Performance Evaluation on Feature Modeling of Halftone Image Texture

Zhiqiang Wen*, Zhiyan Zhong, Degang Ye and Wenqiu Zhu

School of Computer and Communication, Hunan University of Technology, Zhuzhou, 412007, China

Abstract: Three statistics methods, gray-level co-occurrence matrix, autocorrelation function and spectrum statistics, were used to extract feature vector of various halftone images for halftone image classification. The classification performances of three kinds of feature vectors were assessed by three classifiers: radial basis function neural network, least mean square and principal component analysis. Experimental results showed the autocorrelation function is better than other two methods for classification of halftone image. It indicated the best classification performance when the parameter K=64 and L=8.

Keywords: halftone image, inverse halftoning, modeling, statistical method, texture feature.

1. INTRODUCTION

With the development of digital technology, increasing numbers of researchers focused on research of various inverse halftoning. There are two types of inverse halftoning, named universal inverse halftoning, such as neural network [1], color inverse halftoning [2], edge-based LUT [3], lookup table based on image local statistical properties [4], and special inverse halftoning, such as low pass filter [5], maximum a posteriori estimation [6] and projection onto convex sets [7]. The former represent the universal inverse process of all sorts of digital halftoning mode. But these methods will lead to difficulty on optimal image reconstruction due to the lack of halftone image information. The latter describes the inverse halftoning for specific halftone image. But in practice, it is difficult to obtain the optimal reconstruction for unknown type of halftone images. In order to solve the above problem, research on classification of halftone image is necessary. It can improve the quality of the image reconstruction by choosing an inverse halftoning technology to reconstruct image after classifying halftone image. This paper focuses on textural feature modeling of various halftone images.

Texture is an important visual cue. It widely exists in various images, such as natural scenery images and remote sensing image. Of course, it also exists in halftone image. Due to usage of the error-diffusion kernel and dither matrix in digital halftoning, various artificial textures always exist in halftone images. The ordered dithering with the screen matrix will result in the periodic artifact wormlike texture and most of the image details, which exist in the contone images, are lost due to halftoning process. The dispersed ordered dithering will produce the unwanted artificial texture [8]. The so-called "dot gain" will come in clustered ordered dithering images due to non-ideal behavior of printers. The error diffusion will cause the irregular or non-periodic texture of halftone images, such as wormlike pattern, checkerboard pattern and diagonal strip pattern. One of the most observable artifacts of error diffusion images is the periodically repeating patterns appearing in the areas where the input is constant or slowly varying [9]. Fortunately error diffusion gives better halftone quality for printers which do not suffer from dot gain. Basically, the power spectrum of the halftone error will be concentrated in high frequencies, and this type of error is called blue noise [10]. There also are artificial periodic patterns in halftone images produced by dot-diffusion method [8].

How to describe these textural features? Textural feature representation has received considerable attention during the past decades and many methods have been presented in many literatures. Liu L, *et al.* summarized all texture representation methods before 2009 and divided them into four categories as: statistic, model, signal processing and structural method [11]. Among these methods, the statistical method was applied widely due to its simplicity and easiness of implementation. In statistical method, the statistical properties, such as pixel and its first order, second order, and high order statistical properties in its neighborhood, were thoroughly researched. There are many statistical methods: autocorrelation function (AF), gray level co-occurrence matrix (GLCM) [12], spectrum statistic (SS), local binary pattern (LBP) [13], gray level run-length matrix (GLRM) [14], gray level difference matrix (GLDM), and cross diagonal matrix (CDM).

It is proved that GLCM has become a beacon in all statistical methods. Due to its simplicity and low computation complexity, AF was used widely. SS was further studied because of its robustness. Researchers paid more attention to LBP for its rotation invariance and its multi-scale features, however, our experiment results show that LBP gives bad result for halftone image texture representation. Due to bad discriminative ability of texture features and larger computation time, the application of GLRM, GLDM and CDM

1874-4443/14 2014 Bentham Open

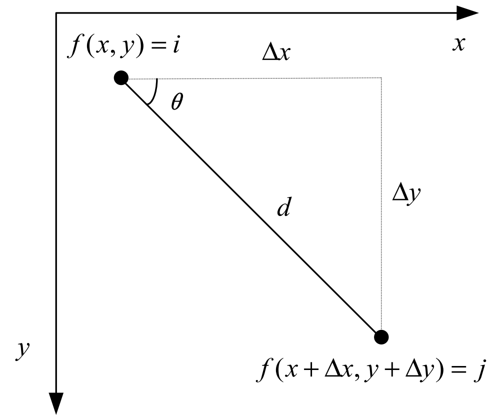


Fig. (1). Sketch map of GLCM.

are limited and there is little researches concerning these methods [11]. For halftone image classification, up to now, there are only several texture modeling methods, e.g., AF [15], GLCM [16, 17], LMS based on Fourier spectrum [18, 19]. These three methods were well for halftone image classification. They were true? So the purpose of this paper is to assess three modeling methods: GLCM, SS and AF from two aspects: feature representation and classification performance by experiments on common datasets.

2. FEATURE MODELING

2.1. GLCM

Based on estimation of second order conditional probability density of image, GLCM described the probability that a pair of pixels with gray value *i* and *j* appear in direction θ and distance *d*. GLCM, a symmetric matrices, is the function of distance *d* and direction θ and its order is decided by gray level of an image. 14 kinds of texture feature can be drawn from GLCM [12]. Element value $n(i, j, d, \theta)$ in GLCM means the statistics frequency numbers of a pair of pixels in which a pixel $f(x, y)$ with gray level *i* is apart from another pixel $f(x+\Delta x, y+\Delta y)$ with gray level *j* about distance *d* in direction θ as shown in Fig. (**1**). For the simplicity, $n(i, j, d, \theta)$ is replaced by probability *p* as follows.

$$p(i,j,d,\theta)=\{p[(x,y),(x+\Delta x,y+\Delta y)]$$

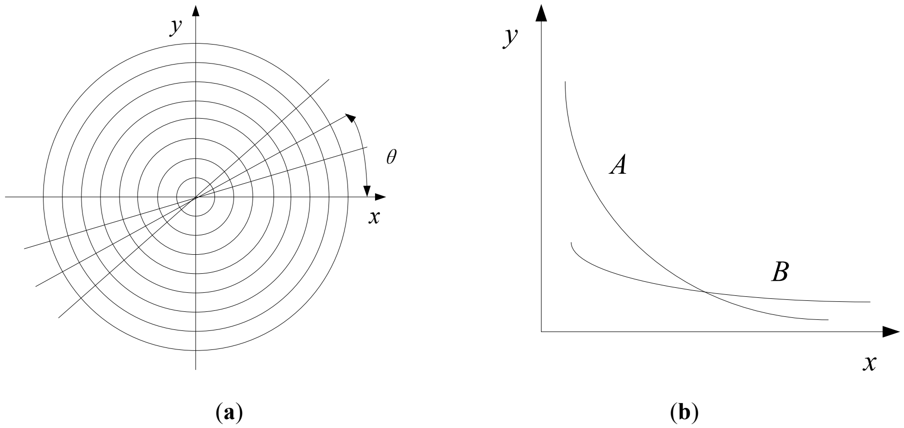
$$|f(x,y)=i,f((x+\Delta x,y+\Delta y)=j)\}$$

satisfying $\tan\theta=\Delta y/\Delta x$ and $d^2=\Delta x^2+\Delta y^2$, where $x=1,\ldots,M$, $y=1,\ldots,N$ and $i, j=0,1,\ldots,G\text{-}1$. (x, y) is pixel position. *M* and *N* denote the width and the height of images respectively. *G* is grey level of image ($G=2$ for halftone images). Usually θ is set as 0°, 45°, 90° and 135° and *d* is set as 1, 2, 3, 4, and 8. For large amount of calculation on 14 kinds of statistics features, 5 kinds of GLCM features (angular second moment, Contrast, Correlation, inverse difference moment and entropy) are commonly recommended for improving computational efficiency [20]. In this paper, the image is divided into sub-block with same size $K\times K$ and θ is set as the above mentioned four angles in each sub-block where $d<K$. 5 kinds of GLCM features are used in each direction to form GLCM feature vector of sub-block, namely there are 20 elements in a GLCM vector. We regard the mean of GLCM vectors in all sub-blocks of halftone images as the GLCM features.

2.2. SS

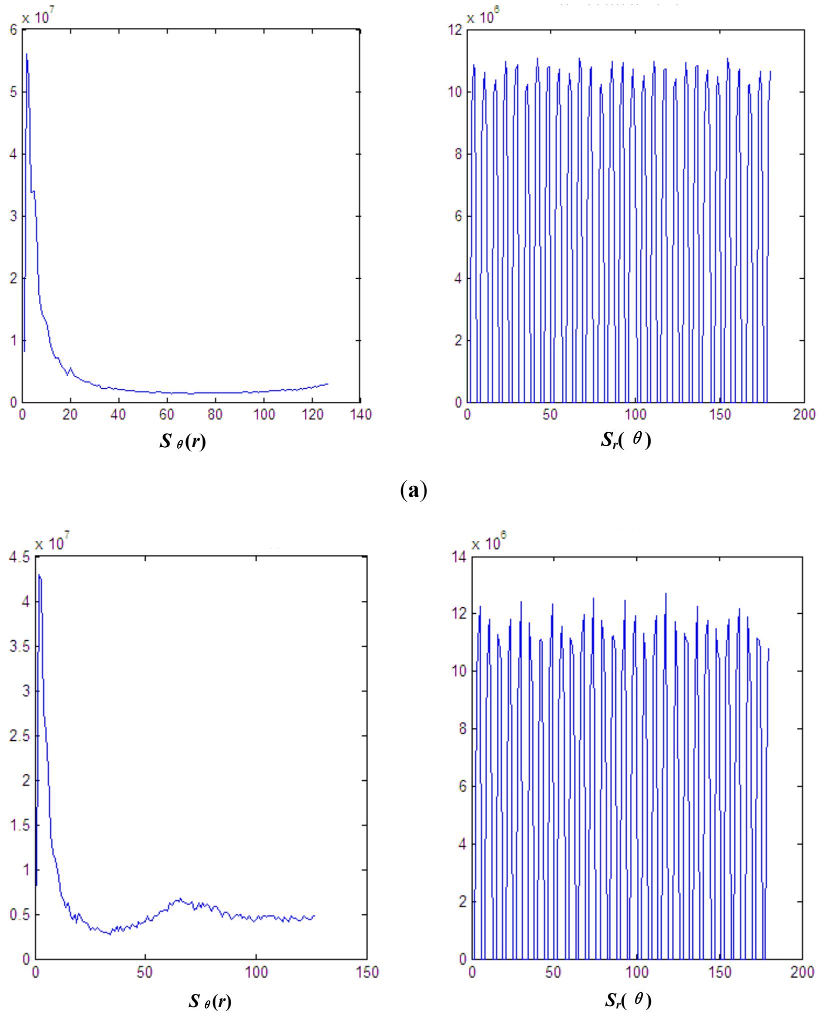
Image spectrum means the periodical or similar periodical texture structure of 2D image described by frequency characteristics of Fourier spectrum. Protuberant peak values in Fourier spectrum represent the principal direction of texture pattern. The position of peak value in plane of spectrum domain describe fundamental period of texture pattern. If the periodic components were removed by a filter, the rest of non-periodic components can be easily described by the spectrum statistic method. In fact, the spectrum map can be transferred into a polar coordinate diagram, as shown in Fig. (**2a**). Fourier transform is signified by $F(r, \theta)$ and the spectrum after Fourier transform is denoted by $S(r, \theta)$, then $S(r, \theta)=|F(r, \theta)|^2$. Given a fixed direction θ, $S(r)$ is a one dimensional function about *r*. In the same way, given a fixed frequency *r*, $S(\theta)$ is a one dimensional function about θ.

Given θ, we can acquire the behavior characteristics of spectrum along the direction from initial point via $S(r)$. Similarly, we can get the behavior characteristics of spectrum along a circle via $S(\theta)$. For discrete image, we can acquire the global description of texture along a direction θ or a



(**a**) (**b**)

Fig. (2). Texture and its spectrum statistic, (**a**) polar coordinate diagram of spectrum and (**b**) distribution of spectrum energy.



(**a**)

(**b**)

Fig. (3). Spectrum energy of two halftone images, (**a**) a halftone image produced by Bay, (**b**) a halftone image produced by Sten.

circle with radius *r* by summing the spectrum in same *r* or *θ*. That is to say, we can get the spectrum features. The formulas for computing spectrum features of images are shown as follows.

$$\begin{cases} S(r)=\sum_{\theta}\left|F(r,\theta)\right|^2=\sum_{\theta}S(r,\theta) \\ S(\theta)=\sum_{r}\left|F(r,\theta)\right|^2=\sum_{r}S(r,\theta) \end{cases}$$

The spectrum energy on the whole image consist of *S*(*r*) and *S*(*θ*). Small *r* result in large *S*(*r*), or it brings about small *S*(*r*). Large *S*(*r*) shows the rough texture structure of image. Conversely, it shows the fine texture structure. The spectrum energy focuses on the value close to the origin point in rough texture, such as A curve shown in Fig. (**2b**). Reversely, the spectrum energy will focuses on the value far from the origin for the fine texture such as B curve in Fig. (**2b**). Fig. (**3**) shows the spectrum energy of two halftone image. From these examples, we can conclude that the different type of halftone image patterns have different curve *S*(*r*) but the curve *S*(*θ*) seem to be same. From above discussion, *S*(*r*) is used for feature representation in this paper. Our method is: after the image is divided into sub-blocks with the same size *K*×*K*, spectrum *S*(*r*) is obtained from each sub-block (*r*≤*K*/2-1). Lastly, the mean of spectrum vectors in all sub-block of images is regarded as the spectrum features. In this method, *K*/2-1 spectrum values form a vector.

2.3. AF

Given a gray image *f*, let *f*(*x*, *y*) denote the pixel value in position (*x*,*y*). An autocorrelation function [15] for extracting texture feature is shown in (1).

$$r(l)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}H(a+x,b+y)\cdot H(a+x,b+j+y) \qquad (1)$$

where *H*(*a*, *b*) denotes a sub-block of halftone images. *a*=0, 1, …, *M*-1 and *b*=0, 1, …, *N*-1. *l*=1, 2,…, *L* and generally let *L* = 16. In [15], sub-block size is set as 1×*L*. The computation time complexity of (1) $O(M^2L^2)$ is still larger. To solve this problem, this paper gives a fast modeling method of poly-directional autocorrelation function. In our method, the correlation between not two sub-blocks but two pixels is regarded as the object of research. In addition, XNOR operation in logical mathematics is used to compute the correla-

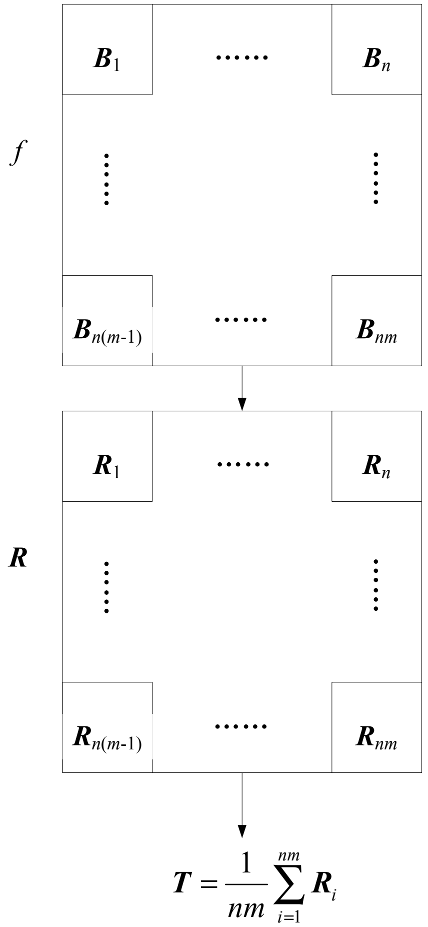


Fig. (4). Schematic diagram of AF feature extraction.

tion value between two pixels for special case of halftone images that there only value can be (0 or 1). Details about our method are shown as follows.

0 degree direction:

$$r_1(l) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\circ f(x+l,y) \tag{2}$$

45 degree direction:

$$r_2(l) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\circ f(x+l,y+l) \tag{3}$$

90 degree direction:

$$r_3(l) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\circ f(x,y+l) \tag{4}$$

Where sign "$\circ$" denotes the XNOR operation $\odot$ and l is an integer satisfying $1 \le l < L$ (L is a constant). From formula (2), (3) and (4), we can get three vectors denoted by $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$, respectively which form a feature vector $\boldsymbol{R}$, namely $\boldsymbol{R}=[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3]$. The element numbers in vector $\boldsymbol{R}$ are $3L$. After normalizing vector $\boldsymbol{R}$, we can get $3L$ features of autocorrelation function on a halftone image. Our method is: after the image is divided into sub-blocks with same size $K\times K$, $3L$ dimension features are obtained via above autocorrelation function from each sub-block to form the sub-block feature vector. Lastly, we regard the mean of these vectors in all sub-block of halftone image as the autocorrelation vectors denoted by $\boldsymbol{T}$. Fig. (**4**) shows the details of our method.

3. EXPERIMENT ENVIRONMENT

Digital halftoning algorithms can be divided into three categories: screening, error diffusion and search based methods [21]. Due to dependence on the screen matrix, screening includes many branch algorithms, such as dot diffusion [7]. Error diffusion compares each pixel with a fixed threshold and the binarization error is diffused ahead to a small set of neighbor pixels that have not yet been processed. Error diffusion also contains many branches as a result of using different distribution coefficients. Floyd-Steinberg filter, Stucki filter and Stenenson filter are the very popular error diffusion filters in various graphics applications. Search-based methods, such as direct binary search (DBS) [22], minimizes a metric of perceived error between the continuous tone image and the halftone image by directly searching for the best configuration of binary pixels in the halftone image. Even though DBS can create high quality halftone images, it is computationally expensive, and only acts as a benchmark [8]. With a low complexity algorithm and the fairly good visual quality of the produced binary images, screening and error diffusion has been used for many applications and are recommended in this paper. Based on above analysis, eleven halftoning methods are selected to produce halftone images for classification and their details are shown in Table **1**. Fig. (**5**) shows eleven halftone images produced respectively by these eleven halftoning methods. Some images provide almost visually same textures as shown in the sub-image (g), (h), (i) and (j) of Fig. (**5**), but the other halftone images provide different textures.

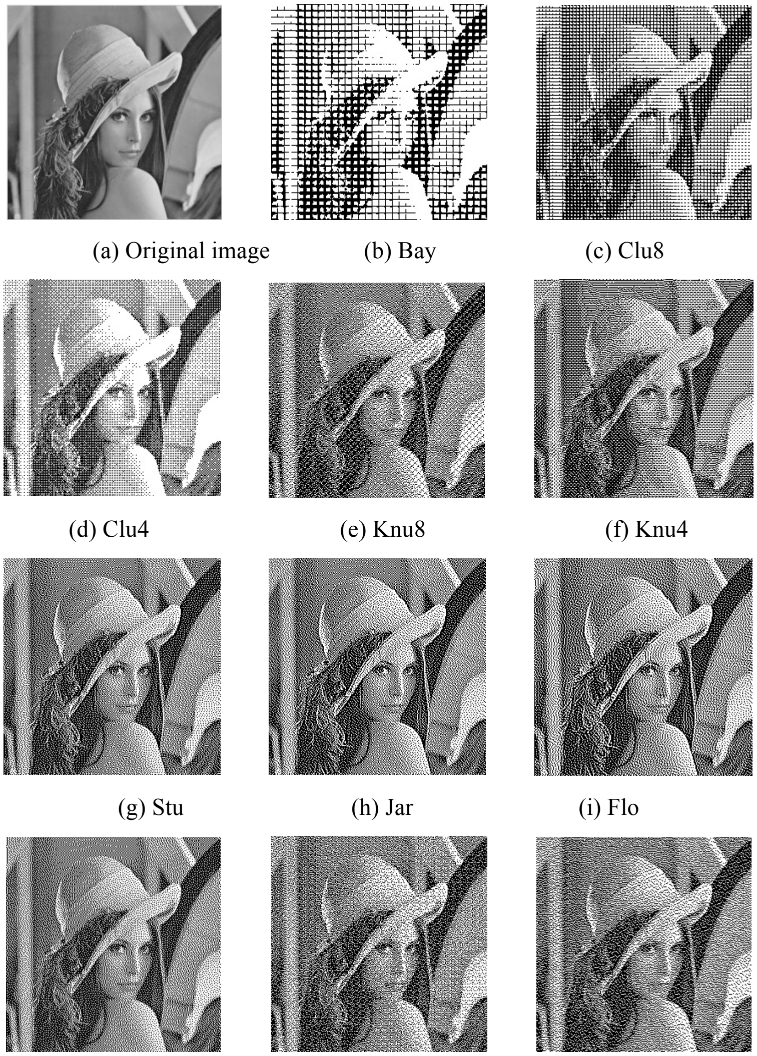
In our experiments, we use VC++6.0 and open CV as programming tool under the environment of windows XP and high performance computer with Intel chip, 2.2GHz CPU frequency and 3G internal memory. The image sets are from a public dataset (http://msp.ee.ntust.edu.tw/) including 1000 original gray image. Each of the original image was translated into eleven halftone images respectively by above eleven digital halftoning methods, so there are 11000 halftone image in which 5500 halftone images are regarded as training data sets and other halftone images as test data set for classification.

4. EVALUATION OF FEATURE REPRESENTATION

We extract the features from 40 halftone images of each type of halftone patterns produced by above three methods to plot the feature curves shown in Figs. (**6-8**) in this experiment. Fig. (**6**) shows the feature curves obtained by SS method where K=32. The dimension of these features is K/2-1, namely 15. Fig. (**7**) indicates the feature curves acquired by AF where K=32 and L=4. Each feature vector contains $3L$ elements. Fig. (**8**) denotes the feature curves obtained by GLCM where K=8 and d=5. Each of these feature curves has 20 feature values which are independent of the parameter K and d. From Fig. (**6-8**), we can conclude that:

Table 1. Digital halftoning methods and their details.

No.	Abb.	Algorithm Type	Representation
1	Bay	ordered dither	Bayer 8×8 dither model and 64 gray level [23]
2	Clu8	ordered dither	Cluster 8×8 dither model and 64 gray level [23]
3	Clu4	ordered dither	Cluster 4×4 dither model and 16 gray level [23]
4	Knu8	dot-diffusion	Knuth 8×8 dither model and 64 gray level [24]
5	Knu4	dot-diffusion	Knuth4×4 dither model and 16 gray level [24]
6	Stu	error diffusion	Stucki filter [25]
7	Jar	error diffusion	Jarvis filter [10]
8	Flo	error diffusion	Floyd-Steinberg filter [10]
9	Sten	error diffusion	Stenenson filter [10]
10	Mes8	ordered dither	Mese-Vaidyanathan 8×8 matrix [7]
11	Mes16	ordered dither	Mese-Vaidyanathan 16×16 matrix [7]



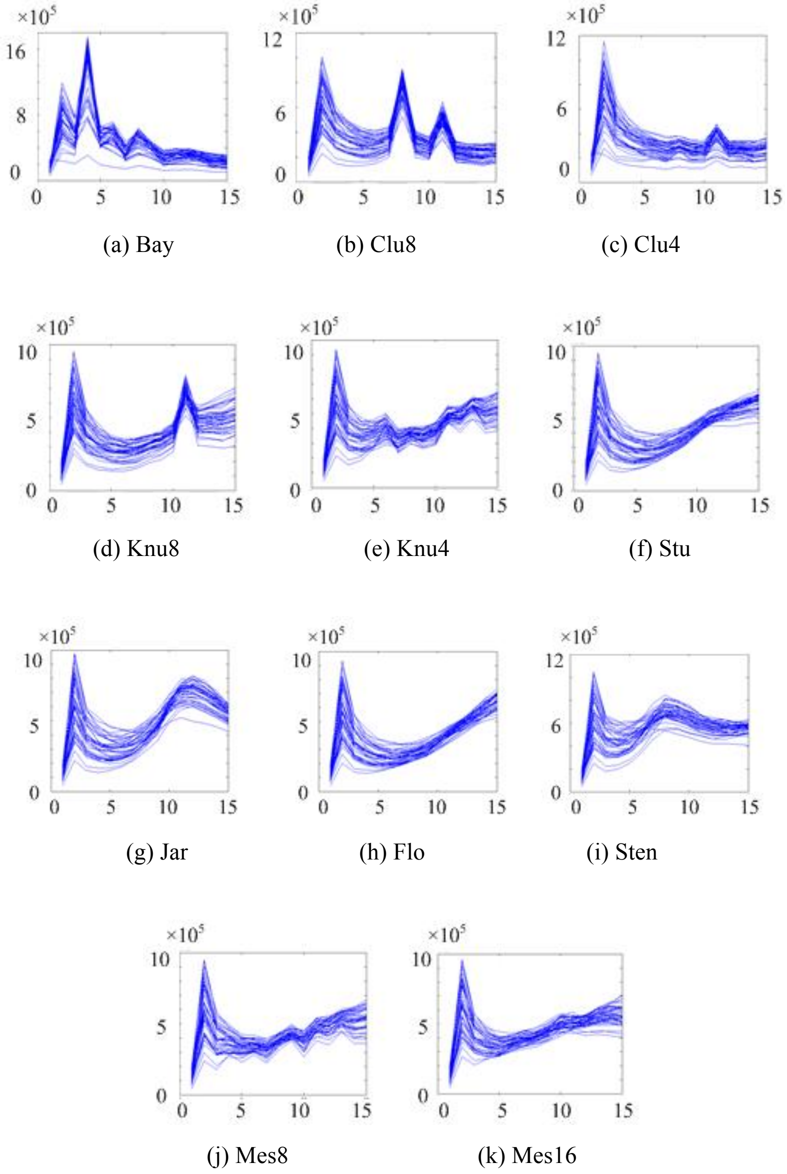
(a) Original image (b) Bay (c) Clu8

(d) Clu4 (e) Knu8 (f) Knu4

(g) Stu (h) Jar (i) Flo

(j) Sten (k) Mes8 (l) Mes16

Fig. (5). Original image and its halftone images.



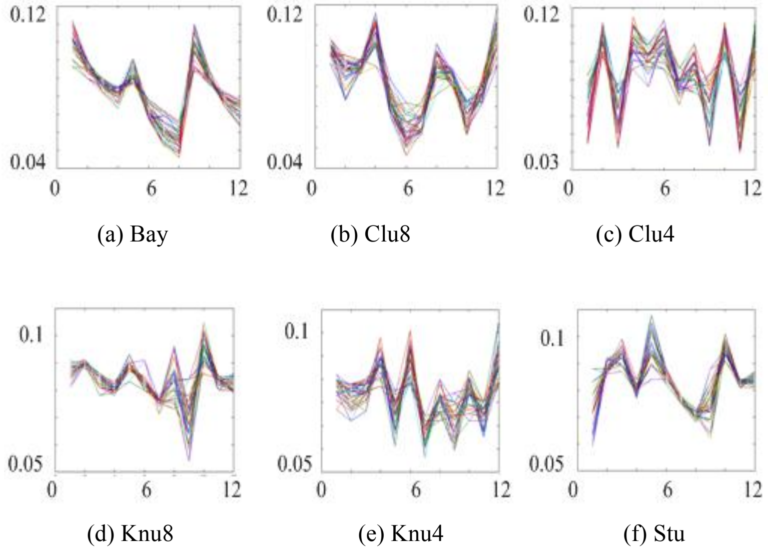
(a) Bay (b) Clu8 (c) Clu4

(d) Knu8 (e) Knu4 (f) Stu

(g) Jar (h) Flo (i) Sten

(j) Mes8 (k) Mes16

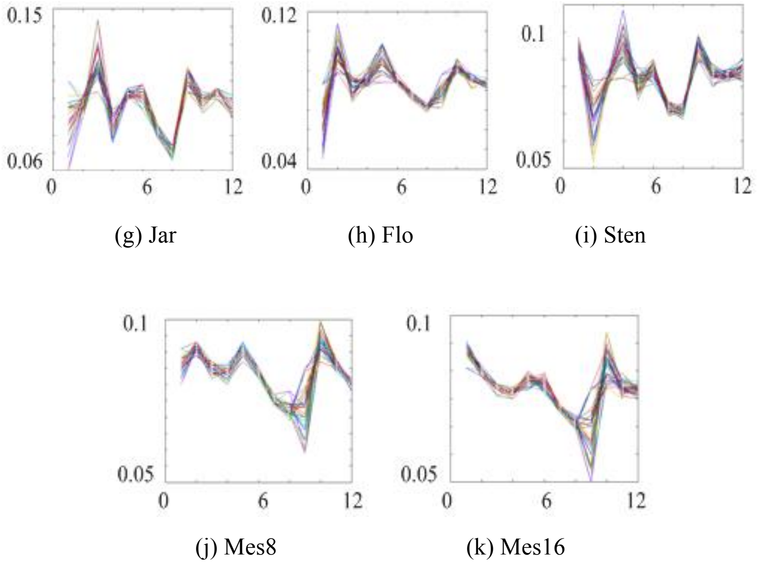
Fig. (6). Feature curves obtained by SS.



(a) Bay (b) Clu8 (c) Clu4

(d) Knu8 (e) Knu4 (f) Stu

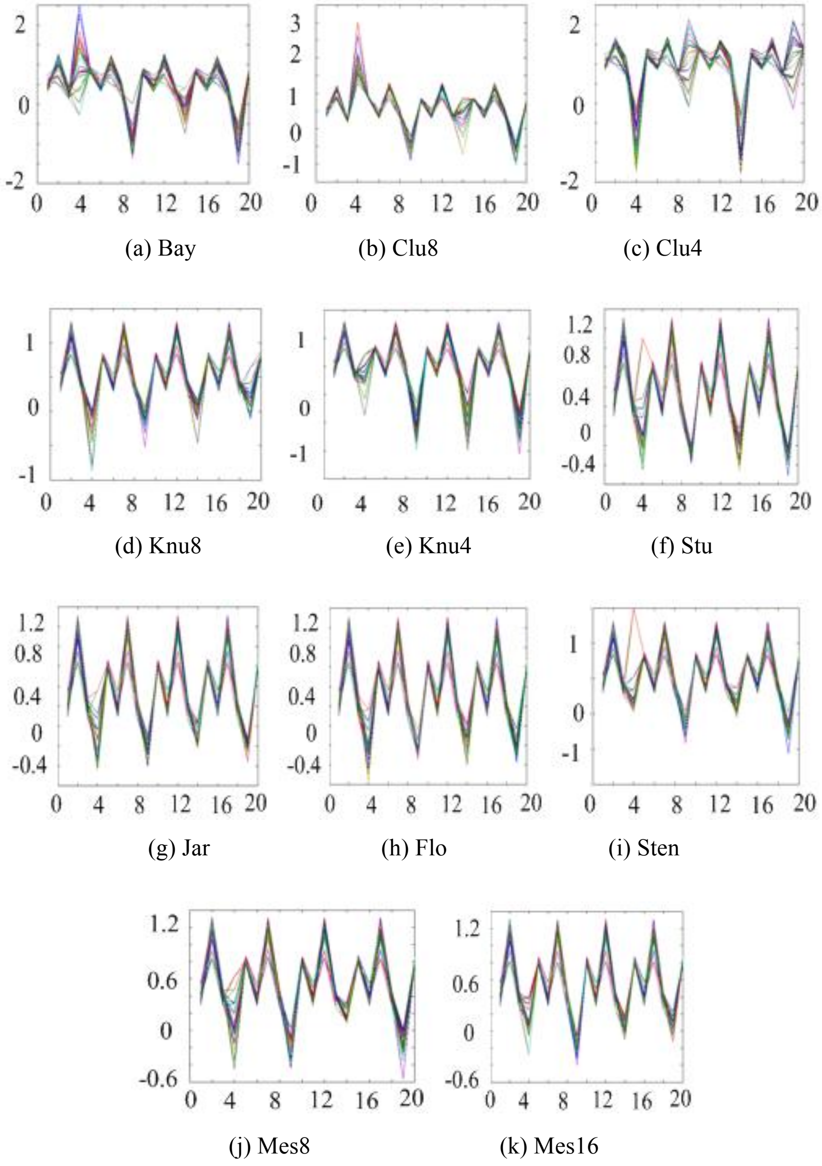
Fig. (**7**). contd…



(g) Jar (h) Flo (i) Sten

(j) Mes8 (k) Mes16

Fig. (7). Feature curves obtained by AF.



(a) Bay (b) Clu8 (c) Clu4

(d) Knu8 (e) Knu4 (f) Stu

(g) Jar (h) Flo (i) Sten

(j) Mes8 (k) Mes16

Fig. (8). Feature curves obtained by GLCM.

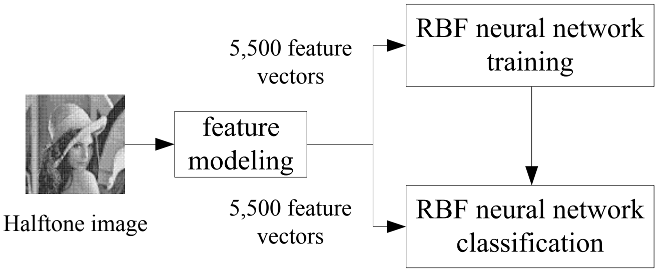


Fig. (9). The framework of RBF classification method.

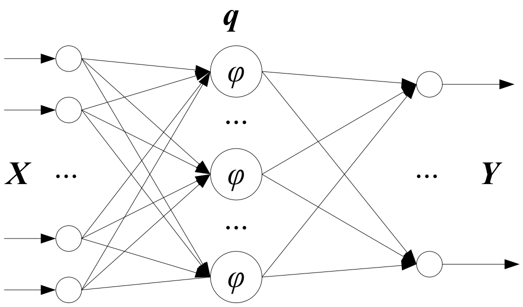


Fig. (10). The structure of RBF network.

1. The feature curves among sub-figures in Fig. (**7**) have the better discrimination ability than that in Fig. (**6**) although the numbers of features in Fig. (**7**) are less than that of Fig. (**6**).
2. Feature curves in Fig. (**7**) are more consistent than that in Fig. (**6**). Though Fig. (**8**) has more feature number than Fig. (**6**), or Fig. (**7**) does, the several sub-figures in Fig. (**8**) are almost the same and the discrimination ability of Fig. (**8**) is worse than that of Fig. (**6**) and Fig. (**7**). Thus it can be seen that the features in Fig. (**7**) are more adaptive for classification than that in Fig. (**6**) or Fig. (**8**).

5. EVALUATION OF CLASSIFICATION

In order to test the validity of above three modeling methods, we use three classifiers to classify all feature vectors obtained by above feature extraction methods respectively and regard the classification accuracy as performance evaluation of classification. For classification accuracy rate in multi-class problem, there are two parameters which can be denoted as average classification accuracy rate P_{avg} and classification accuracy deviation *dev*. In this paper these two parameters are used for the performance evaluation of texture classification.

5.1. Classification Evaluation Based on Neural Network

Back propagation (BP) and RBF are two most popular neural networks, but slow training speed of BP neural network is the limitation for application. Therefore, in this paper, we use RBF neural network as the classifier and the framework of our classification method is shown in Fig. (**9**). Fig. (**10**) shows the structure of RBF network. In Fig. (**10**), the output vector of network $\boldsymbol{Y}=\{y_1,y_2,\ldots,y_{11}\}$. Input vector $\boldsymbol{X}=\{x_1,\ldots,x_n\}$ and $\boldsymbol{q}=\{q_1,\ldots q_n\}$. n=K/2-1 in SS and n=20 in GLCM but n=3L in AF. φ is a radial basis function. y_i in $\boldsymbol{Y}$ describes the category of halftone images, for example, y_i=1 and $y_j=0\,\forall j$ ($1\leq j\leq 11$ and $j\neq i$) means ith class ($1\leq i\leq 11$). The features, which are extracted from 5500 halftone images by above three modeling methods, are used as training data sets and the rest are regarded as the test sample for classification. So we can acquire 5500 output vector $\boldsymbol{Y}$ and class c= arg max$_i\{ y_i,\ 1\leq i\leq 11\}$.

Classification performances of different feature vectors are shown in Table **2**. RBF classifier on feature vectors of AF reaches good performance: P_{avg}=91.04% and *dev*=1.35%. The classification performance of GLCM acquires the worst performance: P_{avg}=75.43% and *dev* =3.79%. When the feature vectors of SS are set as input, RBF classifier gets average classification accuracy rate: P_{avg}=87.2% and *dev*=5.14%. For training time spent *t*, there needs to be more time for training samples in feature vector of GLCM than that of AF or SS. From above, we can conclude that AF is superior to GLCM and SS for halftone image classification.

There are two parameters *K* and *L* in AF, so in our experiments, let *K* and *L* be power of 2 satisfying *L*≤*K*. The classification performances of AF with different parameters *K* and *L* are shown in Table **3**. If *K* = 64 and *L* = 8, RBF classifier reach best performance: P_{avg} =91.86% and *dev*=1.01%. If *K*= 256 and *L*= 64, RBF classifier reach the worst performance: P_{avg}=53.2% and *dev*=8.28%. For training time spent *t*, more

Table 2. Classification performance of RBF.

Method	P_{avg}(%)	*dev*(%)	*t*(*s*)
AF	91.04	1.35	4.126
GLCM	75.43	3.79	5.12
SS	87.2	5.14	2.062

Table 3. Classification accuracy of RBF.

L	K	P_{avg} (%)	dev(%)	t(s)
4	8	90.7	1.65	7.18
4	16	90.49	1.89	3.93
4	32	90.9	1.51	3.81
4	64	90.74	1.72	3.93
4	128	90.99	1.43	3.81
4	256	90.4	2.8	3.68
8	8	90.8	1.31	3.87
8	16	90.3	1.92	4.28
8	32	91.04	1.35	4.12
8	64	91.86	1.01	4.15
8	128	91.11	0.96	4.25
8	256	90.67	1.14	4.25
16	16	85.46	6.53	4.5
16	32	84.43	4.47	4.68
16	64	82.67	5.71	4.87
16	128	83.89	3.92	4.87
16	256	84.13	4.54	4.84
32	32	75.75	6.22	6.43
32	64	61.78	8.58	6.43
32	128	75.46	7.49	6.53
32	256	79.85	5.99	6.47
64	256	53.2	8.28	9.81

features of extraction results in more training time, that is to say, training time becomes longer with the increase of *L*. From above analysis, we can come to the conclusion that *K* should not be too large in AF and not very small either, let *K*=64 and *L*=8, we can get a good classification result.

5.2. Classification Evaluation Based on Least Mean Square

Least mean square (LMS) method can be used as an adaptive filter to reduce noise effects. Due to its simplicity, it has extensive application and became the standard algorithm of adaptive filtering. The idea of classification is realized by building features template via LMS, then using Naive Bayes to achieve classification results [18]. Here by differing from [18], not the spectrums of halftone images but feature vectors extracted by above three methods are regarded as the input of LMS.

Table **4** shows the classification performances of three types of feature vectors. The classifier on feature vectors of AF achieves good performances: P_{avg}=90.67% and *dev*=3.53% which are better than that of GLCM or SS. The classifier has the worst performances when the feature vectors of GLCM act as the input. For training time spent, training time on feature vector of GLCM is more than that of AF or SS. Table **5** shows the classification performances of AF with different parameters *K* and *L*. From Table **5**, we see that LMS classifier reaches best performance: P_{avg}=90.67% and *dev*=3.53% when *K*=64 and *L*=8. LMS classifier gets the worst performance: P_{avg}=70.1% and *dev*=18.25% when *K*=256 and *L*=64.

5.3. Classification Evaluation Based on PCA

Principal component analysis (PCA) is a good method based on variable covariance matrix for information processing, information compression and feature extraction.

Table 4. Classification performances of LMS.

Method	P_{avg} (%)	*dev*(%)	*t*(*s*)
ACF	90.67	3.53	3.348
GLCM	78.85	3.71	6.64
SS	81.03	11.63	2.862

Table 5. Classification accuracy of LMS.

L	K	P_{avg} (%)	dev(%)	t(s)
4	8	72.86	20.07	2.566
4	16	74.84	18.53	2.478
4	32	75.9	18.04	2.51
4	64	76.28	17.45	2.54
4	128	76.05	16.97	2.556
4	256	75.85	18.9	2.566
8	8	84.4	13.48	3.306
8	16	88.52	3.24	3.304
8	32	89.93	2.52	3.296
8	64	90.67	3.53	3.348
8	128	90.39	2.62	3.344
8	256	90.15	2.95	3.324
16	16	87.47	7.74	4.958
16	32	89.25	3.42	4.944
16	64	89.66	2.57	4.94
16	128	89.86	3.38	4.942
16	256	89.65	3.17	4.936
32	32	88.87	4.61	8.444
32	64	86.08	7.1	8.452
32	128	87.27	6.71	8.434
32	256	82.89	16.63	8.446
64	256	70.1	18.25	15.424

Table 6. Classification performances of PCA.

Method	P_{avg} (%)	*dev*(%)	*t*(*s*)
AF	90.3	1.27	0.62
GLCM	79.07	4.12	0.64
SS	83.68	10.37	0.3

Table 7. Classification accuracy of PCA.

L	K	P_{avg} (%)	dev(%)	t(s)	m
4	8	89.54	2.52	2.62	4
4	16	89.36	2.44	1.894	4
4	32	89.35	2.35	1.88	4
4	64	88.54	3.11	1.88	4
4	128	88.35	3.32	1.58	4
4	256	90.1	1.41	1.28	4
8	8	90.3	1.27	0.64	4
8	16	89.27	3.53	0.64	4
8	32	85.94	8.03	0.32	5
8	64	86.95	6.79	1.86	4
8	128	88.24	4.53	2.18	4
8	256	88.21	4.87	2.5	4
16	16	85.25	8.56	2.8	4
16	32	84.26	8.9	0.92	5
16	64	84.06	7.87	2.5	4
16	128	84.52	7.29	2.82	4
16	256	84.95	7.14	3.12	4
32	32	83.11	8.73	2.18	5
32	64	78.9	10.89	4.7	5
32	128	78.89	10.27	4.68	5
32	256	80.3	9.61	5.62	5
64	256	73.7	11.31	9.06	5

PCA uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components, denoted by *m*, is less than or equal to the number of original variables. Largest *m* principal components are extracted to feed the nearest neighbor classifier. In our experiment, features are extracted from 11000 halftone images (1000 halftone image of each type) as training dataset to create the sample matrix $X_{n\times N}$ (categories are known, *n*=3L in AF and *N*=11000) for input of PCA, and then after PCA transformation, we can get the *m*-dimension feature vectors which are regarded as test sample to fed into the nearest neighbor classifier.

Table **6** shows the classification performances of different types of feature vectors. The classifier on feature vectors of AF achieves good performances: P_{avg}=90.3% and *dev*=1.27% which is better than that of GLCM or SS. The classifier has the worst performances when the feature vectors of GLCM act as input. For training time spent, training time on feature vector of GLCM is more than that of AF or SS. Table **7** shows the classification performances with different parameters *K* and *L*. From Table **7**, we find that *K*=8 and *L*=8 mean PCA classifier reach the best performance: P_{avg}=90.3% and *dev*=1.27%. If *K*=256 and *L*=64, PCA classifier get the worst performance: P_{avg}=73.7% and *dev*=11.31%.

CONCLUSION

In this paper, we have done a lot of work to assess the performance of AF, GLCM and SS for halftone image classification. First we give the details of three modeling methods, and then compare the classification performance of different feature vectors produced by three modeling methods. Experimental results show that the AF is more effective than other two methods. Although we have done research thoroughly on texture feature modeling of halftone images, in fact, there is little research on halftone image classification and there is still a lot to study further, for example, a more robust feature modeling method or a learning method is urgently needed for robust classifier on halftone image. These are our work in the future.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61170102 and No. 61350011, Natural Science Fund of Hunan province in China under Grant No. 14JJ2115 and No.12JJ2036.

REFERENCES

[1] W.B. Huang, A.W. Su, and Y.H. Kuo, “Neural network based method for image halftoning and inverse halftoning”, *Expert Systems with Applications*, vol. 34, pp. 2491-2501, 2008.

[2] Y. Fung and Y. Chan, “A POCS-based restoration algorithm for restoring halftoned color- quantized images”, *IEEE Transaction on Images Processing*, vol. 15, no. 7, pp. 1985-1992, 2006.

[3] K. Chung, and S. Wu, “Inverse halftoning algorithm using edge-based lookup table approach”, *IEEE Transitions on Images Processing,* vol. 14, no. 10, pp.1583-1589, 2005.

[4] N. Suetake, G. Tanaka, “Look-up table and gaussian filter-based inverse halftoning method excellent in gray-scale reproducibility of details and flat regions”, *Optical Review*, vol. 16, no. 6, pp. 594-600, 2009.

[5] C. M. Miceli, and K. J. Parker, “Inverse halftoning”. *Journal of Electron Imaging*, vol. 1, pp. 143-151, 1992.

[6] R. Stevenson, “Inverse halftoning *via* MAP estimation”, *IEEE Transitions on Image Processing*, vol. 6, no. 4, pp. 574-583, 1997.

[7] M. Mese, and P. Vaidyanathan, “Optimized halftoning using dot diffusion and methods for inverse halftoning”. *IEEE Transitions on Images Processing*, vol. 9, no. 4, pp. 691-709, 2000.

[8] M. Mese and P. P. Vaidyanathan, “Recent advances in digital halftoning and inverse halftoning methods”, *IEEE Transitions on Circuits and Systems-I: Fundamental Theory and Applications,* vol. 49, no. 6, pp. 790-805, 2002.

[9] Z. Fan and R. Eschbach, “Limit cycle behavior of error diffusion”. *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 1041-1045, 1994.

[10] R.A. Ulichney. “Dithering with blue noise”, *Proceedings of the IEEE*, vol. 76, no. 1, pp. 56-79, 1988.

[11] L. Liu, and G. Kuang. “Overview of image textural feature extraction methods”, *Journal of Image and Graphics,* vol. 14, no. 4, pp. 622-635, 2009.

[12] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Texture features for image classification”, *IEEE Transitition on System, Man and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.

[13] Z. Guo, L. Zhang, and D. Zhang. “A completed modeling of local binary pattern operator for texture classification”, *IEEE Transaction on Images Processing*, vol.19, no.6, pp.1657-1663, 2010.

[14] M. M. Galloway, “Texture analysis using gray level run length”, *Computer Graphics Image Process*, vol. 45, no. 4, pp. 172-179, 1975.

[15] P. C. Chang, and C. S. Yu, “Neural net classification and LMS reconstruction to halftone images,” *Proceedings of SPIE-The International Society for Optical Engineering,* vol. 33, no. 9, pp. 592-602, 1997.

[16] Y. Kong, P. Zeng, and Y. Zhang, “A classification and recognition algorithm for halftone image”, *Journal of Xidian University,* vol. 38, no. 5, pp. 62-69, 2011.

[17] Y. Kong. “*A study of inverse halftoning and quality assessment schemes*”, Xidian University Dissertation, 2008.

[18] Y. F. Liu, J. M. Guo, and J. D. Lee. “Halftone image classification using LMS algorithm and naive bayes”, *IEEE Transitions on Images Processing*, vol. 20, pp. 10, pp. 2837-2847, 2011.

[19] Y.F. Liu, J.M. Guo, and J.D. Lee. “Inverse Halftoning Based on the Bayesian Theorem”, *IEEE Transaction on Images Processing*, vol. 20, no. 4, pp. 1077-1084, 2011.

[20] S. Aksoy, and R. M. Haralick, “Feature normalization and likelihood-based similarity measures for image retrieval”, *Patter Recognition Letters*, vol. 22, no. 5, pp. 563-582. 2001.

[21] P. S. Li and J. P. Allebach, “Look-up-table based halftoning algorithm”, *IEEE Transitions on Images Processing*, vol.9, no.9, pp. 1593-1603, 2000.

[22] D. J. Lieberman, and J. P. Allebach, “Efficient model based halftoning using direct binary search”, *Proceedings. of International Conference on Image Processing*, vol. 1, pp. 775-778, 1997.

[23] R. Ulichney, “*Digital halftoning and the physical reconstruction function*”, Ph.D. Thesis, MIT, Cambridge, MA, 1986.

[24] D. E. Knuth, “Digital halftones by dot diffusion”, *ACM Transition Graphics.*, vol. 6, no. 4, pp. 245–273, 1987.

[25] D. L. Lau and G. R. Arce. “Modern Digital halftoning (2nd)”, *CRC Press*, 2008.

Received: September 22, 2014 Revised: November 04, 2014 Accepted: November 05, 2014