Journal of Signal and Information Processing, 2017, 8, 178-194
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

First Order Fuzzy Transform for Images Compression

Ferdinando Di Martino[1], Salvatore Sessa[2], Irina Perfilieva[3]

[1]Università degli Studi di Napoli Federico II, Dipartimento di Architettura, Napoli, Italy
[2]Università degli Studi di Napoli Federico II, Centro Interdipartimentale di Ricerca Calza Bini, Napoli, Italy
[3]University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, Ostrava, Czech Republic
Email: fdimarti@unina.it, sessa@unina.it, irina.perfilieva@osu.cz

How to cite this paper: Di Martino, F., Sessa, S. and Perfilieva, I. (2017) First Order Fuzzy Transform for Images Compression. *Journal of Signal and Information Processing*, **8**, 178-194.
https://doi.org/10.4236/jsip.2017.83012

Received: May 16, 2017
Accepted: August 22, 2017
Published: August 25, 2017

Abstract

In this paper, we present a new image compression method based on the direct and inverse F^1-transform, a generalization of the concept of fuzzy transform. Under weak compression rates, this method improves the quality of the images with respect to the classical method based on the fuzzy transform.

Keywords

Fuzzy Transform, Generalized Fuzzy Partition, Basic Function, Hilbert Space, Image Compression, PSNR

1. Introduction

We present a new image compression method based on the discrete direct and inverse F^1-transform which is a generalization of the classical fuzzy transform [1] [2] identified as F^0-transform (for brevity, F-transform).

The F-transform compression technique [3] is a lossy compression method used in image and video analysis [4]-[18] and in data analysis [19]-[25] as well. In [26], the concept of the F-transform was extended to the cases with various types of fuzzy partitions. In [1] [27], the F^s-transform ($s \geq 1$), a generalization of the F-transform, was presented: in other terms, the constant components of the F-transform were replaced by polynomials in order to capture more information of the original function. In particular, the F^1-transform was used for the edge detection problem [1] [2]. The aim of this paper is to improve the quality of the decoded images after their compression via the F^1-transform-based method.

Strictly speaking, we divide images of sizes $N \times M$ into smaller images (called blocks) of sizes $N(B) \times M(B)$ and then we code each block into another one of sizes $n(B) \times m(B)$, where $n(B) < N(B)$ and $m(B) < M(B)$. The compression is performed by calculating the direct F^1-transform components with first degree polynomials. Afterwards, we calculate the inverse F^1-transform and obtain the corresponding decoded blocks, recomposed to obtain the final reconstructed image. In Figure 1, we describe this process in detail.

The compression rate is given by $\rho = \left(n(B)\times m(B)\right)/\left(N(B)\times M(B)\right)$. The quality of a decoded image is measured by the Peak Signal to Noise Ratio (PSNR) index.

In Section 2, we recall the definition of h-uniform generalized fuzzy partition and the concept of F^1-transform. In Section 3, a F^1-transform-based compression method is presented and it is applied to images considered as fuzzy relations: there every image is partitioned into smaller blocks and the direct and inverse F^1-transforms are calculated for each block. Then the decoded blocks are recomposed and the PSNR index is calculated. In Section 4, tests are applied to grey image datasets and the results are compared with similar results obtained by using the classical F-transform compression method. Section 5 contains the conclusions.

2. Generalized Fuzzy Partition and F¹-Transform

We recall the main concepts [2] that will be used in the sequel. We consider a set of points (called nodes) $x_0, x_1, x_2, \cdots, x_n, x_{n+1}, n \ge 2$ of $[a,b]$ such that $a = x_0 \le x_1 < x_2 < \cdots < x_n \le x_{n+1} = b$. We say that the fuzzy sets $A_1, \cdots, A_n : [a,b] \to [0,1]$ form a generalized fuzzy partition of $[a,b]$, if for each $k = 1, 2, \cdots, n$, there exist $h'_k, h''_k \ge 0$ such that $h'_k + h''_k > 0$, $[x_k - h'_k, x_k + h''_k] \subseteq [a,b]$ and the following constraints hold:

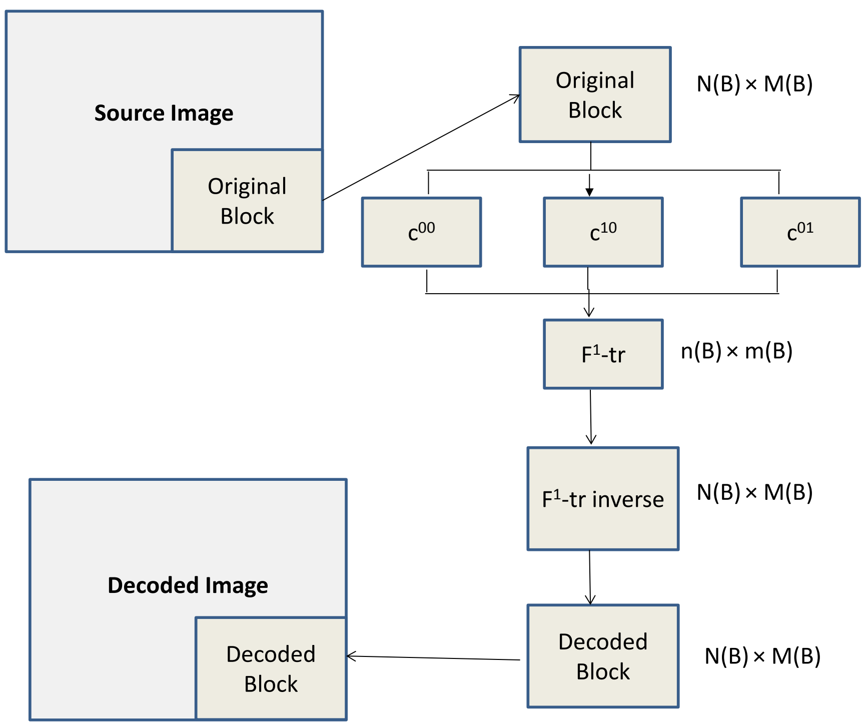


Figure 1. The F^1-transform image compression method.

1) (locality) $A_k(x) > 0$ if $x \in [x_k - h'_k, x_k + h''_k]$ and $A_k(x) = 0$ if $x \notin [x_k - h'_k, x_k + h''_k]$,

2) (continuity) A_k is continuous in $[x_k - h'_k, x_k + h''_k]$,

3) (covering) for each $x \in [a,b]$, $\sum_{k=1}^{n} A_k(x) > 0$.

The fuzzy sets $\{A_1, \cdots, A_n\}$ are called basic functions. If the nodes $x_0, x_1, \cdots, x_n, x_{n+1}$ are equidistant, *i.e.* $x_{k+1} - x_k = h$ for $k = 0,1,2,\cdots,n$, where $h = (b-a)/(n+1)$, if $h' > h/2$ and the following additional properties hold:

4) $h'_1 = h''_n = 0$, $h''_1 = h'_2 = \cdots = h''_{n-1} = h'_n = h'$ and $A_k(x_k - x) = A_k(x_k + x)$ for each $x \in [0, h']$ and $k = 2, \cdots, n$,

5) $A_k(x) = A_{k-1}(x-h)$ and $A_{k+1}(x) = A_k(x-h)$ for every $x \in [x_k, x_{k+1}]$ and $k = 2, \cdots, n$, then $\{A_1, \cdots, A_n\}$ is called an (h, h')-uniform generalized fuzzy partition. In this case we can find a function $A_0 : [-1,1] \to [0,1]$, called generating function, which is assumed to be even, continuous and positive everywhere except on the boundaries, where it vanishes, in such a way we have that for $k = 1, 2, \cdots, n$:

$$A_k(x) = \begin{cases} A_0\left(\dfrac{x - x_k}{h}\right) & x \in [x_k - h, x_k + h] \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

If $h = h'$, then the (h, h')-uniform generalized fuzzy partition is said h-uniform generalized fuzzy partition. We can extend the notion of h-uniform generalized fuzzy partition from an interval to the rectangle $[a,b] \times [c,d]$, so that we have the family of basic functions $\{A_k \times B_l, k = 1, \cdots, n, l = 1, \cdots, m; n, m \geq 2\}$, where $A_k \times B_l$ is the product of the corresponding functions from the h_1-uniform generalized fuzzy partition $\{A_1, \cdots, A_n\}$ of $[a,b]$ and from the h_2-uniform generalized fuzzy partition $\{B_1, \cdots, B_m\}$ of $[c,d]$. Then we can say that $\{A_k \times B_l, k = 1, \cdots, n, l = 1, \cdots, m; n, m \geq 2\}$ is an h-uniform generalized fuzzy partition of $[a,b] \times [c,d]$, where $h = h_1 \cdot h_2$. In the sequel we consider only such h-uniform generalized fuzzy partitions.

Let $A_k(x)$ be a basic function of $[a,b]$ and $L_2(A_k)$ be the Hilbert space of square integrable functions $f : [x_{k-1}, x_{k+1}] \to R$ (reals) with weighted inner product:

$$\langle f, g \rangle_k = \int_{x_{k-1}}^{x_{k+1}} f(x) g(x) A_k(x) \mathrm{d}x$$

Likewise, we define the Hilbert space $L_2(A_k \times B_l)$ of square integrable in two variables functions $f : [x_{k-1}, x_{k+1}] \times [y_{l-1}, y_{l+1}] \to R$ with weighted inner product:

$$\langle f, g \rangle_{kl} = \int_{x_{k-1}}^{x_{k+1}} \int_{y_{l-1}}^{y_{l+1}} f(x) g(x) A_k(x) B_l(y) \mathrm{d}x\mathrm{d}y \tag{2}$$

Two function $f, g \in L_2(A_k \times B_l)$ are orthogonal if $\langle f, g \rangle_{kl} = 0$. Let $L_2^p(A_k)$ and $L_2^r(B_l)$, $p, r \geq 0$ be two linear subspaces of $L_2(A_k)$ and $L_2(B_l)$ with orthogonal basis given by polynomials $\{P_k^i(x)\}_{i=0,\cdots,p}$ and $\{Q_k^j(y)\}_{j=0,\cdots,r}$, respectively.

We consider an integer $s \geq 0$ and all pairs of integers (i, j) such that $0 \leq i + j \leq s$. We introduce a linear subspace $L_2^s\left(A_k \times B_l\right)$ of $L_2\left(A_k \times B_l\right)$ having as orthogonal basis the following:

$$\left\{S_{kl}^{ij}\left(x,y\right) = P_k^i\left(x\right)Q_l^j\left(y\right)\right\}_{i=0,\cdots,p;\ j=0,\cdots,r:i+j\leq s} \tag{3}$$

where s is the maximum degree of polynomials $P_k^i\left(x\right)Q_l^j\left(y\right)$. For $s = 1$, the orthogonal basis of the linear space $L_2^1\left(A_k \times B_l\right)$ is the following:

$$\left\{S_{kl}^{00}\left(x,y\right) = P_k^0\left(x\right)Q_l^0\left(y\right), S_{kl}^{10}\left(x,y\right) = P_k^1\left(x\right)Q_l^0\left(y\right), S_{kl}^{01}\left(x,y\right) = P_k^0\left(x\right)Q_l^1\left(y\right)\right\} \tag{4}$$

Let $L_2\left(\left[a,b\right]\times\left[c,d\right]\right)$ be a set of functions $f:\left[a,b\right]\times\left[c,d\right]\to R$ such that for $k = 1,\cdots,n$, $l = 1,\cdots,m$, $f \mid \left[x_{k-1},x_{k+1}\right]\times\left[y_{k-1},y_{k+1}\right] \in L_2\left(A_k\right)\times L_2\left(B_l\right)$, where the function $f \mid \left[x_{k-1},x_{k+1}\right]\times\left[y_{k-1},y_{k+1}\right]$ is the restriction of f on $\left[x_{k-1},x_{k+1}\right]\times\left[y_{k-1},y_{k+1}\right]$. Then the following theorem holds:

Theorem 1. ([2], lemma 5). Let $f \in L_2\left(\left[a,b\right]\times\left[c,d\right]\right)$. Then the orthogonal projection of f on $L_2^s\left(A_k \times B_l\right)$, $s \geq 0$, is the polynomial of degree s given by

$$F_{kl}^s\left(x,y\right) = \sum_{0\leq i+j\leq s} c_{kl}^{ij} S_{kl}^{ij}\left(x,y\right) \tag{5}$$

for every $\left(x,y\right)\in\left[a,b\right]\times\left[c,d\right]$, where the coefficients c_{kl}^{ij} are given by

$$c_{kl}^{ij} = \frac{\int_{y_{l-1}}^{y_{l+1}}\int_{x_{k-1}}^{x_{k+1}} f\left(x,y\right)S_{kl}^{ij}\left(x,y\right)A_k\left(x\right)B_l\left(y\right)\mathrm{d}x\mathrm{d}y}{\int_{y_{l-1}}^{y_{l+1}}\int_{x_{k-1}}^{x_{k+1}} \left(S_{kl}^{ij}\left(x,y\right)\right)^2 A_k\left(x\right)B_l\left(y\right)\mathrm{d}x\mathrm{d}y} \tag{6}$$

Following [2], let $\left\{A_k \times B_l, k = 1,\cdots,n, l = 1,\cdots,m, n,m \geq 2\right\}$ be an h-uniform generalized fuzzy partition of $\left[a,b\right]\times\left[c,d\right]$ and $f \in L_2\left(A_k \times B_l\right)$. For $s = 1$, the orthogonal basis of the linear subspace $L_2^1\left(A_k \times B_l\right)$ is given by the polynomials:

$$\begin{aligned} S_{kl}^{00}\left(x,y\right) &= P_k^0\left(x\right)Q_l^0\left(y\right) = 1 \\ S_{kl}^{10}\left(x,y\right) &= P_k^1\left(x\right)Q_l^0\left(y\right) = x - x_k \\ S_{kl}^{01}\left(x,y\right) &= P_k^0\left(x\right)Q_l^1\left(y\right) = y - y_l \end{aligned} \tag{7}$$

Let F_{kl}^1 be the orthogonal projection of $f \mid \left[x_{k-1},x_{k+1}\right]\times\left[y_{k-1},y_{k+1}\right]$ on $L_2^1\left(A_k \times B_l\right)$ given point wise as

$$F_{kl}^1\left(x,y\right) = \sum_{0\leq i+j\leq 1} c_{kl}^{ij} S_{kl}^{ij}\left(x,y\right) = c_{kl}^{00} + c_{kl}^{10}\left(x - x_k\right) + c_{kl}^{01}\left(y - y_l\right) \tag{8}$$

for every $\left(x,y\right)\in\left[a,b\right]\times\left[c,d\right]$, where the three coefficients $c_{kl}^{00}, c_{kl}^{10}, c_{kl}^{01}$ are defined by Theorem 1:

$$c_{kl}^{00} = \frac{\int_{y_{l-1}}^{y_{l+1}}\int_{x_{k-1}}^{x_{k+1}} f\left(x,y\right)A_k\left(x\right)B_l\left(y\right)\mathrm{d}x\mathrm{d}y}{\int_{x_{k-1}}^{x_{k+1}} A_k\left(x\right)\mathrm{d}x \int_{y_{l-1}}^{y_{l+1}} B_l\left(y\right)\mathrm{d}y} \tag{9}$$

$$c_{kl}^{10}=\frac{\int_{y_{l-1}}^{y_{l+1}}\int_{x_{k-1}}^{x_{k+1}} f(x,y)(x-x_k)A_k(x)B_l(y)\mathrm{d}x\mathrm{d}y}{\int_{x_{k-1}}^{x_{k+1}} A_k(x)(x-x_k)^2\mathrm{d}x\int_{y_{l-1}}^{y_{l+1}} B_l(y)\mathrm{d}y} \tag{10}$$

$$c_{kl}^{01}=\frac{\int_{y_{l-1}}^{y_{l+1}}\int_{x_{k-1}}^{x_{k+1}} f(x,y)(y-y_l)A_k(x)B_l(y)\mathrm{d}x\mathrm{d}y}{\int_{x_{k-1}}^{x_{k+1}} A_k(x)\mathrm{d}x\int_{y_{l-1}}^{y_{l+1}} B_l(y)(y-y_l)^2\mathrm{d}y} \tag{11}$$

Then the matrix $\boldsymbol{F}_{nm}^1[f]=\left(F_{11}^1,\cdots,F_{nm}^1\right)$, defined from (8), is called F^1-transform of the function $f\in L_2\left(A_k\times B_l\right)$ with respect to the h-uniform generalized fuzzy partition $\left\{A_k\times B_l, k=1,\cdots,n, l=1,\cdots,m; n,m\geq 2\right\}$. We define the inverse F^1-transform of the function $f\in L_2\left(A_k\times B_l\right)$ to be a function $\hat{f}_{nm}^1:[a,b]\times[c,d]\rightarrow R$ as

$$\hat{f}_{nm}^1(x,y)=\frac{\sum_{k=1}^{n}\sum_{l=1}^{m}F_{nm}^1(x,y)A_k(x)B_l(y)}{\sum_{k=1}^{n}\sum_{l=1}^{m}A_k(x)B_l(y)} \tag{12}$$

For sake of completeness, we point out the utility of the concept of inverse F^1-transform which stands in the approximation of the function $f\in L_2\left(A_k\times B_l\right)$ under certain suitable assumptions. For example, we have the following result:

Theorem 2. ([2], theorem 14). Let $\left\{\left(A_k(x),B_l(y)\right), k=1,\cdots,n, l=1,\cdots,m, n,m\geq 2\right\}$ be an h-uniform generalized fuzzy partition of $[a,b]\times[c,d]$ and $\hat{f}_{nm}^1$ be the inverse F^1-transform of f with respect to this fuzzy partition. Moreover let f be four times continuously differentiable on $[a,b]\times[c,d]$ and A_k (resp., B_l) be four times continuously differentiable on $[a,b]$ (resp., $[c,d]$). Then the following holds for every $(x,y)\in[a,b]\times[c,d]$:

$$f(x,y)-\hat{f}_{nm}^1(x,y)=O\left(h^2\right) \tag{13}$$

In other words, the Equality (13) says that we can approximate a function in two variables, four times continuously differentiable on $[a,b]\times[c,d]$, with the inverse F^1-transform (12) unless to O (h^2).

3. F^1-Transform Image Compression Method

We are interested to the case discrete, *i.e.* we consider functions in two variables which assume a finite number of values in $[0,1]$ like finite fuzzy relations. Indeed, let R be a grey image of sizes $N\times M$, $R:(i,j)\in\{1,\cdots,N\}\times\{1,\cdots,M\}\rightarrow[0,1]$, $R(i,j)=R_{ij}$ being the normalized value of the pixel $P(i,j)$, that is $R(i,j)=P(i,j)/N_{lev}$ if N_{lev} is the length of the grey scale. Let $\{A_1,\cdots,A_n\}$ and $\{B_1,\cdots,B_m\}$ be two h-uniform generalized fuzzy partitions of $[a,b]=[1,N]$ and $[c,d]=[1,M]$, respectively, where $a=1$,

$b = N$, $x_k = k, k = 1,2,\cdots,n, n \ll N$, $c = 1$, $d = M$, $y_l = l, l = 1,2,\cdots,m, m \ll M$. Slightly modifying (8), then we can define the (discrete) F[1]-transform $R_{nm}^1 = \left[R_{kl}^1\right]_{n\times m}$ of R the matrix whose entries are defined as

$$R_{kl}^1 = c_{kl}^{00} + c_{kl}^{10} \cdot |i-k| + c_{kl}^{01} |j-l| \tag{14}$$

where c_{kl}^{00}, c_{kl}^{10}, c_{kl}^{01} are given as (by rewriting the Equations (9), (10), (11) in the following form, slightly modified):

$$c_{kl}^{00} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} R_{ij} A_k(i) B_l(j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} A_k(i) B_l(j)} \tag{15}$$

$$c_{kl}^{10} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} R_{ij} |i-k| A_k(i) B_l(j)}{\sum_{i=1}^{N} A_k(i)(i-k)^2 \sum_{j=1}^{M} B_l(j)} \tag{16}$$

$$c_{kl}^{01} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} R_{ij} |j-l| A_k(i) B_l(j)}{\sum_{i=1}^{N} A_k(i) \sum_{j=1}^{N} B_l(j)(j-l)^2} \tag{17}$$

The Formula (14) is considered as a compressed image of the original image R. R_{nm}^1 can be decoded by using the following inverse (discrete) F[1]-transform $R_{NM}^1 = \left[R_{ij}^1\right]_{N\times M}$ defined for every $(i, j) \in \{1,\cdots,N\} \times \{1,\cdots,M\}$ as

$$R_{ij}^1 = \frac{\sum_{k=1}^{n}\sum_{l=1}^{m} R_{kl}^1 A_k(i) B_l(j)}{\sum_{k=1}^{n}\sum_{l=1}^{m} A_k(i) B_l(j)} \tag{18}$$

We divide the image R of sizes $N \times M$ in sub-matrices R^B of sizes $N(B)\times M(B)$, called blocks ([26] [28]), each compressed to a block $\left[R_{kl}^{1B}\right]_{n(B)\times m(B)}$ of sizes $n(B)\times m(B)$ $(3 \le n(B) < N(B), 3 \le m(B) < M(B))$, $k = 1,\cdots,n(B)$, $l = 1,\cdots,m(B)$, via the discrete F[1]-transform, as Formula (14), of components R_{kl}^{1B} given by

$$R_{kl}^{1B} = c_{kl}^{00B} + c_{kl}^{10B} |i-k| + c_{kl}^{01B} |j-l| \tag{19}$$

We rewrite (15), (16), (17) as

$$c_{kl}^{00B} = \frac{\sum_{j=1}^{M(B)}\sum_{i=1}^{N(B)} R_{ij}^B A_k(i) B_l(j)}{\sum_{j=1}^{M(B)}\sum_{i=1}^{N(B)} A_k(i) B_l(j)} \tag{20}$$

$$c_{kl}^{10B} = \frac{\sum_{j=1}^{M(B)}\sum_{i=1}^{N(B)} R_{ij}^B |i-k| A_k(i) B_l(j)}{\sum_{i=1}^{N(B)} A_k(i)(i-k)^2 \sum_{j=1}^{M(B)} B_l(j)} \tag{21}$$

$$c_{kl}^{01B} = \frac{\sum_{j=1}^{M(B)} \sum_{i=1}^{N(B)} R_{ij}^{B} |j-l| A_k(i) B_l(j)}{\sum_{i=1}^{N(B)} A_k(i) \sum_{j=1}^{M(B)} B_l(j)(j-l)^2} \tag{22}$$

The basic functions $\{A_1, \cdots, A_{n(B)}\}$ and $\{B_1, \cdots, B_{m(B)}\}$ form an h-uniform generalized uniform fuzzy partition of $[1, N(B)]$ and $[1, M(B)]$, respectively. They are generated by the basic functions $A_0(x) = 0.5[1+\cos(\pi x)]$ and $B_0(y) = 0.5[1+\cos(\pi y)]$, respectively. Then we have that

$$A_1(x) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_1}(x-x_1)\right) & \text{if } x \in [x_1, x_2] \\ 0 & \text{otherwise} \end{cases}$$
$$A_k(x) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_1}(x-x_k)\right) & \text{if } x \in [x_{k-1}, x_{k+1}] \\ 0 & \text{otherwise} \end{cases} \tag{23}$$
$$A_n(x) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_1}(x-x_n)\right) & \text{if } x \in [x_{n-1}, x_n] \\ 0 & \text{otherwise} \end{cases}$$

where $n = n(B)$, $h_1 = (N(B)-1)/(n-1)$, $x_k = 1 + h_1(k-1), k = 2, \cdots, n-1$ and

$$B_1(y) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_2}(y-y_1)\right) & \text{if } y \in [y_1, y_2] \\ 0 & \text{otherwise} \end{cases}$$
$$B_l(y) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_2}(y-y_t)\right) & \text{if } y \in [y_{t-1}, y_{t+1}] \\ 0 & \text{otherwise} \end{cases} \tag{24}$$
$$B_m(y) = \begin{cases} 0.5\left(1+\cos\frac{\pi}{h_2}(y-y_m)\right) & \text{if } y \in [y_{m-1}, y_m] \\ 0 & \text{otherwise} \end{cases}$$

where $m = m(B)$, $h_2 = (M(B)-1)/(m-1)$, $y_l = 1 + h_2 \cdot (l-1)$, $l = 2, \cdots, m-1$. In Figure 2, we show the basic functions (23) for $N = 16$ and $n = 4$.

The compressed block $[R_{kl}^{1B}]_{n(B)\times m(B)}$ is decoded to a block $[R_{ij}^{1B}]_{N(B)\times M(B)}$ of sizes $N(B)\times M(B)$ by using the inverse F[1]-transform defined for every $(i, j) \in \{1, \cdots, N_B\} \times \{1, \cdots, M_B\}$ as

$$R_{ij}^{1B} = \frac{\sum_{k=1}^{n(B)} \sum_{l=1}^{m(B)} R_{kl}^{1B} A_k(i) B_l(j)}{\sum_{k=1}^{n(B)} \sum_{l=1}^{m(B)} A_k(i) B_l(j)} \tag{25}$$

which approximates the original block R^B. Making the union of all the decoded blocks R^{1B}, we obtain a fuzzy relation (denoted with) R^1 of sizes $N \times M$. Then we measure the RMSE (Root Mean Square Error) given by

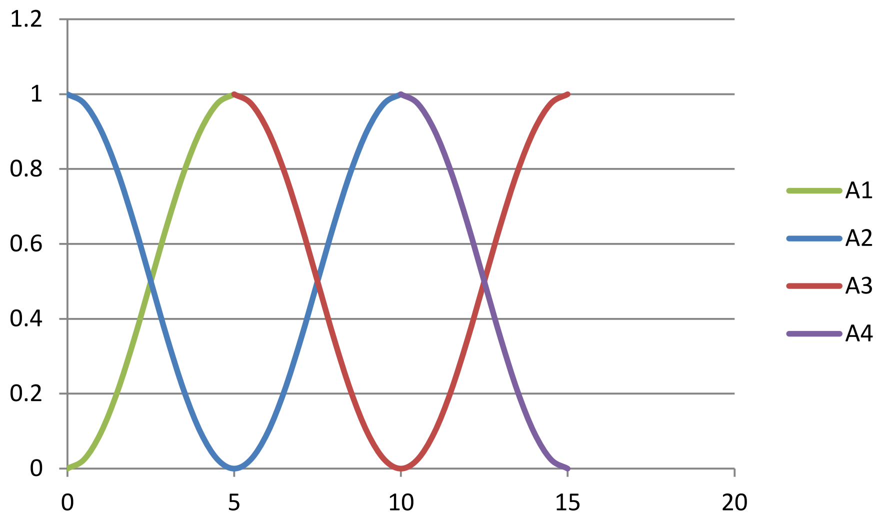


Figure 2. Cosine basic functions ($N = 16$, $n = 4$)

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left(R_{ij} - R_{ij}^{1}\right)^{2}}{N \times M}} \tag{26}$$

which implies that PSNR is the following:

$$\text{PSNR} = 20\log_{10}\frac{N_{lev} - 1}{\text{RMSE}} \tag{27}$$

4. Test Results

We compare our method with the classical F-transform compression method, but here no comparison is made with the one inspired to the Canny method used in [2].

For our tests we have considered the CVG-UGR image database extracting grey images of sizes 256 × 256 (cfr., http://decsai.ugr.es/cvg/dbimagenes/). For brevity, we only give the results for three images as Lena, Einstein and Leopard whose sources are given in Figures 3(a)-(c), respectively.

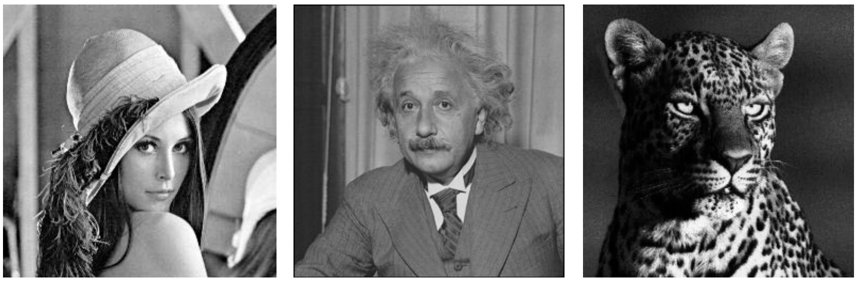
In Table 1, we show the PSNR of the F-transform and F^1-transform methods for some values of the compression rate in the image Lena.

We make the following remarks on Table 1:

- for weak compression rates the quality of the decoded image under the F^1-transform method is better than the one obtained with the F-transform method;
- for strong compression rates the quality of the images decoded in the two methods is similar;
- the difference between the two PSNR's in the two methods overcomes 0.1 for $\rho > 0.25$.

In Figure 4, we show the trend of the PSNR for the two methods.

In Figures 5(a)-(d) (resp., Figures 6(a)-(d)), we show the decoded images of Lena obtained by using the F-transform (resp., F^1-transform) for ρ = 0.0.0625, 0.16, 0.284444 and 0.444444, respectively.



(a) (b) (c)

Figure 3. (a) Lena; (b) Einstein; (c) Leopard.

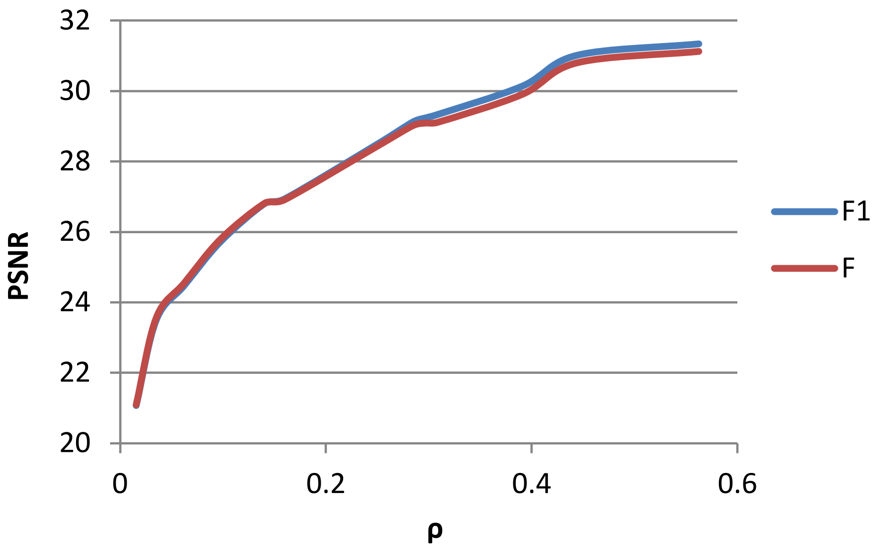


Figure 4. PSNR trend for the source image Lena.

Table 1. PSNR of the F-transform and F^1-transform methods for some values of the compression rate in the image Lena.

ρ	PSNR F-transform	PSNR F^1-transform	(PSNR F^1-tr) - (PSNR F-tr)
0.015625	21.088	21.071	−0.017
0.035156	23.558	23.541	−0.018
0.062500	24.551	24.544	−0.007
0.097656	25.791	25.796	0.005
0.140625	26.812	26.823	0.011
0.160000	26.912	26.941	0.029
0.250000	28.431	28.497	0.066
0.284444	29.012	29.125	0.113
0.297521	29.089	29.247	0.158
0.308642	29.108	29.339	0.231
0.390625	29.899	30.141	0.242
0.444444	30.800	31.023	0.223
0.562500	31.121	31.375	0.254

(a)

(b)

(c)

(d)

Figure 5. (a) F-tr under ρ = 0.0.0625; (b) F-tr under ρ = 0.16; (c) F-tr decoded (ρ = 0.284444); (d) F-tr decoded (ρ = 0.444444).

(a)

(b)

(c)

(d)

Figure 6. (a) F^1-tr decoded (ρ = 0.0.0625); (b) F^1-tr decoded (ρ = 0.16); (c) F^1-tr decoded (ρ = 0.284444); (d) F^1-tr decoded (ρ = 0.444444).

In Table 2 and Figure 7, we show the PSNR obtained using the F-transform and F^1-transform methods for some values of the compression rate in the image Einstein: this table confirms the same results obtained for the image Lena in Table 1.

In Figures 8(a)-(d) (resp., Figures 9(a)-(d)) we show the decoded images of Einstein obtained using the F-transform (resp., F^1-transform) method for ρ = 0.0.0625, 0.16, 0.284444 and 0.444444, respectively.

In Table 3 we show the PSNR values obtained using the F-transform and F^1-transform methods for some values of the compression rate in the image Leopard.

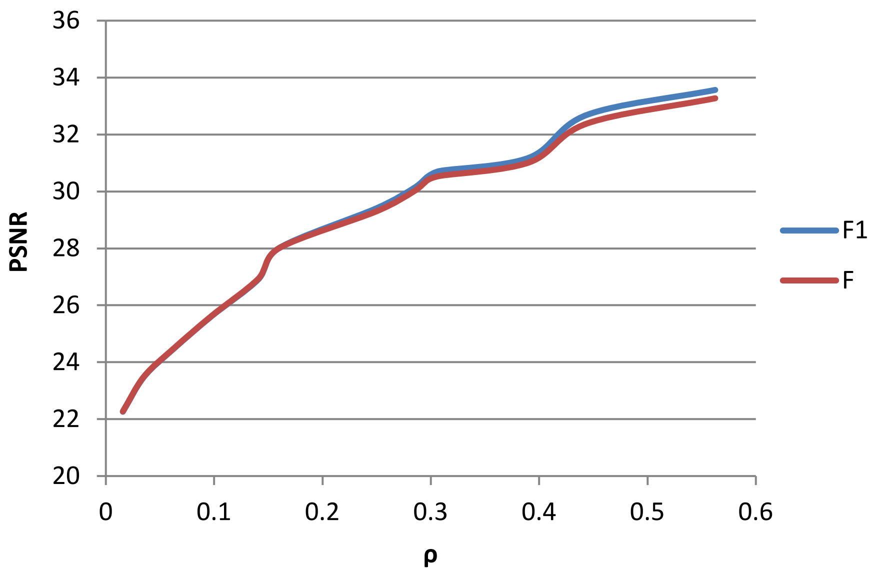


Figure 7. PSNR trend for the source image Einstein.

Table 2. PSNR results obtained for the source image Einstein.

ρ	PSNR F-transform	PSNR F^1-transform	(PSNR F^1-tr) - (PSNR F-tr)
0.015625	22.2701	22.2679	−0.0022
0.035156	23.4968	23.4952	−0.0016
0.062500	24.3781	24.3764	−0.0017
0.097656	25.6269	25.6265	−0.0004
0.140625	26.9260	26.9320	0.0006
0.160000	28.0048	28.0186	0.0138
0.250000	29.3003	29.4154	0.1151
0.284444	30.0018	30.1252	0.1234
0.297521	30.4054	30.5377	0.1323
0.308642	30.5415	30.7242	0.1827
0.390625	31.0126	31.1888	0.1762
0.444444	32.3841	32.6976	0.3135
0.562500	33.2661	33.5678	0.3017

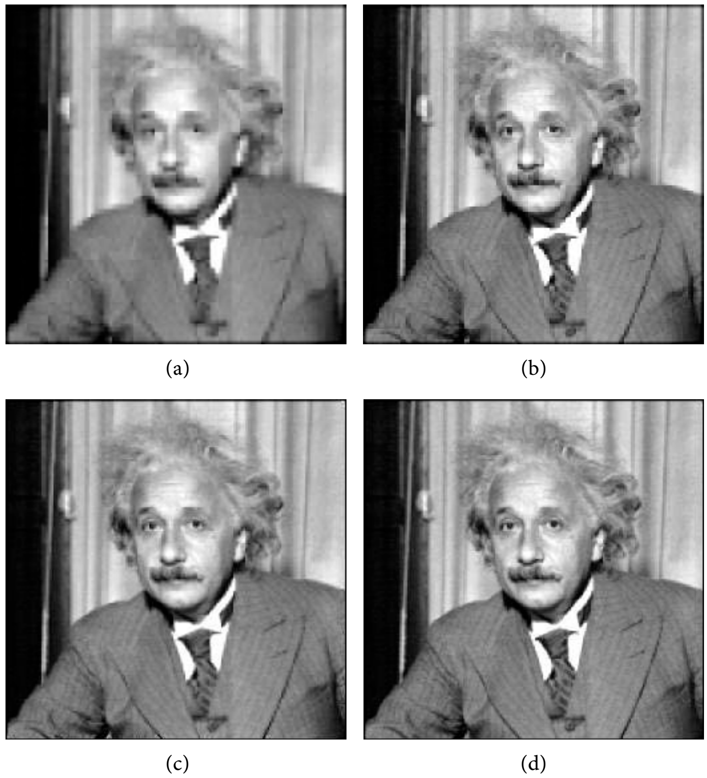


Figure 8. (a) F-tr decoded (ρ = 0.0.0625); (b) F-tr decoded (ρ = 0.16); (c) F-tr decoded (ρ = 0.284444); (d) F-tr decoded (ρ = 0.444444).

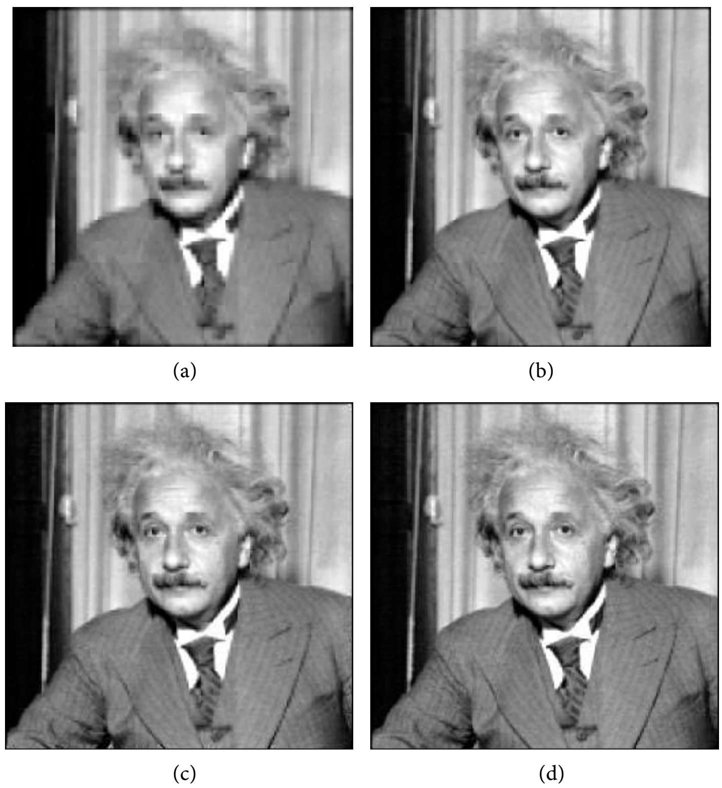


Figure 9. (a) F^1-tr decoded (ρ = 0.0.0625); (b) F^1-tr decoded (ρ = 0.16); (c) F^1-tr decoded (ρ = 0.284444); (d) F^1-tr decoded (ρ = 0.444444).

Table 3 confirms the results obtained for the images Lena and Einstein: the quality of the decoded image obtained by using the F^1-transform is better than the one obtained using the F-transform for weak compression rates. In Figure 10, we show the trend of the PSNR index obtained by using the two methods.

In Figures 11(a)-(d) (resp., Figures 12(a)-(d)), we show the decoded images of Leopard obtained by using the F-transform (resp., F^1-transform) method for ρ = 0.0.0625, 0.16, 0.284444, 0.444444, respectively.

In Figure 13, we show the trend of the difference of PSNR by varying the compression rate for all the images in the dataset above considered.

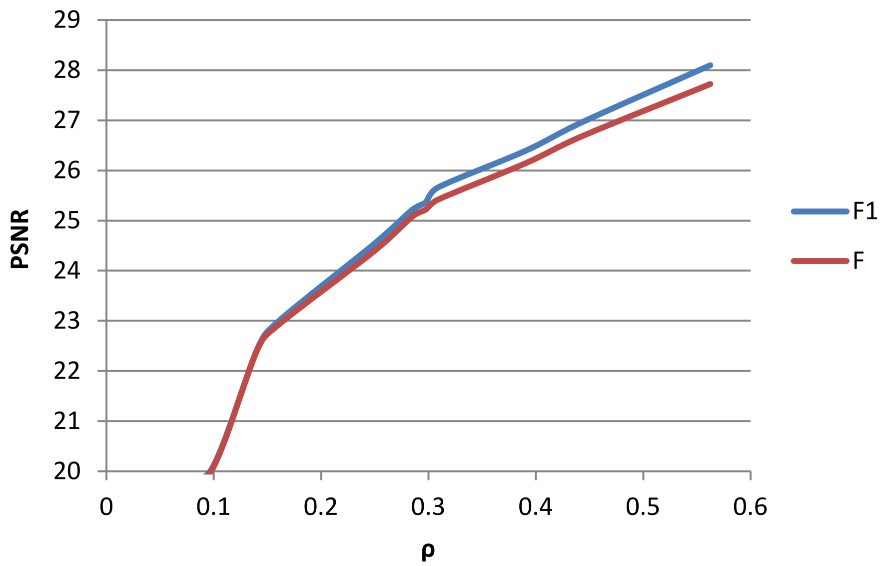


Figure 10. PSNR trend for the source image Leopard.

Table 3. PSNR results obtained for the source image Leopard.

ρ	PSNR F-transform	PSNR F^1-transform	(PSNR F^1-tr) - (PSNR F-tr)
0.015625	17.2997	17.3183	0.0186
0.035156	18.6483	18.6726	0.0243
0.062500	19.6883	19.7067	0.0184
0.097656	20.0131	20.0375	0.0244
0.140625	22.4336	22.4470	0.0134
0.160000	22.9203	22.9892	0.0689
0.250000	24.4041	24.5474	0.1433
0.284444	25.0750	25.2096	0.1346
0.297521	25.2229	25.3673	0.1444
0.308642	25.4181	25.6597	0.2416
0.390625	26.1470	26.3948	0.2478
0.444444	26.6971	26.9762	0.2791
0.562500	27.7235	28.0978	0.3743

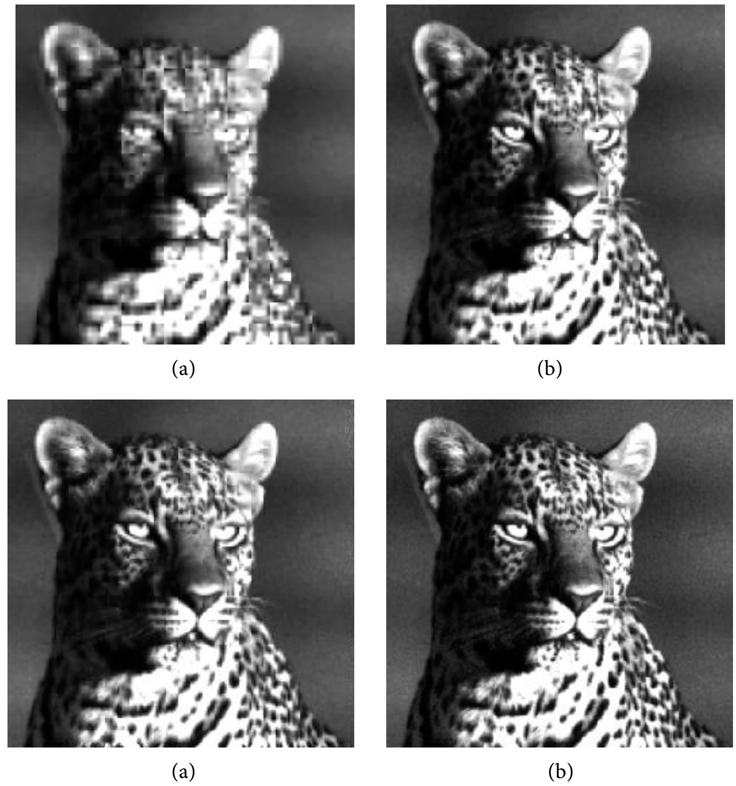


Figure 11. (a) F-tr decoded (ρ = 0.0.0625); (b) F-tr decoded (ρ = 0.16); (c) F-tr decoded (ρ=0.284444); (d) F-tr decoded (ρ=0.444444).

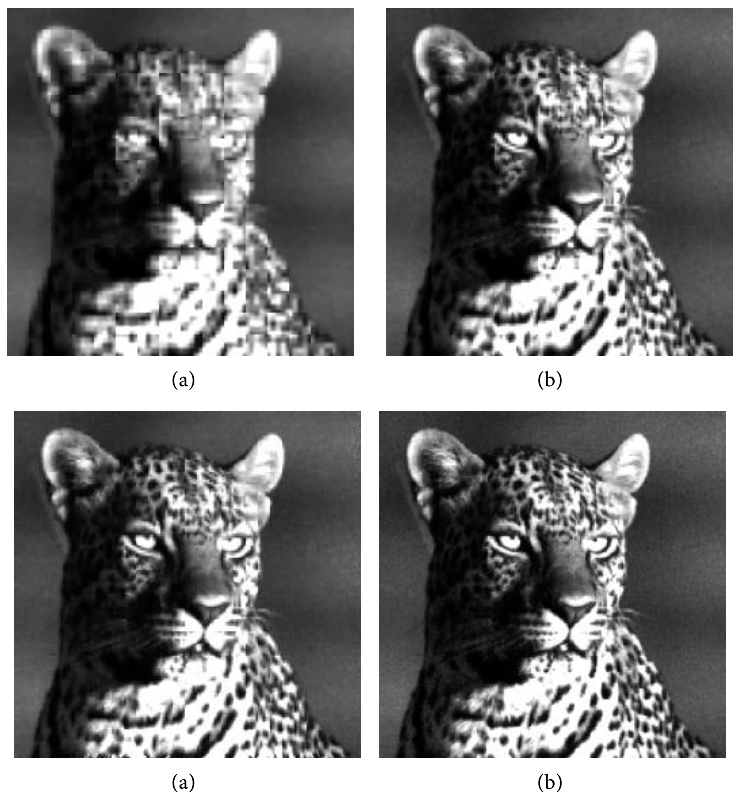


Figure 12. (a) F^1-tr decoded (ρ = 0.0.0625); (b) F^1-tr decoded (ρ = 0.16); (c) F^1-tr decoded (ρ = 0.284444); (d) F^1-tr decoded (ρ = 0.444444).

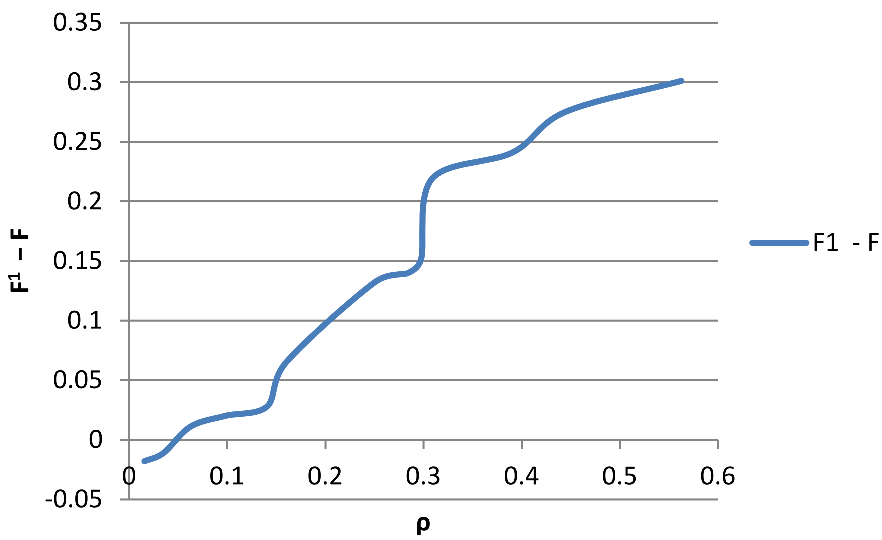


Figure 13. PSNR trend for all the images in the dataset considered.

Summarizing, we can say that the presence of the coefficients of the F^1-transform is negated by noise introduced during the strong compressions, while this effect increases considerably using weak compressions rates.

5. Conclusion

We give an image compression method based on the direct and inverse F^1-transform. The results show that the PSNR of the reconstructed images with the F^1-transform-based compression method is better than the one obtained with the F-transform-based compression. In the tested dataset of images, we find that the difference between the two corresponding PSNR values is greater than 0.1 (resp., 0.25) for $\rho = 0.25$ (resp., $\rho \approx 0.5$). In the next papers, we shall use the F^1-transform in data analysis problems.

Acknowledgements

We also accomplish this research under the auspices of the INDAM-GCNS, Italy. The last author acknowledges a partial support from the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

[1] Perfilieva, I. and Hodáková, P. (2013) F^1-Transform of Functions of Two Variables. *Proceedings of EUSFLAT* 2013, *Advances in Intelligent Systems Research*, Milano, **32**, 547-553.

[2] Perfilieva, I., Hodáková, P. and Hurtik, P. (2014) Differentiation by the F-Transform and Application for Edge Detection. *Fuzzy Sets and Systems*, **288**, 96-114. https://doi.org/10.1016/j.fss.2014.12.013

[3] Perfilieva, I. (2006) Fuzzy Transform: Theory and Application. *Fuzzy Sets and Systems*, **157**, 993-1023. https://doi.org/10.1016/j.fss.2005.11.012

[4] Di Martino, F., Loia, V. and Sessa, S. (2003) A Method in the Compres-

sion/Decompression of Images Using Fuzzy Equations and Fuzzy Similarities. *Proceedings of the* 10*th IFSA World Congress*, Istanbul, 30 June-2 July 2003, 524-527.

[5] Di Martino, F., Loia, V., Perfilieva, I. and Sessa, S. (2008) An Image Coding/Decoding Method Based on Direct and Inverse Fuzzy Transforms. *International Journal of Approximate Systems*, **48**, 110-131. https://doi.org/10.1016/j.ijar.2007.06.008

[6] Di Martino, F. and Sessa, S. (2007) Compression and Decompression of Images with Discrete Fuzzy Transforms. *Information Sciences*, **177**, 2349-2362. https://doi.org/10.1016/j.ins.2006.12.027

[7] Di Martino, F., Loia, V. and Sessa, S. (2010) A Segmentation Method for Images Compressed by Fuzzy Transforms. *Fuzzy Sets and Systems*, **161**, 56-74. https://doi.org/10.1016/j.fss.2009.08.002

[8] Di Martino, F., Loia, V. and Sessa, S. (2010) Fuzzy Transforms for Compression and Decompression of Colour Videos. *Information Sciences*, **180**, 3914-3931. https://doi.org/10.1016/j.ins.2010.06.030

[9] Di Martino, F., Hurlik, P., Perfilieva, I. and Sessa, S. (2014) A Color Image Reduction Based on Fuzzy Transforms. *Information Sciences*, **266**, 101-111. https://doi.org/10.1016/j.ins.2014.01.014

[10] Di Martino, F. and Sessa, S. (2013) Coding B-Frames on Color Videos with Fuzzy Transforms. *Advances on Fuzzy Systems*, **2013**, Article ID: 652429. https://doi.org/10.1155/2013/652429

[11] Di Martino, F., Loia, V. and Sessa, S. (2012) Fragile Watermarking Tamper Detection with Images Compressed by Fuzzy Transform. *Information Sciences*, **195**, 62-90. https://doi.org/10.1016/j.ins.2012.01.014

[12] Di Martino, F., Loia, V. and Sessa, S. (2013) Image Matching by Using Fuzzy Transform. *Advances on Fuzzy Systems*, **2013**, Article ID: 760705, 10 p.

[13] Loia, V. and Sessa, S. (2005) Fuzzy Relation Equations for Coding/Decoding Processes of Images and Videos. *Information Sciences*, **171**, 145-172. https://doi.org/10.1016/j.ins.2004.04.003

[14] Nobuhara, H., Pedrycz, W. and Hirota, K. (2000) Fast Solving Method of Fuzzy Relational Equation and Its Application to Lossy Image Compression. *IEEE Transactions on Fuzzy Systems*, **8**, 325-334. https://doi.org/10.1109/91.855920

[15] Nobuhara, H., Pedrycz, W. and Hirota, K. (2005) Relational Image Compression: Optimizations through the Design of Fuzzy Coders and YUV Color Space. *Soft Computing*, **9**, 471-479. https://doi.org/10.1007/s00500-004-0366-7

[16] Nobuhara, H., Hirota, K., Di Martino, F., Pedrycz, W. and Sessa, S. (2005) Fuzzy Relation Equations for Compression/Decompression Processes of Colour Images in the RGB and YUV Colour Spaces. *Fuzzy Optimization and Decision Making*, **4**, 235-246. https://doi.org/10.1007/s10700-005-1892-1

[17] Perfilieva, I. and De Baets, B. (2010) Fuzzy Transforms of Monotone Functions with Application to Image Compression. *Information Sciences*, **180**, 3304-3315. https://doi.org/10.1016/j.ins.2010.04.029

[18] Vajgl, M., Perfilieva, I. and Hodáková, P. (2012) Advanced F-Transform Based Image Fusion. *Advances in Fuzzy Systems*, **2012**, Article ID: 125086, 9 p. https://doi.org/10.1155/2012/125086

[19] Di Martino, F., Loia, V. and Sessa, S. (2010) Fuzzy Transforms Method and Attribute Dependency in Data Analysis. *Information Sciences*, **180**, 493-505. https://doi.org/10.1016/j.ins.2009.10.012

[20] Di Martino, F., Loia, V. and Sessa, S. (2011) Fuzzy Transforms Method in Prediction Data Analysis. *Fuzzy Sets and Systems*, **180**, 146-163. https://doi.org/10.1016/j.fss.2010.11.009

[21] Holcapek, M. (2011) A Smoothing Filter Based on Fuzzy Transform. *Fuzzy Sets and Systems*, **180**, 69-97. https://doi.org/10.1016/j.fss.2011.05.028

[22] Perfilieva, I., Novak, V. and Dvorak, A. (2008) Fuzzy Transforms in the Analysis of Data. *International Journal of Approximate Reasoning*, **40**, 26-46. https://doi.org/10.1016/j.ijar.2007.06.003

[23] Štěpnička, M. and Polakovič, O. (2009) A Neural Network Approach to the Fuzzy Transform. *Fuzzy Sets and Systems*, **160**, 1037-1047. https://doi.org/10.1016/j.fss.2008.11.029

[24] Štěpnička, M., Dvořák, A., Pavliska, V. and Vavříčková, L. (2011) A Linguistic Approach to Time Series Modeling with the Help of F-Transform. *Fuzzy Sets and Systems*, **180**, 164-184. https://doi.org/10.1016/j.fss.2011.02.017

[25] Troiano, L. and Ktriptani, P. (2011) Supporting Trading Strategies by Inverse Fuzzy Transform. *Fuzzy Sets and Systems*, **180**, 121-145. https://doi.org/10.1016/j.fss.2011.05.004

[26] Bede, B. and Rudas, I.J. (2011) Approximation Properties of Fuzzy Transforms. *Fuzzy Sets and Systems*, **180**, 20-40. https://doi.org/10.1016/j.fss.2011.03.001

[27] Perfilieva, I., Dankova, M. and Bede, B. (2011) Towards a Higher Degree F-Transform. *Fuzzy Sets Systems*, **180**, 3-19. https://doi.org/10.1016/j.fss.2010.11.002

[28] Di Martino, F., Loia, V. and Sessa, S. (2003) A Method for Coding/Decoding Images by Using Fuzzy Relation Equations. In: Bilgic, T., De Baets, B. and Kaynak, O., Eds., *Fuzzy Sets and Systems—IFSA* 2003, *Lecture Notes in Artificial Intelligence*, Vol. 2715, Springer, Berlin, 436-441. https://doi.org/10.1007/3-540-44967-1_52