2015

Blurring and deblurring digital images using the dihedral group

Husein Hadi Abbas Jassim

Zahir M. Hussain
Edith Cowan University

Hind R.M. Shaaban

Kawther B.R. Al-dbag

10.14569/IJARAI.2015.041204

Husein Hadi Abbas Jassim, Zahir M. Hussain, Hind R.M Shaaban and Kawther B.R. Al-dbag, “Blurring and Deblurring Digital Images Using the Dihedral Group” International Journal of Advanced Research in Artificial Intelligence(IJARAI), 4(12), 2015. 10.14569/IJARAI.2015.041204. Available here.

(IJARAI) International Journal of Advanced Research in Artificial Intelligence,
Vol. 4, No.12, 2015

Blurring and Deblurring Digital Images Using the Dihedral Group

Husein Hadi Abbas Jassim
Faculty of Computer Science and Mathematics
University of Kufa
Najaf, Iraq

Hind R.M Shaaban
Faculty of Computer Science and Mathematics
University of Kufa
Najaf, Iraq

Zahir M. Hussain
Faculty of Computer Science and Mathematics
University of Kufa
Najaf, Iraq
Adj. Prof., School of Engineering, ECU, Australia

Kawther B.R. Al-dbag
Faculty of Computer Science and Mathematics
University of Kufa
Najaf, Iraq

***Abstract*—A new method of blurring and deblurring digital images is presented. The approach is based on using new filters generating from average filter and H-filters using the action of the dihedral group. These filters are called HB-filters; used to cause a motion blur and then deblurring affected images. Also, enhancing images using HB-filters is presented as compared to other methods like Average, Gaussian, and Motion. Results and analysis show that the HB-filters are better in peak signal to noise ratio (PSNR) and RMSE.**

Keywords—Dihedral group; Kronecker Product; motion blur and deblur; digital image

I. INTRODUCTION

This template, There are three main categories of image processing, image enhancement, image compression and restoration and measurement extraction [3,6]. A digital image is divided into pixels. Each pixel has a magnitude that represents intensity. The camera uses the recorded image as a faithful representation of the scene that the user saw, but every image is more or less burry. Blurring may arise in the recording of image, because it is unavoidable the scene information "spills over" to neighboring pixels. When there is motion between the camera and image objects during photographing, the motion blur the image. In order to recover motion-blurred images, mathematical model of blurring process are used [1]. Many authors studied motion blur. Often, it is not easy or convenient to eliminate the blur technically. Mathematically, motion blur is modeled as a convolution of point spread function (filters) denoted by (PSF) with the image represented by its intensities. The original image must be recovered by using mathematical model of the blurring process which is called image deblurring [7]. Many researchers introduced algorithms to remove blur such as Average filter AF (or Mean filter), Gaussian filter (GF). The Gaussian filter is equivalent to filtering with a mask of radius R, whose weights are given by Gaussian function: $(x,y)=\frac{1}{2\pi\sigma^2}e^{\frac{-(x^2+y^2)}{2\sigma^2}}$, $x\in R$; where σ is stander deviation of the Gaussian: large σ for more intensive smoothing) [2]. Motion Blur effect filter is a filter that makes the image appear to be moving by adding a blur in a specific direction [10]. The largest subgroup H of dihedral group D_n is found in [4].

In this work, Markov basis *HB* is used to introduce a new filters from Average filter for adding and removing motion blur of image, denoted by *HB*-filters.

II. PRELIMINARY CONCEPTS

This section reviews the preliminaries about H-filters, Dihedral group, Convolution and Deconvolution processes.

A. H-Filters

H-filters are 18 elements as per the following set [5].

$$\mathbf{z}_1=\begin{bmatrix}1&-1&0\\-1&1&0\\0&0&0\end{bmatrix};\ \mathbf{z}_2=\begin{bmatrix}0&0&0\\1&-1&0\\-1&1&0\end{bmatrix};$$

$$\mathbf{z}_3=\begin{bmatrix}1&0&-1\\-1&0&1\\0&0&0\end{bmatrix};\ \mathbf{z}_4=\begin{bmatrix}0&0&0\\1&0&-1\\-1&0&1\end{bmatrix};$$

$$\mathbf{z}_5=\begin{bmatrix}0&1&-1\\0&-1&1\\0&0&0\end{bmatrix};\ \mathbf{z}_6=\begin{bmatrix}0&1&-1\\0&0&0\\0&-1&1\end{bmatrix};$$

$$\mathbf{z}_7=\begin{bmatrix}0&0&0\\0&1&-1\\0&-1&1\end{bmatrix};\ \mathbf{z}_8=\begin{bmatrix}1&-1&0\\0&0&0\\-1&1&0\end{bmatrix};$$

$$\mathbf{z}_9=\begin{bmatrix}1&0&-1\\0&0&0\\-1&0&1\end{bmatrix};\ \mathbf{z}_{10}=\begin{bmatrix}-1&1&0\\1&-1&0\\0&0&0\end{bmatrix};$$

$$\mathbf{z}_{11}=\begin{bmatrix}0&0&0\\-1&1&0\\1&-1&0\end{bmatrix};\ \mathbf{z}_{12}=\begin{bmatrix}-1&0&1\\1&0&-1\\0&0&0\end{bmatrix};$$

$$\mathbf{z}_{13}=\begin{bmatrix}0&0&0\\-1&0&1\\1&0&-1\end{bmatrix};\ \mathbf{z}_{14}=\begin{bmatrix}0&-1&1\\0&1&-1\\0&0&0\end{bmatrix};$$

$$\mathbf{z}_{15}=\begin{bmatrix}0&-1&1\\0&0&0\\0&1&-1\end{bmatrix};\ \mathbf{z}_{16}=\begin{bmatrix}0&0&0\\0&-1&1\\0&1&-1\end{bmatrix};$$

$$\mathbf{z}_{17}=\begin{bmatrix}-1&1&0\\0&0&0\\1&-1&0\end{bmatrix};\ \mathbf{z}_{18}=\begin{bmatrix}-1&0&1\\0&0&0\\1&0&-1\end{bmatrix};$$

B. Definition 1: Dihedral Group

Let n be a positive integer greater than or equal to 3. The group of all symmetries of the regular polygon with n sides, including both rotations and reflections, is called **dihedral group** and denoted by D_n[13]. The 2n elements in D_n can be written as: $\{e, r, r^2, \ldots, r^{n-1}, s, sr, sr^2, sr^{n-1}\}$, where e is the identity element in D_n. In general, we can write D_n as: $D_n=\{s^j r^k : 0 \le k \le n-1, 0 \le j \le 1\}$ which has the following properties:

$r^n = 1$, $sr^k s = r^{-k}$, $(sr^k)^2 = e$, for all $0 \le k \le n-1$. The composition of two elements of the D_n is given by $r^i r^j = r^{i+j}$, $r^i s r^j = sr^{j-i}$, $sr^i r^j = sr^{i+j}$, $sr^i s r^j = r^{j-i}$.

C. 2D Convolution

Assume two discrete 2-dimensional images $f(x, y)$ and $h(x, y)$. Their *convolved* (or *folded*) *sum* is the image $g(x, y)$, the convolution of these two functions is defined as [12]:

$g(x,y) = f(x,y) \otimes h(x,y)$, so

$$f(x,y) \otimes h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n) \tag{1}$$

For $0 \le x, m \le M-1;\ 0 \le y, n \le N-1$,

where $M \times N$ is a size of $h(x,y)$.

III. 2D DISCRETE FOURIER TRANSFORM

The two-dimensional *discrete Fourier transform* (DFT) of the image function *f*(*x*, *y*) is defined as,

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{2}$$

where *f(x, y)* is a digital image of size $M \times N$, and the discrete variable *u* and *v* in the ranges: *u = 0, 1, 2, …, M-1* and *v = 0, 1, 2, …, N-1*[11].

Given the transform *F(u, v)*, we can obtain *f(x, y)* by using the *inverse discrete Fourier transform* (IDFT):

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{3}$$

It can be shown by direct substitution into Eq. 2 and Eq. 3 that the *Fourier transform* pair satisfies the following translation properties:

$$f(x-m, y-n) \Leftrightarrow F(u,v)e^{-i2\pi\left(\frac{um}{M}+\frac{vn}{N}\right)} \tag{4}$$

Now, interested in finding the Fourier transform of Eq. 1:

$\mathcal{F}\big(f(x,y) \otimes h(x,y)\big) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, h(x-m, y-n)] e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$, so by Eq. 4 we have,

$\mathcal{F}\big(f(x,y) \otimes h(x,y)\big) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\, H(u,v)\, e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} = F(u,v)\, H(u,v)$.

This result of the *convolution theorem* is written as:

$$f(x,y) \otimes h(x,y) \Longleftrightarrow F(u,v)H(u,v) \tag{5}$$

The transform of the original image simply by dividing the transform of the degraded image $G(u,v)$, by the degradation function $H(u,v)$ is

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \tag{6}$$

that's called inverse filter [9].

A. Fourier Spectrum

Because the 2-D *DFT* is complex in general [8], it can be expressed in polar form: $F(u,v) = |F(u,v)|e^{-i\emptyset(u,v)}$

where the magnitude,

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{\frac{1}{2}} \tag{7}$$

is called the Fourier (or frequency) spectrum. The power spectrum is defined as,

$P(u.v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$.

As before, *R and J* are the real and imaginary parts of *F(u, v)* and all computations are carried out for the discrete variables *u* = 0, 1, 2, ... , *M* -1 and *v* = 0,1,2, ... , *N*-1. Therefore,

|*F(u, v)*|, $\emptyset(u,v)$, and *P(u, v)* are arrays of size $M \times N$.

B. Image Restoration based on Wiener Deconvolution

The method considers images and noise as random variables, and the objective is to find an estimate $\hat{f}$ of the uncorrupted image f such that the mean square error *(MSE)* between them is minimized. This error measure is given by:

$$e^2 = E\left\{\left(f - \hat{f}\right)^2\right\} \tag{8}$$

Based on these conditions, the minimum of the error function in Eq. 8 is given in the frequency domain by the expression:

$$\begin{aligned}\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2+S_\eta(u,v)}\right] G(u,v) \\ &= \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right] G(u,v)\end{aligned} \tag{9}$$

The terms in Eq. 9 are as follows:

H(u,v) = degradation function& $H^*(u,v)$ = complex conjugate of H(u, v)& $|H(u,v)|^2 = H^*(u,v)H(u,v)$ & $S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise& $S_f(u,v) = |F(u,v)|^2$ = power spectrum of the original image& *G(u, v)*= the transform of the degraded image. Note that if the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter.

IV. THE PROPOSED APPROACH

H-filters are used to generate ***HB-filters*** by adding each element in ***H-filters*** to the average filter, so we got some ***HB-filters*** with dimensions *3-by-3* and each of which has type of blur different from the other.

Then the ***HB-filters*** can be extended using tenser product (by operation $\circledast$) to larger sizes, in order to get a higher degrees of blur in digital images. Take any one of ***HB-filters*** *h(x,y)* of dimension *3-by-3* and extend it by identity matrix I_n, *n-by-n* where *n* is an odd number greater than or equals 3, by Tensor Product *T*:

$$T(x,y) = h(x,y) \circledast I_n(x,y) = \begin{bmatrix} h_{11} \times I_n & h_{12} \times I_n & h_{13} \times I_n \\ h_{21} \times I_n & h_{22} \times I_n & h_{23} \times I_n \\ h_{31} \times I_n & h_{32} \times I_n & h_{33} \times I_n \end{bmatrix}_{3n \times 3n}$$

This filter will be called ***extended HB-filter*** generated from ***HB-filter*** *h(x,y)* and I_n **.**

Example 1.

Choose any one of ***H-filters***: $\mathbf{z}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$

Divide $\mathbf{z}_2$ by 9, and add it to the average filter (A_f) as follows:

$$h_1 = \boldsymbol{z_2} + \boldsymbol{A_f} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \Big/ 9 + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Big/ 9 =$$

$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \Big/ 9$. So, $h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \Big/ 9$ it's one of ***HB-filters***.

Now use the action largest subgroup $\boldsymbol{\mathcal{H}} = \{e, r^{\frac{n}{3}}, r^{\frac{2n}{3}}, sr, sr^{1+\frac{n}{3}}, sr^{1+\frac{2n}{3}}\}$ of dihedral group [4], to generate other ***HB-filters***. So, h_1 can be represented as 9-dimensional column vector,

$$h_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} \Big/ 9 \in \mathbb{Z}^9 ,$$

and calculate element of $\boldsymbol{\mathcal{H}}$ in D_9 as

$$r^{\frac{n}{3}} = r^3$$

$$= \left(1 \ \ \frac{n}{3}+1 \ \ \frac{2n}{3}+1\right)\left(2 \ \ \frac{n}{3}+2 \ \ \frac{2n}{3}+2\right) \dots \left(\frac{n}{3} \ \ \frac{2n}{3} \ \ n\right)$$

$$= (1\ 4\ 7\ )(2\ 5\ 8\ )(3\ 6\ 9\ ) .$$

To find $T_{r^3} h_1$, one has:

$$T_{r^3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

then

$$T_{r^3} h_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} \Big/ 9$$

$$= \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 0 \end{bmatrix} \Big/ 9 = h_2 ,$$

So, $h_2 = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix} \Big/ 9 \in \boldsymbol{HB}$

Similarly, one obtains

$$r^{\frac{2n}{3}} = sr^6$$

$$= \left(1 \ \ \tfrac{2n}{3}+1 \ \ \tfrac{n}{3}+1\right)\left(2 \ \ \tfrac{2n}{3}+2 \ \ \tfrac{n}{3}+2\right) \dots \left(\tfrac{n}{3} \ \ n \ \ \tfrac{2n}{3}\right)$$

$$= (1\ 7\ 4)(2\ 8\ 5)(3\ 9\ 6) \cdot h_1 = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \end{bmatrix} \Big/ 9 = h_3 \, .$$

$$sr = (1 \ \ n)(2 \ \ n-1) \dots \left(\tfrac{n}{3} \ \ \tfrac{2n}{3}+1\right)\left(\tfrac{n}{3}+1 \ \ \tfrac{2n}{3}\right) \dots \left(\tfrac{n-1}{2} \ \ \tfrac{n+3}{2}\right)$$

$$= (1\ 9)(2\ 8)(3\ 7)(4\ 6)(4\ 6) \cdot h_1 = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Big/ 9 = h_4 \, .$$

$$sr^{\frac{n}{3}+1} = sr^4$$

$$= \left(1 \ \ \frac{2n}{3}\right)\left(2 \ \ \frac{2n}{3}-1\right) \dots \left(\frac{n}{3} \ \ \frac{n}{3}+1\right)\left(\frac{2n}{3}+1 \ \ n\right) \dots$$

$$\cdots \left(\frac{5n-3}{6} \ \ \frac{5n+9}{6}\right)$$

$$= (1 \ \ 6)\ (2\ 5)(3\ 4)\ (7\ 9)(7\ 9) \cdot h_1 = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 2 & 0 & 1 \end{bmatrix} \Big/ 9 = h_5 \, .$$

$$sr^{\frac{2n}{3}+1} = \left(1 \ \ \frac{n}{3}\right)\left(2 \ \ \frac{n}{3}-1\right) \cdots \left(\frac{n-3}{6} \ \ \frac{n+9}{6}\right)\left(\frac{n}{3}+1 \ \ n\right) \times$$

$$\times \left(\frac{n}{3}+2 \ \ n-1\right) \cdots \left(\frac{2n}{3} \ \ \frac{2n}{3}+1\right)$$

$$= (1\ 3)\ (2\ 2)\ (1\ 3)\ (4\ 9)\ (5\ 8)\ (6\ 7) \cdot h_1$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \Big/ 9 = h_6$$

All of these filters belong to ***HB-filters***.

Most ***HB-filters*** can be obtained using other ***H-filters***. For example the ***extended HB-filters*** generated from ***HB-filter***

$h(x, y) = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ with I_3 is given by

$$T(x,y) = h(x,y) \circledast I_3(x,y) = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}_{9\times 9}$$

A. *Blurring*

This sub-section describes the standard filters algorithm for addition blur of an image by using the convolution theorem.

Blur algorithm

Consider an image matrix *f(x, y)* of dimension *m-by-n*, which can be written as follows:

$$f(x,y) = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m} & \cdots & f_{mn} \end{bmatrix}_{m\times n}$$

. And ***HB-filter*** *h(x,y) p-by-q* dimension defined as, $h(x,y) = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1q} \\ h_{21} & h_{22} & \cdots & h_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p1} & h_{p2} & \cdots & h_{pq} \end{bmatrix}_{p\times q}$.

Step1: In the beginning add *f(x,y)* by *p-1* rows with zeros from up and down, and *p-1* columns with zeros from left and right, such that the result is *{m+2(p-1)}-by-{n+2(q-1)}* dimensions, as follows:

$$f(x,y) = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0_{1j} \\ \vdots & \vdots & \vdots & \dots & 0 & 0 & 0 \\ 0 & \cdots & f_{11} & & f_{1n} & \vdots & \vdots \\ \vdots & & \vdots & \ddots & \vdots & & \\ 0 & 0 & f_{m1} & \dots & f_{mn} & 0 & 0 \\ 0_{i1} & 0 & 0 & & 0 & 0 & 0_{ij} \\ & & 0 & & 0 & & \end{bmatrix}_{i\times j},$$

where *i= m+2(p-1)* and *j= m+2(q-1)*.

Step2: Reverse $h(x,y)$ (that used in blurring) for two directions,

$$h(x,y) = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1q} \\ h_{21} & h_{22} & \cdots & h_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p1} & h_{p2} & \cdots & h_{pq} \end{bmatrix}$$

$$\xRightarrow{rev} h(x,y) = \begin{bmatrix} h_{pq} & \cdots & h_{p2} & h_{p1} \\ h_{2q} & \cdots & h_{22} & h_{21} \\ \vdots & \ddots & \vdots & \vdots \\ h_{1q} & \cdots & h_{12} & h_{11} \end{bmatrix}_{p\times q}$$

Step3: Make the two arrays as follows:

$$h(x,y) = \begin{bmatrix} h_{pq} & \cdots & h_{p2} & h_{p1} \\ h_{2q} & \cdots & h_{22} & h_{21} \\ \vdots & \ddots & \vdots & \vdots \\ h_{1q} & \cdots & h_{12} & h_{11} \end{bmatrix}$$

$$f(x,y) = \begin{bmatrix} 0 & \cdots & 0_{1q} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & 0 & 0 \\ 0_{p1} & \cdots & f_{11} & f_{12} & 0 & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots & 0 & 0 & \vdots \\ 0 & & f_{m1} & f_{m2} & \ddots & \vdots & \vdots \\ 0 & \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{i\times j}.$$

Step4: Calculate the convolution equation for all pixels of blurred matrix *g(x,y)*:

$$g(x,y) = f(x,y) \otimes h(x,y) = \sum_{i=1}^{p} \sum_{j=1}^{q} f(i,j)h(i,j)$$

So,

$$\begin{aligned} g(1,1) = {} & (0 \times h_{pq}) + (0 \times h_{p2}) + (0 \times h_{p1}) + (0 \times h_{2p}) \\ & + \cdots + (0 \times h_{22}) + (0 \times h_{21}) + (0 \times h_{1q}) \\ & + \cdots + (0 \times h_{12}) + (f_{11} \times h_{11}) \\ = {} & (f_{11} \times h_{11}). \end{aligned}$$

After that shift the filter *h(x,y)* as much as one column as follows:

$$h(x,y) = \begin{bmatrix} h_{pq} & \cdots & h_{p2} & h_{p1} \\ h_{2q} & \cdots & h_{22} & h_{21} \\ \vdots & \ddots & \vdots & \vdots \\ h_{1q} & \cdots & h_{12} & h_{11} \end{bmatrix}$$

$$f(x,y) = \begin{bmatrix} 0 & 0 & \cdots & 0 & \dots & 0 & 0 & 0_{1j} \\ & \vdots & \ddots & \vdots & \dots & 0 & 0 & 0 \\ 0 & 0 & f_{11} & f_{12} & & f_{1n} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & f_{m1} & f_{m2} & \dots & f_{mn} & & \\ 0 & 0 & 0 & 0 & & 0 & 0 & 0 \\ 0_{i1} & 0 & 0 & 0 & & 0 & 0 & 0_{ij} \end{bmatrix}_{i\times j}$$

Also,

$$\begin{aligned} g(1,2) = {} & (0 \times h_{pq}) + \cdots + (0 \times h_{p2}) + (0 \times h_{p1}) \\ & + (0 \times h_{2q}) + \cdots + (0 \times h_{22}) + (0 \times h_{21}) \\ & + \cdots + (0 \times h_{1q}) + \cdots + (f_{11} \times h_{12}) \\ & + (f_{12} \times h_{11}) = (f_{11} \times h_{12}) + (f_{12} \times h_{11}) \end{aligned}$$

Now repeat step 4 to obtain digital image convolution *g(x,y)* at all times that the two arrays overlap. We continue until we find *g(r, c)*, where *r & c=m+ (p-1)*, then the final form of the blurred matrix *g(x,y)* is:

$$g(x,y) = \begin{bmatrix} g_{11} & \cdots & g_{1c} \\ \vdots & \ddots & \vdots \\ g_{r} & \cdots & g_{rc} \end{bmatrix}_{r\times c}.$$

Step5: Delete from *g(x,y)* as much as $\frac{p-1}{2}$ rows from up and down, and $\frac{p-1}{2}$ columns from left and right, such that the blurred matrix *g(x,y)* becomes *m-by-n* in dimension:

$$g(x,y) = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{m} & \cdots & g_{mn} \end{bmatrix}_{m\times n}.$$

Example 2.

Suppose the image matrix *f(x,y)* is:

$f(x,y)=\begin{bmatrix}209 & 90 & 60\\ 0 & 77 & 30\\ 100 & 46 & 20\end{bmatrix}_{3\times3}$. We blur this matrix with one of the ***HB-filters***: $h(x,y)=\begin{bmatrix}2 & 0 & 1\\ 0 & 2 & 1\\ 1 & 1 & 1\end{bmatrix}/9$.

Step1: Add two rows from up and down, and two columns from left and right of zeros for the matrix *f(x, y)*, such that becomes *7-by-7* dimension, as follows:

$$f(x,y)=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&209&90&60&0&0\\0&0&0&77&30&0&0\\0&0&100&46&20&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\end{bmatrix}_{7\times7} .$$

Step2: Reverse the filter *h(x,y)* for two directions:

$$h(x,y)=\begin{bmatrix}2&0&1\\0&2&1\\1&1&1\end{bmatrix}/9 \overset{rev}{\Longrightarrow}\begin{bmatrix}1&1&1\\1&2&0\\1&0&2\end{bmatrix}/9$$

Step3: Make the two arrays, as the following form:

$$h(x,y)=\begin{bmatrix}1&1&1\\1&2&0\\1&0&2\end{bmatrix}/9$$

$$f(x,y)=\begin{bmatrix}0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&209&90&60&0&0\\0&0&0&77&30&0&0\\0&0&100&46&20&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\end{bmatrix}_{7\times7}$$

Step4: Calculate the convolution equation for all pixels of blurred matrix *g(x,y)*:

$$g(x,y)=f(x,y)\otimes h(x,y)=\sum_{m_1=1}^{3}\sum_{n_1=1}^{3}f(m_1,n_1)h(m_1,n_1)$$

Now, $g(1,1)=(209\times0.2222)=26.4444$

After that, shift the filter *h(x,y)* as much as one column, then repeat the same step.

So, $g(1,2)=(90\times0.2222)=20$
$g(1,3)=(209\times0.1111)+(60\times0.2222)=36.5556$
$\vdots$
$g(5,5)=(20\times0.2222)=2.2222$

The final form of the blurred matrix $g(x,y)$ is:

$$\begin{bmatrix}46.4444&20&36.5556&10&6.6667\\0&63.5556&94.8889&31.8889&10\\45.4444&43.4444&72.5556&37&12.2222\\0&30.7778&33.2222&21.4444&5.5556\\11.1111&16.2222&18.444&7.3333&2.2222\end{bmatrix}_{5\times5}$$

Step5: Delete from *g(x,y)* as much as *one* row from up and down, and *one* column from left and right, such that the result is the blurred matrix $g_1(x,y)$ *3-by-3* dimension,

$$g_1(x,y)=\begin{bmatrix}63.5556&49.39&31.57\\43.4444&72.5556&37\\30.7778&33.2222&21.4444\end{bmatrix}_{3\times3} .$$

B. *Deblurring*

Here we express the proposed deblurring method.

Deblur Algorithm

Weiner deconvolution for the matrix *g(x,y)* and *h(x,y)* is given by:

$$\hat{F}(u,v)=\left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2+S_\eta(u,v)/S_f(u,v)}\right]G(u,v) .$$

Suppose there is no noise (i.e.$\frac{S_\eta(u,v)}{S_f(u,v)}=0$), then the noise of power spectrum vanishes and the Weiner reduces to the invers filter, so one has: $\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}$.

Step 1: Find Fourier transform of the blurred matrix *g(x,y) r-by-c* dimensions,

$$G(u,v)=\sum_{x=1}^{m}\sum_{y=1}^{n}g(x,y)\,e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} .$$

Step 2: Find Fourier transform of **HB-filter** h(x,y). $H(u,v)=\sum_{x=1}^{m}\sum_{y=1}^{n}h(x,y)\,e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$,

If the dimension of *h(x,y)* is less than dimension of *g(x,y)*, we will add zeros for *h(x,y)* to create as same as the dimension of the image matrix *g(x,y)* before doing the transform, such that the result is *m-by-n* in dimension.

Step 3: Calculate the transform of estimated image $\hat{F}(u,v)$.

Step 4: Find estimated image $\hat{f}(x,y)$ by taking inverse Fourier transform of $\hat{F}(u,v)$, by follows:

$$\hat{f}(x,y)=\frac{1}{MN}\sum_{u=1}^{m}\sum_{v=1}^{n}\hat{F}(u,v)\,e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} .$$

Step 5: Remove zeros from $\hat{f}(x,y)$as much as $(p-1)/2$ of last rows and columns, where resulted dimensions equal to dimensions original image matrix *f(x,y)*.

Example 2. We will take blurred matrix *g(x,y)* from ex.2, $g(x,y)=$

$$\begin{bmatrix}46.4444&20&36.5556&10&6.6667\\0&63.5556&94.8889&31.8889&10\\45.4444&43.4444&72.5556&37&12.2222\\0&30.7778&33.2222&21.4444&5.5556\\11.1111&16.2222&18.444&7.3333&2.2222\end{bmatrix}_{5\times5} ,$$

with ***HB-filter***, $h(x,y)=\begin{bmatrix}2&0&1\\0&2&1\\1&1&1\end{bmatrix}_{3\times3}/9$.

Now, from the Weiner equation, suppose that $\frac{S_\eta(u,v)}{S_f(u,v)}=0$, then the Weiner reduces to the invers filter as following, $\hat{F}(u,v)=\frac{G(u,v)}{H(u,v)}$.

Step 1: Find Fourier transform of the matrix $g(x,y)$, $G(u,v)=\sum_{x=1}^{m}\sum_{y=1}^{n}g(x,y)\,e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$

Now, $G(1,1)=\sum_{x=1}^{5}\sum_{y=1}^{5} g(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{y}{5}\right)}$

$$=\left(g(1,1)e^{-j2\pi\left(\frac{1}{5}+\frac{1}{5}\right)}\right)+\left(g(1,2)e^{-j2\pi\left(\frac{1}{5}+\frac{2}{5}\right)}\right)$$
$$+\left(g(1,3)e^{-j2\pi\left(\frac{1}{5}+\frac{3}{5}\right)}\right)$$
$$+\left(g(1,4)e^{-j2\pi\left(\frac{1}{5}+\frac{4}{5}\right)}\right)+\cdots$$
$$+\left(g(5,5)e^{-j2\pi\left(\frac{5}{5}+\frac{5}{5}\right)}\right)$$
$$=46.4444e^{-j\left(\frac{4}{5}\right)\pi}+20e^{-j\left(\frac{6}{5}\right)\pi}+36.5556e^{-j\left(\frac{8}{5}\right)\pi}$$
$$+10e^{-j2\pi}+\cdots+2.2222e^{-j4\pi}=632+0j$$

$$G(1,2)=\sum_{x=1}^{5}\sum_{y=1}^{5} g(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{2y}{5}\right)}$$
$$=-89.44-191.15j$$

$$G(1,3)=\sum_{x=1}^{5}\sum_{y=1}^{5} g(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{3y}{5}\right)}$$
$$=30.94+17.24\mathrm{j}$$

$\vdots$

$$G(5,5)=\sum_{x=1}^{5}\sum_{y=1}^{5} g(x,y)\, e^{-j2\pi\left(\frac{5x}{5}+\frac{5y}{5}\right)}$$
$$=-1.13-45.84\mathrm{j}$$

So, the final form of *G (u,v)* be

$$\begin{bmatrix} 632+0j & -89.44-191.15j & 30.94+17.24j & 30.94-17.24j & -89.44+191.15j \\ -59.29-165.44j & -1.13+45.84j & 7.69+13.15 & 17.02+4.9j & 101.27+20.83j \\ 42.45-55.03j & 31.43+42.17j & 42.35+97.85j & 98.29+36.24j & 42.97+17.47j \\ 42.45-55.03j & 42.97-17.47j & 98.29-36.24j & 42.35-97.85j & 31.43-42.17j \\ -59.29+165.44j & 101.27-20.83j & 17.02-4.9j & 7.69-13.15j & -1.13-45.84j \end{bmatrix}_{5\times5}$$

Step 2: Because of the dimension of *h(x,y)* is less than dimension of *g(x,y)*, then add zeros for *h(x,y)* to create as same as the dimensions of the image matrix *g(x,y)*, so we have:

$$h(x,y)=\begin{bmatrix} 2 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5\times5} \Big/ 9,$$

After that, we are doing the Fourier transform of *h(x,y)*:
$H(u,v)=\sum_{x=1}^{m}\sum_{y=1}^{n} h(x,y)\, e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$

Now, $H(1,1)=\sum_{x=1}^{5}\sum_{y=1}^{5} h(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{y}{5}\right)}$

$$=\left(h(1,1)e^{-j2\pi\left(\frac{1}{5}+\frac{1}{5}\right)}\right)+\left(h(1,2)e^{-j2\pi\left(\frac{1}{5}+\frac{2}{5}\right)}\right)$$
$$+\left(h(1,3)e^{-j2\pi\left(\frac{1}{5}+\frac{3}{5}\right)}\right)$$
$$+\left(h(1,4)e^{-j2\pi\left(\frac{1}{5}+\frac{4}{5}\right)}\right)+\cdots$$
$$+\left(h(5,5)e^{-j2\pi\left(\frac{5}{5}+\frac{5}{5}\right)}\right)$$
$$=2e^{-j\left(\frac{4}{5}\right)\pi}+0e^{-j\left(\frac{6}{5}\right)\pi}+1e^{-j\left(\frac{8}{5}\right)\pi}+0e^{-j2\pi}+\cdots+0e^{-j4\pi}$$
$$=1+0j$$

$$H(1,2)=\sum_{x=1}^{5}\sum_{y=1}^{5} h(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{2y}{5}\right)}$$
$$=0.1667-0.5129j$$

$$H(1,3)=\sum_{x=1}^{5}\sum_{y=1}^{5} h(x,y)\, e^{-j2\pi\left(\frac{x}{5}+\frac{3y}{5}\right)}$$
$$=0.1667+0.1211j$$

$\vdots$

$$H(5,5)=\sum_{x=1}^{5}\sum_{y=1}^{5} h(x,y)\, e^{-j2\pi\left(\frac{5x}{5}+\frac{5y}{5}\right)}$$
$$=-0.2828+0.0249j$$

So, the final form of *H(u,v)* is:

$$H(u,v)=\begin{bmatrix} 1+0j & 0.1667-0.5129j & 0.1667+0.1211j & 0.1667-0.1211j & 0.1667+0.5129j \\ 0.1667-0.5129j & -0.2828-0.0249j & 0.1667+0.171j & 0.1667+0.0404j & 0.4444+0j \\ 0.1667+0.1211j & 0.1667+0.171j & 0.3383+0.2767j & 0.4444+0j & 0.1667-0.0404j \\ 0.1667-0.1211j & 0.1667+0.0404j & 0.4444+0j & 0.3383-0.2767j & 0.1667-0.171j \\ 0.1667+0.5129j & 0.4444-0j & 0.1667-0.0404j & 0.1667-0.171j & -0.2828+0.0249j \end{bmatrix}_{5\times5}$$

Step 3: Calculate the Fourier transform of estimated image.

$$\hat{F}(u,v)=G(u,v)/H(u,v)=\begin{bmatrix} 632+0\mathrm{j} & 285.83-267.23\mathrm{j} & 170.67-20.58\mathrm{j} & 170.67+20.58\mathrm{j} & 285.83+267.23\mathrm{j} \\ 257.77-199.34j & -10.23-161.21j & 61.93+15.37j & 103.17+4.41j & 227.85+46.87j \\ 323.73+94.98j & 218.33+29.01j & 216.73+111.98j & 221.15+81.54j & 219.57+158.02j \\ 323.73-94.98j & 219.57-158.02j & 221.15-81.54j & 216.73-111.98j & 218.33-29.01j \\ 257.77+199.34j & 227.85-46.87j & 103.17-4.41j & 61.93-15.37j & -10.23+161.21j \end{bmatrix}_{5\times5}$$

Step 4: Find inverse Fourier transform with only real numbers $\hat{f}(x,y)$ of an array $\hat{F}(u,v)$.

$\hat{f}(x,y)=\frac{1}{MN}\sum_{u=1}^{m}\sum_{v=1}^{n}\hat{F}(u,v)\, e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$,
So,

$$\hat{f}(1,1)=\frac{1}{5\times5}\sum_{u=1}^{m}\sum_{v=1}^{n}\hat{F}(u,v)\, e^{j2\pi\left(\frac{u}{5}+\frac{v}{5}\right)}$$
$$=\frac{1}{5\times5}\Big(\hat{F}(1,1)e^{j2\pi\left(\frac{1}{5}+\frac{1}{5}\right)}$$
$$+\hat{F}(1,2)e^{j2\pi\left(\frac{1}{5}+\frac{2}{5}\right)}+\hat{F}(1,3)e^{j2\pi\left(\frac{1}{5}+\frac{3}{5}\right)}+\cdots$$
$$+\hat{F}(5,5)e^{j2\pi\left(\frac{5}{5}+\frac{5}{5}\right)}\Big)$$
$$=\frac{1}{25}\Big((632+0\mathrm{j})e^{j\left(\frac{4}{5}\right)\pi}+(285.83$$
$$-267.23\mathrm{j})e^{j\left(\frac{6}{5}\right)\pi}+(170.67$$
$$-20.58\mathrm{j})e^{j\left(\frac{8}{5}\right)\pi}+\cdots+(-10.23$$
$$+161.12\mathrm{j})e^{j4\pi}\Big)=209$$

$\hat{f}(1,2)=90$
$\hat{f}(1,3)=60$
$\vdots$
$\hat{f}(5,5)=0$

Now, the final of estimated image $\hat{f}(x,y)$ is $\hat{f}(x,y)=\begin{bmatrix} 209 & 90 & 60 & 0 & 0 \\ 0 & 77 & 30 & 0 & 0 \\ 100 & 46 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5\times5}$.

Step 5: Remove the last two rows and columns of zeros from $\hat{f}(x,y)$:

$\hat{f}(x,y)=\begin{bmatrix} 209 & 90 & 60 \\ 0 & 77 & 30 \\ 100 & 46 & 20 \end{bmatrix}_{3\times3}$, where the original matrix *f(x,y)* is: $g(x,y)=\begin{bmatrix} 209 & 90 & 60 \\ 0 & 77 & 30 \\ 100 & 46 & 20 \end{bmatrix}_{3\times3}$.

Now, we give the (original, blurred, estimated) block image to explain the image enhancement in ex.2 and ex.3 as shown in Fig.1.

Fig. 1. Image blocks in ex.2 & ex.3. Left: original image f(x,y). Middle: blurred g(x,y). Right: estimated image $\hat{f}(x, y)$

TABLE I. THE COMPARISON OF BETWEEN DIFFERENT FILTERS

	Degree of blur	Image blur	Aver. filter	Gauss. filter	Motion filter	Proposed filter
PSNR	9×9	21.44	7.25	21.45	13.78	45.53
	21×21	18.03	7.01	18.04	12.7	49.9
	27×27	17.02	7.03	17.02	11.79	46.23
RMSE	9×9	21.61	110.66	21.58	52.18	1.35
	21×21	31.98	113.72	31.96	59.1	0.81
	27×27	35.95	113.45	35.94	65.65	1.24

C. Comparison with other filters

HB-filters are compared in PNSR (in dB) and RMSE with the (*AF*, *GF*, and *MF*) filters. The proposed method and the other methods are applied on (256× 256) Pepper RGB image, by using (jpg. format) as in Table I. The application of proposed method and some other methods on the color images (in jpg. format) of different blur is shown in Fig.2.

V. CONLCLUSION

Nlur has been added and removed from digital images using HB-filters. The HB-filters perform well for grayscale, binary and color (jpg, png) images with different blur degrees. Results show that the HB method has higher PSNR and less RMSE than Average, Gaussian and Motion methods.

ACKNOWLEDGMENT

We would like to thank the University of Kufa for financial support.

REFERENCES

[1] B. Jiang, A. Yang, C. wang, and Z. Hou, "Comparison of Motion-blurred Image Restoration Using Wiener Filter and Spatial Difference Technique", International Journal of Signal Processing, vol. 7, pp.11-22, 2014.

[2] D. Majerova, "Image Processing by Means of Lukasiewicz Algebra with Square Root", Institute of Chemical Technology, Pregue, Department of Computing and Control Engineering, 2004.

[3] Gonzalez, R.C., Woods, R.E., "Digital Image Processing", 2nd Ed, Prentice-Hall of India Pvt. Ltd, 2002.

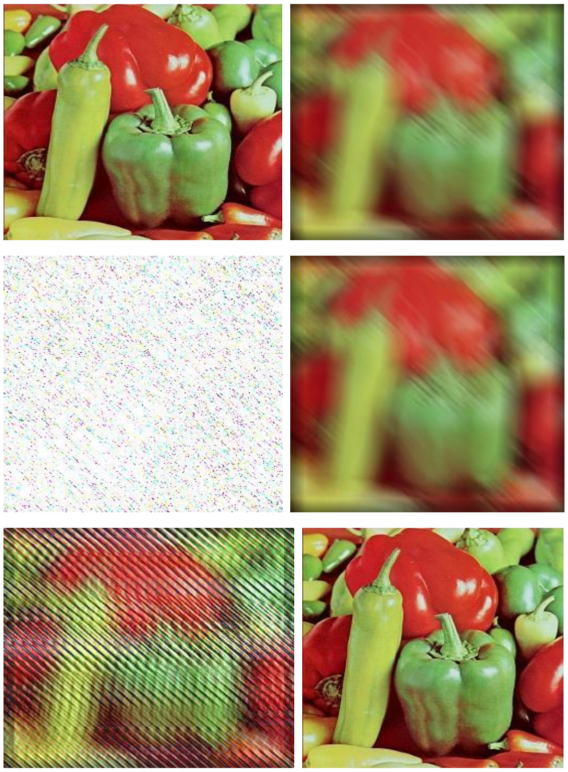

Fig. 2. Application on Pepper (jpg. format) RGB image with degree of blur 27*27. Top Left: Original. Top Right: Blur image PSNR=17.02, RMSE=35.95. Middle Left: A.F, PSNR=7.03, RMSE=113.45. Middle Right: G.F, PSNR=17.02, RMSE =35.94> Bottom Left: M.F, PSNR=11.79, RMSE =65.65. Bottom Right: Proposed, PSNR=46.23, RMSE=1.24

[4] H. H. Abbass, and H. S. Mohammed Hussein "An Invariant Markov basis Under the Action of Largest Subgroup of Dihedral Group D_{3^m} ", European Journal of Scientific Research, Vol. 125, pp. 265-277, 2014.

[5] H. H. Abbass, and H. S. Mohammed Hussein " On Toric Ideals for $3 \times \frac{n}{3}$ -Contingency Tables with Fixed Two Dimensional Marginals n is a multiple of 3", European Journal of Scientific Research, Vol. 123, pp. 83-98, 2014.

[6] Madasu Hanmandlu Member IEEE and Devendra Jha “An Optimal Fuzzy System for color image Enhancement” , IEEE Trans image process , 2006.

[7] M. Dobes, L.Machala, and T. Furst, "Digital Signal Processing", Elsevier Inc, 1677-1686, March, 2010.

[8] R. C. Gonzalez and R. E. Woods , " Digital Image Processing ", Prentice Hall,3rd Edition, 2008.

[9] R. C. Gonzalez "Digital Image Processing", ISBN 0201180758, 9780201180756 ,prentice Hall , 2002.

[10] S. Aoki, A. Takemura, "The Largest Group of Invariance for Markov Bases and Toric ideals", J. Symbolic Computation ,pp. 342–358, 2008.

[11] W. K. Pratt, " Digital Image Processing", A Wiey-Inters-Cience Publication, ISBN: 978-0-471-76777-0, TA 1632.p 7, 4 Edition, 2007.

[12] X. Jiany, D. Cheng, S. Wachenfeld, and K. Rothaus, "Image Processing and Pattern Recognition", Department of Mathematics and Computer Science, University of Muenster, Winter, 2005.

[13] W. J. Gilbert," Modern Algebra With Application ",John Wiley, Inc., New Jersey, U.S.A, 2004.