First Order Fuzzy Transform for Images Compression

Ferdinando Di Martino¹, Salvatore Sessa², Irina Perfilieva³

¹Università degli Studi di Napoli Federico II, Dipartimento di Architettura, Napoli, Italy
²Università degli Studi di Napoli Federico II, Centro Interdipartimentale di Ricerca Calza Bini, Napoli, Italy
³University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, Ostrava, Czech Republic

Email: fdimarti@unina.it, sessa@unina.it, irina.perfilieva@osu.cz

Abstract
In this paper, we present a new image compression method based on the direct and inverse F¹-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 F¹-transform which is a generalization of the classical fuzzy transform [1] [2] identified as F⁰-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 F¹-transform (s ≥ 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 F¹-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 F¹-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 \( \rho = \frac{n(B) \times m(B)}{N(B) \times M(B)} \). 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 F\(^1\)-Transform

We recall the main concepts [2] that will be used in the sequel. We consider a set of points (called nodes) \( x_0, x_1, x_2, \ldots, x_n, n \geq 2 \) of \([a, b]\) such that \( a = x_0 < x_1 < x_2 < \cdots < x_n = b \). We say that the fuzzy sets \( A_1, \ldots, A_n : [a, b] \rightarrow [0, 1] \) form a generalized fuzzy partition of \([a, b]\), if for each \( k = 1, 2, \ldots, n \), there exist \( h'_k, h''_k \geq 0 \) 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](image_url)

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

4) \( h'_k = h''_k = \cdots = h''_{n+1} = h'_k = h' \) and \( A_k(x - x) = A_k(x + x) \) for each \( x \in [0, h'] \) and \( k = 2, \ldots, n \),

5) \( L_k(x) = A_{k+1}(x - h) \) and \( A_{k+1}(x) = A_k(x - h) \) for every \( x \in [x_k, x_{k+1}] \) and \( k = 2, \ldots, n \), then \( \{ A_1, \ldots, A_n \} \) is called an \((h, h')\) -uniform generalized fuzzy partition. In this case we can find a function \( A_0 : [-1, 1] \rightarrow [0, 1] \), 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 \( k = 1, 2, \ldots, n \):

\[
A_k(x) = \begin{cases} \frac{x - x_k}{h} & x \in [x_k - h, x_k + h] \\ 0 & \text{otherwise} \end{cases}
\] (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 \( \{ A_k \times B_l, k = 1, \ldots, n, l = 1, \ldots, m; n, m \geq 2 \} \), where \( A_k \times B_l \) is the product of the corresponding functions from the \( h_1 \)-uniform generalized fuzzy partition \( \{ A_1, \ldots, A_n \} \) of \([a, b]\) and from the \( h_2 \)-uniform generalized fuzzy partition \( \{ B_1, \ldots, B_m \} \) of \([c, d]\). Then we can say that \( \{ A_k \times B_l, k = 1, \ldots, n, l = 1, \ldots, m; n, m \geq 2 \} \) is an \( h \)-uniform generalized fuzzy partition of \([a, b] \times [c, d]\), where \( h = h_1, h_2 \). 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 \( f : [x_{k-1}, x_{k+1}] \rightarrow \mathbb{R} \) (reals) with weighted inner product:

\[
\langle f, g \rangle_h = \int_{x_{k-1}}^{x_{k+1}} f(x)g(x)A_k(x)dx
\]

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

\[
\langle f, g \rangle_{hl} = \int_{x_{k-1}}^{x_{k+1}} \int_{y_{l-1}}^{y_{l+1}} f(x,y)g(x,y)A_k(x)B_l(y)dxdy
\] (2)

Two function \( f, g \in L_2(A_k \times B_l) \) are orthogonal if \( \langle f, g \rangle_{hl} = 0 \). 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 \( \{ P^p_i(x) \}_{i=0, \ldots, p} \) and \( \{ Q^r_j(y) \}_{j=0, \ldots, r} \), re-
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 \( L_2^s(A_i \times B_j) \) of \( L_2(A_i \times B_j) \) having as orthogonal basis the following:

\[
S^0_{ij}(x, y) = P^0_i(x)Q^0_j(y) \quad \text{if } i = 0, \ldots, p; j = 0, \ldots, r; i + j \leq s
\]

where \( s \) is the maximum degree of polynomials \( P^0_i(x)Q^0_j(y) \). For \( s = 1 \), the orthogonal basis of the linear space \( L_2^1(A_i \times B_j) \) is the following:

\[
\begin{align*}
S^0_{00}(x, y) &= P^0_0(x)Q^0_0(y) = 1 \\
S^0_{10}(x, y) &= P^0_1(x)Q^0_0(y) = x - x_0 \\
S^0_{01}(x, y) &= P^0_0(x)Q^0_1(y) = y - y_0
\end{align*}
\]

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

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

\[
F^s_{ij}(x, y) = \sum_{0 \leq i + j \leq s} c^s_{ij} S^s_{ij}(x, y)
\]

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

\[
c^s_{ij} = \frac{\int_{x_{k-1}}^{x_k} \int_{y_{l-1}}^{y_l} f(x, y) S^s_{ij}(x, y) A_i(x) B_j(y) \, dx \, dy}{\int_{x_{k-1}}^{x_k} \int_{y_{l-1}}^{y_l} (S^s_{ij}(x, y))^2 A_i(x) B_j(y) \, dx \, dy}
\]

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

\[
\begin{align*}
S^0_{00}(x, y) &= P^0_0(x)Q^0_0(y) = 1 \\
S^0_{10}(x, y) &= P^0_1(x)Q^0_0(y) = x - x_0 \\
S^0_{01}(x, y) &= P^0_0(x)Q^0_1(y) = y - y_0
\end{align*}
\]

Let \( F^1_{ij} \) be the orthogonal projection of \( f \mid [x_{k-1}, x_k] \times [y_{l-1}, y_l] \) on \( L_2^1(A_i \times B_j) \) given point wise as

\[
F^1_{ij}(x, y) = \sum_{0 \leq i + j \leq 1} c^0_{ij} S^0_{ij}(x, y) = c^0_{ij} + c^0_{ij}(x - x_0) + c^0_{ij}(y - y_0)
\]

for every \((x, y) \in [a, b] \times [c, d]\), where the three coefficients \( c^0_{ij}, c^0_{ij}, c^0_{ij} \) are defined by Theorem 1:
Then the matrix $F_{nm}^{1} = \left( F_{nm}^{1}, \cdots, F_{nm}^{N} \right)$, 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 $\{ A_{k} \times B_{l}, k = 1, \cdots, n, l = 1, \cdots, m; n, m \geq 2 \}$. 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 $\hat{f}_{nm}^{1} : [a, b] \times [c, d] \to R$ as

$$\hat{f}_{nm}^{1}(x, y) = \frac{\sum_{k=1}^{n} \sum_{l=1}^{m} F_{nm}^{1}(x, y) A_{k}(x) B_{l}(y)}{A_{k}(x) B_{l}(y)}$$  \hspace{1cm} (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 $\{ (A_{k}(x), B_{l}(y)) \}, k = 1, \cdots, n, l = 1, \cdots, m; n, m \geq 2 \}$ 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 $(x, y) \in [a, b] \times [c, d]$:

$$f(x, y) - \hat{f}_{nm}^{1}(x, y) = O(h^{2})$$  \hspace{1cm} (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 $N \times M$, $R : (i, j) \in \{1, \cdots, N\} \times \{1, \cdots, M\} \to [0,1]$, $R(i, j) = R_{ij}$ being the normalized value of the pixel $P(i, j)$, that is $R(i, j) = P(i, j)/N_{ov}$ if $N_{ov}$ 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 $a = 1$, $b = N$,
Slightly modifying (8), then we can define the (discrete) F1-transform of R the matrix whose entries are defined as

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

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

$$c_{ij}^{00} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} R_y A_i (i) B_j (j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_i (i) B_j (j)} \tag{15}$$

$$c_{ij}^{01} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} R_y |j-l| A_i (i) B_j (j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_i (i) \sum_{j=1}^{N} B_j (j) (j-l)^2} \tag{16}$$

$$c_{ij}^{01} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} R_y |i-k| A_i (i) B_j (j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_i (i) \sum_{j=1}^{N} B_j (j) (j-l)^2} \tag{17}$$

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) F1-transform

$$R_{ij}^1 = \sum_{k=1}^{M} \sum_{l=1}^{N} R_{ij}^1 A_k (i) B_l (j) \tag{18}$$

We divide the image R of sizes $N \times M$ in sub-matrices $R^0$ of sizes $N (B) \times M (B)$, called blocks ([26] [28]), each compressed to a block $[R_{ij}^{1B}]_{x \in \mathbb{Z}} \in \mathbb{Z}$ of sizes $n(B) \times m(B)$, via the discrete F1-transform, as Formula (14), of components $R_{ij}^{1B}$ given by

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

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

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

$$c_{ij}^{01B} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} R_{ij}^B |i-k| A_i (i) B_j (j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_i (i) (i-k)^2 \sum_{j=1}^{N} B_j (j)} \tag{21}$$
The basic functions \( \{ A_1, \ldots, A_{n(B)} \} \) and \( \{ B_1, \ldots, B_{m(B)} \} \) form an \( h \)-uniform generalized uniform fuzzy partition of \([1, N(B)]\) and \([1, M(B)]\), respectively. They are generated by the basic functions \( A_k(x) = 0.5[1 + \cos(\pi x)] \) and \( B_l(y) = 0.5[1 + \cos(\pi y)] \), respectively. Then we have that

\[
A_k(x) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_k} (x - x_k) \right) & \text{if } x \in [x_k, x_{k+1}] \\
0 & \text{otherwise}
\end{cases}
\]

\[
A_k(x) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_k} (x - x_k) \right) & \text{if } x \in [x_{k+1}, x_k] \\
0 & \text{otherwise}
\end{cases}
\]

\[
A_n(x) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_k} (x - x_n) \right) & \text{if } x \in [x_{n-1}, x_n] \\
0 & \text{otherwise}
\end{cases}
\]

where \( n = n(B) \), \( h_k = (N(B) - 1) / (n - 1) \), \( x_k = 1 + h_k (k - 1), k = 2, \ldots, n - 1 \) and

\[
B_l(y) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_l} (y - y_l) \right) & \text{if } y \in [y_l, y_{l+1}] \\
0 & \text{otherwise}
\end{cases}
\]

\[
B_l(y) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_l} (y - y_l) \right) & \text{if } y \in [y_{l+1}, y_l] \\
0 & \text{otherwise}
\end{cases}
\]

\[
B_m(y) = \begin{cases} 
0.5 \left(1 + \cos \frac{\pi}{h_m} (y - y_m) \right) & \text{if } y \in [y_{m-1}, y_m] \\
0 & \text{otherwise}
\end{cases}
\]

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

The compressed block \( \left[ R^{IB}_{ij} \right]_{i=1}^{n(B)}\times_{j=1}^{m(B)} \) is decoded to a block \( \left[ R^{I}_y \right]_{i=1}^{n(B)}\times_{j=1}^{M(B)} \) of sizes \( N(B)\times M(B) \) by using the inverse \( F^1 \)-transform defined for every \( (i, j) \in \{1, \ldots, N_B\} \times \{1, \ldots, M_B\} \) as

\[
R^{I}_y = \frac{\sum_{j=1}^{M(B)} \sum_{i=1}^{n(B)} R^{IB}_{ij} A_k(i) B_l(j)}{\sum_{i=1}^{n(B)} \sum_{j=1}^{M(B)} A_k(i) B_l(j)}
\]

which approximates the original block \( R^B \). Making the union of all the decoded blocks \( R^B \), we obtain a fuzzy relation (denoted with) \( R \) of sizes \( N \times M \). Then we measure the RMSE (Root Mean Square Error) given by
Figure 2. Cosine basic functions \((N = 16, \ n = 4)\)

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

which implies that PSNR is the following:

\[
\text{PSNR} = 20 \log_{10} \frac{N_{\text{base}} - 1}{\text{RMSE}} \tag{27}
\]

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 \(256 \times 256\) (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 \(\rho = 0.00625, 0.16, 0.284444\) and 0.444444, respectively.
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.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>PSNR F-transform</th>
<th>PSNR $F^1$-transform</th>
<th>$(\text{PSNR } F^1 \text{-tr}) - (\text{PSNR } F \text{-tr})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.015625</td>
<td>21.088</td>
<td>21.071</td>
<td>$-0.017$</td>
</tr>
<tr>
<td>0.035156</td>
<td>23.558</td>
<td>23.541</td>
<td>$-0.018$</td>
</tr>
<tr>
<td>0.062500</td>
<td>24.551</td>
<td>24.544</td>
<td>$-0.007$</td>
</tr>
<tr>
<td>0.097656</td>
<td>25.791</td>
<td>25.796</td>
<td>$0.005$</td>
</tr>
<tr>
<td>0.140625</td>
<td>26.812</td>
<td>26.823</td>
<td>$0.011$</td>
</tr>
<tr>
<td>0.160000</td>
<td>26.912</td>
<td>26.941</td>
<td>$0.029$</td>
</tr>
<tr>
<td>0.250000</td>
<td>28.431</td>
<td>28.497</td>
<td>$0.066$</td>
</tr>
<tr>
<td>0.284444</td>
<td>29.012</td>
<td>29.125</td>
<td>$0.113$</td>
</tr>
<tr>
<td>0.297521</td>
<td>29.089</td>
<td>29.247</td>
<td>$0.158$</td>
</tr>
<tr>
<td>0.308642</td>
<td>29.108</td>
<td>29.339</td>
<td>$0.231$</td>
</tr>
<tr>
<td>0.390625</td>
<td>29.899</td>
<td>30.141</td>
<td>$0.242$</td>
</tr>
<tr>
<td>0.444444</td>
<td>30.800</td>
<td>31.023</td>
<td>$0.223$</td>
</tr>
<tr>
<td>0.562500</td>
<td>31.121</td>
<td>31.375</td>
<td>$0.254$</td>
</tr>
</tbody>
</table>
Figure 5. (a) F-tr under $\rho = 0.0625$; (b) F-tr under $\rho = 0.16$; (c) F-tr decoded ($\rho = 0.284444$); (d) F-tr decoded ($\rho = 0.444444$).

Figure 6. (a) $F^1$-tr decoded ($\rho = 0.0625$); (b) $F^1$-tr decoded ($\rho = 0.16$); (c) $F^1$-tr decoded ($\rho = 0.284444$); (d) $F^1$-tr decoded ($\rho = 0.444444$).
In Table 2 and Figure 7, we show the PSNR obtained using the F-transform and F1-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., F1-transform) method for $\rho = 0.0.0625, 0.16, 0.284444$ and $0.444444$, respectively.

In Table 3 we show the PSNR values obtained using the F-transform and F1-transform methods for some values of the compression rate in the image Leopard.

![Figure 7. PSNR trend for the source image Einstein.](image)

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

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>PSNR F-transform</th>
<th>PSNR F1-transform</th>
<th>(PSNR F1-tr) - (PSNR F-tr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.015625</td>
<td>22.2701</td>
<td>22.2679</td>
<td>−0.0022</td>
</tr>
<tr>
<td>0.035156</td>
<td>23.4968</td>
<td>23.4952</td>
<td>−0.0016</td>
</tr>
<tr>
<td>0.062500</td>
<td>24.3781</td>
<td>24.3764</td>
<td>−0.0017</td>
</tr>
<tr>
<td>0.097656</td>
<td>25.6269</td>
<td>25.6265</td>
<td>−0.0004</td>
</tr>
<tr>
<td>0.140625</td>
<td>26.9260</td>
<td>26.9320</td>
<td>0.0006</td>
</tr>
<tr>
<td>0.160000</td>
<td>28.0048</td>
<td>28.0186</td>
<td>0.0138</td>
</tr>
<tr>
<td>0.250000</td>
<td>29.3003</td>
<td>29.4154</td>
<td>0.1151</td>
</tr>
<tr>
<td>0.284444</td>
<td>30.0018</td>
<td>30.1252</td>
<td>0.1234</td>
</tr>
<tr>
<td>0.297521</td>
<td>30.4054</td>
<td>30.5377</td>
<td>0.1323</td>
</tr>
<tr>
<td>0.308642</td>
<td>30.5415</td>
<td>30.7242</td>
<td>0.1827</td>
</tr>
<tr>
<td>0.390625</td>
<td>31.0126</td>
<td>31.1888</td>
<td>0.1762</td>
</tr>
<tr>
<td>0.444444</td>
<td>32.3841</td>
<td>32.6976</td>
<td>0.3135</td>
</tr>
<tr>
<td>0.562500</td>
<td>33.2661</td>
<td>33.5678</td>
<td>0.3017</td>
</tr>
</tbody>
</table>
Figure 8. (a) F-tr decoded ($\rho = 0.0625$); (b) F-tr decoded ($\rho = 0.16$); (c) F-tr decoded ($\rho = 0.284444$); (d) F-tr decoded ($\rho = 0.444444$).

Figure 9. (a) F^1-tr decoded ($\rho = 0.0625$); (b) F^1-tr decoded ($\rho = 0.16$); (c) F^1-tr decoded ($\rho = 0.284444$); (d) F^1-tr decoded ($\rho = 0.444444$).
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 $\rho = 0.0.0625, 0.16, 0.284444, 0.444444$, 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.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>PSNR $F$-transform</th>
<th>PSNR $F^1$-transform</th>
<th>$(\text{PSNR } F^1\text{-tr}) - (\text{PSNR } F\text{-tr})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.015625</td>
<td>17.2997</td>
<td>17.3183</td>
<td>0.0186</td>
</tr>
<tr>
<td>0.035156</td>
<td>18.6483</td>
<td>18.6726</td>
<td>0.0243</td>
</tr>
<tr>
<td>0.062500</td>
<td>19.6883</td>
<td>19.7067</td>
<td>0.0184</td>
</tr>
<tr>
<td>0.097656</td>
<td>20.0131</td>
<td>20.0375</td>
<td>0.0244</td>
</tr>
<tr>
<td>0.140625</td>
<td>22.4336</td>
<td>22.4470</td>
<td>0.0134</td>
</tr>
<tr>
<td>0.160000</td>
<td>22.9203</td>
<td>22.9892</td>
<td>0.0689</td>
</tr>
<tr>
<td>0.250000</td>
<td>24.4041</td>
<td>24.5474</td>
<td>0.1433</td>
</tr>
<tr>
<td>0.284444</td>
<td>25.0750</td>
<td>25.2096</td>
<td>0.1346</td>
</tr>
<tr>
<td>0.297521</td>
<td>25.2229</td>
<td>25.3673</td>
<td>0.1444</td>
</tr>
<tr>
<td>0.308642</td>
<td>25.4181</td>
<td>25.6597</td>
<td>0.2416</td>
</tr>
<tr>
<td>0.390625</td>
<td>26.1470</td>
<td>26.3948</td>
<td>0.2478</td>
</tr>
<tr>
<td>0.444444</td>
<td>26.6971</td>
<td>26.9762</td>
<td>0.2791</td>
</tr>
<tr>
<td>0.562500</td>
<td>27.7235</td>
<td>28.0978</td>
<td>0.3743</td>
</tr>
</tbody>
</table>
Figure 11. (a) F-tr decoded ($\rho = 0.0625$); (b) F-tr decoded ($\rho = 0.16$); (c) F-tr decoded ($\rho = 0.284444$); (d) F-tr decoded ($\rho = 0.444444$).

Figure 12. (a) F¹-tr decoded ($\rho = 0.0625$); (b) F¹-tr decoded ($\rho = 0.16$); (c) F¹-tr decoded ($\rho = 0.284444$); (d) F¹-tr decoded ($\rho = 0.444444$).
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


---

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.
A wide selection of journals (inclusive of 9 subjects, more than 200 journals)
Providing 24-hour high-quality service
User-friendly online submission system
Fair and swift peer-review system
Efficient typesetting and proofreading procedure
Display of the result of downloads and visits, as well as the number of cited articles
Maximum dissemination of your research work

Submit your manuscript at: [http://papersubmission.scirp.org/](http://papersubmission.scirp.org/)
Or contact jsip@scirp.org