7. *F*-Transform

Irina Perfilieva

The theory of the *F*-transform is presented and discussed from the perspective of the latest developments and applications. Various fuzzy partitions are considered. The definition of the *F*-transform is given with respect to a generalized fuzzy partition, and the main properties of the *F*-transform are listed. The applications to image processing, namely image compression, fusion and edge detection, are discussed with sufficient technical details.

7.1 Fuzzy Modeling

Fuzzy modeling is still regarded as a modern technique with a nonclassical background. The goal of this chapter is to bridge standard mathematical methods and methods for the construction of fuzzy approximation models. We will present the theory of the *fuzzy transform* (the *F-transform*), which was introduced in [7.1] for the purpose of encompassing both classical (usually, integral) transforms and approximation models based on fuzzy IF–THEN rules (*fuzzy approximation models*). We start with an informal characterization of integral transforms, and from this discussion, we examine the similarities and differences among integral transforms, the *F*-transform, and fuzzy approximation models. An integral transform is performed using some kernel. The kernel is represented by a function of two variables and can be understood as a *collection of local factors* or closeness areas around elements of an original space. Each factor is then assigned an average value of a transforming object (usually, a function). Consequently, the transformed object is a new function defined on a space of *local factors*. The *F*-transform can be implicitly characterized by a *discrete* kernel that is associated with a finite collection of fuzzy subsets (local factors or closeness areas around chosen *nodes*) of an original space. We say that this collection establishes a *fuzzy partition* of the space. Then, similar to integral transforms, the *F*-transform assigns an average value of a transforming object to each fuzzy subset from the fuzzy partition of the space. Consequently, the *F*-transformed object is a finite vector of average values.

Similar to the *F*-transform, a fuzzy approximation model can also be implicitly characterized by a discrete kernel that establishes a fuzzy partition of an original space. Each element of the established fuzzy partition is a fuzzy set in the IF part (antecedent) of the re-

spective fuzzy IF–THEN rule. The rule characterizes a correspondence between an antecedent and an average value of a transforming object (singleton model) or a fuzzy subset of a space of object values (fuzzy set model).

To emphasize the differences among integral transforms, the F-transform, and fuzzy approximation models, we note that the last two are actually finite collections of local descriptions of a considered object. Each collection produces a global description of the considered object in the form of the direct F-transform or the system of fuzzy IF–THEN rules.

The idea of producing collections of local descriptions by fuzzy IF–THEN rules originates from the early works of *Zadeh* [7.2–5] and from the *Takagi–Sugeno* [7.6] approximation models.

Similar to the conventional integral transforms (the Fourier and Laplace transforms, for example), the F-transform performs a transformation of an original universe of functions into a universe of their *skeleton models* (vectors of F-transform components) for which further computations are easier (see, e.g., an application to the initial value problem with fuzzy initial conditions [7.7]). In this respect, the F-transform can be as useful in applications as traditional transforms (see applications to image compression [7.8, 9] and time series processing [7.10–14], for example). Moreover, sometimes the F-transform can be more efficient than its counterparts; see the details below.

The structure of this chapter is as follows. In Sect. 7.2, we consider various fuzzy partitions: uniform and with and without the Ruspini condition, among others; in Sect. 7.3, definitions of the F-transforms (direct and inverse) and their main properties are considered; in Sect. 7.4, the discrete F-transform is defined; in Sect. 7.5, the direct and inverse F-transform of a function of two variables is introduced; in Sect. 7.6, a higher degree F-transform is considered; in Sect. 7.7, applications of the F-transform and F^1-transform to image processing are discussed.

7.2 Fuzzy Partitions

In this section, we present a short overview of various fuzzy partitions of a universe in which transforming objects (functions) are defined. As we learned from Sect. 7.1, a fuzzy partition is a finite collection of fuzzy subsets of the universe that determines a discrete kernel and thus a respective transform. Therefore, we have as many F-transforms as fuzzy partitions.

7.2.1 Fuzzy Partition with the Ruspini Condition

The *fuzzy partition with the Ruspini condition* (7.1) (simply, *Ruspini partition*) was introduced in [7.1]. This condition implies normality of the respective fuzzy partition, i. e., the *partition-of-unity*. It then leads to a simplified version of the inverse F-transform. In later publications [7.15, 16], the Ruspini condition was weakened to obtain an additional degree of freedom and a better approximation by the inverse F-transform.

Definition 7.1

Let $x_1 < \cdots < x_n$ be fixed nodes within $[a, b]$ such that $x_1 = a, x_n = b$ and $n \geq 2$. We say that the fuzzy sets $A_1, \ldots, A_n$, identified with their membership functions defined on $[a, b]$, establish a Ruspini partition of $[a, b]$ if they fulfill the following conditions for $k = 1, \ldots, n$:

1. $A_k : [a, b] \to [0, 1]\,, \quad A_k(x_k) = 1$
2. $A_k(x) = 0$ if $x \notin (x_{k-1}, x_{k+1})$, where for uniformity of notation, we set $x_0 = a$ and $x_{n+1} = b$
3. $A_k(x)$ is continuous
4. $A_k(x)$, for $k = 2, \ldots, n$, strictly increases on $[x_{k-1}, x_k]$ and $A_k(x)$ for $k = 1, \ldots, n-1$, strictly decreases on $[x_k, x_{k+1}]$
5. for all $x \in [a, b]$,

$$\sum_{k=1}^{n} A_k(x) = 1\,. \tag{7.1}$$

The condition (7.1) is known as the Ruspini condition. The membership functions $A_1, \ldots, A_n$ are called the *basic functions*. A point $x \in [a, b]$ is *covered* by the basic function A_k if $A_k(x) > 0$.

The shape of the basic functions is not predetermined and therefore, it can be chosen according to additional requirements (e.g., smoothness). Let us give examples of various fuzzy partitions with the Ruspini condition. In Fig. 7.1, two such partitions with triangular and cosine basic functions are shown. The following formulas represent generic fuzzy partitions with the

Ruspini condition and triangular functions

$$A_1(x) = \begin{cases} 1 - \dfrac{(x - x_1)}{h_1}, & x \in [x_1, x_2] \,, \\ 0 \,, & \text{otherwise} \,, \end{cases}$$

$$A_k(x) = \begin{cases} \dfrac{(x - x_{k-1})}{h_{k-1}}, & x \in [x_{k-1}, x_k] \,, \\ 1 - \dfrac{(x - x_k)}{h_k}, & x \in [x_k, x_{k+1}] \,, \\ 0 \,, & \text{otherwise} \,, \end{cases}$$

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

where $k = 2, \ldots n-1$ and $h_k = x_{k+1} - x_k$.

We say that a Ruspini partition of $[a, b]$ is *h-uniform* if its nodes $x_1, \ldots, x_n$, where $n \geq 3$, are equidistant, i. e., $x_k = a + h(k-1)$, for $k = 1, \ldots, n$, where $h = (b-a)/(n-1)$, and the two additional properties are met:

6. $A_k(x_k - x) = A_k(x_k + x)$, for all $x \in [0, h]$, $k = 2, \ldots, n-1$,
7. $A_k(x) = A_{k-1}(x - h)$, for all $k = 2, \ldots, n-1$ and $x \in [x_k, x_{k+1}]$, and $A_{k+1}(x) = A_k(x - h)$, for all $k = 2, \ldots, n-1$ and $x \in [x_k, x_{k+1}]$.

7.2.2 Fuzzy Partitions with the Generalized Ruspini Condition

Fuzzy partitions with the generalized Ruspini condition were introduced in [7.15]. The generalization consists in replacing *partition-of-unity* (7.1) by *fuzzy r-partition* (7.2). This type of partition was investigated in [7.15, 17], where the focus was on smoothing or filtering data using the inverse F-transform. The following definition is taken from [7.15].

Definition 7.2

Let $r \geq 1$ and $n \geq 2$ be fixed integers such that $r \leq n$. Let $a = x_1 < \cdots < x_n = b$ be nodes within $[a, b]$, and let $x_{1-r} < \cdots < x_0 < a$ and $b < x_{n+1} < \cdots < x_{n+r}$ be nodes outside of $[a, b]$. A fuzzy r-partition of $[a, b]$ is a family of $n + 2r - 2$ continuous, normal, convex fuzzy sets

$$A_{2-r}^{(r)}, \ldots, A_1^{(r)}, \ldots, A_n^{(r)}, \ldots, A_{n+r-1}^{(r)}$$

such that the following conditions are fulfilled:

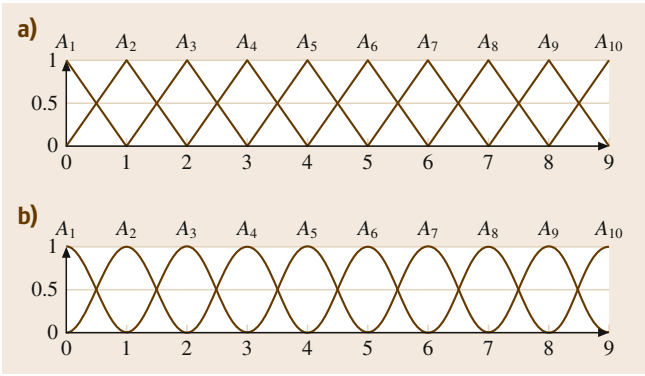

Fig. 7.1a,b Two Ruspini partitions with triangular **(a)** and cosine basic functions **(b)**

1. For $k = 1, \ldots, n$, $A_k^{(r)}$ is a continuous function on $[a, b]$ such that $A_k^{(r)}(x_k) = 1$ and $A_k^{(r)}(x) = 0$ for $x \notin [\max(x_{k-r}, a), \min(x_{k+r}, b)]$
2. For $k = 1, \ldots, n$, $A_k^{(r)}$ is increasing on $[\max(x_{k-r}, a), x_k]$ and decreasing on $[x_k, \min(x_{k+r}, b)]$
3. For $k = -r+2, \ldots, 0$, $A_k^{(r)}$ is decreasing on $[\max(x_k, a), x_{k+r}]$
4. For $k = n+1, \ldots, n+r-1$, $A_k^{(r)}$ is increasing on $[x_{k-r}, \min(x_k, b)]$
5. For all $x \in [a, b]$, the following *partition-of-r* condition holds

$$\sum_{k=-r+2}^{n+r-1} A_k^{(r)}(x) = r \,. \tag{7.2}$$

If $r = 1$, then a fuzzy r-partition in the sense of Definition 7.2 becomes the standard fuzzy partition in the sense of Definition 7.1, i. e., the *partition-of-unity*. In Fig. 7.2, the fuzzy 2-partition with triangular basic functions is shown.

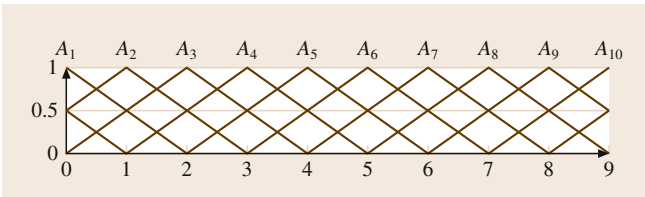

Fig. 7.2 An example of a fuzzy 2-partition with triangular basic functions

7.2.3 Generalized Fuzzy Partitions

A *generalized fuzzy partition* appeared in [7.16] in connection with the notion of the higher degree F-transform. Its even weaker version was implicitly introduced in [7.18] with the purpose of meeting the requirements of image compression. We summarize both these notions and propose the following definition.

Definition 7.3

Let $[a,b]$ be an interval on $\mathbb{R}$, $n \geq 2$, and let $x_0, x_1, \ldots, x_n, x_{n+1}$ be nodes such that

$$a = x_0 \leq x_1 < \cdots < x_n \leq x_{n+1} = b\,.$$

We say that the fuzzy sets

$$A_1, \ldots, A_n : [a,b] \to [0,1]$$

constitute a *generalized fuzzy partition* of $[a,b]$ if for every $k = 1, \ldots, n$ 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 three conditions are fulfilled:

1. (locality) – $A_k(x) > 0$ if $x \in (x_k - h'_k, x_k + h''_k)$, and $A_k(x) = 0$ if $x \in [a,b] \setminus [x_k - h'_k, x_k + h''_k]$
2. (continuity) – A_k is continuous on $[x_k - h'_k, x_k + h''_k]$
3. (covering) – for $x \in [a,b]$, $\sum_{k=1}^{n} A_k(x) > 0$.

It is important to remark that by conditions of *locality* and *continuity*,

$$\int_a^b A_k(x)\mathrm{d}x > 0\,.$$

An (h,h')-*uniform* generalized fuzzy partition of $[a,b]$ is defined for equidistant nodes

$$x_k = a + h(k-1)\,, k = 1, \ldots, n\,,$$

where $h = (b-a)/(n-1)$, $h' > h/2$ and two additional properties are satisfied:

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

An (h,h')-uniform generalized fuzzy partition of $[a,b]$ can also be defined using the *generating function* $A_0 : [-1,1] \to [0,1]$, which is assumed to be *even*, continuous, and positive everywhere except for on boundaries, where it vanishes. (The function $A_0 : [-1,1] \to \mathbb{R}$ is even if for all $x \in [0,1]$, $A_0(-x) = A_0(x)$.) Then, basic functions A_k of an (h,h')-uniform generalized fuzzy partition are shifted copies of A_0 in the sense that

$$A_1(x) = \begin{cases} A_0\left(\dfrac{x - x_1}{h'}\right), & x \in [x_1, x_1 + h']\,, \\ 0\,, & \text{otherwise}\,, \end{cases}$$

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

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

$$A_n(x) = \begin{cases} A_0\left(\dfrac{x - x_n}{h'}\right), & x \in [x_n - h', x_n]\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{7.3}$$

As an example, we note that the function $A_0(x) = 1 - |x|$ is a generating function for all uniform triangular partitions. The difference between them is in parameters h

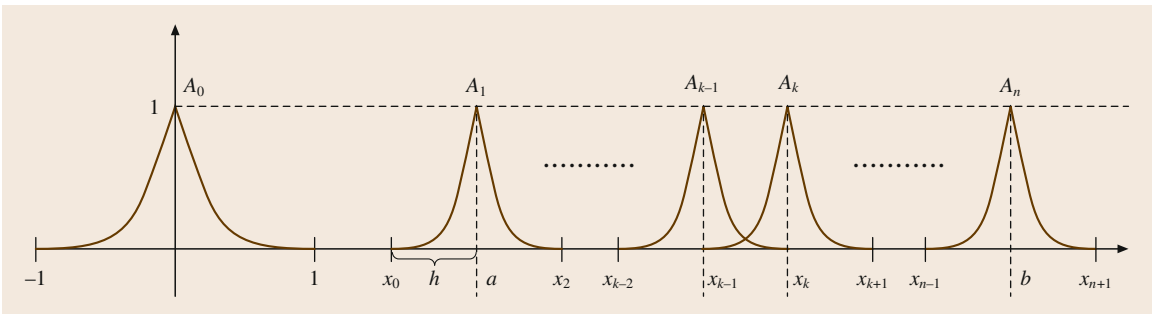

Fig. 7.3 Generating function A_0 of an h-uniform generalized fuzzy partition (after [7.19])

and h'. An (h,h)-uniform generalized fuzzy partition is simply called an h-uniform one (Fig. 7.3).

Remark 7.1
A generalized fuzzy partition can also be considered in connection with radial membership functions; see [7.20]. In this case, every basic function has a generic representation in terms of a kernel $\varphi : \mathbb{R}^+ \to \mathbb{R}$ such that

$$A_k(x) = \varphi(\|x - x_k\|_2) , \quad k = 1, \ldots, n .$$

7.3 Fuzzy Transform

The F-transform establishes a correspondence between a set of continuous functions on an interval of real numbers and the set of n-dimensional (real) vectors. Each component of the resulting vector is a weighted local mean of a corresponding function over an area covered by a corresponding basic function. The vector of the F-transform components is a simplified representation of an original function that can be used instead of the original function in many applications. Among them, let us mention applications to image compression [7.8, 9], image fusion, image reduction, time series processing [7.10–14], and the initial value problem with fuzzy initial conditions [7.7].

7.3.1 Direct F-Transform

In this section, we give the definition of the F-transform according to [7.1] and recall the main properties of it. We assume that the universe is an interval $[a,b]$ and $x_1 < \cdots < x_n$ are fixed nodes from $[a,b]$ such that $x_1 = a$, $x_n = b$ and $n \geq 2$. Let us formally extend the set of nodes by $x_0 = a$ and $x_{n+1} = b$. Let $A_1, \ldots, A_n$ be the basic functions that form a fuzzy partition of $[a,b]$ according to Definition 7.3. Let $C([a,b])$ be the set of continuous functions on the interval $[a,b]$. The following definition introduces the fuzzy transform of a function $f \in C([a,b])$.

Definition 7.4
Let $A_1, \ldots, A_n$ be the basic functions that form a generalized fuzzy partition of $[a,b]$ and f be any function from $C([a,b])$. We say that the n-tuple of real numbers $\mathbf{F}[f] = (F_1, \ldots, F_n)$ given by

$$F_k = \frac{\int_a^b f(x)A_k(x)\mathrm{d}x}{\int_a^b A_k(x)\mathrm{d}x} , \qquad k = 1, \ldots, n , \tag{7.4}$$

is the (integral) F-transform of f with respect to $A_1, \ldots, A_n$.

The elements $F_1, \ldots, F_n$ are called the *components of the F-transform*. If $A_1, \ldots, A_n$ is an h-uniform Ruspini partition, then (7.4) may be simplified as follows,

$$F_1 = \frac{2}{h}\int_{x_1}^{x_2} f(x)A_1(x)\mathrm{d}x ,$$

$$F_n = \frac{2}{h}\int_{x_{n-1}}^{x_n} f(x)A_n(x)\mathrm{d}x ,$$

$$F_k = \frac{1}{h}\int_{x_{k-1}}^{x_{k+1}} f(x)A_k(x)\mathrm{d}x , \quad k = 2, \ldots, n-1 . \tag{7.5}$$

The following is a list of some properties of the F-transform of f with respect to a generalized fuzzy partition of $[a,b]$:

(a) If for all $x \in [a,b]$, $f(x) = C$, then $F_k = C$, $k = 1, \ldots, n$.
(b) If $f = \alpha g + \beta h$, then $\mathbf{F}[f] = \alpha\mathbf{F}[g] + \beta\mathbf{F}[h]$.
(c) If $[c,d] = \{f(x) \mid x \in [a,b]\}$, then $F_k = \min_{[c,d]} \int_a^b (f(x) - y)^2 A_k(x)\mathrm{d}x$, $k = 1, \ldots, n$.
(d) If f is twice continuously differentiable on $[a,b]$, then $F_k = f(x_k) + O(h^2)$, $k = 1, \ldots, n$. (This is true for an h-uniform Ruspini partition of $[a,b]$ only. A similar estimation of the F-transform component F_k as a linear combination of $f(x_k - r + 1), \ldots, f(x_k), \ldots, f(x_k + r - 1)$ can be established for a fuzzy r-partition [7.15].)
(e) If a generalized fuzzy partition is (h,h')-uniform, then for each $k = 1, \ldots, n-1$,

$$|f(t) - F_k| \leq 2\omega(\tilde{h}, f) ,$$
$$|f(t) - F_{k+1}| \leq 2\omega(\tilde{h}, f) ,$$

where $\tilde{h} = \max(h, h')$, $t \in [x_k, x_k + \tilde{h}]$, and

$$\omega(\tilde{h}, f) = \max_{|\delta| \leq \tilde{h}} \max_{x \in [a, b-\delta]} |f(x + \delta) - f(x)| . \tag{7.6}$$

(f)

$$\int_a^b f(x)\mathrm{d}x = h\left(\frac{F_1}{2} + \frac{F_n}{2} + \sum_{k=2}^{n-1} F_k\right) .$$

(This is true for an h-uniform Ruspini partition of $[a, b]$ only.)

7.3.2 Inverse *F*-Transform

It is clear that an original nonconstant function f cannot be precisely reconstructed from its F-transform $\mathbf{F}[f]$ because we lose information when passing from f to $\mathbf{F}[f]$. However, the inverse F-transform $\hat{f}$ that can be reconstructed (using the inversion formula (7.7)) approximates f in such a way that universal convergence can be established.

Definition 7.5

Let $A_1, \ldots, A_n$ be the basic functions that form a generalized fuzzy partition of $[a, b]$ and f be a function from $C([a, b])$. Let $\mathbf{F}[f] = (F_1, \ldots, F_n)$ be the F-transform of f with respect to $A_1, \ldots, A_n$. Then, the function $\hat{f} : [a, b] \to \mathbb{R}$ represented by

$$\hat{f}(x) = \frac{\sum_{k=1}^{n} F_k A_k(x)}{\sum_{k=1}^{n} A_k(x)} , \qquad (7.7)$$

is called *the inverse F-transform*.

Remark 7.2

If a fuzzy partition of $[a, b]$ fulfills the generalized Ruspini condition (7.2) with $r \geq 1$, then the inversion formula (7.7) can be simplified to

$$\hat{f}(x) = \frac{1}{r} \sum_{k=1}^{n} F_k A_k(x)$$

or to (in the case of the Ruspini partition for which $r = 1$)

$$\hat{f}(x) = \sum_{k=1}^{n} F_k A_k(x) .$$

The following theorem demonstrates that the inverse F-transform $\hat{f}$ can approximate a continuous function f with arbitrary precision. Thus, it explains why the F-transform has convincing applications in various fields, including image and time series processing, and data mining [7.21]. In Fig. 7.4, we illustrate how the inverse F-transform approximates the function $10\mathrm{e}^{-(x-\pi)^2}$.

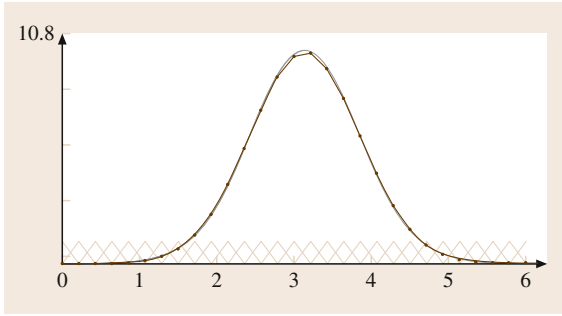

Fig. 7.4 The function $f(x) = 10\mathrm{e}^{-(x-\pi)^2}$ (*gray*) and its inverse F-transform (*brown*) with respect to the uniform Ruspini partition of $[0, 6]$ by 29 triangular-shaped basic functions. The F-transform components are marked by *small circles*

Theorem 7.1

Let f be a continuous function on $[a, b]$. Then, for any $\varepsilon > 0$, there exist n_ε and a generalized fuzzy partition $A_1, \ldots, A_{n_\varepsilon}$ of $[a, b]$ such that for all $x \in [a, b]$,

$$eq8|f(x) - \hat{f}_\varepsilon(x)| \leq \varepsilon , \qquad (7.8)$$

where $\hat{f}_\varepsilon$ is the inverse F-transform of f with respect to the fuzzy partition $A_1, \ldots, A_{n_\varepsilon}$.

From Theorem 7.2, which is given below, we learn that for a pointwise approximation (as in Theorem 7.1), it is sufficient to compute the F-transform with respect to the simplest triangular fuzzy partition. Therefore, almost all applications of the F-transform are based on this type of partition.

Theorem 7.2

Let f be any continuous function on $[a, b]$, and let $A'_1, \ldots, A'_n$ and $A''_1, \ldots, A''_n$, for $n \geq 3$, be the basic functions that form different (h, h')-uniform generalized fuzzy partitions of $[a, b]$. Let $\hat{f}'$ and $\hat{f}''$ be the two inverse F-transforms of f with respect to different sets of basic functions $A'_1, \ldots, A'_n$ or $A''_1, \ldots, A''_n$. Then, for arbitrary $x \in [a, b]$,

$$|\hat{f}'(x) - \hat{f}''(x)| \leq 4\omega(\tilde{h}, f) ,$$

where $h = \frac{b-a}{n-1}$, $\tilde{h} = \max(h, h')$ and $\omega(\tilde{h}, f)$ is the modulus of continuity (7.6) of f on the interval $[a, b]$.

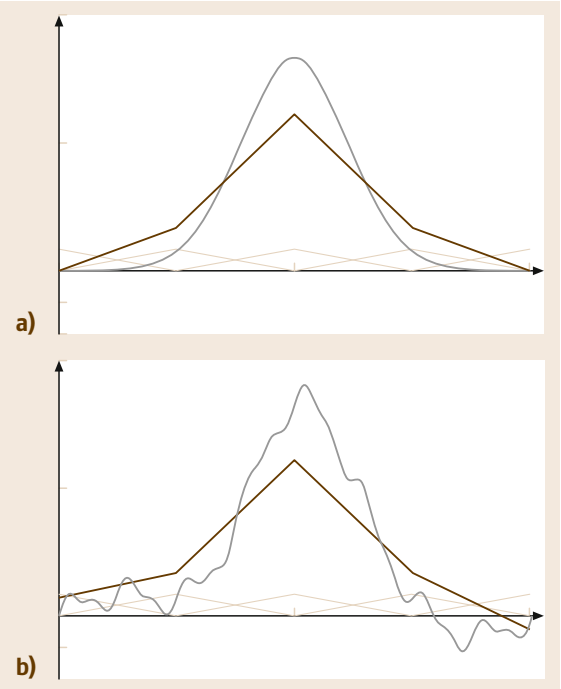

Fig. 7.5 **(a)** Function $f(x) = 10e^{-(x-\pi)^2}$ (*gray*) and its inverse F-transform (*brown*) with respect to the Ruspini partition given by the triangular-shaped basic functions $A_1, \ldots, A_5$ (*gray*). **(b)** Noisy function $f + s$ (*gray*), where $s(x) = \sin(2x) + 0.6\sin(8x) + 0.3\sin(16x)$, and its inverse F-transform (*brown*) with respect to the same fuzzy partition. Both inverse F-transforms $\hat{f}$ and $\widehat{f+s}$ are equal on $[x_2, x_4]$

The proofs of Theorems 7.1 and 7.2 can be obtained from the respective proofs in [7.22, Theorems 2 and 3] after some necessary changes caused by the usage of the generalized fuzzy partition.

Below, we list some properties of the inverse F-transform $\hat{f}$ of f that were considered and proved in [7.1, 15, 23]. If not specially mentioned, it is assumed that the F-transform is computed with respect to a generalized fuzzy partition of $[a, b]$:

(a) If for all $x \in [a, b]$, $f(x) = C$, then $\hat{f}(x) = C$
(b) If $f = \alpha g + \beta h$, then $\hat{f} = \alpha\hat{g} + \beta\hat{h}$
(c) $\int_a^b f(x)\mathrm{d}x = \int_a^b \hat{f}(x)\mathrm{d}x$ (This is true for the fuzzy r-partition ($r \geq 1$) of $[a, b]$ only.)
(d) Let $A_1, \ldots, A_n$ be an h-uniform Ruspini partition of $[a, b]$, where $h = (b-a)/(n-1)$ and $n > 3$. Let $s : [a, b] \to \mathbb{R}$ be a continuous function such that one of the following two conditions are fulfilled:
   (i) s is $2h$-periodical and for all $x \in [0, h]$, $s(x_k - x) = -s(x_k + x)$, where $k = 2, \ldots, n-1$
   (ii) s is h-periodical and $\int_{x_{k-1}}^{x_k} s(x)\mathrm{d}x = 0$, where $k = 2, \ldots, n-1$.

   Then, for $x \in [x_2, x_{n-1}]$,

$$\hat{f} = \widehat{f+s}\,.$$

The last property is known as *noise removal*. This phrase implies that both functions f (non-noisy) and $f + s$ (noisy) have the same inverse F-transform. The noise is represented by s and characterized by conditions (i) or (ii). We illustrate this property in Fig. 7.5.

7.4 Discrete *F*-Transform

The discrete case of the F-transform, for which an original function f is defined (may be computed) on a finite set $P = \{p_1, \ldots, p_l\} \subseteq [a, b]$, was introduced in [7.1]. We will adapt the mentioned definition to the case of a generalized fuzzy partition of $[a, b]$.

We assume that the domain P of the function f is *sufficiently dense with respect to the fixed partition*, i. e.,

$$(\forall k)(\exists j)A_k(p_j) > 0\,.$$

Then, the (discrete) F-transform of f is defined as follows.

Definition 7.6

Let $A_1, \ldots, A_n$, for $n > 2$, be the basic functions that form a generalized fuzzy partition of $[a, b]$, and let function f be defined on the set $P = \{p_1, \ldots, p_l\} \subseteq [a, b]$, which is sufficiently dense with respect to the partition. We say that the n-tuple of real numbers $(F_1, \ldots, F_n)$ is the discrete F-transform of f with respect to $A_1, \ldots, A_n$ if

$$F_k = \frac{\sum_{j=1}^{l} f(p_j)A_k(p_j)}{\sum_{j=1}^{l} A_k(p_j)}\,. \tag{7.9}$$

It is not difficult to demonstrate that the components of the discrete F-transform have similar properties to those listed in Sect. 7.3.1.

In the discrete case, we define the inverse F-transform on the same set P on which the original function is defined.

Definition 7.7

Let $A_1, \ldots, A_n$, for $n > 2$, be the basic functions that form a generalized fuzzy partition of $[a, b]$, and let function f be defined on the set $P = \{p_1, \ldots, p_l\} \subseteq [a, b]$, which is sufficiently dense with respect to the partition. Moreover, let $\mathbf{F}[f] = (F_1, \ldots, F_n)$ be the discrete F-transform of f w.r.t. $A_1, \ldots, A_n$. Then, the function $\hat{f} : P \to \mathbb{R}$ represented by

$$\hat{f}(p_j) = \frac{\sum_{k=1}^{n} F_k A_k(p_j)}{\sum_{k=1}^{n} A_k(p_j)} \tag{7.10}$$

is *the inverse discrete F-transform* of f.

Remark 7.3

If a fuzzy partition of $[a, b]$ fulfills the generalized Ruspini condition (7.2) with $r \geq 1$, i. e., for all $p_j \in P$, $\sum_{k=1}^{n} A_k(p_j) = r$, then the inversion formula (7.10) can be simplified to

$$\hat{f}(p_j) = \frac{1}{r} \sum_{k=1}^{n} F_k A_k(p_j)$$

or (in the case of Ruspini partition, i. e., $r = 1$) to

$$\hat{f}(p_j) = \sum_{k=1}^{n} F_k A_k(p_j) \, .$$

Analogous to Theorem 7.1, we can show that the inverse discrete F-transform $\hat{f}$ can approximate the original discrete function f on P with arbitrary precision [7.1]. Moreover, the properties (a)–(c) that are listed in Sect. 7.3.2 have valid discrete analogies.

An interesting comparison between the discrete F-transform and the least-square approximation was made in [7.20]. It was demonstrated that the discrete F-transform is invariant with respect to the interpolating and least-squares approximation of the set $\{(p_j, f(p_j)) \mid j = 1, \ldots, l\}$. This means that the best approximation of f on P in the form of $\sum_{i=1}^{n} \alpha_i A_i$, where $n \leq l$, has the same direct discrete F-transform as the original f.

7.5 *F*-Transforms of Functions of Two Variables

The direct and inverse F-transform of a function of two (and more) variables is a direct generalization of the case of one variable. We introduce it briefly and refer to [7.1] for more details.

Suppose that the universe is a rectangle $[a, b] \times [c, d] \subseteq \mathbb{R} \times \mathbb{R}$ and that $x_1 < \cdots < x_n$ are the fixed nodes of $[a, b]$ and $y_1 < \cdots < y_m$ are the fixed nodes of $[c, d]$ such that $x_1 = a$, $x_n = b$, $y_1 = c$, $x_m = d$ and $n, m \geq 2$. Let us formally extend the set of nodes by setting $x_0 = a$, $y_0 = c$, $x_{n+1} = b$, and $y_{m+1} = d$. Assume that $A_1, \ldots, A_n$ are the basic functions that form a generalized fuzzy partition of $[a, b]$ and $B_1, \ldots, B_m$ are basic functions that form a generalized fuzzy partition of $[c, d]$. Then, the rectangle $[a, b] \times [c, d]$ is partitioned into fuzzy sets $A_k \times B_l$ with the membership functions $(A_k \times B_l)(x, y) = A_k(x) B_l(y)$, $k = 1, \ldots, n$, $l = 1, \ldots, m$. Let $C([a, b] \times [c, d])$ be the set of continuous functions of two variables on the domain and $f \in C([a, b] \times [c, d])$.

Definition 7.8

Let $A_1, \ldots, A_n$ be the basic functions that form a generalized fuzzy partition of $[a, b]$ and $B_1, \ldots, B_m$ be the basic functions that form a generalized fuzzy partition of $[c, d]$. Let f be any function from $C([a, b] \times [c, d])$. We say that the $n \times m$-matrix of real numbers $\mathbf{F}[f] = (F_{kl})_{n \times m}$ is the (integral) F-transform of f with respect to $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ if for each $k = 1, \ldots, n$, $l = 1, \ldots, m$,

$$F_{kl} = \frac{\int_c^d \int_a^b f(x, y) A_k(x) B_l(y) \mathrm{d}x\mathrm{d}y}{\int_c^d \int_a^b A_k(x) B_l(y) \mathrm{d}x\mathrm{d}y} \, . \tag{7.11}$$

The components F_{kl} (7.11) have properties (adapted to the case of two variables) similar to those listed in Sect. 7.3.1. For example, the property (e) has the following form (we assume that $A_1, \ldots, A_n$ form an h_1-uniform Ruspini partition of $[a, b]$ and $B_1, \ldots, B_m$ form an h_2-uniform Ruspini partition of $[c, d]$)

$$\begin{aligned} &\int_c^d \int_a^b f(x, y) \mathrm{d}x\mathrm{d}y \\ &= \frac{h_1 h_2}{4} (F_{11} + F_{1m} + F_{n1} + F_{nm}) \\ &\quad + \frac{h_1 h_2}{2} \left(\sum_{k=2}^{n-1} F_{k1} + \sum_{k=2}^{n-1} F_{km} + \sum_{l=2}^{m-1} F_{1l} + \sum_{l=2}^{m-1} F_{nl} \right) \\ &\quad + h_1 h_2 \sum_{k=2}^{n-1} \sum_{l=2}^{m-1} F_{kl} \, . \end{aligned}$$

In the discrete case, when an original function f is known only at points $(p_i, q_j) \in [a, b] \times [c, d]$, where $i =$

$1,\dots,N$ and $j=1,\dots,M$, the (discrete) F-transform of f can be introduced in a manner analogous to the case of a function of one variable. This case is important for applications of the F-transform to image processing [7.8, 9, 18, 24–26].

Definition 7.9

Let a function f be given at nodes $(p_i,q_j)\in[a,b]\times[c,d]$, for which $i=1,\dots,N$ and $j=1,\dots,M$, and $A_1,\dots,A_n$ and $B_1,\dots,B_m$, where $n<N$ and $m<M$, be the basic functions that form generalized fuzzy partitions of $[a,b]$ and $[c,d]$, respectively. Suppose that sets P and Q of these nodes are sufficiently dense with respect to the chosen partitions. We say that the $n\times m$-matrix of real numbers $\mathbf{F}[f]=(F_{kl})_{nm}$ is the discrete F-transform of f with respect to $A_1,\dots,A_n$ and $B_1,\dots,B_m$ if

$$F_{kl}=\frac{\sum_{j=1}^{M}\sum_{i=1}^{N}f(p_i,q_j)A_k(p_i)B_l(q_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N}A_k(p_i)B_l(q_j)} \tag{7.12}$$

holds for all $k=1,\dots,n$, $l=1,\dots,m$.

The inverse F-transform of a function of two variables is a simple extension of (7.7). It will be given below for the continuous version of a function.

Definition 7.10

Let $A_1,\dots,A_n$ and $B_1,\dots,B_m$ be the basic functions that form generalized fuzzy partitions of $[a,b]$ and $[c,d]$, respectively. Let f be a function from $C([a,b]\times[c,d])$ and $\mathbf{F}[f]$ be the F-transform of f with respect to $A_1,\dots,A_n$ and $B_1,\dots,B_m$. Then, the function $\hat{f}:[a,b]\times[c,d]\to\mathbb{R}$ represented by

$$\hat{f}(x,y)=\frac{\sum_{k=1}^{n}\sum_{l=1}^{m}F_{kl}A_k(x)B_l(y)}{\sum_{k=1}^{n}\sum_{l=1}^{m}A_k(x)B_l(y)} \tag{7.13}$$

is called the *the inverse F-transform*.

Similar to the case of a function of one variable, we can prove that the inverse F-transform $\hat{f}$ can approximate the original continuous function f with arbitrary precision, and the (adapted) properties (a)–(c), which are listed in Sect. 7.3.2, are fulfilled.

7.6 F^1-Transform

In [7.16], a higher degree F-transform was introduced for the purpose for advanced applications in time series and image processing [7.26, 27]. In this section, we give a description of the F^1-transform, which has working applications, and refer to [7.16] for the F^m-transform for which $m>1$.

Throughout this section, we assume that $A_1,\dots,A_n$, $n>2$ is an h-uniform generalized fuzzy partition of $[a,b]$ such that there exists a generating function $A_0:[-1,1]\to[0,1]$ such that for all $k=1,\dots,n$, A_k is defined by (7.3) (the illustration is in Fig. 7.3).

Let k be a fixed integer from $\{1,\dots,n\}$, and let $L_2(A_k)$ be a normed space of square-integrable functions $f:[x_{k-1},x_{k+1}]\to\mathbb{R}$, where the norm $\|f\|_k$ is given by

$$\|f\|_k=\sqrt{\frac{\int_{x_{k-1}}^{x_{k+1}}f^2(x)A_k(x)\mathrm{d}x}{\int_{x_{k-1}}^{x_{k+1}}A_k(x)\mathrm{d}x}}\,.$$

By $L_2(A_1,\dots,A_n)$ we denote a set of functions $f:[a,b]\to\mathbb{R}$ such that for all $k=1,\dots,n$, $f|_{[x_{k-1},x_{k+1}]}\in L_2(A_k)$, where $f|_{[x_{k-1},x_{k+1}]}$ is the restriction of f on $[x_{k-1},x_{k+1}]$.

For any function f from $L_2(A_1,\dots,A_n)$ we define the F^1-*transform* of f with respect to $A_1,\dots,A_n$ as the vector of linear functions

$$\mathbf{F}^1[f]=(c_{1,0}+c_{1,1}(x-x_1),\dots,c_{n,0}+c_{n,1}(x-x_n))\,, \tag{7.14}$$

where for every $k=1,\dots,n$,

$$\begin{aligned}c_{k,0}&=\frac{\int_{x_{k-1}}^{x_{k+1}}f(x)A_k(x)\mathrm{d}x}{hs_0}\,,\\ c_{k,1}&=\frac{\int_{x_{k-1}}^{x_{k+1}}f(x)(x-x_k)A_k(x)\mathrm{d}x}{\int_{x_{k-1}}^{x_{k+1}}(x-x_k)^2A_k(x)\mathrm{d}x}\,,\end{aligned} \tag{7.15}$$

and

$$s_0=\int_{-1}^{1}A_0(x)\mathrm{d}x\,.$$

The kth component of the vector $\mathbf{F}^1[f]$ is denoted by $F_k^1[f]$.

The following is a list of properties of the F^1-transform of f with respect to a generalized fuzzy partition of $[a,b]$. They are particular cases of the properties of the F^m-transform proved in [7.16]:

(a) Let F_k and $c_{k,0}+c_{k,1}(x-x_k)$, for $k=1,\dots,n$, be respective kth components of $\mathbf{F}^1[f]$ and $\mathbf{F}[f]$. Then, $F_k=c_{k,0}$.

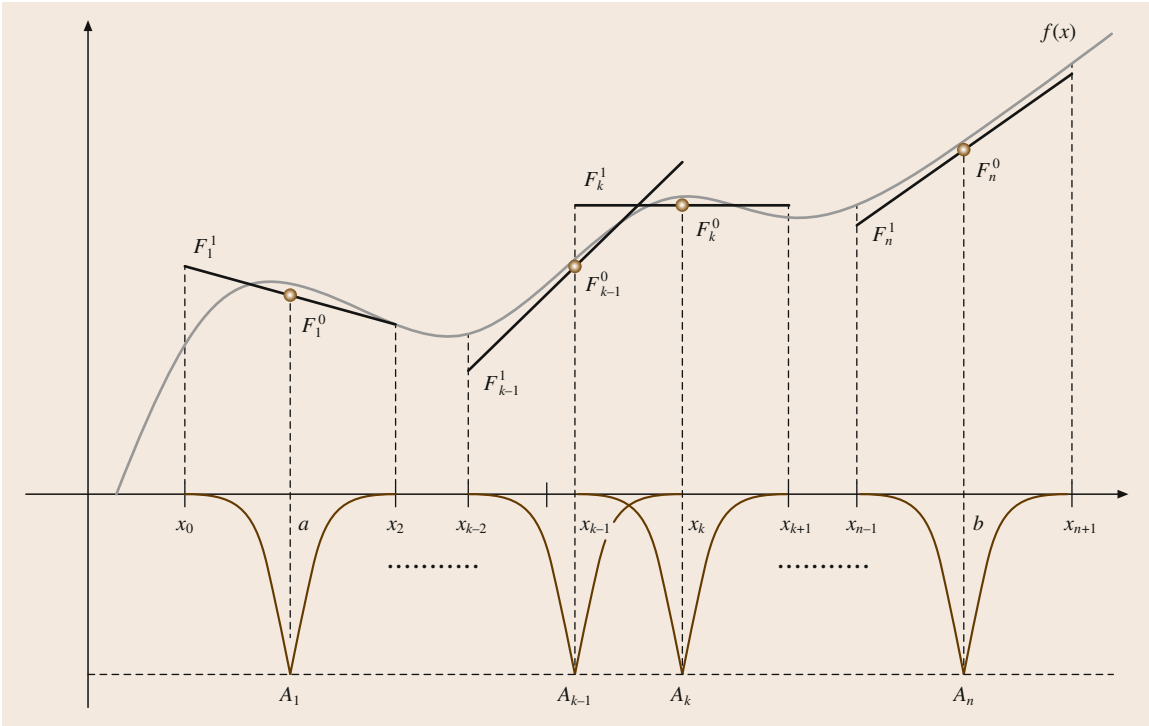

Fig. 7.6 Function f, its F^1-transform components $F_1^1, \ldots, F_k^1, \ldots, F_n^1$ (*linear segments*) and its F^0-transform components $F_1^0, \ldots, F_k^0, \ldots, F_n^0$ (*star nodes*) (after [7.16])

(b) If for all $x \in [a,b]$, $f(x) = d + cx$, then all the components of F^1-transform of $d + cx$ are equal to $(d + cx_k) + c(x - x_k)$, $k = 1, \ldots, n$.
(c) If $f = \alpha g + \beta h$, then $\mathbf{F}^1[f] = \alpha \mathbf{F}^1[g] + \beta \mathbf{F}^1[h]$.
(d) $c_{k,0} + c_{k,1}(x - x_k) = \min \|f(x) - (d + c(x - x_k)\|_k$, $k = 1, \ldots, n$, where min is considered over the set of functions of the form $(d + c(x - x_k))$.
(e) If f is four times continuously differentiable on $[a,b]$, then

$$c_{k,0} = f(x_k) + O(h^2)\,,$$
$$c_{k,1} = f'(x_k) + O(h)\,, \quad k = 1, \ldots, n\,.$$

In Fig. 7.6, we show a schematic representation of the F^1-transform components of a generic function f.

Finally, we give simplified expressions of F^1-transform components with respect to an h-uniform triangular fuzzy partition [7.16]

$$c_{k,0} = \frac{\int_{x_{k-1}}^{x_{k+1}} f(x) A_k(x) \mathrm{d}x}{h}\,, \tag{7.16}$$

$$c_{k,1} = \frac{12 \int_{x_{k-1}}^{x_{k+1}} f(x)(x - x_k) A_k(x) \mathrm{d}x}{h^3}\,, \tag{7.17}$$

where $k = 1, \ldots, n$.

7.7 Applications

In this section, we consider applications of the F-transform and F^1-transform to image processing.

7.7.1 Image Compression and Reconstruction

A method of lossy image compression and reconstruction using fuzzy relations was proposed in [7.19]. The dominant idea was a choice of suitable granulation (represented by a fuzzy relation) of an image domain. We will refer to this method as FEQ. F-transform image compression (FTR) is based on the same idea of granulation but connects it with fuzzy partitions [7.1, 9]. In the cited papers, two approaches were proposed: a uniform fuzzy partition of the entire domain [7.1] and a two-step partition [7.9] in which initially the entire do-

main is partitioned into blocks and second, each block is uniformly partitioned into fuzzy sets. Both approaches were compared with JPEG and other compression techniques (including FEQ) [7.9], and the conclusion was that the F-transform-based method is slightly worse than JPEG but better than FEQ. Two further improvements of the F-transform-based compression have been proposed in [7.18, 28], where an advantage over JPEG was achieved in many cases.

In this section, after reiterating the principles of image compression and reconstruction using the F-transform and its inverse, we explain how a proper choice of a fuzzy partition improves the quality of the reconstructed image. A detailed elaboration and comparison with other existing techniques is in [7.18, 28] and will be presented in subsequent papers.

Principles of Image Compression Using the F-Transform

Let a grayscale image of size $N \times M$ pixels be represented by a function of two variables $u : N \times M \to [0, 1]$. The value $u(i, j)$ represents the intensity range of each pixel in the gray scale. The problem of image compression is to reduce the image's size to save space or transmission time. A desirable size $n \times m$ (where $n < N$ and $m < M$) of a compressed image can be obtained from the *compression ratio*, $\rho = nm/(NM)$. If a compression method is lossy (JPEG, FEQ, and the F-transform, for example), then the respective reconstruction $\hat{u}$ to a full size image is compared with the original image using the two quality indices PSNR (peak signal-to-noise ratio) and RMSE (root-mean-square error), where

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

and

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} [u(i,j) - \hat{u}(i,j)]^2}{NM}} .$$

Simple F-Transform Compression

In [7.1], we proposed representing a compressed image by the $n \times m$ matrix $\mathbf{U}$ of F-transform components

$$\mathbf{U} = \begin{pmatrix} U_{11} & \dots & U_{1m} \\ \vdots & \vdots & \vdots \\ U_{n1} & \dots & u_{nm} \end{pmatrix} ,$$

computed over uniform fuzzy partitions (usually, triangular) $A_1, \dots, A_n$ and $B_1, \dots, B_m$ of the entire domains $[1, N]$ and $[1, M]$, respectively

$$U_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} u(i,j) A_k(i) B_l(j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k(i) B_l(j)} ,$$
$$k = 1, \dots, n ; \quad l = 1, \dots, m .$$

We proposed reconstructing $\mathbf{U}$ to a full-size image using the inverse F-transform of u such that

$$\hat{u}(i,j) = \sum_{k=1}^{n} \sum_{l=1}^{m} U_{kl} A_k(i) B_l(j) .$$

This method does not take advantage of any property of the original image and therefore, its quality is not very high. Let us illustrate it on the image *Cameraman* taken from the Corel Gallery. In Fig. 7.7, we show the original image and its reconstruction using the simple F-transform compression described above. The compression ratio is $\rho = 0.25$, and $\mathrm{PSNR} = 25.422$ (compare with $\mathrm{PSNR} = 38.8$ for JPEG with a similar compression ratio).

F-Transform Compression with Block Decomposition

This F-transform-based compression [7.9] was inspired by the JPEG method in which, at first, the entire domain was decomposed into blocks and then, each block was compressed according to a compression ratio. In [7.9], the same principle is used. In the first step, a decomposition into blocks of the same size is performed, where the size (chosen experimentally) is such that a certain quality of approximation by the inverse F-transform should be guaranteed (Theorem 7.1). Each block is then uniformly partitioned into cosine-shaped fuzzy sets and compressed by the simple F-transform method according to a compression ratio. In comparison with the simple F-transform compression, this method considers the peculiarities of the original images when making

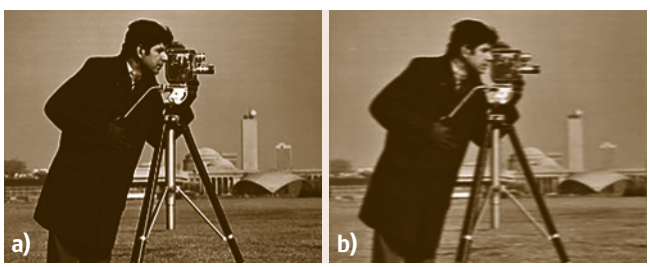

Fig. 7.7a,b Original image *Cameraman* **(a)** and its reconstruction after applying the simple F-transform compression **(b)** with $\mathrm{PSNR} = 25.422$

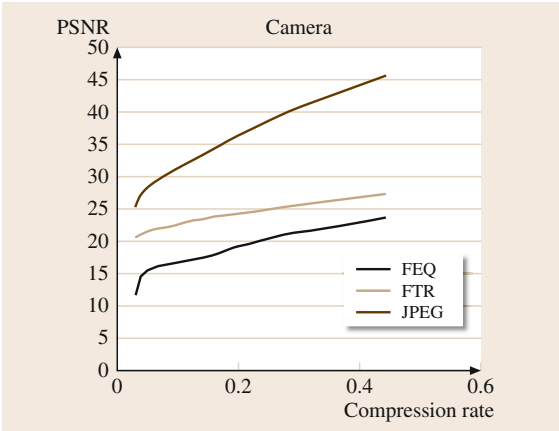

Fig. 7.8 The PSNR values of the image *Cameraman* compressed using three methods: FEQ, the F-transform with block decomposition, and JPEG (after [7.29])

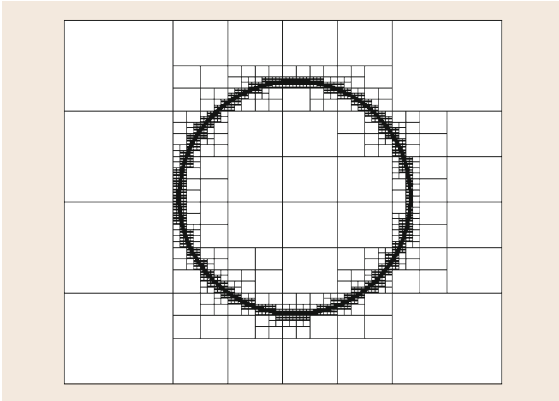

Fig. 7.9 The quad tree algorithm and the generalized fuzzy partition on its base

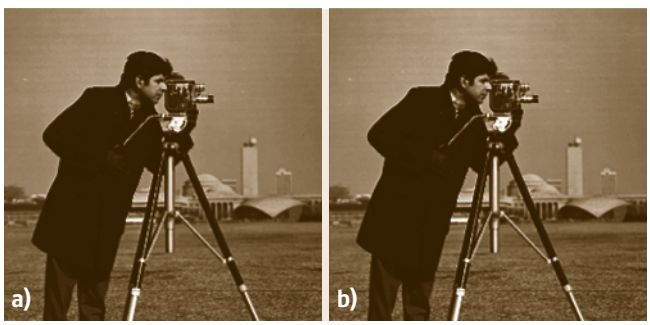

Fig. 7.10a,b Two reconstructions of the image *Cameraman* after applying the advanced F-transform compression (the ratio is 0.188) with the histogram restoring **(a)** and without it **(b)**. The PSNR values are 29 **(a)** and 30 **(b)**

the block decomposition. In Fig. 7.8, we show the PSNR quality measure of the image *Cameraman* compressed using three methods: FEQ, F-transform with block decomposition and JPEG. It is easily observed that the JPEG method is still better than the F-transform with block decomposition, whereas the latter is better than FEQ. However, for the particular image *Cameraman* and the compression ratio $\rho = 0.25$, the value of PSNR of the F-transform with block decomposition is similar to that of the simple F-transform compression: 25.0676 versus 25.422, respectively. This means that the uniform partition, even when applied to both steps independently, is not effective with respect to the quality estimated by PSNR. In the next subsection, we propose an F-transform compression method [7.18] that is almost nonlossy and is based on a nonuniform generalized partition adapted to each particular image.

Advanced Image Compression

If we analyze the properties of the F-transform (Sect. 7.3.1), then it is immediate from (a) that the more the function behaves like a constant, the better is the approximation quality of the inverse F-transform. Thus, the following recommendation regarding the choice of a proper generalized fuzzy partition can be made:

- A generalized fuzzy partition of the domain $[1, N] \times [1, M]$ into fuzzy sets $A_k \times B_l$, where $k = 1, \ldots, n$ and $l = 1, \ldots, m$, should guarantee that the difference between extremal values of the image over each $A_k \times B_l$ is not greater than $\varepsilon > 0$ or (if the preceding condition cannot be fulfilled) the area of $A_k \times B_l$ is not greater than $\delta > 0$.

There are several algorithms that can produce a generalized fuzzy partition with the mentioned property. In [7.18], we used the quad tree algorithm for this purpose; see the illustration in Fig. 7.9.

Let us add that the advanced image compression algorithm [7.18] uses the following two tricks to increase the quality of the reconstructed image:

- Preserve sharp edges
- Restore the histogram of the original image.

Figure 7.10 shows how the histogram restoration influences the quality of the reconstructed image. In Fig. 7.11, we see that the PSNR values of the advanced F-transform and the JPEG are almost equal.

7.7.2 Image Fusion

Image fusion aims to integrate complementary distorted multisensor, multitemporal, and/or multiview scenes into one new image that contains the *best* parts of each scene. Thus, the primary problem in image fusion is to find the least distorted scene for every pixel.

A local focus measure is traditionally used for the selection of an undistorted scene. The scene that maximizes the focus measure is selected. Usually, the focus measure is a measure of high-frequency occurrences in the image spectrum. This measure is used when a source of distortion is connected with blurring, which suppresses high frequencies in an image. In this case, it is desirable that a focus measure decreases with an increase in blurring.

There are various fusion methodologies that are currently in use. They can be classified according to the primary technique: aggregation operators [7.22], fuzzy methods [7.30], optimization methods (e.g., neural networks and genetic algorithms [7.29]), and multiscale decomposition methods based on various transforms (e.g., discrete wavelet transforms; [7.31]).

The F-transform approach to image fusion was proposed in [7.32, 33]. The primary idea is a combination of (at least) two fusion operators, both of which are based on the F-transform. The first fusion operator is applied to the F-transform components of scenes and is based on a robust partition of the scene domain. The second fusion operator is applied to the residuals of scenes with respect to inverse F-transforms with fused components and is based on a finer partition of the same domain. Although this approach is not explicitly based on focus measures, it uses the fusion operator, which is able to choose an undistorted scene among the available blurred scenes.

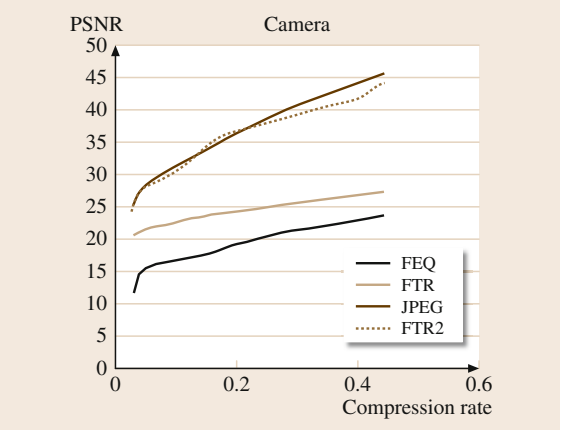

Fig. 7.11 The PSNR values of the image *Cameraman* compressed using four methods: FEQ, the F-transform with block decomposition, the advanced F-transform, and JPEG

Principles of Image Fusion Using the F-Transform

In this subsection, we present a short overview of the two methods of fusion that were proposed in [7.32, 33] and introduce a new method [7.34] that is a weighted combination of those two. We will demonstrate that the new method is computationally more effective than the first two.

The F-transform fusion is based on a certain decomposition of an image. We assume that the image u is a discrete real function $u = u(x, y)$ defined on the $N \times M$ array of pixels $P = \{(i, j) \mid i = 1, \ldots, N, j = 1, \ldots, M\}$ such that $u : P \to \mathbb{R}$. Moreover, let fuzzy sets $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$, where $2 \le n \le N, 2 \le m \le M$, establish uniform Ruspini partitions of $[1, N]$ and $[1, M]$, respectively. We begin with the following representation of u on P,

$$u(x, y) = u_{nm}(x, y) + e(x, y) \,, \tag{7.18}$$

$$e(x, y) = u(x, y) - u_{nm}(x, y) \,, \tag{7.19}$$

where u_{nm} is the inverse F-transform of u and e is the respective first difference. If we replace e in (7.18) by its inverse F-transform e_{NM} with respect to the finest partition of $[1, N] \times [1, M]$, the above representation can then be rewritten as follows,

$$u(x, y) = u_{nm}(x, y) + e_{NM}(x, y) \,. \tag{7.20}$$

We call (7.20) a *one-level decomposition* of u on P. If u is smooth, then the function e_{NM} is small (this claim follows from the property (e) in Sect. 7.3.1), and we can stop at this level. In the opposite case, we continue with the decomposition of the first difference e in (7.18). We decompose e into its inverse F-transform $e_{n'm'}$ (with respect to a finer fuzzy partition of $[1, N] \times [1, M]$ with $n' : n < n' \le N$ and $m' : m < m' \le M$ basic functions) and the second difference e'. Thus, we obtain the *second-level decomposition* of u on P

$$u(x, y) = u_{nm}(x, y) + e_{n'm'}(x, y) + e'(x, y) \,,$$
$$e'(x, y) = e(x, y) - e_{n'm'}(x, y) \,.$$

In the same manner, we can obtain a *higher level decomposition* of u on P

$$u(x,y) = u_{n_1 m_1}(x,y) + e^{(1)}_{n_2 m_2}(x,y) + \cdots + e^{(k-2)}_{n_{k-1} m_{k-1}}(x,y) + e^{(k-1)}(x,y)\,, \tag{7.21}$$

where

$$\begin{aligned} &0 < n_1 \le n_2 \le \cdots \le n_{k-1} \le N\,,\\ &0 < m_1 \le m_2 \le \cdots \le m_{k-1} \le M\,,\\ &e^{(1)}(x,y) = u(x,y) - u_{n_1 m_1}(x,y)\,,\\ &e^{(i)}(x,y) = e^{(i-1)}(x,y) - e^{(i-1)}_{n_i m_i}(x,y)\,,\\ &i = 2,\ldots,k-1\,. \end{aligned} \tag{7.22}$$

Three Algorithms for Image Fusion

In [7.33], we proposed two algorithms:

1. The *simple* F-transform-based fusion algorithm (SA) and
2. The *complete* F-transform-based fusion algorithm (CA).

The principal role in the fusion algorithms CA and SA is played by the *fusion operator* $\kappa : \mathbb{R}^K \to \mathbb{R}$, which is defined as follows:

$$\kappa(x_1,\ldots,x_K) = x_p\,, \quad \text{if } |x_p| = \max(|x_1|,\ldots,|x_K|)\,. \tag{7.23}$$

The Simple F-Transform-Based Fusion Algorithm

In this subsection, we present a *block* description of the SA without technical details, which can be found in [7.33]. We assume that $K \ge 2$ input (channel) images $c_1,\ldots,c_K$ with various types of degradation are given. Our aim is to recognize undistorted parts in the given images and to fuse them into one image. The algorithm is based on the decompositions given in (7.20), which are applied to each channel image:

1. Choose values n and m such that $2 \le n \le N, 2 \le m \le M$ and create a fuzzy partition of $[1,N]\times[1,M]$ by fuzzy sets $A_k \times B_l$, where $k = 1,\ldots,n$ and $l = 1,\ldots,m$.
2. Decompose the input images $c_1,\ldots,c_K$ into inverse F-transforms and error functions according to the one-level decomposition (7.20).
3. Apply the fusion operator (7.23) to the respective F-transform components of $c_1,\ldots,c_K$, and obtain the fused F-transform components of a new image.
4. Apply the fusion operator to the respective F-transform components of the error functions e_i, $i = 1,\ldots,K$, and obtain the fused F-transform components of a new error function.
5. Reconstruct the fused image from the inverse F-transforms with the fused components of the new image and the fused components of the new error function.

The SA-based fusion is very efficient if we can guess values n and m that characterize a proper fuzzy partition. Usually, this is performed manually according to the user's skills. The dependence on fuzzy partition parameters can be considered as a primary shortcoming of this otherwise effective algorithm. Two recommendations follow from our experience:

- For complex images (with many small details), higher values of n and m yield better results.
- If a triangular shape for a basic function is chosen, than the generic choice of n and m is such that the corresponding values of n_p and m_p are equal to 3 (recall that n_p is the number of points that are covered by every full basic function A_k).

The Complete F-Transform-Based Fusion Algorithm

The CA-based fusion does not depend on the choice of only one fuzzy partition (as in the case of the SA) because it runs through a sequence (7.22) of increasing values of n and m. The algorithm is based on the decomposition presented in (7.21), which is applied to each channel image. The description of the CA is similar to that of the SA except for step 4, which is repeated in a cycle. Therefore, the quality of fusion is high, but the implementation of the CA is rather slow and memory consuming, especially for large images. For an illustration, Fig. 7.12, Tables 7.1 and 7.2.

Table 7.1 Basic characteristics of the three algorithms applied to the image *Balls*

Image	Resolution	Time (s)			Memory (MB)		
		CA	SA	ESA	CA	SA	ESA
Balls	1600×1200	340	1.2	36	270	58	152

Table 7.2 MSE (mean-square error) and PSNR characteristics of the three fusion methods applied to the image *Balls*

Image set	MSE			PSNR		
	CA	SA	ESA	CA	SA	ESA
Balls	1.28	6.03	0.86	48.91	43.81	52.57

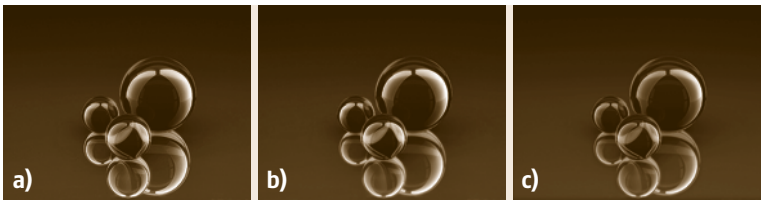

Fig. 7.12a–c The SA **(a)**, CA **(b)** and ESA **(c)** fusions of the image *Balls*. The ESA fusion has the best quality (Table 7.2)

Enhanced Simple Fusion Algorithm

In [7.34], we proposed an algorithm that is as fast as the SA and as efficient as the CA. We aimed at achieving the following goals:

- Avoid running through a long sequence of possible partitions (as in the case of CA).
- Automatically adjust the parameters of the fusion algorithm according to the level of blurring and the location of blurred areas in input images.

The algorithm adds another run of the F-transform over the first difference (7.18). The explanation is as follows: the first run of the F-transform is aimed at edge detection in each input image, whereas the second run propagates only sharp edges (and their local areas) to the fused image. We refer to this algorithm as to *enhanced simple algorithm* (ESA) and give its informal description:

for all input (channel) images **do**
  Compute the *inverse F-transform*
  Compute the *first absolute difference* between the original image and the inverse F-transform of it
  Compute the *second absolute difference* between the first one and its inverse F-transform and set them as the pixel weights
end for
for all pixels in an image **do**
  Compute the value of *sow* – the sum of the weights over all input images
  for all input images **do**
    Compute the value of *wr* – the ratio between the weight of a current pixel and *sow*
  end for
  Compute the fused value of a pixel in the resulting image as a weighted (by *wr*) sum of input image values
end for

The primary advantages of the ESA are:

- *Time* – the execution time is smaller than for the CA (Table 7.1).
- *Quality* – the quality of the ESA fusion is better than that of the SA and for particular cases (Table 7.2), it is better than that of the CA.

Because of space limitations, we present only one illustration of the F-transform fusion performed using the three algorithms, SA, CA, and ESA. We chose the image *Balls* with geometric figures to demonstrate that our fusion methods are able to reconstruct edges. In Fig. 7.13, two (channel) inputs of the image *Balls* are given, and in Fig. 7.12, three fusions of the same image are demonstrated.

In Table 7.1, we demonstrate that the complexity (measured by the execution time or by the memory used) of the ESA is greater than the complexity of the SA and less than the complexity of the CA.

In Table 7.2, we demonstrate that for the particular image *Balls*, the quality of fusion (measured by the values of MSE and PSNR) of the ESA result is better (the MSE value is smaller) than the quality of the SA result and even than the quality of the CA result.

7.7.3 F^1-Transform Edge Detector

Edge detection is inevitable in image processing. In particular, it is a first step in feature extraction and image segmentation. We focused on the Canny edge detector [7.35], which is widely used in computer vision. It was developed to ensure three basic criteria: good detection, good localization, and minimal response. In

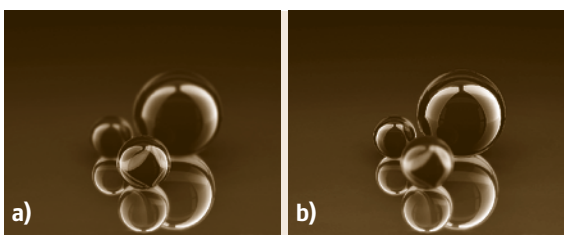

Fig. 7.13a,b Two inputs for the image *Balls*. The *central ball* is blurred in **(a)**, and conversely, it is the only sharp ball in **(b)**

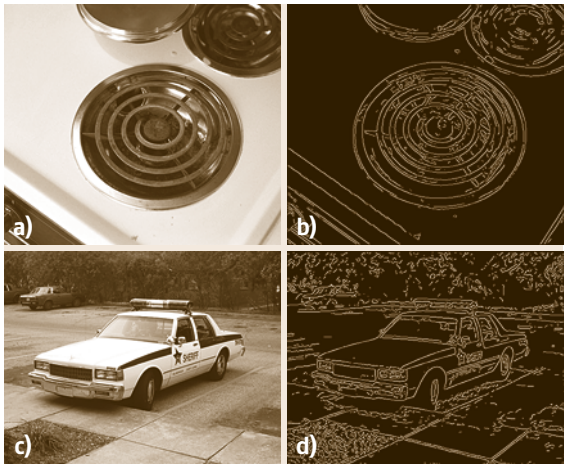

Fig. 7.14a–d Original images (a,c) and their F^1-transform edges (b,d)

these aspects, the Canny detector can be considered an *optimal* edge detector. In [7.26], we proposed using the F^1-transform with the purpose of simplifying the first two steps of the Canny algorithm. Below, we provide the details of our proposal.

The Canny algorithm is a multistep procedure for detecting edges as the local maxima of the gradient magnitude. The first step, performed using a Gaussian filter, is image smoothing and filtering noise. The second step is computation of a gradient of the image function to find the local maxima of the gradient magnitude and the gradient's direction at each point. This step is performed using a convolution of the original image with directional masks (edge detection operators, such as those of *Roberts, Prewitt*, and *Sobel*, are some examples of these filters). The next step is called nonmaximum suppression [7.36], and it selects those points whose gradient magnitudes are maximal in the corresponding gradient direction. The final step is tracing edges and hysteresis thresholding, which leads to preserving the continuity of edges.

In our experiment, we removed the first two steps in the Canny algorithm and replaced them by computation of approximate gradient values using the F^1-transform. The reason is that the F^1-transform (similar to the ordinary F-transform) filters out noise when computing approximate values of the first partial derivatives given by (7.15). We assume that the image is represented by a discrete function $u : P \to \mathbb{R}$ of two variables, where $P = \{(i,j) \mid i = 1,\dots,N, j = 1,\dots,M\}$ is an $N \times M$ array of pixels, and the fuzzy sets $A_1,\dots,A_n$ and $B_1,\dots,B_m$ establish a uniform triangular fuzzy partition of $[1,N]$ and $[1,M]$, respectively.

Let $x_1,\dots,x_n \in [1,N]$ and $y_1,\dots,y_m \in [1,M]$ be the h_x and h_y-equidistant nodes of $[1,N]$ and $[1,M]$, respectively.

According to property (e) in Sect. 7.6, the coefficients $c_{k,1}$ of the linear polynomials of the F^1-transform components are approximate values of the first partial derivatives of the image function at nodes (x_k, y_l) (for simplicity, we assume $k = 2,\dots,n-1$ and $l = 2,\dots,m$), where by (7.17) and (7.5) the following hold,

$$c_{k,1}(y_l) = \frac{12}{h_x^3 h_y} \sum_{i=1}^{N} \sum_{j=1}^{M} u(i,j)(i - x_k) A_k(i) B_l(j) \,,$$

$$c_{l,1}(x_k) = \frac{12}{h_x h_y^3} \sum_{i=1}^{N} \sum_{j=1}^{M} u(i,j)(j - y_l) A_k(i) B_l(j) \,.$$

Then, we can write approximations of the first partial derivatives as the respective inverse F-transforms

$$G_x(i,j) \approx \sum_{k=1}^{n} \sum_{l=1}^{m} c_{k,1}(y_l) A_k(i) B_l(j) \,,$$

and

$$G_y(i,j) \approx \sum_{k=1}^{n} \sum_{l=1}^{m} c_{l,1}(x_k) A_k(i) B_l(j) \,.$$

All other steps of the Canny algorithm – namely, finding the local maxima of the gradient magnitude and its direction, nonmaximum suppression, tracing edges through the image and hysteresis thresholding – are the same as in the original procedure.

In the two examples in Fig. 7.14, we demonstrate the results of the F^1-transform edge detector on images chosen from the dataset available at ftp://figment.csee.usf.edu/pub/ROC/edge_comparison_dataset.tar.gz.

We observe that many thin edges/lines are detected as well as their connectedness and smoothness. Moreover, the following properties are retained:

- Smoothness of circular lines
- Concentricness circles
- Smoothness of sharp connections.

7.8 Conclusions

In this chapter, the theory of the F-transform has been discussed from the perspective of the latest developments and applications. The importance of a proper choice of fuzzy partition has been stressed. Various fuzzy partitions have been considered, including the most general partition (currently known). The definition of the F-transform has been adapted to the generalized fuzzy partition, and the main properties of the F-transform have been re-established. The applications to image processing, namely image compression, fusion and edge detection, have been discussed with sufficient technical details.

References

7.1 I. Perfilieva: Fuzzy transforms: Theory and applications, Fuzzy Sets Syst. **157**, 993–1023 (2006)

7.2 L.A. Zadeh: Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Syst. Man Cybern. **SMC-3**, 28–44 (1973)

7.3 L.A. Zadeh: The concept of a linguistic variable and its application to approximate reasoning, Part I, Inf. Sci. **8**, 199–257 (1975)

7.4 L.A. Zadeh: The concept of a linguistic variable and its application to approximate reasoning, Part II, Inf. Sci. **8**, 301–357 (1975)

7.5 L.A. Zadeh: The concept of a linguistic variable and its application to approximate reasoning, Part III, Inf. Sci. **9**, 43–80 (1975)

7.6 T. Takagi, M. Sugeno: Fuzzy identification of systems and its application to modeling and control, IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)

7.7 I. Perfilieva, H. De Meyer, B. De Baets, D. Pisková: Cauchy problem with fuzzy initial condition and its approximate solution with the help of fuzzy transform, Proc. WCCI 2008, IEEE Int. Conf. Fuzzy Syst., Hong Kong (2008) pp. 2285–2290

7.8 I. Perfilieva: Fuzzy transforms and their applications to image compression, Lect. Notes Artif. Intell. **3849**, 19–31 (2006)

7.9 F. Di Martino, V. Loia, I. Perfilieva, S. Sessa: An image coding/decoding method based on direct and inverse fuzzy transforms, Int. J. Approx. Reason. **48**, 110–131 (2008)

7.10 V. Novák, M. Štěpnička, A. Dvořák, I. Perfilieva, V. Pavliska, L. Vavřičková: Analysis of seasonal time series using fuzzy approach, Int. J. Gen. Syst. **39**, 305–328 (2010)

7.11 I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, M. Štěpnička: Analysis and prediction of time series using fuzzy transform, Proc. WCCI 2008, IEEE Int. Conf. Neural Netw., Hong Kong (2008) pp. 3875–3879

7.12 F. Di Martino, V. Loia, S. Sessa: Fuzzy transforms method in prediction data analysis, Fuzzy Sets Syst. **180**, 146–163 (2011)

7.13 M. Štěpnička, A. Dvořák, V. Pavliska, L. Vavřičková: A linguistic approach to time series modeling with the help of F-transform, Fuzzy Sets Syst. **180**, 164–184 (2011)

7.14 L. Troiano, P. Kriplani: Supporting trading strategies by inverse fuzzy transform, Fuzzy Sets Syst. **180**, 121–145 (2011)

7.15 L. Stefanini: F-transform with parametric generalized fuzzy partitions, Fuzzy Sets Syst. **180**, 98–120 (2011)

7.16 I. Perfilieva, M. Daňková, B. Bede: Towards F-transform of a higher degree, Fuzzy Sets Syst. **180**, 3–19 (2011)

7.17 M. Holčapek, T. Tichý: A smoothing filter based on fuzzy transform, Fuzzy Sets Syst. **180**, 69–97 (2011)

7.18 P. Hurtik, I. Perfilieva: Image compression methodology based on fuzzy transform, Int. Jt. Conf. CISIS'12-ICEUTE'12-SOCO'12 Special Sessions, Adv. Intell. Soft Comput., Vol. 189 (Springer, Berlin, Heidelberg 2013) pp. 525–532

7.19 K. Hirota, W. Pedrycz: Fuzzy relational compression, IEEE Trans. Syst. Man Cybern. **29**, 407–415 (1999)

7.20 G. Patanè: Fuzzy transform and least-squares approximation: Analogies, differences, and generalizations, Fuzzy Sets Syst. **180**, 41–54 (2011)

7.21 I. Perfilieva, V. Novák, A. Dvořák: Fuzzy transform in the analysis of data, Int. J. Approx. Reason. **48**, 36–46 (2008)

7.22 R.S. Blum: Robust image fusion using a statistical signal processing approach, Inf. Fusion **6**, 119–128 (2005)

7.23 I. Perfilieva, R. Valášek: Fuzzy transforms in removing noise. In: *Computational Intelligence, Theory and Applications*, ed. by B. Reusch (Springer, Heidelberg 2005) pp. 225–234

7.24 F. Di Martino, V. Loia, S. Sessa: A segmentation method for images compressed by fuzzy transforms, Fuzzy Sets Syst. **161**, 56–74 (2010)

7.25 I. Perfilieva, M. Daňková: Image fusion on the basis of fuzzy transforms, Proc. 8th Int. FLINS Conf., Madrid (2008) pp. 471–476

7.26 I. Perfilieva, P. Hodáková, P. Hurtik: F^1-transform edge detector inspired by Canny's algorithm. In: *Advances on Computational Intelligence (IPMU2012)*, ed. by S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, R.R. Yager (Catania, Italy 2012) pp. 230–239

7.27 V. Novák, I. Perfilieva, V. Pavliska: The use of higher-order F-transform in time series analysis, World Congr. IFSA 2011 AFSS 2011, Surabaya (2011) pp. 2211–2216
7.28 I. Perfilieva, B. De Baets: Fuzzy transform of monotonous functions, Inf. Sci. **180**, 3304–3315 (2010)
7.29 A. Mumtaz, A. Masjid: Genetic algorithms and its applications to image fusion, IEEE Int. Conf. Emerg. Technol., Rawalpindi (2008) pp. 6–10
7.30 R. Ranjan, H. Singh, T. Meitzler, G.R. Gerhart: Iterative image fusion technique using fuzzy and neuro fuzzy logic and applications, Proc. IEEE Fuzzy Inf. Process. Soc., Detroit (2005) pp. 706–710
7.31 G. Piella: A general framework for multiresolution image fusion: From pixels to regions, Inf. Fusion **4**, 259–280 (2003)
7.32 I. Perfilieva, M. Daňková, H.P.M. Vajgl: The use of f-transform for image fusion algorithms, Proc. Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR2010), Cergy Pontoise (2010) pp. 472–477
7.33 I. Perfilieva, M. Daňková, P. Hodáková, M. Vajgl: F-transform based image fusion. In: *Image Fusion*, ed. by O. Ukimura (InTech, Rijeka 2011), pp. 3–22, available online from http://www.intechopen.com/books/image-fusion/f-transform-based-image-fusion
7.34 M. Vajgl, I. Perfilieva, P. Hodáková: Advanced F-transform-based image fusion, Adv. Fuzzy Syst. **2012**, 125086 (2012)
7.35 J. Canny: A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**(6), 679–698 (1986)
7.36 A. Rosenfeld, M. Thurston: Edge and curve detection for visual scene analysis, IEEE Trans. Comput. **C-20**(5), 562–569 (1971)