Objective Approach to SPECT Filtering

Dana Majerová[1], *Otto Lang*[2], *Jaromír Kukal*[1] *and Aleš Procházka*[1]

[1] ICT Prague, Department of Computing and Control Engineering
[2] Charles University Prague, 3rd Faculty of Medicine

Abstract. *The SPECT technique is based on inverse Radon transformation with digital filtering inside. The Metz, Hanning and Butterworth filters are used in standard software packages. The resulting images were subjects of multidimensional PCA based on extended Oja learning. The resulting 2D maps of various filters were obtained as results of MATLAB calculations.*

Keywords: *SPECT, Metz, Hanning, Butterworth, enhancement, built-in filter, Oja learning, principal component analysis, filter categorization, MATLAB.*

1 Introduction

The general question of optimum filtering or rather de-noising arises during the computer processing of SPECT images. The filtering of digital image occurs in frequency domain inside the complex process of inverse Radon transformation. The Metz, Hanning and Butterworth filters of various order, cutoff and enhancement are frequently used as built-in tools inside the commercial software packages. The article investigates the technique for SPECT filter categorization using Oja neuron for principal component analysis of de-noised 2D images. The first two principal components were used for the mapping of properties of various filters with different parameters. The standard software for the analysis of Oja neuron and its learning paradigm has been well known since 1982. The linear neuron is learn to obtain the main eigenvector and eigenvalue of signal covariance matrix. Lately the previous principle was extended to the other eigenvalues and eigenvectors. It enables to compare the results with referential filter used in practice. The step from referential Metz filter toward Butterworth filters with enhancement is then possible to plan.

2 Filter Description

There are three types of built-in filters included in the standard software for the inverse Radon transformation. The Metz filter is described by its frequency spectrum as

$$|F_M(\omega)| = \frac{1-\left(1-\mathrm{MTF}^2\right)^n}{\mathrm{MTF}}$$

where $F_M(\omega)$ is the spectral function, ω is a frequency, $\mathrm{MTF} = \mathrm{MTF}(\omega, \mathrm{cutoff})$ is the modulation transfer function and n is the order of Metz filter.

The Hanning filter is the second one and its frequency spectrum is

$$|F_H(\omega)| = (1 + A \cdot \omega) \cdot \left(\frac{1 + cos\left(\pi\left(\frac{\omega}{Nq}\right)^B\right)}{2} \right)^n$$

where $F_H(\omega)$ is the spectral function, ω is a frequency, A is the enhancement parameter ($A = 0$), B is a self-adjusting parameter, n is the order of Hanning filter and $Nq = Nq(\mathrm{cutoff})$ is the Nyquist frequency.

Finally, the Butterworth filter is described by the frequency spectrum

$$|F_B(\omega)| = \frac{1 + A \cdot \omega}{\sqrt{1 + \left(\frac{\omega}{B \cdot Nq}\right)^{2n}}}$$

where $F_B(w)$ is the spectral function, ω is a frequency, $A \geq 0$ is the enhancement parameter, B is a self-adjusting parameter, n is the order of Butterworth filter and $Nq = Nq(\text{cutoff})$ is the Nyquist frequency.

3 Extended Oja Learning

Let $\mathbf{X}$ be original pattern set. Let $NORMALIZE(\mathbf{X})$ be function for pattern set normalization. The $OJA(\mathbf{X})$ be function for finding the main eigenvalue and corresponding eigenvector. Let $PROJ(\mathbf{X}, \vec{w})$ be function for projection into orthogonal subspace. Then the **Extended Oja Learning** is defined as

$$\mathbf{X}_1 = NORMALIZE(\mathbf{X})$$

$$\text{for s} = 1, \ldots, \text{H}:$$

$$(\lambda_s, \vec{w_s}) = OJA(\mathbf{X}_s)$$

$$\mathbf{X}_{s+1} = PROJ(\mathbf{X}_s, \vec{w_s})$$

The results of extended method are

$$\vec{\lambda} = (\lambda_1 | \ldots | \lambda_H) \in (\mathbf{R}_0^+)^H,$$

$$\mathbf{W} = (\vec{w_1} | \ldots | \vec{w_H}) \in \mathbf{R}^{n \times H}.$$

Now $\mathbf{Y} = \mathbf{X}_1 \mathbf{W}$ is a picture of given pattern set and

$$\text{Eff} = \frac{1}{\text{n}} \sum_{\text{s}=1}^{\text{H}} \lambda_{\text{s}}$$

is the efficiency of the transformation.

4 Experimental Part

The Extended Oja Learning was realized in the Matlab environment as a set of functions for normalization, Oja learning, projection and the main function for the whole method (see Figs. 5–8). Then one scan of vessel system of human brain was measured using the SPECT technique. The Metz filter of 3rd order with cutoff = 31 was used first as referential filter because of frequently used for a good practical purposes on given apparatus.

After this first experiment the Metz filters were studied for various orders $n = 2, \ldots, 5$ and cutoffs $= 31, \ldots 43$. The resulting 2D images were proceeded as vectors using the Extended Oja Learning. The result of principle component analysis (PCA) is demonstrated in Fig. 1. First two components PCA_1, PCA_2 were directly obtained from matrix Y as $\text{PCA}_1 = y_{k1}$, $\text{PCA}_2 = y_{k2}$ for $H = 2$ where $k = 1, \ldots, m$ is the index of filter. The values λ_1, λ_2 are first two elements of vector $\vec{\lambda}$. Every Metz filter is represented by empty circle except referential filter which is black one. Filters of the same order are connected by lines. The number in the circle neighborhood represents the value of cutoff. The extreme values of normalized components $\text{PCA}_1/\lambda_1^{1/2}$, $\text{PCA}_2/\lambda_2^{1/2}$ represent extreme cases of filtering by Metz filter while the referential filter is near the center. It means the referential Metz filter is a good compromise from the others.

The Hanning filters were studied in the third experiment. The poor properties of Hanning filters of various orders $n = 2, \ldots, 6$ and cutoffs $= 10, \ldots, 18$ are demonstrated in Fig. 2. The black circle with the minimum value of PCA_2 is the symbol of referential Metz filter again. The Hanning filter of 2^{nd} order and cutoff $= 10$ is the nearest filter to the referential one. The other Hanning filters are worse. Then using the Hanning filters for SPECT brain map de-noising is not recommended.

The fourth experiment was oriented to Butterworth filters without enhancement. The resulting PCA map is demonstrated in Fig. 3. Every Butterworth filter is represented by empty circle and the referential Metz filter is black one. Filters of the same order are connected by lines. The number in the circle neighborhood represents the value of cutoff. The extreme values of normalized components $\mathrm{PCA}_1/\lambda_1^{1/2}$, $\mathrm{PCA}_2/\lambda_2^{1/2}$ represent extreme cases of filtering by Butterworth filter. The Butterwoth filter of 5^{th} with cutoff $= 14$ the nearest to the referential Metz filter. Then the Butterworth filters recommended both for technical and biomedical applications are able to substitute the referential Metz filter.

The last experiment began with the best Butterworth filter of the 5^{th} order and cutoff $= 14$ without any enhancement. Then the influence of enhancement and cutoff were studied for the Butterworth filters of 5^{th} order. The resulting PCA map is in Fig. 4 where the solid line connects the filters of the same cutoff and the dotted line connects the filters of the same value of enhancement.

5 Conclusions

The technique of principal component analysis based on Oja learning was used for SPECT built-in filter comparative study. The referential Metz filter as subjective optimum one was used for the other filter mapping. The main results of comparative study are recognizing the inadequate properties of Hanning filter and very good properties of Butterworth filter for SPECT brain vessel analysis. The Butterworth filter of 5^{th} order is very close to the referential Metz filter. Using the 2% (0.020) enhancement the better subjective result was obtained. The side product of our work is the methodology of objective filter mapping with single subjective optimum referential filter including the general software for PCA analysis of large 2D images with thousands of pixels. The comparative methodology is also recommended in the cases of referential filter absence.

References

[1] Kak A. C., Slaney M.: *Principles of Computerized Tomographic Imaging.* Society of Industrial and Applied Mathematics. 2001.

[2] Kukal J., Majerová D. and Kukalová M.: Coming Back to Oja Learning. In: *MENDEL 2002*, 217–220. 2002.

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

[4] Oja E.: *A Simplified Neuron Model as a Principal Component Analyzer.* Journal of Mathematical Biology, 15, 267–273. 1982.

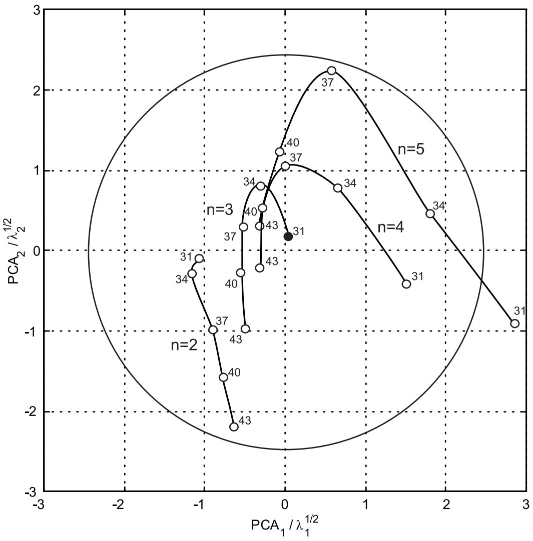

Figure 1: Metz filters

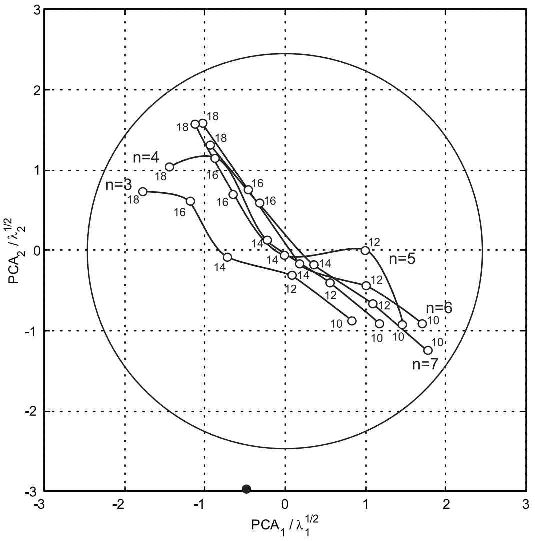

Figure 2: Hanning filters

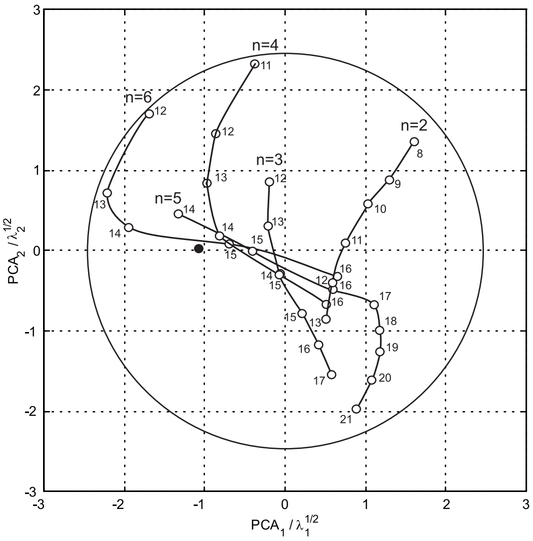

Figure 3: Butterworth filters without enhancement

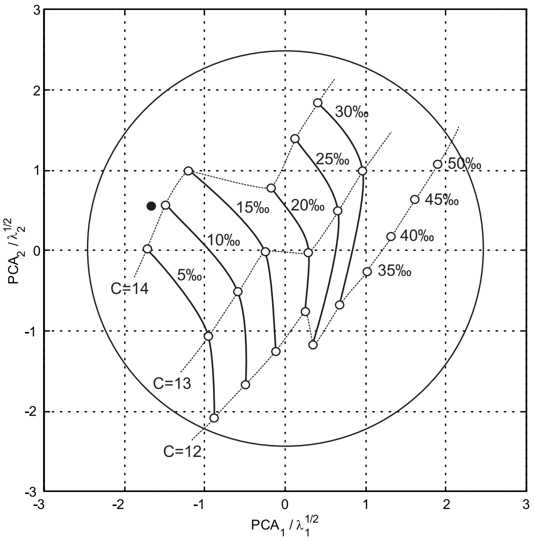

Figure 4: Butterworth filter of 5th order with enhancement

Contacts

Dana Majerová, Jaromír Kukal, 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/

Otto Lang
CHARLES UNIVERSITY, Prague
3rd Faculty of Medicine
Šrobárova 50, 100 34 Prague 10
Phone: +420 267 163 186, fax: +420 267 162 660
E-mail: lang@fnkv.cz
WWW address: http://www.fnkv.cz/kliniky/nukl/index.html

```
function [x,ex,dx]=ojanormalize(x)

[m,n]=size(x);
ex=mean(x);
for j=1:n
      x(:,j)=x(:,j)-ex(j);
end
dx=zeros(1,n);
for j=1:n
      dx(j)=sum(x(:,j).^2)/(m-1);
end
dx=max(dx,1e-100);
for j=1:n
      x(:,j)=x(:,j)/sqrt(dx(j));
end
```

Figure 5: Function `ojanormalize`

```
function [w,y,lambda]=ojalearn(x,zeta0,n0,nmax)

[m,n]=size(x);
w=ones(n,1)/sqrt(n);
for k=1:nmax
      zeta=zeta0*min(1,n0/k);
      i=unidrnd(m,1,1);
      y=x(i,:)*w;
      w=w+zeta*y*(x(i,:)'-y*w);
      w=w/norm(w);
end
y=x*w;
if max(w)<-min(w)
      w=-w;
end
lambda=sum(y.*y)/m;
```

Figure 6: Function `ojalearn`

```
function x=ojanext(x,w)

[m,n]=size(x);
for i=1:m
      xxx=x(i,:);
      x(i,:)=xxx-(xxx*w)*w';
end
```

Figure 7: Function `ojanext`

```
function [w,y,lambda,ex,dx]=ojapca(x,h,zeta0,n0,nmax)
% PCA using Oja learning
% [w,y,lambda,ex,dx]=ojapca(x,h,zeta0,n0,nmax);
% w ......  weight matrix (n,h)
% y ......  output matrix (m,h)
% lambda eigenvalue vector (1,h)
% ex .....  average x vector (1,n)
% dx .....  variance x vector (1,n)
% x ......  input matrix (m,n)
% h ......  PCA dimension (defaoult=2)
% zeta0 ..  learning rate (defaoult=0.5)
% n0 .....  searching period (defaoult=1)
% nmax ...  number of Oja learning steps (defaoult=100000*n0)
% Example:
% [w,y,lambda,ex,dx]=ojapca([1 2 3; 3 4 5; 5 6 7; 7 8 9; 9 10 11]);

[x,ex,dx]=ojanormalize(x);
disp('normalized')
if nargin==1; h=2; end
if nargin<=2; zeta0=0.5; end
if nargin<=3; n0=1; end
if nargin<=4; nmax=100000*n0; end
w=[];
y=[];
lambda=[];
for k=1:h
      [ww,yy,ll]=ojalearn(x,zeta0,n0,nmax);
      disp('learned')
      w=[w ww];
      y=[y yy];
      lambda=[lambda ll];
      if k<h
           x=ojanext(x,ww);
           disp('next')
      end
end
```

Figure 8: Function `ojapca`