Modified (LTB) Algorithm For Image Compression By Using Standard MV-Algebra

خوارزمية (LTB) المعدلة لضغط الصور بإستخدام جبر MV القياسي

Assist. prof. Husein Hadi Abbass
University of Kufa \Collage of Education for Girls \
Department of Mathematics
MSC_hussien@yahoo.com

Karrar Dheiaa Mohammed
University of Kufa \ Collage of Mathematics and Computer Science \
Department of Mathematics
Karrar_d_alsabti@yahoo.com

Abstract

In this paper ,we introduce a new method as a modified of the (LTB) algorithm for image compression, by using the standard MV-algebra (A,⊕ , ~ ,0), where A=[0,1] is the unit interval of real numbers with the internal binary operation ⊕ which is defined by x⊕y=min {1, x+y }and the internal unary operation ~ x =1-x, for all x,y belong to A and a constant 0. We prove some mathematical properties of standard MV-algebra to use them as a mathematical support in our modified method, we call this method (CSLTB). A comparison was made between our method and (LTB algorithm , JPEG algorithm) by the PSNR (Peak signal Noise Ratio) and RMSE (Root Mean Square Error).Show that our method has better performance.

الخلاصة:-

قدمنا في هذا البحث طريقة جديدة كتحسين لخوارزمية (LTB) لضغط الصور باستخدام جبر MV القياسي (A , ⊕ , ~ , 0), حيث ان A=[0,1] الفترة الواحدية المغلقة من الأعداد الحقيقية مع العملية الثنائية الداخلية ⊕ والتي تعرف كما يلي x ⊕ y = min { 1 , x + y } والعملية الأحادية الداخلية ~ x =1-x , لكل x و y تنتمي الى A والثابت 0 . وقد برهنا بعض الخواص الرياضية لجبر MV القياسي لاستخدامها كأساس رياضياتي لطريقتنا المعدلة , والتي أطلقنا عليها أسم (CSLTB) . وتمت المقارنة بين طريقتنا وبين كل من (خوارزمية (LTB) وخوارزمية (JPEG)) باستخدام المقاييس الآتية (PSNR) (Peak Signal Noise Ratio) (نسبة قمم الضوضاء للإشارة) وكذلك (RMSE) (Root Mean Square Error) (الجذر التربيعي للخطأ) . لبيان جودة و كفاءة عمل طريقتنا .

Introducion

Image can be defined as a two dimensional light intensity function f(x, y), where x and y denotes spatial co-ordinates and the value of 'f 'at any point is directly proportional to the brightness (gray level) of the image at that point.[19].Interest in image compression dates back more than 35 years. The initial focus of research efforts in this field was on the development of analog methods for reducing size of data representation of an image, the practical application of the theoretic work that began in the 1940s, by C.E Shannon and others [20].

On the other hand Chang devised MV-algebras to study many-valued logics, introduced by Jan Łukasiewicz in 1920. In particular, MV-algebras form the algebraic semantics of Łukasiewicz logic, [6].Given an MV-algebra A, an A-valuation is a homomorphism from the algebra of propositional formulas (in the language consisting of ⊕, ~ , and 0) into A. Formulas mapped to 1 (or ~ 0) for all A-valuations are called A-tautologies. If the standard MV-algebra over [0,1] is employed, the set of all [0,1]-tautologies determines so-called infinite-valued Łukasiewicz logic [21].

In 1965 L.A.Zadeh introduced the notion of fuzzy subset of a set, the theory of fuzzy relations is widely used in many applications [2,3,11,16] and particularly in the field of image processing ([8]-[10],[13] - [15], [18] and [23]). As a matter of fact, fuzzy relations fit the problem of processing the representation of an image as a matrix with the range of its elements previously normalized in [0 , 1], [17,22] .
In 2006, A. D. NOLA and C. Russo provided a mathematical support for some techniques of image processing based on the theory of fuzzy sets.They used the Lukasiewicz Transform by mean of the partition of unity in MV-algebra . The Lukasiewicz Transform and its residual are semimodules homomorphism and their compositions yields the identity. These algebra tools allowed them to define the algorithm for image processing Lukasiewicz Transform Based (LTB) [1] . In this paper ,we give a new method as a modified (LTB) method as we mentioned in the abstract.

1. Some Basic Concepts

In this section,we review some basic definitions and notations of semimodules over semirings and MV- algebra with its proves that we need in our work.

Definition (1.1):- [1]

A semi ring is an algebraic structure (S, +, . , 0, 1), with two internal binary operations, + and . , and two constants $0,1 \in S$ such that

(S1) (S,+, 0) is a commutative monoid,
(S2) (S, . , 1) is a monoid,
(S3) $x . (y + z) = xy + xz$ and $(x + y) . z = xz + yz$ for all $x, y, z \in S$,
(S4) $0x = x0 = 0$ for all $x \in S$.

A semi ring is said to be commutative iff the commutative property holds for the multiplication too.We will consider only non-trivial semi rings, i. e. semi rings such that $0 \neq 1$.

Definition(1.2) :- [1]

Let S be a semiring. A left S-semi module is a commutative monoid $(M, +_M, 0)$ with an external operation, called scalar multiplication, with coefficients in S

$$\bullet : (s,m) \in S \times M \mapsto s \bullet m \in M ,$$

Satisfying, for all $s, s' \in S$ and $m,m' \in M$, the following condition :

(M1) $(ss') \bullet m = s \bullet (s' \bullet m)$;
(M2) $s \bullet (m +_M m') = s \bullet m +_M s \bullet m'$;
(M3) $(s + s') \bullet m = s \bullet m +_M s' \bullet m$;
(M4) $1 \bullet m = m$;
(M5) $s \bullet 0_M = 0_M = 0 \bullet m$.

The definition of a right S-semi module is analogous.
If S and S′ are semi rings and M is both a left S-semi module and a right S′-semi module, M will be called an (S , S′)-bisemimodule iff it satisfies the following additional condition :

(M6) $(s \bullet_S m) \bullet_{S'} s' = s \bullet_S (m \bullet_{S'} s')$,for all $s \in S$, $s' \in S'$ and $m \in M$,

Where •s and •s′ mean the external products with scalars in S and in S′ respectively. In particular, if S is commutative, any left or right S –semi module is an (S,S)-bisemimodule and we will call it , shortly , an S-bisemimodule . Note that we will always omit the "•" symbol
Throughout this section S will denote the semi ring (S, +, . , 0, 1), and $M = (M,+_M ,0)$ will be a left semimodule over S. Obviously all the following definitions and results hold both for right and left S-semimodules.

Definition(1.3):-[6]

An algebraic structure $A=(A ,\oplus, . , \sim ,0)$, is said to be an MV -algebra iff it satisfies the following equations:

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$;
2. $x \oplus y = y \oplus x$;
3. $x \oplus 0 = x$;

4. $\sim\sim x = x$;
5. $x \oplus \sim 0 = \sim 0$
6. $\sim(\sim x \oplus y) \oplus y = \sim(\sim y \oplus x) \oplus x$.

Remark(1.4):-[1]
On every MV – algebra it is possible to define another constant and two further operations as follows $1= \sim 0$, $x \odot y = \sim(\sim x \oplus \sim y)$, $x \ominus y = x \odot \sim y$
The following properties follow directly from these definitions
(1) $\sim 1= 0$
(2) $x \oplus y = \sim(\sim x \odot \sim y)$
(3) $x \oplus 1 = 1$
(4) $(x \ominus y) \oplus y = (y \ominus x) \oplus x$
(5) $x \oplus \sim x = 1$
In follows we will often denote an MV-algebra by $(A, \oplus, \odot, \sim, 0, 1)$

Definition (1.5):- [6]
Let A be an MV-algebra ,the binary relation $\leq$ is defined on A by
$x \leq y$ iff $\sim x \oplus y =1$. for all $x, y \in A$.

Remark (1.6):-[5]
The relation $\leq$ which is defined in the definition (1.5) is a partial order ,called natural order of A. This relation also determines a lattice structure with 0,1 respectively infimum and supremum elements , $\vee$ and $\wedge$ defined as follows:-
$x \vee y = (x \odot \sim y) \oplus y = (x \ominus y) \oplus y$
$x \wedge y = \sim(\sim x \vee \sim y) = x \odot (\sim x \oplus y)$
Definition (1.7):- [21]
The MV – algebra is called complete if it is complete as a lattice ,and that is an MV- chain if it is totally ordered by its natural order .

Remark (1.8):- [4]
Let $A=(A, \oplus, \odot, \sim, 0,1)$ be a complete MV – algebra , X be a non empty set , and A^X is the set of all functions from A to X . Then $A^X=(A^X, \oplus, \odot, \sim, 0,1)$ is a complete MV-algebra with operations defined pointwise , for any MV- algebra A it is possible define semirings $L^{\wedge} =(A, \wedge, \oplus, 0, 1)$ and
$L^{\vee} =(A, \vee, \oplus, 0, 1)$,called semirings reduct of A.
Moreover the monoid $M=(A^X, \vee, 0)$ is a bisemimodule over both $L^{\wedge}$ and $L^{\vee}$ with scalar multiplication $af = \sim a \odot f$ and $af = a \odot f$, for all $a \in A$ and $f \in A^x$. analogously $(A^n, \vee, 0)$ is a bisemimodule over $L^{\wedge}$ and $L^{\vee}$ for any $n \in N$. M is called a MV- $L^{\wedge}$ - Semimodule (respectively : MV- $L^{\vee}$ semimodule) over A.
If A is an MV- chain ,the MV – semimodules over it will be called Lukasiewicz $L^{\wedge}$- semimodule and Lukasiewicz $L^{\vee}$ - semimodule respectively .

Definition (1.9):- [2]
Let A be an MV- algebra a finite sequence of element of A ,$(a_0, \ldots, a_{n-1})$ is a partition of unity if $a_0 + \ldots + a_{n-1} =1$.

Proposition (1.10):-[1]
Let $n \in N$,$n > k$ and $k \in Z$ .Then $\{p_0, p_1, \ldots, p_{n-1}\}$ is a partition of unity in the MV- algebra $[0,1]^{[0,1]}$ where

$$P_0(x) = \begin{cases} -(n-1)x+1 & \text{if} \quad 0 \le x \le \frac{1}{n-1} \\ 0 & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$P_{n-1}(x) = \begin{cases} (n-1)x-(n-2) & \text{if} \quad \frac{n-2}{n-1} \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

and

$$P_k(x) = \begin{cases} (n-1)x-(k-1) & \text{if} \quad \frac{k-1}{n-1} \le x \le \frac{k}{n-1} \\ -(n-1)x+k+1 & \text{if} \quad \frac{k}{n-1} \le x \le \frac{k+1}{n-1} \\ 0 & \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

for $k = 1,2,\ldots,n-2$

Definition (1.11):- [7,12]
Let S be a semiring and let M=(M , + , 0) be an S – semimodule ,furthermore , let X be a non empty set and n $\in$ N . A semimodule homomorphism
$h_n : M^x \to M^n$ is called transform of order n , if there exists a homommorphism $\lambda_n : M^n \to M^x$ having the following properitys

(T1) $h_n \circ \lambda_n \circ h_n = h_n$

(T2) $\lambda_n \circ h_n \circ \lambda_n = \lambda_n$

Definition (1.12):- [1]
Consider the MV- algebra A = $([0,1], \oplus, \odot, \sim, 0,1)$ and the set X = [0,1] .The operator
$H_n : [0,1]^{[0,1]} \to [0,1]^n$ defined by

$$H_n : f \in [0,1]^{[0,1]} \longmapsto \left(\bigvee_{x\in[0,1]} f(x) \odot p_k(x) \right)_{k=0}^{n-1} \in [0,1]^n,$$

is called Lukasiewicz transform, where $P_0, \ldots, P_{n-1}$ is the partition of unity defined by proposition (1.10).

Theorem (1.13):- [1]
H_n is a Lukasiewicz $\hat{L}$ - semimodule homomorphism from $([0,1]^{[0,1]}, \vee, 0)$ to $([0,1]^n, \vee, 0)$.

Definition (1.14):- [7]
Let (X, $\le$) and (Y, $\le$) be two ordered sets. A map h: x $\to$ y is said to be residuated if it is isotone and, for all y $\in$ Y , the set {x$\in$ X : h(x) $\le$ y} admits the greatest element, denoted by $h^{\#}$ (y). The map
$h^{\#}$: Y $\to$ X is called the residual ,or the residual map, of h.

Theorem (1.15):- [1]
The map H_n is residuated and Λ_n is its residual map. Moreover $H_n \circ \Lambda_n = I_{[0,1]^n}$, where Λ_n is the Lukasiewicz inverse transform defined by

$$\Lambda_n : v = (v_0, \ldots, v_{n-1}) \in [0,1]^n \to \sim \left(\bigvee_{k=0}^{n-1} (\sim v_k \odot p_k) \right) \in [0,1]^{[0,1]}$$

Definition (1.16):- [16]

Let X be a non-empty set and I be the closed interval [0, 1] of the real line (real numbers). A ***fuzzy subset A in X (a fuzzy subset of X)*** is a function from X in to I.

Remark (1.17):-

1) According to the theorem (1.15) H_n is transform of order n from the semimodule $([0,1]^{[0,1]} , \vee , 0)$ to $([0,1]^n , \vee , 0)$.
2) Every element in $[0,1]^{[0,1]}$ is a fuzzy subset of [0,1] .

2. The Lukasiewicz Transform Based (LTB) Algorithm for ImageProcessing

In [1] the authers introduced (LTB) algorithm , and we describe it brifly as follow :-

Step 1 :- Fuzzyfy the image matrix of the dimension (m x n) into a fuzzy matrix $X=(x_{ij})$ of dimension (m x n) (i.e each image matrix must be seen as a [0,1] valued function defined on (a subset) of [0,1]) . Every gray image is treated as afuzzy matrix and the (RGB) colour image is treated as three fuzzy matrices.

Step 2 :- Rewrite the fuzzy image matrix $X = (x_{ij})$ as a (mn) vector $(x'_0 ,\dots,x'_{mn-1})$ by setting for all (k = 0 , … ,mn-1) , $x'_k = x_{q(k,n)\ r(k,n)}$ where q(k,n) and r(k,n) are , respectively , quotient and remainder of the Euclidean division k/n .

Step 3 :- Set $D_X = \{ \frac{k}{mn-1} : k = 0 , \dots , mn-1\} \subset [0,1]$

and define the function

$$f : \frac{k}{mn-1} \in D_X \rightarrow x'_k \in [0,1]$$

Step 4 :- Find the partition of unity $(p_0,\dots,p_{n-1})$ defined by proposition (1.10) .

Step 5 :- Apply the Lukasiewicz transform $H_n(fx)$ to compress the image .

Step 6 :- To decompress the image apply $\Lambda_n(H_n(fx))$ the residual of the Lukasiewicz transform .

Step 7 :- Rewrite $\Lambda_n(H_n(fx))$ as an ordinary matrix.

3. The Main Result

The techniqe of (LTB) image processing based on the MV- algebra ([0,1], $\oplus$, ~ ,0) (i.e the LTB algorithm depend on the definition of the binary operation $\oplus$ and the unary operation ~ and the constant 0).

In this section we give a new method as a modefied of the (LTB) for image processing based on the standard MV – algebra ([0,1], $\oplus$, ~ ,0) endowed with binary operation $\oplus$ defined by $x \oplus y = \min \{ 1 , x + y \}$ and the unary operation $\sim x = 1-x$, for all $x,y \in A$ and the constant 0 .

Now , we prove some mathematical results about the standard MV- algebra as a mathematical support to our algorithm.

Proposition (3.1) :-

If $([0,1] , \oplus , \sim , 0)$ is a standard MV- algebra and $x, y \in [0,1]$ then

(i) $x \odot y = \max \{ 0, x + y - 1\}$

(ii) $x \ominus y = \max \{ 0, x-y \}$

(iii) $x \vee y = \max \{ x , y \}$

(iv) $x \wedge y = \min \{ x , y \}$

Proof:-

(i) From the definition of the binary operation $\odot$ in remark (1.4) and since ([0,1] , $\oplus$, ~ , 0) is a standard MV- algebra where

$$x \odot y = \sim (\sim x \oplus \sim y) = 1-(1 - x \oplus 1 - y) = 1- \min \{ 1 , 1 - x + 1- y \}$$
$$= 1- \{ \min \{1 , 2 - x - y \}\}= \max \{ 1-1 , 1-\{2 - x - y \}\} = \max \{ 0 , x + y - 1 \}.$$

(ii) By the definition of the binary operation $\ominus$ in remark (1.4) and since ([0,1] , $\oplus$, ~ , 0) is a standard MV – algebra and by (i) we have

$x \ominus y = x \odot \sim y = \max \{0, x + \sim y -1\}$ [from (i)]

$=\max \{ 0 , x + 1- y -1 \} = \max \{ 0 , x - y \}$.

(iii) From the definition of $\vee$ in remark (1.6) , ([0,1] , $\oplus$, ~ , 0) is a standard MV- algebra and by (i) we get

$x \vee y = (x \odot \sim y) \oplus y = \min \{1,(x \odot \sim y) + y \}$

$= \min \{ 1 , \max \{ 0 , x + \sim y -1\} +y\}$

$= \min \{ 1 , \max\{ 0 , x + 1 - y - 1 \} + y \}$

$= \min \{ 1 , \max \{ 0 , x - y \} + y \}$

$$= \begin{Bmatrix} x & if & x \geq y \\ y & if & x < y \end{Bmatrix}$$

$= \max \{ x , y \} \Longrightarrow x \vee y =\max \{ x , y \}$.

(iv) By definition of $\wedge$ in remark (1.6) , ([0,1] , $\oplus$, ~ , 0) is a standard MV – algebra and from(iii), we have

$x \wedge y = \sim (\sim x \vee \sim y) = 1 - (1 - x \vee 1 - y) = 1- \max \{1-x ,1-y\} = \min \{1- (1-x), 1- (1-y) ,$

$= \min \{ x , y \}$ ■.

Proposition (3.2) :-

Let ([0,1], $\oplus$, $\odot$, ~ , 0 , 1) be the standard MV- algebra , $f \in [0,1]^{[0,1]}$ and P_0 , P_1 ,…,P_{n-1} be the partition of unity defined by proposition (1.10) .Then there exists $x' \in [0,1]$ such that

$P_k(x') + f(x') \geq 1$, for all k = 0,1,2,…,n-1.

Proof:-

If k=0 and $x' = 0 \in [0,1]$, then $P_0(0) =1 \Rightarrow P_k (x') +f(x') \geq 1$

[Since f is a function from [0,1] to [0,1] (fuzzy subset of [0,1]) Therfore $0 \leq f(x) \leq 1$, $\forall$ $x \in [0,1]$]

$\Rightarrow$ If k = n-1 and $x' = 1 \in [0,1]$, then

$P_{n-1}(x') = P_{n-1}(1) = (n-1) 1 - (n-2) = 1$

$\Rightarrow P_{n-1} (x') + f(x') \geq 1$ [since $0 \leq f(x') \leq 1$]

If k = 2,…,n-2 and $x' = \dfrac{k}{n-1}$, then

$P_k(x') = P_k(\dfrac{k}{n-1}) = (n-1) \dfrac{k}{n-1} - (k-1) = 1$

$\Rightarrow P_k (x') + f(x') \geq 1$ ■.

Theorem (3.3):-

Let ([0,1], $\oplus$, $\odot$, ~ , 0 , 1) be the standard MV-algebra and $P_0,P_1,\ldots,P_{n-1}$ be the partation of unity defined by proposition (1.10) . Then $H_n(f) = (\sup_{x\in[0,1]} \{f(x) +P_k(x) -1\})_{k=0}^{n-1}$, $\forall f \in [0,1]^{[0,1]}$.

Proof:-

From the definition of Lukasiewicz Transform (definition (1.12)) , we have

$$H_n(f) = \Big(\bigvee_{x\in[0,1]} f(x) \odot P_k(x) \Big)_{k=o}^{n-1}$$

By proposition (3.1) (i) and (iii) we have

$$\bigvee_{x\in[0,1]} f(x) \odot P_k(x) = \sup_{x\in[0,1]} \{ \max \{0 , f(x) + P_k(x) -1\}\}$$

from proposition (3.2) there exists $x' \in [0,1]$ such that

$f(x') + P_k(x') \geq 1 \Rightarrow f(x') + P_k(x') - 1 \geq 0$

$$\Rightarrow \sup_{x\in[0,1]} \{ \max\{0, f(x) + P_k(x) - 1\}\} = \sup_{x\in[0,1]} \{f(x) + P_k(x) - 1\}$$

$$H_n(f) = \left(\sup_{x\in[0,1]} \{f(x) + P_k(x) - 1\} \right)_{k=0}^{n-1}$$

Definition (3.4):-

We call the Lukasiewicz Transform in theorem (3.3) the standard Lukasiewicz Transform , denoted by SH_n .

corollary (3.5):-

Let ([0,1] , $\oplus$, Θ , ~ , 0 , 1) be the standard MV-algebra and $P_0, P_1, \ldots, P_{n-1}$ be the partation of unity defined by propsition (1.10) . Then $SH_n(\sim f) = SH_n(1- f) = \left(\sup_{x\in[0,1]} \{P_k(x) - f(x)\} \right)_{k=0}^{n-1}$, $\forall\ f \in [0,1]$

proof :-

Since ([0,1] , $\oplus$, Θ , ~ , 0 , 1) is the standard MV-algebra ,then ~ f = 1-f .

From theorem (3.3), we have $SH_n((\sim f) = SH_n(1- f) = \left(\sup_{x\in[0,1]} \{1- f(x) + P_k(x) - 1\} \right)_{k=0}^{n-1}$

$$= \left(\sup_{x\in[0,1]} \{P_k(x) - f(x)\} \right)_{k=o}^{n-1} \blacksquare.$$

Theorem (3.6):-

Let ([0,1] , $\oplus$, Θ , ~ , 0 , 1) be the standard MV – algebra , $P_0, P_1, \ldots, P_{n-1}$ be the partation of unity defined by proposition (1.10) , $f \in [0,1]^{[0,1]}$, $SH_n(f) = (v_k)_{k=0}^{n-1}$ and $S\Lambda_n$ be the residual map of the standard Lukasiewicz transformed SH_n .

Then $S\Lambda_n((v_k)_{k=0}^{n-1}) = 1- \sup_{k=0}^{n-1} \{ \max\{ 0 , P_k(x) - v_k\}\}$ for all $x \in [0,1]$.

Proof:-

From theorem (1.15) , we have

$$S\Lambda_n((v_k)_{k=o}^{n-1}) = \sim (\bigvee_{k=0}^{n-1} (\sim v_k \,\Theta\, P_k))$$

Now, let $x \in [0,1]$.

By proposition (3.1) ,we have

$$\sim (\bigvee_{k=0}^{n-1} (\sim v_k \,\Theta\, P_k)) (x) = 1- (\bigvee_{k=0}^{n-1} (\sim v_k \,\Theta\, P_k(x)))$$

$$= 1- \sup_{k=0}^{n-1} \{ \max\{ 0 , 1 - v_k + P_k(x) - 1\}\}$$

$$= 1- \sup_{k=0}^{n-1} \{ \max\{ 0 , P_k(x) - v_k \}\} \blacksquare.$$

corollary (3.7):-

Let ([0,1] , ⊕, ʘ , ~ , 0 , 1) be the standard MV – algebra , $P_0,P_1,\ldots,P_{n-1}$ be the partation of unity defined by proposition (1.10) , $f \in [0,1]^{[0,1]}$ then $S\Lambda_n(SH_n(\sim f)) = S\Lambda_n(SH_n(1-f))$

$$= \sup_{k=0}^{n-1} \{ \max \{ 0 , P_k(x) - v_k \}$$

Proof:-

$S\Lambda_n(SH_n(1-f)) = S\Lambda_n(SH_n(1) - SH_n(f))$

$= S\Lambda_n(SH_n(1)) - S\Lambda_n(SH_n(f))$

$= 1- (1- \sup_{k=0}^{n-1} \{ \max \{ 0 , P_k(x) – v_k)$ (By Theorem (3.6))

$= \sup_{k=0}^{n-1} \{ \max \{ 0 , P_k(x) - v_k \}$ ■.

We call our algorithm " Complement of fuzzy image vector Standard Lukasiewicz Transform Based", denoted by (CSLTB) . Since the standard Lukasiewicz Transform SH_n which we define in theorem (3.3)and defintion (3.4) to compress the fuzzy image complement vector ~ f =(1-f) .
Now we describe the (CSLTB) algorithm .

Step 1 :-

Fzzyfy the image matrix by dividing each element of image matrix by 256 , because the gray level is ranging between (0 – 255). Each gray image is treated a fuzzy matrix and the (RGB) colour image as three fuzzy matrix .

Step 2 :-

Rewrite the fuzzy image matrix $X=(x_{ij})$ as (mn) vector as in the (step 2) in (LTB) algorithm.

Step 3 :-

Define the function f as in the (step 3) in (LTB) algorithm .

Step 4 :-

Find $P_0,\ldots,P_{n-1}$ the partition of unity defined by proposition (1.10).

Step 5 :-

Apply the standard Lukasiewicz transformed $SH_n(1-f)$ by using corollary (3.5) to compress the image .

Step 6 :-

To decompress the image apply the residual map of the standard Lukasiewicz Transform $S\Lambda_n(SH_n(1-f))$ by using corollary (3.7).

Step 7 :-

Rewrite $S\Lambda_n(SH_n(1-f))$ as ordinary (m x n) matrix by multiplying each element by 256 to get the ordinary decompression image matrix .

Example (3.8):-

In this example we Take a square block of (4 x 4) to explain (CSLTB) algorithm (our method) as in the following steps:-

$$M = \begin{bmatrix} 110 & 110 & 118 & 118 \\ 108 & 111 & 125 & 122 \\ 106 & 119 & 129 & 127 \\ 110 & 126 & 130 & 133 \end{bmatrix}$$

Step 1:-

Now we fuzzfying the matrix M by dividing each element of M by 256 to get the following matrix:-

$$X = \begin{bmatrix} 0.42 & 0.42 & 0.46 & 0.46 \\ 0.42 & 0.43 & 0.48 & 0.47 \\ 0.41 & 0.46 & 0.5 & 0.49 \\ 0.42 & 0.49 & 0.5 & 0.51 \end{bmatrix}$$

Step 2 :-

Now we rewrite the fuzzy matrix X as a matrix of one row of dimension (1 x 16) or as the vector of 16 components as follow :-

[0.42 0.42 0.46 0.46 0.42 0.43 0.48 0.47 0.41 0.46 0.5 0.49 0.42 0.49 0.5 0.51]

Step 3 :-

Set the following:-

$$D_X=\left\{\frac{k}{(mxn)-1} : k = 0,1,\ldots,(mxn)-1\right\}$$

$$D_X=\left\{\frac{k}{(4x4)-1} : k = 0,1,\ldots,(4x4)-1\right\}$$

D_X={ 0 , 0.06 , 0.13 , 0.2 , 0.26 , 0.33 , 0.4 , 0.46 , 0.53 , 0.6 , 0.66 , 0.73 , 0.8 , 0.86 , 0.93 , 1 }

and define

$$f = \frac{k}{(mxn)-1} \in D_X \longrightarrow X`_k$$

f(0) = 0.42 , f(0.06) = 0.42 , f(0.13) =0.46 ,… ,f(1) = 0.51

Step 4 :-

Now we find P_0 , P_1 , P_2 , P_3 by using proposition (1.10)

$$P_0(x) = \begin{cases} -3x+1 & \text{if} \quad 0 \le x \le \frac{1}{3} \\ 0 & \text{otherwise} \end{cases}$$

$$P_1(x) = \begin{cases} 3x & \text{if} \quad 0 \le x \le \frac{1}{3} \\ -3x+2 & \text{if} \quad \frac{1}{3} \le x \le \frac{2}{3} \\ 0 & \text{otherwise} \end{cases}$$

$$P_2(x) = \begin{cases} 3x-1 & \text{if} \quad \frac{1}{3} \le x \le \frac{2}{3} \\ -3x+3 & \text{if} \quad \frac{2}{3} \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

$$P_3(x) = \begin{cases} 3x-2 & \text{if} \quad \frac{2}{3} \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 5 :-

Compress the complement of image vector (1- f) by using SH_n (1-f) (corollary (3.5))

$$SH_n(1-f) = \left(\sup_{x\in[0,1]} (P_0(x) - f(x)),\ \sup_{x\in[0,1]} (P_1(x) - f(x)),\ \sup_{x\in[0,1]} (P_2(x) - f(x)),\ \sup_{x\in[0,1]} (P_3(x) - f(x)) \right)$$

x	0	0.06	0.13	0.2	0.26	0.33
$P_0(x)$	1	0.82	0.61	0.4	0.22	0.01
f(x)	0.42	0.42	0.46	0.46	0.42	0.43
$P_0(x) - f(x)$	0.58	0.4	0.15	- 0.06	- 0.2	- 0.42

Table (1) How we find $P_0(x) - f(x)$ to get v_0

$$\sup_{x\in[0,1]} (P_0(x) - f(x)) = \sup \{ 0.58 , 0.4 , 0.15 , - 0.06 , - 0.2 , - 0.42 \} = 0.58 = v_0$$

x	0	0.06	0.13	0.2	0.26	0.33	0.4	0.46	0.53	0.6	0.66
$P_1(x)$	0	0.18	0.39	0.6	0.78	0.99	0.8	0.62	0.41	0.2	0.02
f(x)	0.42	0.42	0.46	0.46	0.42	0.43	0.48	0.47	0.41	0.46	0.5
$P_1(x) - f(x)$	- 0.42	-0.24	-0.07	0.14	0.36	0.56	0.32	0.15	0	- 0.26	-0.48

Table (2) How we find $P_1(x) - f(x)$ to get v_1

$$\sup_{x\in[0,1]} (P_1(x) - f(x)) = \sup \{ - 0.42 , - 0.24 , - 0.07 , 0.14 , 0.36 , 0.56 , 0.32 , 0.15 , 0 , - 0.26 ,- 0.48\}$$

$v_1 = 0.56$

x	0.33	0.4	0.46	0.53	0.6	0.66	0.73	0.8	0.86	0.93	1
$P_2(x)$	-0.01	0.2	0.38	0.59	0.8	0.98	0.81	0.6	0.42	0.21	0
f(x)	0.43	0.48	0.47	0.41	0.46	0.5	0.49	0.42	0.49	0.5	0.51
$P_2(x) - f(x)$	-0.44	-0.28	-0.09	0.18	0.34	0.48	0.32	0.18	-0.07	-0.29	-0.51

Table (3) How we find $P_2(x) - f(x)$ to get v_2

$$\sup_{x\in[0,1]} (P_2(x) - f(x)) = \sup\{-0.44 , -0.28 , -0.09 , 0.18 , 0.34 , 0.48 , 0.32 , 0.18 , - 0.07 , - 0.29 , - 0.51\}$$

$v_2 = 0.48$

x	0.66	0.73	0.8	0.86	0.93	1
$P_3(x)$	-0.02	0.19	0.4	0.58	0.79	1
f(x)	0.5	0.49	0.42	0.49	0.5	0.51
$P_3(x) - f(x)$	-0.52	-0.3	-0.02	0.09	0.29	0.49

Table (4) How we find $P_3(x) - f(x)$ to get v_3

$$\sup_{x\in[0,1]} (P_3(x) - f(x)) = \sup \{-0.52 , -0.3 , -0.02 , 0.09 , 0.29 , 0.49 \} = 0.49 = v3$$

Then $SH_4 (\sim f) = (v_0,v_1,v_2,v_3) = (0.58,0.56,0.48,0.49)$

In this step we use SH_4 to compress the (4x4) fuzzy matrix to (2x2) matrix or we compress the fuzzy image vector of (16) components to the vector of (4) components .

Step 6 :-

To decompress the compression vector , we apply the residual map of standard Lukasiewicz transform by using corollary (3.7)as follow :-

$$S\Lambda_4(SH_4(1-f)) = \sup_{k=0}^{3}\{\max\{0, P_k(x) - v_k\}$$

Now we complete the above example as follow :-

X	P_0-V_0 P_0-0.58	Max{0,P_0-v_0}	P_1-V_1 P_1-0.56	Max{0,P_1-v_1}	P_2-V_2 P_2-0.48	Max{0,P_2-v_2}	P_3-V_3 P_3-0.49	Max{0,P_3-v_3}	$\sup_{k=0}^{3}$ (P_k-V_k)
0	0.42	0.42	-0.56	0	-0.48	0	-0.49	0	0.42
0.06	0.24	0.24	-0.38	0	-0.48	0	-0.49	0	0.24
0.13	0.03	0.03	-0.17	0	-0.48	0	-0.49	0	0.03
0.2	-0.18	0	0.04	0.04	-0.48	0	-0.49	0	0.04
0.26	-0.36	0	0.22	0.22	-0.48	0	-0.49	0	0.22
0.33	-0.57	0	0.43	0.43	-0.49	0	-0.49	0	0.43
0.4	-0.58	0	0.24	0.24	-0.28	0	-0.49	0	0.24
0.46	-0.58	0	0.06	0.06	-0.1	0	-0.49	0	0.06
0.53	-0.58	0	-0.15	0	0.11	0.11	-0.49	0	0.11
0.6	-0.58	0	-0.36	0	0.32	0.32	-0.49	0	0.32
0.66	-0.58	0	-0.54	0	0.5	0.5	-0.51	0	0.5
0.73	-0.58	0	-0.56	0	0.33	0.33	-0.3	0	0.33
0.8	-0.58	0	-0.56	0	0.12	0.12	-0.09	0	0.12
0.86	-0.58	0	-0.56	0	-0.06	0	0.09	0.09	0.09
0.93	-0.58	0	-0.56	0	-0.27	0	0.3	0.3	0.3
1	-0.58	0	-0.56	0	-0.48	0	0.51	0.51	0.51

Table (5) How we apply $S\Lambda_n$ to find the decompression matrix

$S\Lambda_4(SH_4(1-f(x)))$ = [0.42,0.24,0.03,0.04,0.22,0.43,0.24,0.06,0.11,0.32,0.5,0.33,0.12,0.09,0.3,0.51]

Now we rewrite this vector as ordinary vector such that we multiply each element in this vector by 256 to get

The decompress image vector = [108 , 61 , 8 , 10 , 56 , 110 , 61 , 15 , 28 , 82 , 128 , 85 , 31 , 23 , 77 , 131]

$$\text{Decompression ordinary image matrix} \quad DM = \begin{bmatrix} 108 & 61 & 8 & 10 \\ 56 & 110 & 61 & 15 \\ 28 & 82 & 128 & 85 \\ 31 & 23 & 77 & 131 \end{bmatrix}$$

Original Image	Our Method, 2x2, ρ = 0.5	LTB method, 2x2, ρ = 0.5	JPEG method,2x2, ρ = 0.5
			
	Rmse = 1.2162 Psnr = 46.4309	Rmse = 58.4469 Psnr = 12.7956	Rmse = 2.4985 Psnr = 40.1773
	Our Method, 4x4, ρ = 0.25	LTB method, 4x4, ρ = 0.25	JPEG method, 4x4, ρ = 0.25
			
	Rmse = 6.0652 Psnr = 32.4739	Rmse = 68.3310 Psnr = 11.4384	Rmse = 6.1120 Psnr = 32.4071

Table (6) Comparison between our method image ,(LTB algorithm and JPEG algorithm) image, (gray image)

Fig (1) Bridge original image
(360 x 360)

Fig (2) our method (8x8)
(360 x 360)

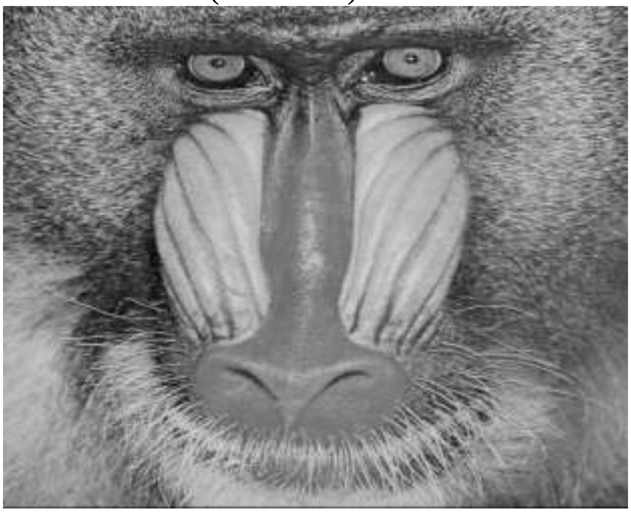


Fig (5) Mandrill original image
(512 x 512)

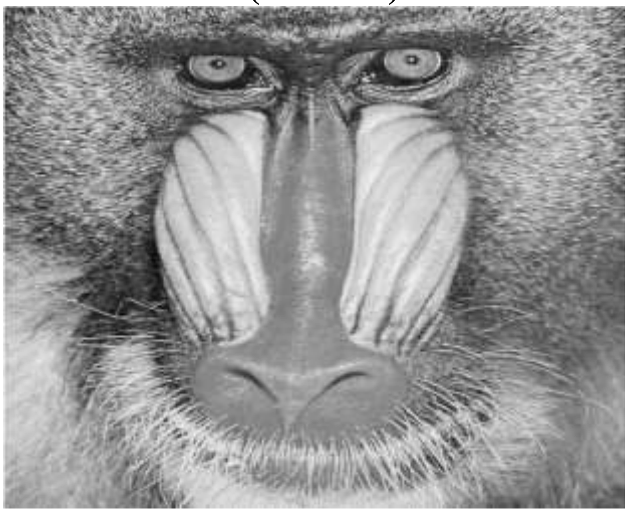


Fig (6) our method (8x8)
(512 x 512)

Fig (5) Lena original image
(512 x 512)

Fig (6) our method (8x8)
(512 x 512)

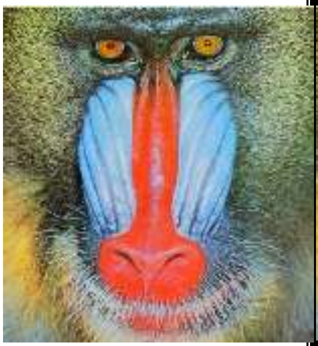
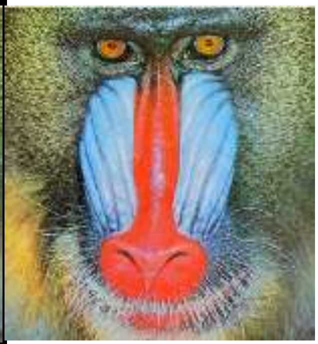
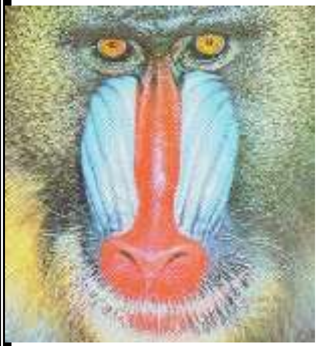
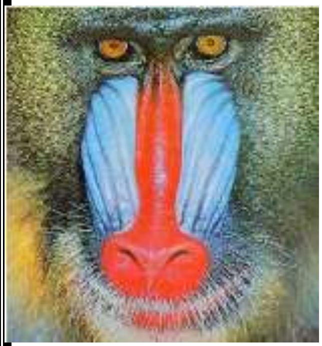
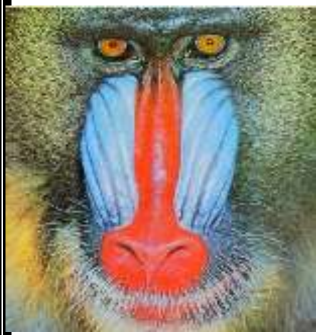
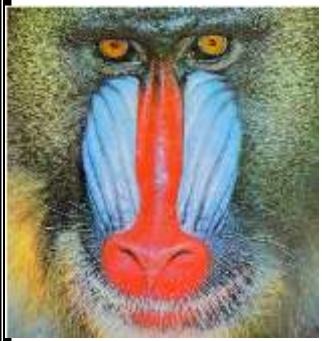
Original Image	**Our Method, 2x2** ρ **= 0.5**	**LTB method, 2x2** ρ **= 0.5**	**JPEG method, 2x2** ρ **= 0.5**
			
	Rmse = 1.6650 Psnr = 43.7023	Rmse = 55.8967 Psnr = 13.1831	Rmse = 6.4739 Psnr = 31.9075
	Our Method, 4x4, ρ **= 0.25**	**LTB method, 4x4,** ρ **= 0.25**	**JPEG method, 4x4** ρ **= 0.25**
			
	Rmse = 6.7813 Psnr = 31.5045	Rmse = 65.7072 Psnr = 11.7785	Rmse = 7.1230 Psnr = 31.0775

Table (7) Comparison between our method image ,(LTB algorithm and JPEG algorithm) image, (color image)

Fig (1) original image
(360 x 360)

Fig (2) our method (360 x 360),
(8x8) blocks (ρ =0.39)

Fig (3) (LTB) algorithm (360 x 360),
(8x8) blocks , (ρ =0.39)

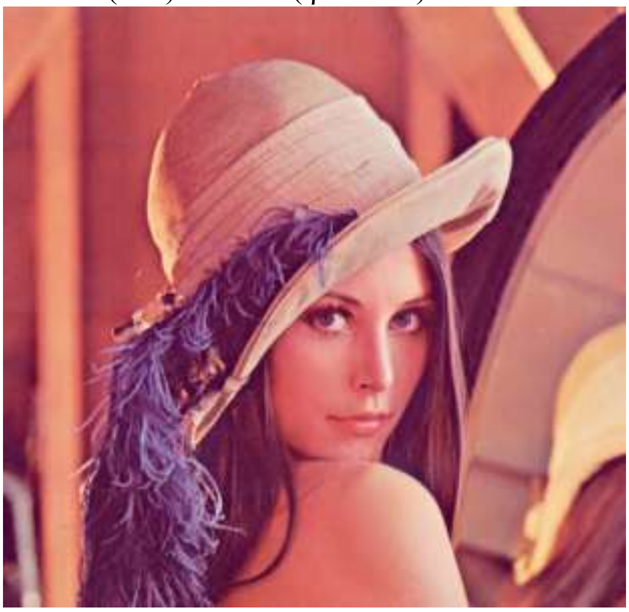


Fig (1) Lena original image (512 x 512)

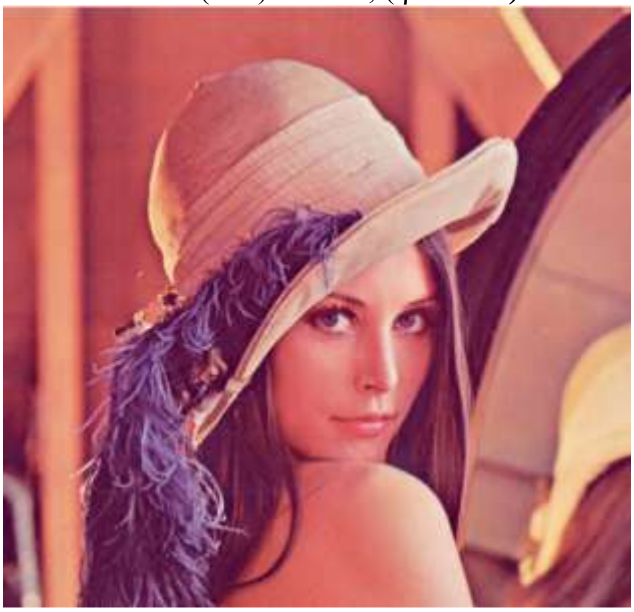


Fig (2) our method (512 x 512)
(8x8) blocks , (ρ =0.39)

IMAGE	JPEG RMSE	LTB RMSE	OUR METHOD RMSE	JPEG PSNR	LTB PSNR	OUR METHOD PSNR
Bridge Gray	3.6168	56.3420	2.7965	36.9643	13.1141	39.1984
Mandril l RGB	6.5917	53.5997	2.2150	31.7509	13.5476	41.2232

Table (8) Comparison between our method and (LTB algorithm, JPEG algorithm)

Conclusions:-

We deduce from this paper which contains the new method to compress images (gray and color) as a modified of (LTB) algorithm we created it to get a result better than the (LTB) algorithm and we get our goal from this method and so, we observe that the result of our methods which we call it (NSLTB) is batter than JPEG and (LTB) algorithms, results in (gray and color) images by objective and subjective as we viewed in the above tables and figures .

Since the JPEG algorithm still the standard algorithm for image compression then the important thing in (CSLTB) is that its result batter than the JPEG algorithm result and so we observe that in the gray image the difference between (CSLTB) and JPEG algorithm results is small but still (CSLTB) result better than JPEG algorithm result and in color image we saw the large difference between them and we think the reason of this difference between gray and color images "of (CSLTB) results and JPEG algorithm results" because there exist one matrix corresponding the gray image and there are three matrix for color image one matrix for each color of (RGB) i.e.(red , green , blue) respectively, such that we compress one matrix for gray image and compress three matrix for color image.

Refrence

[1] A. D. Nola and C. Russo, "Lukasiewicz Transform Based Algorithm for Image Processing", IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006.

[2] A. D. Nola, A. Lettieri, "On Normal Forms in Lukasiewicz Logic",Arch. Math. Logic, 43, no. 6, 795–823, 2004.

[3] A. D. Nola, S. Sessa, W. Pedrycz, E. Sanchez, , "Fuzzy relation equations and their applications to knowledge engineering", Kluwer, Dordrecht, 1989.

[4] A. D. Nola, B. Gerla, "Algebras of Lukasiewicz's logic and their semiring reducts", Idempotent mathematics and mathematical physics, 131{144, Contemp. Math., 377, Amer. Math. Soc., Providence,RI,2005.

[5] B.A. Davey,; H. A. Priestley, "Introduction to Lattices and Order", Cambridge University Press, ISBN 978-0-521-78451-1(2002).

[6] C. C. Chang, "Algebraic analysis of many-valued logics," Transactions of the American Mathematical Society 88: 476–490, (1958).

[7] G.Cohen, S.Gaubert, J.-P. Quadrat," Hahn-Banach Separation Theorem for Max-Plus Semimodules", pp. 325{334 in: J.-L. Menaldi, E. Rofman, A. Sulem (Eds.), Optimal Control and Partial Di erential Equations, Proceedings of Conference in honour of Professor Alain Bensoussan's 60th birthday, Dec. 4-5 2000, Paris, published by IOS Press, Amsterdam, 2000.

[8]H.Nobuhara,W. Pedrycz, K. Hirota," A digital watermarking algorithm using image compression method based on fuzzy relational equations", IEEE Inter. Conference Fuzzy Syst., CD-ROM In: Proc Hawaii, 2002.

[9] H. Nobuhara, W. Pedrycz, K. Hirota, "Fast solving method of fuzzy relational equations and its application to image compression/reconstruction", IEEE Trans. Fuzzy Syst., 8 (3), 325-334, 2000.

[10] H. Nobuhara, Y. Takama, K. Hirota, "Image compression/reconstruction based on various types of fuzzy relational equations", Trans. Inst. Electrical Eng. Japan, 121-C(6), 1102- 1113, 2001.

[11] I. Perfilieva, Fuzzy Transforms: "Theory and Applications, Fuzzy Sets and Systems", 157, 993-1023, 2006.

[12] J. S.Golan, "The theory of semirings with applications in mathematics and theoretical computer science", Longman Scientific and Technical, Harlow and Wiley, New York, 1992.

[13] K. Hirota, H. Nobuhara, W. Pedrycz, "Relational image compression: optimizations through the design of fuzzy coders and YUV color space", Soft Computing, 9, 471{479, 2005.

[14] K. Hirota, K. Kawamoto, H. Nobuhara, S.I. Yoshida, "On a lossy image compression /reconstruction method based on fuzzy relational equations", Iran. J. Fuzzy Syst., 1, 33{42, 2004.

[15] K. Hirota, W. Pedrycz ,"Fuzzy relational compression", IEEE Trans. Syst. Man Cyber.{Part B, 29 (3), 407{415, 1999.

[16] L.A.Zadah " fuzzy sets " , information and control vol 8/pp 338 - 353/1965 .

[17] Madasu Hanmandlu Member IEEE and Devendra Jha "An Optimal Fuzzy System for color image Enhancement".

[18] M. Nachtegael, D.Van der Weken, D. Van De Ville, E.E. Kerre, W. Philips, I. Lemahieu, " An overview of fuzzy and fuzzy-classical lters for noise reduction", Proceedings of the 10th FUZZ-IEEE Conference, 3-6,2001.

[19] Nitin Kumar Kansal "Fuzzy Techniques For Image Enhancement" Computer Science and Engineering Department Thapar University Patiala-147004 , Master Thesis ,June 2010

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

[21] R.L.O. Cignoli, I.M.L. D'Ottaviano, D. Mundici, "Algebraic Foundations of Many-valued Reasoning", Trends in Studia Logica, Kluwer, Dordrecht, 2000.

[22] S. N. Sivanandam, S. Sumathi and S. N. Deepa "Introduction to Fuzzy Logic using MATLAB Springer",-Verlag Berlin Heidelberg 2007.

[23] V. Loia, S. Sessa, 'Fuzzy relation equations for coding/decoding processes of images and Videos", Inform. Sci., 171, no. 1-3, 145{172, 2005.