IRAFM

UNIVERSITAS OSTRAVIENSIS

University of Ostrava

Institute for Research and Applications of Fuzzy Modeling

Edge detection using F-transform

Martina Daňková, Petra Hodáková, Irina Perfilieva, Marek Vajgl

Research report No. 157

2011

Submitted/to appear:

Preliminary version of the extended abstract aimed for ISDA 2011

Supported by:

Supported by projects 1M0572 of MŠMT ČR (DAR) and SGS7/PRF/2011 (Soft Computing Methods in Particular problems of Image Processing)

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic

tel.: +420-59-7091401 fax: +420-59-6120478
e-mail: e-mail martina.dankova@osu.cz

Edge detection using F-transform

Martina Daňková, Petra Hodáková, Irina Perfilieva, Marek Vajgl
Institute for Research and Applications of Fuzzy Modeling
University of Ostrava
Ostrava, Czech Republic
Email: martina.dankova@osu.cz

***Abstract*—This contribution introduces the technique of F-transform as a tool for handling the problem of edge detection. An explanation of the problem together with a detailed description of the F-transform technique will be given. Consequently, an algorithm for solving the problem will be provided together with a study of its behavior w.r.t. various parameter setting as well as w.r.t. various existing methods.**

***Keywords*-F-transform; Image processing; Edge detection;**

I. Introduction

The problem of edge detection is one of the most attractive problems for the image processing community. Mainly due to various important applications in the practical sphere (microscopy, medicine, safety management etc.) and also a wide spectrum of methods use the edge detection as a preprocessing technique. The development in this field has become immense in the last two decades. Techniques such as Wavelet transform, neural networks and also statistical or soft computing techniques has been employed to improve the original approaches represented by works published in [1], [2], [3] or to introduce a completely new approaches to detect edges (see e.g. [4], applying Wavelets [5], [6], statistical methods and optimization [7], [8], soft computing [9], [10], [11]).

Generally, the edge detection problem is one of the problems that are intuitively easy to describe and understand but, it is hard to formalize mathematically since there is no explicit definition of the term "edge". Existing algorithms differ mainly in the mathematical characterization of this term and consequently, by methods that search for pixels in the input image matching these characteristics. Symbolically, if u is an input image (considered as a function of two variables) then the edge detection can be viewed as its transformation to a many-valued (fuzzy) set U that is visualized as image. The membership degrees of U represent property of belonging to an edge, i.e. (x,y) is an edge pixel to the degree $U(x,y)$.

The main purpose of this contribution is to introduce the F-transform technique as a new method for the edge detection problem. The original motivation for F-transform came from fuzzy modeling [12], [13]. It performs a transformation of an original universe of functions into a universe of their "skeleton models" (vectors of F-transform components) where further computation is easier (see e.g., an application to the initial value problem with fuzzy initial condition [14]). In this respect, F-transform can be as useful in applications as traditional transforms (see applications to image compression [15], image fusion [16], [17], time series processing [18], etc.). Moreover, sometimes F-transform can be more efficient than its counterparts.

This contribution is organized as follows: Section II introduces the F-transform technique and gives an overview of its properties; Section III is devoted to a theoretical explanation of an efficiency of F-transform in the problem of edge detection and moreover it goes into the details of the algorithm; and in Section IV, some particular images will be analyzed using F-transform based algorithm and moreover, a comparison with some existing methods will be provided for these images. Finally, conclusions, comments and some future trends in our research will be drawn in Section V.

II. F-transform.

To find edges, we propose a new algorithm which is based on the technique of F-transform (a short name of fuzzy transform). Before going into the details of edge detection algorithm, let us give a general characterization and necessary details of the used technique. We assume that the reader is familiar with the notion of *fuzzy set* and a way how is it represented.

Below, we explain F-transform in more details and adapt our explanation to the purpose of this chapter (we refer to [12] for a complete description). The explanation will be given on the example of a discrete function that corresponds to the image u.

Let u be represented by the 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. If $(i,j) \in P$ is a pixel then the $u(i,j)$ represents its intensity range. We extend the domain P of u to the Cartesian product of intervals $[1,N] \times [1,M]$ and assume that u is a partial function on $[1,N] \times [1,M]$.

At first, we will introduce the notion of a *fuzzy partition*. It will be defined for the interval $[1,N]$ and then extend to $[1,N] \times [1,M]$.

Let $[1,N] = \{x \mid 1 \le x \le N\}$ be an interval on the real line $\mathbb{R}$, $n \ge 2$, a number of fuzzy sets in a fuzzy partition of $[1,N]$, and $h = \frac{N-1}{n-1}$ a distance between nodes $x_1,\dots,x_n \in$

$[1,N]$, where $x_1 = 1$, $x_k = x_1 + (k-1)h$, $k = 1,\dots,n$. Fuzzy sets $A_1,\dots,A_n : [1,N] \to [0,1]$ establish an *h-uniform fuzzy partition* of $[1,N]$ if the following requirements are fulfilled:

1) for every $k = 1,\dots,n$, $A_k(x) = 0$ if $x \in [1,N] \setminus [x_{k-1}, x_{k+1}]$ where $x_0 = x_1$, $x_{n+1} = x_N$;
2) for every $k = 1,\dots,n$, A_k is continuous on $[x_{k-1}, x_{k+1}]$ where $x_0 = x_1$, $x_{n+1} = x_N$;
3) for every $i = 1,\dots,N$, $\sum_{k=1}^{n} A_k(i) = 1$;
4) for every $k = 1,\dots,n$, $\sum_{i=1}^{N} A_k(i) > 0$;
5) for every $k = 2,\dots,n-1$, A_k is symmetrical with respect to the line $x = x_k$.

It is easy to see that if the fuzzy partition $A_1,\dots,A_n$ is h-uniform, then there exists an even function

$$A : [-h,h] \to [0,1]$$

such that for all $k = 2,\dots,n-1$,

$$A_k(x) = A(x - x_k) = A(x_k - x),\ x \in [x_{k-1}, x_{k+1}].$$

We call A a *generating function* of an h-uniform fuzzy partition.

The example of a triangular-shaped uniform fuzzy partition $A_1,\dots,A_n$, $n \geq 2$, of the interval $[1,N]$ is given below.

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

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

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

Note that the shape (triangular, sinusoidal, etc.) of a basic function in a fuzzy partition is not predetermined and can be chosen according to extra requirements.

If fuzzy sets $A_1,\dots,A_n$ establish a fuzzy partition of $[1,N]$ and $B_1,\dots,B_m$ do the same for $[1,M]$ then the Cartesian product $\{A_1,\dots,A_n\} \times \{B_1,\dots,B_m\}$ of these fuzzy partitions is the set of all fuzzy sets $A_k \times B_l$, $k = 1,\dots,n$, $l = 1,\dots,m$. The membership function $A_k \times B_l : [1,N] \times [1,M] \to [0,1]$ is equal to the product $A_k \cdot B_l$ of the respective membership functions. Fuzzy sets $A_k \times B_l$, $k = 1,\dots,n$, $l = 1,\dots,m$ establish a fuzzy partition of the Cartesian product $[1,N] \times [1,M]$.

Now we can introduce the F-transform of $u : P \to \mathbb{R}$ with respect to the chosen partition of $[1,N] \times [1,M]$.

Let $u : P \to \mathbb{R}$ and fuzzy sets $A_k \times B_l$, $k = 1,\dots,n$, $l = 1,\dots,m$ establish a fuzzy partition of $[1,N] \times [1,M]$. The (direct) *F-transform* of u (with respect to the chosen partition) is an image of the mapping $F[u] : \{A_1,\dots,A_n\} \times \{B_1,\dots,B_m\} \to \mathbb{R}$ defined by

$$F[u](A_k \times B_l) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} u(i,j) A_k(i) B_l(j)}{\sum_{i=1}^{N} \sum_{j=1}^{M} A_k(i) B_l(j)}, \quad (1)$$

where $k = 1,\dots,n, l = 1,\dots,m$. The value $F[u](A_k \times B_l)$ is called an *F-transform component* of u and is denoted by $F[u]_{kl}$. Components $F[u]_{kl}$ can be arranged into the matrix representation or into the vector representation as follows:

$$(F[u]_{11},\dots,F[u]_{1m},\dots,F[u]_{n1},\dots,F[u]_{nm}). \quad (2)$$

The *inverse F-transform* of u is a function on P which is represented by the following inversion formula where $i = 1,\dots,N, j = 1,\dots,M$:

$$u_{nm}(i,j) = \sum_{k=1}^{n} \sum_{l=1}^{m} F[u]_{kl} A_k(i) B_l(j). \quad (3)$$

It can be shown that the inverse F-transform u_{nm} approximates the original function u on the domain P. The proof can be found in [12], [13].

Example 1 *Let a discrete real function $u = u(x,y)$ be defined on the $N \times M$ array of pixels $P = \{(i,j) \mid i = 1,\dots,N, j = 1,\dots,M\}$ so that $u : P \to \mathbb{R}$. We will characterize F-transforms of u for two extreme fuzzy partitions.*

The largest partition of $[1,N] \times [1,M]$ contains only one fuzzy set $A_1 \times B_1$ such that for all $(x,y) \in [1,N] \times [1,M]$, $A_1 \times B_1(x,y) = 1$. The respective F-transform component $F[u](A_1 \times B_1)$ and the respective inverse F-transform u_{nm} are as follows:

$$F[u](A_1 \times B_1) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} u(i,j)}{NM},$$
$$u_{nm}(i,j) = F[u](A_1 \times B_1), \quad i = 1,\dots,N, j = 1,\dots,M.$$

It is easy to see that $F[u](A_1 \times B_1)$ is the arithmetic mean of u.

The finest partition of $[1,N] \times [1,M]$ is established by $N \times M$ fuzzy sets $A_k \times B_l$ such that for all $k = 1,\dots,N$, $l = 1,\dots,M$, $A_k \times B_l(x_k, y_l) = 1$ and for all $r = 1,\dots,N$, $s = 1,\dots,M$, $(k,l) \neq (r,s)$, $A_k \times B_l(x_r, y_s) = 0$. The respective F-transform components $F[u](A_k \times B_l)$, $k = 1,\dots,N$, $l = 1,\dots,M$, and the respective inverse F-transform u_{nm} are as follows:

$$F[u](A_k \times B_l) = u(k,l),$$
$$u_{nm}(i,j) = u(i,j), \quad i = 1,\dots,N, j = 1,\dots,M.$$

It is easy to see that $u_{nm} = u$.

The following statement (for the proof see [19]) expresses a useful property of the F-transform components. It can be characterized by saying that each component $F[u]_{kl}$ is a local mean value of u over a support set of the respective fuzzy set $A_k \times B_l$.

P1. The kl-th component $F[u]_{kl}$ $(k = 1,\dots,n, l = 1,\dots,m)$ minimizes the function

$$\Phi(y) = \sum_{j=1}^{l} (u(i,j) - y)^2 A_k(i) B_l(j)$$

The next statement [20] is useful for a verification that the proposed edge detection technique works correctly. The statement describes a representation of the discrete Fourier transform of the F-transform components.

P2. Let $\mathbf{Z}_l = \{0, 1, \ldots, l-1\}$ and $\hat{f}$ be the Fourier transform of a function $f : \mathbf{Z}_l \to \mathbb{R}$. Let $n \geq 3$ and $A_1, \ldots, A_{n-1}$ be a triangular-shaped h-uniform fuzzy partition $[a, b]$ where $h = \frac{b-a}{n}$. Let $F : \mathbf{Z}_l \to R$ be the discrete function given by

$$F(t) = \sum_{j=0}^{l-1} A(t-j) f(j);\ t = 0, \ldots, l-1,$$

which contains the F-transform components of f among its values.

Then the Fourier transform of F is given by

$$\begin{aligned} \hat{F}(0) &= \hat{f}(0); \\ \hat{F}(k) &\approx \frac{mn^2}{2\pi^2 k^2} \exp(-2\pi i k/n)(1 - \cos\frac{2\pi k}{n}) \cdot \hat{f}(k); \\ & k = 1, \ldots, l-1, \end{aligned} \quad (4)$$

where m is a fixed parameter.

By **P2.**, the influence of the Fourier coefficient $\hat{f}(k)$ in the representation (4) is weakened by the factor $\frac{1}{k^2}$. In other words, every F-transform component works as a low-pass filter of an original function.

Having in mind the two above mentioned facts, namely: the inverse F-transform approximates an original function, and the F-transform components are low-pass filters, we come to the conclusion that the difference between an original function and its inverse F-transform works as a high-pass filter of the former. Therefore, the mentioned above difference can be used for the edge detection problem.

III. F-transform based Algorithm for Edge Detection

Informally, we understand edges in a picture as visually important places that can be distinguished mainly by a significant change of intensity. Further on, we aim to emphasize these places by assigning degrees of membership to a particular edge. For an image $u : P \to \mathbb{R}$, the edge elements can be collected in the set as follows:

$$\begin{aligned} B = \{(x_i, y_j) \in P | \ \exists (x, y) \in P\ (d((x_i, y_j), (x, y)) \leq \delta) \\ \text{and } |u(x_i, y_j) - u(x, y)| \geq \varepsilon\}, \end{aligned} \quad (5)$$

where d is a distance, δ determines a size of the "important" neighborhood or in another words, a thickness of the edge, and ε specifies a "significance" of the intensity change.

For our approach, it is important that the F-transform components are local weighted arithmetic means with weights taken as membership degrees of the respective fuzzy set in the fuzzy partition, see the property **P1**. An important part of the preliminary analysis of the input image is a choice of the parameters of F-transform that are in our case the numbers n, m of fuzzy sets in the fuzzy partition of $[1, N] \times [1, M]$. **P2** provides a relationship between the Fourier transform of the F-transform components and a frequency spectrum of the input image, which gives a direction for estimation of n, m so that the particular frequency is captured by the components.

Let us explain a way of characterization of edge elements using F-transform components. The numbers n, m of fuzzy sets in a respective fuzzy partition determine its robustness. This relates to a size of a chosen neighborhood which in (5) was characterized by the inequality $d((x_i, y_j), (x, y)) \leq \delta$. In our approach, the above inequality is generalized by $A_i(x) \cdot B_j(y)$, where δ relates to lengths of supports of A_i, B_j. Both numbers n, m and shapes of fuzzy sets A_i, B_j in a fuzzy partition determine components $F[u]_{i,j}$ of the F-transform of u. The difference $|u(x, y) - F[u]_{i,j}|$ approximates the respective difference $|u(x_i, y_j) - u(x, y)|$ in (5). Thus we come to the following generalization of (5):

$$\begin{aligned} B = \{[x, y] \in P | \ (\exists i, j)(x \in A_i) \text{ and } (y \in B_j) \\ \text{and } |u(x, y) - F[u]_{i,j}| \succcurlyeq \varepsilon\}, \end{aligned} \quad (6)$$

where B is now fuzzy set with the membership function in $[0, 1]$, $\exists$ is computed as addition, "and" is product, $(x \in A_i)$ is interpreted as $A_i(x)$ (the same for B_j, B), and finally $|u(x, y) - F[u]_{i,j}| \succcurlyeq \varepsilon$ is the rescaled $|u(x, y) - F[u]_{i,j}|$ onto $[0, 1]$. Even thought this interpretation is not easy to be understood the above fuzzy class definition (6) of B is very natural and intuitive because it uses the usual symbols of the classical set theory.

Now let us summarize the main steps of the edge detection algorithm that uses F-transform (**FTransform-EDA**):

Input: Image u, the numbers of fuzzy sets in the fuzzy partition n, m

1) Compute $F[u]$ – the direct F-transform of u by (1);
2) Compute $u_{n,m}$ – the inverse F-transform using the components $F[u]$ by (3);
3) Compute the error function $e(x) = |u(x) - u_{n,m}(x)|$ for all $x \in P$.
4) Rescale and round the values of e from $[0, \max_{x \in P} e(x)]$ to the integers in $[0, 255]$, which results in the new image e_r.

Output: Image e_r.

IV. Experimental Results and Comparisons

As it was mentioned in the previous section, the input arguments n, m determine the thickness of the edge. Hence, the bigger are n, m the thinner are edges. It follows that the edges of all objects in focus can be determined from the change of intensities of the pixels of the closest neighborhoods, i.e. setting n, m such that each A_i, B_j covers min. 4 pixels (i.e. $h \geq 2$ pixels) see Figure 3.

Naturally, problem comes with blurred images, with a various depth of the focus or objects that are in (partial) shadow. Let us present here the second case and show how the setting of the edge thickness (i.e. the setting of

(a) Original image.

(b) e_r with 4 pixels covered

(c) e_r with 10 pixels covered

(d) e_r with 100 pixels covered

Figure 1. Examples of various setting of the input parameters of **FTransform-EDA**

n,m) affected the result. Figure 2(a) shows an exemplary image where the object of an interest is blurred. Obviously, edge elements of this object do not correspond with the characterization of edge elements for an object in focus, i.e. the intensity change in a small neighborhood of the edge element is not significant. Hence, the edges of the blurred objects cannot be captured using small values n,m, see Figure 2(b). A different situation comes with higher values of n,m (see Figure 2(c)): in this case, lower intensity changes are captured by the components of F-transform, edges are thicker and the edge of the blurred object is specified in a negative way which means its edge elements have the degrees of memberships close to zero.

Now, let us make a comparison with standard methods for the edge detection problem. Due to the space limitation, we will provide a comparison only for one image (Figure 3(a)) with a complex scene comprising from a heterogenous background and some objects in a foreground. The results of **FTransform-EDA** are on Figure 3(b). Outputs of the standard methods are on Figure 4. To compare our results with the binary images (outputs of the chosen standard methods) we have to do a thresholding, see Figure 3(c).

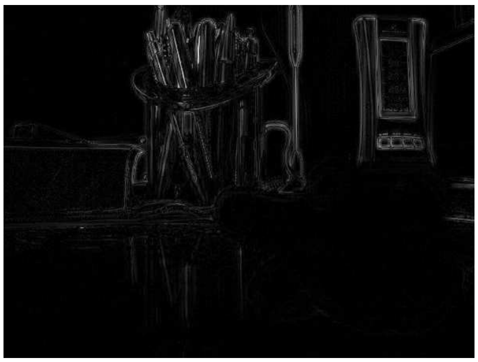
As standard methods we take gradient-based algorithms - the Prewitt algorithm, the Sobel algorithm and the multi-stage Canny algorithm. We recall [?] that these methods deal with the *edge* characterized by properties of the picture fragment which is given by a magnitude of change of intensity and a direction of the greatest increase of the picture values.

The Prewitt and Sobel algorithms have a major drawback of being very sensitive to noise. The size of the kernel filter and coefficients are fixed and cannot be adapted to a given image. These methods determine thicker edges and then they give an imprecise localization. The Canny algorithm is more flexible and depends on the adjustable parameters. It can be described as an "optimal" edge detection algorithm. But it is computationally more expensive compared to the Sobel and Prewitt algorithms. In comparison with those methods, the **FTransform-EDA** behaves acceptably well. According to a visual comparison, we can say that it outperforms the Sobel and Prewitt algorithms but for this particular image, the Canny algorithm wins.

V. Conclusion

This contribution has focused on the application of F-transform to the problem of the edge detection. We proposed the algorithm that uses an error function created on the basis of F-transform technique. A detailed description of this algorithm has been given together with a motivation, explanation and justification of its suitability in the given problem. Finally, various examples were tested and we showed that the proposed approach can be successfully applied to the images with a complex scene as well as to the blurred images. In a comparison with standard methods

(a) Original image.



(b) e_r with 10 pixels covered

(c) e_r with 100 pixels covered

Figure 2. **FTransform-EDA** applied to image with objects in various focus.

(a) Original image.

(b) e_r with 4 pixels covered

(c) thresholded Figure 3(b)

Figure 3. **FTransform-EDA** applied to image with a complex scene.

we have demonstrated that the edge detection algorithm based on F-transform performs acceptably well. The deeper analysis, improvements related to the automatization of the thresholding as well as the input parameters together with a comparison study is a matter of the future research.

Acknowledgment

We gratefully acknowledge support of the projects 1M0572 of MŠMT ČR (DAR) and SGS7/PRF/2011 (Soft Computing Methods in Particular problems of Image Processing).

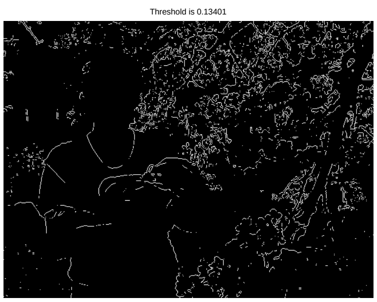
References

[1] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679 –698, nov. 1986.

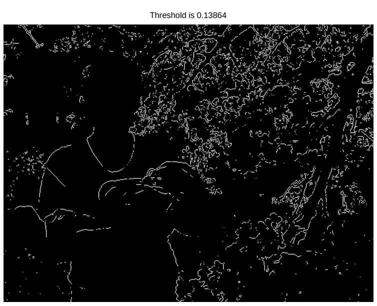
[2] R. M. Haralick, “Digital step edges from zero crossing of second directional derivatives,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, no. 1, pp. 58 –68, jan. 1984.

[3] W. E. L. Grimson and E. C. Hildreth, “Comments on “digital step edges from zero crossings of second directional derivatives”,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-7, no. 1, pp. 121 –127, jan. 1985.

(a) Canny algorithm.



(b) Prewitt algorithm



(c) Sobel algorithm

Figure 4. Standard methods applied to Figure 3(a).

[4] F. Pellegrino, W. Vanzella, and V. Torre, "Edge detection revisited," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 3, pp. 1500 –1518, june 2004.

[5] W. Jiang, K.-M. Lam, and T.-Z. Shen, "Efficient edge detection using simplified gabor wavelets," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 4, pp. 1036 –1047, aug. 2009.

[6] M. Tello Alonso, C. Lopez-Martinez, J. Mallorqui, and P. Salembier, "Edge enhancement algorithm based on the wavelet transform for automatic edge detection in sar images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 1, pp. 222 –235, jan. 2011.

[7] M. Law and A. Chung, "Weighted local variance-based edge detection and its application to vascular segmentation in magnetic resonance angiography," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 9, pp. 1224 –1241, sept. 2007.

[8] H. Moon, R. Chellappa, and A. Rosenfeld, "Optimal edge-based shape detection," *Image Processing, IEEE Transactions on*, vol. 11, no. 11, pp. 1209 – 1227, nov 2002.

[9] J. Bezdek, R. Chandrasekhar, and Y. Attikouzel, "A geometric approach to edge detection," *Fuzzy Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 52 –75, feb 1998.

[10] T. Law, H. Itoh, and H. Seki, "Image filtering, edge detection, and edge tracing using fuzzy reasoning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 5, pp. 481 –491, may 1996.

[11] J. Wu, Z. Yin, and Y. Xiong, "The fast multilevel fuzzy edge detection of blurry images," *Signal Processing Letters, IEEE*, vol. 14, no. 5, pp. 344 –347, may 2007.

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

[13] ——, "Fuzzy transforms: A challenge to conventional transforms," in *Advances in Images and Electron Physics*, P. W. Hawkes, Ed. San Diego: Elsevier Academic Press, 2007, vol. 147, pp. 137–196.

[14] I. Perfilieva, H. De Meyer, B. De Baets, and D. Plskova, "Cauchy problem with fuzzy initial condition and its approximate solution with the help of fuzzy transform," in *Proc. of WCCI 2008, IEEE Int. Conf. on Fuzzy Systems*, Hong Kong, 2008, pp. 2285–2290.

[15] I. Perfilieva, V. Pavliska, M. Vajgl, and B. De Baets, "Advanced image compression on the basis of fuzzy transforms," in *Proc. Conf. IPMU'2008*, Torremolinos (Malaga), Spain, 2008, pp. 1167–1174.

[16] M. Daňková and R. Valášek, "Full fuzzy transform and the problem of image fusion," *Journal of Electrical Engineering*, vol. 12, pp. 82–84, 2006.

[17] P. H. Irina Perfilieva, Martina Daňková and M. Vajgl, "F-transform based image fusion," in *Image Fusion*, O. Ukimura, Ed. InTech, 2011, pp. 3–22. [Online]. Available: http://www.intechopen.com/articles/show/title/f-transform-based-image-fusion

[18] I. Perfilieva, V. Novák, V. Pavliska, A. Dvořák, and M. Štěpnička, "Analysis and prediction of time series using fuzzy transform," in *Proc. of WCCI 2008, IEEE Int. Conf. on Neural Networks*, Hong Kong, 2008, pp. 3875–3879.

[19] I. Perfilieva and R. Valášek, "Fuzzy transforms in removing noise," in *Computational Intelligence, Theory and Applications*, B. Reusch, Ed. Heidelberg: Springer, 2005, pp. 225–234.

[20] I. Perfilieva and P. Hodáková, "Fuzzy and fourier transforms," in *Proc. of the LFA-EUSFLAT'2011*, France, 2011.