Image Reconstruction using Partial Fuzzy Transform

Martina Daňková*, Viktor Pavliska†
National Supercomputing Centre IT4Innovations
Institute for Research and Applications of Fuzzy Modeling,
University of Ostrava
30. dubna 22, 701 03 Ostrava 1,
Czech Republic
Email: *martina.dankova@osu.cz, †viktor.pavliska@osu.cz

***Abstract*—We will introduce various extensions of ordinary binary operations to a specially chosen single dummy element representing an undefined or missing value. We will apply these extensions to the Perfilieva's fuzzy transform technique and extend it appropriately to operate with this dummy element too. The result of this extension will be called partial fuzzy transform. Moreover, we will demonstrate the usefulness of our approach using partial fuzzy transform technique on an image reconstruction problem.**

Keywords–Undefined; Missing data; Error propagation; Fuzzy transform; Image processing.

I. INTRODUCTION

Undefined or Missing (U/M) data as a source of various bugs are a common problem in most scientific research domains. Causes are various, e.g., computation exceptions, non-terminating computation, mishandling of samples, measurement error, non-response or division by zero. Such data are needed to be represented in order to be correctly handled. For this purpose, there exist various markers in programming languages, e.g., `Null`, `NaN`, `Inf`. In this paper, we propose to use a single dummy element denoted by $*$ to represent U/M data, and we extend binary functions to operate with $*$ also [1]. As an output, we obtain several families of binary functions due to variability of treating U/M data. Remark that these extensions are analogous to the connectives of a partial fuzzy logic [2].

Our main aim is to demonstrate the usefulness of the above-mentioned extensions. For this purpose, we chose the ordinary Fuzzy Transform (FT) technique introduced in [3] and its application to an image reconstruction problem introduced in [4]. A brief description of FT, algorithms and their implementations for Image Reconstruction (IR) using FT can be found in [5].

We have observed that algorithms for IR using FT in all available sources lack an explanation of what happens in case of exceptions in a computation process. We refer mainly to the problem of the division by 0 when computing components of the direct FT which often occurs in damaged parts of an image. Moreover, the description of Multi-Step Reconstruction Algorithm (MSRA) from [4] does not correspond with experimental results presented in [4]. It is because U/M parts of an input image are considered as having value 0 which influences the final infilling of U/M areas of images in each iteration step. The main problem lies in Step 5 of MSRA which states the following: "Update the mask by deleting pixels whose reconstructed values are strictly greater than zero". Fulfilling this requirement leads to gradual falling (with a growing number of iterations) of infilled values to 0 and consequently to the darkening of the centers of infilled areas (see Figure 5). However, this is not the case of the experimental results presented in [4]. Surprisingly, their results are almost identical with ours (see Figure 6). A discussion on differences between algorithms from [4] and our approach will be explained in Section VI.

In this paper, we will present algorithms for an image reconstruction based on FT which will be able to handle U/M parts of images represented by $*$. For this purpose, we will choose suitable extensions of elementary operations to $*$ according to our intuitive expectations on the behavior of FT on U/M parts of an image which will result in a definition of the so-called partial FT. In our approach, we encode U/M areas of an image by $*$ and reconstructed parts are computed only from the real image values.

The paper is organized as follows. In Section II, we recall three basic extensions of binary functions from [1]. Perfilieva's FT and partial FT are presented in Section III and IV, respectively. Next, in Section V, we introduce algorithms for image reconstruction using partial FT together with illustrative examples. Finally, features of the used formalism are summarized in Section VI.

II. EXTENSIONS OF OPERATIONS TO U/M DATA

Extensions of connectives to undefined truth values in partial fuzzy logic [2] can be analogously carried for an arbitrary binary function $o\colon X^2 \to X$ as follows:

$$\begin{array}{c|cc} o^{\mathrm{B}} & y & * \\ \hline x & o(x,y) & * \\ * & * & * \end{array} \qquad \begin{array}{c|cc} o^{\mathrm{S}} & y & * \\ \hline x & o(x,y) & x \\ * & y & * \end{array} \tag{1}$$

where $* \notin X$. We call o^{B} the *Bochvar-extension* of o, o^{S} the *Sobociński-extension* of o

Moreover, if $a \in X$ is an absorbing element of o, i.e., $o(a,x) = o(x,a) = a$ then, we introduce the *Kleene-extension* of o:

$$\begin{array}{c|ccc} o^{\mathrm{K}} & a & y & * \\ \hline a & a & a & a \\ x & a & o(x,y) & * \\ * & a & * & * \end{array} \tag{2}$$

Note that

- $o^{\mathrm{B}}, o^{\mathrm{S}}, o^{\mathrm{K}}$ operate on $(X \cup \{*\})^2$ and take values from $X \cup \{*\}$.
- These extensions are motivated by classical four valued logics.

 ISBN: 978-1-61208-709-2

- Sobociński-style operations treat U/M inputs encoded by $*$ as irrelevant and ignore them. In the case of Bochvar-style operations, $*$ represents a fatal error, and it terminates a computation. Kleene-style operations treat $*$ as a vincible error where $*$ is treated as a fatal error up to the case of an absorbing element which overwrites $*$.
- An extension should be chosen due to a required behavior of o in the case of U/M data.

III. Ordinary Fuzzy Transform

Recall that FT is a well-established technique in the image processing domain. Let us recall some applications, e.g., [6] introduces an algorithm for pattern matching and provides a comparison with existing methods, an image fusion was elaborated in [7], and an image contrast enhancement can be found in [8]. A higher order ordinary FT [9] was used in an edge detection problem [10], as well as in an improvement of image reconstruction method [11].

FT uses weighted arithmetic mean to compute transformation components and invert them as linear combination of components with their weights. We recall a discrete FT from [3] adjusted for 2-D case, where $X = [a, b] \times [c, d] \neq \emptyset \subset \mathbb{R}^2$ and a continuous function $f\colon X \to \mathbb{R}$ is given only at some non-empty finite set of points $D \neq \emptyset \subseteq X$.

Let $m, n \geq 2$, $\{x_i\}_{i=0}^{m+1}$ and $\{y_j\}_{j=0}^{n+1}$ be nodes such that $a = x_0 = x_1 < \ldots < x_m = x_{m+1} = b$ and $c = y_0 = y_1 < \ldots < y_n = y_{n+1} = d$. Consider two finite sets of fuzzy sets $\mathbf{A} = \{A_1, \ldots, A_m\}$ and $\mathbf{B} = \{B_1, \ldots, B_n\}$, where each A_k and B_l are identified with their membership functions $A_k\colon [a, b] \to [0, 1]$, $B_l\colon [c, d] \to [0, 1]$ for each $k = 1, \ldots, m$ and $l = 1, \ldots, n$.

Let $\mathbf{A}$ fulfill the following conditions, for each $k = 1, \ldots, m$:

1) Locality: $A_k(x) = 0$ if $x \in [a, x_{k-1}] \cup [x_{k+1}, b]$.
2) Continuity: A_k is continuous on $[a, b]$.
3) Normality: $A_k(x) = 1$ for some $x \in (x_{k-1}, x_{k+1})$.
4) Ruspini's condition: $\sum_{i=1}^{n} A_i(x) = 1$ for all $x \in [a, b]$.

We shall say that $\mathbf{A}$ form a fuzzy partition of $[a, b]$.

Often, we deal with the so called h-uniform fuzzy partition $\mathbf{A}$ (see [4]) determined by the generating function $A : [-1, 1] \mapsto [0, 1]$, which is assumed to be even, continuous, bell-shaped and fulfill $A(0) = 1$. Fuzzy sets from a h-uniform fuzzy partition are created as a shifted copy of a generating function A.

Definition 1 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *form fuzzy partitions of* $[a, b]$ *and* $[c, d]$*, respectively. Let* $m = |\mathbf{A}|$*,* $n = |\mathbf{B}|$*, and moreover, let* $\mathbf{A}$ *and* $\mathbf{B}$ cover D*, i.e., for each* $A \in \mathbf{A}$*,* $B \in \mathbf{B}$ *there exist* $(x, y) \in D$ *such that* $A(x) > 0$ *and* $B(y) > 0$*. Then, a (direct discrete)* fuzzy transform *of* f *w.r.t.* $\mathbf{A}$*,* $\mathbf{B}$ *and* D *is defined as a* $m \times n$ *matrix*

$$F_{m,n}[f] = \begin{bmatrix} F_{11} & F_{12} & \ldots & F_{1n} \\ F_{21} & F_{22} & \ldots & F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ F_{m1} & F_{m2} & \ldots & F_{mn} \end{bmatrix}$$

where the (k, l)*-th* component $F_{k,l}$ *is equal to*

$$F_{k,l} = \frac{\sum_{(c,d)\in D} f(c, d) A_k(c) B_l(d)}{\sum_{(c,d)\in D} A_k(c) B_l(d)} \tag{3}$$

for each $k = 1, \ldots, m$ *and* $l = 1, \ldots, n$*.*

Observe that the matrix of components consists of weighted arithmetic means with weights given by the values of the respective fuzzy sets from $\mathbf{A}$ and $\mathbf{B}$. A linear combination of components from $F_{m,n}[f]$ and fuzzy sets from $\mathbf{A}$ and $\mathbf{B}$ returns a continuous function on X called inverse fuzzy transform.

Definition 2 *The* inverse discrete fuzzy transform *of* f *w.r.t.* $\mathbf{A}$*,* $\mathbf{B}$ *and* D *is a function* $f_{F,m,n}\colon X \to \mathbb{R}$ *defined as*

$$f_{F,m,n}(x, y) = \sum_{k=1}^{m} \sum_{l=1}^{n} F_{k,l} \cdot A_k(x) \cdot B_l(y), \quad (x, y) \in X. \tag{4}$$

We refer to the direct and inverse discrete FT from the above definitions as the *ordinary* FT.

IV. Partial Fuzzy Transform

Since $f\colon X \to \mathbb{R}$ is given only at some non-empty finite set of points, hence, it is total on D and partial on X. Therefore, it can be considered undefined on $X \setminus D$. This fact can be formalized by extension of real-line by a dummy element $*$ (represents U/M value) to $\mathbb{R}_* = \mathbb{R} \cup \{*\}$ and f to $f^*\colon X \to \mathbb{R}_*$ as follows:

$$f^*(x) = \begin{cases} f(x) & \text{for } x \in D; \\ * & \text{otherwise.} \end{cases} \tag{5}$$

Now, we can introduce FT components for f^* as extended weighted arithmetic means.

Definition 3 *Let* $\mathbf{A}$*,* $\mathbf{B}$ *and* m, n *be as in Definition 1 and*

$$P_{kl} = \{(x, y) \in X | A_k(x) \cdot B_l(y) > 0\}.$$

Then, a (direct discrete) partial fuzzy transform *of* f^* *w.r.t.* $\mathbf{A}$*,* $\mathbf{B}$ *and* D *is defined as a matrix of* components

$$F^*_{m,n}[f^*] = \begin{bmatrix} F^*_{11} & F^*_{12} & \ldots & F^*_{1n} \\ F^*_{21} & F^*_{22} & \ldots & F^*_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ F^*_{m1} & F^*_{m2} & \ldots & F^*_{mn} \end{bmatrix}$$

where

$$F^*_{k,l} = \frac{\sum^{\mathrm{S}}_{(x,y)\in P_{kl}} f^*(x, y) \cdot^{\mathrm{B}} (A_k(x) \cdot B_l(y))}{\sum_{(c,d)\in D\cap P_{kl}} A_k(c) \cdot B_l(d)}^{\mathrm{B}} \tag{6}$$

for $k = 1, \ldots, n$*.*

In this definition, we do not consider $\mathbf{A}$ and $\mathbf{B}$ to cover D. It follows that $F^*_{k,l} \neq *$ if and only if there exists $(x, y) \in D$ such that $x \in [x_{k-1}, x_{k+1}]$ and $y \in [y_{l-1}, y_{l+1}]$.

Observe that (6) can be equivalently written as

$$F^*_{k,l} = \frac{\sum^{\mathrm{S}}_{(x,y)\in P_{kl}} f^*(x,y) \cdot^{\mathrm{B}} (A_k(x)\cdot B_l(y))}{\sum_{(x,y)\in P_{kl}} A_k(x)\cdot B_l(y)\cdot \chi_D(x,y)}^{\mathrm{B}} \quad (7)$$

where χ_D denotes characteristic function of D.

An inverse transformation of $F^*_{m,n}[f^*]$ to the space of *partial* functions is defined as follows:

Definition 4 *The* inverse discrete partial fuzzy transform *of* f^* *w.r.t.* $\mathbf{A}$, $\mathbf{B}$ *and* D *is a function* $f^*_{F,m,n}\colon X \to \mathbb{R}_*$ *such that*

$$f^*_{F,m,n}(x,y) = \sum_{k=1}^{m}{}^{\mathrm{B}} \sum_{l=1}^{n}{}^{\mathrm{B}} (F^*_{k,l} \cdot^{\mathrm{K}} (A_k(x)\cdot B_l(y))) \quad (8)$$

for all $(x,y) \in X$.

Convention 5 *We denote a partial fuzzy transform briefly by FT*.*

This definition of FT* that operates on $*$-extended reals allows us to fill in "small" gaps on X between the given data D by means of real values given by the inverse FT* while "big" gaps are filled only on the edges and remain undefined otherwise. A meaning of "small" and "big" gaps is captured by the width of supports of fuzzy sets in $\mathbf{A}$ and $\mathbf{B}$.

V. Image Reconstruction using Partial Fuzzy Transform

In this contribution IR problem does not mean a fusion of several images, but a problem of reconstruction of undefined, missing or corrupted part of a single input image. Methods for solving this problem are often called image inpainting (or image interpolation) methods. There are several approaches to the problem of IR [12]. The F-transform based method falls to the class of the interpolation techniques. For an overview and a comparison of interpolation techniques, we refer to [13].

In this section, we will provide IR algorithms based on FT* which fill in U/M regions of the input image I. We will assume that information about these regions (denoted by ω) is available in the form of the input mask S having value $*$ for U/M data and 1 otherwise. A goal of IR algorithms is to replace ω by values gained through computation on undamaged regions of an input image. Here, we use FT* to do so.

Remark that inverse FT and FT* transform a matrix of components to the space of 2-D continuous functions having values in $\mathbb{R}$. Therefore, we round the received real values of the inverse FT* to the closest natural number from the interval $[0,255]$ and $*$ remains untouched.

Recall that gray-scale digital images are represented as functions on $X \subset \mathbb{N}^2$ with values in $[0,255] \subset \mathbb{N}$, i.e., as mappings $I : X \mapsto [0,255]$, where $X = [1,M] \times [1,N]$ and $[1,M], [1,N]$ are closed intervals on $\mathbb{N}$.

In the following, let us consider a non-empty set $\omega \subset X$ and parameters $s, h \in \mathbb{N}$.

A. A Simple IR Algorithm Based on Partial FT

Let us briefly sketch steps of the simple IR algorithm based on FT* with the inputs I, ω, $h > 1$ and a generating function A:

- Rewrite values of I on ω by $*$ and denote the result by I^*.
- Compute m, n from h and generate h-uniform partitions $\mathbf{A}$ and $\mathbf{B}$ using A.
- Compute the direct and inverse FT* of I^* w.r.t. $\mathbf{A}$, $\mathbf{B}$ and D.
- Reconstruct a part of ω in I^* using the inverse FT*.

Inputs: Image I, damaged areas ω, width $h > 1$, generator A.

1) Set $I^*(x,y) = \begin{cases} *, & \forall (x,y) \in \omega; \\ I(x,y), & \text{otherwise.} \end{cases}$
2) Set $D = X \setminus \omega$.
3) Compute m, n from h and generate h-uniform partitions $\mathbf{A}$ and $\mathbf{B}$ using A.
4) Compute the direct FT* of I^* w.r.t. $\mathbf{A}$, $\mathbf{B}$ and D, i.e., $F^*_{m,n}[I^*]$ by (6).
5) Compute the inverse FT* of I^* w.r.t. $\mathbf{A}$, $\mathbf{B}$ and D, i.e., $I^*_{F,m,n}$ by (8).
6) Set $I^*(x,y) = I^*_{F,m,n}(x,y)$, for all $(x,y) \in \omega$.
7) Rewrite all $*$ by 0 in I^*.

Output: Image I^*.

B. An Iterative IR Algorithm Based on Partial FT

In this algorithm, we repeat the simple IR algorithm described above with an increasing width h (determined by the step s) until the whole ω is reconstructed. A detailed description follows.

Inputs: Image I, damaged areas ω, width $h > 1$, generator A, step $s > 0$.

1) Set $I^*(x,y) = \begin{cases} *, & \forall (x,y) \in \omega; \\ I(x,y), & \text{otherwise.} \end{cases}$
2) Set $D = X \setminus \omega$.
3) Compute m, n from h and generate h-uniform partitions $\mathbf{A}$ and $\mathbf{B}$ using A.
4) Compute the direct FT* of I^* w.r.t. $\mathbf{A}$, $\mathbf{B}$ and D, i.e., $F^*_{m,n}[I^*]$ by (6).
5) Compute the inverse FT* of I^* w.r.t. $\mathbf{A}$, $\mathbf{B}$ and D, i.e., $I^*_{F,m,n}$ by (8).
6) Set $I^*(x,y) = I^*_{F,m,n}(x,y)$, for all $(x,y) \in \omega$.
7) Set $h = h + s$ and $\omega = \{(x,y) \in I^* | I^*(x,y) = *\}$.
8) If $\omega \neq \emptyset$ then go to 2) else output I^*.

Output: Image I^*.

Remark 6 *In the case of RGB (where* $I\colon X \mapsto [0,255]^3$*) or another image representation model, we run the selected algorithm in each channel separately with the same settings of the input parameters.*

Example 7 *Consider* $A(x) = 1-|x|$*. We apply the simple and iterative IR algorithms based on FT* to an image with small, as well as large U/M areas visualized in Figure 1. Figures 2 and 3 show an effect of the simple algorithm with* $h = 2$ *and* $h = 3$*, respectively. A suitable choice of* h *(the parameter determining fuzzy partitions) leads to an infilling of U/M areas*

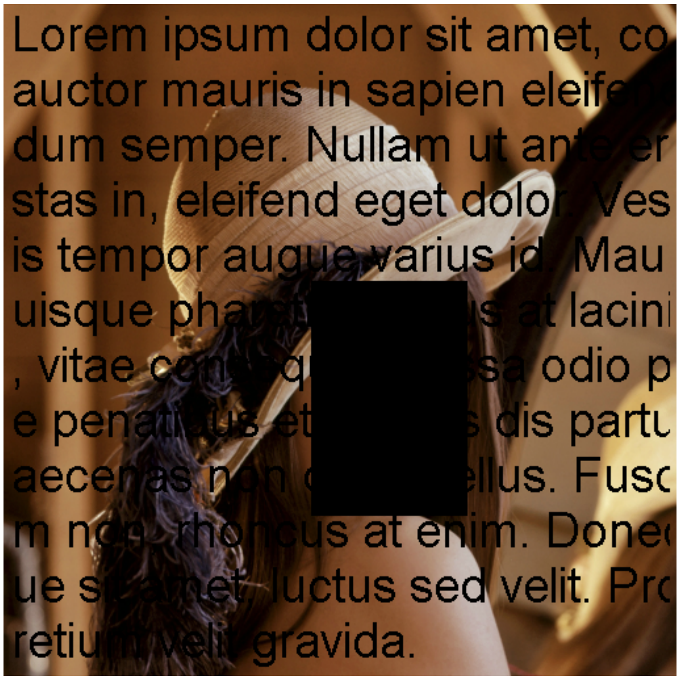

Figure 1. An original image with U/M areas.

Figure 2. Application of the simple IR algorithm based on FT* to the image from Figure 1 with $h = 2$.

of a maximal vertical and horizontal width smaller than h*. Finally, an output of the iterative IR algorithm based on FT* with* $h = 2$ *and* $s = 1$ *is depicted in Figure 4. In this particular example, both algorithms produce fine reconstructions of small U/M areas. In the case of the big U/M area in Figure 1, which covers Lena's face, some context based IR algorithm would be more applicable.*

Figure 3. Application of the simple IR algorithm based on FT* to the image from Figure 1 with $h = 3$.

Figure 4. Application of the iterative IR algorithm based on FT* to the image from Figure 4.

 ISBN: 978-1-61208-709-2

Figure 5. MSRA (due to the description in [4]) applied to Figure 1.

(a) IMSRA applied to Figure 4. (b) Difference between Figure 4 and IMSRA.

Figure 6. An example of an output of the Implementation of MSRA (IMSRA) provided in [5].

VI. CONCLUSIONS

The most important features of FT* is a significant simplification and correction of MSRA from [4] (and also other later sources). Indeed, experimental results of MSRA are almost identical with outputs of our iterative IR algorithm based on FT* (see Figure 6). Observe that FT* handles automatically U/M parts and they do not need to be explicitly encoded in (6) and (8). Moreover, this particular application shows the usefulness of the introduced formalism that deals with one error code $*$ for an arbitrary exception and various extensions of operations to $*$. Hence, we can choose a suitable extension of the user operation due to our requirements on the behaviour of $*$ and consequently, we do not need to take care of any exception because it is correctly handled using extended operations. In our opinion, this approach may be useful also in other research domains and problem-solving techniques where U/M data are present.

ACKNOWLEDGMENT

The work is supported by projects 16–19170S of GA ČR and LQ1602 of MŠMT ČR (NPU II).

REFERENCES

[1] L. Běhounek and M. Daňková, “Extending aggregation functions for undefined inputs,” in International Symposium on Aggregation and Structures, 2018, pp. 15–16.

[2] L. Běhounek and V. Novák, “Towards fuzzy partial logic,” in Proceedings of the IEEE 45th International Symposium on Multiple-Valued Logics (ISMVL 2015), 2015, pp. 139–144.

[3] I. Perfilieva, “Fuzzy transforms: Theory and applications,” Fuzzy Sets and Systems, vol. 157, 2006, pp. 992–1023.

[4] I. Perfilieva and P. Vlašánek, “Image reconstruction by means of F-transform,” Knowledge-Based Systems, vol. 70, 2014, pp. 55–63.

[5] P. Vlašánek, “Fuzzy image processing tutorials,” https://docs.opencv.org/4.0.1/d7/d36/tutorial_fuzzy.html, December 2018.

[6] P. Hurtik and P. Števuliáková, “Pattern matching: overview, benchmark and comparison with f-transform general matching algorithm,” Soft Computing, vol. 21, no. 13, 2017, pp. 3525–3536.

[7] F. Di Martino and S. Sessa, “Complete image fusion method based on fuzzy transforms,” Soft Computing, 2017, pp. 1–11.

[8] C. Reshmalakshmi, M. Sasikumar, and G. Shiny, “Fuzzy transform for low-contrast image enhancement,” International Journal of Applied Engineering Research, vol. 13, no. 11, 2018, pp. 9103–9108.

[9] I. Perfilieva, M. Daňková, and B. Barnabas, “Towards a higher degree f-transform,” Fuzzy Sets and Systems, vol. 180, no. 1, 2011, pp. 3–19.

[10] I. Perfilieva, P. Hodáková, and P. Hurtík, “Differentiation by the f-transform and application to edge detection,” Fuzzy Sets and Systems, vol. 288, 2016, pp. 96–114.

[11] P. Vlašánek and I. Perfilieva, “Patch based inpainting inspired by the f1-transform,” International Journal of Hybrid Intelligent Systems, vol. 13, no. 1, 2016, pp. 39–48.

[12] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH ’00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[13] S. Fadnavis, “Image interpolation techniques in digital image processing: An overview,” International Journal of Engineering Research and Applications, vol. 4, 2014, pp. 70–73.