Pre-Processing of Hyperspectral Images Using Nonlinear Filters

V. Behar, V. Bogdanova

Key Words: Hyperspectral imaging; random field simulation; edge-preserving filtration.

Abstract. In this paper the use of a set of nonlinear edge-preserving filters is proposed as a pre-processing stage with the purpose to improve the quality of hyperspectral images before object detection. The capability of each nonlinear filter to improve images, corrupted by spatially and spectrally correlated Gaussian noise, is evaluated in terms of the average Improvement factor in the Peak Signal to Noise Ratio (IPSNR), estimated at the filter output. The simulation results demonstrate that this pre-processing procedure is efficient only in case the spatial and spectral correlation coefficients of noise do not exceed the value of 0.6.

1. Introduction

A significant recent achievement in remote sensing is the development of hyperspectral sensors and the corresponding software used to analyze and interpret the resulting image data [1]. During the past decade hyperspectral image analysis has become one of the most powerful and fastest growing technologies in the field of remote sensing. The hyperspectral sensors collect information as a set of 2D images. Each 2D image represents a range of the electromagnetic spectrum known as a spectral band. These 2D images are then combined, forming a three-dimensional data file (a cube of data), which includes spatial information (2D images) of the object, added with spectral information (1D) of each spatial coordinate. On the other hand, the spectrum received at this point of the object corresponds to each point of the image [2]. Hyperspectral imaging allows by finding objects with their physical and chemical structure to identify a specific structure or a condition of the vegetable cover, to define the geological structure, to reveal the chemical composition of water. The hyperspectral sensors are used in the following areas:

• In agriculture: spatial distribution of different types of land, assessment of the condition of crops, areas of plants distribution (including weed and narcotic-containing), violations of the vegetable and soil cover, definition of types of the damp and salted soils, etc. [3,4].
• In forestry: assessment of the condition of a forest stand and areas of distribution of wreckers, drawing maps of distribution of breeds of trees, determination of the quantitative parameters of stocks of wood and reservoirs condition, identification of minerals, detection of temperature anomalies [5].
• In ecology: identification and localization of the defects of oil and gas pipelines, detection of an oil film, etc. [6].

In this paper we study the possibility to use nonlinear filters at the pre-processing stage in order to improve SNR of hyperspectral images before object detection. Several well known nonlinear edge preserving filters, such as SNN, Kuwahara (three versions), Cubic, Rational, Sticks, Anisotropic Diffusion, Nonlinear Gaussian are studied for the aim of pre-processing of hyperspectral images [9-15]. The efficiency of nonlinear filtration is evaluated in terms of the average Improvement in the Peak SNR (PSNR) estimated at the filter output.

2. Signal Model

In accordance with [8] we assume that the noise in a recorded 3D hyperspectral image is additive and signal independent. We assume that a 3D hyperspectral image can be considered as a set of 2D images recorded in K spectral band. The process of formation of a 3D hyperspectral image can be written as

\[ P(k) = Y(k) = F(k) + N(k), \quad k=1,...,K, \]

where \( Y(k) \) is the intensity of the 2D signal received in the \( k \)-th spectral band (spectral channel), \( F(k) \) is the reflectance of the object in the \( k \)-th spectral band, \( N(k) \) is a zero-mean 2D Gaussian process independent of \( F \), which is stationary and correlated along both \( X \) and \( Y \) dimensions. Under this assumption, the intensity of a hyperspectral image can be modeled as a 3D random process with Rice distribution:
(2) \[ P(i,j,k) = [f(i,j,k) + n_i(i,j,k)]^2 + [f(i,j,k) + n_i(i,j,k)]^2 \]

In (2) \( P(i,j,k) \) is the intensity of a 2D image at pixel position \((i, j)\) recorded in the \(k\)-th spectral band and \(f(i,j,k)\) is the \(k\)-th component of the spectral reflectance at pixel position \((i, j)\). The terms \(n_i\) and \(n_i\) are zero-mean Gaussian processes independent of \(i, j\) at pixel position \((i, j)\) before and after the image filtering. If the filtering improves the image quality before and after the image filtering. If the filtering improves the SNR of the images while preserving important image elements. The present study attempts to achieve this aim by testing several variants of nonlinear edge-preserving filters, which have been designated in literature as the most appropriate ones for image noise reduction.

3. Image Processing

As shown in Figure 1, the image processing investigated in this study includes the following stages: noisy image formation according to (1) and (2); image filtration using a set of nonlinear filters and PSNR estimation.

\[
\text{Matrix of reflections} \rightarrow \text{Matrix of Gaussian noise} \rightarrow \text{Image Formation} \rightarrow \text{Image Filtering} \rightarrow \text{PSNR Estimation} \rightarrow \text{Average IPSNR Estimation}
\]

**Figure 1.** Block-scheme of image processing

The Peak Signal-to-Noise Ratio (PSNR) is used to measure the image quality before and after filtering. This quality measure is evaluated as the ratio between the maximum possible pixel power and the power of the corrupting noise. Usually PSNR is expressed in terms of the logarithmic decibel scale

\[
\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right)
\]

In (3) \( \text{MAX}_I \) is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per a sample, this is 255. According to (1), given a noise-free \( I \times J \) monochromatic image \( F \) and its noisy version \( P \), MSE (Mean Square Error) is defined as

\[
\text{MSE} = 10 \log_{10} \left( \frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} [P(i,j) - F(i,j)]^2 \right)
\]

Since a higher PSNR at the filter output generally indicates filtration of higher quality, the quality of image filtration can be evaluated in terms of the Improvement in PSNR at the filter output (IPSNR)

\[
\text{IPSNR} = \text{PSNR}_{\text{out}} - \text{PSNR}_{\text{in}}
\]

Here \( \text{PSNR}_{\text{in}} \) and \( \text{PSNR}_{\text{out}} \) are PSNR values evaluated before and after the image filtering. If the filtering improves the image quality, IPSNR accepts positive values.

4. Filter Description

Unlike linear filtering, nonlinear filtering may increase the SNR of the images while preserving important image elements. The present study attempts to achieve this aim by testing several variants of nonlinear edge-preserving filters, which have been designated in literature as the most appropriate ones for image noise reduction.

4.1. Symmetric-Nearest-Neighbourhood (SNN) Filter

The kernel of this filter is divided into sub-windows [8]. All sub-windows must include the central pixel of the kernel, which, however, is not accounted in processing. The pixel intensity, which is closest to the intensity of the central pixel, is extracted from each sub-window. The intensity of the central pixel of the output image is determined as the average of the intensities of the pixels extracted from each sub-window. The algorithm of the filter is given in Table 1.

The values of C1, C2, C3 and C4 are determined as follows:

\[
C1 = \begin{cases} P1, & \text{if } |P5 - P1| < |P5 - P9| \\ P9, & \text{otherwise} \end{cases}
\]

\[
C2 = \begin{cases} P2, & \text{if } |P5 - P2| < |P5 - P8| \\ P8, & \text{otherwise} \end{cases}
\]

\[
C3 = \begin{cases} P3, & \text{if } |P5 - P3| < |P5 - P7| \\ P7, & \text{otherwise} \end{cases}
\]

\[
C4 = \begin{cases} P4, & \text{if } |P5 - P4| < |P5 - P6| \\ P6, & \text{otherwise} \end{cases}
\]

4.2. Kuwahara-Nagao Filters

The kernel of this filter is divided into four sub-windows \((S_1, S_2, S_3, S_4)\), each of them including a central pixel of the filter kernel [8]. Each sub-window is used to estimate the variance of the intensity \(\sigma_i^2, \sigma_j^2, \sigma_k^2, \sigma_\gamma^2\) and \(\sigma_v^2\), and next to find a sub-window with the minimal variance \(S_{\text{min}}\), where the average intensity \(m(S_{\text{min}})\) is determined. The following three modifications of Kuwahara filter are used in practice:

- **Kuwahara-1.** The intensity of the central pixel in the window of the output image is replaced by \(m(S_{\text{min}})\);
- **Kuwahara-2.** The intensity of the central pixel in the window of the output image is replaced by the pixel intensity from the sub-window \(S_{\text{min}}\) with the smallest difference from \(m(S_{\text{min}})\);
- **Kuwahara-3.** The intensity of the central pixel in the window of the output image is replaced by the pixel intensity...
from the sub-window $S_{\text{min}}$ with the largest difference from $m$ ($S_{\text{min}}$).

In Table 2 three Kuwahara filters are presented with a kernel window of size 3×3. Four sub-windows are determined in the kernel window of the filter: $S_1=\{P1, P2, P4, P5\}$; $S_2=\{P2, P3, P5, P6\}$; $S_3=\{P4, P5, P7, P8\}$ and $S_4=\{P5, P6, P8, P9\}$. Assuming that the sub-window $S_4$ has the smallest variance of the pixel intensity, i.e., $S_4 = S_{\text{min}}$, the intensity of the central pixel in the window of the output image accepts values according to the algorithms in Table 2.

### 4.3. Rational Filter

According to [9], the kernel window of this filter is divided into sub-windows in a manner shown in Table 3.

The intensity of the central pixel in the window of the output image is formed as

\[(P5)_F = \alpha \cdot P5 + Z\]

where

\[\alpha = \frac{2w}{w \cdot k(P2-P8)^2 + 1} + \frac{2w}{w \cdot k(P4-P6)^2 + 1} + \frac{2w}{w \cdot k(P1-P9)^2 + \sqrt{2}} + \frac{2w}{w \cdot k(P3-P7)^2 + \sqrt{2}}\]

and $w = 0.16; k = 0.01$

### 4.4. Cubic Filter

According to [10], in the kernel window of this filter a sub-window is defined as shown in Table 4.

The kernel sub-window includes four pixels P2, P4, P6, P8 and P5. The intensity of the central pixel in the window of the output image is formed as

\[(P5)_F = P5 + \lambda \cdot (P2 - 2\cdot r_2 / 3 + r_6 / 12 + r_8 / 3 + r_9 / 12 - r_5 / 12 - - r_7 / 12 - r_9 / 3 + r_6 / 6),\]

where

\[r_1 = P5; r_2 = P2; r_3 = P(P2); r_4 = P(P5); r_5 = P8(P2); r_6 = P6(P2); r_7 = P4(P2); r_8 = P2P5P8; r_9 = P2P4P8.\]

The parameter $\lambda$ in (8) controls the level of smoothing in the filtered image.

### 4.5. Nonlinear Gaussian Filter

According to [11] the algorithm for nonlinear Gaussian filtration of the image $P(p)$ can be described as:
are the divergence information technologies and control

nonlinear Gaussian filter range \([1, 1.5]\). In general, the smoothing effect of a single

\(10\)

is the filter parameter and

\(13\)

where

\(16\)

with symmetric boundary conditions

\(d_{i-1,j}^n = d_{i,j}^n; d_{M,j}^n = d_{M-1,j}^n; j = 0, 1, 1, ..., N - 1\)

\(d_{i,j}^n = d_{i,0}^n; d_{i,N}^n = d_{i,N-1}^n; i = 0, 1, 2, ..., M - 1\)

The diffusion coefficients \(c_{i,j}\) are given by

\(15\)

where

\(\nabla^n_r P_{i,j}^n = \left[(P_{i+1,j}^n - P_{i,j}^n), (P_{i+1,j+1}^n - P_{i,j+1}^n)ight]\)

\(\nabla^n_l P_{i,j}^n = \left[(P_{i-1,j}^n - P_{i,j}^n), (P_{i,j-1}^n - P_{i,j}^n)\right]\)

\(\nabla^n P_{i,j}^n = P_{i+1,j}^n + P_{i-1,j}^n + P_{i,j+1}^n + P_{i,j-1}^n - 4P_{i,j}^n\),

with symmetric conditions:

\(P_{i,j}^n = P_{i,j}^0; P_{M,j}^n = P_{M-1,j}^0; j = 0, 1, 1, ..., N - 1\)

\(P_{i,j}^n = P_{i,j}^0; P_{i,N}^n = P_{i,N-1}^0; i = 0, 1, 2, ..., M - 1\)

The diffusion coefficients \(c_{i,j}\) in (12) are obtained as a result of discretization of the diffusion function \(c(q)\), which can be chosen as

\(16\)

\(17\)

In (16) and (17) \(q_i(t)\) is the scale function and \(q(x, y, t)\) is the instantaneous coefficient of variation determined by

\(18\)

where \(P(n, m)\) is the intensity at pixel (\(n, m\)). The instantaneous coefficient of variation \(q(n, m, t)\) serves as an edge detector in the noise image. This function exhibits high values at edges or on high-contrast features and produces low values in homogeneous regions. The scale function \(q_i(t)\) efficiently controls the amount of smoothing applied to the image. It is estimated as:

\(19\)

\(\begin{array}{c}
\text{Table 4. Cubic filter with a kernel of size 3x3} \\
\end{array}\)

<table>
<thead>
<tr>
<th>Filter kernel</th>
<th>Sub-window</th>
<th>Filter algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>P2</td>
<td>P3</td>
</tr>
<tr>
<td>P4</td>
<td>P5</td>
<td>P6</td>
</tr>
<tr>
<td>P7</td>
<td>P8</td>
<td>P9</td>
</tr>
</tbody>
</table>

\begin{align*}
\text{(P5)} &= P5 + \lambda t(-2r1/3 + r2/6 - r3/12 + r4/3 + r5/12 - r6/12 - r7/12 - r8/3 + r9/6) \\
\end{align*}
where \( \text{var}[z(t)] \) and \( z(t) \) are the intensity variance and mean value over a homogeneous area at \( t \), respectively. The scale function \( q_0(t) \) can be approximated by
\[
(20) \quad q_0(t) = d_0 \exp(-\rho t)
\]
where \( \rho \) is a constant and \( d_0 \) is the coefficient of noise variation in the observed image.

### 4.7. Sticks Filter

The pixel intensity of the filtered image is formed as the maximum value of \( “2n - 2” \) pre-filtered images. The kernel of these sub-filters \( s_i \) has the form of a stick, which includes \( n \) pixels with angular orientation \( \theta \) [14]. A set of these sub-filter kernels is \( \mathcal{S} = \{s_{ji}, i = 1, \ldots, 2n - 2 \}. \) In this study we used a filter with 8 sub-filters, the kernels of which include 5 pixels, and the stick thickness is \( k = 1 \) (figure 2).

After pre-filtering of the input image the following \( (2n - 2) \) images are formed:
\[
(21) \quad P_x(x, y) = \max\{F_i(x, y)\}, i = 1 \ldots 2n - 2.
\]

The final filtered image is formed as
\[
(22) \quad P_f(x, y) = \max \{P_i(x, y)\}, i = 1 \ldots 2n - 2.
\]

### 5. Numerical Results

In this study we tested the possibility of the usage of nonlinear filtration as pre-processing of hyperspectral images. For this aim, seven 2D images (obtained at seven different spectral bands) of the same object recorded by a spectrometer of high quality are used as reflections matrices \( F_1, F_2, \ldots, F_7 \). For each reflection matrix \( F_i \) \( (k = 1 \ldots 7) \), the noise matrix \( N_i \) is simulated as a stationary and correlated 2D Gaussian process, as a result of which seven noisy images \( P_1, P_2, \ldots, P_7 \) are formed in accordance with (2). In simulation of noise, the noise correlation coefficients \( \rho_X \) (along X-axis), \( \rho_Y \) (along Y-axis) and \( \rho_\theta \) (along numbers of spectral bands) vary within the range from 0 up to 0.8. For the sake of simplicity, we consider the variant when \( \rho_X = \rho_Y = \rho_\theta = \rho. \) In noise simulation, the noise variances \( \sigma^2_x \) (along X-axis), \( \sigma^2_y \) (along Y-axis) and \( \sigma^2_\theta \) (along numbers of spectral bands) are determined as a function of the SNR value. For simplicity we have assumed that \( \sigma^2_x = \sigma^2_y = \sigma^2_\theta = \sigma^2 \), which are determined according to a given SNR value in decibels:
\[
(23) \quad \sigma = \sum_{i} \sum_{j} \sum_{k} F_{i,j,k} / SNR \cdot J \cdot K, \text{ where } SNR = 10^{\text{SNR}_{dB}/20}.
\]

In (23) \( (i \times J) \) is the image size in the \( k \)-th spectral band and \( K \) is the number of spectral bands. All noisy images are simulated by algorithm (2) and then they are filtered by a set of different nonlinear filters (SNN, Kuwahara-1,2,3, Rational, Cubic, Gaussian, AAD in two variants and Sticks). In tables 5-7 the values of the average improvement factor \( IPSNR \) are presented, evaluated with respect to the filter type and noise parameters (SNR in dB and \( \rho \)).

### Table 5. Average improvement factor depending on the correlation coefficient \( \rho \) (SNR=0dB)

<table>
<thead>
<tr>
<th>Filter</th>
<th>( \rho )</th>
<th>0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>1.8837</td>
<td>1.3950</td>
<td>0.8950</td>
<td>0.4886</td>
<td>0.2169</td>
<td></td>
</tr>
<tr>
<td>AADm</td>
<td>1.7450</td>
<td>1.2760</td>
<td>0.7956</td>
<td>0.4037</td>
<td>0.1462</td>
<td></td>
</tr>
<tr>
<td>AADY</td>
<td>1.7719</td>
<td>1.3498</td>
<td>0.9006</td>
<td>0.5063</td>
<td>0.2137</td>
<td></td>
</tr>
<tr>
<td>Rat</td>
<td>1.7283</td>
<td>1.3540</td>
<td>0.9583</td>
<td>0.5947</td>
<td>0.2863</td>
<td></td>
</tr>
<tr>
<td>Sticks</td>
<td>-1.5390</td>
<td>-1.5475</td>
<td>-1.3953</td>
<td>-1.0803</td>
<td>-0.6579</td>
<td></td>
</tr>
<tr>
<td>CUB</td>
<td>5.0235</td>
<td>5.2973</td>
<td>4.6684</td>
<td>2.7153</td>
<td>-1.9313</td>
<td></td>
</tr>
<tr>
<td>Kuw1</td>
<td>2.2021</td>
<td>1.6357</td>
<td>0.9972</td>
<td>0.4478</td>
<td>0.1388</td>
<td></td>
</tr>
<tr>
<td>Kuw2</td>
<td>2.1521</td>
<td>1.6011</td>
<td>0.9971</td>
<td>0.4390</td>
<td>0.1364</td>
<td></td>
</tr>
<tr>
<td>Kuw3</td>
<td>1.3613</td>
<td>0.9937</td>
<td>0.5795</td>
<td>0.2517</td>
<td>0.0774</td>
<td></td>
</tr>
<tr>
<td>SNN</td>
<td>2.3329</td>
<td>1.6753</td>
<td>0.9763</td>
<td>0.4337</td>
<td>0.1231</td>
<td></td>
</tr>
</tbody>
</table>

### Table 6. Average improvement factor depending on \( \rho \) and SNR

| Average improvement factor in dB: SNR = 5 |
|------------------------|------------------|------------------|------------------|
| Filter | \( \rho \) | 0   | 0.2 | 0.4 | 0.6 | 0.8  |
| Gaussian | 1.3113 | 0.8313 | 0.5140 | 0.3089 | 0.1724 |
| AADm    | 1.3248 | 0.8038 | 0.4599 | 0.2428 | 0.1070 |
| AADY    | 1.0995 | 0.7660 | 0.5185 | 0.3237 | 0.1671 |
| Rat     | 0.7932 | 0.6676 | 0.5487 | 0.4033 | 0.2363 |
| Sticks  | -2.333 | -1.9849 | -1.4913 | -1.0024 | -0.5828 |
| CUB     | 3.9786 | 6.1312 | 6.1735 | 4.2189 | -0.5962 |
| Kuw1    | 1.4019 | 0.8997 | 0.5137 | 0.2410 | 0.1036 |
| Kuw2    | 1.3403 | 0.8661 | 0.4977 | 0.2376 | 0.1027 |
| Kuw3    | 0.8856 | 0.5429 | 0.3052 | 0.1473 | 0.0857 |
| SNN     | 1.5504 | 0.9320 | 0.5183 | 0.2372 | 0.0860 |

### Table 7. Average improvement factor depending on \( \rho \) and SNR

| Average improvement factor in dB: SNR = 10 |
|------------------------|------------------|------------------|------------------|
| Filter | \( \rho \) | 0   | 0.2 | 0.4 | 0.6 | 0.8  |
| Gaussian | 0.4638 | 0.3695 | 0.3057 | 0.2353 | 0.1568 |
| AADm    | 0.5865 | 0.3972 | 0.2749 | 0.1779 | 0.0933 |
| AADY    | 0.1242 | 0.2837 | 0.3064 | 0.2486 | 0.1506 |
| Rat     | -0.3966 | 0.1278 | 0.3274 | 0.3252 | 0.2186 |
| Sticks  | -3.2519 | -2.2991 | -1.5181 | -0.9609 | -0.5511 |
| CUB     | 2.4146 | 6.3081 | 6.8855 | 4.8139 | -0.0017 |
| Kuw1    | 0.2713 | 0.3402 | 0.2799 | 0.1858 | 0.0973 |
| Kuw2    | 0.2017 | 0.3111 | 0.2674 | 0.1811 | 0.0964 |
| Kuw3    | 0.2230 | 0.2315 | 0.1821 | 0.1200 | 0.0644 |
| SNN     | 0.4786 | 0.4028 | 0.3043 | 0.1778 | 0.0787 |
The value of PSNR is evaluated at each filter output for each spectral band. The average improvement factor is

\[
IPSNR_{ave} = \frac{1}{K} \sum_{k=1}^{K} PSNR_{k}(k) - \frac{1}{K} \sum_{k=1}^{K} PSNR(k).
\]

Analyzing the numerical results shows that in general the efficiency of the nonlinear filtration as a preprocessing procedure of hyperspectral images depends strongly on the noise parameters – the noise intensity and correlation properties. It can be seen that the average improvement factor of all nonlinear filters decreases with increase of both SNR and \(q\). The numerical results show that all filters considered, except for Sticks filter, improve the quality of hyperspectral images. However, the Cubic filter is the most efficient among them when the noise correlation coefficients do not exceed the value of 0.6 (given in bold fonts in the tables). In this case the usage of the Cubic filter allows improvement of the PSNR of hyperspectral images by 3-6 dB.

Conclusions

It is shown that nonlinear filtration can be used as a preprocessing step which can improve the SNR of hyperspectral images before their further processing. This preprocessing procedure is effective only in case the spatial and spectral correlation coefficients of noise do not exceed the value of 0.6. The next step of implementation is verification of the explored algorithms over a data set received by a laboratory hyperspectral camera installation.

Acknowledgment

The research work reported in the paper is partly supported by Project AComIn “Advanced Computing for Innovation”, Grant 316087, funded by FP7 Capacity Programme (Research Potential of Convergence Regions) and by Project No DFNI – 101/8, funded by the Bulgarian Science Fund.

References


Manuscript received on 15.10.2013

Vera Petrowna Behar was born in Bologoe, Russia. She received the M.S. degree in Applied Mathematics in the Saint-Petersburg State Institute of Fine Mechanics and Optics (LITMO), and the Ph.D degree in radar signal processing in the Central Laboratory for Parallel Data Processing, Sofia, Bulgaria. Up to date she is an Assoc. Prof. in radar signal processing and works in the Institute of Information & Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria. Her research experience is in signal & image processing, CFAR detection, adaptive array processing, Forward Scattering radar imaging, medical ultrasound imaging; signal transform (Hough, Wavelet, SFT) and Monte Carlo simulation. At the moment she works in the field of research related to adaptive array processing for navigation and sonar receivers, GPS signal processing and hyperspectral image processing.

Contacts:
Institute of Information and Communication Technologies, BAS
25A Acad. G. Bonchev St., 1113 Sofia, Bulgaria
e-mail: behar@bas.bg

Violeta Bogdanova is born in Sofia. Her University Degree is Engineer of Automation, High Chemistry and Technology Institute, Sofia, Bulgaria, 1976. She an Assistant Professor at the Bulgarian Academy of Sciences, IICT, Sofia, Bulgaria. Her current research interests are in the area of signal and image processing.

Contacts:
Institute of Information and Communication Technologies, BAS
25A Acad. G. Bonchev St., 1113 Sofia, Bulgaria
e-mail: vbogdan@bas.bg