NONLINEAR FILTERING OF 3D MRI IN MATLAB

J. Kukal, D. Majerová, V. Musoko, A. Pavelka and A. Procházka

ICT Prague, Department of Computing and Control Engineering

Abstract. *The 3D image of human head obtained by T2 magnetic resonance imaging (MRI) can be represented by three-dimensional matrix of voxel intensities. There is a chance to use eight neighbor voxels for the 3D interpolation. The linear interpolation can be realized as a mean of eight. The other nonlinear statistics like median, trimmed average or BES can be used for the 3D filtering. The software for the 3D nonlinear denoising of MRI signal was built in the Matlab environment. The filter properties was also studied and compared. The great advantage of proposed filter techniques can be visualized on the 3D brain models.*

Keywords: *nonlinear filter, 3D image, MRI, denoising, Matlab.*

1 Primer of LA_{sqrt}

The mathematical background of 3D image processing is *Łukasiewicz algebra enriched by a square root function* (LA_{sqrt}) which is defined as

$$\text{LA}_{sqrt} = \{\mathbf{L}, \wedge, \vee, \otimes, \rightarrow, sqrt, 0, 1\}$$

where $\mathbf{L} = [0,1] \subset \mathbf{R}$, conjunction ($\wedge$), disjunction ($\vee$), Łukasiewicz multiplication ($\otimes$) and residuum ($\rightarrow$) are basic operations and *sqrt* is the square root function.

It is useful to introduce several derived operators, e.g. negation, equivalence, non-equivalence, addition or subtraction. There is possible to describe the basic and derived operators as *basic functions* as described in Tab. 1.

function	**formula**
negation	$\psi_1(x) = \neg x = 1 - x$
square root	$\psi_2(x) = sqrt(x) = (1 + x)/2$
conjunction	$\psi_3(x, y) = x \wedge y = \min(x, y)$
disjunction	$\psi_4(x, y) = x \vee y = \max(x, y)$
Łukasiewicz multiplication	$\psi_5(x, y) = x \otimes y = \max(x + y - 1, 0)$
residuum	$\psi_6(x, y) = x \rightarrow y = \min(1 - x + y, 1)$
equivalence	$\psi_7(x, y) = x \leftrightarrow y = 1 - \lvert x - y\rvert$
non-equivalence	$\psi_8(x, y) = x \circ y = \lvert x - y\rvert$
addition	$\psi_9(x, y) = x \oplus y = \min(x + y, 1)$
subtraction	$\psi_{10}(x, y) = x \ominus y = \max(x - y, 0)$
multiplication by integer	$\psi_{11}(x, n) = n \odot x = \min(n \cdot x, 1)$
integer power	$\psi_{12}(x, n) = x^n = \max(n \cdot x - n + 1, 0)$

Table 1: Basic functions in LA_{sqrt} for $x, y \in \mathbf{L},\ n \in \mathbf{N}_0$

The *fuzzy logic expression* (FLE) is defined by the rules:

- Any free variable $x \in \mathbf{L}$ is FLE.
- Any constant $a \in \mathbf{L}$ is FLE.

- $\psi_i(FLE)$ is FLE for $i = 1, 2$,
- $\psi_j(FLE, FLE)$ is FLE for $j = 3, \ldots, 10$,
- $\psi_k(FLE, n)$ is FLE for $k = 11, 12$ and $n \in \mathbf{N}_0$

where ψ_m $(m = 1, \ldots, 12)$ are the basic functions in LA$_{sqrt}$.

Let $n \in \mathbf{N}$, $\vec{x} \in \mathbf{L}^n$ and $\varphi: \mathbf{L}^n \rightarrow \mathbf{L}$. If $\varphi(\vec{x})$ is FLE then φ is called a *fuzzy logic function* (FLF) in LA$_{sqrt}$. The main advantage of LA$_{sqrt}$ is a constrain sensitivity of any FLF to its input variables as proven in [3].

2 Useful FLFs for 3D Denoising

Let $S = (x_1, \ldots, x_n)$ be a list of values $x_k \in [0, 1]$. Let $O = (x_{(1)}, \ldots, x_{(n)})$ be an ordered list of values from S. Let $y \in [0, 1]$ be the output of 3D denoising filter. Then the *FLF denoising filter* is based on the formula

$$y = f(x_1, \ldots, x_n)$$

where $f: [0, 1]^n \rightarrow [0, 1]$ is a FLF.

There are several FLFs useful for 3D denoising:

- mean of eight $f_1 = \frac{1}{8} \cdot \sum\limits_{k=1}^{8} x_k$,
- double trimmed mean of eight $f_2 = \frac{1}{4} \cdot \sum\limits_{k=3}^{6} x_{(k)}$,
- median of eight $f_3 = \frac{1}{2} \cdot \left(x_{(4)} + x_{(5)}\right)$,
- BES of eight $f_4 = \frac{1}{4} \cdot \left(x_{(2)} + x_{(4)} + x_{(5)} + x_{(7)}\right)$,
- mean of four $f_5 = \frac{1}{4} \cdot \sum\limits_{k=1}^{4} x_k$,
- median of four $f_6 = \frac{1}{2} \cdot \left(x_{(2)} + x_{(3)}\right)$,
- trimmed mean of six $f_7 = \frac{1}{4} \cdot \sum\limits_{k=2}^{5} x_{(k)}$,
- median of six $f_8 = \frac{1}{2} \cdot \left(x_{(3)} + x_{(4)}\right)$,

3 Direct FLF Denoising

Let the values $x_1, \ldots, x_8$ are obtained from the $2 \times 2 \times 2$ cube of eight neighbor voxels from the original 3D image. The direct FLF denoising is a process of "body centered" interpolation which is not necessary linear one. We can use one of $f_1, \ldots, f_4$ FLFs and apply them to the original corner values $x_1, \ldots, x_8$.

4 Hierarchical FLF Denoising

The 2^3 cube of eight voxels can be decomposed in three directions to the three pairs of 2^2 squares. Then the six lists of size four are formed

$$\begin{aligned} S_1 &= (x_1, x_2, x_3, x_4) \\ S_2 &= (x_5, x_6, x_7, x_8) \\ S_3 &= (x_1, x_2, x_5, x_6) \\ S_4 &= (x_3, x_4, x_7, x_8) \\ S_5 &= (x_1, x_3, x_5, x_7) \\ S_6 &= (x_2, x_4, x_6, x_8) \end{aligned}$$

The first step of hierarchical FLF processing is based on the "face centered" interpolation using the average of four

$$h_k = f_5(S_k) \quad \text{for } k = 1, \ldots, 6$$

or the median of four

$$h_k = f_6(S_k) \quad \text{for } k = 1, \ldots, 6.$$

The second and last step is based on the denoising by the six points interpolation using the trimmed average of six as

$$y = f_7(h_1, \ldots, h_6)$$

or rather the median of six as

$$y = f_8(h_1, \ldots, h_6).$$

There are four possibilities how to combine functions f_5, f_6 with f_7, f_8 ones.

5 Filter Testing

A set of eight FLF 3D filters was tested using the impulse and Gaussian noise. The impulse noise was represented by a list where $x_1 = 1$ and $x_k = 0$ for $k = 2, \ldots, 8$. The Gaussian noise was studied in the case when $x_k \sim N(0.5, 0.01)$ for $k = 1, \ldots, 8$. Then the standard deviation of Gaussian noise is $\sigma = 0.1$. The results are collected in the Tab. 2.

Filter	**1st level**	**2nd level**	y_I	s_G
F1	f_1	*	0.125	0.0354
F2	f_2	*	0.000	0.0355
F3	f_3	*	0.000	0.0353
F4	f_4	*	0.000	0.0363
F5	f_5	f_7	0.125	0.0354
F6	f_5	f_8	0.125	0.0354
F7	f_6	f_7	0.000	0.0354
F8	f_6	f_8	0.000	0.0355

Table 2: FLF filter properties

Every filter is described by a single FLF or by a pair of two FLFs. The output response to a single impulse noise is denoted as y_I and its ideal value is zero. It means that any filter, which use the poor average function (f_1, f_5) in the first step, is not robust in a statistical sense. The remaining filters F2, F3, F4, F7 and F8 are robust ones.

The experimental values of standard deviation s_G are also included in the Tab. 2 for the Gaussian noise. The best filter with a minimum value of s_G is F3.

6 Biomedical Application

The set of eight FLF filters were applied to 3D MRI T2 image and the results were compared using the MATLAB environment. A sample of results are depicted in the Figs. 1,2. The original slide No. 100 is depicted in the Fig. 1. The result of the best robust filter (F3) is depicted in the Fig. 2.

7 Conclusion

Eight 3D FLF filters were developed in LA_{sqrt}. After the realization in MATLAB environment, the process of their testing brought two results. The filters F2, F3, F4, F7 and F8 are robust ones. Then the best robust filter is F3. The great advantage of previous approach is in the simplicity of non-linear robust filtering.

References

[1] Mitra, S. K., Kaiser, J. F.: *Handbook for Digital Signal Processing.* John Wiley & Sons. New York. 1993.

[2] Hodges, J. L., Lehmann, E. L. On Medians and Quasi Medians. *Journal of the American Statistical Association*, 1967, 62, p.926–931.

[3] Majerová, D., Kukal, J. Multicriteria approach to 2D image de-noising by means of łukasiewicz algebra with square root. *Neural Network World*, 2002, 12, p. 333–348.

Jaromír Kukal, Dana Majerová, Victor Musoko, Aleš Pavelka, Aleš Procházka

INSTITUTE OF CHEMICAL TECHNOLOGY, Prague
Department of Computing and Control Engineering
Technická 5, 166 28 Prague 6 Dejvice
Phone: +420 224 354 170, fax: +420 224 355 053
E-mails: {Dana.Majerova, Jaromir.Kukal, Ales.Prochazka}@vscht.cz
WWW address: http://uprt.vscht.cz/

Figure 1: Original image

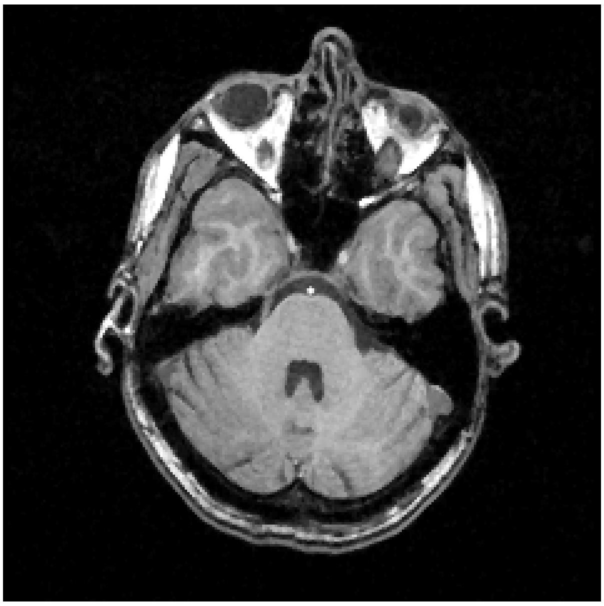

Figure 2: The best robust FLF filter