IT4Innovations
Centre of Excellence

Advances in F-transform based image fusion

Martina Daňková Irina Perfilieva

Centre of Excellence IT4Innovations
division of the University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
Ostrava 1, Czech Republic

11'th FSTA 2012, Liptovský Ján, 30.1.-3.2, 2012

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj pro inovace

Outline

Seminal paper on F-transform:

[1] Perfilieva, I., Fuzzy Transform: Theory and Applications. Fuzzy sets and systems, 2006, Vol. 157 (8), 992–1023.

Based on results obtained with Radek Valášek:

[2] Daňková, M., Valášek, R.. Full Fuzzy Transform and the Problem of Image Fusion. Journal of electrical engineering. 2006, Vol. 12, 82–84.

and subsequently followed by

[3] Perfilieva, I., Daňková, M. Image Fusion on the Basis of Fuzzy Transforms. Computational Intelligence in Decision and Control. New Jersey: World Scientific, 2008, 471-476.

Image fusion – problem specification

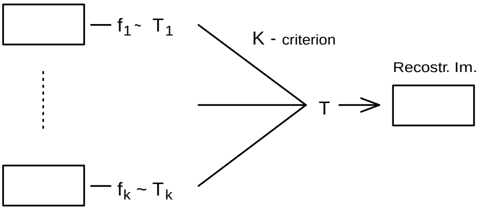
- What?

  "k" partially damaged images → one "good" image

- Tool?

  Aggregation operators, F-transform, Mathematical morphology, Wavelet transform

- How (Principle)?



F-transform tool

Let u be represented by the function $u : P \to \mathbb{R}$ of two variables where $P = \{(i, j) \mid i = 1, \ldots, N, j = 1, \ldots, M\}$ is an $N \times M$ array of pixels and $\mathbb{R}$ is the set of reals.

- Partition
  $A_1, \ldots, A_n \underset{\sim}{\subset} [1, N]$ establish a *fuzzy partition* of $[1, N]$ if the following requirements are fulfilled:
  (i) Locality: for every $k = 1, \ldots, n$, $A_k(x) = 0$ if $x \in [1, N] \setminus [x_{k-1}, x_{k+1}]$ where $x_0 = x_1$, $x_{N+1} = x_N$;
  (ii) Continuity: for every $k = 1, \ldots, n$, A_k is continuous on $[x_{k-1}, x_{k+1}]$ where $x_0 = x_1$, $x_{N+1} = x_N$;
  (iii) Ruspini: for every $i = 1, \ldots, N$, $\sum_{k=1}^{n} A_k(i) = 1$;
  (iv) Non-emptiness: for every $k = 1, \ldots, n$, $\sum_{i=1}^{N} A_k(i) > 0$.

F-transform tool – Direct F-transform

Let $u : P \to \mathbb{R}$ and fuzzy sets $A_k \subsetneqq [1, N], B_l \subsetneqq [1, M]$, $k = 1, \ldots, n$, $l = 1, \ldots, m$ establish a fuzzy partition of $[1, N] \times [1, M]$.

- Direct F-transform of u is an image of the mapping $F[u] : \{A_1, \ldots, A_n\} \times \{B_1, \ldots, B_m\} \to \mathbb{R}$ defined by

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

- Each component $F[u]_{kl}$ is a local mean value of u over a support set of the respective fuzzy sets (A_k, B_l).

F-transform tool – Inverse F-transform

Components $F[u]_{kl}$ can be arranged into the matrix or vector representation as follows:

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

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

$$u_{nm}(i,j) = \sum_{k=1}^{n} \sum_{l=1}^{m} F[u]_{kl} A_k(i) B_l(j). \tag{2}$$

Properties of F-transform

Let $[a, b]$ be h-uniformly partitioned by $A_1, \dots, A_n$, where $n > 2$ and $h = (b - a)/(n - 1)$,
f be a continuous function on $[a, b]$,
$F_1, \dots, F_n$ be the F-transform components of f w.r.t. $A_1, \dots, A_n$.

P1. The k-th component F_k $(k = 1, \dots, n)$ minimizes the function

$$\Phi(y) = \int_a^b (f(x) - y)^2 A_k(x) dx.$$

Properties of F-transform

P2. For each $k = 1, \ldots, n-1$, and for each $t \in [x_k, x_{k+1}]$ the following estimations hold:

$$|f(t) - F_k| \leq 2\omega(h, f), \qquad |f(t) - F_{k+1}| \leq 2\omega(h, f)$$

where

$$\omega(h, f) = \max_{|\delta| \leq h} \max_{x \in [a, b-\delta]} |f(x+\delta) - f(x)|$$

is the modulus of continuity of f on $[a, b]$.

Properties of F-transform

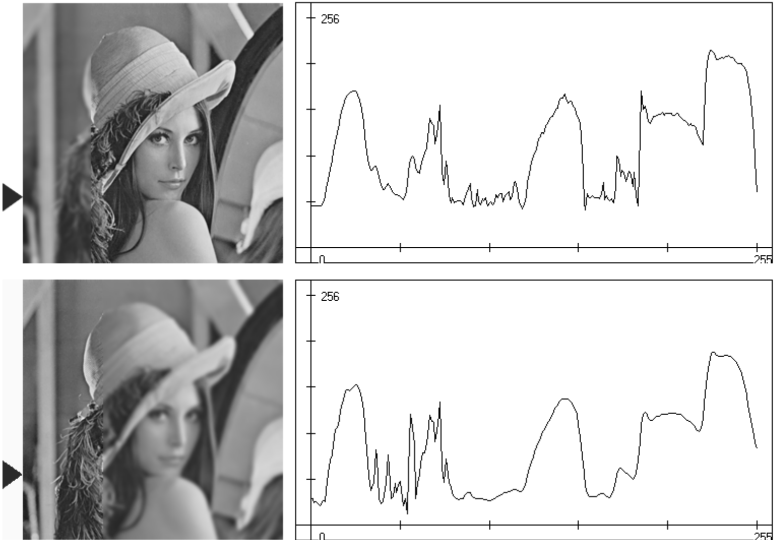
P3. F-transform of a continuous periodical function f on $[a, b]$ with period $2h$ such that

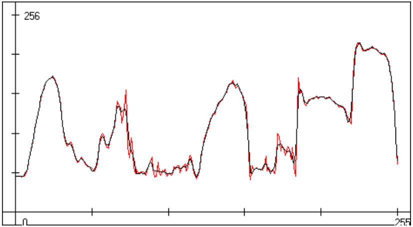
$$f(x_k - x) = -f(x_k + x), \quad x \in [x_{k-1}, x_{k+1}].$$

Then all components of the direct F-transform are equal to zero.

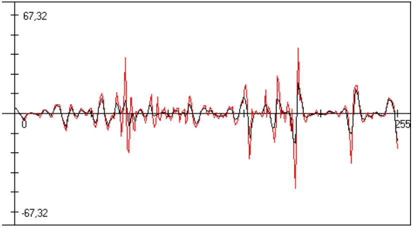
- F-transform can remove a noise with the above characteristic.

Motivation for the use of F-transform

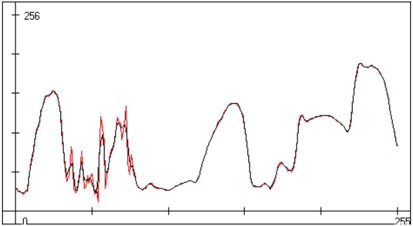




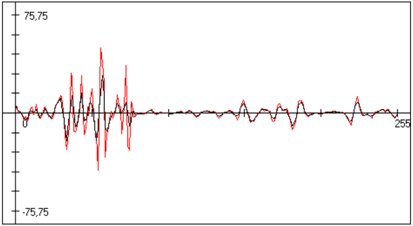
(a) $f^{(1)}$ and $f^{(1)}_{128}$



(b) $f^{(1)} - f^{(1)}_{128}$ and its approximation



(c) $f^{(2)}$ and $f^{(2)}_{128}$



(d) $f^{(2)} - f^{(2)}_{128}$ and its approximation

Expected result of fusion using F-transform

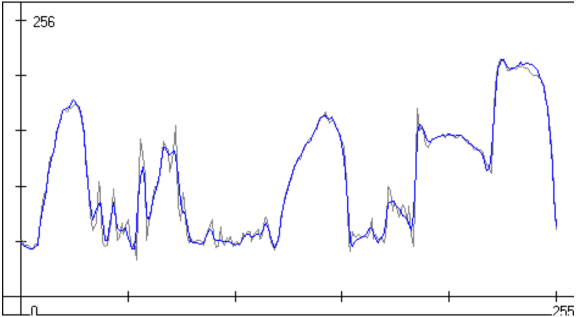


Figure: Fused function.

Justification of using F-transform

- By **P2**, smaller the modulus of continuity higher the quality of approximation of an input image by its inverse fuzzy transform.
- If a certain part of an input image is affected by degradation producing small variations between functional values in a close neighborhood, then by **P1**, the respective fuzzy transform component capture the weighted arithmetic mean and the error function is close to zero at that part.

Decomposition for image fusion

- By decomposition we extract relevant information.
- We distinguish One-level and Higher-level image decompositions.

Further, we deal with a real-valued function $u = u(x, y)$ defined on the $N \times M$ array of pixels
$P = \{(i, j) \mid i = 1, \ldots, N, j = 1, \ldots, M\}$ so that $u : P \to \mathbb{R}$.

One-level decomposition

We stem from the following representation of u on P:

$$u(x,y) = u_{nm}(x,y) + e(x,y), \text{ where } 0 < n \leq N, 0 < m \leq M,$$
$$e(x,y) = u(x,y) - u_{nm}(x,y), \forall (x,y) \in P.$$

where $n < N$, $m < M$, u_{nm} is the inverse F-transform of u and e is the respective residuum.

Image = F-transform of Image + Error

Higher-level decomposition

We decompose the error function e using its inverse F-transform with respect to a finer fuzzy partition of P that consists of $n' : n < n' \leq N$ and $m' : m < m' \leq M$ basic functions respectively. Thus we will obtain 2-level decomposition

$$u(x,y) = u_{nm}(x,y) + e_{n'm'}(x,y) + e'(x,y),$$
$$e'(x,y) = e(x,y) - e_{n'm'}(x,y), \ \forall (x,y) \in P.$$

Image = F-transform of Image + F-transform of Error + Error

Higher-level decomposition

The higher-level decomposition ($(k-1)$-level) formula:

$$u(x,y) = u_{n_1 m_1}(x,y) + \sum_{r=1}^{k-2} e^{(r)}_{n_{r+1} m_{r+1}}(x,y) + e^{(k-1)}(x,y),$$

where

$$\begin{aligned}
&0 < n_1 \leq n_2 \leq \ldots \leq n_{k-1} \leq N,\\
&0 < m_1 \leq m_2 \leq \ldots \leq m_{k-1} \leq M,\\
&e^{(1)}(x,y) = u(x,y) - u_{n_1 m_1}(x,y),\\
&e^{(i)}(x,y) = e^{(i-1)}(x,y) - e^{(i-1)}_{n_i m_i}(x,y),\\
&\text{for } i = 2,\ldots,k-1 \text{ and } (x,y) \in P.
\end{aligned}$$

F-transform based image fusion algorithms

There are two algorithms:

1. Simple F-transform based image fusion algorithm – based on one-level decomposition.
2. Complete F-transform based algorithm – based on higher-level decomposition.

Fusion operator $\kappa : \mathbb{R}^K \to \mathbb{R}$ which is defined as follows:

$$\kappa(x_1, \dots, x_K) = x_p, \text{ if } |x_p| = \max(|x_1|, \dots, |x_K|). \tag{3}$$

Using κ we chose components with maximal absolute values, which should correspond to parts of the least degraded input images.

Simple algorithm

Based on one-level decomposition

(1) Decompose input images into inverse F-transforms and error functions using one-level decomposition formula.

(2) Apply fusion operator to the inverse F-transforms to produce a fused F-transform.

(3) Apply fusion operator to the error functions to produce a fused error function.

(4) Reconstruct the fused image from the fused F-transform and the fused error function.

Complete algorithm

Based on higher-level decomposition

(1) Decompose input images into inverse F-transforms and error functions using higher-level decomposition formula.
(2) Apply fusion operator to the inverse F-transforms to produce a fused F-transforms.
(3) Apply fusion operator to the error functions to produce a fused error function.
(4) Reconstruct the fused image from the fused F-transforms and the fused error function.

Complexity

- A computational complexity of CA is higher than SA.
- Complexity of an inverse F-transform is

$$\mathcal{O}(N \cdot M)$$

- CA uses higher number of F-transforms multiplied by the number of images K.
- And finally, there is the fusion operator, which compares $Kk_{\max}mn + KMN$ values.

Hence, a speed of CA depends mainly on the number of iterations processed in the algorithm.

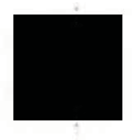
Fusion of blurred images

(a) 1'st blurred image (b) 2'nd blurred image (c) O – Original image

Fusion of blurred images

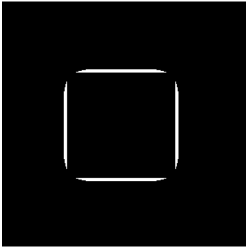
(d) F_1 – fused image by SA with $n, m = 1$

(e) F_2 – fused image by SA with $n, m = 9$

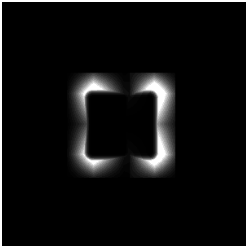


(f) F_3 – fused image by CA with $n_{start}, m_{start} = 1, k_{\max} = 8, step = 2$

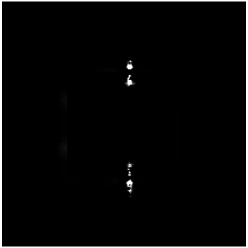
Fusion of blurred images



(g) $|O - F_1|$



(h) $|O - F_2|$



(i) $|O - F_3|$

Fusion of multi-channel color images

(j) C_1 – 1'st image (background in focus)

(k) C_2 – 2'nd image (toy in focus)

Fusion of multi-channel color images

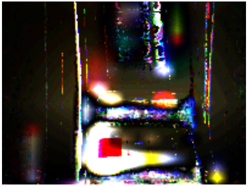
(l) F_1 – fused image by CA with $n_{start}, m_{start} = 1, k_{\max} = 5, step = 2$

(m) F_2 – fused image by CA with $n_{start}, m_{start} = 10, k_{\max} = 4, step = 2$

Fusion of multi-channel color images

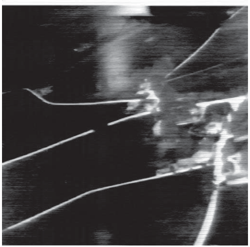
(n) R – region of interest cut from C_1

(o) $|R - R_{F_1}|$



(p) $|R - R_{F_2}|$

Fusion of multi-sensor images

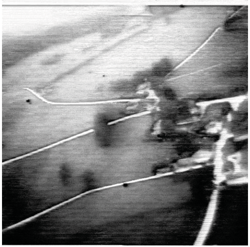


(q) FLIR image

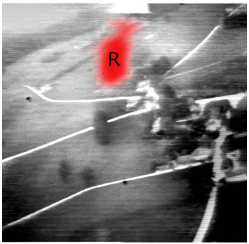
(r) LLTV image

(s) Fusion by means of multiresolution analysis

Fusion of multi-sensor images



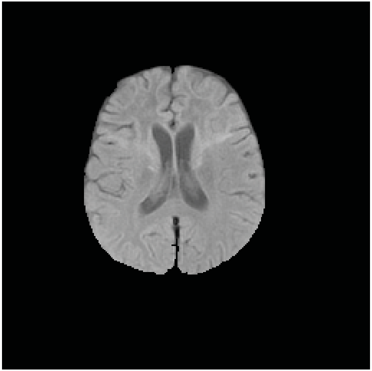
(t) SA $n, m = 100$



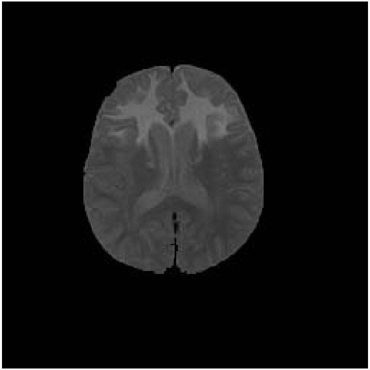
(u) Problematic part R

(v) CA with $n_{start}, m_{start} = 40$, $k_{max} = 3$, $step = 2$

Fusion of multi-sensor images



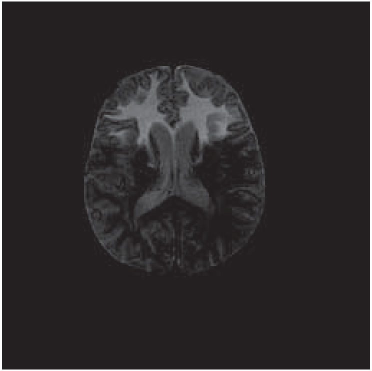
(w) 1'st MRI image



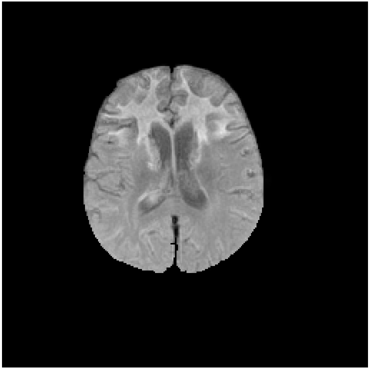
(x) 2'nd MRI image

Figure: One axial slice of dual-echo magnetic resonance imaging acquisitions (pathological brain image), Courtesy Professor Catherine Adamsbaum, Saint Vincent de Paul Hospital, Paris.

Fusion of multi-sensor images



(a) 2'nd image with modified contrast



(b) CA with $n_{start}, m_{start} = 10$, $k_{max} = 5$, $step = 2$

F-transform tool

Let $\mathcal{L} = \langle [0,1], *, \rightarrow, \wedge, \vee, 0, 1 \rangle$ be a residuated lattice.

- Partition

  $A_1, \ldots, A_n \underset{\sim}{\subset} [1, N]$ establish a *fuzzy partition* of $[1, N]$ if the following requirements are fulfilled:

  (i) Locality: for every $k = 1, \ldots, n$, $A_k(x) = 0$ if $x \in [1, N] \setminus [x_{k-1}, x_{k+2}]$ where $x_0 = x_1$, $x_{N+1} = x_N$;

  (ii) Continuity: for every $k = 1, \ldots, n$, A_k is continuous on $[x_{k-1}, x_{k+2}]$ where $x_0 = x_1$, $x_{N+1} = x_N$;

  (iii) Ruspini-like: for every $i = 1, \ldots, N$, $\bigvee_{k=1}^{n} A_k(i) = 1$;

  (iv) Non-emptiness: for every $k = 1, \ldots, n$, $\sum_{i=1}^{N} A_k(i) > 0$.

Let $u : P \rightarrow [0, 1]$ and fuzzy sets $A_k \underset{\sim}{\subset} [1, N]$, $B_l \underset{\sim}{\subset} [1, M]$, $k = 1, \ldots, n$, $l = 1, \ldots, m$ establish a fuzzy partition of $[1, N] \times [1, M]$.

Direct and inverse F-transform

- Direct F-transform of u is an image of the mapping $F[u] : \{A_1, \ldots, A_n\} \times \{B_1, \ldots, B_m\} \to \mathbb{R}$ defined by

$$F[u](A_k, B_l) = \bigwedge_{i=1}^{N} \bigwedge_{j=1}^{M} [A_k(i) * B_l(j)] \to u(i, j), \tag{4}$$

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

$$u_{nm}(i, j) = \bigvee_{k=1}^{n} \bigvee_{l=1}^{m} [A_k(i) * B_l(j) * F[u]_{kl}]. \tag{5}$$

Properties of F-transform

P1. Each component $F[u]_{kl}$ is a local minimum value of u over a support set of the respective fuzzy sets (A_k, B_l).

P2. $u_{nm}(i,j) \leq u(i,j), \ \forall (i,j)$ - lower approximation.

P3. $F[u]_{kl}, \ \forall k, l$ are the best components from all g_{kl} satisfying

$$\bigvee_{k=1}^{n} \bigvee_{l=1}^{m} [A_k(i) * B_l(j) * g_{kl}] \leq u(i,j).$$

One-level decomposition

We stem from the following representation of u on P:

$$u(x, y) = u_{nm}(x, y) + e(x, y), \text{ where } 0 < n \leq N, 0 < m \leq M,$$
$$e(x, y) = u(x, y) - u_{nm}(x, y), \forall(x, y) \in P.$$

where $n < N$, $m < M$, u_{nm} is the inverse F-transform of u and e is the respective residuum.

Image = F-transform of Image + Error

Simple algorithm

Based on one-level decomposition

(1) Decompose input images into inverse F-transforms and error functions using one-level decomposition formula.

(2) Apply fusion operator to the inverse F-transforms to produce a fused F-transform.

(3) Apply fusion operator to the error functions to produce a fused error function.

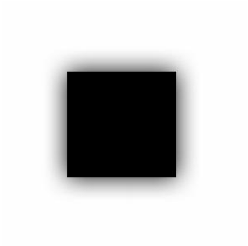
(4) Reconstruct the fused image from the fused F-transform and the fused error function.

Fusion operator $\kappa : \mathbb{R}^K \to [0, 1]$ which is defined as follows:

$$\kappa(x_1, \ldots, x_K) = \max(x_1, \ldots, x_K). \tag{6}$$

Using κ we chose components with maximal values.

Fusion by F-transform over a residuated lattice



(c) Fused squares

(d) Fused toy

(e) Fused road

Simple algorithm - generally

Based on one-level decomposition

(1) Decompose input images into inverse F-transforms and error functions using one-level decomposition formula.

(2) Apply 1'st fusion operator to the inverse F-transforms to produce a fused F-transform.

(3) Apply 2'nd fusion operator to the error functions to produce a fused error function.

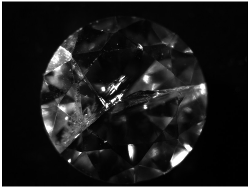
(4) Reconstruct the fused image from the fused F-transform and the fused error function.

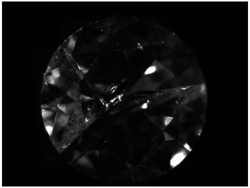
Fusion operators $\kappa_{1,2} : \mathbb{R}^K \to [0,1]$ – aggregation operator that can be defined e.g. as

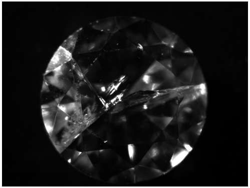
$$\kappa_1(x_1, \dots, x_K) = \min(x_1, \dots, x_K),$$
$$\kappa_2(x_1, \dots, x_K) = \max(x_1, \dots, x_K).$$

Using $\kappa_{1,2}$ we aggregate components that fits our purposes.

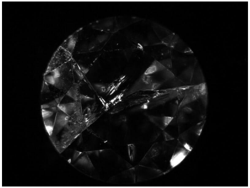
Fusion using various fusion operators


(f) Stone 1


(g) Stone 2


(h) Stone 3

Fusion using various fusion operators



(i) Fusion 1

(j) Fusion 2

(k) Fusion 3

Conclusions

- +++ SA proceeds very fast.
- - SA demands manual setting.
- - CA rapidly slow down with high number of iterations.
- + CA provides highly desirable result with maximal setting.
- + higher contrast elements are propagated.
- - by P3. SA and CA increase noise from input images.

Hence, the optimal choice is

SA with preset numbers of components $(N/2, M/2)$.