

Research Article

Generating Suitable Basic Functions Used in Image Reconstruction by F-Transform

Pavel Vlašánek

Department of Informatics and Computers, University of Ostrava, 30. Dubna 22, 701 03 Ostrava, Czech Republic

Correspondence should be addressed to Pavel Vlašánek; pavel.vlasanek@osu.cz

Received 5 July 2013; Accepted 12 July 2013

Academic Editor: Salvatore Sessa

Copyright © 2013 Pavel Vlašánek. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image reconstruction technique based on F-transform uses clearly defined basic functions. These functions have strong impact on the quality of reconstruction. We can use some predefined shape and radius, but also we can create a new one from the scratch. The aim of this paper is to analyze the creating process and based on that find best basic function for input set of damaged testing images.

1. Introduction

Image reconstruction aims at recovering damaged parts. In real situation this damage can be caused by various sources on various surfaces. Input information has to be digitalized as a first step and after that divided on damaged and nondamaged parts. Typical situation lays on photography. We can distinguish among many types of damage like stains, scratches, text, or noise. We can say that every kind of damage covers typical way of damaging process. Unwanted time or date stamp is covered by *text*. Scratches, cracks, and folds are covered by *inpaint*. These are shown in Figure 1 which includes also *noise*.

Target of reconstruction is removing damaged parts from input image and replacing them by the parts with recomputed values. These values are computed from the neighborhood ones. The technique mentioned in this paper is based on F-transform which brings particular way of valuation neighborhood pixels and their usage in the computation [1, 2]. Other methods [3, 4] differ by the choice at the technique.

In this paper, we will focus on the valuation part of the computation. Quality of reconstruction will be measured by RMSE value (RMSE stands for the root mean square error). In Figure 2 you can see damaged input images and reconstructed ones with usage of the ideal basic function described later. We will analyse influence of changing parameter on target quality.

2. F-Transform

In image reconstruction, a discrete version of the F-transform is used. Details can be seen in [5, 6]. In this section two-dimensional (2D) variant and also conditions for proper functioning will be briefly introduced.

2.1. Fuzzy Partition with Ruspini Condition. Let $x_1 < \dots < x_n$ 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, \dots, A_n , identified with their membership functions defined on $[a, b]$, establish a fuzzy partition with Ruspini condition of $[a, b]$ if they fulfill the following conditions for $k = 1, \dots, n$:

- (1) $A_k: [a, b] \rightarrow [0, 1]$, $A_k(x_k) = 1$;
- (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 $k = 2, \dots, n$, increases on $[x_{k-1}, x_k]$, and $A_k(x)$, for $k = 1, \dots, n - 1$, strictly decreases on $[x_k, x_{k+1}]$;
- (5) for all $x \in [a, b]$,

$$\sum_{k=1}^n A_k(x) = 1. \quad (1)$$

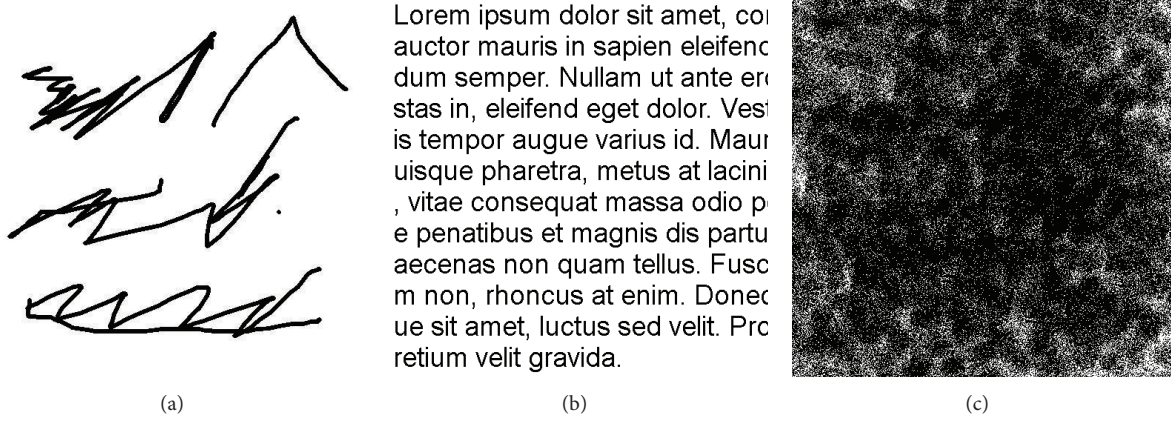


FIGURE 1: (a) Inpaint damage; (b) text damage; (c) noise damage.

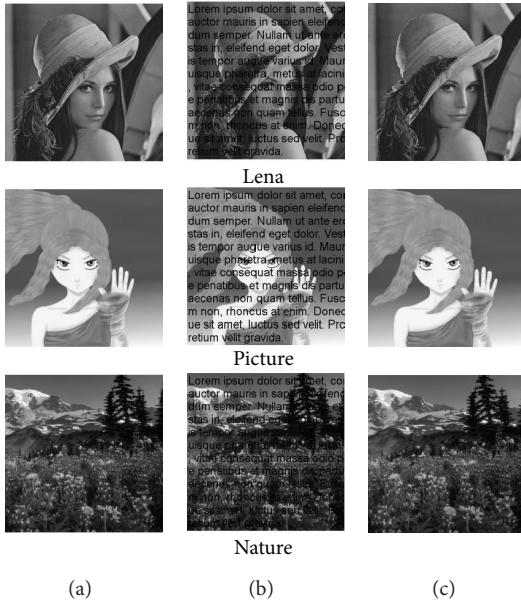


FIGURE 2: (a) Original image; (b) damaged image; (c) reconstructed image.

The condition (1) is known as the *Ruspini condition*. The membership functions A_1, \dots, A_n are called *basic functions*. A point $x \in [a, b]$ is *covered* by basic function A_k if $A_k(x) > 0$.

2.2. Shape of the Basic Function. 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 Figure 3, two such partitions with triangular and cosine basic functions are shown. The formulas given below represent generic fuzzy partitions with the Ruspini condition and triangular functions:

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

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

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

(2)

where $k = 2, \dots, n-1$ 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, \dots, x_n , where $n \geq 3$, are *h-equidistant*; that is, $x_k = a + h(k-1)$, for $k = 1, \dots, n$, where $h = (b-a)/(n-1)$, and two additional properties are met:

$$(6) A_k(x_k - x) = A_k(x_k + x), \text{ for all } x \in [0, h], k = 2, \dots, n-1;$$

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

An *h-uniform* fuzzy partition of $[a, b]$ can be determined by the so called *generating function* $A_0: [-1, 1] \rightarrow [0, 1]$, which is assumed to be *even* (The function $A_0: [-1, 1] \rightarrow \mathbb{R}$ is even if for all $x \in [0, 1]$, $A_0(-x) = A_0(x)$), continuous, have a bell shape, and fulfill $A_0(0) = 1$. Basic functions A_k of an *h-uniform* fuzzy partition with generating function A_0 are shifted copies of A_0 in the sense that

$$A_1(x) = \begin{cases} A_0\left(\frac{x - x_1}{h}\right), & x \in [x_1, x_2], \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

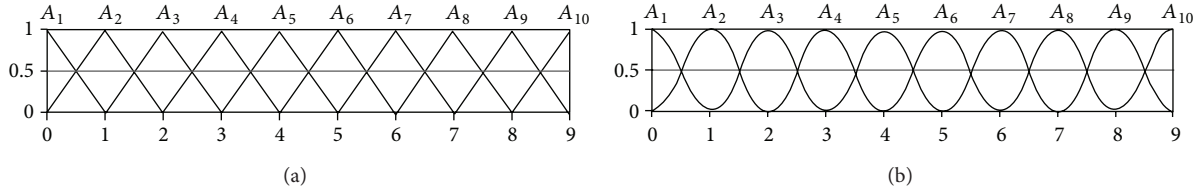


FIGURE 3: (a) Ruspini partitions with triangular basic function; (b) Ruspini partitions with cosine basic function.

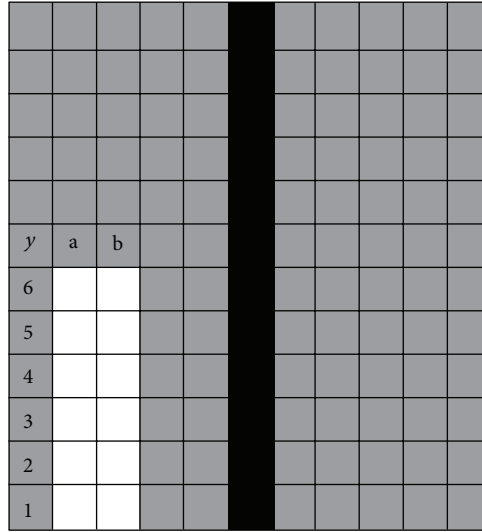


FIGURE 4: Template for basic function definition.

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

$$A_k(x) = \begin{cases} A_0\left(\frac{x - x_k}{h}\right), & x \in [x_{k-1}, x_{k+1}], \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

$$A_n(x) = \begin{cases} A_0\left(\frac{x - x_n}{h}\right), & x \in [x_n - 1, x_n], \\ 0, & \text{otherwise.} \end{cases}$$

As an example, we notice that the function $A_0(x) = 1 - |x|$ is a generating function for any h -uniform triangular partition. In the sequel, we will be using h -uniform fuzzy partitions only and refer to h as to a *radius* of partition [7].

2.3. Discrete 2D F-Transform. We say that the $n \times m$ -matrix of real numbers $[U_{kl}]$ is called *the (discrete) F-transform* of u with respect to $\{A_1, \dots, A_n\}$ and $\{B_1, \dots, B_m\}$ if for all $k = 1, \dots, n, l = 1, \dots, m$,

$$U_{kl} = \frac{\sum_{j=1}^M \sum_{i=1}^N u(p_i, q_j) A_k(p_i) B_l(q_j)}{\sum_{j=1}^M \sum_{i=1}^N A_k(p_i) B_l(q_j)}. \quad (5)$$

The elements U_{kl} are called components of the F-transform.

The *inverse F-transform* is defined as follows:

$$\hat{u}(i, j) = \sum_{k=1}^n \sum_{l=1}^m U_{kl} A_k(i) B_l(j). \quad (6)$$

We use discrete basic functions A_k and B_l . Our results show that sufficient image reconstruction is enough to use maximal radius 4. Discrete basic function with respect to Ruspini condition is building on top of template in Figure 4.

You can see white squares in Figure 4. These squares can be marked as parts of the basic function. One marked per column. After that is automatically computed mirrored part of the basic function as can be seen in Figure 5. We choose $y = 4$ for the column a . Value in column 10 is the same, and because of Ruspini condition, values in columns 5 and 7 are easily computable as

$$f(y_5) = 1 - f(y_a),$$

$$f(y_7) = 1 - f(y_a) = f(y_5). \quad (7)$$

Because of using Ruspini condition, we can use copies of the basic function for covering the whole range. Height of the template determines values $f(y)$ on the y axis. For better usability we choose value 0 for first row, 1 for last one, and twice 0.5 on the middle. Two basic functions with marked values can be seen in Figure 6.

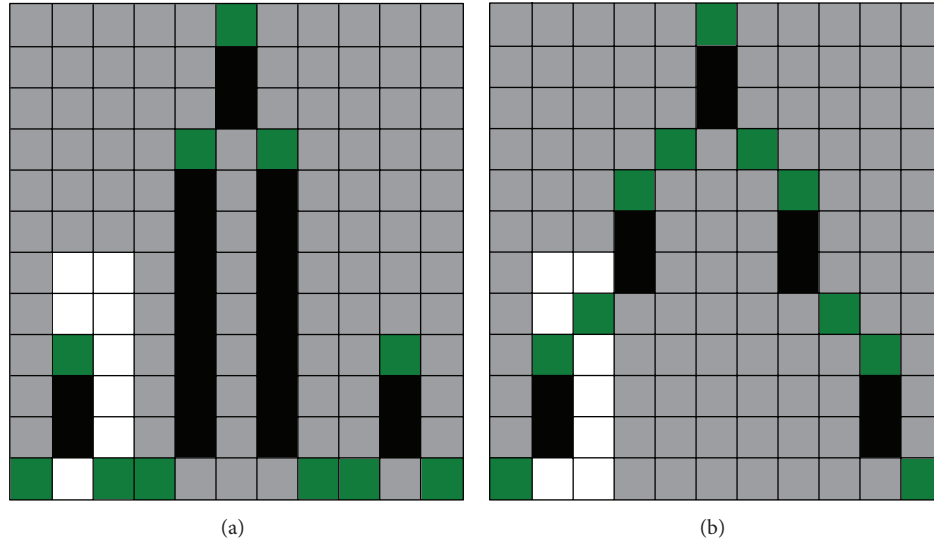


FIGURE 5: (a) Result after selection of fourth pixel in a column; (b) result after selection of fifth pixel in b column.

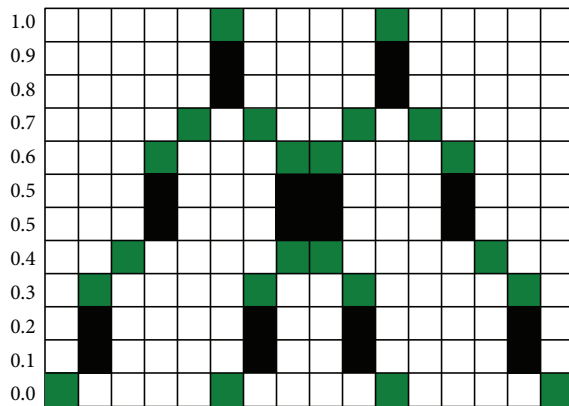


FIGURE 6: Template for basic function definition.

The overlapping part of basic functions have functional values equal 1 in each column. For better visualization, black vertical lines are also plotted.

3. Influence of Basic Functions

Shapes of the basic functions can be *nondecreasing*, *oscillating*, or *nonincreasing*. Differences and influences on computation are described in [8]. We choose basic functions with radius 2 in the first step of reconstruction and radius 4 in the second step because of fully sufficient usage on the input set of testing images. In this paper, we focus on building shape of basic function step by step based on results provided by RMSE. We identified two ways of this process which will be demonstrated on template in Figure 4 where $\max = 6$.

Column by Column. Find the best value for current column and continue with next column:

- (1) choose radius $r = 2$,

- (2) choose active column as $c = "a,"$
- (3) choose current row as $y = 1$,
- (4) based on those values create a basic function by mirroring and with respect to Ruspini condition formula (7),
- (5) use the basic function for image reconstruction,
- (6) compare reconstructed image with original undamaged one by RMSE,
- (7) if $y = \max$, then $r = 4$, $c = "b"$ and continue by step 3,
- (8) change current row as $y = y + 1$,
- (9) continue by step 4.

Radius by Radius. Find the best value for current radius and continue with next radius:

- (1) choose current radius as $r = 2$,
- (2) choose current column as $c = "a,"$
- (3) choose current row as $y = 1$,
- (4) based on those values create a basic function,
- (5) use the basic function for image reconstruction,
- (6) compare reconstructed image with the original undamaged by RMSE,
- (7) if $y = \max$ and $r = 4$, then $c = "b"$ and continue by step 3,
- (8) if $y = \max$, then $r = 4$ and continue by step 3,
- (9) choose current row as $y = y + 1$,
- (10) continue by step 4.

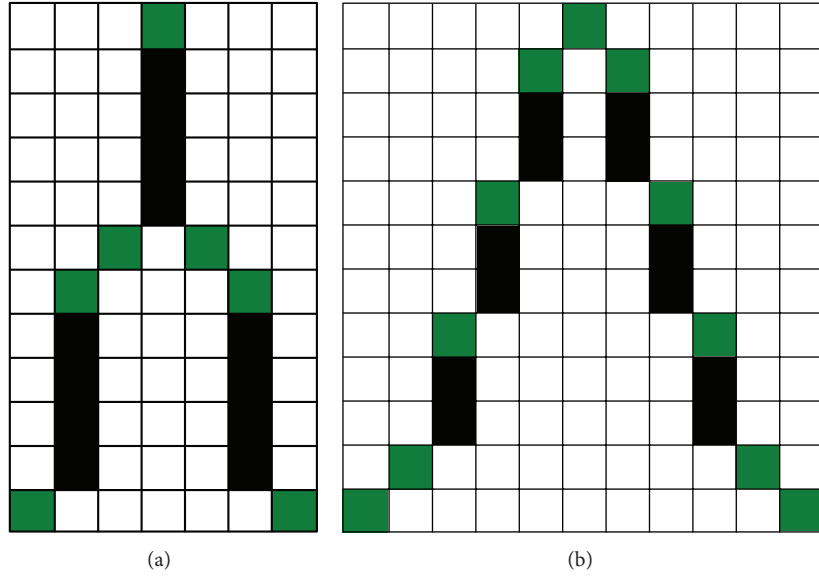


FIGURE 7: (a) Basic function for radius 2; (b) basic function for radius 4.

TABLE 1: RMSE values of the *column by column* basic function creation.

y	a	b	a	b	a	b
	Lena		Picture		Nature	
1	62.99	5.16	46.72	4.01	71.25	7.55
2	49.99	5.04	36.63	3.89	56.39	7.39
3	49.98	4.97	36.62	3.82	56.37	7.30
4	49.97	4.93	36.62	3.79	56.36	7.27
5	49.97	4.92	36.61	3.80	56.35	7.27
6	49.96	4.95	36.61	3.84	56.35	7.32

First way *column by column* is computed in Table 1. You can see that there is for column a best RMSE value for $y = 6$. It means that we choose sixth row for column a . Next column b is the best RMSE value in row $y = 5$ for the *Lena* and *Nature* images and $y = 4$ for the *Picture* image. We choose basic function with values $y = 6$ for column a and $y = 5$ for column b . This function is in Figure 8.

Second way of basic function creation *radius by radius* is computed in Table 2. From the table comes that the best value is $y = 2$ for a and $y = 5$ for column b . This basic function is shown in Figure 7.

4. Conclusion

As a result we can say that step by step process converges to oscillating or linear shape. *Column by column* way is the best solution oscillating basic function. Value $y = 6$ for column a and $y = 5$ for column b provides RMSE values 4.92 for *Lena*, 3.80 for *Picture*, and 7.27 for *Nature*. Because of very close results for *Picture* image between $b = 4$ and $b = 5$, we choose

TABLE 2: RMSE values of the *radius by radius* basic function creation.

a/b	1	2	3	4	5	6
Lena						
1	6.12	5.71	5.58	5.51	5.50	5.52
2	5.16	4.97	4.85	4.79	4.78	4.81
3	5.12	4.96	4.86	4.81	4.80	4.83
4	5.11	4.97	4.88	4.83	4.83	4.86
5	5.13	5.00	4.92	4.87	4.87	4.90
6	5.16	5.04	5.97	4.93	4.92	4.95
Picture						
1	4.93	4.32	4.21	4.16	4.15	4.19
2	3.93	3.76	3.65	3.61	3.61	3.65
3	3.91	3.76	3.67	3.63	3.64	3.68
4	3.92	3.79	3.71	3.67	3.68	3.72
5	3.96	3.83	3.76	3.73	3.73	3.77
6	4.01	3.89	3.82	3.80	3.80	3.84
Nature						
1	8.50	7.68	7.45	7.33	7.30	7.33
2	7.63	7.33	7.16	7.07	7.06	7.10
3	7.53	7.30	7.15	7.09	7.09	7.13
4	7.50	7.30	7.18	7.13	7.14	7.19
5	7.51	7.34	7.23	7.19	7.20	7.25
6	7.55	7.39	7.30	7.27	7.27	7.32

value $b = 5$ for all of them. *Radius by radius* is the best linear function with $y = 2$ for column a and $y = 5$ for column b . For *Lena* RMSE is equal to 4.78, for *Picture* is equal to 3.61, and for *Nature* is equal to 7.06.

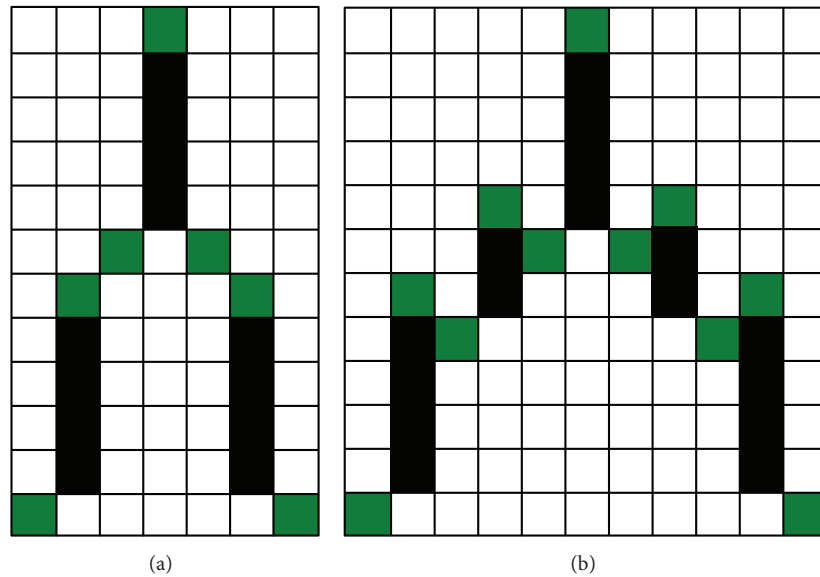


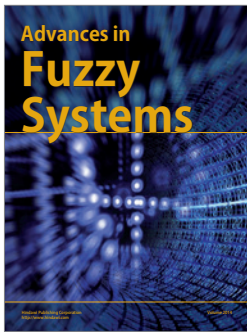
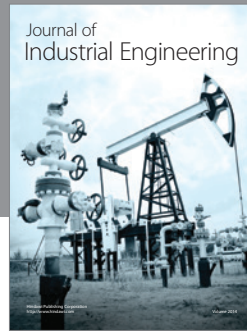
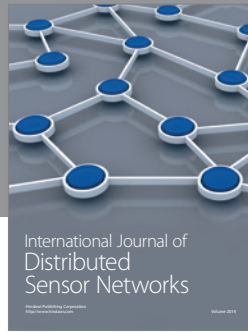
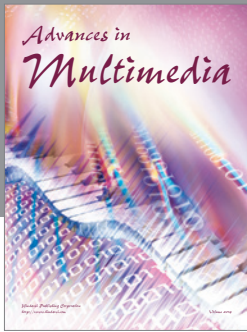
FIGURE 8: (a) Basic function for radius 2; (b) basic function for radius 4.

Acknowledgment

This work was supported by SGS14/PRF/2013 (advanced techniques of applications of soft computing methods in image processing).

References

- [1] P. Vlasanek, I. Perfilieva, and M. Wrublova, "Fuzzy transform for image reconstruction, Uncertainty modeling in knowledge engineering and decision making," in *Proceedings of the 10th International FLINS Conference*, pp. 615–620, World Scientific, Istanbul, Turkey, August 2012.
- [2] P. Vlasanek and I. Perfilieva, "Image reconstruction with usage of the F-Transform," in *Proceedings of the Advances in Intelligent Systems and Computing, International Joint Conference (CISIS'12-ICEUTE'12-SOCO '12)*, A. Herrero, V. Snášel, A. Abraham et al., Eds., vol. 189, pp. 507–514, Springer, 2012.
- [3] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolating methods for image resampling," *IEEE Transactions on Medical Imaging*, vol. MI-2, no. 1, pp. 31–39, 1983.
- [4] K. Uhlir and V. Skala, "Radial basis function use for the restoration of damaged images," in *Computer Vision and Graphics*, K. Wojciechowski, B. Smolka, H. Palus et al., Eds., vol. 32, pp. 839–844, Kluwer Academic, 2006.
- [5] F. Di Martino, V. Loia, I. Perfilieva, and S. Sessa, "An image coding/decoding method based on direct and inverse fuzzy transforms," *International Journal of Approximate Reasoning*, vol. 48, no. 1, pp. 110–131, 2008.
- [6] I. Perfilieva, "Fuzzy transforms: theory and applications," *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 993–1023, 2006.
- [7] I. Perfilieva and P. Vlasanek, *Image Reconstruction by means of F-transform*, Elsevier Editorial System for Knowledge-Based Systems, 2013.
- [8] P. Vlasanek and I. Perfilieva, "Influence of various types of basic functions on image reconstruction using F-transform," *European Society for Fuzzy Logic and Technology*. In press.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

