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

Image Composition Using F-transform

Marek Vajgl, Petr Hurtik, Irina Perfilieva, Petra Hodáková
Centre of Excellence IT4Innovations
Division of the University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
Ostrava, Czech Republic
Email: marek.vajgl@osu.cz, petr.hurtik@osu.cz, irina.perfilieva@osu.cz, petra.hodakova@osu.cz

***Abstract*—The contribution describes newly developed technique used to improve image quality by fusion of information from the multiple images into one resulting image containing better information than each of the input ones. The presented approach is based on the F-Transform, integral transform used to detect gradients, similarity and image fusion, and noise reduction.**

I. Introduction

Usually image obtained by camera is not in a perfect quality, especially in the case of low light conditions or in the case of images taken by mobile phones or web cameras. There exist several techniques to improve their quality. A lot of techniques are based on one image modification with standard algorithms (median filter for noise reduction, Laplace filter for image sharpening, brightness or contrast modification etc.), other approaches are based on the composition of several images. The set of the input images is usually represented by images capturing the same view with changed conditions as light or some hardware filter. The most known application of multiple image composition is called HDR (e.g. see [1], [2]).

The presented approach in this contribution is aimed on soft-computing method based on *Fuzzy transformation* (see [3])(also referred as *F-transform*) that consists of two steps: direct and inverse transform. The F-transform is based on a simple idea of covering a domain of a continuous function by finite number of fuzzy granules whereas the parts affecting every element are overlapped. The F-transform has been introduced by Irina Perfilieva and the main theoretical preliminaries were described in [3] and [4]. Later a lot of practical application of the F-transform were developed, especially image compression ([5], [6]) is a good example where a user can control strength of compression and quality by number of used components. By adjusting level of components the quality (distortion) of the image can be adjusted. Then F-transform based fusion ([9], [10]) can be used to pick up parts with best image information representation. Another interesting application of F-transform is an image reduction and interpolation [7]: the direct F-transform can reduce (shrink) image and the inverse F-transform can be used as interpolation method. The F-transform of higher degree [12] (especially first degree) can approximate original function even better and can be used for image gradient computation [11] and following edge detection.

This contribution demonstrates how the F-transform can be used in multiple image composition: F^1-transform is used for gradient detection in features extraction step; F^0-transform is used for image similarity measurement and final image fusion. By the composition of these three steps the set of images should be registered and then fused into one image with improved quality, especially with a noise reduction.

II. F-transform

A. Fuzzy partition with Ruspini condition

The *fuzzy partition with the Ruspini condition* (1) (simply, *Ruspini partition*) was introduced in [3]. This condition implies normality of the respective fuzzy partition, i.e., the "partition-of-unity". It then leads to a simplified version of the inverse F-transform. In later publications [12], the Ruspini condition was weakened to obtain an additional degree of freedom and a better approximation by the inverse F-transform.

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*. The shape of the basic functions is not predetermined and therefore, it can be chosen according to additional requirements (e.g., smoothness). However, usage of triangular basic functions is the most common in an image processing.

B. Generalized fuzzy partitions

A *generalized fuzzy partition* appeared in [12] in connection with the notion of the higher-degree F-transform. Its even weaker version was implicitly introduced in [5] for the purpose of meeting the requirements of image compression. We summarize both these notions and propose the following definition. Let $[a, b]$ be an interval on the real line $\mathbb{R}$, $n > 2$, and let $x_1, \ldots, x_n$ be nodes such that $a \leq x_1 < \ldots < x_n \leq b$. Let $[a, b]$ be covered by the intervals $[x_k - h'_k, x_k + h''_k] \subseteq [a, b]$,

$k = 1, \dots, n$, such that their left and right margins $h'_k, h''_k \geq 0$ fulfill $h'_k + h''_k > 0$.

We say that fuzzy sets $A_1, \dots, A_n : [a, b] \to [0, 1]$ constitute a *generalized fuzzy partition* of $[a, b]$ (with nodes $x_1, \dots, x_n$ and margins h'_k, h''_k, $k = 1, \dots, n$), if for every $k = 1, \dots, n$, the following three conditions are fulfilled:

1) (*locality*) — $A_k(x) > 0$ if $x \in (x_k - h'_k, x_k + h''_k)$, and $A_k(x) = 0$ if $x \in [a, b] \setminus (x_k - h'_k, x_k + h''_k)$;
2) (*continuity*) — A_k is continuous on $[x_k - h'_k, x_k + h''_k]$;
3) (*covering*) — for $x \in [a, b]$, $\sum_{k=1}^{n} A_k(x) > 0$.

An (h, h')-*uniform* generalized fuzzy partition of $[a, b]$ is defined for equidistant nodes $x_k = a + h(k-1)$, $k = 1, \dots, n$, where $h = (b - a)/(n - 1)$, $h' > h/2$ and two additional properties are satisfied:

4) $A_k(x) = A_{k-1}(x - h)$ 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 $x \in [x_k, x_{k+1}]$.
5) $h'_1 = h''_n = 0$, $h''_1 = h'_2 = \dots = h''_{n-1} = h'_n = h'$ and for all $k = 2, \dots, n - 1$ and all $x \in [0, h']$, $A_k(x_k - x) = A_k(x_k + x)$.

An (h, h')-uniform generalized fuzzy partition of $[a, b]$ can also be defined using the *generating function* $A_0 : [-1, 1] \to [0, 1]$, which is assumed to be *even*[1], continuous and positive everywhere except for on boundaries, where it vanishes. Then, basic functions A_k of an (h, h')-uniform generalized fuzzy partition are shifted copies of A_0 in the sense that

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

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

$$A_k(x) = \begin{cases} A_0\left(\frac{x - x_k}{h'}\right), & x \in [x_k - h', x_k + h'], \\ 0, & otherwise. \end{cases}, \tag{2}$$

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

C. F^0 transform

The direct and inverse F-transform as a function of two (and more) variables is a generalization of the case of one variable F-transform. We introduce the discrete version only due to its application in image processing. Let us refer to [4] for more details.

Suppose that the universe is a rectangle $[a, b] \times [c, d] \subseteq \mathbb{R} \times \mathbb{R}$ and that $x_1 < \dots < x_n$ are fixed nodes of $[a, b]$ and $y_1 < \dots < y_m$ are fixed nodes of $[c, d]$ such that $x_1 = a$, $x_n = b$, $y_1 = c$, $y_m = d$ and $n, m \geq 2$. Assume that $A_1, \dots, A_n$ are basic functions that form a generalized fuzzy partition of $[a, b]$ and $B_1, \dots, B_m$ are basic functions that form a generalized fuzzy partition of $[c, d]$. Then, the rectangle $[a, b] \times [c, d]$ is partitioned into fuzzy sets $A_k \times B_l$ with the membership functions $(A_k \times B_l)(x, y) = A_k(x) B_l(y)$, $k = 1, \dots, n$, $l = 1, \dots, m$.

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

In the discrete case, an original function f is assumed to be known only at points $(p_i, q_j) \in [a, b] \times [c, d]$, where $i = 1, \dots, N$ and $j = 1, \dots, M$. In this case, the (discrete) F-transform of f can be introduced in a manner analogous to the case of a function of one variable.

Let a function f be given at points $(p_i, q_j) \in [a, b] \times [c, d]$, for which $i = 1, \dots, N$ and $j = 1, \dots, M$, and $A_1, \dots, A_n$ and $B_1, \dots, B_m$, where $n < N$ and $m < M$, be basic functions that form generalized fuzzy partitions of $[a, b]$ and $[c, d]$ respectively. Suppose that sets P and Q of these points are sufficiently dense. We say that the $n \times m$-matrix of real numbers $\mathbf{F}[f] = (F_{kl})_{nm}$ is the discrete F-transform of f with respect to $A_1, \dots, A_n$ and $B_1, \dots, B_m$ if

$$F_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} f(p_i, q_j) A_k(p_i) B_l(q_j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k(p_i) B_l(q_j)} \tag{3}$$

holds for all $k = 1, \dots, n$, $l = 1, \dots, m$.

The inverse F-transform of a discrete function f of two variables is defined as follows. Let $A_1, \dots, A_n$ and $B_1, \dots, B_m$ be basic functions that form generalized fuzzy partitions of $[a, b]$ and $[c, d]$, respectively. Let function f be defined on the set of points $(p_i, q_j) \in P \times Q$ where $P = \{p_1, \dots, p_N\} \subseteq [a, b]$, $Q = \{q_1, \dots, q_M\} \subseteq [c, d]$ and both sets P and Q are sufficiently dense with respect to corresponding partitions. Moreover, let $\mathbf{F}[f] = (F_{kl})_{nm}$ be the discrete F-transform of f w.r.t. $A_1, \dots, A_n$ and $B_1, \dots, B_m$. Then, the function $\hat{f} : P \times Q \to \mathbb{R}$ represented by

$$\hat{f}(p_i, q_j) = \frac{\sum_{k=1}^{n} \sum_{l=1}^{m} F_{kl} A_k(p_i) B_l(q_j)}{\sum_{k=1}^{n} \sum_{l=1}^{m} A_k(p_i) B_l(q_j)} \tag{4}$$

is called the *inverse F-transform* of f.

D. F^1-transform

We can generalize the F-transform with constant components to the F^1-transform with linear components. The latter are orthogonal projections of an original function f onto a linear subspace of functions with the basis of polynomials $P_k^0 = 1, P_k^1 = (x - x_k)$. We say that the n-tuple

$$F^1[f] = [F_1^1, \dots, F_n^1] \tag{5}$$

is the F^1-*transform* of f w.r.t. $A_1, \dots, A_n$ where the $k-$th component F_k^1 is defined by

$$F_k^1 = c_{k,0} P_k^0 + c_{k,1} P_k^1, \ k = 1, \dots, n. \tag{6}$$

For the h-uniform fuzzy partition and the triangular-shaped basic functions we can compute the coefficients $c_{k,0}, c_{k,1}$ for each $k = 1, \dots, n$ as follows

$$c_{k,0} = \frac{1}{h} \sum_{i=1}^{N} f(p_i) A_k(p_i), \tag{7}$$

$$c_{k,1} = \frac{12}{h^3} \sum_{i=1}^{N} f(p_i)(p_i - x_k) A_k(p_i). \tag{8}$$

It can be shown that the coefficient $c_{k,0}$ is equal to the F-transform component F_k, $k = 1, \dots, n$. The next theorem shows the important property of the coefficient $c_{k,1}$ which will be useful for the proposed edge detection technique.

The theorem is formulated for the continuous version of the F^1-transform. Let $A_1, \ldots, A_n$, be an h-uniform partition of $[a, b]$, let functions f and $A_k,\ k = 1, \ldots, n$ be four times continuously differentiable on $[a, b]$, and let $F^1[f] = (c_{1,0} + c_{1,1}(x - x_1), \ldots, c_{n,0} + c_{n,1}(x - x_n))$ be the F^1-transform of f with respect to $A_1, \ldots, A_n$,. Then, for every $k = 1, \ldots, n$, the following estimation holds true:

$$c_{k,1} = f'(x_k) + O(h). \tag{9}$$

We refer to [12] for a proof of **Theorem II-D** and for a detailed description of the F^1-transform.

III. Image composition

This section describes application of the F-transform theory in image composition. The developed method is divided into three steps: A. Feature extraction, B. Feature matching, and C. Image fusion. An example of a result obtained by image composition algorithm on two input images can be seen at figure 5.

A. Feature extraction

Let us remark that there exist many algorithms for feature extraction; the most used are FAST, ORB or SIFT [13]. In this contribution, we are taking the problem of feature extraction as a procedure that selects small corner areas in the image. According to the accepted terminology, we call the latter *point features*. Point features extracted from a reference and sensed images should be detected on the similar places even if the sensed image is rotated, resized or has different intensity. We propose an original technique of point features detection using the first degree F-transform (F^1-transform) adopted from [11].

By **Theorem II-D**, coefficients $c_{k,1}$ of the F^1-transform give us a vector whose components approximate the first derivative of the original function at certain nodes. We use these coefficients as components of the inverse F-transform and we get the approximation of the first derivative of the original image function in each pixel.

Let triangular fuzzy sets $A_1, \ldots, A_n$ establish a fuzzy partition of $[1, N]$ and triangular $B_1, \ldots, B_m$ do the same for $[1, M]$. Let $x_1, \ldots, x_n \in [1, N]$, $h_x = x_{k+1} - x_k$, $k = 1, \ldots, n$ and $y_1, \ldots, y_m \in [1, M]$, $h_y = y_{l+1} - y_l$, $l = 1, \ldots, m$ be nodes on $[1, N]$, $[1, M]$ respectively. Then we can determine the approximation of the first derivative for each $(p_i, p_j) \in D$ in the horizontal direction

$$G_x(p_i, p_j) \approx \sum_{k=1}^{n} \sum_{l=1}^{m} c_{k,1}(y_l) A_k(p_i) B_l(p_j) \tag{10}$$

and in the vertical direction

$$G_y(p_i, p_j) \approx \sum_{k=1}^{n} \sum_{l=1}^{m} c_{l,1}(x_k) A_k(p_i) B_l(p_j) \tag{11}$$

as the inverse F-transform of the image function u where the coefficients $c_{k,1}(y_l)$, $c_{l,1}(x_k)$, $k = 1, \ldots, n$, $l = 1, \ldots, m$ are given by the F^1-transform

$$c_{k,1}(y_l) = \frac{12}{h_x^3} \sum_{i=1}^{N} f(p_i, y_l)(p_i - x_k) A_k(p_i), \tag{12}$$

$$c_{l,1}(x_k) = \frac{12}{h_y^3} \sum_{j=1}^{M} f(x_k, p_j)(p_j - y_l) B_l(p_j). \tag{13}$$

Then, the gradient magnitude G of an edge at point (p_i, p_j) is computed as

$$G(p_i, p_j) = \sqrt{G_x(p_i, p_j)^2 + G_y(p_i, p_j)^2} \tag{14}$$

and the gradient angle Θ is determined by

$$\Theta(p_i, p_j) = \arctan \frac{G_y(p_i, p_j)}{G_x(p_i, p_j)} \tag{15}$$

where for simplicity the gradient angle will be quantized by: $\Theta^Q : \Theta \to \{0, 45, 90, 135\}$.

We say that a corner is a set of neighboring pixels (we call them corner points) where at least three different quantized angles show up. A center of gravity of the corner is called a feature point. Many corner points can be found in an image. It may happen that corner points are close to each other. In this case we have to choose only one of them. We modify computer graphic *flood fill* algorithm to detect clusters of close corner points and then compute centers of gravity of each cluster. These centers constitute the set of point features.

B. Feature matching

In this step, a correspondence between the point features detected in the reference and sensed images is established. The reference image is the first of the image set, all followed are sensed and centered according to the referenced one. As a main technique (among various similarity measures or spatial relationships) we propose to measure similarity by a (inverse) distance between F-transform components of various levels.

In more details, the lowest (first) level is comprised by the F-transform components of image f and corresponds to the discretization given by the respective fuzzy partition of the domain. This first level $F^{(1)}[f]$ is given by the F-transform of f so that

$$F^{(1)}[f] = F[f] = (F_{11}, ..., F_{nm}). \tag{16}$$

The vector of the F-transform components $(F_{11}, ..., F_{nm})$ is a linear representation of a respective matrix of components. This first level serves as a new image for the F-transform components of the second level and so on. For a higher level ℓ we propose the following recursive formula:

$$F^{(\ell)}[f] = F[F^{(\ell-1)}] = (F_{11}^{(\ell-1)}, ..., F_{n_{(\ell-1)} m_{(\ell-1)}}^{(\ell-1)}). \tag{17}$$

The top (last) level $F^{(t)}[f]$ consists of only one final component F^{fin}.

The F-transform based similarity S of two image functions $f, g \in \mathcal{I}$ is proposed to be as follows:

$$S(f, g) = 1 - |F^{fin} - G^{fin}| \cdot \frac{\sum_{k=1}^{n} \sum_{l=1}^{m} |F_{kl} - G_{kl}|}{nm} \tag{18}$$

where F^{fin}, G^{fin} are the top F-transform components of f and g, and F_{kl}, G_{kl}, $k = 1, \ldots, n$, $l = 1, \ldots, m$ are the first level F-transform components of f and g, respectively. The justification that S is a similarity measure with respect to the product t-norm was given in [6].

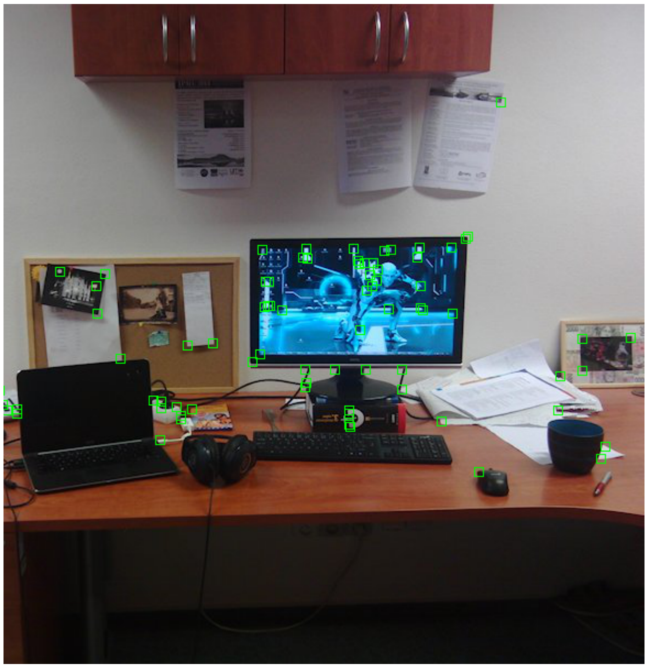


Figure 1. Image A with detected point features

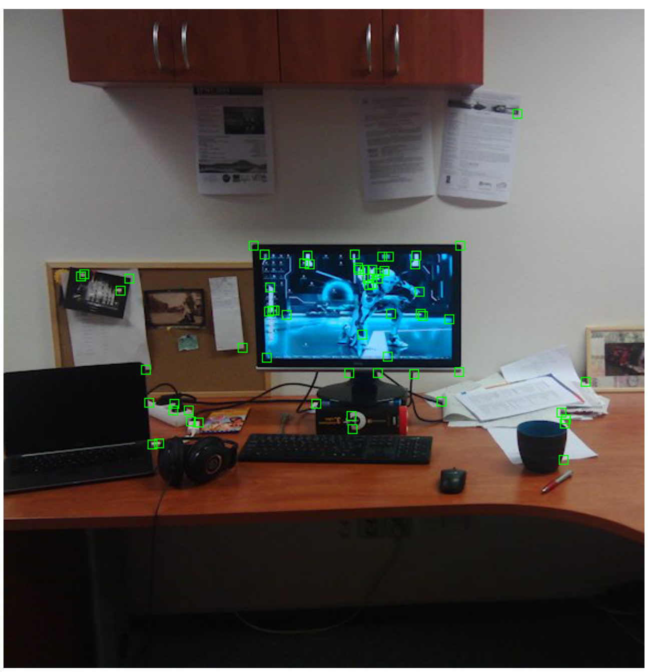


Figure 2. Image B with detected point features

The F-transform similarity is used to compare two areas around feature points of two images. If two feature points have similarity higher than threshold value, they are remembered and relative shift between them are stored. Finally, the sensed image is moved by shift obtained from these step. Registered images can be fused. Without the registering process, the final result image should containt defective artifacts.

C. Image fusion

The last step of the process is the merge of the registered images $u_1, ..., u_k$ into one result image. The image fusion based on F-transform is used for to do this. Image fusion is important field of image processing area, which generally aims to the following task: how to obtain result image which is somehow "better" than each of the input images.

Generally, let u is ideal image and $u_1, \ldots, u_K$ are acquired (input) images, then the relation between each u_i and u can be expressed by

$$u_i(x, y) = d_i(u(x, y)) + e_i(x, y), \;\; i = 1, \ldots, K \quad (19)$$

where d_i is an unknown operator describing the image degradation, and e_i is some random noise. The aim of the fusion is to obtain fused image $\hat{u}$ such that it is closer to u (and therefore "better") than any of $u_1, \ldots, u_K$.

If we apply F-transform based image fusion, we may introduce following representation of u on P:

$$u(x, y) = u_{nm}(x, y) + e(x, y), \quad (20)$$

$$e(x, y) = u(x, y) - u_{nm}(x, y), \quad (21)$$

where $0 < n \leq N$, $0 < m \leq M$, and u_{nm} is the inverse F-transform of u and e is the respective first difference. Value e represents *residuals* of the image u (see 21). If we replace e in (20) by its inverse F-transform e_{NM} with respect to the finest partition of $[1, N] \times [1, M]$, the above representation can then be rewritten as follows:

$$u(x, y) = u_{nm}(x, y) + e_{NM}(x, y), \forall(x, y) \in P. \quad (22)$$

We call (22) a *one-level decomposition* of u on P.

If function u is smooth, then the function e_{NM} is small. If an input image contains a significant noise or sharp changes (e.g. at edges), the residuals e_{NM} are higher. For two compared images, the lower residuals mean that the image has less noise or is less sharp. This property can be used to obtain the most sharpest image from the input ones, and also to reduce noise by filtering it from the input images and join them together. If one level decomposition is not enough, second (third etc.) level decomposition can be calculated over $e(x, y)$, until required quality is achieved. The detailed algorithm was presented in [9]. However, this processing is very time consuming, therefore some improvements of the original algorithm were researched ([16] or [17]). Finally, three different algorithms were tested to fuse feature–matched images presented in this article. Because lack of the space, only the one algorithm (called IESA) producing best results will be briefly explained.

Let $u^1, ..., u^k$ is the set of k input algorithms obtained by *feature matching* part of explained solution. The IESA algorithm consists of three main steps (there are some values set as fixed constants. Those constants are set by experiments and recommendations presented in [17]):

1) Set n, m values to 5.
2) For each of input image u^i (where $i \in [1, k]$): decompose image into inversed F-transform representation u^i_{nm} and calculate residuals $e^i_{nm} = u^i - u^i_{nm}$.
3) For each obtained residuals e^i_{nm} from previous step calculate blurred residuals $\hat{e}^i_{nm}$ using direct and inverse F-transform with $n = m = 25$.

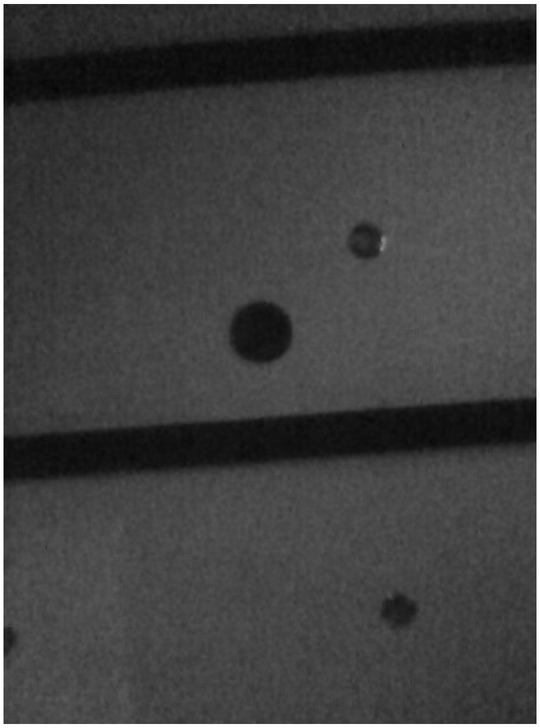


Figure 3. Input for image fusion (one of seven input images)

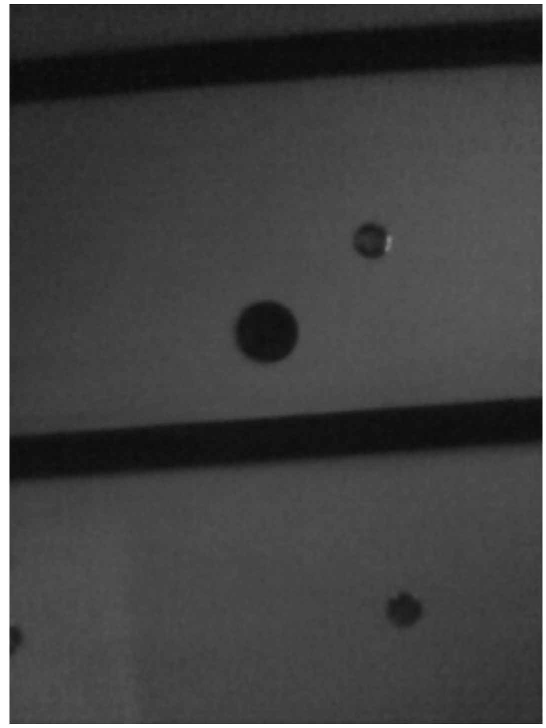


Figure 4. Image fusion - reduced noise (one output image)

4) For each residual value $\hat{e}^i_{nm}$ set $\hat{e}^i_{nm}(x,y) = 0$ if $\hat{e}^i_{nm} < 0.9$. This will suppress low residual values produced by noise.
5) Calculate final image f that for each pixel $f(x,y)$:
   a) From all residuals $\hat{e}^i_{nm}$ find the index j with the highest pixel value $\hat{e}^i_{nm}(x,y)$ in absolute: $abs(\hat{e}^j_{nm}(x,y)) \geq abs(\hat{e}^i_{nm}(x,y))$, $i \in [1,k]$.
   b) Set pixel value $f(x,y) = u^j(x,y)$.

The IESA algorithm takes the best part from each input image and creates result image with suppressed noise, emphasizing sharpest parts from each image. For multiple input images (Figure 3) the result can be seen at Figure 4. Another example can be seen at figure 5.

As a result of image fusion, one final image with improved quality and suppressed noise is produced from the set of input images.

IV. Conclusion

The article presented new approach based on F-transform technique to produce fused image with improved quality against the set of input images. The presented algorithm is applicable for the input images with varying in image scale, shift and light conditions. Further research will be aimed to support other kinds of variance, like image angle (different point of view of the camera) and rotation.

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 project and SGS18/PrF/2014 project.

References

[1] G. Qiu, J. Guan, J. Duan and M. Chen: "Tone Mapping for HDR Image using Optimization - A New Closed Form Solution", *Proceedings of the 18th International Conference on Pattern Recognition* (2006)

[2] Vavilin, A.; Jo, K. "Fast HDR image generation from multi-exposed multiple-view LDR images". *Visual Information Processing (EUVIP), 2011 3rd European Workshop on*, 105 – 110

[3] Perfilieva, I: Fuzzy transforms: Theory and applications, *Fuzzy Sets and Systems* 157, 993–1023 (2006).

[4] Perfilieva, I., De Baets, B.: Fuzzy Transform of Monotonous Functions with Applications to Image Processing, *Information Sciences* 180, 3304-3315 (2010).

[5] Hurtik, P., Perfilieva, I.: Image compression methodology based on fuzzy transform, *Advances in Intelligent and Soft Computing. Proc. Intern. Conf. on Soft Computing Models in Industrial and Environmental Applications (SoCo2012)*, 525–532 (2012).

[6] Hurtik, P., Perfilieva, I.: Image compression methodology based on fuzzy transform using block similarity. 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13). Atlantis Press (2013).

[7] Hurtik, P., and Perfilieva, I.: Image Reduction/Enlargement Methods Based on the F-Transform. Asturias: European centre for soft computing, 3–10 (2013).

[8] Zitova, B., Flusser, J.: Image registration methods: a survey. *Image and vision computing* 21(11), 977-1000 (2003).

[9] Perfilieva, I., Daňková, M.: Image fusion on the basis of fuzzy transforms, *Proc. 8th Int. FLINS Conf.*, Madrid, 471–476 (2008).

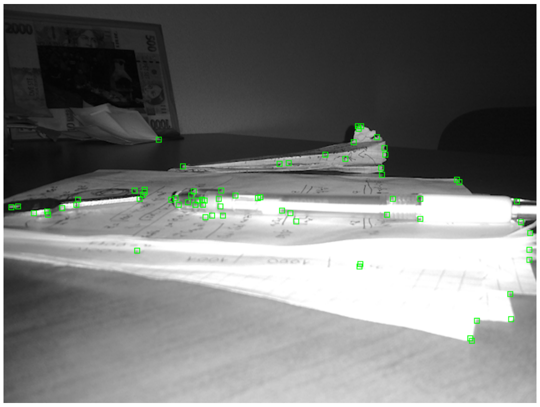
[10] Vajgl, M., Perfilieva, I., Hod'áková, P.: Advanced F-Transform-Based Image Fusion, *Advances in Fuzzy Systems* 2012, (2012).

[11] Perfilieva, I., Hodáková, P., Hurtik, P.: F^1-transform edge detector inspired by Canny's algorithm, *Advances on Computational Intelligence (IPMU2012)*, 230–239 (2012).

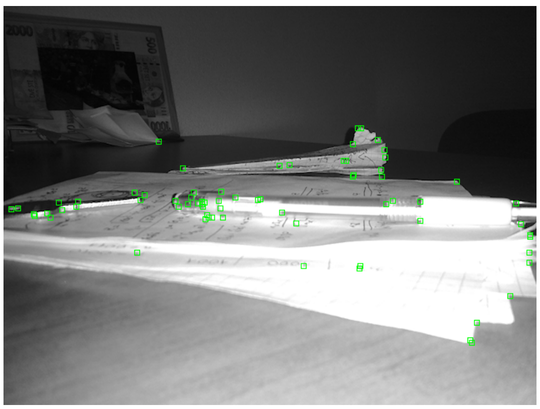
[12] Perfilieva, I., Daňková, M., Bede, B.: Towards F-transform of a higher degree, *Fuzzy Sets and Systems* 180, 3–19 (2011).

[13] Lowe, D, G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, (2004).

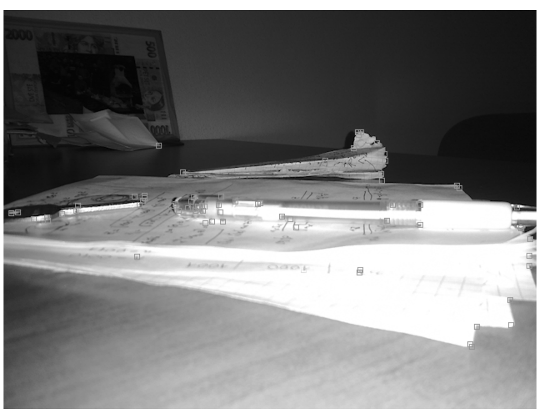
[14] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features.*Computer Vision – ECCV 2006.*, (2006).



(a) Input image A with detected point features



(b) Input image B with detected point features



(c) Result after image fusion

Figure 5. Example with table picture

[15] Wang, Z., Bovik, A. C., Sheikh H. R., Simoncelli, E. P.: Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing* 13, 600–612 (2004).

[16] I. Prefiljeva, M. Vajgl: Novel Image Fusion Based on F-transform. *2nd World Conference on Soft Computing Proceedings*. Letterpress publishing house, 2012. pp. 165–171. ISBN 9789952452372

[17] M. Vajgl, I. Perfilijeva: Improved F-Transform based image fusion. *15th International Conference IPMU 2014*, in processing, 2014.