A COLOR IMAGE CODING/DECODING METHOD BASED ON NEWLY CONSTRUCTED FUZZY TRANSFORMS

SHARMISTHA BHATTACHARYA (HALDER), PURNENDU DAS, SANJU DAS, SAPTARSHI PAUL

Abstract: A color image or a RGB image can be described by an array of $m \times n \times 3$ pixels. The value of the pixel lies between 0 and 255. In this paper we compress a color image by using newly defined fuzzy transform and also we reconstruct an approximation of the original image from the compressed image by using inverse fuzzy transform. The quality of the reconstructed image is judged by calculating the R.M.S.E. and P.S.N.R. with respect to the original image.

Keywords: Fuzzy set, Fuzzy transform, Inverse fuzzy transform, Image coding and decoding, PSNR.

Introduction: In classical Mathematics, various types of transforms are introduced (e.g. Laplace transform, Fourier transform, wavelet transform etc.) by various researchers. In 2001 Irina Perfilieva introduced fuzzy transform in her paper [2]. Latter on fuzzy transform is applied in to various fields, like image processing, data mining etc in the papers [5, 10]. The fuzzy transform provides a relation between the space of continuous functions defined on a bounded domain of real line R and R^n .Similarly inverse fuzzy transform identified each vector of Rn with a continuous map. In paper [7], I.Perfilieva prove that the function obtained by using inverse fuzzy transform approximates the original function with a very small precision. Similarly the discrete fuzzy transform and discrete inverse fuzzy transform can be applied for a function whose values at particular points are known.

The compression and decompression of the gray image by fuzzy transform method was introduced in the paper [], by F. Di Martino, V. Loia, I.Perfilieva and S. Sessa. In this paper we extend the compression and decompression techniques for coloring images by using fuzzy transform. We use two dimensional discrete fuzzy transform for compression of color image and for getting decompressed image we use discrete inverse fuzzy transform.

Fuzzy transforms: The central idea of the fuzzy transform is to partition the domain of the function by fuzzy sets. A fuzzy partition is defined as follows

Definition 2.1([5]): Let $[a, b]$ be an interval of real numbers and $x_1 < x_2 < \cdots < x_n$ be fixed nodes within $[a, b]$ such that $x_1 = a$, $x_n = b$ and $n \geq 2$ .We say that fuzzy sets $A_1, A_2, \dots, A_n$ identified with their membership functions $A_1(x), A_2(x), \dots, A_n(x)$ and defined on $[a, b]$ form a fuzzy partition of $[a, b]$ if they fulfill the following conditions for $i = 1, 2, \dots, n$.

1. $A_i : [a , b] \to [0 , 1]$, $A_i(x_i) = 1$.
2. $A_i(x) = 0$ if $x \notin (x_{i-1}, x_{i+1})$
3. $A_i(x)$ is continuous.
4. $A_i(x)$ is monotonically increasing on $[x_{i-1}, x_i]$ and monotonically decreasing on $[x_i, x_{i+1}]$.
5. $\sum_{i=1}^{n} A_i(x) = 1$, for all x.
6. $A_i(x_i - x) = A_i(x_i + x)$, for all $x \in [0, h]$, $i = 2, 3, \dots, n-1$, $n > 2$.
7. $A_{i+1}(x) = A_i(x - h)$, for all $x \in [a + h, b]$, for $i = 2, 3, \dots, n-2$, $n > 2$.
8. Where h is the uniform distance between two nodes.
9. **Definition 2.2([6]):** Let $f(x)$ be a continuous function on $[a,b]$ and $A_1(x), A_2(x), \dots, A_n(x)$ be basis functions determining a uniform fuzzy partition of $[a , b]$. Then the n-tuple of real numbers $[F_1, F_2, \dots, F_n]$ such that

$$F_i = \frac{\int_a^b f(x) A_i(x) dx}{\int_a^b A_i(x) dx}, \quad i = 1, 2, \dots, n. \tag{1}$$

will be called the F- transform of f w.r.t. the given basis functions. Real's F_i are called components of the F-transform.

Lemma 2.1([5]): Let f be any continuous function defined on $[a, b]$, but function f is twice continuously differentiable in (a, b) and let $A_1(x), A_2(x), \dots, A_n(x)$ be basis functions determining a uniform fuzzy partition of $[a, b]$ .Then for each $i = 1, 2, \dots, n$

$F_i = f(x_i) + O(h^2)$

Now a question arises in the minds that can we get back the original function by its fuzzy transform. The answer is we can reconstruct an approximate function to the original function. For that purpose Perfilieva define inverse fuzzy transform.

Definition 2.3([6]): Let $A_1, A_2, \dots, A_n$ be basic functions which form a uniform fuzzy partition of $[a, b]$ and f be a function from $c([a, b])$. Let $F_n[f] = [F_1, F_2, \dots, F_n]$ be the fuzzy transform of f with respect to $A_1, A_2, \dots, A_n$. Then the function defined by $f_{F,n}(x) = \sum_{i=1}^{n} F_i . A_i(x)$ is called the inverse fuzzy transform of f with respect to $A_1, A_2, \dots, A_n$.

The following theorem shows that the inverse fuzzy transform can approximate the original continuous function f with a very small precision.

Theorem 2.1 ([6]): Let f be a continuous functions defined on $[a, b]$. Then for any $\varepsilon > 0$ there exist n_ε and a uniform fuzzy partition $A_1, A_2, \dots, A_{n_\varepsilon}$ of $[a, b]$ such that for all $x \in [a, b]$

$\left| f(x) - f_{F,n_\varepsilon}(x) \right| \leq \varepsilon$

Note that the Theorem2.1 concerns the continuous, but now we will deal the discrete case, that is we only know that the function f assumes determined values in some points $p_1, p_2, \cdots, p_m$ of $[a,b]$. Assume that the set P of these nodes is sufficiently dense with respect to the fixed partition, i.e. for each $i = 1,2,\cdots,n$ there exist an index $j \in \{1,2,\cdots,m\}$, such that $A_i(p_j) > 0$. Then I. Perfilieva define the n-tuple $[F_1, F_2, \cdots, F_n]$ as the discrete fuzzy transform of f with respect to $\{A_1, A_2, \cdots, A_n\}$, where each F_i , is given as

$$F_i = \frac{\sum_{j=1}^{m} f(p_j)A_i(p_j)}{\sum_{j=1}^{m} A_i(p_j)} \quad (2)$$

for $i = 1,2,\cdots,n$.

Similarly I. Perfilieva also define the discrete inverse fuzzy transform of f with respect to $\{A_1, A_2, \cdots, A_n\}$ to be the following function defined on the same set of points $\{p_1, p_2, \cdots, p_m\}$ of $[a,b]$:

$$f_{F,n}(p_j) = \sum_{i=1}^{n} F_i A_i(p_j) \quad (3)$$

Analogously to the Theorem2.1, the following approximation theorem can be given:

Theorem 2.2([6]): Let $f(x)$ be a function assigned on a set P of points $p_1, p_2, \cdots, p_m$ of $[a,b]$. Then for every $\varepsilon > 0$, there exist an integer $n(\varepsilon)$ and a related fuzzy partition $A_1, A_2, \cdots, A_n$ of $[a,b]$ such that, P is sufficiently dense with respect to $A_1, A_2, \cdots, A_n$ and for every $p_j \in [a,b]$, $|f(p_j) - f_{F,n}(p_j)| < \varepsilon$, $j = 1,2,\cdots,m$ holds true.

Two variable fuzzy transform: We will consider a rectangle $D = [a,b] \times [c,d] \subset R^2$ as a common domain of all real valued functions. The main idea consists in construction of two fuzzy partition one of $[a,b]$ and another of $[c,d]$.

Definition3.1 ([2]): Let a fuzzy partition of $[a,b]$ be given by basic functions $\{A_1, A_2, \cdots, A_n\}$ with nodes $a = x_1 < x_2 < \cdots < x_n = b$ and another fuzzy partition of $[c,d]$ be given by basic functions $\{B_1, B_2, \cdots, B_m\}$, with nodes $c = y_1 < y_2 < \cdots < y_m = d$. Then the fuzzy partition of D is given by the fuzzy Cartesian product of $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, with respect to the product t-norm of these two fuzzy partition.

Definition 3.2([6]): Let a fuzzy partition of D is given by $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, and let $f \in C(D)$. We say that a real matrix $F^2[f] = [F_{ij}]_{n\times m}$ given by

$$F_{ij} = \frac{\int_a^b \int_c^d f(x,y)A_i(x)B_j(y)dxdy}{\int_a^b \int_c^d A_i(x)B_j(y)dxdy} \quad (4)$$

for $i = 1,2,\cdots,n$ and $j = 1,2,\cdots,m$ is called the fuzzy transform of f with respect to the given fuzzy partition. The reals F_{ij} are called the components of the fuzzy transform of f.

Definition 3.3([6]): Let a fuzzy partition of D is given by $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, and let $f \in C(D)$. Let $F^2[f] = [F_{ij}]_{n\times m}$ be the fuzzy transform of f with respect to $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$. Then the function

$$f_F(x,y) = \sum_{i=}^{2n} \sum_{j=}^{2m} BF_{ij}.A_i(x)B_j(y) \quad (5)$$

is called the inverse fuzzy transform of f for two variables.

Theorem 3.1([6]): Let f be a continuous function defined on D. Then for any $\varepsilon > 0$, there exist natural numbers n and m and a fuzzy partition $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$ of D such that for all $(x,y) \in D$, $|f(x,y) - f_F(x,y)| \le \varepsilon$

In the discrete case, we assume that the function f assumes determined values in some points $(p_i, q_j) \in [a,b] \times [c,d]$, where $i = 1,2,\cdots,N$ and $j = 1,2,\cdots,M$. Moreover the sets, $P = \{p_1, p_2, \cdots, p_N\} \times \{q_1, q_2, \cdots, q_M\}$, must be sufficiently dense with respect to the chosen partitions.

In this case Perfilieva define the matrix $[F_{kl}]_{n\times m}$ to be the discrete fuzzy transform of f with respect to $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, where for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots m$

$$F_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} f(p_i,q_j)A_k(p_i)B_l(q_j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k(p_i)B_l(q_j)} \quad (6)$$

The discrete inverse fuzzy transform of f with respect to $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, to be the following function which is defined on the same points $(p_i, q_j) \in [a,b] \times [c,d]$, with $i = 1,2,\cdots,N$ and $j = 1,2,\cdots,M$.

$$f_F(p_i, q_j) = \sum_{k=1}^{n} \sum_{l=1}^{m} F_{kl} A_k(p_i) B_l(q_j) \quad (7)$$

Theorem 3.2([6]): Let $f(x,y)$ be a function assigned on the points $(p_i, q_j) \in [a,b] \times [c,d]$, with $i = 1,2,\cdots,N$ and $j = 1,2,\cdots,M$. Then for every $\varepsilon > 0$, there exist two integers $n(\varepsilon), m(\varepsilon)$ with respect to the fuzzy partitions $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$, of D, such that the set of points $P = \{p_1, p_2, \cdots, p_N\} \times \{q_1, q_2, \cdots, q_M\}$, are sufficiently dense with respect to $\{A_1, A_2, \cdots, A_n\} \times \{B_1, B_2, \cdots, B_m\}$ and for every $(p_i, q_j) \in [a,b] \times [c,d]$, with $i = 1,2,\cdots,N$ and $j = 1,2,\cdots,M$. $|f(p_i, q_j) - f_F(p_i, q_j)| < \varepsilon$ holds true.

Coding/Decoding of RGB images: Let I be a RGB image divided in N × M pixels. Then I can be treated as a array of order N × M × 3. We divide the image I in three separate components Red, Green and Blue each of order, N × M × 1, and we denote these by matrices R, G and B, i.e. the matrix R, G and B corresponds the red, green and blue components of the image I, respectively. Now each of these components can be treated as a function from $\{1,2,\cdots,N\} \times \{1,2,\cdots,M\}$ to the real interval [0, 255]. That means the red components is a function given as, $R : \{1,2,\cdots,N\} \times \{1,2,\cdots,M\} \to [0,255]$ and $R(i,j)$ denotes the value of the pixel at (i,j) position in the red components matrix. Similarly the other two components G and B are also be treated as a function from $\{1,2,\cdots,N\} \times \{1,2,\cdots,M\}$ to the real interval [0, 255]. Now for compressing the original Image I, we compress its three components red, green and blue. That is we will compress the matrix$R(i,j)$, $G(i,j)$ and $B(i,j)$ and for compressing these matrix we use discrete fuzzy transform in two variables $[F_{kl}]$, defined for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots,m$ as

$$F_{kl}=\frac{\sum_{j=1}^{M}\sum_{i=1}^{N}R(i,j)A_k(i)B_l(j)}{\sum_{j=1}^{M}\sum_{i=1}^{N}A_k(i)B_l(j)} \tag{8}$$

Where by simplicity of notation we have assume, $p_i = i, q_j = j, a = c = 1, b = N \; and \; d = M$. Here $A_1, A_2, \cdots, A_n$ and $B_1, B_2, \cdots, B_m$, $< N, m < M$, are basic functions which forms a fuzzy partition of the interval $[1, N]$ and $[1, M]$ respectively. Resulting these we get a compressed matrix of order $m \times n$. Now for getting the original red components from these compressed matrix we used inverse fuzzy transform for every $(i, j) \in \{1,2,\cdots,N\} \times \{1,2,\cdots,M\}$ and is defined as follows

$$R^F=\sum_{k=1}^{n}\sum_{l=1}^{m}F_{kl}A_k(i)B_l(j) \tag{9}$$

We define the basic functions $\{A_1, A_2, \cdots, A_n\}$ and $\{B_1, B_2, \cdots, B_m\}$, used in equations (8) which form an uniform fuzzy partition of $[1, N]$ and $[1, M]$ respectively as follows:

$$A_1(x) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{h}(x - x_1)\right) & if \; x\epsilon[x_1, x_2], \\ 0 & otherwise, \end{cases}$$

$$A_k(x) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{h}(x - x_k)\right) & if \; x\epsilon[x_{k-1}, x_{k+1}], \\ 0 & otherwise, \end{cases}$$

$$A_n(x) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{h}(x - x_n)\right) & if \; x\epsilon[x_{n-1}, x_{n+1}], \\ 0 & otherwise, \end{cases} \tag{10}$$

where $k = 2, 3, \cdots, n, \; h = \frac{N-1}{n-1}$, $and \; x_k = 1 + h\,(k - 1)$ and

$$B_1(y) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{s}(y - y_1)\right) & if \; y \,\epsilon\, [y_1, y_2], \\ 0 & otherwise, \end{cases}$$

$$B_t(y) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{s}(y - y_t)\right) & if \; y \,\epsilon\, [y_{t-1}, y_{t+1}], \\ 0 & otherwise, \end{cases}$$

$$B_m(y) = \begin{cases} 0.5\left(1 + \cos\frac{\Pi}{s}(y - y_m)\right) & if \; y \,\epsilon\, [y_{m-1}, y_{m+1}], \\ 0 & otherwise, \end{cases} \tag{11}$$

where $t = 2, 3, \cdots, m, \; s = \frac{M-1}{m-1} \; and \; y_t = 1 + s\,(t - 1)$

Now by using these basic functions and by using formula in equation (8), we compressed the each of the components red, green and blue. That means by using fuzzy transform we get a compressed image of the original image. From this compressed image we construct a approximated image of the original image by using inverse fuzzy transform. After getting the approximated image we study the quality of the reconstructed image by using peak signal to noise ratio in short PSNR, which is defined as follows: $PSNR = 20\log_{10}\frac{255}{RMSE}$

where RMSE means root means square error and is given by the following formula:

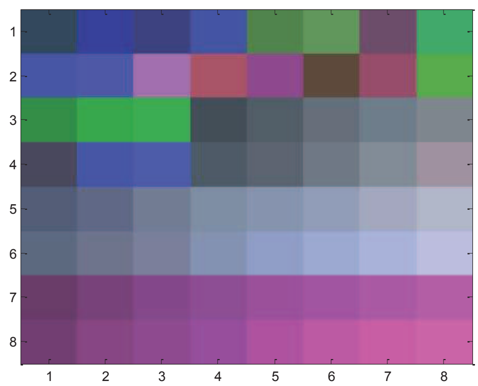
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{K=1}^{3}(A(i,j,k) - A\_NM\hat{\ }F\;(i,j,k))^2}{N \times M \times 3}}$$

where $A(i, j, k)$ is the original array of color matrix image and $A^F_{NM}(i, j, k)$ is the new array of the reconstructed image by using inverse fuzzy transform method.

It is noted that if the RMSE is approximately becomes zero then we say that the approximated image is exactly equal to the original image and consequently the PSNR value will becomes very large. That means that smaller the RMSE value more accurate the approximated image and if the PSNR value is large we say that the approximated image is more similar with the original image.

Example4.1: Consider the following color image of sizes 8×8 , with value pixels between 0 and 255 corresponding to the image of figure 1, and its red, green and blue components are also given below.

$$R = \begin{pmatrix} 54 & 57 & 62 & 69 & 84 & 98 & 110 & 69 \\ 71 & 75 & 162 & 169 & 142 & 94 & 150 & 90 \\ 54 & 57 & 62 & 69 & 84 & 102 & 110 & 125 \\ 74 & 67 & 72 & 79 & 94 & 112 & 130 & 160 \\ 84 & 97 & 114 & 127 & 134 & 146 & 162 & 177 \\ 94 & 110 & 122 & 132 & 145 & 156 & 168 & 186 \\ 105 & 121 & 132 & 142 & 156 & 166 & 180 & 194 \\ 115 & 134 & 142 & 153 & 176 & 196 & 215 & 225 \end{pmatrix}$$



$$G = \begin{pmatrix} 74 & 62 & 67 & 84 & 132 & 150 & 78 & 169 \\ 84 & 87 & 112 & 84 & 74 & 75 & 78 & 171 \\ 142 & 167 & 172 & 79 & 94 & 112 & 125 & 135 \\ 73 & 83 & 87 & 91 & 101 & 122 & 140 & 145 \\ 94 & 105 & 124 & 142 & 148 & 158 & 168 & 182 \\ 105 & 116 & 128 & 147 & 158 & 170 & 180 & 192 \\ 62 & 68 & 74 & 78 & 82 & 85 & 87 & 90 \\ 64 & 71 & 75 & 80 & 83 & 87 & 90 & 94 \end{pmatrix}$$

$$B = \begin{pmatrix} 94 & 157 & 127 & 169 & 78 & 92 & 107 & 108 \\ 178 & 187 & 177 & 103 & 140 & 62 & 107 & 78 \\ 74 & 77 & 82 & 89 & 104 & 124 & 137 & 140 \\ 94 & 186 & 205 & 102 & 114 & 134 & 150 & 161 \\ 120 & 135 & 148 & 164 & 175 & 185 & 192 & 200 \\ 130 & 140 & 155 & 178 & 200 & 210 & 231 & 241 \\ 107 & 121 & 139 & 147 & 154 & 166 & 180 & 187 \\ 115 & 130 & 143 & 156 & 165 & 175 & 185 & 194 \end{pmatrix}$$

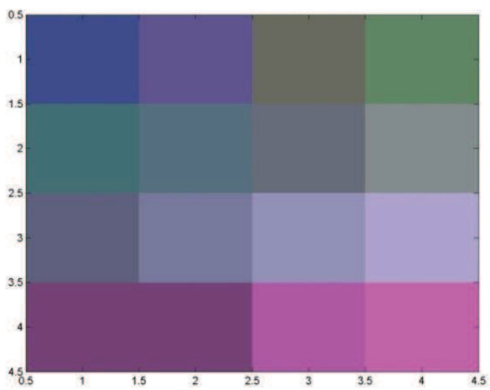
Now for compressing this image by using fuzzy transform method we need a fuzzy partition of the domain $[1,8] \times [1,8]$, for doing these we use basic functions in equations (10) and (11), by taking N = 8, M = 8, h = 7/3, k = 7/3. By using these basic fuzzy functions we compress the components of the image separately. We denote the compress matrix of the red, blue and green components as R′, G′ and B′ respectively each of size 3 × 3 (that means compression rate $\rho = \frac{4\times4}{8\times8} = 0.25$) and by using MATLAB, we evaluated these matrix and are given below:

$$R' = \begin{pmatrix} 63 & 96 & 105 & 96 \\ 67 & 87 & 108 & 132 \\ 96 & 120 & 147 & 175 \\ 118 & 118 & 178 & 207 \end{pmatrix}$$

$$G' = \begin{pmatrix} 77 & 85 & 107 & 134.5 \\ 111 & 111 & 108 & 139 \\ 96 & 119 & 144 & 163 \\ 67 & 67 & 87 & 94 \end{pmatrix}$$

$$B' = \begin{pmatrix} 140 & 143 & 98 & 101 \\ 116 & 127 & 123 & 141 \\ 128 & 156 & 184 & 207 \\ 118 & 119 & 169 & 189 \end{pmatrix}$$

Now by using the matrix R′, G′ and B′ as red, green and blue components respectively the compressed image is given below:



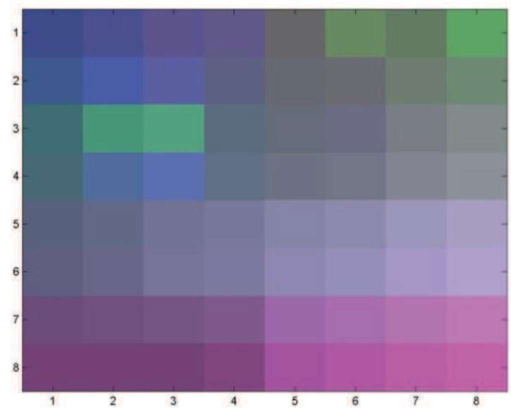
Now by apparent looking the image of figure1 and this compressed image are of quite similar but the advantage of these compressed image is that it requires only 16 data but the original image requires 64 data to store. Also whatever the quality of the image lost due to data compression can also be nearly managed by constructing decomposed image from this compressed image by using inverse fuzzy transform and this decomposed image nearly equal to the original image. Now for using inverse fuzzy transform we use the formula () for each components of the compressed matrix and consequently we get the reconstructed red, green and blue components, which combined to form the reconstructed RGB image and due to theorem we can say that these reconstructed image approximates the original image. The reconstructed red, green and blue matrices are given below:

$$R'' = \begin{pmatrix} 63 & 75 & 94 & 97 & 103 & 104 & 100 & 96 \\ 64 & 75 & 92 & 96 & 104 & 106 & 108 & 90 \\ 66 & 74 & 86 & 91 & 103 & 108 & 121 & 130 \\ 73 & 80 & 92 & 97 & 110 & 116 & 130 & 140 \\ 90 & 100 & 112 & 119 & 134 & 141 & 156 & 167 \\ 97 & 106 & 119 & 126 & 143 & 150 & 166 & 176 \\ 109 & 113 & 119 & 128 & 157 & 167 & 183 & 194 \\ 118 & 118 & 118 & 129 & 167 & 179 & 195 & 208 \end{pmatrix}$$

$$G'' = \begin{pmatrix} 77 & 80 & 85 & 89 & 103 & 108 & 123 & 135 \\ 90 & 92 & 95 & 97 & 105 & 106 & 125 & 137 \\ 109 & 109 & 110 & 109 & 108 & 109 & 126 & 138 \\ 107 & 109 & 112 & 112 & 114 & 116 & 132 & 143 \\ 98 & 106 & 117 & 121 & 133 & 138 & 150 & 159 \\ 95 & 103 & 116 & 122 & 137 & 142 & 152 & 160 \\ 78 & 81 & 87 & 91 & 105 & 109 & 116 & 120 \\ 67 & 67 & 67 & 71 & 83 & 87 & 91 & 94 \end{pmatrix}$$

$$B'' = \begin{pmatrix} 140 & 142 & 143 & 135 & 106 & 98 & 100 & 101 \\ 131 & 133 & 137 & 132 & 114 & 107 & 113 & 116 \\ 117 & 7121 & 127 & 126 & 122 & 122 & 132 & 139 \\ 118 & 123 & 182 & 132 & 133 & 135 & 145 & 153 \\ 125 & 135 & 149 & 154 & 169 & 174 & 186 & 195 \\ 127 & 138 & 153 & 160 & 178 & 185 & 198 & 207 \\ 122 & 126 & 133 & 141 & 167 & 176 & 188 & 196 \\ 118 & 118 & 118 & 128 & 160 & 170 & 181 & 189 \end{pmatrix}$$

and the corresponding reconstructed image is given below:



From this reconstructed image we evaluate the RMSE by using MATLAB and is found as equal to 1. That means root means square error is very small so we

can say that nearly approximates the original image. Subsequently we evaluate the PSNR value and is equal to 110.82, which can be treated as very good.
This technique can be used for several compressed ratio and the compressed matrix can again be compressed by using the same technique. If we use a series of compression for a image, then for getting an approximate image of the original image we must do the decomposition of image in reverse sequence.

Conclusion: In this paper we have developed a method for compression and decompression of RGB images by using fuzzy transform method. By using this method we can reduce the memory size of the picture since in this method we reduce the data required to store a image. We have also shown that form the compressed data we are able to approximate the original data.

References:

1. B. Kosko, “Fuzzy systems as universal approximators”, *IEEE Internet. Conf. On Fuzzy System, San Diego, CA, 1992, pp.1143-116.*
2. B. Bede, I.J.Rudas, “Approximation properties of fuzzy transform”, *Fuzzy sets and Systems,180(2011),20-40.*
3. I. Perfilieva, E.Chaldeeva, “Fuzzy transformation”, *in: Proc. of IFSA’ 2001 WorldCongress, Vancouver, Canada, 2001.*
4. M. Venkateswarlu , B. Rajini Kanth, A Simple Method for the Determination of Magneto Rheological Properties of Nanoparticulated Cobalt Ferrite Based Magnetorhelogical Fluids; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 20-23
5. I. Perfilieva, “Logical approximation”, *soft Comput. 7(2002) pp.73-78.*
6. I. Perfilieva, “Fuzzy transforms and their applications to data Compression”, *in: Proc. Int. Conf.on Fuzz – IEEE 2005, Reno, Nevada,USA, May 22-25, 2005, pp.294-299.*
7. I. Perfilieva, “Fuzzy transforms, in: J.F.Peters, et al.(Eds)”,*Transactions on Rough set II, Lecture Notes in Computer Science, vol.3135, 2004, pp.63-81.*
8. I. Perfilieva, “Fuzzy transforms: Theory and Applications”, *Fuzzy sets and Systems, 157(2006) 993-1023.*
9. Arnab Acharyya, Application of Image Halftone Technique in Visual Secret Sharing of Text Image; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 24-27
10. I. Perfilieva, “Fuzzy Transforms and their applications to data Compression. In I. Bloch et al. (Eds.)”, *Fuzzy logic and Applications, LNAI 3849, Springer, 2006, 19-31.*
11. I. Perfilieva, Fuzzy transforms: a challenge to conventional transforms, Advances in Imaging and Electron Physics 47 (2007) 137–196.
12. Ar. Arpan Dasgupta, Dr. Madhumita Roy, Comparing Visual Comfort in Daylight Pattern Between An Old and A Modern office Building; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 28-35
13. I. Perfilieva, M. Daˇnková, Image fusion on the basis of fuzzy transforms, in: Computational Intelligence in Decision and Control, World Scientific, New Jersey, 2008, pp. 471–476.
14. Dhaneswar Kalita, Sub-Linear Dc Photoconductivity of thermally Evaporated Polycrystalline CDSE Thin Films; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 36-44

***

Sharmistha Bhattacharya (Halder)/Associate Professor/Dept. of Mathematics/
Tripura University/Suryamaninagar/Agartala/ Tripura/ India- 799022/
Purnendu Das/ Assistant Professor/Dept. of Computer Science/Assam University/ Silchar/ India-788011/
Sanju Das/ Technical Assistant/ Dept. of Computer Science/ Assam University/ Silchar/ India-788011/
Saptarshi Paul/ Assistant Professor/Dept. of Computer Science/Assam University/ Silchar/