A NEW METHOD OF GRAY IMAGE CODING/DECODING BY USING NEWLY CONSTRUCTED SHEPARD KERNEL BASED FUZZY TRANSFORMS

Purnendu Das
Department of Computer Science, Assam University, Silchar, India
Bishwa Ranjan Roy
Department of Computer Science, Assam University, Silchar, India
Sanju Das
Department of Computer Science, Assam University, Silchar, India
Saptarshi Paul
Department of Computer Science, Assam University, Silchar, India

Abstract: An image can be described by an array of m×n pixels. The value of the pixel lies between 0 and 255. In this paper we compress an image by using newly defined Shepard kernel based 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, Shepard Kernel.

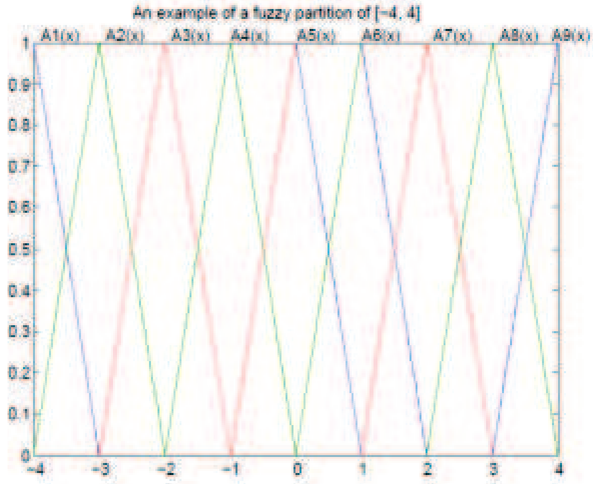
Introduction: In classical Mathematics, various types of transforms are introduced (e.g. Laplace transform, Fourier transform, wavelet transform etc.) by various researchers. In 2001 Irina Perfilieva introduced fuzzy transform in her paper [3]. Latter on fuzzy transform is applied in to various fields, 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 R^n with a continuous map. The central idea of the fuzzy transform is to partition the domain of the function by fuzzy sets.

Definition 1.1([6]): Let $[a, b]$ be an interval of real numbers and $x_1 < x_2 < \ldots < x_n$ be fixed nodes within $[a, b]$ such that $x_1 = a$, $x_n = b$ and $n \geq 2$ .We say that fuzzy sets $A_1, A_2, \ldots, A_n$ identified with their membership functions $A_1(x), A_2(x), \ldots\ldots, A_n(x)$ and defined on $[a, b]$ form a fuzzy partition of $[a, b]$ if they fulfill the following conditions for $i = 1, 2, \ldots, n$.

1. $A_i : [a , b] \rightarrow [0 , 1]$, $A_i(x_i) = 1$.
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 $x \in [0, h]$, $i = 2,\ldots,n-1$, $n > 2$.
7. $A_{i+1}(x) = A_i(x - h)$, for all $x \in [a + h, b]$, for $i = 2, 3,\ldots, n-2$, $n > 2$.

Where h is the uniform distance between two nodes.
The shape of basic functions is not predetermined and therefore it can be chosen additionally according to further requirements.

The following figure shows a fuzzy partition of the interval [−4, 4], with triangular membership functions.



The following expression gives the formal representation of such triangular membership functions.

$$A_1(x) = \begin{cases} -3 - x, x \in [x_1, x_2] \\ 0, \quad otherwise \end{cases}$$

and for $i = 2, 3, \dots, n-1$.

$$A_i(x) = \begin{cases} x - x_{i-1}, & if\ x \in [x_{i-1}, x_i] \\ 1 - x + x_i, & if\ x \in [x_i, x_{i+1}] \\ 0, & otherwise \end{cases}$$

$$A_n(x) = \begin{cases} x - x_{n-1}, & if\ x \in [x_{n-1}, x_n] \\ 0, & otherwise \end{cases}$$

Fuzzy Transform: In this section we first give the definition of fuzzy transform given by Irina Perfilieva in 2006.

Definition 2.1([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 $[a, b]$ .The n-tuple of real numbers $[F_1, F_2, \dots, F_n]$ such that

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

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 2.2([5]): Let f be any continuous function defined on $[a, b]$,but function f is twice continuously differentiable in (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 for each $i = 1, 2, \dots, n$

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

Proof: Perfilieva (2004) .

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 $A_1, A_2, \ldots, A_n$.
The following theorem shows that the inverse fuzzy transform can approximate the original continuous function f with a very small precision.

Theorem 2.4([8]): Let f, be a continuous functions defined on $[a, b]$. Then for any ε > o there exist n_ε and a uniform fuzzy partition $A_1, A_2, \ldots, A_{n_\varepsilon}$ 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 Theorem2.4 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

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

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 $[a, b]$:

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

Analogously to the Theorem2.4, the following approximation theorem can be given:

Theorem 2.5([7]): 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 $p_j \in [a, b]$,

$$\left|f(p_j) - f_{F,n}(p_j)\right| < \varepsilon, \; j = 1,2, \cdots, m$$

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 $[c, d]$.

Definition 3.1 ([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 ([7]): 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

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

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

$f_F(x,y) = \sum_{i=}^{2n}\sum_{j=}^{2m} BF_{ij}.A_i(x)B_j(y)$ (5)
is called the inverse fuzzy transform of f for two variables.

Theorem 3.4 ([**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, $P = \{p_1, p_2, \cdots, p_N\} \times \{q_1, q_2, \cdots, q_M\}$, msut 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 $l = 1,2,\cdots m$
$F_{kl} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f(p_i,q_j)A_k(p_i)B_l(q_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} A_k(p_i)B_l(q_j)}$ (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 $j = 1,2,\cdots,M$.
$f_F(p_i, q_j) = \sum_{k=1}^{n}\sum_{l=1}^{m} F_{kl}A_k(p_i)B_l(q_j)$ (7)

Theorem 3.5([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.

Shepard Kernel based F-Transform: Shepard Kernels are well known functions in the field of numerical analysis. In the paper "Approximation properties of fuzzy transform", by B. Bede and I.J. Rudas [2], a new type of fuzzy transform has been introduced by using Shepard kernels as basic functions, and named these transform as Shepard kernels based fuzzy transform. In this section we apply these new transform for approximating continuous function and study its applications for compression and decompression of gray image.

The Shepard kernels basic functions are defined as follows:

Definition 4.1 ([2]): Consider the closed interval$[a,b]$, and a partition of these intervals $\{y_1, y_2, \cdots, y_k\}$, then Shepard Kernels A_i for $i = 1,2,\cdots,k$ is defined as

$$A_i(x) = \begin{cases} \frac{|x - y_i|^{-\lambda}}{\sum_{j=0}^{k} |x - y_j|^{-\lambda}}, & if\ x \in [a,b]\backslash\{y_1, y_2, \cdots, y_k\} \\ \delta_{ij} & if\ x \in \{y_1, y_2, \cdots, y_k\} \end{cases}$$

Where δ_{ij} is Kronecker's delta and λ is a parameter.
One can check that the range of these Shepard Kernals functions is [0, 1]. Thus we can say that Shepard Kernels are fuzzy set defined on a closed interval. Also from the definition of the Shepard Kernels it can be shown that $\sum_{i=1}^{k} A_i(x) = 1$. These Shepard Kernels are continuous functions. Thus we can say that these Shepard Kernels satisfies the important properties of the fuzzy partition of closed intervals defined

in Definition1.1 except the support of the fuzzy set and B. Bede and I. J. Rudas shows that for proving the main theorem of fuzzy transform this property is not needed.

Definition 4.2 ([2]): Let [a, b] be any closed interval and f(x) be any real valued continuous functions defined on [a, b] . Let A_i for $i = 1,2,\cdots,k$ are the Shepard kernels defined on [a, b]. Then the n-tuple of real numbers [$F_1, F_2, \ldots, F_k$] such that

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

is called the Shepard kernels based F- transform of f . Real's F_i are called components of the Shepard kernels based F-transform.

Definition 4.3([3]): Let $A_1, A_2, \ldots, A_k$ be basic functions which form a fuzzy partition of [a ,b] and f be a function from C([a, b]). Let $F_k[f] = [F_1, F_2, \ldots, F_k]$ be the Shepard kernel based fuzzy transform of f with respect to $A_1, A_2, \ldots, A_k$. Then the function

$f_{F,k}(x) = \sum_{i=1}^{k} F_i.A_i(x)$

is called the inverse Shepard kernel based fuzzy transform.

Now we state the main important theorem of this section.

Theorem 4.4: Let f be a continuous functions defined on [a, b]. Then for any ε > 0 there exist n_ε and a Shepard kernel based fuzzy partition $A_1, A_2, \ldots, A_k$ of [a, b] such that for all $x \in$ [a ,b]

$|f(x) - f_{F,k}(x)| \le \varepsilon$.

The proof of the above theorem is similar to the proof of Theorem 5.2 in [2].

Example 4.5: Consider the closed interval [0, 4] and let f be a function defined on [0, 4] as $f(x) = x^2 + 1$. We approximate these functions by using Shepard kernel based fuzzy transform. We consider a partition {0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4}. Then the graph of the Shepard kernel basic functions taken λ=3 is given below:

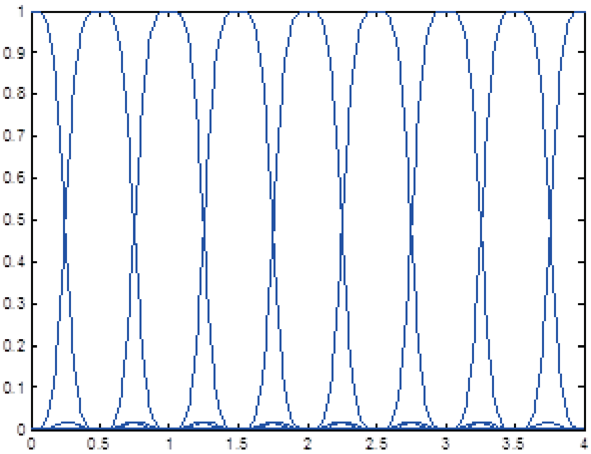


Figure 4.1: Shepard Kernel Based Fuzzy Partition of the Closed Interval [0, 4]

Now we have to evaluate the Shepard kernel based F-transform components of the function $f(x) = x^2 + 1$, with respect to the above Shepard kernels fuzzy partition. We evaluate these components by using software MATLAB and the corresponding values are

$$F_1 = 1.0484, F_2 = 1.2958, F_3 = 2.0442,$$
$$F_4 = 3.2889, F_5 = 5.0442, F_6 = 7.2904,$$
$$F_7 = 10.0154, F_8 = 13.2143, F_9 = 16.8166$$

Now by using Definition3.3 we evaluate the inverse Shepard kernel based fuzzy transform and the graph of these approximated curve and the original curve is given below, where red graph is the original curve and blue graph is our approximated curve:

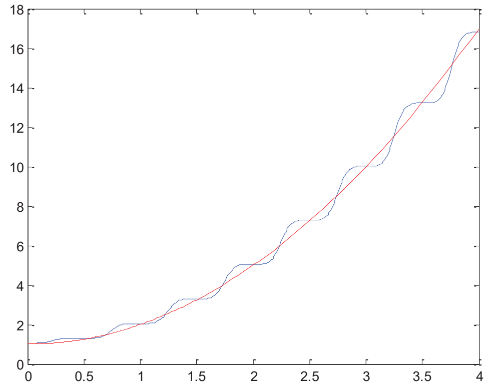


Figure 4.2: Graph of the Exact Function and The Graph of the Approximated Function

From the above figure we can see that the Shepard kernel based fuzzy transform approximates continuous functions very much accurately.

Two Variable Discrete Shepard Kernel based F-Transforms:

Definition 5.1.: Let a function $f(x,y)$ be given at nodes$(p_i,q_j) \in [a,b]\times[c,d]$, where $= 1,2,\cdots,N$, $j = 1,2,\cdots,M$. Let $A_1(x), A_2(x),\cdots,A_n(x)$ are the Shepard kernel based fuzzy partition of $[a,b]$, and $B_1(y), B_2(y)\cdots,B_m(y)$ are the Shepard kernel based fuzzy partition of $[c,d]$, where $n \ll N, m \ll M$. Then the components$[F_{kl}]$, defined for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots,m$ as,

$$F_{kl} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f(p_i,q_j)A_i(p_i)B_j(q_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} A_i(p_i)B_j(q_j)} \quad (8)$$

is called the discrete Shepard kernel based fuzzy transform of $f(x,y)$.

Definition5.2:Let F_{kl} , be the Shepard kernel based fuzzy transform components of the function $f(x,y)$ be given at nodes $(p_i,q_j) \in [a,b]\times[c,d]$, where $i = 1,2,\cdots,N$, $j = 1,2,\cdots,M$, with respect to the Shepard kernel based fuzzy partition $A_1(x), A_2(x)\cdots,A_n(x)$ of [a, b] and $B_1(y), B_2(y)\cdots,B_m(y)$ of [c, d], then the inverse Shepard kernel based fuzzy transform is also defined on the same points $(p_i,q_j) \in [a,b]\times[c,d]$, with $i = 1,2,\cdots,N$, $j = 1,2,\cdots,M$ and is defined as

$$f_{nm}^{F}(p_i,q_j) = \sum_{k=1}^{n}\sum_{l=1}^{m} F_{kl}A_k(p_i)B_l(q_j) \quad (9)$$

Theorem 5.3: Let $f(x,y)$ be a function assigned on the points $(p_i,q_j) \in [a,b]\times[c,d]$, for $i = 1,2,\cdots,N$, $j = 1,2,\cdots,M$. Then for every ε >o, there exist two integers $n(\varepsilon), m(\varepsilon)$ and related Shepard kernel based fuzzy partitions $A_1(x), A_2(x)\cdots,A_n(x)$ of [a, b] and $B_1(y), B_2(y)\cdots,B_m(y)$ of [c, d], such that the sets of points $P = \{p_1,p_2,\cdots,p_N\}$ and $Q = \{q_1,q_2,\cdots,q_M\}$ are sufficiently dense with respect to $\{A_1,A_2\cdots,A_{n(\varepsilon)}\}$ and $\{B_1,B_2\cdots,B_{m(\varepsilon)}\}$, and for every $(p_i,q_j) \in [a,b]\times[c,d]$, $i \in \{1,2,\cdots,N\}$, $j \in \{1,2,\cdots,M\}$

$$\left|f(p_i,q_j) - f_{nm}^{F}(p_i,q_j)\right| < \varepsilon$$

holds true.

Proof: we omit the proof since the proof is the generalization of the Theorem 3 in [12].

Coding/Decoding of Gray Image by Shepard Kernel Based F-Transform: Let I be a gray image, divided in N × M pixels. Then I can be treated as a matrix of order N × M , whose values lies in the closed interval [o, 1]. Thus we can say that I can be treated as a function $I : \{1,2,\cdots,N\}\times\{1,2,\cdots,M\} \to [0,255]$ and $I(i,j)$ denotes the intensity value of the pixel at (i,j)th position in the matrix. Thus for storing this image in computer one needs a space equal to store $N.M$ real data. We want to compress this image so that for storing this image we need $nm,$ real data, where $n < N, m < M$. Now for compressing the image $I(i,j)$, we define discrete Shepard kernel based fuzzy transform for two variables given below:

Definition 6.1.: Let a function $f(x,y)$ be given at nodes$(p_i, q_j) \in [a,b] \times [c,d]$, where $= 1,2,\cdots,N$, $j = 1,2,\cdots,M$. Let $A_1(x), A_2(x), \cdots, A_n(x)$ are the Shepard kernel based fuzzy partition of $[a,b]$, and $B_1(y), B_2(y) \cdots, B_m(y)$ are the Shepard kernel based fuzzy partition of $[c,d]$, where $n \ll N, m \ll M$. Then the components$[F_{kl}]$, defined for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots,m$ as,

$$F_{kl} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f(p_i,q_j)A_i(p_i)B_j(q_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} A_i(p_i)B_j(q_j)}$$

is called the discrete Shepard kernel based fuzzy transform of $f(x,y)$.

Definition 6.2: Let F_{kl} , be the Shepard kernel based fuzzy transform components of the function $f(x,y)$ be given at nodes $(p_i, q_j) \in [a,b] \times [c,d]$, where $i = 1,2,\cdots,N,\ j = 1,2,\cdots,M$, with respect to the Shepard kernel based fuzzy partition $A_1(x), A_2(x) \cdots, A_n(x)$ of [a, b] and $B_1(y), B_2(y) \cdots, B_m(y)$ of [c, d], then the inverse Shepard kernel based fuzzy transform is also defined on the same points $(p_i, q_j) \in [a,b] \times [c,d]$, with $i = 1,2,\cdots,N,\ j = 1,2,\cdots,M$ and is defined as

$$f_{nm}^{F}(p_i, q_j) = \sum_{k=1}^{n}\sum_{l=1}^{m} F_{kl}A_k(p_i)B_l(q_j) \quad (2)$$

Theorem6.3: Let $f(x,y)$ be a function assigned on the points $(p_i, q_j) \in [a,b] \times [c,d]$, for $i = 1,2,\cdots,N,\ j = 1,2,\cdots,M$. Then for every ε >0, there exist two integers $n(\varepsilon), m(\varepsilon)$ and releted Shepard kernel based fuzzy partitions $A_1(x), A_2(x) \cdots, A_n(x)$ of [a, b] and $B_1(y), B_2(y) \cdots, B_m(y)$ of [c, d], such that the sets of points $P = \{p_1, p_2, \cdots, p_N\}$ and $Q = \{q_1, q_2, \cdots, q_M\}$ are sufficiently dense with respect to $\{A_1, A_2 \cdots, A_{n(\varepsilon)}\}$ and $\{B_1, B_2 \cdots, B_{m(\varepsilon)}\}$, and for every $(p_i, q_j) \in [a,b] \times [c,d],\ i \in \{1,2,\cdots,N\},\ j \in \{1,2,\cdots,M\}$

$$\left|f(p_i,q_j) - f_{nm}^{F}(p_i,q_j)\right| < \varepsilon$$

holds true.

Proof: we omit the proof since the proof is the generalization of the Theorem 3 in [12].

Now we reverse back to our original problem of compression of grey image. Since the image I has $N \times M$ pixels we want to approximate this image by image of $n \times m$ pixels with $n < N, m < M$. For these purpose we take the discrete Shepard kernel based fuzzy transform of $I\ (i,j)$, defined for $i = \{1,2,\cdots,N\}$and $j = \{1,2,\cdots,M\}$. Thus after using Shepard kernel based fuzzy transform on $I\ (i,j)$, we will get the components$[F_{kl}]$, defined for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots,m$ as given in Definition4.1 with respect to the Shepard kernel based fuzzy partition $A_1(x), A_2(x) \cdots, A_n(x)$ of [1, N] and $B_1(y), B_2(y) \cdots, B_m(y)$ of [1, M] as given below.

$$A_i(x) = \begin{cases} \frac{|x-i|^{-\lambda}}{\sum_{j=1}^{n}|x-j|^{-\lambda}}, & if\ x \in [1,N]\backslash\{1,2,\cdots,n\} \\ \delta_{ij} & if\ x \in \{1,2,\cdots,n\} \end{cases} \quad (3)$$

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

and

$$B_j(y) = \begin{cases} \frac{|y-j|^{-\lambda}}{\sum_{i=1}^{n}|y-i|^{-\lambda}}, & if\ x \in [1,M]\backslash\{1,2,\cdots,m\} \\ \delta_{ji} & if\ x \in \{1,2,\cdots,m\} \end{cases} \quad (4)$$

for $j = 1,2,\cdots,m$

We take these Shepard kernel based fuzzy transform components$[F_{kl}]$, defined for each $k = 1,2,\cdots,n$ and $l = 1,2,\cdots,m$, as our new pixel values for the compressed image. Now for getting the original image we will use inverse Shepard kernel based fuzzy transform as given in Definition6.2 and in this way we will get an approximation of the original image from the compressed image. This new image is called the decompressed image. The quality of the new image is evaluated by using peak signal noise ratio (PSNR) and is given below

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\mathrm{RMSE}},$$

Where RMSE means root means square error and is given by the formula

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(A(i,j) - A_{NM}^{F})^2}{N \times M}}$$

Where $A(i,j)$ is the original array of the gray image and $A_{NM}^{F}(i,j)$ is the new array of the reconstructed image by using inverse fuzzy transform.

Example 6.4: Consider the following gray image in figure1 of 10×10 pixels, whose pixel values lies between 0 and 255. The corresponding pixel matrix I of the image is also given below.

$$I = \begin{bmatrix} 24 & 34 & 43 & 54 & 68 & 81 & 104 & 123 & 132 & 152 \\ 28 & 32 & 45 & 56 & 78 & 83 & 88 & 92 & 121 & 147 \\ 48 & 72 & 98 & 111 & 122 & 142 & 164 & 185 & 205 & 225 \\ 57 & 68 & 97 & 118 & 149 & 175 & 195 & 210 & 225 & 240 \\ 74 & 85 & 106 & 118 & 138 & 151 & 172 & 185 & 150 & 110 \\ 43 & 58 & 64 & 78 & 85 & 95 & 103 & 112 & 127 & 138 \\ 87 & 105 & 132 & 154 & 178 & 210 & 217 & 216 & 232 & 254 \\ 53 & 67 & 78 & 92 & 112 & 115 & 126 & 135 & 140 & 145 \\ 110 & 134 & 158 & 178 & 137 & 107 & 87 & 72 & 63 & 48 \\ 68 & 67 & 78 & 100 & 110 & 118 & 128 & 145 & 154 & 174 \end{bmatrix}$$

and the corresponding gray image is given below:

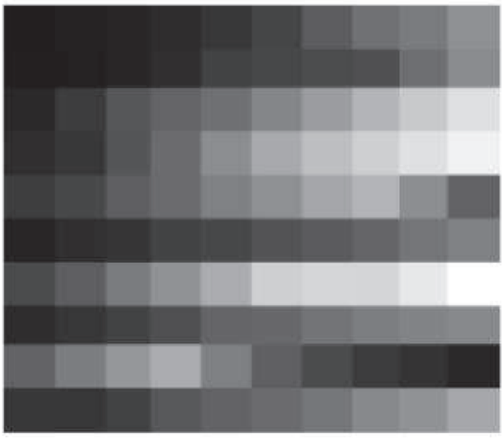


Figure 6.1: Gray Image

Now we want to compress this 10×10 pixel image to 6×6 pixel image. For doing this we need a Shepard kernel based fuzzy partition of $[1,10] \times [1,10]$. Consider $x_1 = 1 < x_2 < x_3 < x_4 < x_5 < x_6 = 10$ be a partition of $[1, 10]$ with equidistant nodes. Similarly for simplicity of notation we take another partition of $[1,10]$ as $y_1 = 1 < y_2 < y_3 < y_4 < y_5 < y_6 = 10$. With respect to this partition we form the Shepard kernel based fuzzy partition of $[1,10] \times [1,10]$ as $\{A_1, A_2, A_3, A_4, A_5, A_6\} \times \{B_1, B_2, B_3, B_4, B_5, B_6\}$ with λ =3 by using equation (3) and (4). Now with respect to this fuzzy partition we evaluate the Shepard kernel based fuzzy partition by using Definition6.1 and the corresponding compressed image matrix is given below:

$$\begin{bmatrix} 27.7279 & 41.2349 & 63.8198 & 90.7230 & 119.7865 & 145.5039 \\ 45.8051 & 71.8842 & 100.2276 & 129.6225 & 159.5120 & 190.9011 \\ 68.7810 & 68.7810 & 130.6352 & 169.6745 & 189.6016 & 174.3629 \\ 68.7457 & 92.2432 & 122.4641 & 152.0401 & 165.6288 & 186.4958 \\ 80.7324 & 105.5583 & 125.6623 & 117.7831 & 117.2185 & 116.5984 \\ 80.0114 & 94.2034 & 117.8059 & 117.1186 & 128.4269 & 140.3359 \end{bmatrix}$$

and the gray image corresponding to the above compressed matrix is

Figure 6.2: Compressed Gray Image

This image is the compressed image of the original image. The memory space required in computer to store this image is 0.36 times the memory required to the original image. So we can say that this process will save the memory location. Also from this image we can get back an approximation of the original image by using inverse Shepard kernel based fuzzy transform. If we apply the inverse Shepard kernel based fuzzy transform on the compressed image matrix , then we find the following reconstructed matrix in round figure:

$$\begin{bmatrix} 27 & 37 & 41 & 61 & 64 & 90 & 93 & 120 & 127 & 146 \\ 30 & 45 & 51 & 76 & 79 & 107 & 110 & 126 & 145 & 155 \\ 46 & 64 & 72 & 98 & 101 & 129 & 133 & 160 & 169 & 191 \\ 66 & 69 & 70 & 124 & 128 & 163 & 165 & 185 & 191 & 195 \\ 68 & 70 & 79 & 123 & 130 & 167 & 170 & 189 & 173 & 154 \\ 68 & 85 & 91 & 119 & 123 & 151 & 153 & 165 & 172 & 186 \\ 70 & 86 & 93 & 120 & 133 & 148 & 150 & 170 & 185 & 198 \\ 80 & 97 & 105 & 123 & 125 & 117 & 117 & 117 & 118 & 116 \\ 80 & 95 & 101 & 120 & 122 & 118 & 118 & 92 & 82 & 76 \\ 78 & 83 & 88 & 105 & 120 & 128 & 136 & 156 & 162 & 184 \end{bmatrix}$$

and the corresponding reconstructed image is given below:

Figure 6.3: Reconstructed Gray Image from the Compressed Image

For evaluating the quality of this reconstructed image in comparison to the original image we evaluate the root mean square error (RMSE) by using MATLAB and is found as equal to 27.6. So we can say that in this case the root mean square error is very small and thus our reconstructed image approximates the original image. Subsequently we evaluate the PSNR value and is equal to 44.46, 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 gray images by using Shepard Kernel based 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. The method can be extended for RGB image also.

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. F. Di Martino, Salvatro Sessa: Compression and decompression of images with discrete fuzzy transforms, *Information Sciences,* 177(2007), 2349-2362.
4. I. Perfilieva, E.Chaldeeva, "Fuzzy transformation", *in: Proc. of IFSA' 2001 WorldCongress, Vancouver, Canada, 2001.*
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. 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.*
10. I. Perfilieva, Fuzzy transforms: a challenge to conventional transforms, Advances in Imaging and Electron Physics 47 (2007) 137–196.
11. 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.
12. Stepnicka M: Fuzzy transform and its applications to problems in engineering practice, Ph.D thesis, submitted in University of Ostrava, Faculty of Science, Department of Mathematics, in 2007.

***