2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

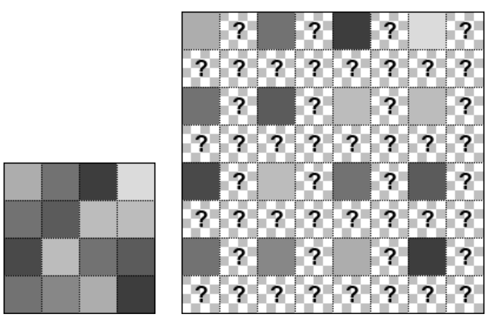
Interpolation techniques versus F-transform in application to image reconstruction

Vlašánek Pavel and Perfilieva Irina

***Abstract*— Many interpolation techniques are available for image reconstruction, with differences in time complexity, memory complexity and quality. In this article, we compare the application of bilinear interpolation, nearest neighbor interpolation and the F-transform approximation technique to the problem of image reconstruction. Based on our results, F-transform achieves the best results in terms of quality.**

I. Introduction

A typical technique in image reconstruction is resampling that is specified as up- or downsampling. When we upsample an image, we obtain many pixels with unknown values, and when we downsample an image, we must discard many pixels. Fig. 1 shows a 4×4 input image and its upsampled 8×8 version. The unknown pixels are marked by "?". Their intensities are not presented in the original image and must be computed.



(a) (b)

Fig. 1. a) Input image, 4×4 pixels; b) upsampled image to 8×8 pixels in the regular grid.

The most common resampling technique is interpolation [2], [3], [4]. It is often used in, e.g., medical [1] image processing. In our contribution, we will focus on the problem of image reconstruction, which is similar to the problem of upsampling. Both are focused on the replacement of damaged (reconstruction) or unknown (upsampling) pixels in an image, with values computed from neighboring pixels. In reconstruction, if we discard the damaged pixels (saying that they are "new"), then the computation of their intensities is performed on the same basis as resampling. Thus, reconstruction uses pixels from a neighborhood of damaged ones and techniques of interpolation, extrapolation or approximation.

Vlašánek Pavel is with Dept. of Informatics and Computers, University of Ostrava, Ostrava, Czech Republic; Perfilieva Irina is with the University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, NSC IT4Innovations, Ostrava, Czech Republic (email: {pavel.vlasanek, irina.perfilieva}@osu.cz).

In our previous work [17], we showed that the technique of F-transform is highly suitable for image reconstruction. We compared it to the technique of RBF -interpolation and showed its advantages in speed and quality. The aim of this research is to compare the effectiveness of the F-transform technique and of simple interpolation techniques such as the nearest neighbor and bilinear interpolation techniques. They are often used in upsampling where the known pixels establish a regular grid. In this contribution, we will extend these interpolation techniques to the case of a non-regular grid, which is more common in reconstruction problems. We will compare the extended interpolation techniques with the F-transform technique. We compare their qualities by RMSE[1] and SSIM[2][5]. The value of RMSE expresses a distance between the reconstructed and original images. The value of SSIM is computed on the basis of a more advanced technique that considers the perception abilities of the human eye. Let us adopt the following notation and use it throughout the paper. A partially damaged image u is a discrete function that is defined on a domain P and damaged on a domain P^d. The characteristic function of P^d is called the "mask". The goal of image reconstruction is to produce an image $\hat{u}$ that is defined on $P \cup P^d$ and coincides with u on P. In other words, for all $(i,j) \in P, u(i,j) = \hat{u}(i,j)$. Let $u(Q)$ denote the intensity of pixel Q in the range $\{0, 1, \ldots, 255\}$.

II. Interpolation

We extend the above interpolation techniques to an irregular grid. In figures Fig. 1 "a)" and Fig. 2, we demonstrate two types of distributions of known pixels: regular and irregular grids. In the case of a regular grid, computing the intensities of unknown pixels is easy due to the available analytic expressions for interpolation or approximation techniques. For example, if we want to double the size of our image, there is a given pattern of known pixels, as seen in Fig. 1 "b)". The extension of an irregular grid will be described next. As a demonstration, we will use the image of Lena (USC-SIPI Image Database), which we artificially damaged (see Fig. 3) with the mask shown in Fig. 9, case a).

A. *Nearest neighbor*

The new intensities of the pixels are determined as follows

$$\hat{u}(i,j) = u(Q_{op}),$$

[1]RMSE stands for the root mean square error. The lower the value of RMSE, the better the quality of the reconstruction is. $RMSE = 0$ represents identical images.

[2]SSIM stands for the structural similarity. Its values are in the range $[0, 1]$; higher values are better.

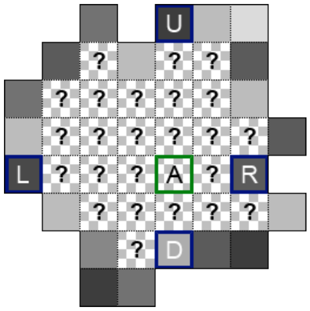


Fig. 2. Irregular grid of known pixels.

Fig. 3. Corrupted image of Lena.

where Q_{op} is the nearest known pixel with respect to the pixel at position (i, j). Because we want to minimize the complexity of this technique, we will ignore diagonal directions. Let us assume that we want to compute the intensity of the (unknown) pixel marked as A in Fig. 2. First, we will find the nearest two known pixels $Q(x, z_0)$ and $Q(t_0, y)$ in the vertical and horizontal directions, respectively, where

$$z_0 = \arg\min_z(|A(x, y) - Q(x, z)|),$$
$$t_0 = \arg\min_t(|A(x, y) - Q(t, y)|).$$

Then, the nearest known pixel $Q(x*, y*)$ is chosen as follows:

$$Q(x*, y*) = \begin{cases} Q(x, z_0), & \text{if } |y - z_0| \leq |x - t_0|, \\ Q(t_0, y), & \text{otherwise .} \end{cases}$$

The result for the example in Fig. 3 is shown in Fig. 4.

B. Bilinear interpolation

Let us assume that we have four given pixels $Q_{00} = (i_0, j_0), Q_{01} = (i_0, j_1), Q_{10} = (i_1, j_0), Q_{11} = (i_1, j_1)$. Moreover, $i_0 \leq i \leq i_1$ and $j_0 \leq j \leq j_1$.

$$\begin{aligned} \hat{u}(i, j) = \tfrac{1}{(i_1 - i_0)(j_1 - j_0)}(\quad & u(Q_{00})(i_1 - i)(j_1 - j) + \\ & u(Q_{10})(i - i_0)(j_1 - j) + \\ & u(Q_{01})(i_1 - i)(j - j_0) + \\ & u(Q_{11})(i - i_0)(j - j_0)) \end{aligned}$$

In the irregular grid, we compute the linear interpolation for all unknown rows and columns. That is, for every unknown

Fig. 4. Lena after nearest neighbor interpolation of the unknown parts.

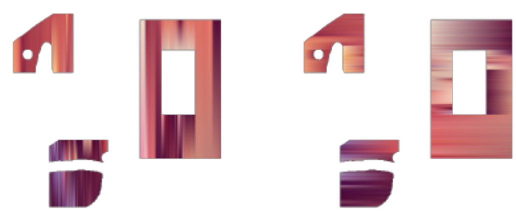
pixel, two values are available, one in each direction. The sum of these two intensities divided by 2 is used as the new intensity value of the unknown pixel. For our example in Fig. 2, we compute the linear interpolation for the row with pixel A as follows

$$w(x, y) = w(x-1, y) + \frac{R_x - L_x}{u(R) - u(L)}; \quad w(L_x, L_y) = u(L)$$

and for the column as

$$n(x, y) = n(x, y-1) + \frac{D_y - U_y}{u(D) - u(U)}; \quad n(U_x, U_y) = u(U)$$

, where the subscript x or y stands for the x or y coordinate of the point. The results of these two interpolation directions are shown in Fig. 5 for the example image Fig. 3. The whole



(a) (b)

Fig. 5. a) Linear interpolation of the columns; b) linear interpolation of the rows.

reconstructed image of Lena is in Fig. 6.

III. IMAGE APPROXIMATION

In comparison with image interpolation, image approximation produces image u_{app}, which differs from u on the domain $P \cup P^d$. It is important that u_{app} and u are near each other on P. The final reconstruction $\hat{u} = u_{app}|_{P^d}$. We propose to apply the F-transform technique [14] to produce u_{app}. This technique will then be compared with the interpolation methods described in the previous sections.

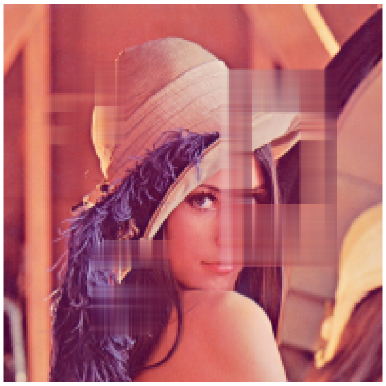


Fig. 6. Lena after bilinear interpolation of the unknown parts.

A. F-transform

In the last ten years, the theory of F-transforms has been intensively developed in many directions [6], [7], [8], [9], [10], [11], [12], [13]. In image processing, it has had successful applications in image compression and reduction, image fusion, edge detection and image reconstruction [14], [6], [15], [16], [17], [18]. The F-transform is a technique that places a continuous/discrete function in correspondence with a finite vector of its F-transform components. In image processing, where images are identified by intensity functions of two arguments, the F-transform of the latter is given by a matrix of components. We recall the definition of the F-transform [14] and provide it for a function of two variables defined on the set of pixels $P = \{(i,j) \mid i,j = 0,1,\ldots,255\}$. First, we recall the definition of a fuzzy partition [14]. In this research, we use the one with the Ruspini condition. Let us recall that a fuzzy set on X is identified with its membership function, which is a mapping from X to $[0,1]$.

1) Fuzzy partition with Ruspini condition: A *fuzzy partition with the Ruspini condition* (simply, *Ruspini partition*) was introduced in [14]. The Ruspini condition implies the normality of the respective fuzzy partition, i.e., the "partition-of-unity". It then leads to a simplified version of the inverse F-transform.

Definition 1: Let $x_1 < \ldots < x_n$ 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) $A_k : [a,b] \to [0,1]$, $A_k(x_k) = 1$;
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 $x \in [a,b]$,

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

The condition (1) is known as the Ruspini condition. The membership functions $A_1,\ldots,A_n$ are called *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 Figure 7, two such partitions with triangular and cosine basic functions are shown. We say that a Ruspini partition

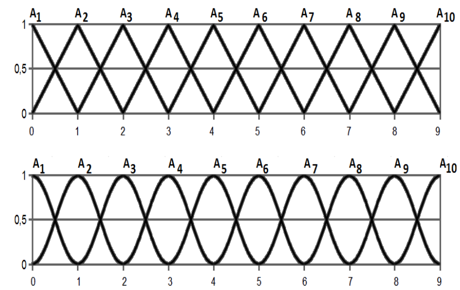


Fig. 7. Two Ruspini partitions with triangular (left) and cosine (right) basic functions.

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

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 $x \in [x_k, x_{k+1}]$.

An h-uniform fuzzy partition of $[a,b]$ can be determined by the so called *generating function* $A_0 : [-1,1] \to [0,1]$, which is assumed to be *even*[3], continuous, have a bell shape and fulfill $A_0(0) = 1$. The 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 $A_k(x) = A_0(\frac{x-x_k}{h})$, where $A_k(x) > 0$. From this point forward, we will be using h-uniform fuzzy partitions only and refer to h as a *radius* of partition.

B. Discrete F-transform

In this section, we introduce the F-transform of an image u that is considered as a function $u : [0,N] \times [0,N] \to [0,1]$, where $N = 255$. It is assumed that the image is gray-scaled and that it is defined at points (pixels) that belong to the set $P = \{(i,j) \mid i,j = 0,1,\ldots,N\}$. Let $A_1,\ldots,A_n$ and $B_1,\ldots,B_m$ be basic functions and $A_1,\ldots,A_n : [0,N] \to [0,1]$ and $B_1,\ldots,B_m : [0,N] \to [0,1]$ be two fuzzy partitions of $[0,N]$ (not necessarily different). Assume that the set of pixels P is *sufficiently dense with respect to the chosen partitions*, which means that $(\forall k)(\exists i \in [0,N])\ A_k(i) > 0$, and $(\forall l)(\exists j \in [0,N])\ B_l(j) > 0$. We say

[3] The function $A_0 : [-1,1] \to \mathbb{R}$ is even if, for all $x \in [0,1]$, $A_0(-x) = A_0(x)$.

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 $k = 1, \dots, n,\ l = 1, \dots, m$,

$$U_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} u(p_i, q_j) A_k(p_i) B_l(q_j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k(p_i) B_l(q_j)}. \tag{2}$$

The elements U_{kl} are called the *components of the F-transform*. The *inverse F-transform* $\hat{u} : P \to [0,1]$ of the function u with respect to $\{A_1, \dots, A_n\}$ and $\{B_1, \dots, B_m\}$ is defined as follows:

$$\hat{u}(i,j) = \sum_{k=1}^{n} \sum_{l=1}^{m} U_{kl} A_k(i) B_l(j). \tag{3}$$

The function $\hat{u}$ approximates the original function u on the whole domain $P = \{(i,j) \mid i,j = 0,1,\dots,N\}$ with a given precision. Moreover, the following estimate was established in [19] for every continuous function u on a domain P and its inverse F-transform $\hat{u}$, computed with respect to h-uniform fuzzy partitions $\{A_1, \dots, A_n\}$ and $\{B_1, \dots, B_m\}$ of $[0, N]$:

$$\max_{t \in P} |\hat{u}(t) - u(t)| \le C\omega(h, u), \tag{4}$$

where C is a constant, $t = (i,j)$ and $\omega(h,u)$ is the modulus of continuity of u on P.[4] Formula (4) shows that the smaller the value of h, the better the estimate of the difference between u and $\hat{u}$ is. These facts together justify the reconstruction methods described below. The result of the approximation for the example Fig. 3 is shown in Fig. 8.

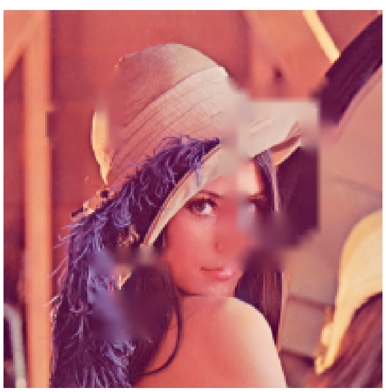


Fig. 8. Lena after F-transform approximation of the unknown parts.

C. Description of the algorithm

We propose an algorithm to produce a reconstruction as a result of combining a non-damaged part of an original image with several inverse F-transforms, computed on a sequence of uniform fuzzy partitions with increasing radii. The main idea of the algorithm is as follows: in the first step, we choose the finest h-uniform fuzzy partition of P, apply the F-transform and reconstruct the damaged pixels $(i,j) \in P^d$ that satisfy the following property:

- there are basic functions A_k and B_l, and there is a pixel $(i',j') \in P \setminus P^d$ such that $A_k(i) \cdot A_k(i') > 0$ and $B_l(j) \cdot B_l(j') > 0$.

We then recompute the damaged area P^d by deleting the already reconstructed pixels and, if P^d is not empty, repeat the procedure with a larger value of h. The following describes the reconstruction algorithm that takes u and the characteristic function m_{P^d} of P^d (called the mask) as inputs. The output will be the reconstruction $\hat{u}$. The algorithm uses the notation introduced above.

1) Choose radius $h = 2$.
2) Establish an h-uniform fuzzy partition $A_1, \dots, A_n$ and $B_1, \dots, B_m$ of P.
3) Compute the inverse F-transform $\hat{u}$ of image u.
4) Update P^d by deleting the reconstructed pixels, and update the mask m_{P^d}.
5) Update the image $\hat{u}$. If the mask is NOT identically equal to 0, then update the radius $h := h + 1$ and proceed to Step 2. Otherwise, proceed to Step 7.
6) Print output.

IV. Results

All techniques introduced above were tested on a set of 55 color images[5] with three types of damaged parts, according to their masks shown in Fig. 9. Fig. 10-13 show examples

(a) (b) (c)

Fig. 9. a) Larger contiguous areas; b) smaller contiguous areas; c) noise

of the four images from the testing set damaged by the masks shown in Fig. 9. There is a clear and visible difference in the smoothness of the reconstructed parts. The principal features are as follows: non-connected reconstruction by nearest neighbor interpolation, long linear stripes in the case of bilinear interpolation and blurred connections in the case of the F-transform. Tab. I, II, III show the results of the comparison between the reconstructed and original images for three chosen types of damage. Our conclusion is split into two parts. a)

- From the quality of reconstruction point of view, (measured by RMSE or SSIM)
  - the F-transform is the best technique in all tested cases, with the largest difference in damage type ”c)” (the best mean values in all tables are marked by the bold font);
  - of the interpolation methods, the bilinear interpolation shows better results than the nearest neighbor

[4]Generally, $\omega(h,f) = \max_{|\delta| \le h} \max_{x \in X} |f(x+\delta) - f(x)|$.

[5]`http://decsai.ugr.es/cvg/dbimagenes/c512.php`

(a) (b) (c)

Fig. 10. Reconstruction of the image *anhinga*. Rows from top to bottom show masks a), b) and c) from Fig. 9. The technique used for reconstruction is the same throughout a column, where a) F-transform, b) bilinear interpolation and c) nearest neighbor interpolation.

(a) (b) (c)

Fig. 11. Reconstruction of the image *athens*. Rows from top to bottom show masks a), b) and c) from Fig. 9. The technique used for reconstruction is the same throughout a column, where a) F-transform, b) bilinear interpolation and c) nearest neighbor interpolation.

interpolation for all types of damages “a)”, “b)” or “c)”.

- From the runtime point of view (measured in seconds), the F-transform is the slowest technique. However, on a typical computer with a 2.5 GHz CPU and 2 GB RAM, the reconstructions of 512×512 images with the damage type “b)” or “c)” is performed in under 1 sec. Damage type “a)” requires more than 2 seconds.

From the visual perception perspective, the F-transform provides smooth and clear output in comparison with the above interpolations.

V. Conclusion

We have compared two interpolation techniques, nearest neighbor and bilinear, with the F-transform technique for the problem of the reconstruction of damaged areas in images. A set of 55 color images with size 512×512 has been tested. Three differently distributed types of damaged areas were applied. The results of reconstruction are compared in tables I, II and III from the perspectives of quality and runtime. From the quality perspective (measured by RMSE and SSIM), we conclude that the F-transform shows the best results. In detail, the mean values of RMSE (the smaller, the better) for bilinear interpolation and F-transform for damage type ”c)” are

$$\text{bilinear: } 19.625; \quad \text{F-transform: } 16.836,$$

and the mean values of SSIM (the higher, the better) are

$$\text{bilinear: } 0.8975; \quad \text{F-transform: } 0.9240.$$

In conclusion, we recommend the F-transform as a solution for image reconstruction. The F-transform with linear basic functions provides smooth output and high quality. Moreover, the best result in comparison with the above interpolations is achieved in the case of damage type “noise”. Future research will focus on comparing the F-transform with other interpolation techniques.

VI. Acknowledgement

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and SGS18/PRF/2014 (Advanced techniques of applications of soft computing methods in image processing).

References

[1] T. M. Lehmann and C. Gonner and K. Spitzer, ”Survey: Interpolation methods in medical image processing”, *IEEE Transactions on Medical Imaging*, 18, pp. 1049-1075, 1999

[2] A. Amanatiadis and I. Andreadis, ”A survey on evaluation methods for image interpolation”, *Measurement Science and Technology*, 20, 2009

[3] P. Thevenaz and T. Blu and M. Unser: ”Interpolation revisited”, *IEEE Transactions on Medical Imaging*, 19, pp. 739–758, 2000

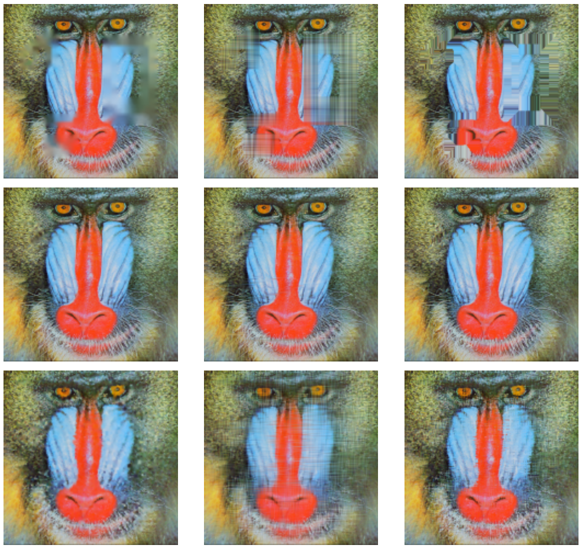
[4] A. Amanatiadis and I. Andreadis, ”Performance evaluation techniques for image scaling algorithms”, *Imaging Systems and Techniques - Workshop on Imaging Systems and Techniques*, pp 114–128, 2008

[5] Z. Wang and A. C. Bovik and H.R. Sheikh and E. P. Simoncelli, ”Image Quality Assessment: From Error Visibility to Structural Similarity,” *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13, pp. 600–612, 2004.

[6] F. Di Martino and V. Loia and I. Perfilieva and S. Sessa, ”An image coding/decoding method based on direct and inverse fuzzy transforms,” *International Journal of Approximate Reasoning*, 48, pp. 110–131, 2008.

[7] I. Perfilieva and V. Novák and A. Dvořák, ”Fuzzy transform in the analysis of data,” *Int. Journ. of Appr. Reasoning*, 48, pp 36–46, 2008.

[8] I. Perfilieva and B. De Baets, ”Fuzzy transform of monotonous functions,” *Information Sciences*, 180, pp 3304–3315, 2010.



(a) (b) (c)

Fig. 12. Reconstruction of the image *baboon*. Rows from top to bottom show masks a), b) and c) from Fig. 9. The technique used for reconstruction is the same throughout a column, where a) F-transform, b) bilinear interpolation and c) nearest neighbor interpolation.

[9] F. Di Martino and V. Loia and S. Sessa, "A segmentation method for images compressed by fuzzy transforms," *Fuzzy Sets and Systems*, 161, pp 56–74, 2010.

[10] V. Kreinovich and I. Perfilieva, "Fuzzy transforms of higher order approximate derivatives," *A theorem, Fuzzy Sets and Systems*, pp 55-68, 2011.

[11] I. Perfilieva and M. Daňková and B. Bede, "Towards F-transform of a higher degree" *Fuzzy Sets and Systems*, 180, pp 3–19, 2011.

[12] L. Stefanini, "F-transform with parametric generalized fuzzy partitions," *Fuzzy Sets and Systems*, 180, pp 98–120, 2011.

[13] M. Štěpnička and A. Dvořák and V. Pavliska and L. Vavřičková, "A linguistic approach to time series modeling with the help of F-transform," *Fuzzy Sets and Systems*, 180, pp 164–184, 2011.

[14] I. Perfilieva, "Fuzzy transforms: Theory and applications," *Fuzzy Sets and Systems*, 157, pp 993–1023, 2006.

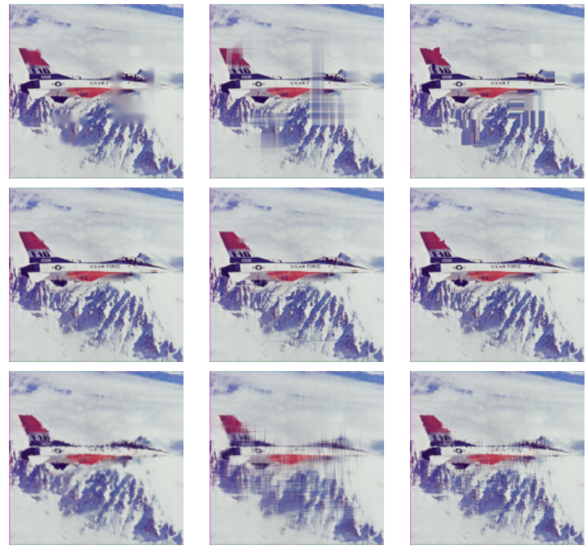
[15] M. Vajgl and I. Perfilieva and P. Hoďáková, "Advanced F-transform-based image fusion," *Advances in Fuzzy Systems*, 2012.

[16] I. Perfilieva and Hoďáková and P. Hurtik, "F^1-transform edge detector inspired by Canny's algorithm" *Advances on Computational Intelligence*, pp. 230–239, 2012.

[17] P. Vlašánek and I. Perfilieva, "Image reconstruction with usage of the F-Transform," *International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*, pp 507–514, 2012.

[18] P. Vlašánek and I. Perfilieva and M. Wrublová, "Fuzzy transform for image reconstruction" *Uncertainty Modeling in Knowledge Engineering and Decision Making*, pp 615–620, 2012.

[19] I. Perfilieva, "Fuzzy transforms: A challenge to conventional transforms," *Advances in Images and Electron Physics*, 147, pp. 137–196, 2007.



(a) (b) (c)

Fig. 13. Reconstruction of the image *avion*. Rows from top to bottom show masks a), b) and c) from Fig. 9. The technique used for reconstruction is the same throughout a column, where a) F-transform, b) bilinear interpolation and c) nearest neighbor interpolation.

	/Larger contiguous areas		
	SSIM		
Stat	Nearest	Bilinear	F-transform
Min.	0.8421	0.8585	0.8684
1st Qu.	0.9061	0.9203	0.9348
Median	0.9245	0.9382	0.9449
Mean	0.9215	0.9349	**0.9426**
3rd Qu.	0.9416	0.9481	0.9573
Max.	0.9816	0.9802	0.9862
	RMSE		
Stat	Nearest	Bilinear	F-transform
Min.	8.355	6.707	7.372
1st Qu.	15.010	13.248	12.610
Median	18.051	15.580	14.800
Mean	18.695	16.604	**15.735**
3rd Qu.	21.432	19.413	17.997
Max.	31.906	31.429	28.753
	time (ms)		
Stat	Nearest	Bilinear	F-transform
Min.	249.0	124.0	2138
1st Qu.	265.0	125.0	2184
Median	266.0	140.0	2200
Mean	271.7	134.0	2210
3rd Qu.	281.0	140.2	2231
Max.	297.0	156.0	2293

TABLE I

SSIM, RMSE AND RUNTIME FOR DAMAGE TYPE "A)"

	Smaller contiguous areas		
	SSIM		
Stat	Nearest	Bilinear	F-transform
Min.	0.9597	0.9704	0.9735
1st Qu.	0.9801	0.9859	0.9862
Median	0.9882	0.9913	0.9910
Mean	0.9863	0.9897	**0.9902**
3rd Qu.	0.9927	0.9944	0.9946
Max.	0.9990	0.9988	0.9981
	RMSE		
Stat	Nearest	Bilinear	F-transform
Min.	3.010	3.184	3.201
1st Qu.	5.288	4.796	4.488
Median	6.966	6.276	6.156
Mean	7.392	6.419	**6.237**
3rd Qu.	8.670	7.600	7.177
Max.	14.880	12.460	11.812
	time (ms)		
Stat	Nearest	Bilinear	F-transform
Min.	124.0	124.0	842.0
1st Qu.	125.0	125.0	873.0
Median	140.0	125.0	874.0
Mean	133.4	130.8	879.2
3rd Qu.	140.0	140.0	889.0
Max.	156.0	156.0	920.0

TABLE II

SSIM, RMSE AND RUNTIME FOR DAMAGE TYPE “B)”.

	Noise		
	SSIM		
Stat	Nearest	Bilinear	F-transform
Min.	0.7057	0.7461	0.7866
1st Qu.	0.8401	0.8662	0.8960
Median	0.8848	0.9066	0.9354
Mean	0.8783	0.8975	**0.9240**
3rd Qu.	0.9294	0.9389	0.9641
Max.	0.9755	0.9774	0.9923
	RMSE		
Stat	Nearest	Bilinear	F-transform
Min.	10.99	9.364	8.485
1st Qu.	16.36	14.429	11.821
Median	21.81	19.046	15.816
Mean	22.51	19.625	**16.836**
3rd Qu.	26.50	22.932	20.083
Max.	40.45	34.206	32.913
	time (ms)		
Stat	Nearest	Bilinear	F-transform
Min.	218.0	156.0	827.0
1st Qu.	234.0	171.0	842.0
Median	249.0	172.0	850.5
Mean	244.3	172.1	852.7
3rd Qu.	250.0	172.0	858.0
Max.	266.0	187.0	905.0

TABLE III

SSIM, RMSE AND RUNTIME FOR DAMAGE TYPE “C)”.