

Hyperspectral Data Compression Tradeoff

Emmanuel Christophe

Abstract Hyperspectral data are a challenge for data compression. Several factors make the constraints particularly stringent and the challenge exciting. First is the size of the data: as a third dimension is added, the amount of data increases dramatically making the compression necessary at different steps of the processing chain. Also different properties are required at different stages of the processing chain with variable tradeoff. Second, the differences in spatial and spectral relation between values make the more traditional 3D compression algorithms obsolete. And finally, the high expectations from the scientists using hyperspectral data require the assurance that the compression will not degrade the data quality. All these aspects are investigated in the present chapter and the different possible tradeoffs are explored. In conclusion, we see that a number of challenges remain, of which the most important is to find an easier way to qualify the different algorithm proposals.

Keywords Compression • Hyperspectral • Quality evaluation • Quality comparison • Data Acquisition • Lossless compression • Lossy compression • Near-lossless compression

1 Introduction

For the past 20 years hyperspectral data are a challenge for data compression. Several factors make the constraints particularly stringent and the challenge exciting. First is the size of the data: as a third dimension is added, the amount of

E. Christophe (✉)

Centre for Remote Imaging, Sensing and Processing, National University of Singapore,
Singapore, Singapore

e-mail: emmanuel.christophe@gmail.com

data increases dramatically making the compression necessary at different steps of the processing chain. Second, the differences in spatial and spectral relation between values make the more traditional 3D compression algorithms obsolete. And finally, the high expectations from the scientists using hyperspectral data require the assurance that the compression will not degrade the data quality.

In [Sect. 2](#), the different steps of the processing chain, where compression is required for hyperspectral data, are detailed: the specific requirements for each situation are explained and the different possible tradeoffs are explored. The following [Sect. 3](#) goes more deeply into the exploration of some key characteristics of hyperspectral data that can be successfully used by compression algorithms. Examples are drawn from the recent published literature on the subject. Finally, in [Sect. 4](#), requirements for an accurate assessment of image quality are explored.

2 Data Acquisition Process and Compression Properties

Compression is a way to reduce the amount of data to be transmitted or processed. The compression can be lossless without any impact on the data, or lossy, when the data values are distorted in the process and the original data cannot be retrieved in their original form. Compression is a tradeoff between processing capabilities and size (whether it is storage or transmission). Lossy compression adds a third dimension to the equation: data quality.

Before defining the compression algorithms, it is important to understand the context in which they operate and the constraints that led to their definition. There is not much in common in the requirements for compressing data onboard a satellite and compressing data on-ground. In the first case, computational power is limited and any error is unrecoverable, while in the second case, compression is used for speeding up network transfer or processing but the whole data can be transmitted if necessary.

The properties required for these algorithms will be strongly dependent on the aim.

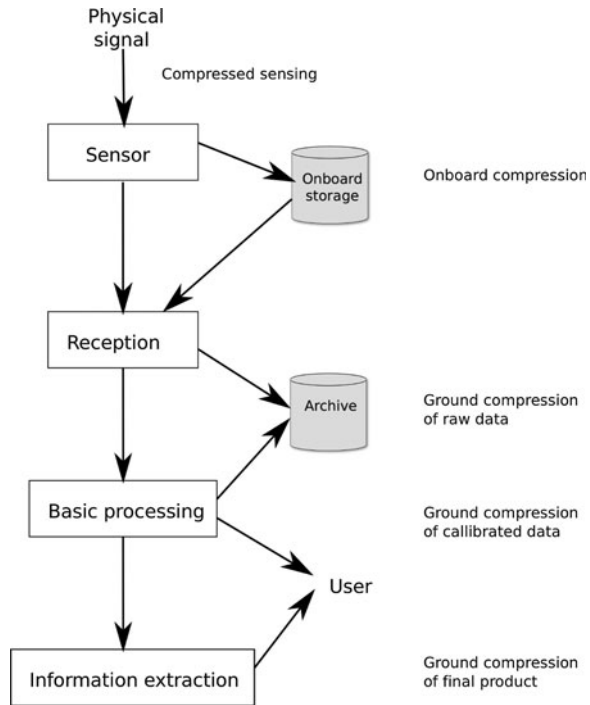
2.1 Data Acquisition Process

The first important question is to find out where the compression is going to take place. This will define the data to work on, the constraints on the algorithm and the desirable properties.

The processing chain is similar, whether the data are acquired by space-borne or air-borne sensor. Data compression usually occurs at several levels in the chain, where different tradeoffs take place. [Figure 1](#) presents a typical processing chain.

The first place where data compression can occur in the processing chain is in the acquisition process itself. This is quite a recent paradigm, widely known as

Fig. 1 Data acquisition chain



compressed sensing but it will be treated in [Reconstructions from Compressive Random Projections of Hyperspectral Imagery](#). We will focus in the present chapter on more traditional techniques.

After the signal acquisition, the information will be either stored onboard or directly transmitted. Direct transmission usually requires constant bitrate: this requirement can be mitigated by the use of memory buffers. The transmission which occurs in noisy environment requires redundancy coding. Both compression (source coding) and redundancy coding (channel coding) can be combined in a single operation using joint source and channel coding [1]. If the hyperspectral instrument is space-borne, all this processing is subjected to stringent requirement in term of complexity. The constraints of onboard compression are detailed in [Sect. 3](#).

Once the signal is received on the ground, it is transmitted over cable network and stored for future use. At this part of the processing, the complexity constraint is greatly relaxed. A transcoding step can occur to keep the data in a more practical format. However, in some situation the delay between the reception and the final product must remain short. Different properties of the encoded bitstream are expected from the user. These properties are detailed in [Sect. 4](#).

Figure 1 presents the most common processing chain where minimal operations are performed onboard. However, due to the evolution of technology, some simple

operations such as calibration can be applied before compression of the signal. This will lead to different properties of the data to be compressed that should be considered during algorithm evaluation by working on the right data. These considerations are detailed in [Sect. 5](#).

One important factor to keep in mind when designing a compression system is the end-user. Depending on who the end-user is and what information is intended to be retrieved from the data, the optimal compression solution can be very different from one case to another. The first point to consider is whether the objectives of the mission are specific or generic. A specific mission would intend to use hyperspectral data to obtain a detailed land cover map of the area for example. It could also be used to raise a warning when some anomalies are detected. In these cases, the purpose of the mission is clearly identified and the final product fully defined. On the contrary, a generic mission does not preclude any possible application. In this case, the mission has to transfer the information to the final user in a form that is as close as possible to the physical measurement.

For both these situations, specific or generic application, the compression should have no impact from the point of view of the application. No impact does not necessarily mean no differences as the error could stay within the confidence interval of the application itself. More details on the error are presented in [Sect. 4](#).

2.2 *Lossy, Lossless, Near-Lossless*

Lossless compression algorithms enable the users to retrieve exactly the original data from the compressed bitstream. They are generally based on a predictor followed by an entropy coder of the residuals. The most recent publications in this domain [2–5] converge towards a compression ratio around 3:1. Such a compression ratio is insufficient to meet the constraint for onboard systems [6]. However, they are highly relevant for archiving the data and distribution to the end-user.

Lossy compression on the contrary introduces a distortion in the data and the original data cannot be retrieved in its exact form. These methods generally have parameters that can be adjusted to move along the rate–distortion curve. Reducing the bitrate increase the distortion and vice-versa.

When the distortion remains small, the algorithm can be qualified as near-lossless. Two main definitions appear in the literature for *near-lossless compression* of hyperspectral data. The first definition [6] considers that the compression is near-lossless if the noise it introduces remains below the sensor noise: the data quality remains the same. The other definition [7] considers that an algorithm is near-lossless if the distortion is bounded. We will stick to the former definition which guaranties no distortion from the application point of view: the compression remains in the noise of the sensor.

2.3 Onboard

Amazing acquisition capabilities of satellites make them the ideal candidates for regular monitoring. Many fields would benefit from regular observations. Since the launch of Hyperion on EO-1 on November 2000, the feasibility of hyperspectral space sensors has been demonstrated. Several projects in the coming years will probably increase the amount of hyperspectral data available.

For satellite sensors, the trend is towards an increase in spatial resolution, radiometric precision and possibly the number of spectral bands, leading to a dramatic increase in the amount of bits generated by such sensors. Often, continuous acquisition of data is desired, which requires scan-based compression capabilities. Scan-based compression denotes the ability to begin the compression of the image when the end of the image is still under acquisition. But due to the amount of data collected and the limited transmission capacity, there is no doubt that data has to be compressed onboard. Onboard compression presents several challenges. First, if the compression is lossy, losses are irrecoverable; this is to contrast with data compressed for transmission to the user, where the compression can be modified and data retransferred if it appears that there is a need for a higher quality. This fact makes it particularly challenging to accept onboard lossy compression even if the impact is proven to be negligible.

The second point concerns limited processing capabilities: electronics onboard a satellite need to be protected from radiation, work in a vacuum environment, have a low power consumption, limited heating, and support these conditions for several years. All these conditions cause a lag of several years in terms of processing power capabilities between consumer electronics and satellite electronics.

As onboard storage is limited, the data need to be processed on the flow as they are acquired: the start of the scene is compressed and transmitted before the end of the scene is even acquired. Satellite acquisition is done in pushbroom mode where the spectral dimension and one spatial dimension are acquired simultaneously, while the second spatial dimension is created by the satellite motion. Data ordering such as bits interleaved per pixel (BIP) or bits interleaved by line (BIL) are representative of this acquisition process.

Another consequence of this limited onboard storage is that data is often transmitted while it is acquired. One requirement to enable this transmission is a constant throughput of the compression system. This requirement can be alleviated by the use of buffers.

Desirable properties for onboard compression are summarized in Table 1.

2.4 Image Distribution

For image distribution, the challenges are very different. Data transmission is not the main problem, but the constraint is rather on processing and visualization. Due

Table 1 Desirable properties for onboard compression

Constant throughput
Low complexity
On the flow coding
Error resilient

to the huge amount of data involved, even compressed images are significant in size. In this situation, progressive data encoding enables quick browsing of the image with limited computational or network resources.

When the sensor resolution is below 1 m, images containing more than $30,000 \times 30,000$ pixels are not exceptional. In these cases, it is important to be able to decode only portions of the whole image. This feature is called random access decoding.

Resolution scalability is another feature which is appreciated within the remote sensing community. Resolution scalability enables the generation of a quicklook of the entire image using just few bits of coded data with very limited computation. It also allows the generation of low resolution images which can be used by applications that do not require fine resolution. More and more applications of remote sensing data are applied within a multiresolution framework [8, 9], often combining data from different sensors. Hyperspectral data should not be an exception to this trend. Hyperspectral data applications are still in their infancy and it is not easy to foresee what the new application requirements will be, but we can expect that these data will be combined with data from other sensors by automated algorithms.

Strong transfer constraints are ever more present in real remote sensing applications as in the case of the *International Charter: space and major disasters* [10]. Resolution scalability is necessary to dramatically reduce the bitrate and provide only the necessary information for the application.

For ground compression, error recovery is not so critical as most of the time information can be transmitted again on demand.

As the main purpose at this level is to make the image available, it is important to ensure the wide availability of the decompression algorithm. This is where the usage of an established standard is particularly relevant. Image users have a wide variety of software to analyze and process the images. This software usually implements standard formats. If the data are distributed in a specific format, transcoding into a standard format is generally required before processing. Having the data already in a standard format can save this transcoding step.

Finally, the raw pixel data is not the only product of interest for the user. First, auxiliary data are required to apply correction to the image (geometry or radiometry corrections for example), to extract geographic information, to combine with other images, etc. Going further, value added products can also be distributed directly by the data provider: classification, end-members, etc. In these cases, it is important that the format handles this information seamlessly with the image data.

Desirable properties for compression for image distribution are summarized in Table 1.

2.5 Data Availability

When designing a compression algorithm, the choice of the data on which it is going to be evaluated is important. The choice of the quality measurement is also critical and will be presented in Sect. 4. Several considerations need to be taken into account for the choice of the data (Table 2).

The first factor is availability: is there any dataset available that is representative of the mission? If not, simulations take a particularly important role. In some cases, similar data might be available from an instrument operating in different conditions (airborne sensor instead of spaceborne sensor). This is the case for example of the Aviris data sets [11]. These datasets are widely used and enable a quick and easy comparison with previous published results. Figure 2 illustrates a color composition of the four popular datasets: Moffett Field (Fig. 2a) and Jasper Ridge (Fig. 2b) represents a mix of urban area and vegetation, the two other tracks, Cuprite (Fig. 2c) and Lunar Lake (Fig. 2d) are more focused on geology application as the content is mostly minerals.

The second point is the data level to consider. If onboard compression is targeted, radiance data should probably be considered or even better, uncalibrated data if they are representative of the targeted sensor. If compression for the final user distribution is targeted, the reflectance product or even the final product can be compressed.

The third point concerns the processing required to simulate the targeted sensor. For example, if Aviris data are used to qualify a hyperspectral compression system that would be onboard of a satellite, it is unlikely that the same signal to noise ratio could be reached. In this case, additional noise should be added to the data before the compression. Some specific artifacts should also be considered. In [5], for example, it is shown that the algorithms giving the best results on unprocessed data are not the same than the best ones on calibrated data (radiance).

3 Trends in Compression Algorithms

Hyperspectral data presents an enticing challenge with an original relation between spatial and spectral information and a high information value which may rely on subtle variations of the spectrum. As a consequence, efficient compression remains an open problem. Several publications tackle the problem taking a diversity of approaches.

Table 2 Desirable properties for image distribution

Random access
Progressive decoding
Established standard
Access to value added products

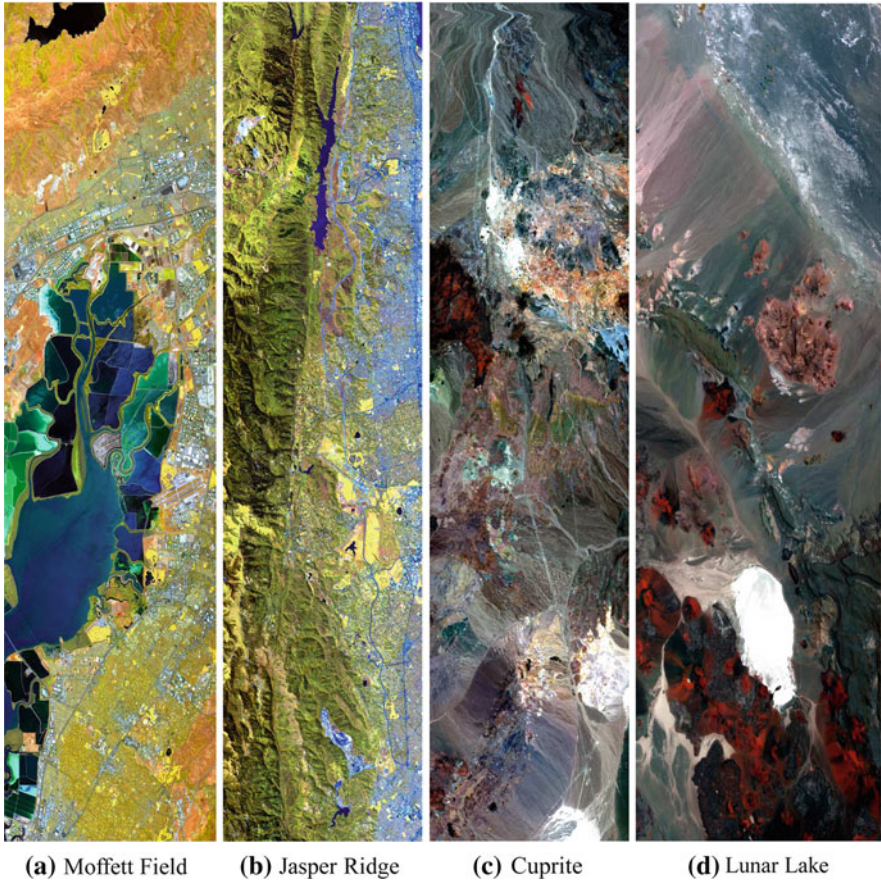


Fig. 2 Classic data sets used for compression algorithms evaluation

We can separate these methods into three groups: prediction, vector quantization and transform coding. These three different approaches have been successively refined, leading to an important diversity of methods. Some of the most recent papers on the subject for prediction-based methods are [2–5, 7, 12–14], most of them in lossless compression; vector quantization recently appears in [6, 15], and transform methods in [16–28].

3.1 Prediction-Based

Directly following the main trend for lossless compression algorithms for 2D images, several adaptations for hyperspectral image compression are devised based on prediction methods. In these approaches, the data are first decorrelated by

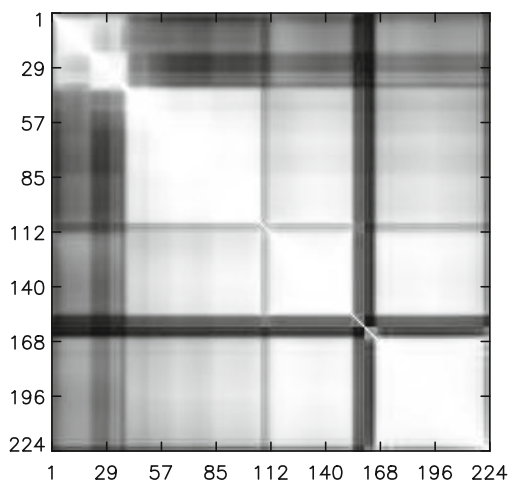
a predictor. In a second step, the prediction error is coded by an entropy coder. The predictor takes advantage of the strong correlation between spectral bands (as presented in Fig. 3). It also relies on correlation with neighboring pixel values.

As shown on Fig. 3, the correlation is not only between neighboring bands, but also between bands far apart in the spectrum. This is particularly striking for the visible part of the spectrum, which is highly correlated with the infrared (for bands 20 and 120, the correlation is above 0.6 for example), but not so much with the near-infrared (for bands 20 and 60, the correlation is around 0.3 for example). This is mainly due to the specific response of the vegetation in the near-infrared region with a strong signal due to chlorophyll. In [29] for example, it is shown that optimal reordering of the bands for Aviris can lead to a gain of 18.5% in compression performance. However, the optimal reordering might not be feasible onboard [7] and some simplifications are often used. Most of the time, only the previous band is used as a predictor. The most promising method in the domain of prediction-based compression seems to be the use of lookup tables (LUT) [2, 5, 13, 30] or the adaptation of CALIC [4, 7].

3.2 Vector Quantization

Vector quantization (VQ) of hyperspectral data is very tempting as one of the most popular application of hyperspectral data is classification. When the classification algorithm only considers pixels one by one, each pixel is assigned to the nearest class (in term of classification distance). This naturally brings the notion of codebook, each codeword being the spectrum of one material in the scene. Only the codebook (the classes) and the map (classification) have to be transmitted. This

Fig. 3 Interband correlation for the Moffett hyperspectral image on a gray level scale: white corresponds to highly correlated bands while black to uncorrelated ones. Abscissa and ordinate represents the band number



is a significant reduction of the data. However, most of the time, as generic applications are targeted, the method is more complex and provides much more than a classification.

VQ compression has two separate steps: a training step, where the codebook is constructed and a coding step where each vector is assigned to a codeword. One of the common methods to generate a codebook is the Generalized Lloyd Algorithm (GLA). However, high computational costs of this algorithm presents a challenge for the compression of hyperspectral data [15]. Most of the work focuses on simplifying this step to relax the complexity constraints.

Work is going on within the Canadian Space Agency [6, 15, 31] as well as in other teams [32–34] on the use of vector quantization for the compression of hyperspectral data. In general, the targeted compression rate is high (typically 100) with a significant distortion on the image but not on classification applications where the impact is negligible. However, when the compression rate remains small, the vector quantization algorithms remains acceptable for a wider range of applications [6].

3.3 Transform Methods

Transform coding works in two steps, the first step is to transform the data in a domain where the representation of the data is more compact (energy compaction) and less correlated. The second step is to encode this information as efficiently as possible. It is during this last step, encoding, that the information loss occurs, usually through quantization. Most of the algorithms developed for hyperspectral data compression revolve around this scheme with variations on how the two steps are defined.

3.3.1 Transform

The correlation between spectral bands is important in hyperspectral data. The spectral variations are usually much slower than the spatial variations. The consequence is that hyperspectral images are more compressible than traditional images. Figure 3 presents the correlation of spectral bands with each other. The correlation coefficient is often above 0.9, even for spectral bands separated by several hundred nanometers.

From the point of view of signal theory, the most efficient transform in terms of energy compaction and decorrelation is the Karhunen–Loeve Transform (KLT) which is strongly related to the Principal Component Analysis (PCA). In [18], it is shown that using KLT transform to decorrelate the spectral bands lead to a quality gain of more than 20 dB. The main drawback is that the transform is costly in terms of computation (Table 3). The basis vectors depend on the data. In the case of an onboard compression system, a full KLT transform cannot be implemented

Table 3 Surface of silicium required to implement a KLT transform (on ASIC): without including the computation of the transform matrix according to the number of spectral bands

# Bands	Surface (mm ²)
16	6
64	99
128	400
256	1,500
The limit in 2006 was around 110 mm ² [35]	

for 200 spectral bands. Several solutions specifically target the simplification of the KLT transform.

One of the first solutions to avoid the complexity of the KLT is to precompute the transform coefficients on a set of typical images and reuse these coefficients for all images. But unfortunately, if it works for multispectral images with few bands [36], it does not for hyperspectral images as the variations in the spectra between pixels become too important to be efficiently decorrelated by an average KLT.

Some papers, such as [28, 37] design simplified versions of the KLT transform to enable its implementation onboard satellites.

The other most popular transform is the wavelet family. In [18], it is shown that using wavelet transform to decorrelate the spectral bands leads to a quality gain of more than 15 dB. The extension of the 2D wavelet transform to the 3D space led to wide range of possibilities. The first variation is on which wavelet to use. The standard 9/7 and 5/3 wavelets, which were also adopted by the JPEG 2000 standard are the most popular. Due to complexity and memory constraints, these wavelets are usually separable. The second variation is on the order in which the separable wavelets should be applied. The straightforward extension of the Mallat decomposition to 3D does not lead to the best results and another simple decomposition appears to be nearly optimal [18]. Several papers uses this decomposition which is becoming the standard [16, 19, 28, 38]. The decomposition is illustrated on Fig. 4: first the multiresolution wavelet transform is fully applied to each spectrum, then the dyadic 2D wavelet decomposition is applied on each resulting plane.

3.3.2 Coding

Once the energy is compacted to a small number of coefficients, several methods are used to code these values.

All the subtleties of the different coding methods rely on the more efficient way to order these data and/or how to predict them. The prediction methods are related to the methods presented in Sect. 3.1, but remain different as the correlation is much lower here.

Usually, the first step is quantization, which is the step where the distortion takes place. Often this occurs indirectly during a bitplane coding. Bitplane coding is a way to navigate in the binary data to be encoded starting from the one with the

Fig. 4 3D Wavelet decomposition commonly used for hyperspectral images

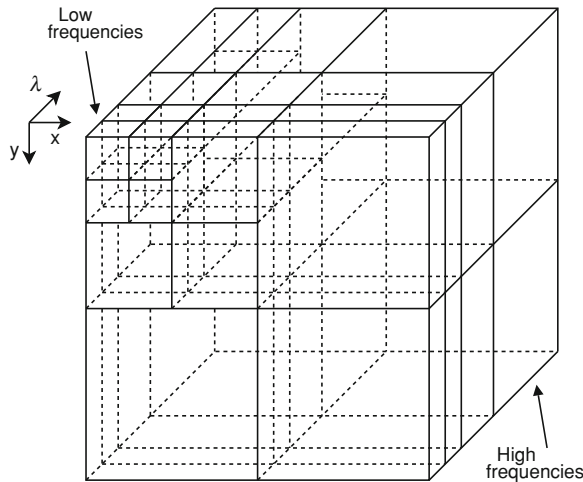


Table 4 Example of bitplane coding

Bitplane	q	5	63	173
7	128	0	0	1
6	64	0	0	0
5	32	0	1	1
4	16	0	1	0
3	8	0	1	1
2	4	1	1	1
1	2	0	1	0
0	1	1	1	1

This would typically lead to a bitstream 001000011010011..

greater impact first. This enables progressive decoding of the final bitstream. For example, with the example presented in Table 4, the numbers 5, 63, 173 are all going to be encoded starting by the first bitplane (quantization step $q = 128$), and then progressively refined by successive smaller quantization. This process ensures that the value with the most energy will be coded and transmitted first. As the distribution of the value to be coded, which is the result of the transform or the residual of a predictor, is close to a Laplacian distribution, for the higher bitplane, most of the values will be zero which can be very efficiently coded.

The order in which these bits are going to be visited can be further refined. As the data are sparse, there is a high number of zeros (at least in the higher bitplane). The idea is to maximize the number of zeros that can be encoded together. Some strategies exist to further increase the amount of zeros in the stream to be encoded, such as the use of signed binary digits [39]. The main strategies to

take benefit of long streams of zeros are to exploit the fact that small wavelet coefficients are clustered in similar spatial areas: if the wavelet coefficient is small in the low frequency band, it is likely to be small in the high frequency band for the same spatial location. This fact is used by zerotree algorithms such as EZW, SPIHT and SPECK that have been successfully adapted to hyperspectral data [19, 40, 41]. In these algorithms, the visiting order of the value is designed to maximize the probability to code large chunks of zeros using only one symbol.

Once the data visiting order is designed, the data coding itself takes place: using the minimum number of symbols to code the stream. Arithmetic coders have been very successful, but the implementation complexity can be a deterrent. Simpler coders such as run-length, Lempel–Ziv algorithms are also used.

Most of algorithms currently used for compression are a combination of these different steps, some of them being optional: JPEG 2000 combines the wavelet transform with a contextual arithmetic coder [42]. 3D-SPIHT combines the wavelet transform with tree ordering, without the requirement of specific coding thereafter.

3.4 *Lossy to Lossless*

One of the current trends in the definition of new compression algorithms for hyperspectral data is to try to get the best of both worlds and provide a progressive compressed bitstream which is able to reach lossless quality. Several possibilities exist:

- use a lossless algorithm that is able to do progressive encoding;
- use an hybrid solution combining a lossy algorithm with error encoding techniques.

For the first case, JPEG2000 can be used with the 5/3 integer wavelet transform. The bitstream is progressive: decoding only the beginning of the compressed data leads to data with a lower quality but adapted to some applications. If the full stream is decoded, the data are recovered without any distortion. One main drawback of the method is the relatively low quality obtained for intermediate bitrates: the integer wavelet transform is not as efficient as the 9/7 for the decorrelation of hyperspectral images.

The second solution encodes the residual error of the lossy encoding. The residual error can be encoded using a DPCM scheme for example as in [25]. The performance of the lossy compression part is preserved and the residual error is used only if required, but this causes an increase in the complexity of the algorithm, particularly in terms of memory handling.

These methods are most likely to find an application in the ground segment for data archiving where the complexity constraints are relaxed and when no transcoding losses are tolerated.

3.5 *What is in Use Now?*

All these major trends have been successfully implemented and/or used in real situation. Here are some examples; note that these have been mainly used in the case of a demonstration mission to show the capabilities of hyperspectral data.

A system based on onboard classification was planned for the canceled mission of Cois on the Nemo satellite. This system, named Optical Real-time Adaptive Signature Identification System (Orasis), enables compression ratios of 30:1 while preserving good quality for classification applications [43].

On the transform side, the SPIHT algorithm is a good candidate for onboard hyperspectral data compression. A modified version of SPIHT is currently flying towards the 67P/Churyumov-Gerasimenko comet and is targeted to reach in 2014 (Rosetta mission) among other examples. This modified version of SPIHT is used to compress the hyperspectral data of the VIRTIS instrument [44]. This interest is not restricted to hyperspectral data. The current development of the Consultative Committee for Space Data Systems (CCSDS, which gathers experts from different space agencies as NASA, ESA and CNES) is oriented towards zero-trees principles [45]. The CCSDS currently has a group working on hyperspectral data compression targeting to reach a standard by 2011.

The vector quantization solution is quite advanced in terms of progress and demonstrated feasibility with hardware implementation on FPGA [6]. More importantly, this algorithm was also submitted to an extensive acceptance study by hyperspectral data users [46]. This study, using a double-blind setup, has demonstrated that compression rate of 10:1 seems acceptable for all applications and compression rates of 30:1 are for most of them. This is a gain of a factor 3 to 10 compared to lossless compression.

Of course when compression is used to distribute the data to the end user, established standards benefit from the wide availability of software able to read and process the data. The JPEG 2000 format is increasingly popular for the distribution of high resolution satellite data.

4 Ensuring Sufficient Quality

4.1 *Why Bothering with Lossy Compression?*

Given the fidelity requirement of the final applications whether it is target recognition (see [A Divide-and-Conquer Paradigm for Hyperspectral Classification and Target Recognition](#)), classification (see [Decision Fusion of Multiple Classifiers for Vegetation Mapping and Monitoring Applications by Means of Hyperspectral Data](#)) spectral unmixing (see [Recent Developments in Endmember Extraction and Spectral Unmixing](#)) or change detection (see [Change Detection in VHR Multispectral Images: Estimation and Reduction of](#)

Registration Noise Effects), any loss of information caused by compression is unacceptable. This is one of the main reasons why lossless compression is still so popular on hyperspectral data. However, the consideration has to be taken in a wider range than just image to image comparison. We have to look at the mission globally to find the optimal tradeoff. Compression enables gathering more data, the cost being a slight distortion on the final product. The question that needs to be answered is *does the increase in acquisition capability (increasing information) offset the quality loss (decreasing information)?*

For example, the MERIS sensor onboard the ENVISAT satellite acquires hyperspectral data in 520 spectral bands before averaging some of them and discarding other to produce the final selectable 15 band products [47]. To further reduce the output rate, averaging is also performed on the spatial domain.

In the setup of a specific application, the answer to this question can be validated quite easily using simulations. Using compression could enable an increase in resolution (providing more details), an increase in swath (reducing the revisit delay, thus improving multitemporal resolution), more spectral bands or an increase duty cycle (increasing the amount of images collected per orbit).

If every application would benefit from an increase in the amount of data collected, most of them would also suffer if the data quality is impacted. This is especially true when generic applications are targeted. In these conditions, it makes no sense to target compression ratio higher than 20:1 and a bitrate between 1 and 2 bit per pixel per band (bpppb) seems a reasonable target.

Lossless coding is very reassuring from the point of view of the user. This is the assurance that the compression algorithm will not change the data at all. But if it is considered in the more global situation of the mission trade-off, given the fact that sensor noise affects the data anyway, lossless compression is definitely not the optimal choice.

4.2 Quality Evaluation

With qualifying lossy compression comes the problem of quality evaluation. The important point is the impact on the end-user, but it is particularly difficult to evaluate or compare algorithms from the application point of view. The most convincing measure is to show the impact on a real application using ground truth before and after compression. However, a realistic evaluation is not often done in the literature as compression specialists are rarely also application specialists. The first shortcut which is often taken is to use a statistical distortion measure (such as SNR, PSNR or MSE). But such measures, even if widely used, have well-known drawbacks: see [48] for a review on the topic. The second widely popular shortcut is to measure how well the compression preserves the results of a benchmark application. This can be referred as Preservation of Application Results (PAR), which is a more general case of the Preservation of Classification (POC) presented

in [16]. Both these cases are different than measuring the true impact of the compression on the applications.

There is currently no universal method to provide a quality evaluation. For example, if we review papers on lossy hyperspectral compression published in IEEE journals in the last three years [6, 17, 18, 20, 21–28], six papers present only statistical measurement (SNR, spectral angle) [17, 18, 20, 21, 22, 23], five present additional examples on applications comparing with the results obtained on the original image, classification [24] or anomaly detection [25–28]. Only two compare with real ground truth [28], for the classification (this paper also evaluate the anomaly detection performance, but as PAR) and only [6] provides extensive results on a wide range of real applications with the participation of several experts using the set up described in [46]. These results are not surprising, and, given the difficulty to set up a correct evaluation, such a set up cannot be expected for each paper.

Choosing a suitable quality measure is not an easy task and the amount of existing criteria to quantify the quality of compressed hyperspectral data is significant: for example see [49] for a non-exhaustive list of quality criteria for hyperspectral images.

The Preservation of Application Results (PAR) supposes that results obtained from the original data (classification, anomaly detection, ...) are as close as possible to the ground truth. This only is an approximation of what we really want to measure: the classification accuracy compared to the ground truth or the real anomaly detection rate. The ideal is of course to compare the results with a ground truth, but this is not easily available.

There is a trend towards standardizing the datasets used for the evaluation of hyperspectral compression algorithms (see Sect. 5). This is already a great improvement. The trend should continue towards the availability of standard application algorithms with ground truth to make the evaluation and comparison of quality more objective. The website [50] of the Data Fusion Contest (DFC) 2008 [51] proposes the automatic evaluation of hyperspectral classification with ground truth. This system can be used to qualify the impact of hyperspectral data compression on this particular application. Another system proposing an automatic evaluation of anomaly detection would be a very valuable complement to the existing one. Anomaly detection is an important application of hyperspectral remote sensing and is neglected by most evaluations (none of the aforementioned papers compare anomaly detection with a real ground truth). A third one that would be a perfect complement would be a spectral unmixing application. With these three applications, a much better evaluation of the impact of hyperspectral compression could be done.

So we have to separate problems here: how to compare the different algorithms between each other and how to get a precise evaluation of the impact on the targeted application. Ideally, these two problems would be one, but given the number of algorithms available and number of existing applications, it is more convenient to rely on simpler measure for comparison.

4.3 Making Comparison Easier

In Sect. 5, we insisted on choosing representative data for the targeted application. But once again, as comparison is important, results should also be provided for a classic case. If the algorithm is highly specialized for one particular type of image, the results can be compared also for a standard algorithm on this case and contrasted with the ad-hoc proposed algorithm. If the algorithm is targeting the minimization of error for a particular application, once again, it can be contrasted with a standard algorithm. As it is by definition a standard, JPEG 2000, seems the ideal candidate for this task. Several implementations are freely available and easy to use. Section 5 will provide the results on the classic images.

5 Reference Results

As it is a widely available solution and standardized, the result for JPEG 2000 compression are presented for reference on some popular data sets. The user should be able to reproduce these results without trouble and compare with the implementation of its own algorithm.

However, for simplicity and because it is among the most widely used, we choose the popular SNR which can be easily converted to the ever popular PSNR or MSE (meaning that the ranking between algorithms would be the same).

When computing the SNR, one has to be careful about the variance computation which introduces an additional source of error. Hyperspectral images contains an important number of pixels on a wide range of values, computational artefacts (which becomes significant when millions of small values are added) appear in some publications. Depending on the algorithm used for computation, one has to be careful to use double precision to avoid such artifacts.

Table 5 presents the results for the popular Aviris data set for JPEG 2000 lossless compression. JPEG 2000 is used without any fancy options. The only non classic option is the use of the multicomponent decomposition (MCT) using wavelets as defined in the standard [42, 52]. Five levels of decomposition are used in the spatial and spectral directions. The decomposition is equivalent to the one illustrated in Fig. 4. The rate allocation is done considering all the wavelet subbands together (default behavior of Kakadu).

The implementation used to obtain those results is Kakadu v6.2.1 [53], most of the options are the default one apart from the MCT which requires specific parameters.

The dataset are the first scene of the three popular tracks, in radiance and reflectance. The original scenes are 614×512 pixels. The results are presented for the original scenes, but also for some common extracts: 512×512 pixels and 256×256 starting from the top left.

Table 6 presents the results obtained with JPEG 2000 in lossy configuration. The results are presented in terms of SNR and maximum error. These results can

Table 5 Lossless performances

Scene	Size	Rate (bpppb)	Compression ratio
Moffett Field (sc 1) Radiance	614	5.684	2.815
	512	5.654	2.830
	256	5.557	2.879
Jasper Ridge (sc 1) Radiance	614	5.598	2.858
	512	5.547	2.885
	256	5.390	2.968
Cuprite (sc 1) Radiance	614	5.291	3.024
	512	5.286	3.027
	256	5.261	3.041
Moffett Field (sc 1) Reflectance	614	6.865	2.331
	512	6.844	2.338
	256	6.767	2.365
Jasper Ridge (sc 1) Reflectance	614	6.619	2.417
	512	6.573	2.434
	256	6.428	2.489
Cuprite (sc 1) Reflectance	614	6.755	2.369
	512	6.763	2.366
	256	6.784	2.359

Table 6 Lossy performances at 1.0 bpppb

Scene	Size	SNR	MAD
Moffett Field (sc 1) radiance	614	45.233	90
	512	45.453	91
	256	45.898	87
Jasper Ridge (sc 1) radiance	614	44.605	96
	512	44.807	78
	256	45.367	60
Cuprite (sc 1) radiance	614	50.772	58
	512	50.920	54
	256	51.259	51
Moffett Field (sc 1) reflectance	614	36.110	444
	512	36.438	444
	256	37.865	260
Jasper Ridge (sc 1) reflectance	614	36.446	225
	512	36.983	201
	256	37.647	127
Cuprite (sc 1) reflectance	614	34.995	283
	512	34.952	283
	256	34.829	291

easily be obtained for other bitrates as both the images and the JPEG2000 implementation are available. Any new proposal for an hyperspectral compression algorithm can be easily compared with this reference to provide a convenient comparison point for the reader.

6 Conclusion

Despite the numerous algorithms proposed, a number of challenges remain in the area of hyperspectral image compression. One of the main challenges is the evaluation of the impact of lossy compression. The lack of confidence of the final users in these evaluations is probably the main reason for the reluctance to accept near lossless compression in spite of the significant advantage in acquisition capabilities.

An extensive study conducted in a double blind setup shows that a factor of 3 can be obtained with near lossless compression compared to lossless compression with no impact from the user point of view. This shows that near-lossless compression is the best tradeoff for onboard compression of generic missions. However the procedure to evaluate the impact of the distortion needs to be refined as it is not conceivable to conduct a new extensive study with end-users for each new algorithm proposal.

Once the procedure for impact evaluation is accepted, more advanced concepts to reduce the data volume with an acceptable complexity can be proposed. This is the case of compressed sensing which proposes a shift in the compression paradigm, shifting most of the complexity at the decoding step.

Acknowledgments The author wishes to thank Corinne Mailhes for valuable suggestions that improved this chapter and for carefully reading the early versions.

References

1. Abousleman, G., Lam, T.-T., Karam, L.: Robust hyperspectral image coding with channel-optimized trellis-coded quantization. *IEEE Trans. Geosci. Remote Sens.* **40**(4), 820–830 (2002)
2. Mielikainen, J., Toivanen, P.: Lossless compression of hyperspectral images using a quantized index to lookup tables. *Geosci. Remote Sens. Lett.* **5**(3), 474–478 (2008)
3. Huo, C., Zhang, R., Peng, T.: Lossless compression of hyperspectral images based on searching optimal multibands for prediction. *Geosci. Remote Sens. Lett.* **6**(2), 339–343 (2009)
4. Magli, E.: Multiband lossless compression of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **47**(4), 1168–1178 (2009)
5. Kiely, A.B., Klimesh, M.A.: Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **47**(8), 2672–2678 (2009)
6. Qian, S.-E., Bergeron, M., Cunningham, I., Gagnon, L., Hollinger, A.: Near lossless data compression onboard a hyperspectral satellite. *IEEE Trans. Aerospace Electron. Syst.* **42**(3), 851–866 (2006)
7. Magli, E., Olmo, G., Quacchio, E.: Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC. *IEEE Geosci. Remote Sens. Lett.* **1**(1), 21–25 (2004)
8. Krishnamachari, S., Chellappa, R.: Multiresolution Gauss–Markov random field models for texture segmentation. *IEEE Trans. Image Process.* **39**(2), 251–267 (1997)
9. Bruce, L.M., Morgan, C., Larsen, S.: Automated detection of subpixel hyperspectral targets with continuous and discrete wavelet transforms. *IEEE Trans. Geosci. Remote Sens.* **39**(10), 2217–2226 (2001)

10. International charter: Space and major disasters. http://www.disasterscharter.org/main_e.html
11. Jet Propulsion Laboratory: AVIRIS free standard data product. <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>
12. Zhang, J., Liu, G.: An efficient reordering prediction-based lossless compression algorithm for hyperspectral images. *Geosci. Remote Sens. Lett.* **4**(2), 283–287 (2007)
13. Aiazzi, B., Baronti, S., Alparone, L.: Lossless compression of hyperspectral images using multiband lookup tables. *Geosci. Remote Sens. Lett.* **16**(6), 481–484 (2009)
14. Wang, H., Babacan, S.D., Sayood, K.: Lossless hyperspectral-image compression using context-based conditional average. *IEEE Trans. Geosci. Remote Sens.* **45**(12), 4187–4193 (2007)
15. Qian, S.-E.: Hyperspectral data compression using a fast vector quantization algorithm. *IEEE Trans. Geosci. Remote Sens.* **42**(8), 1791–1798 (2004)
16. Fowler, J.E., Rucker, J.T.: 3D wavelet-based compression of hyperspectral imagery. In: Chang, C.-I. (ed.) *Hyperspectral Data Exploitation: Theory and Applications*, Chapter 14, pp. 379–407. Wiley, Hoboken (2007)
17. Penna, B., Tillo, T., Magli, E., Olmo, G.: Progressive 3-D coding of hyperspectral images based on JPEG 2000. *IEEE Geosci. Remote Sens. Lett.* **3**(1), 125–129 (2006)
18. Christophe, E., Mailhes, C., Duhamel, P.: Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3D wavelet coding. *IEEE Trans. Image Process.* **17**(12), 2334–2346 (2008)
19. Christophe, E., Pearlman, W.A.: Three-dimensional SPIHT coding of volume images with random access and resolution scalability. *EURASIP J. Image Video Process.* (2008). doi: [10.1155/2008/248905](https://doi.org/10.1155/2008/248905)
20. Cheung, N.-M., Wang, H., Ortega, A.: Sampling-based correlation estimation for distributed source coding under rate and complexity constraints. *IEEE Trans. Image Process.* **17**(11), 2122–2137 (2008)
21. Wang, L., Wu, J., Jiao, L., Shi, G.: Lossy-to-lossless hyperspectral image compression based on multiplierless reversible integer TDLT/KLT. *IEEE Geosci. Remote Sens. Lett.* **6**(3), 587–591 (2009)
22. García-Vilchez, F., Serra-Sagristà, J.: Extending the CCSDS recommendation for image data compression for remote sensing scenarios. *IEEE Trans. Geosci. Remote Sens.* **47**(10), 3431–3445 (2009)
23. Du, Q., Fowler, J.E., Zhu, W.: On the impact of atmospheric correction on lossy compression of multispectral and hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **47**(1), 130–132 (2009)
24. Zhang, J., Fowler, J.E., Liu, G.: Lossy-to-lossless compression of hyperspectral imagery using three-dimensional TCE and an integer KLT. *IEEE Geosci. Remote Sens. Lett.* **5**(4), 814–818 (2008)
25. Carvajal, G., Penna, B., Magli, E.: Unified lossy and near-lossless hyperspectral image compression based on JPEG 2000. *IEEE Geosci. Remote Sens. Lett.* **5**(4), 593–597 (2008)
26. Du, Q., Zhu, W., Fowler, J.E.: Anomaly-based JPEG2000 compression of hyperspectral imagery. *IEEE Geosci. Remote Sens. Lett.* **5**(4), 696–700 (2008)
27. Penna, B., Tillo, T., Magli, E., Olmo, G.: Hyperspectral image compression employing a model of anomalous pixels. *IEEE Geosci. Remote Sens. Lett.* **4**(4), 664–668 (2007)
28. Penna, B., Tillo, T., Magli, E., Olmo, G.: Transform coding techniques for lossy hyperspectral data compression. *IEEE Trans. Geosci. Remote Sens.* **45**(5), 1408–1421 (2007)
29. Tate, S.R.: Band ordering in lossless compression of multispectral image. *IEEE Trans. Geosci. Remote Sens.* **46**(4), 477–483 (1997)
30. Mielikainen, J.: Lossless compression of hyperspectral images using lookup tables. *IEEE Signal Process. Lett.* **13**(3), 157–160 (2006)
31. Qian, S.-E., Hollinger, A., Williams, D., Manak, D.: Vector quantization using spectral index-based multiple subcodebooks for hyperspectral data compression. *IEEE Trans. Geosci. Remote Sens.* **38**(3), 1183–1190 (2000)

32. Motta, G., Rizzo, F., Storer, J.A.: Compression of hyperspectral imagery. In: Data Compression Conference, DCC, vol. 8. IEEE, Mar. 2003, pp. 333–342
33. Ryan, M.J., Arnold, J.F.: Lossy compression of hyperspectral data using vector quantization. *Remote Sens. Environ.* **61**, 419–436 (1997)
34. Ryan, M., Pickering, M.: An improved M-NVQ algorithm for the compression of hyperspectral data. In: IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2000, vol. 2, pp. 600–602 (2000)
35. Implementation du décorellateur multispectral—R&T Compression. Alcatel Alenia Space, Tech. Rep. 100137101A, Nov (2006)
36. Thiebaut, C., Christophe, E., Lebedeff, D., Latry, C.: CNES studies of on-board compression for multispectral and hyperspectral images. In: SPIE, Satellite Data Compression, Communications, and Archiving III, vol. 6683. SPIE, August (2007)
37. Penna, B., Tillo, T., Magli, E., Olmo, G.: A new low complexity KLT for lossy hyperspectral data compression. In IEEE International Geoscience and Remote Sensing Symposium, IGARSS'06, August (2006), pp. 3525–3528
38. Liu, G., Zhao, F.: Efficient compression algorithm for hyperspectral images based on correlation coefficients adaptive 3D zerotree coding. *IET Image Process.* **2**(2), 72–82 (2008)
39. Christophe, E., Duhamel, P., Mailhes, C.: Adaptation of zerotrees using signed binary digit representations for 3 dimensional image coding. *EURASIP J. Image Video Process.* (2007)
40. Tang, X., Pearlman, W.A.: Scalable hyperspectral image coding. In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'05, vol. 2, pp. 401–404 (2005)
41. Cho, Y., Pearlman, W.A., Said, A.: Low complexity resolution progressive image coding algorithm: progres (progressive resolution decompression). In: IEEE International Conference on Image Processing, vol. 3, pp. 49–52 (2005)
42. Information technology—JPEG 2000 image coding system: Core coding system, ISO/IEC Std. 15 444-1 (2002)
43. Bowles, J., Gillis, D., Palmadesso, P.: New improvements in the ORASIS algorithm. *Aerospace Conference Proceedings* **3**, 293–298 (2000)
44. Langevin, Y., Forni, O.: Image and spectral image compression for four experiments on the ROSETTA and Mars Express missions of ESA. In Applications of Digital Image Processing XXIII, vol. 4115. SPIE, 2000, pp. 364–373.
45. Yeh, P.-S., Armbruster, P., Kiely, A., Masschelein, B., Moury, G., Schaefer, C., Thiebaut, C.: The new CCSDS image compression recommendation. In IEEE Aerospace Conference. IEEE, March (2005)
46. Qian, S.-E., Hollinger, A., Bergeron, M., Cunningham, I., Nadeau, C., Jolly, G., Zwick, H.: A multi-disciplinary user acceptability study of hyperspectral data compressed using onboard near lossless vector quantization algorithm. *Int. J. Remote Sens.* **26**(10), 2163–2195 (2005)
47. Rast, M., Bezy, J.L., Bruzzi, S.: The ESA medium resolution imaging spectrometer MERIS - a review of the instrument and its mission. *Int. J. Remote Sens.* **20**(9), 1681–1702 (1999)
48. Wang, Z., Bovik, A.C.: Mean square error: love it or leave it?. *IEEE Signal Process. Mag.* **26**(1), 98–117 (2009)
49. Christophe, E., Léger, D., Mailhes, C.: Quality criteria benchmark for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **43**(09), 2103–2114 (2005)
50. DFTC fusion contest (2008). <http://tlclab.unipv.it/dftc/home.do?id=3>
51. Licciardi, G., Pacifici, F., Tuia, D., Prasad, S., West, T., Giacco, F., Inglada, J., Christophe, E., Chanussot, J., Gamba, P.: Decision fusion for the classification of hyperspectral data: Outcome of the 2008 GRS-S Data Fusion Contest. *IEEE Trans. Geosci. Remote Sens.* **47**(11), 3857–3865 (2009)
52. Information technology – JPEG 2000 image coding system: Extensions, ISO/IEC Std. 15 444-2, (2004)
53. Taubman, D.: Kakadu Software v 6.0 (2009). <http://www.kakadusoftware.com/>

Optical Remote Sensing
Advances in Signal Processing and Exploitation
Techniques

Prasad, S.; Bruce, L.M.; Chanussot, J. (Eds.)

2011, VIII, 344 p., Hardcover

ISBN: 978-3-642-14211-6