainly focuses on different masking models kind of attacks. In [12], Yan Haowen proposed a watermarking technique by shuffling the cover image, extracting the feature points of the data which are grouped as clusters and then the watermark is embedded in the LSBs. This scheme is proposed mainly to protect copyrights. To the best of our knowledge, no intensive experiments were conducted which give the main drawback of this technique. An enhancement of a watermarking algorithm based on kernel fuzzy clustering and singular value decomposition in the complex wavelet transform domain is proposed in [13]. The host image is decomposed by complex wavelet transform. Then, the singular value of the low-frequency coefficients is selected as an embedded object. Finally, image low-frequency background and high-frequency texture features are used as fuzzy clustering feature vectors to determine the different embedding strength. The results show that the proposed scheme performs well against different kinds of attacks. Against image rotation (5^o, 15^o) the Normalized Correlation (NC) is going from 0.93 to 0.98 depending images. In this paper we are proposing a new approach of image clustering which is called Content Addressing Method (CAM) for color images. Using the defined clusters, watermarks are embedded and extracted. The aim of this paper is to show the robustness of this scheme against image rotation attacks.

This work was supported in part by the Deanship of Scientific Research,Qassim University, under Grant SRD-2701. Financial support is gratefully acknowledged.
M.T. Ben Othman is with Computer College, Qassim University, Saudi Arabia. (Corresponding author phone: +966502221867, e-mails: mtothman@gmail.com, maathaman@qu.edu.sa).

III. Image Clustering

The clustering method is shown in Fig. 1. A configurable number of bits for each pixel of the cover image are used as parameter of a function to produce a value used as address regrouping all entries sharing the same feature. Unlike geographic clustering, in our new proposed clustering technique, the pixels that belong to the same cluster may be distinct to each other in geographical point of view as shown Fig. 2. This helps reduce the effect of attacks. On the other side, clustering is regrouping the pixels with same predefined features and in our case will also be used to store portion of watermark data that has specific criteria which help retrieval and adds robustness of the watermark process. The more the distribution of pixels over clusters is uniform the more the watermark is robust against attacks. In this paper, this function is taken as simple as concatenating three bits from each. This yields to an address of 9 bits which gives a maximum of 512 entries in each cluster.

The clustering function *clusAdd*(x,y,z) should be chosen so the distribution goes uniform as much as it is possible.

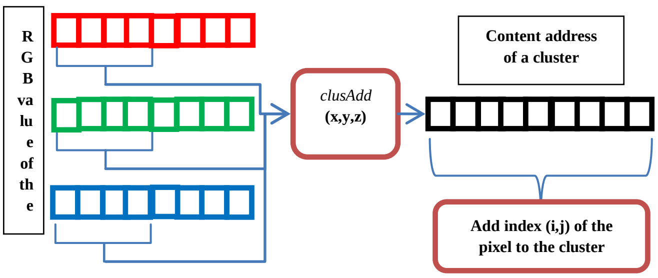


Fig. 1: Clustering Method

The clustering algorithm is going through all cover image pixels and including each pixel indexes in its cluster. Depending on its size, a cluster is divided into sub-clusters.

B. *Embedding Process*

The cover image is scanned using the clusters indexing instead of its own indexing as described in the embedding process in Fig. 4. The pixels in each cluster are divided into a number of portions (or sub-cluster), we are using 8, depending on the number of bits that are used to maintain the sequence number of the portion in the cluster (we are using 3 bits). Into each portion are duplicated three bits from the watermark image.

```
embed_watermark(cover_image, watermark, clusters_set)
{
  For each cluster in clusters_set
  {
      // depending on dispersal rate (dr) and size (sz) divide cluster into sub-clusters.
     preprocess(cluster, dr, sz);
     For each pixel in cluster
      {
       //embed data with parity bit and sub_cluster sequence.
        Embed_data();
      }
  }
}
```

Fig. 4: Embedding Process

The three bits of the watermark image are embedded into the three of the four LSB bits of one of RGB values (say R). Each sub-cluster is used to hold the same portion of watermark data.

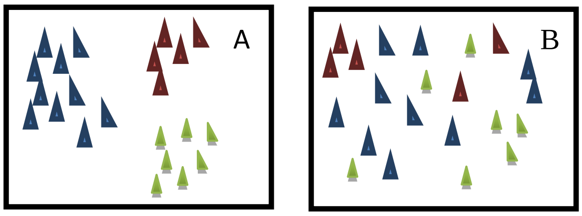


Fig. 2: Geographic (A) vs. Dispersal (B) clustering

IV. Proposed Watermark Scheme

A. *Watermark Scheme algorithm*

As shown in Fig. 3, our watermarking scheme has 5 steps: 1) the cover image is divided into clusters, 2) the watermark image is imbedded into the cover image using clusters, 3) attack the watermarked image, 4) run clustering method again for result image, and 5) extract watermark from the attacked image using clusters.

```
Watermark_Algorithm ()
{
   clustering(cover_image);
   embed_watermark(cover_image, watermark, clusters_set);
   attacks(watermarked_image);
   clustering(attacked_image);
   extract_watermark(attacked_image, clusters_set);
}
```

Fig. 3: Watermark Scheme Algorithm

The fourth bit is used as parity bit (for error detection). A sub-cluster counter is maintained to distinguish between the embedded data and to maintain their sequencing. This counter is written in three of the fourth LSBs of one of RGB values (say B). A parity bit is also maintained in the fourth bit for error detection. A parity bit is also calculated for each of the four MSB of RGB values and stored in three of the four LSB bits of one of RGB values (say G) which in their turn are protected by a parity bit.

In this paper, we focused on geometric attacks. The used scheme may not be robust against noise which directly affects the LSBs. This can be managed by reconfiguring the utilized LSBs depending on the cluster, the sub-cluster and the number of embedded bits. This will be experimented in a future work.

C. *Extracting Process*

Fig. 5 describes the extracting process algorithm. Before executing the extract algorithm a new clustering process should take place. Again the image should be scanned using the clustering indexing. The watermark should be found in sequence from the first cluster until the last cluster. If the cluster is not empty, all pixels in this cluster are checked, parity bits are verified and a decision table is built.

```
watermark = extract_watermark(image, clusters_set)
{
 For each cluster in clusters_set
  {
   For each pixel in cluster
     {
      Error=ParityCheck();
      If (no Error)
       {
        Seq=extractSeq();
        Wat=extractData();
        decisionTable(Seq,Wat)=decisionTable(Seq,Wat)+1;
       }
     }
  Build watermark portion from decisionTable. The index of the enter with maximum
value in a row forms the data to extract from the sub-scluster.
  }
}
```

Fig. 5: Extract Process

V. Experimental Results and Analysis

We used 512×512×3 color images as a host carrier signal, and 64×64 binary image as the watermark signal, as shown in Fig.6 and Fig.7. The correlation coefficient (NC) in equation (1) is used for measuring the quantitative similarity between the extracted and embedded watermarking:

$$\mathrm{NC} = \frac{\sum_x \sum_y i(x,y) w(x,y)}{\sum_x \sum_y i^2(x,y)} \qquad (1)$$

Where *i* denotes the embedded original watermark and *w* denotes the extracted watermark.

The difference between watermarked image and original host image is evaluated using the Peak Signal to Noise Ratio (PSNR). The PSNR formula is giving in equation (2):

$$\mathrm{PSNR} = 10 \log \left(\frac{\max\left(i^2(x,y)\right)}{\frac{1}{M\mathrm{x}N} \sum_{y=1}^{M} \sum_{x=1}^{N} \left(i(x,y) - w(x,y)\right)^2} \right) \quad (2)$$

a-Peppers b-Mandrill c-Hollyhocks

Fig. 6: Host carrier signal images

Copyright image

Fig. 7: Watermark signal

The proposed solution does not affect the normal visualization of the cover image as it is proved by the PSNR presented in table 1.

Table 1: PSNR of cover image

	peppers	mandrill	Hollyhocks
Complex Wavelet Transform Domain	45.11	44.83	NA
Our solution	41.56	44.68	44.00

In previous research papers [14], [15] we compared our results with the results provided in [13] under the same environment. Table 2 shows that using our scheme, under any rotation attack, the watermark is completely extracted without any distortion (NC=1).

Table 2: NC of the watermark image

		Rotation 5°	Rotation 15°
Spatial domain	peppers	0.927	0.885
	mandrill	0.918	0.869
Wavelet domain	peppers	0.945	0.913
	mandrill	0.937	0.896
Complex Wavelet Transform Domain [13]	peppers	0.981	0.946
	mandrill	0.966	0.930
Our solution	peppers	1.000	1.000
	mandrill	1.000	1.000
	Hollyhocks	1.000	1.000

D. *Pixels Distribution over Clusters*

Table 3 presents the overall clusters changes between original and attacked images. The maximum changes rate shown in these measurements is less than 6%. The clusters distribution is shown in Fig. 8.

Table 3: Overall clustering changes rate (%)

	Rotation 15°	**Rotation 45°**	**Rotation 90°**
Peppers	**2.03**	**5.55**	**0**
Mandrill	**2.49**	**4.91**	**0**
Hollyhocks	**1.88**	**3.62**	**0**

Fig. 9 shows that global clusters changes rate for all rotation degrees from 1 until 359 are: 5.63% for peppers, 5.21% for Mandrill, and 3.62% for Hollyhocks. Note that for angles 90°, 180°, 270° the changes rate is always 0. This is because there is no interpolation in these rotation degrees. All this proves that our proposal clustering method is nearly invariant clustering which solidifies the robustness.

E. *Cluster Dispersal Rate*

We define the dispersal rate (DR) of a set of pixels in the image as the closest area containing these pixels into the image over the total area. Even though, we can get better dispersal rate representativeness taking the smallest polygon and by including the number of selected pixels but for simplicity we considered only the smallest rectangle holding the set of pixels armed by the fact that condense area is liable to attacks. We believe that the higher cluster dispersal rate we can get the more robust proposed watermark scheme we can build. We define the dispersal rate in two levels:

1- The Cluster Dispersal Rate (CDR): at the cluster level which depends on the image itself and the clustering function. This will be considered in a future work. Cluster Distribution Rate is calculated according to the Equation 3.
2- The Sub-cluster Dispersal Rate (SDR): at the level of sub-cluster within the cluster which depends of the sub-cluster construction. This is research work will focus on this rate impact. Equation 4 gives the average SRD of sub-clusters in a given cluster.

$$\mathrm{CDR}_c = \frac{(x_{max\ c} - x_{min\ c}) * (y_{max\ c} - y_{min\ c}) * 100}{\mathrm{MxN}}, \qquad (3)$$

$$\text{where, M and N image sizes}$$

$$\mathrm{avg}(\mathrm{SDR}_c) = \frac{(x_{max\ sc} - x_{min\ sc}) * (y_{maxs\ c} - y_{min\ sc}) * 100}{nsc_c * (x_{max\ c} - x_{min\ c}) * (y_{max\ c} - y_{min\ c})}, \qquad (4)$$

$$\text{where, } \mathrm{nsc}_c \text{ is number of sub\_clusters in C}$$

We used two ways to build sub-clusters:

1- The first is the easy one where the cluster is divided into sub-clusters in sequence (first sub-cluster is built with the first found pixels in the cluster and so on).
2- The second way consists of building sub-clusters with the highest SDR. The cluster area is divided into geographic regions and sub-clusters are built in such a way pixels are added from each region.

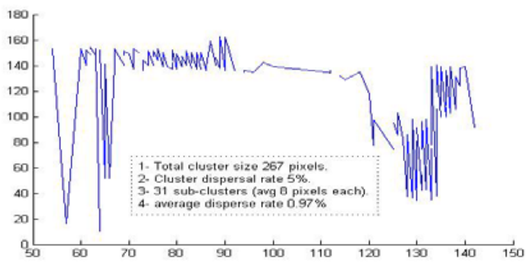


Fig. 10: In-sequence sub-clusters allocation

Fig. 10 and Fig. 11 give, respectively, the in-sequence and highest-SDR sub-clusters allocation for the cluster 435 of Hollyhocks. This cluster contains 267 pixels divided into 31 sub-clusters and has a CDR of 5%. The average SDR in the first way of sub-clustering is 0.97% whereas in the second is 49%.

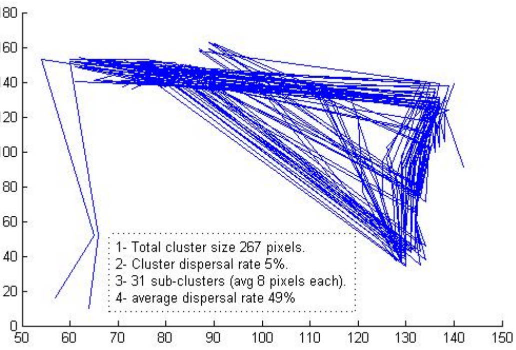


Fig. 11: Highest SDR sub-clusters allocation

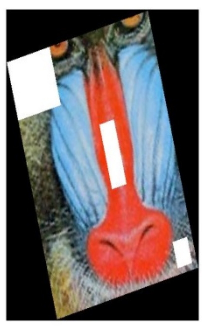
Table 4 presents the average sub-clusters dispersal rate (SDR) for each image in both clustering ways. We can see that the proposed second way over-performs the first in terms of surface occupation. If each sub-cluster can hold 3 bits, like in our experiments, the minimum number of sub-clusters given in peppers image can hold more than twice the embedded watermark.

Table 4: Average sub-clusters dispersal rate (SDR)

	# of clusters	# of sub-clusters	First way	Second way
peppers	344	3386	26.66	48.92
mandrill	371	5318	23.25	42.06
Hollyhocks	463	10309	12.44	32.25

The performance of the second way of sub-clustering over the first is observed with the cropping and mixed attacks as shown in Fig. 12. For all rotation degrees the performance is the same as the watermark can be extracted with NC equal 1 in both sub-clustering ways.

Hollyhocks: crop 50-100-205-415 (**NC_1=0.96, NC_2=1**)



Mandrill: crop 70-60-300-400 with touches and rotation 15^o (**NC_1=0.97, NC_2=0.99**)

Peppers: -crop 50-50-430-480 with touches (**NC_1=0.91, NC_2=0.92**)   Mandrill: crop 70-60-300-400 with touches (**NC_1=0.97, NC_2=0.99**)

Fig. 12: Cropping and mixed attacks results with improved DSR

The obtained results can be improved by increasing the minimum size of sub-clusters and further by considering the increase the average of CDR.

Fig. 13: Hollyhocks mosaic (NC=1)

Fig. 14: Mandrill mosaic (NC=1)

Fig. 15: Peppers mosaic (NC=1)

As can be seen in Fig. 13-15 the watermark technique is totally robust against mosaic attack. The mosaic algorithm is shown in Fig. 16.

```
function [puzzle ] = mosaic(cover, plength , pwidth )

s=size(cover);

pieces = (s(1)/ plength)*(s(2)/ pwidth);
pIndexI=zeros(1,peices);
pIndexJ=zeros(1,peices);

n=1;
for i=1:plength:s(1)
   for j=1:pwidth:s(2)
     pIndexI(1,n)=i;
     pIndexJ(1,n)=j;
     n=n+1;
   end
end

m=randperm(pieces);  //random distribution
n=1;
for i=1:plength:s(1)
   for j=1:pwidth:s(2);
       mI=pIndexI(1,m(n));
       mJ=pIndexJ(1,m(n));
       k(mI:mI+l-1,mJ:mJ+c-1,:)= cover(i:i+l-1,j:j+c-1,:);
       n=n+1;
   end
end
end
```

Fig. 16: mosaic algorithm

Moreover, unlike most of the other researches where tests are generally ran using a single attack, our proposed watermarking scheme prove the robustness even using combined attacks. Fig. 17 and Fig. 18 show such combination.

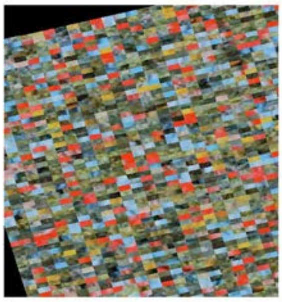


Fig. 17: Mandrill mosaic 8-16 rot 15 crop 75-65-360-460 (NC=0.82)

The mosaic is formed by cutting the image on pieces of puzzle, with equal size for simplicity, and then randomly distributed. The random distribution makes that the performance may drop with cropping attack as some entire sub-clusters may be totally removed. This can be clearly seen in Fig. 17 where the image was cut on pieces of 8x16 pixels, rotated by 15° and cropped to a window of [75,65,360,460]. The NC is 0.82.

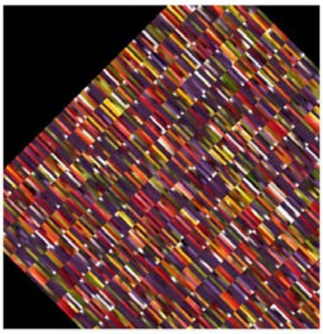


Fig. 18: Peppers mosaic 32-4 rot 45 crop 60-65-460-470 (NC=0.99)

In Fig. 18 the NC is equal to 0.99 with a mosaic formed by 32x4 pieces and ten rotated to 45° and then cropped to a window [60,65,460,470].

VI. Conclusion

In this paper we proposed a new image clustering technique based on the Content Addressable Method. This clustering technique aims to be used for watermarking in color images. The cover image is divided into a set of clusters which are built using Content Addressable Method. Each cluster is divided into sub-clusters. Each sub-cluster is used to hold the same watermark portion. The robustness of our scheme comes from the fact that it resists on image rotation, mosaic, and cropping attacks. Rotation and mosaic attacks do not affect the robustness of our scheme when only one single attack is applied at a time. We used two sub-clusters allocation techniques. Our experiments show that the more the SDR is greater the best results obtained. On the other hand, we have promised results by combining several attacks together.

Future work will be concentrated on looking for more improvement of the average SDR as well as the average of CDR which may have a considerable better impact on the watermark extraction results. Moreover, we will explore other directions to use the same technique against other kind of attacks like noise which directly affects the LSBs and Jpeg compression.

Acknowledgment

The Author gratefully acknowledges financial support for the present work (project ID 2701) from the Deanship of Scientific Research at Qassim University.

References

[1] Sasank Reddy, Andrew Parker, Josh Hyman, Jeff Burke, Deborah Estrin, Mark Hansen, "Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype", EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors, June 2007.

[2] Yongqing Sun, Shinji Ozawa, "Semantic-meaningful Content-based Image Retrieval in Wavelet Domain", MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, November 2003.

[3] Asmatullah Chaudhry, Mehdi Hassan, Asifullah Khan, Jin Young Kim, Tran Anh Tuan, "Image clustering using improved spatial fuzzy C-means", ICUIMC '12: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, February 2012.

[4] Yu Zhao, Zhiqiang Gao, Baoxiu Mi, "Fast image segmentation based on K-means algorithm", ICIMCS '12: Proceedings of the 4th International Conference on Internet Multimedia Computing and Service, September 2012.

[5] Ahmed, N. ; Jalil, A., "Image Clustering Using Discriminant Image Features", 11th International Conference on Frontiers of Information Technology (FIT), 2013, DOI: 10.1109/FIT.2013.13
Publication Year: 2013 , Page(s): 31 – 36.

[6] Zhen Wang ; Meng Yang, "A fast clustering algorithm in image segmentation", 2nd International Conference on Computer Engineering and Technology (ICCET), 2010 Volume: 6, DOI: 10.1109/ICCET.2010.5486041, Publication Year: 2010 , Page(s): V6-592 - V6-594 .

[7] Maheshwari, M. ; Silakari, S. ; Motwani, M., "Image Clustering Using Color and Texture", First International Conference on Computational Intelligence, Communication Systems and Networks, 2009. CICSYN'09. DOI: 10.1109/CICSYN.2009.69, Publication Year: 2009 , Page(s): 403 - 408 .

[8] Xu Wangming ; Liu Xinhai ; Fang Kangling, "Content-based image clustering via multi-view visual vocabularies", 31st Chinese Control Conference (CCC), 2012 Publication Year: 2012 , Page(s): 3974 – 3977.

[9] Shukui Bo, Yongju Jing, "Image Clustering Using Mean Shift Algorithm",
Fourth International Conference on Computational Intelligence and Communication Networks (CICN), 2012, DOI: 10.1109/CICN.2012.128, Publication Year: 2012 , Page(s): 327 – 330.

[10] Singh, D. ; Singh, A. ," A New Framework for Texture Based Image Content with Comparative Analysis of Clustering Techniques", Fourth International Conference on Computational Intelligence and Communication Networks (CICN), 2012, DOI: 10.1109/CICN.2012.22, Publication Year: 2012 , Page(s): 232 – 236.

[11] L. An, X. Gao, X. Li, D. Tao, C. Deng, J. Li, Robust Reversible Watermarking via Clustering and Enhanced Pixel-Wise Masking, IEEE Transactions on Image Processing, Vol.21 , Issue: 8, DOI: 10.1109/TIP.2012.2191564, Publication Year: 2012, pp. 3598-3611.

[12] Y. Haowen , Watermarking Algorithm for Vector Point Clusters, 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2011, DOI: 10.1109/wicom.2011.6040148, Publication Year: 2011, pp. 1–4.

[13] J. Fan, Y. Wu, Watermarking Algorithm Based on Kernel Fuzzy Clustering and Singular Value Decomposition in the Complex Wavelet Transform Domain, International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), 2011, Vol.3, DOI: 10.1109/ICM.2011.121, Publication Year: 2011, pp. 42–46.

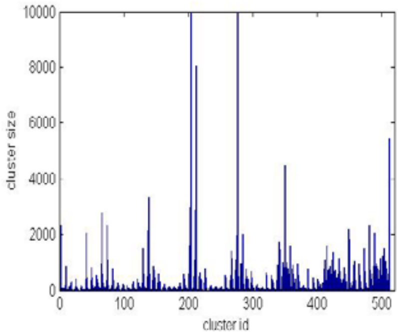
[14] M.T. Ben Othman, New Image Watermarking Scheme based on Image Content Addressing Method, accepted in 13th WSEAS International Conference on Applied Computer and Applied Computational Science ACACOS'14, Kuala Lumpur, Malaysia, April 23-25, 2014.

[15] M.T. Ben Othman, Digital Image Watermarking based on Clustering, 3rd International Conference on Circuits, Systems, Communications, Computers and Applications (CSCCA '14), Florence, Italy; 11/2014

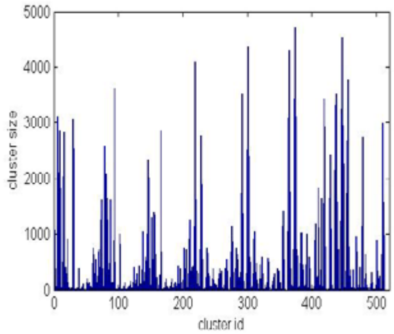
[16] H.V. Dang, W. Kinsner, Y. Wang, Multiobjective Image Data Hiding Based on Neural Networks and Memetic Optimization, WSEAS Transactions on Signal Processing, Vol. 10, 2014.

[17] Jia Mo, Zhaofeng Ma, Yixian Yang, Xinxin Niu, Shuhua Xu, Application of ICA and contourlet transform in image watermarking, WSEAS Transactions on Mathematics, Issue 4, Volume 12, April 2013.

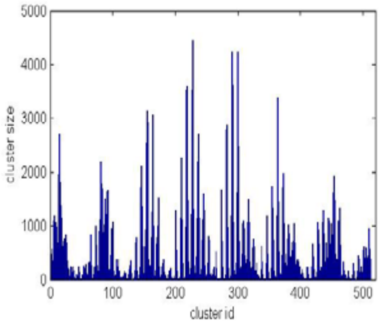
Mohamed Tahar Ben Othman received his Ph.D. in Computer Science from The National Institute Polytechnic of Grenoble INPG France in 1993, His Master degree in Computer Science from ENSIMAG "École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble" in 1989. He received a degree of Senior Engineer Diploma in Computer Science from Faculty of Science of Tunis. This author became a Member (M) of IEEE in 1997, and a Senior Member (SM) in 2007. He worked as post-doc researcher in LGI (Laboratoire de Genie Logiciel) in Grenoble, France between 1993 and 1995, Dean of the Faculty of Science and Engineering between 1995 and 1997 at the University of Science and Technology in Sanaa, Yemen, as Senor Software Engineer in Nortel Networks, Canada, between 1998 and 2001 and Assistant and Associate (since 2011) Professor in Computer College at Qassim University in Saudi Arabia from 2002 until present. His research interest areas are information security, wireless networks, Adhoc Networks, communication protocols, and bioinformatics.



A.1: peppers-cluster distribution

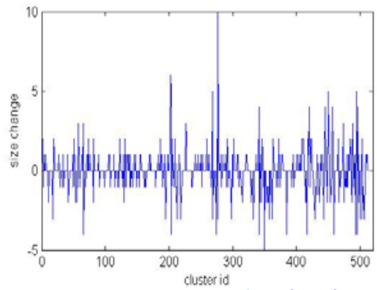


A.2 :Mandrill -cluster distribution

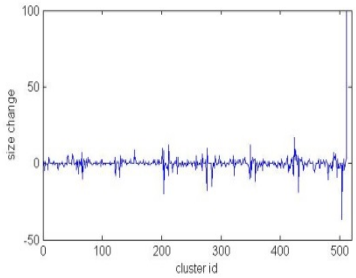


A.3 :Hollyhocks -cluster distribution

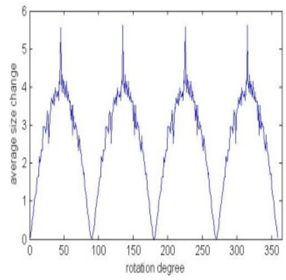
Fig. 8: Cluster distribution



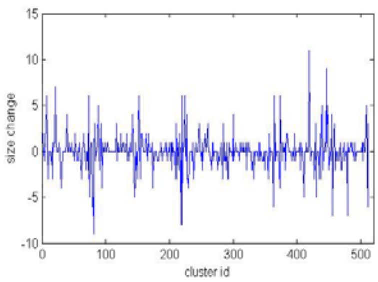
B.1: peppers-changes from 0º to 15º



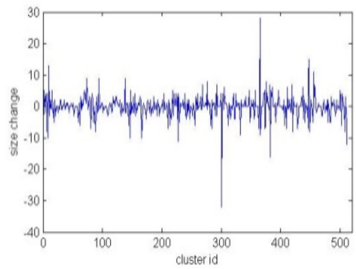
B.2 :peppers-changes from 0º to 45º



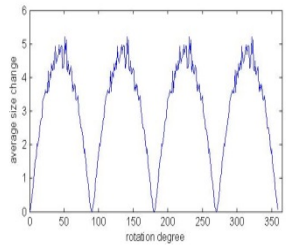
B.3 :peppers-overall changes from 0º to 360º



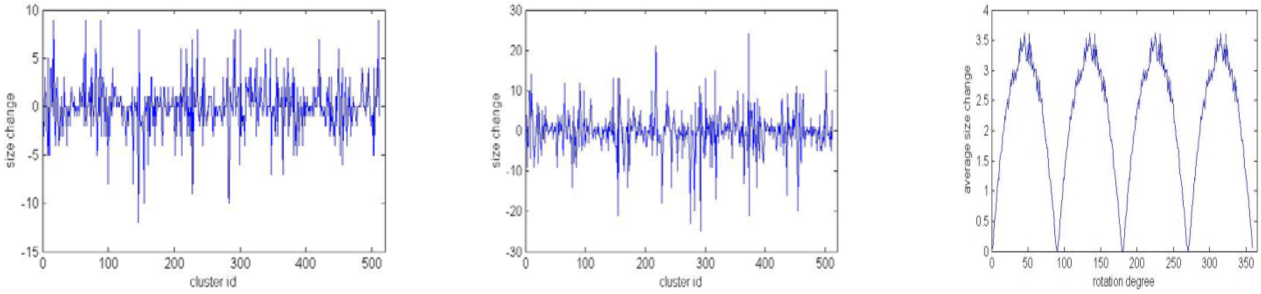
C.1 :Mandrill-changes from 0º to 15º



C.2 :Mandrill-changes from 0º to 45º



C.3 :Mandrill -overall changes from 0º to 360º



D.1 :Hollyhocks -changes from 0^o to 15^o D.2 :Hollyhocks -changes from 0^o to 45^o D.3 :Hollyhocks -overall changes from 0^o to 360^o

Fig. 9: Clusters changes