14

Compression of Spectral Images

Arto Kaarna
Lappeenranta University of Technology
Finland

1. Introduction

In this chapter we describe methods how to compress spectral imaging data. Normally the spectral data is presented as spectral images which can be considered as generalizations of colour images. Rapid technological development in spectral imaging devices has initiated the need for the compression of raw data. Spectral imaging has been central to many remote sensing applications like geology and environment monitoring. Nowadays, new application areas have arisen in industry, for example in the quality control of assembly line products and in applications, where the traditional three-chromaticity colour measurements are not accurate enough. Spectral imaging produces large amounts of raw data which will be processed later in various applications. Image compression provides a possibility to reduce the amount of raw data for storing and transmission purposes. The image compression can be either lossless or lossy. In the lossy compression the quality of the reconstructed data should be estimated to evaluate the usefullness of the reconstructed data. The lossy compression is justified in the sense that the compression ratios are much higher than in the lossless case where the reconstructed data is identical to the raw data.

Spectral images are now available for different applications due to the development in the spectral imaging systems (Hauta-Kasari et al., 1999; Hyvärinen et al., 1998). Geoscience and remote sensing have been the main application areas of spectral images but nowadays several new application areas have emerged in industry. For example, applications in quality control, exact colour measurement, and colour reproduction use spectral information, since RGB colour information only is not sufficient.

Image compression has been one of the main research topics in image processing. The compression methods are usually developed for images visible to humans, i.e. for grey-scale or RGB colour images. Applications in the field of remote sensing and recent advances in industrial applications, however require the compression of spectral images (Vaughn & Wilkinson, 1995). Some compression methods are lossless (Memon et al., 1994; Roger & Cavenor, 1996), but most of the methods are lossy (Abousleman et al., 1997; Gelli & Poggi, 1999). Some applications can accept data which is compressed by a lossy scheme, but naturally the important features in the data must be present. If the lossy compression method cancels out any of the important features for the applications, then the lossless compression is the only possibility to decrease the amount of the raw data.

Compression is required due to the large amounts of data captured in the images. Regular digital cameras in everyday use apply JPEG or TIFF-compression. Images displayed in web-

Source: Vision Systems: Segmentation and Pattern Recognition, ISBN 987-3-902613-05-9,
Edited by: Goro Obinata and Ashish Dutta, pp.546, I-Tech, Vienna, Austria, June 2007

pages are compressed with the same methods. Compression in these applications is accepted as a normal procedure as long as the visual quality is not reduced.
With spectral images the memory or the transmission requirements are very high. Observations of Earth in spatial, spectral, temporal and radiometric methods produce data volume which is growing faster than the transmission bandwidth (Abousleman et al., 2002; Aiazzi et al., 2001). This means, that for long term storing or transmission, these databases should be compressed. The compression should be such that the spatial and spectral quality of the reconstructed image is high enough for the application. Table 1 shows examples of spectral imaging systems developed for remote sensing (Kerekes &Baum, 2002; Lillesand & Kiefer, 2000; AVIRIS, 2006; HyMap, 2006; HYDICE, 2006; Landsat, 2006; Hyperion 2006; Ikonos, 2006; OrbView, 2006; Aisa Eagle, 2006).

Name	# of channels	Spatial resolution, m	Radiometric resolution, bits	Raw data: kB/km²
Airborne				
M7	12	10	8	120
AVIRIS	224	20	12	840
HYDICE	210	3	12	35000
HyMap	200	2	16	100000
Aisa Eagle	244	0.5	12	1400000
Spaceborne				
ERTS/MSS	4	80	8	0.6
Landsat/TM	7	30	8	7.8
Hyperion/EO-1	220	30	12	366.7
IKONOS	4/1	4/1	11	1719
OrbView-5	4/1	1.64/0.41	11	2045
OrbView-4	200	8	8	31250

Table 1. Examples of remote sensing systems. The spatial resolution for airborne sensors depends on the flight altitude. kB means kilobytes.

As an example, one spectral Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) (AVIRIS, 2006) tape, taken in one day, can have up to 16 GB of raw data. Large amounts of data are also recorded in an application for the quality control of ceramic tiles (Kälviäinen et al., 1998): imaging of 25 ceramic tiles made up a spectral database of size 312 megabytes. Nowadays, there are several conferences where new spectral imaging systems for industrial applications are presented (MCS, 2006; EI, 2006; IGARSS, 2006).
When the client's application is known in advance the data for it can be extracted from the original database. For example, in mineral mapping the spectral range from around 2.0μm to 2.5μm is sufficient. Infrared systems utilize also a narrow band above the visual range for example in night time vision systems. If the colour features are enough, one can extract 30 bands out of 224 from the AVIRIS images for that specific application. In all previous cases and for various client requirements, the high quality database or even the original database must be present for the data extraction.

As the imaging systems have developed, at the same time the resources for storing the images are advanced due to the technological changes. In Table 2 we show some development features in hard drive technologies and properties (Thompson & Best, 2000; Hughes, 2002; Grochowski & Halem, 2003; Moreiro, 2006).

Feature	1970	1980	1990	2000	2009
Density, Mb/cm^2	$1*10^0$	$5*10^1$	$3*10^2$	$4*10^4$	$5*10^5$
Internal data rate, Mb/s	0.8	2	4	50	200
Capacity, GB	0.03	0.3	1	100	1600
Price, $/MB	NA	200	8	0.05	<0.002

Table 2. Advances in hard drive features. NA stands for information Not Available.

A similar growth pace as for the hard disk drives is experienced also in digital transmission both in wired and wireless cases: in average, every fifteen years the capacities have become thousand-fold.

The spectral imaging systems produce a vector of values for each pixel of the image. The values depend on the resolution of the imaging system and they are normally presented as 8 bit, 12 bit or 16 bit values. Thus, a spectral image can be considered as a set of two-dimensional, equal size images. Now, compression methods can be similar to the methods applied to greyscale images or to RGB-colour images. For lossless compression also regular text compression methods can be applied. This simple approach may be usable, when a) the original image should be perfectly reconstructed, b) the compression method should be widely available, and c) high compression ratios are not required. These methods include entropy modelling followed by Huffman coding, arithmetic coding or Burrows-Wheeler transform. The standard Unix tool, gzip, is based on Lempel-Ziv coding (Ziv & Lempel, 1977). It gives the average lossless compression ratio 1.41 for a set of four Moffet Field scenes and 1.39 for a set of five Jasper Ridge scenes from the AVIRIS dataset (AVIRIS, 2006). Much better lossless compression ratios are received if the composition of the spectral images is observed. Best lossless compression methods are most often based on predictive coding combined with entropy modeling (Aiazzi et al., 2002; Aiazzi et al., 2001; Aiazzi et al., 1999; Benazza-Benyahia et al., 2001; Mielikäinen & Toivanen, 2003; Mielikäinen, 2006). Also integer transforms (Kaarna, 2001) or lossless vector quantization (Ryan & Arnold, 1997-1) is possible for the perfect reconstruction.

A lossy compression procedure for spectral images consists of three phases. The first phase decorrelates the raw data in spatial and spectral dimensions, the second phase quantizes the coefficients from the first phase. The third phase utilizes some lossless scheme to encode the quantized coefficients. This procedure is depicted in Fig. 1.

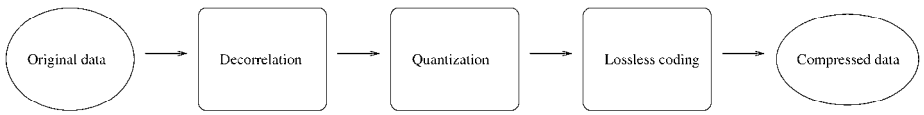


Fig. 1. The three phases in the lossy compression.

A compression procedure is of any practical interest only if it has an inverse procedure which reconstructs the original data or image. An inverse procedure includes the same phases as the compression procedure in Fig. 1, but they are processed in reverse order. First, the compressed data is decoded resulting in the quantized coefficients. Then the quantized coefficients are restored to their original values and these values compose the original data. In Fig. 2 the decompression, the inverse procedure for compression, is depicted.

Fig. 2. Decompression, the inverse procedure for compression.

Different methods and their parameter values are possible in the compression procedure. Depending on the selections the reconstructed data can be equal to the original data or some information can be lost. The methods and parameters are selected such that the important features of the image are present and the information lost can be regarded as an observation noise or otherwise irrelevant for the application. The evaluation of the quality of the reconstructed data is necessary when using lossy compression. The quality measurements are most often based on the pixelwise or bandwise difference between the original image and the reconstructed image resulting to logarithmic signal-to-noise ratio (Rabbani & Jones, 1991). Specific measures for spectral images include both the percentage maximum absolute distortion measure (PMAD) (Ryan & Arnold, 1997-2; Ryan & Arnold, 1998) and the blockwise distortion measure for multispectral images BDMM (Kaarna & Parkkinen, 2002). PMAD guarantees that every value in the reconstructed image is within a maximum distance from the original value. The maximum distance is relative to the original value. BDMM correlates blockwise filtering of the original and the reconstructed image to the visual quality of the distorted images.
In the following sections we introduce methods for lossy and lossless compression of spectral images. Then we describe how to evaluate the quality of reconstructed data in the lossy case. Finally, we show experimental results and evaluate different compression methods.

2. Lossy compression of spectral images

A comprehensive study on theoretic aspects of lossy source coding can be found from (Berger & Gibson, 1998). Data compression is thoroughly considered in (Donoho et al., 1998). Both scalar and vector quantization is widely surveyed in (Gray & Neuhoff, 1998). The wavelet transform is described in detail in (Daubechies, 1992; Taubman & Marcellin, 2002).
Lossy compression methods achieve remarkably higher compression ratios than lossless compression by neglecting some unessential data in the compression phase. Several lossy compression methods have been developed for the compression of spectral images. Some of them are two-dimensional methods applied separately to each band of the spectral image

(Abousleman et al., 1994), and some methods have been further enhanced from the two-dimensional methods to be three-dimensional (Abousleman et al., 1997; Kaarna & Parkkinen 1999). Most of the recent methods apply separate subtasks to the spectral and spatial dimensions due to their dissimilar characteristics (JPEG2000, 2006; Aware, 2006; Kaarna et al., 2000; Kaarna et al., 2006).
A rough classification of the compression methods for the spectral images include the principal component analysis (PCA) for the decorrelation of the spectral data, the wavelet transform for the spatial compression of images, predictive methods applied simultaneously to the spectral and spatial dimensions of the image, and finally the vector quantization of the spectra in the image. Each of these methods can alone compress the image, but in practise, best compression results are obtained through combining these methods.

2.1 Vector quantization

Clustering is an unsupervised method to classify patterns in an image. Patterns within a cluster are more similar to eachother than they are to a pattern belonging to another cluster. Thus, a lossy compression method can be established on that notation: each member of a cluster are represented by the cluster center. The compressed data consists of cluster centers and an index image.
Vector quantization utilizies the previous idea (Ryan & Arnold, 1997-1; Ryan & Arnold, 1997-2). First, a decomposition of the image into a set of vectors is performed. With spectral images the decomposition naturally consists of the spectral vectors. Then a codebook is generated from a training set of vectors using an iterative algorithm. Finally, each spectral vector of the image is quantized to the closest vector in the codebook according to the selected distortion measure. The compressed data consists of a codebook and a set of indices to the codebook. One index is required for each spectrum of the image.
The generalized Lloyd algorithm (GLA) tries to optimize the codebook C. The algoritmic presentation of the GLA is :

Algorithm 1:

Step 1: Select the initial codebook C_1, set $m=1$.
Step 2: With the given codebook C_m perform one iteration to generate an improved codebook C_{m+1}.
Step 3: Compute the average distortions for C_{m+1}. If the change from the previous iteration is small enough, then stop.
Otherwise set $m = m+1$ and continue from Step 2.

The Step 2 of Algorithm 1 is generally implemented using a Nearest Neighbor condition:
Algorithm 2:
Step 1: Using the codebook $C_m = y_i$ partition the training set T into clusters R_i with the NN condition: $R_i = \{ x \in T : d(x,y_i) \leq d(x,y_j);\ all\ j \neq i \}$.
Step 2: Compute the centroids for the clusters $\{cent(R_i)\}$ to obtain an improved codebook $C_{m+1} = \{cent(R_i)\}$.

In vector quantization each vector is represented by the centroid of a cluster it belongs to. The resulted data from the VQ consists of the cluster centroids and of an index image, which

describes the inclusion of each vector into one cluster. In Fig. 3. we illustrate the vector quantization for the compression of spectral images.

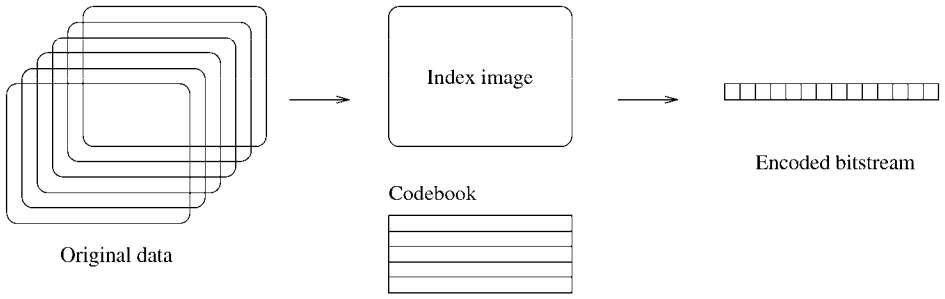


Fig. 3. Vector quantization in lossy compression of spectral images.

Similar approach to vector quantization was defined in (Toivanen et al., 1999). The procedure produced an index image and a codebook which was generated with a Self-Organizing Map (SOM). The results were good compared to a clustering method in (Kaarna et al., 1998).

Even though vector quantization is theoretically optimal in lossy coding, the implementation issues constrain the performance (Poggi & Ragozini, 2002). Computational complexity and memory requirements have been the main drawbacks that have been attacked (Poggi & Ragozini, 2002; Kaarna et al., 2000; Ryan & Arnold, 1997-2; Kamano et al., 2001).

A tree-structured product-codebook was designed to support progressive transmission (Poggi & Ragozini, 2002). In product-codebook VQ, all codewords are of type $x_{ij} = u_i * v_j$, where * is the decomposition rule. The component codebooks were organized in a tree-structure in order to speed up the codeword selection. The optimal design of the components codebooks is a complex task, but a suboptimal solution clearly lowered the computational requirements. A tree-structure was also applied in (Kaarna et al., 2000) to accelarate the look-up functions in clustering. The leaves of the tree consisted of short linear lists and the search operation combined both the tree-structured and linear look-up functions.

An important feature in vector quantization is how to define an appropriate distortion measure for two vectors (Ryan & Arnold, 1997-2). The Euclidian distance between the two vectors X and Y is defined as

$$E = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{1}$$

where x_i and y_i are components of vectors X and Y, respectively. A drawback for the Euclidian distance is that it doesn't account for the various shapes of the vectors. The PMAD distortion measure was developed to guarantee that every pixel $B'(s_1,s_2,\lambda)$ of the compressed and reconstructed image is within a maximum distance of $p*100\%$ from its original value $B(s_1,s_2,\lambda)$, i.e. $(1-p)B(s_1,s_2,\lambda) < B'(s_1,s_2,\lambda) < (1+p)B(s_1,s_2,\lambda)$. Using this distortion measure, lossy compression ratios cr up to $cr=17$ were received with airborne multispectral images.

One large codebook can be replaced with two codebooks (Kamano et al., 2001). The first one, a relative small codebook, with few training sets was generated. The second codebook was generated from the residual data between the original image and the first codebook output. The proposed scheme improved the coding efficiency and reduced the transmission rate according to the numerical experiments.

2.2 Spectral decorrelation with PCA

In image compression, the principal component analysis (PCA) produces optimal results in the sense of the mean-square error reconstruction (Karhunen & Joutsensalo, 1995).
The principal component analysis is based on the covariance matrix $C = E[(x-\mu)(x-\mu)^T]$, $\mu = E[x]$ of the original data. In practical calculations the matrix C is replaced by an estimated $\hat{C}$

$$\hat{C} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu^*)(x_i - \mu^*)^T \tag{2}$$

where x_i is a sample vector and μ^* is the estimated mean vector of the sample set. The sum is over all the n samples of the set. From the estimated $\hat{C}$ the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and the respective eigenvectors $u_1, u_2, \ldots, u_n$ are calculated. Due to the properties of the autocorrelation matrix, the eigenvalues λ_i , $i=1,n$ are all real and nonnegative.
As soon as the eigenvalues λ_i are known and without loss of generality the indexing is such that $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, the reconstruction x^* of x is obtained as

$$x^* = \sum_{i=1}^{p}(x^T u_i)u_i \tag{3}$$

where p, $p < n$ is selected such that the required quality in reconstruction will be achieved. In Fig. 4 the principle of the PCA compression of spectral images is shown.

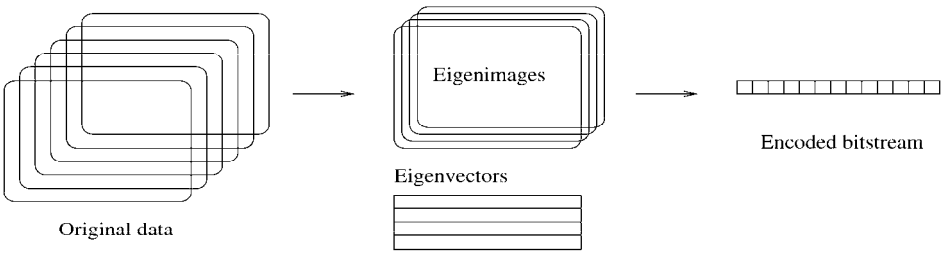


Fig. 4. The PCA in lossy compression of spectral images. In this example, the number p of eigenimages/eigenvectors is $p=4$.

2.3 Transform coding

Function transforms have been used for centuries to solve problems i.e. in mathematics, physics and engineering (Zayed, 1996). For example, in audio signal processing it would be interesting to know what frequencies are included in the measured signal. This problem can be solved using the Fourier transform.

In general, a transform is a mathematical operation where a function or data f in domain u is transformed into another function or data F in domain U: $f \rightarrow F$. The purpose of the transform is normally one of the following:

- After the transform it would be easier to solve the original problem.
- The transformed data gives a new insight to the problem at hand.
- The data in the domain $F(U)$ is measured experimentally and the function f needs to be constructed from this data.

The transforms $f \rightarrow F$ of any practical value has an inverse transform, where the original function f is completely constructed from F, i.e. $F \rightarrow f$. Thus, the transform pair is used to solve the original problem using data F in domain U, and then the solution is transformed back to the data f in domain u.

A popular transform in engineering is the Fourier transform. The transform was developed by Joseph Fourier in 1822 as he demonstrated, that most signals of practical interest can be expanded into a series of sinusoidal functions. Later on, this continuos transform has been developed to be applicable in discrete computations (Proakis & Manolakis, 1994).

The wavelet transform f^w of a function $f(t)$ also provides a time-frequency localization (Chui, 1992; Daubechies, 1988; Daubechies, 1992; Mallat, 1998; Vetterli & Kovačević, 1995) as

$$f^w(a,b) = |a|^{-1/2} \int f(t)\psi\left(\frac{t-b}{a}\right)dt \tag{4}$$

where ψ is called a mother wavelet with zero average, $\int\psi(t)dt = 0$. The mother wavelet ψ(t) is defined as a double-indexed function as

$$\psi^{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right) \tag{5}$$

Practical applications, like the signal compression, require fast implementations. In signal processing community, the wavelet transform is implemented with convolution as an filtering operation and the conjugate mirror filters are used as filter banks. The orthogonal wavelet transform is implemented by the cascading conjugate mirror filters. The perfect reconstruction is achieved with this implementation. The orthonormal bases of wavelets can be constructed using multiresolution analysis.

The fast discrete wavelet transform is computed using perfect reconstruction filter banks. Vetterli showed (Vetterli, 1986), that perfect reconstruction was always possible using FIR-filters. The multiresolution approximation lead to two discrete, finite length filters, and, thus, a filter bank was a solution to a fast implementation.

Using the definition

$$f(t) = \sum_{n=-\infty}^{\infty} a_0[n]\phi(t-n) \in V_0 \tag{6}$$

where $\phi(t)$ a is the scaling function, and due to the properties of multiresolution, $\{\phi(t-n)\}_{n \in Z}$ is orthonormal, then

$$a_0[n] = \langle f(t), \phi(t-n) \rangle \tag{7}$$

The approximation a_{j+1} in the next coarser level of the multiresolution is obtained by

$$a_{j+1}[p] = \sum_{n=-\infty}^{\infty} h[n-2p]a_j[n] \tag{8}$$

and the difference d_{j+1} between the two levels by

$$d_{j+1}[p] = \sum_{n=-\infty}^{\infty} g[n-2p]a_j[n] \tag{9}$$

where $g[n]$ is defined using the discrete filter $h[n]$ as

$$g[n] = (-1)^{1-n} h[1-n] \tag{10}$$

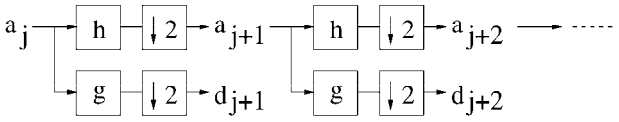
At the reconstruction of the data the coefficients are obtained as

$$a_j[p] = \sum_{n=-\infty}^{\infty} h[p-2n]a_{j+1}[n] + \sum_{n=-\infty}^{\infty} g[p-2n]d_{j+1}[n] \tag{11}$$

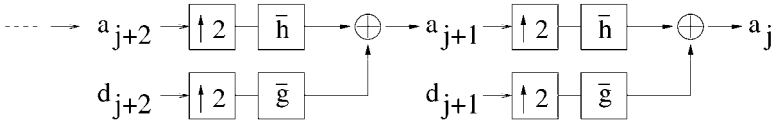
and finally, the discrete values f_d of the original function are recovered from

$$f_d[p] = \sum_{n=-\infty}^{\infty} a_0[n]\phi_d[p-n][n] \tag{12}$$

Since the scaling and the wavelet filters h ang g are finite, the infinite sums in Eqs. 8-12 are computed using the convolution. The discrete wavelet transform is illustrated in Fig. 5: part a) shows the transform and part b) the inverse transform.



a)



b)

Fig. 5. The discrete wavelet transform: a) the transform, b) the inverse transform.

The transform coefficients are the values of a_{j+N} and d_{j+i}, $i=1,...,N$. Downsampling by two (↓2) is performed in the transform and upsampling by two (↑2) in the inverse transform. In practice, Eq. 12 is not used and the values for a_0 are obtained directly as discretized values $f[n]$ of $f(t)$. Due to the perfect reconstruction property, the inverse transform returns the discretized values $f[n]$ directly as coefficients a_0.

The orthogonal, compactly supported wavelets described above has several enhancements. They include biorthogonal wavelets (Cohen et al., 1992), which allow symmetric wavelets.

This was a result of a modification in the multiresolution approximation. Another modification in the multiresolution lead to the wavelet packet analysis (Coifman & Wickerhauser, 1992). Wavelet packets are described as a full transform of the coefficients. Since in the original transform only the coefficients a_j are transformed, see Fig. 5, then in the wavelet packet analysis also the coefficients d_j are transformed in a similar way.

2.4 Linear and non-linear compression

The principal component analysis (PCA) is a widely used statistical technique in pattern recognition, image processing, and signal processing. PCA is an optimal solution in minimization of the mean-square representation error $E\{||x - x^*||^2\}$ where the data x is approximated using a lower dimensional linear subspace x^* (Karhunen & Joutsensalo, 1995). The principal component analysis provides orthonormal basis functions that optimally decorrelate the data. Other methods, like DCT or wavelets, approximate this optimal decorrelation. The justification for the wavelet transform in signal compression comes from the nonlinear approximation (Daubechies, 1998; Donoho et al., 1998; Devore et al., 1992), where the linear combination of N basis functions is used instead of the first N basis functions. In the linear approximation, the space S_n spanned by the first N basis functions $\boldsymbol{\Phi}_n$ is

$$S_n = \left\{ \sum_{n=1}^{N} c_n \boldsymbol{\Phi}_n ; c_n \in C \right\} \tag{13}$$

and in the nonlinear approximation the space S_n is

$$S_n = \left\{ \sum_{n} c_n \boldsymbol{\Phi}_n ; c_n \in C, \#\{n, c_n \neq 0\} \leq N \right\} \tag{14}$$

In nonlinear approximation the wavelet coefficients are ordered according to their significance and the most significant coefficients and their addressing are included in the bit stream.

2.5 Nonlinear compression through the three-dimensional wavelet transform

In two-dimensional case the construction of the wavelet transform starts from a tensor product of two one-dimensional multiresolution analyses (Daubechies, 1992; Mallat, 1989), $\boldsymbol{V_0} = V_0 \otimes V_0$, where V_j, $j \in Z$ is a multiresolution of $L^2(R)$. The multiresolution ladder is similar to that of one-dimensional case,, and now the multiresolution is

$$\begin{aligned} &(1) \quad \ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots \\ &(2) \quad V_0 = V_0 \otimes V_0 \\ &(3) \quad F \in V_j \Leftrightarrow F(2^j \cdot, 2^j \cdot) \in V_0, F(x_1, x_2) = f(x_1) f(x_2), f, g \in V_0 \end{aligned} \tag{15}$$

and the product

$$\boldsymbol{\Phi}_{0,n,m}(x_1, x_2) = \phi_{0,n}(x_1) \phi_{o,m}(x_2) = \phi(x_1 - n) \phi_{0,m}(x_2 - m), n, m \in Z \tag{16}$$

is an orthonormal basis for $\boldsymbol{V_0}$. The basis for $\boldsymbol{V_j}$ is obtained (Mallat, 1998) as

$$\Phi_{j,n,m}(x_1,x_2)=\phi_{j,n}(x_1)\phi_{j,m}(x_2)=\frac{1}{2^j}\Phi(2^{-j}x_1-n)\Phi(2^{-j}x_2-m) \tag{17}$$

The orthogonal complement in V_{j-1} for V_j is W_j

$$\begin{aligned} V_{j-1}&=V_{j-1}\otimes V_{j-1}=(V_j\oplus W_j)\otimes V_j\oplus W_j \\ &=V_j\otimes V_j\oplus\left[(V_j\otimes W_j)\oplus(W_j\otimes V_j)\oplus(W_j\otimes W_j)\right] \\ &=V_j\oplus W_j \end{aligned} \tag{18}$$

and, thus, W_j consists of three parts, whose bases Ψ are combinations of one-dimensional scaling function ϕ and wavelet function ψ:

$$\begin{aligned} \Psi^h(x_1,x_2)&=\phi(x_1)\psi(x_2) \\ \Psi^v(x_1,x_2)&=\psi(x_1)\phi(x_2) \\ \Psi^d(x_1,x_2)&=\psi(x_1)\psi(x_2) \end{aligned} \tag{19}$$

The set $\{\Psi^{\lambda}_{j,n};\ j\in Z,\ n\in Z^2,\ \lambda=h,v,d\}$ is an orthonormal basis for $L^2(R^2)$ (Daubechies, 1992). In this construction the sampling is done separately in vertical and horizontal directions, but the wavelet bases are nonseparable.

The fast two-dimensional wavelet transform is performed using filtering operations on vertical and horizontal dimensions of the image. The original image is filtered into quadrants and then the approximation quadrant is filtered further on. If the size of the original image in $N * N$ then each quadrant is of size $N/2 * N/2$. The transform has the perfect reconstruction property.

Similar approach as in the two-dimensional case gives the three-dimensional wavelets that are applied to the three-dimensional data like spectral images. If the separation of the spectral dimension is not applied, then the multiresolution analysis gives the configuration for the transform as

$$\begin{aligned} V_{j-1}&=V_{j-1}\otimes V_{j-1}\otimes V_{j-1} \\ &=(V_j\oplus W_j)\otimes(V_j\oplus W_j)\otimes(V_j\oplus W_j) \\ &=(V_j\oplus W_j)\otimes\{(V_j\otimes V_j)\oplus(V_j\otimes W_j)\oplus(W_j\otimes V_j)\oplus(W_j\otimes W_j)\} \\ &=(V_j\otimes V_j\otimes V_j)\oplus \\ &\quad\{(V_j\otimes V_j\otimes W_j)\oplus(V_j\otimes W_j\otimes V_j)\oplus(V_j\otimes W_j\otimes W_j)\oplus \\ &\quad(W_j\otimes V_j\otimes V_j)\oplus(W_j\otimes V_j\otimes W_j)\oplus(W_j\otimes W_j\otimes V_j)\oplus(W_j\otimes W_j\otimes W_j)\} \end{aligned} \tag{20}$$

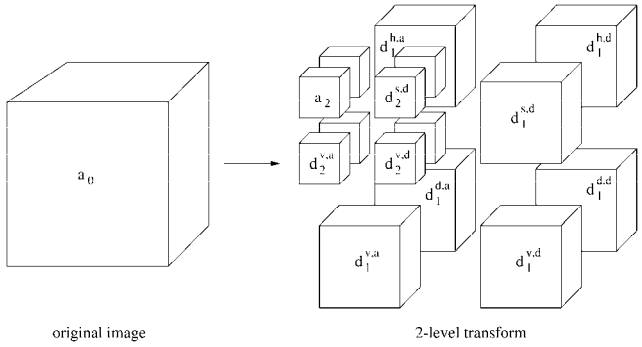
The scaling function for the basis V_0 is

$$\Phi_{0,n_1,n_2,n_3}(x_1,x_2,x_3)=\phi(x_1-n_1)\phi(x_2-n_2)\phi(x_3-n_3),\quad n_1,n_2,n_3\in Z \tag{21}$$

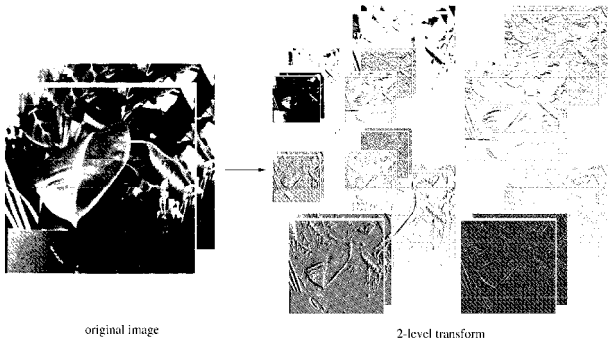
and the filtering of the spectral image is done using one scaling function and seven wavelets, which are defined as

$$
\begin{aligned}
\Phi^{s,a}(x_1,x_2,x_3) &= \phi(x_1)\phi(x_2)\phi(x_3)\\
\Psi^{h,a}(x_1,x_2,x_3) &= \phi(x_1)\phi(x_2)\psi(x_3)\\
\Psi^{v,a}(x_1,x_2,x_3) &= \phi(x_1)\psi(x_2)\phi(x_3)\\
\Psi^{d,a}(x_1,x_2,x_3) &= \phi(x_1)\psi(x_2)\psi(x_3)\\
\Psi^{s,d}(x_1,x_2,x_3) &= \psi(x_1)\phi(x_2)\phi(x_3)\\
\Psi^{h,d}(x_1,x_2,x_3) &= \psi(x_1)\phi(x_2)\psi(x_3)\\
\Psi^{v,d}(x_1,x_2,x_3) &= \psi(x_1)\psi(x_2)\phi(x_3)\\
\Psi^{d,d}(x_1,x_2,x_3) &= \psi(x_1)\psi(x_2)\psi(x_3)
\end{aligned}
\tag{22}
$$

where all dimensions are dilated similarly and the sampling is done separately along each dimension of the three-dimensional image. The original spectral image of size $N * N * N$ is filtered into octants of size N/2 * N/2 * N/2 as illustrated in Fig. 6.



a)



b)

Fig. 6. Three-dimensional wavelet transform applied twice. a) The principle of the transform, the coefficients *a* come from the low-pass filtering and the coefficients *d* from the high-pass filtering. b) Three-dimensional transform applied to one Bristol-image (Parraga et al., 1998).

Similar procedure will produce wavelets in higher dimensions than three. A theorem says (Mallat, 1998) that the family obtained by dilating and translating the 2^p-1 wavelets for $\alpha \neq 0$

$$\left\{ 2^{-\frac{pj}{2}} \Psi^{\alpha} \left(\frac{x_1 - 2^j n_1}{2^j}, \ldots, \frac{x_p - 2^j n_p}{2^j} \right) \right\}_{1 \leq \alpha < 2^p, (j, n_1, \ldots, n_p) \in Z^{p+1}} \tag{23}$$

is an orthonormal basis for $L^2(R^p)$. The configuration of the three-dimensional transform in Eqs. 20, 21, and 22 is compatible with this theorem.
The multiwavelet based transform is slightly more complicated due to preprocessing, computations, and housekeeping (Kaarna & Parkkinen, 1999). This transform has similar variants as the scalar case above. The multiwavelet transform with two scaling functions compatible with Eq. 22, Fig. 6 would contain the coefficients of the first scaling function in the front part of each cubic block and the coefficients from the second scaling function in the back part of each cubic block. The similar division applies to the coefficients from the two wavelet functions.

2.6 Linear compression through spectral decorrelation and spatial compression

A reference method for image compression is based on the principal component analysis (PCA) as a spectral decorrelation method combined with a two-dimensional transform as the spatial compression method. Both the discrete cosine transform (Rabbani & Jones, 1991) and the wavelet transform are used as the spatial compression methods (Kaarna & Parkkinen, 2001). Also the discrete cosine transform has been enhanced to the hyperspectral images (Abousleman et al., 1995).
In Fig. 7 we illustrate the compression method, where the spectral decorrelation by PCA is followed the spatial wavelet transform.

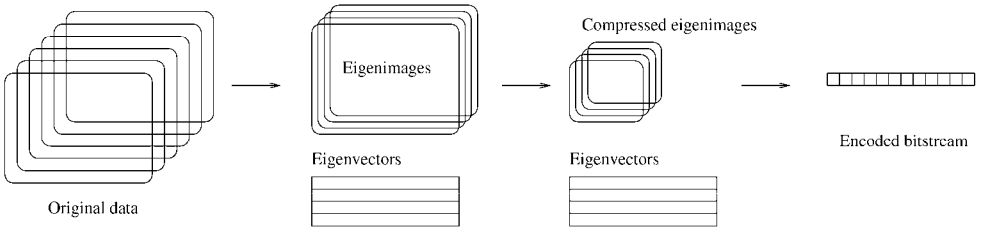


Fig. 7. PCA and the 2D wavelet transform in the lossy compression of spectral images.

The new image compression standard, JPEG2000 was basically defined only for colour images, but the Part 2, extensions, includes also transforms for multiple component imagery (Taubman & Marcellin, 2002). The linear block transform can be considered as a matrix multiplication as described in Section 2.2 as the PCA transform. The multiplication has an inverse operation and, thus, the data can be reconstructed. Also the wavelet transform is defined suitable for a point transform in JPEG2000. The work is currently underway, so the exact definitions are open (Taubman & Marcellin, 2002; JPEG2000, 2006).

2.7 Encoding in the Lossy Compression

The encoding phase of the compression is accomplished in a lossless manner, see Fig. 1. In encoding the quantized coefficients are coded to minimize the amount of data to be stored or transmitted. In addition, the new representation should contain the same information as the original data. In decoding the new representation is coded back to the original data. The encoding methods are divided into statistical methods and dictionary methods. In statistical methods, like static Huffman coding and arithmetic coding, the probabilities of the source symbols are required and two passes of the source are needed for encoding. In dictionary based methods, adaptive methods like the Ziv-Lempel-algorithms LZ77 and LZ78, only one pass is sufficient for encoding (Lelewer & Hirschberg, 1987; Ziv & Lempel, 1977; Ziv & Lempel, 1978).

In LZ77 (Ziv & Lempel, 1977), the source characters were encoded using a window of length *N*. The first *N-F* source symbols were already encoded and the last *F* source symbols constituted a lookahead buffer. The next source symbols in the lookahead buffer *F* were encoded by searching the longest match from the *N-F* source symbols in the window *N*. The match was coded using a pointer and the length of the match. In decoding, no search was needed, since the data was copied from the pointer position (Lelewer & Hirschberg, 1987; Ziv & Lempel, 1977). A modification to the previous method is the LZ78-encoding (Ziv & Lempel, 1978; Bell et al., 1989). Now the source symbols seen so far are split into phrases, where each phrase is the longest matching phrase seen so far plus one source symbol. Each phrase is coded as an index to its prefix plus the extra symbol. The new phrase is also added to the list of phrases that may be referenced. After the introduction of the original LZ-methods, there have appeared several enhancements and modifications to these methods, see e.g. (Bell et al., 1989; Lelewer & Hirschberg, 1987).

In arithmetic coding the source symbols are coded to a magnitude in range *[0,1)* (Langdon, 1984; Rissanen & Langdon, 1979). Initially, the range was split by the probabilities of single source symbols. New source symbols split the existing subranges in a similar way. Finally, all source symbols were manipulated and the subranges gave the codes. The encoding carried the prefix property. In decoding, the model of the source used by the encoder must be known. The encoded value was compared to the known probabilities in range *[0, 1)*, then in the subranges, and finally the decoder output the original source symbols.

SPIHT (Said & Pearlman, 1995) is an effective wavelet-based compression method for two-dimensional images. Color images are compressed through applying the method in each R, G, and B-band separately. For spectral images this approach has been extended to simultaneously manipulate all the bands of the spectral image (Dragotti et al., 2000). The approach combines the wavelet transform with the coding of the selected wavelet coefficients. The wavelet coefficients are coded using a hierarchical tree structure. In Fig. 8, the two-dimensional tree structure is depicted. In the three-dimensional case the structure is extended to include also the spectral dimension. Then the two-dimensional "squares" of coefficients become three-dimensional "cubics". The extension is analogious to that depicted in Fig. 6 for the wavelet transform.

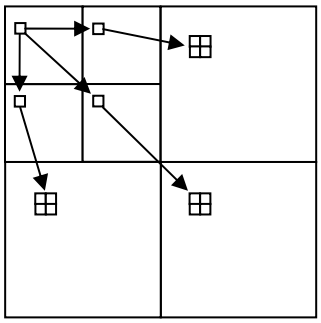


Fig. 8. The tree structure in 2D SPIHT.

2.8 Results from lossy compression

In the experiments we applied the integer PCA and wavelet transform to four AVIRIS images: Jasper Ridge, Moffet Field, Lunar Lake, and Cuprite (AVIRIS, 2006). The spatial size of each image was 608*512 and the number of bands was 224. The resolution of the original images was 16 bits. Thus, each image occupied 139,460,608 bytes of disk space in the raw form. Band 200 from Moffet Field image is displayed in Fig. 9.

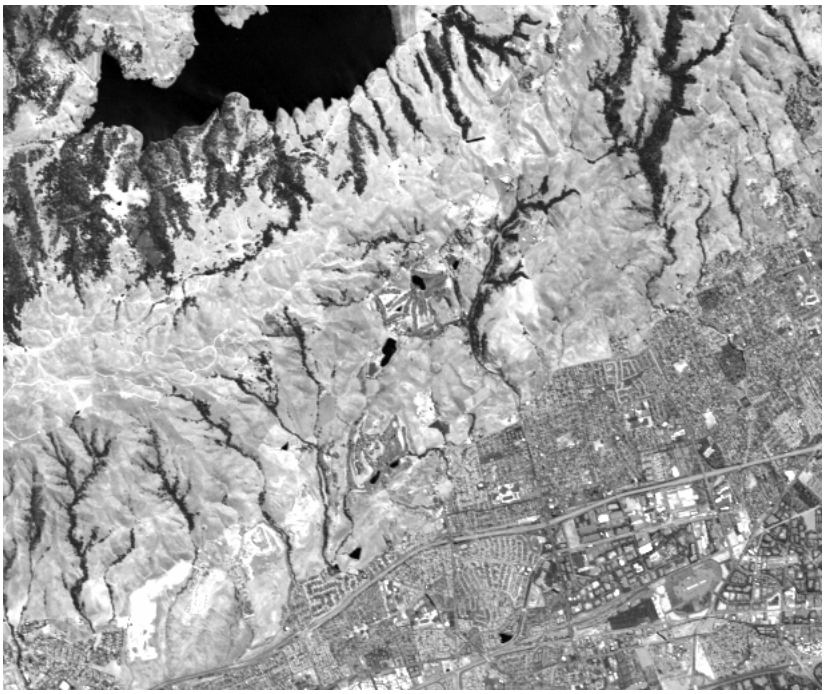


Fig. 9. Spectral band 200 from Moffet Field AVIRIS image.

In Fig. 10 the results from the PCA decorrelation with the 2D wavelet tranform are shown, see Fig 7. With PCA, the variable-bit-rate approach was also applied: the bit-allocation

between the eigenimages was entropy-based (Kaarna et al., 2006). The results from the 3D wavelet transform are also included, see Fig. 6. The horizontal axis is the compression ratio (CR) and the reconstruction quality as PSNR in *dB* (Eq. 28) is in the vertical axis.

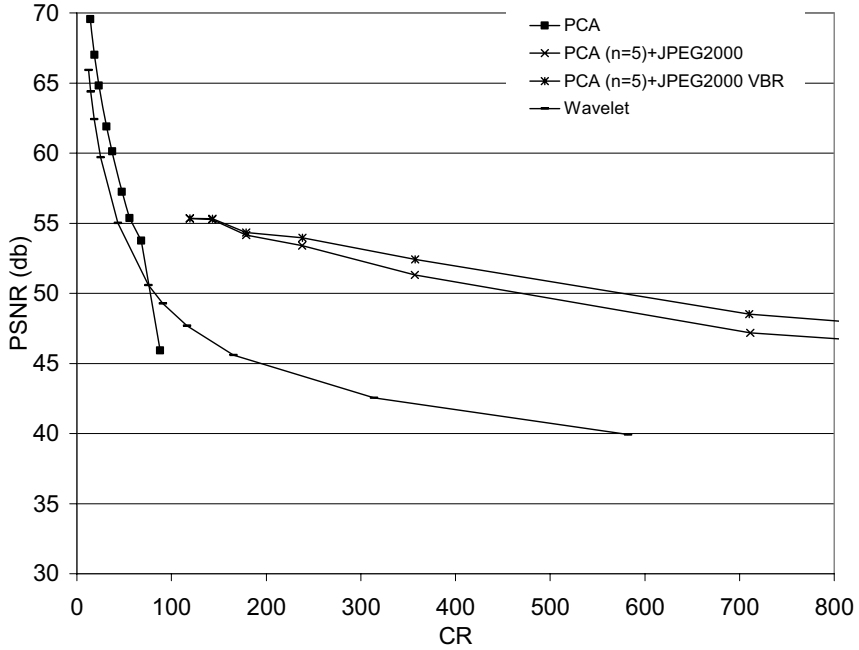


Fig. 10. Results from the lossy compression of AVIRIS spectral images. In PCA, *p=5* (Eq. 3).

3. Lossless compression of spectral images

The lossless compression is also called image coding when all the information in the image is included in the bit stream and the representation of the information is changed into a more compact form. In lossless compression of the spectral images the methods described in Section 2.7 can be used, since they are universal coding approaches for any digital information. These entropy-based textual coding approaches produced compression ratios from 1.5 to 2.0 for a set of AVIRIS-images (Kaarna & Parkkinen, 2001).

For spectral images, the lossless compression can be done by vector quantization and arithmetic coding (Ryan & Arnold, 1997-1). First, the vector quantization is applied to the spectra of the image; second, the residual image is created and then it is coded using arithmetic coding. The residual image contains only integer values obtained after rounding the difference between the closest vector center and the original spectral values. In addition, addresses from vector quantization and a set of other parameters are stored as side information. Also other lossless coding methods exist, e.g. they are based on band ordering (Tate, 1997; Toivanen et al., 2005) or spectral and spatial noncausal prediction (Memon et al., 1994). Thus, in the lossless compression of spectral images better results are achieved through the transform coding and especially, with the predictive coding.

3.1 Transform coding in lossless compression of spectral images

We applied the principal component analysis (PCA) to define the approximation image (Kaarna, 2001). PCA will produce a set of base vectors, which minimize the approximation error in the L^2-sense. The problem of heavy computations in PCA is solved by selecting only a small number of spectra from the image for the calculation of the base vectors. Also an integer version of PCA is needed. The calculation of the eigenvalues and eigenvectors is done with floating values, but the double precision results from the analysis are transformed to integer values such, that the required number of correct digits is maintained in the reverse transform. The PCA transform was described in Section 2.2.

Since the approximation image is available, then the residual image can be calculated. The residual image is then further compressed with the integer wavelet transform. Reversible integer-to-integer wavelet transforms have shown good performance in lossless colour and grey-scale image coding (Adams & Kossentini, 2000). The integer wavelet transform is based on the lifting scheme: different filters were derived by combining the prediction step with the update step (Calderbank et al., 1998). The integer wavelet transform is one-dimensional in nature. In the two-dimensional case, the one-dimensional transform is applied to the rows and columns of the image. In the three-dimensional case, the one-dimensional transform is applied to the spatial and spectral domains separately. The approach is the same as in the general case, see Fig. 8.

Similarly to the floating case, there exists different integer wavelet transforms (Adams & Kossentini, 2000; Calderbank et al., 1998; Daubechies, 1998). The oldest form of the integer wavelet transform subtracts the even samples from the odd samples to get the difference d_1 and the new approximation a_1 as

$$\begin{aligned} d_{1,l} &= a_{0,2l+1} - a_{0,2l} \\ a_{i,l} &= a_{0,2l} + \lfloor d_{1,l} / 2 \rfloor \end{aligned} \tag{24}$$

where the original data is stored in a_0. The second subscript refers to the index of the sample vector. The exact reconstruction comes from calculating the values in reverse order as

$$\begin{aligned} a_{0,2l} &= a_{1,l} - \lfloor d_{1,l} / 2 \rfloor \\ a_{0,2l+1} &= a_{0,2l} + d_{1,l} \end{aligned} \tag{25}$$

In general, the integer wavelet transform consists of the prediction and of the update based on the lifting where the number of vanishing moments is increased. In (Adams & Kossentini, 2000), the best lossless compression results for grey-scale images were obtained with the 5/3-transform, the forward 5/3-transform is defined as

$$\begin{aligned} d_{1,l} &= a_{0,2l+1} - \lfloor 1/2\left(a_{0,2l+2} + a_{0,2l}\right) \rfloor \\ a_{i,l} &= a_{0,2l} + \lfloor 1/4\left(d_{1,l} + d_{1,l-1} + 1/2\right) \rfloor \end{aligned} \tag{26}$$

where a_0 refers to the even samples and d_0 to the odd samples of the original signal. We implemented also other integer wavelet transforms, but the final results were calculated with the 5/3-transform.

The zero order entropies of the AVIRIS test images, see section 2.8, are tabulated in Table 3, column ent_o. The test images were compressed with the lossless Burrows-Wheeler algorithm (Nelson, 1996) and the bitrates are in Table 3, the second column (Bit-rate). The compression