

# Influence of various types of basic functions on image reconstruction using F-transform

Pavel Vlašánek<sup>1</sup> Irina Perfilieva<sup>2</sup>

<sup>1</sup>Department of Informatics and Computers, University of Ostrava, Czech Republic

<sup>2</sup>Centre of Excellence IT4Innovations - Division of the University of Ostrava - IRAFM, Czech Republic

## Abstract

Technique of image reconstruction using F-transform uses basic functions for computation. Radius and partition of these functions affects quality of the reconstruction. In this article we are focusing on influence of the basic function shape on the quality of reconstruction and providing results of nondecreasing, nonincreasing and oscillating examples.

**Keywords:** F-transform, basic function, image reconstruction

## 1. Introduction

In general, there are many choices and possibilities in reconstruction process which consists in removing damaged parts from input image and replacing it by a part with similar to original values of removed pixels. Areas which include damaged pixels which we want to remove should be recomputed and replaced by new pixels based on the neighborhood ones. F-transform may be used for this purpose [1] and will be elaborated in the article. Various methods [2] differ by the choice at the technique, type of damage, etc. Various types of damage covers stains, scratches, text and noise as shown in figure 1.

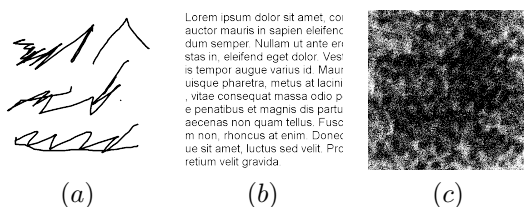


Figure 1: a) inpaint damage; b) text damage; c) noise damage

We will focused on increasing quality of reconstruction measured by RMSE value<sup>1</sup>. In figure 2 you can see input image and result. We will analysed influence of changing parametr on target quality.

<sup>1</sup>RMSE stands for the Root Mean Square Error.

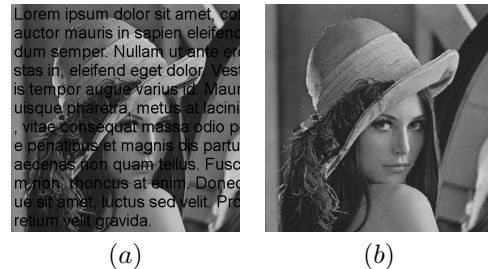


Figure 2: a) image damaged by text; b) result after reconstruction

## 2. F-transform

In image reconstruction a discrete version of the F-transform is used. Details can be seen in [3, 4]. 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  $x \in [a, b]$ ,

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

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:

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

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

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

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

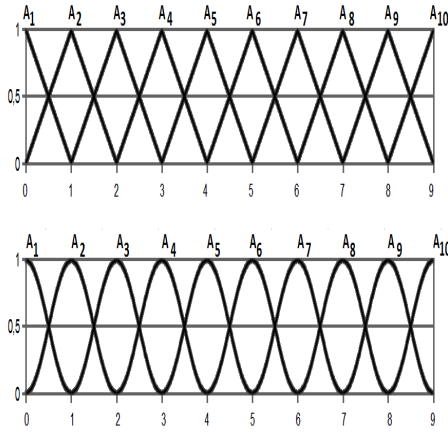


Figure 3: Two Ruspini partitions with triangular (up) and cosine (down) basic functions.

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*, i.e.,  $x_k = a + h(k-1)$ , for  $k = 1, \dots, n$ , where  $h = (b-a)/(n-1)$ , and two additional properties are met:

6.  $A_k(x_k - x) = A_k(x_k + x)$ , for all  $x \in [0, h]$ ,  $k = 2, \dots, n-1$ ,
7.  $A_k(x) = A_{k-1}(x - h)$ , for all  $k = 2, \dots, n-1$  and  $x \in [x_k, x_{k+1}]$ , and  $A_{k+1}(x) = A_k(x - h)$ , for all  $k = 2, \dots, n-1$  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*  $A_0 : [-1, 1] \rightarrow [0, 1]$ , which is assumed to be *even*<sup>2</sup>, 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  $A_0$

<sup>2</sup>The function  $A_0 : [-1, 1] \rightarrow \mathbb{R}$  is even if for all  $x \in [0, 1]$ ,  $A_0(-x) = A_0(x)$ .

in the sense that

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

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

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

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

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 [5].

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

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

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

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

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

### 3. Image reconstruction

From mathematical point of view image is a two dimensional discrete function, say  $u : [0, N] \times [0, N] \rightarrow [0, 255]$ . Every pixel has the coordinates  $x, y$  and color. In grayscale image it is usual to call color as *intensity*. It happens that an image is damaged. In this case we assume that the damaged part can be separated from undamaged. It always helps, if a damage can be classified into classes: textover, scratches, noise, etc. The purpose of image reconstruction is to remove the damaged part and to replace it by the part which is computed from the local neighborhood. In real life it may be a photography without inscribed time stamp, a fresco without cracks or a surface without stains for example.

The idea of image reconstruction consists in interpolation [6] or approximation [7] of the missing intensity from the closest neighborhood. Because of that we have to distinguish between damaged and undamaged part. Let us assume that we are able to create another image called *mask* which includes only black and white pixels. The mask has the same

dimension as the original image, and its black pixels cover places where the original image contains damage. In figure 4, you can see what is meant by this.

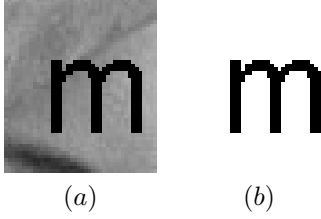


Figure 4: a) original image; b) mask

Because of discrete nature of the input data we have to use the discrete variant of F-transform and also inverse F-transform.

### 3.1. One-step reconstruction

Let  $u^c$  be a damaged image on the set  $P = \{(i, j) \mid i, j = 0, 1, \dots, N\}$  with the damaged part located at  $P^c \subset P$ . The following mask will be associated with  $P^c$ :

$$m^c(i, j) = \begin{cases} 0, & \text{if } (i, j) \in P^c, \\ 1, & \text{otherwise.} \end{cases}$$

Let us compute a new (combined) image  $u$  where

$$\begin{aligned} u(i, j) &= \min(u^c(i, j), m^c(i, j)) = \\ &= \begin{cases} 0, & \text{if } (i, j) \in P^c, \\ u^c(i, j), & \text{otherwise.} \end{cases} \end{aligned}$$

The following text is an informal description of an algorithm that takes  $u^c$  and  $m^c$  as inputs and computes the reconstruction  $u^r$  as a combination of the non-damaged part of  $u^c$  with the inverse F-transform of the combined image  $u$ .

Let us choose a uniform fuzzy partition  $A_1, \dots, A_n$  and  $B_1, \dots, B_m$  of  $P$  that fulfills the following property:

1. for every damaged pixel  $(i, j) \in P^c$  there are basic functions  $A_k$  and  $B_l$  and there is pixel  $(i', j') \in P \setminus P^c$  such that  $A_k(i) > 0$ ,  $A_k(i') > 0$ ,  $B_l(j) > 0$ ,  $B_l(j') > 0$ .

This means that every damaged pixel is covered by some combination of basic functions such that this combination also covers at least one non-damaged pixel. Then the inverse F-transform  $\hat{u}$  of the combined image  $u$  replaces the zero value of the mask  $m^c$  at every damaged pixel  $(i, j)$  in  $u$  by the value of  $\hat{u}(i, j)$  that is computed using values of  $u$  at non-damaged pixels  $(i', j')$ . The output reconstructed image  $u^r$  is a combination of  $u^c$  and  $\hat{u}$ :

$$u^r(i, j) = \begin{cases} \hat{u}(i, j), & \text{if } (i, j) \in P^c, \\ u^c(i, j), & \text{otherwise.} \end{cases}$$

Type of damage	One-Step	Multi-Step
Noise	14.6	11.72
Text over image	5.19	4.56
Scratch	5.62	4.87

Table 1: RMSE values of the one- and multi-step reconstructions.

### 3.2. Multi-step reconstruction

The one-step reconstruction can be successively applied to those images that are damaged on relatively small areas. In the above formulated assumption, we formalized this condition demanding that every damaged pixel should be covered by a combination of basic functions such that it covers at least one non-damaged pixel (property **P**). However, it is not always that a fuzzy partition with this property exists. Therefore, we propose another algorithm which produces reconstruction as a result of combination of 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 following is a description of the multi-step reconstruction algorithm that takes  $u^c$  and  $m^c$  as inputs and computes the reconstruction  $u^r$ . The algorithm uses the introduced above denotation.

1. Compute the combined image  $u$  where  $u(i, j) = \min(u^c(i, j), m^c(i, j))$ . Choose radius  $h = 2$ .
2. Establish a  $h$ -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 the combined image  $u$ .
4. Compute the corresponding to  $h$  reconstruction  $u^h$ : replace the zero value of a damaged pixel  $(i, j)$  by the value  $\hat{u}(i, j)$ .
5. Update the mask  $m^c$  by deleting pixels whose reconstructed values are strictly greater than zero.
6. Update the combined image  $u$  so that  $u(i, j) = \min(u^h(i, j), m^c(i, j))$ . If the mask is NOT identically equal to 0, then update the radius  $h := h + 1$  and go to Step 2. Otherwise go to Step 7.
7. Put  $u^r = u^h$  and print output.

Besides simplifying the problem of choosing radius of a uniform fuzzy partition, the multi-step reconstruction has better quality than the one-step reconstruction. This follows from the estimate that shows that the smaller is the value of a radius, the closer is the inverse F-transform  $\hat{u}$  to  $u$ . In table 1, we justify the above postulated claim numerically and demonstrate the RMSE values of the one- and multi-step reconstructions of the damaged image *Lena* with the three types of damage which we name as *noise*, *text over image* and *scratch*.

#### 4. Influence of basic functions

In experiments, we choose a generating function  $A_0$  of a partition (with the Ruspini condition) and create the corresponding partition by shifting  $A_0$  (section 2.2). Because we work with uniform partitions only, the generating functions are symmetrical. Therefore it is enough to choose their left parts. In experiments, we extend the variety of basic functions by excluding condition 4 in the definition in section 2.1. The F-transform based reconstructions with various basic functions were applied to the images of *Lena*, *cloth*, *drawing* and *nature* (as can be seen in figure 5) which has been damaged by *text* (Figure 2).

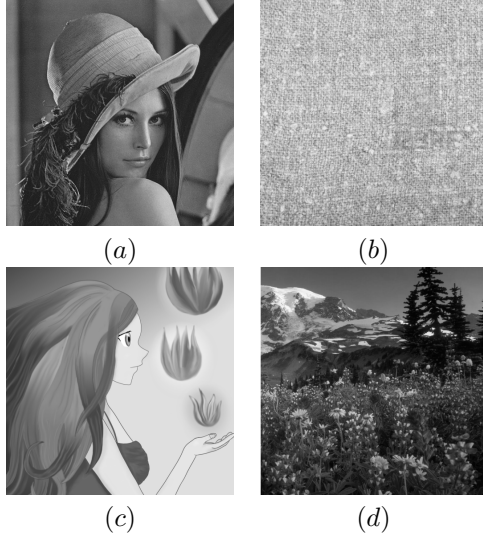


Figure 5: a) Lena; b) cloth; c) drawing; d) nature

We applied the multi-step reconstruction algorithm with radii from 2 to 6. The following shapes of generating functions were used:  $a_i, b_i, c_i, d_i$  (Figure 6) and  $a_o, b_o, c_o, d_o$  (Figure 8). For demonstration pictures *Lena* image was used.

In table 2, we show the values of the RMSE applied to the original (non-damaged) and reconstructed images.

Reconstruction has been performed by the F-transforms with basic functions generated by  $a_i, b_i, c_i, d_i$  (left nondecreasing) and  $a_o, b_o, c_o, d_o$  (oscillating). Symbols "-" in columns 5, 6 mean that all damaged pixels were reconstructed at preceding steps. Based on table 2 we can state that 4 is the maximal value of a radius of a basic function which solves the reconstruction problem for testing kind of damage.

From table 2, we conclude that:

- the best quality is achieved by generated function  $b_o, b_i, c_i$ , whose shape is *triangle* or almost *triangle*
- the worse quality is achieved by the generated function  $d_i$

f/r	2	3	4	5	6
Lena					
$a_i$	49.98	5.59	4.75	-	-
$b_i$	49.96	5.50	4.64	-	-
$c_i$	49.97	5.51	4.65	-	-
$d_i$	62.99	32.71	6.66	6.66	6.66
$a_o$	49.96	5.93	5.10	-	-
$b_o$	49.96	5.47	4.60	-	-
$c_o$	49.98	5.64	4.80	-	-
$d_o$	49.98	5.59	4.75	-	-
Cloth					
$a_i$	22.95	7.83	7.75	-	-
$b_i$	22.90	7.76	7.68	-	-
$c_i$	28.89	7.66	7.58	-	-
$d_i$	28.48	16.35	8.81	8.81	8.81
$a_o$	22.89	8.12	8.04	-	-
$b_o$	22.89	7.72	7.64	-	-
$c_o$	22.95	7.88	7.80	-	-
$d_o$	22.95	7.83	7.75	-	-
Drawing					
$a_i$	33.91	3.97	3.90	-	-
$b_i$	33.90	3.92	3.84	-	-
$c_i$	33.90	3.83	3.76	-	-
$d_i$	42.47	21.98	4.82	4.82	4.82
$a_o$	33.90	4.43	4.37	-	-
$b_o$	33.90	3.90	3.82	-	-
$c_o$	33.91	4.04	3.97	-	-
$d_o$	33.91	3.97	3.90	-	-
Nature					
$a_i$	56.37	7.60	6.96	-	-
$b_i$	56.35	7.51	6.86	-	-
$c_i$	56.35	7.39	6.72	-	-
$d_i$	71.25	37.25	9.22	9.22	9.22
$a_o$	56.35	8.15	7.55	-	-
$b_o$	56.35	7.46	6.80	-	-
$c_o$	56.37	7.67	7.03	-	-
$d_o$	56.37	7.61	6.96	-	-

Table 2: The values of the RMSE.

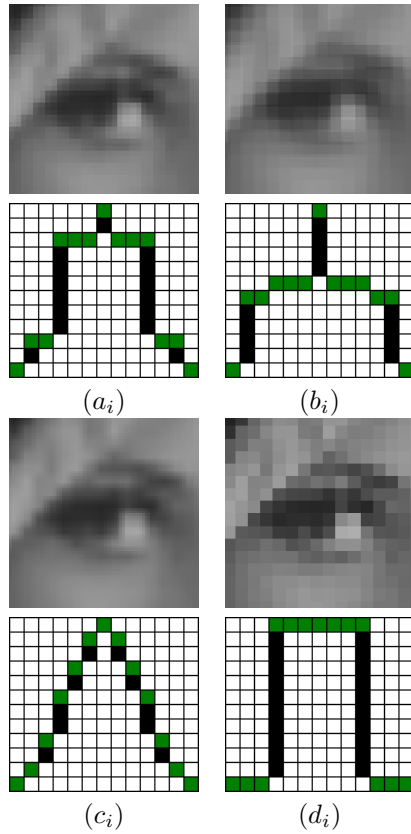


Figure 6: Nondecreasing basic functions and the corresponding inverse F-transforms (detail of Lena's eye).

- the quality of reconstruction is robust with respect to small oscillations in shapes of basic functions, compare the respective RMSE values of  $a_i$  and  $a_o$ ,  $b_i$  and  $b_o$ ,  $c_i$  and  $c_o$ ,  $d_i$  and  $d_o$ .

## 5. Conclusion

We analysed the influence of various types of basic functions on image reconstruction using F-transform. Based on the achieved results, we see that the triangular shaped basic functions are the best ones for the reconstruction problem. The F-transform based reconstructions whose basic functions have small oscillations  $b_i, c_i, b_o$  produce visual better results than their counterparts.

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

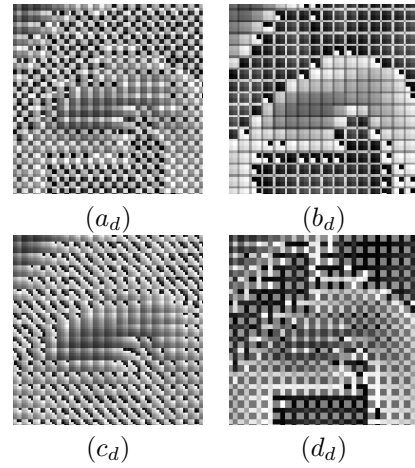


Figure 7: Nonincreasing basic function application (detail of Lena's eye). We have used the same basic functions as in nondecreasing example but reversed upside down.

## References

- [1] P. Vlasanek, I. Perfilieva, M. Wrublova, Fuzzy transform for image reconstruction, *Uncertainty Modeling in Knowledge Engineering and Decision Making*, 615–620, (World Scientific, 2012).
- [2] J. A. Parker, R. V. Kenyon, D. E. Troxel, Comparison of interpolating methods for image resampling, *IEEE Transactions on medical imaging*, (1983).
- [3] F. Di Martino, V. Loia, I. Perfilieva, S. Sessa, An image coding/decoding method based on direct and inverse fuzzy transforms, *International Journal of Approximate Reasoning*, 110–131, (2008).
- [4] I. Perfilieva, Fuzzy transforms: Theory and applications, *Fuzzy Sets and Systems*, 993–1023, (2006).
- [5] I. Perfilieva, P. Vlasanek, Image Reconstruction by means of F-transform, *Elsevier Editorial System for Knowledge-Based Systems*, Submitted, (2013).
- [6] K. Uhler, V. Skala, Radial basis function use for the restoration of damaged images, *Computer vision and graphics*, 839–844, (Springer, 2006).
- [7] P. Vlasanek, I. Perfilieva, Image reconstruction with usage of the F-Transform, *International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*, 507–514, (Springer, 2012).

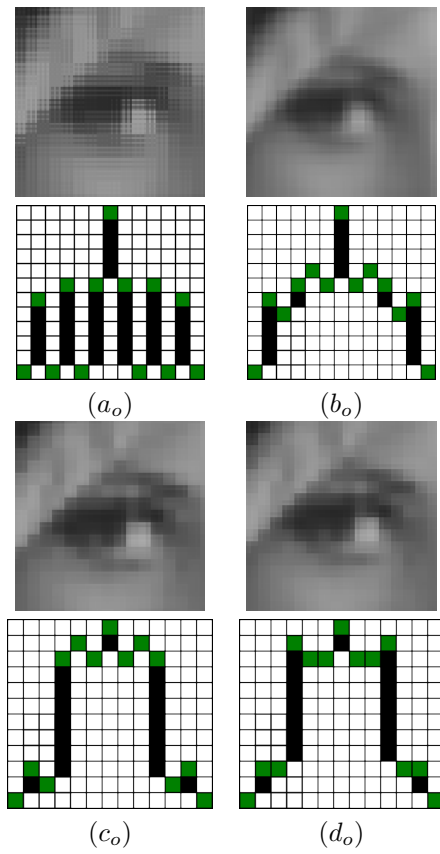


Figure 8: Oscillating basic function application (detail of Lena's eye).