Segmentation-Based and Region-Adaptive Lossless Image Compression Underpinned by a Stellar-Field Image Model

Christian Dieter Grünler

A thesis submitted in partial fulfilment of the requirement of
Staffordshire University for the degree of
Doctor of Philosophy

Collaborating establishment:
Baden-Württemberg Cooperative State University (BW CSU),
Stuttgart, Germany

January 2010

To Sarah.

"This lossless compression is mandatory in cases where the images are very precious, as in the case of space imagery, where data is very costly to obtain and must be accurately preserved for scientific exploration" [Gadiel Seroussi[a]]

[a] Gadiel Seroussi, director of the Information Theory Research Group at HP Labs. Source: http://www.hpl.hp.com/news/2004/jan-mar/hp_mars.html (03.03.2007).

Abstract

The central question addressed in this research is whether lossless compression of stellar-field images can be enhanced in terms of compression ratio, by using image segmentation and region-adaptive bit-allocation which are based on a suitable image model. Therefore, special properties of stellar-field images, which compression algorithms could exploit, are studied. The research proposes and develops novel lossless compression algorithms for the compaction of stellar-field images. The proposed algorithms are based on image segmentation coupled to a domain-specific image data model and to a region-adaptive allocation of pixel bits. The algorithms exploit the distinctive characteristics of stellar-field images and aim to meet the requirements for compressing scientific-quality astronomical images. The image data model used is anchored on the property of a stellar-field image encapsulated in the characterisation of this type of images as consisting of "dot-like bright objects on a noisy background". These novel algorithms segment the dot-like bright objects, corresponding to the high-dynamic-range areas of the image, from the noise-like low-dynamic-range background sky areas. Following the segmentation of the image, the algorithms perform region-adaptive image compression tuned to each specific component of the image data model.

Besides the development of novel algorithms, the research also presents a survey of the state-of-the-art of compression algorithms for astronomical images. It reviews and compares existing methods claimed to be able to achieve lossless compression of stellar-field images and contributes an evaluation of a set of existing methods. Experiments to evaluate the performance of the algorithms investigated in this research were conducted using a set of standard astronomical test images.

The results of the experiments show that the novel algorithms developed in this research can achieve compression ratios comparable to, and often better than existing methods. The evaluation results show that a significant compaction can be achieved by image segmentation and region-adaptive bit-allocation, anchored on a domain-specific image data model. Based on the evaluation results, this research suggests application classes for the tested algorithms.

On the test image set, existing methods which do not explicitly exploit the special characteristics of astronomical images were shown to lead to average compression ratios of 1.97 up to 3.92. Great differences were found between the results on 16-bit-per-pixel images and those on 32-bit-per-pixel images. For these existing methods, the average results on 16-bit-per-pixel images range from 1.37 up to 2.81, and from 3.81 up to 6.42 for 32-bit-per-pixel images. Therefore, it is concluded that for archiving data, compression methods may indeed save costs for storage media or data transfer time, especially if a large part of the raw images is encoded with 32 bits per pixel.

With average compression ratios on the test image set in the range of 3.37 to 3.82, the simplest among the new algorithms developed in this research achieved a result which is comparable to the best existing methods. These simple algorithms use general-purpose methods, which have limited performance, for encoding the data streams of separate image regions corresponding to components of a stellar-field image. The most advanced of the new algorithms, which uses data encoders tuned to each image signal component, outperformed existing methods by about 10 percent (average of 4.29 on the test image set), in terms of size efficiency; it can yield a compression ratio of 7.87. Especially for applications where high volumes of image data have to be stored, the most advanced of the new algorithms should also be considered.

Acknowledgements

First of all, I thank my supervisor in Stuttgart, Prof. Dr. Hans Weghorn, for his continuous support and his help. Hans was always there to listen and give advice whenever there were questions and uncertainties. He taught me to ask the necessary questions, express the new ideas and he guided me through the wide open spaces of this topic. Without his spontaneous willingness and his dedication, the special programme that I have taken part in would never have been established. Also, my very special thanks go to my supervisor in Stafford, Dr. Claude Chibelushi, who was always supportive and willing to talk about all ideas, to proofread my written work, and to ask me good questions to help me think through my problems. He showed me ways to approach a research problem and gave insightful comments.

Besides my advisors, I would like to thank Prof. Dr. Bernadette Sharp for her warm welcome in Stafford, her support, the inspiring discussions in Stuttgart and all the necessary information and help which was especially needed in my distance-learning and part-time student situation. I also want to thank her for the interesting lectures she and her colleagues gave during the Research Methods course in September 2004. Also, I want to thank Prof. Dr. Elzmann from the Baden-Württemberg Cooperative State University for his valuable hints for my visits in Stafford and his support for this programme.

I also want to acknowledge the support given by ESO/ESA, for my participation in the Astronomical Data Analysis and Software Systems Conference 2005 in El Escorial, Spain. I want to thank them for a range of interesting talks, discussions and demonstrations as well as for the encouraging large number of people who were interested in my research.

I also want to acknowledge Prof. Dr. Fionn Murtagh (from Royal Holloway, University of London) who granted me access to the MR/1 multiresolution package to test a range of existing compression algorithms. My work has depended on the generosity of authors of free software, I am grateful to them for granting free access to their software.

Also thanks to the staff at the Baden-Württemberg Cooperative State University for their helpfulness. I also want to thank my former room mate at the Baden-Württemberg Cooperative State University in Stuttgart, Heinz Wolfram, for all the jolly discussions and the coffee breaks which kept me in good spirits.

I also want to thank my family and parents for their support. Last, but not least, I thank Sarah my wonderful friend, for her encouragement to pursue my interests, for her support, her patience, for listening to my complaints and frustrations, for believing in me, and for her love.

Contents

List of Tables

List of Figures

List of Abbreviations

BWT	Burrows-Wheeler Transform
CCD	Charge-Coupled Device
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
ESA	European Space Agency
ESO	European Southern Observatory
FITS	Flexible Image Transport System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
JPEG	Joint Photographic Experts Group
JPEG-LS	Low complexity lossless JPEG
LSB	Least Significant Bit
LZ	Lempel-Ziv
MP3	MPEG 1 Audio Layer 3
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
NASA	National Aeronautics and Space Administration
PBM	Portable Bitmap
PLIO	Pixel List I/O
PMT	Pyramidal Median Transform
PNG	Portable Network Graphics
RLE	Run Length Encoding
SHC	Signed Huffman Coding
SPHIT	Set Partitioning in Hierarchical Trees
TCP/IP	Transmission Control Protocol over Internet Protocol
UDP	User Datagram Protocol
VO	Virtual Observatory

Chapter 1

Introduction

Astronomy provides exciting new findings, which touch and explain fundamentals of human origin and life. Modern research in astronomy and its advances depend heavily on raw data that is collected with complex instruments like huge telescope systems or even spacecrafts. Since no single university or research institute can afford operating such facilities, independent service institutions like the European Southern Observatory[1] and the European Space Agency[2] were founded for operating huge telescope plants (in the Chilean desert mountains, for example), or for performing space-based missions for astronomical observations. Presently, there are ongoing preparations to set up observation stations in the Antarctica. With expenditures in the order of a billion euros a year, these activities in the northern and southern hemisphere are financed by tax proceeds from the European Union. This gives sufficient reasons for allowing access to the collected astronomical raw data by a wider spectrum of users than only a selected community of researchers.

The stakeholders – consisting of the general public, professionals from other fields, hobby researchers, and also education institutions at any level, like schools – have the right to access electronically this valuable data.

Unfortunately, astronomy faces the problem of high and even accelerating growth in the amount of the collected raw data (Seaman et al. 2006). This growth makes the operation of all-embracing public archives a complex problem. In particular during recent years, a dramatic change in the generation, storage, and access of data has taken place in astronomy.

In the past, astronomical data used to be collected with a minimal degree

[1] http://www.eso.org (10.08.2009)

[2] http://www.esa.int (10.08.2009)

of automation. Data of a certain astronomical object of interest were collected for a certain purpose and institution, on demand. While in the past observation tasks had to be scheduled and observers sometimes had to wait for months or years until access to a telescope was granted, now, generation of astronomical data has turned into mass production (Szalay et al. 2000). Automated survey telescopes image the entire sky every night and image data are often generated continuously in an automated process at observatories and telescope facilities. Furthermore, data sets are available at virtual observatories. These sources of primary data are linked to grid-like, geographically distributed sky archives (Golombek 2004, Djorgovski 2003, Szalay et al. 2000). The advent of virtual observatories, which provide high-quality astronomy data, will enrich astronomy-oriented projects (Quinn and Gorski 2004).

Data, for use by professional observers and by other people interested in astronomy, are stored at different locations. About half of all astronomical data in the world is at least in principle "public" (Szalay et al. 2001), ready to be analyzed and accessed by astronomers all over the world. With this, the data for some specific scientific investigation typically are available at the virtual observatory. The request for certain data can be fulfilled quickly and without waiting until access to a telescope is granted.

1.1 Context

Virtual observatories require huge data processing and storage capacity for various reasons. First, the number of large telescope facilities has grown dramatically during the past few years (Quinn and Gorski 2004). Also the resolution of imaging devices (such as charge-coupled device cameras) has been increasing continuously, as detectors with higher resolutions become available (Howell 2000). In addition to better spatial resolutions, astronomical projects are no longer limited to one or two wavelengths. Projects typically involve observations of different wavebands, including gamma-rays, X-rays, optical, infrared, and radio frequencies (Szalay et al. 2000). These observations will provide a better understanding of the physics underlying the observed phenomena, but they will also provide more data than ever before (Golombek 2004). Furthermore, survey telescopes currently being planned will image the entire sky continuously, and therefore generate Petabytes of image data. The growth of the amount of astronomical data even exceeds

Moore's law; the size of the data is doubling every year (Szalay et al. 2001). Despite the falling prices of disk space and network bandwidth (costs per Gigabyte), high annual investments for replacing and enhancing storage and network capacity have to be made. It can be argued that an international virtual observatory with interoperating data centres could only become a success if problems of limited storage capacities and limited bandwidth would be solved. The costs for online storage, archives, and bandwidth could be reduced to a fraction of the current annual costs by using efficient – in terms of the compression-ratio that they achieve – compression methods for the astronomical data sets.

Antu telescope (ESO, Paranal)

Isophot infrared satelite (ESO)

3.6 m Telescope (ESO, La Silla)

Hubble Space Telescope (NASA)

Large Telescopes (ESA, La Silla)

SOFIA airborne infrared telescope (DLR, NASA)

Figure 1.1: Different sources of astronomical images: ground- and space-based telescopes, including equipment for gamma-ray, X-ray, and infrared data collection (images courtesy of ESA and NASA).

Unfortunately, the requirements for compressing astronomical images differ greatly from compression requirements for the purpose of entertainment. For the latter, a loss caused by compression is often accepted for primary data — as is the case, for example, in every digital camera that creates a

JPEG image. In most cases this is not even noticed by the user.

The requirements of different users of astronomical images vary, and for some users lossy compressed astronomical data can be adequate. In his paper discussing the communication of images from the "New Generation Astronomical Telescope", McNerney (2000) identifies three categories of users and their requirements for astronomical images: the general public, schools and colleges, and professional astronomers. The general public is mainly interested in the appearance of astronomical objects on preferably colourful posters. Schools and colleges require less accurate data than professional astronomers.

Scientific data analysis usually requires the best quality of data. No information loss can be accepted for many medical and astronomy archiving applications, for example. In addition, lossless methods are often needed for legal reasons, and for backing up statistical databases and experimental results (Moffat et al. 1997). The possibility of verifying scientific investigations also drives the need for lossless compression in astronomy. Moreover, lossless compression is required, because the future value of images is not known and images are still obtained at high costs.

Professional astronomers often insist that they can only accept lossless compression for the captured photographs (White 1992); in general, they require images of high photometric and astrometric accuracy. The need for lossless compression is also driven by the fact that astronomers are often disinclined to run their precious data, collected during observations, through an algorithm which can introduce irreversible artefacts. Still, even for astronomers, lossy compression is an option for fast previewing and selecting images over networks like the Internet. Nevertheless, careful scientific explorations have to be verified on undegraded data.

Hence, the work presented in this thesis focuses on the lossless compression of original data acquired directly from a telescope, because only this data will allow the verification of scientific findings. Although they offer only moderate compression rates, lossless techniques guarantee reversible decoding (Penrose and Dodgson 1999). The research which is presented here focuses on the compression of stellar-field images, like the Hubble deep field image (Figure 4.2), which are a common type of astronomical images. A typical stellar-field or galaxy image consists of a nearly flat background sprinkled with point sources and occasional extended sources (White and Percival 1994). A significant area of the image is filled exclusively by the background sky (Puetter et al. 2005). Close-ups are, compared to the other types, rarely used for scientific applications although these images are often shown, sometimes even coloured, on posters and in presentations geared towards the general public.

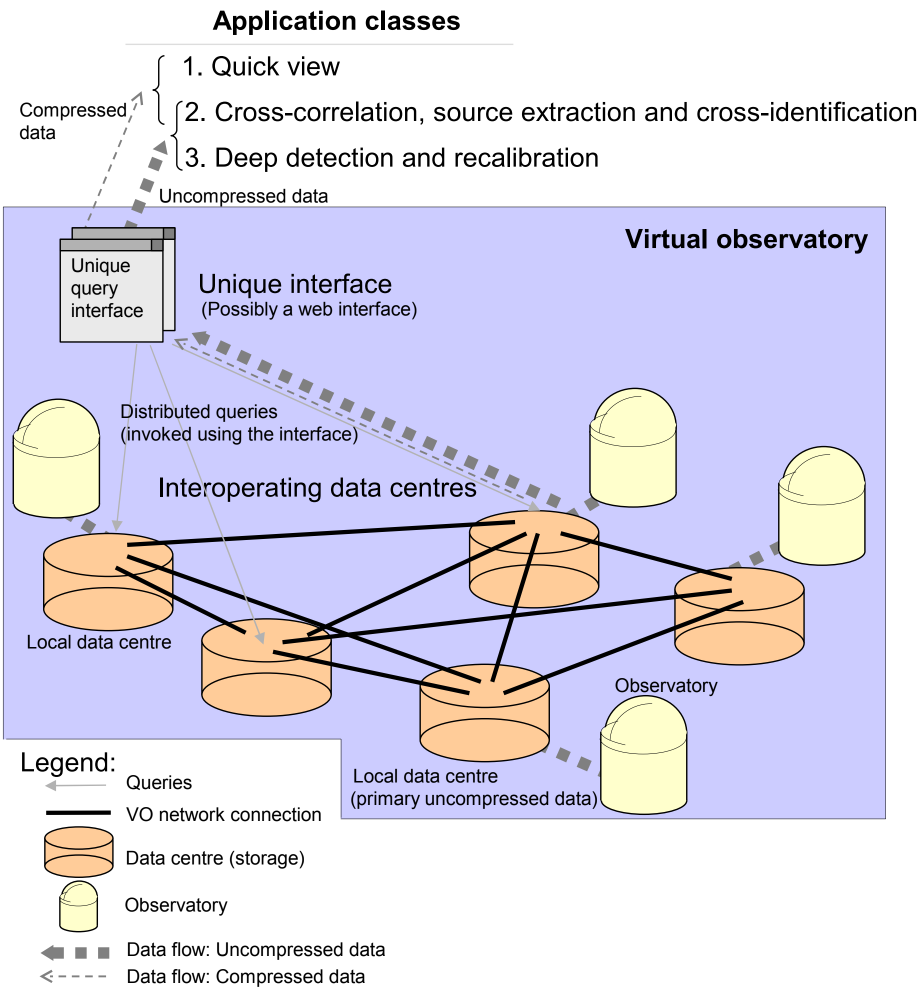

Application classes
1. Quick view
2. Cross-correlation, source extraction and cross-identification
3. Deep detection and recalibration
Compressed data
Uncompressed data
Virtual observatory
Unique query interface
Unique interface
(Possibly a web interface)
Distributed queries
(invoked using the interface)
Interoperating data centres
Local data centre
Observatory
Local data centre
(primary uncompressed data)
Legend:
Queries
VO network connection
Data centre (storage)
Observatory
Data flow: Uncompressed data
Data flow: Compressed data

Figure 1.2: Data flow model for virtual observatories (application classes of (Louys, Starck, Mei, Bonarel and Murtagh 1999)).

A virtual observatory consists of sources of primary data, which are linked to a grid-like, geographically distributed sky archive (Golombek 2004, Djorgovski 2003). Data are accessible from all observatories, or even from the Internet and may be stored at different locations (Szalay et al. 2001). The data are generated at observatories and telescope facilities all over the world and stored in centres associated with the observatory (Figure 1.2). Via a computer interface, an end-user may invoke a query for data of interest, which generates distributed sub-queries to data archives all over the world. Based on the application types given in (Louys, Starck, Mei, Bonarel and Murtagh 1999), different application classes for the usage of astronomical image data may be defined with respect to the compression requirements:

1. **Quick preview:** Lossy compressed data can be used for quick inspection and navigation through vast image repositories.

2. **Cross-correlation for catalogue overlay, source extraction, and cross-identification:** Lossy or lossless compressed data may be used depending on data properties and the precision requirements of astronomical examinations.

3. **Deep detection and recalibration:** Lossless compressed data has to be used if high precision requirements for astrometric and photometric applications exist or recalibration is needed.

Depending on the requirements of the user, it should be possible to retrieve data in a compressed or uncompressed format. Non-lossy compressed primary data should be available for application classes 2 and 3 – even if it is too large for online access. After the selection of data, and possibly first tentative examinations on lossy compressed data, only the lossless compressed data of interest has to be retrieved. In that case, a distribution on storage media is possible as well, especially for the verification of new findings.

The investigation presented in this thesis focuses on the lossless archival or transfer of primary data. Unprocessed astronomical data only contains integer values, due to the digitisation process. Hence, lossless compression techniques for raw integer images are evaluated in this investigation. In order to meet the key requirements for the archival of primary data, for this research, five main requirements have been identified:

1. **Losslessness:** This has to be guaranteed for the archival of scientific-grade primary data.

2. **Size-efficiency:** Compression algorithm should lead to high compression ratios for astronomical images.

3. **Reliability:** Compression algorithms have to work with both 16-bit-per-pixel and 32-bit-per-pixel integer data reliably.

4. **Adaptation to data properties:** The algorithms should take advantage of the special properties of astronomical images and exploit them as much as possible.

5. **Computational efficiency:** High data rates close to disk I/O and the rate at which the CCD data can be read out, have to be achieved with the compression algorithms to keep up with the data rate that is generated by the telescope. Computational complexity should also be reasonable for online storage of large unsplit images.

The research work presented in this thesis aims to fulfil the first four requirements. Due to the long exposure times which are required to capture astronomical images, the computational complexity of the algorithm is typically less important than its size-efficiency. Also, the computational performance of a compression method depends heavily on the hardware platform (classical single or multiple processor versus pipeline processors, application-specific integrated circuits, or FPGAs, ...) and the software implementation which is used. Therefore, no run-time efficiency measurement is performed in the investigation reported herein.

1.2 Research Questions

The central research question addressed in this work is: Can lossless compression, using segmentation and region-adaptive bit-allocation which are based on image modelling, be applied effectively (in terms of enhancing the compression ratio) to astronomical stellar-field images? A subsidiary question also considered on the way towards an answer to the central research question is: What contribution may existing data compression methods make to reduce the size of astronomical image data?

1.3 Aim of the Investigation

This research aims to demonstrate that an efficient compression technique, based on image modelling, segmentation, and region-adaptive bit-allocation, meets the requirements of astronomical image storage and transmission. Special characteristics of astronomical stellar-field images should be exploited.

Furthermore, the research aims to identify and evaluate existing compression methods which are capable of fulfilling the requirements for the archival of astronomical image data.

1.4 Objectives of the Investigation

The objectives of the research are:

1. To perform a thorough literature review of existing compression methods and identify their strengths and limitations.
2. To conduct a thorough evaluation of some existing methods and analyse the results.
3. To find a suitable model that describes the properties of astronomical images. This model shall form the foundation for the development of segmentation-based and region-adaptive compression algorithms.
4. To develop region-adaptive compression algorithms based on image segmentation and modelling, and implement them into a software prototype to measure achievable compression ratios. Lossless compression techniques shall be used to compress all the information contained in the images (Weghorn 2002).
5. To evaluate the compression ratios of the algorithms developed in this research. For this investigation, a representative sample of standard images shall be used, possible data sets are provided by (ESO 2004, IRAF 2004).

1.5 Contribution to Knowledge

The thesis extends current knowledge by contributing a study of special properties of astronomical images and of the data redundancy within stellar-field images, which compression algorithms could exploit. The thesis also contributes a set of original methods for astronomical image compression based on a second-generation image compression approach.

In addition, the thesis contributes the findings of experiments where the novel methods developed in the thesis work are evaluated against existing lossless compression algorithms, which can be applied to the primary data of virtual observatories without irreversible distortion. The thesis suggests application classes for the tested algorithms. The thesis also presents a survey of the state-of-the art of compression algorithms for astronomical images.

1.6 Brief Description of the Methodology

This study follows an experimental method. Research questions and the corresponding hypotheses are formulated. Existing methods are reviewed and novel compression algorithms addressing the research questions are developed. Implementations of the novel algorithms are evaluated based on controlled experiments and compared to existing methods.

First, a literature survey of existing compression methods, especially lossless methods, is conducted. Then, special statistical properties of astronomical images are determined and described. Based on the analysis of astronomical image data, a model of a typical astronomical image signal is established. On the basis of this model, algorithms to compress astronomical images are developed and tested in controlled experiments.

During the development of the novel algorithms, specific properties of astronomical data are accounted for. Efficient encoding methods, for the different data components of astronomical images, are studied, and evaluations are conducted to determine the encoders best matched to the distinct data components within the image. Compression ratios achievable by the developed algorithms are evaluated.

The performance – in terms of achievable compression ratio – of the newly developed algorithms is compared to existing methods identified in the literature survey. Finally, application classes for both, existing methods and the newly developed algorithms, are defined.

Further methodological details relating to the experimental designs, materials, and procedures are given in Chapter 6, which reports the performance evaluation of the compression techniques developed in this research and of existing compression techniques.

1.7 Thesis Structure

This thesis consists of seven chapters. After this introduction chapter, Chapter 2 presents an overview of the theoretical foundation of all compression methods, information theory and a short systematic overview of methods used in data compression. A range of existing methods for data encoding, which are related to this work, are discussed. Finally, image compression methods coupled to properties of visual data are discussed in detail.

Chapter 3 presents a literature survey of compression methods which can be used for astronomical images. Existing state-of-the art methods for lossy compression, methods that allow lossless or lossy compression and purely lossless methods are described in detail. Also, the standard for the compaction of images in FITS format, a common file format for astronomical images, is presented. The methods are categorised according to the decorrelation step. This categorisation is derived in Chapter 2.

Then, an overview of astronomical images and image categories is given in Chapter 4. Models of astronomical stellar field data are discussed based on mathematical models for CCDs and stellar field image properties. An image data model used for the segmentation in this research is presented and its relation to other astronomical image data models is discussed. This image data property model forms the basis for the new compression algorithms. Detailed knowledge of data properties is required to adapt the novel compression algorithms closely to the properties of stellar-field images. Finally, a first sample calculation presented in the same chapter shows that a significant data reduction is possible by decomposing the image into its basic components, according to the model proposed.

Chapter 5 introduces three new compression algorithms developed in this research project, which are based on image segmentation and region-adaptive bit-allocation and which are underpinned by a stellar-field image model. They can be considered as second-generation image compression methods for astronomical images. The three algorithm versions represent algorithmic options, each with its own trade-off between complexity and optimality of compression. Details of each algorithm are given, the choices of encoding methods and segmentation parameters are discussed. As a successive improvement – towards more optimal compression ratio – the three algorithmic options have been conceived in a phased development.

To determine the size-efficiency of existing methods and newly developed compression algorithms, a performance evaluation is given in Chapter 6. A set of test images is presented which is used for the evaluation of both existing methods and the novel algorithms reported in this thesis. Existing methods have been – whenever possible – evaluated using the standard set of astronomical test images to determine their size-compaction efficiency. The systematic comparison of existing methods alone will be useful to users of astronomical image data. Despite of the range of different existing methods available, these are rarely applied in practice.

Chapter 6 also contains the evaluation of the novel compression algorithms. Results from both the evaluation of existing methods and of implementations of the novel algorithms are compared and the size-efficiency of the novel algorithms is established.

This thesis ends with Chapter 7, which highlights the key conclusions, summarises findings for the different algorithms developed and assessed experimentally, and discusses possible future work.

Chapter 2

Information Theory and Compression

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point."* (Shannon 1948)

The main aim of this chapter is to provide an overview of existing compression methods, which are related to the work developed in this research, and to present their theoretical foundation, which is information theory. This chapter consists of three parts.

Section 2.1 presents a short summary of concepts which are relevant for the research presented here.

Section 2.2 presents a short overview of methods used in data compression, which categorises the methods and their purposes. The focus of the section is on clarifying the difference between encoders, decorrelation, and data reduction methods. Samples for all categories of such methods are given, and their properties, advantages, and disadvantages are discussed. The overview also presents an existing architecture of compression methods (Figure 2.2) and assigns a range of modern MP3 and MPEG-4 algorithms (Pereira and Ebrahimi 2002, Sikora 2003) to it. Different decorrelation methods are critically reviewed and the discussion highlights that for lossless compression of scientific-grade stellar-field images, a transform-based decorrelation step is not a good option. The systematic overview also shows that a close coupling of data properties and compression is essential, especially in the case of lossless compression.

Finally, a review of some recent developments in image compression is given in Section 2.3. It highlights two main developments in the area of compression, the closer-and-closer adaptation of methods to their application class, and the trend from pixel and pixel-difference-based methods towards exploiting higher-order redundancy using context adaptive methods, such as segmentation-based pre-processing.

2.1 Foundation of Data Compression Methods

This section, presenting the foundation of data compression methods, highlights the importance of data models for size-efficient compression. Also, this section introduces and defines basic compression-related terms, and discusses data correlation and redundancy, which are relevant for the choice of encoders of the new algorithms.

2.1.1 Definition of Basic Compression Terms

Prior to discussing compression techniques in detail, the required terms have to be defined precisely. The definition of the terms and the notation are inspired by various books on data and image-compression (Salomon 2000, Sayood 2000, Strutz 2005), in order to follow an easily understandable "standard".

Definition 1 (Bit rate) *The average number of bits required to represent a letter with a symbol is called the bit rate. For a given amount of data D (in bits) consisting of N symbols, the bit rate is:*

$$r = \frac{D}{N} \text{ [bit/symbol]} \tag{2.1}$$

For images, the bit rate is typically given in bits-per-pixel [bpp].

Definition 2 (Compression ratio) *The compression ratio c is used to describe the reduction of the size of the data. The compression ratio is the quotient of storage required before and after compression.*

$$c = \frac{\textit{Original data size}}{\textit{Compressed size}} \tag{2.2}$$

Compression ratios $c > 1$ indicate a reduction of the required storage space, while compression ratios $c < 1$ indicate an enlargement rather than a compaction of the data. Often the compression ratio is written in the form c:1.

Definition 3 (Reconstruction error) *The reconstruction error e, ocassionally called residual, is the difference between the original data o and the reconstructed data after compression. In Equation 2.3, $C(x)$ stands for the compression operation, $D(x)$ for the decompression operation.*

$$e = o - D(C(o)) \tag{2.3}$$

Definition 4 (Near-lossless compression) *With a near-lossless compression method $C_n(x)$, the maximum of the reconstruction error e is limited to a predefined small value ϵ. Equation 2.4 holds for near-lossless compression methods; $max(x)$ stands for a function that determines the maximum:*

$$max(\,|o - D(C_n(o))|\,) \leq \epsilon \tag{2.4}$$

For example, JPEG-LS (Weinberger et al. 2000) provides such a near-lossless mode of operation.

2.1.2 Information and Entropy

> *"First, this entropy is not known and depends heavily on the model used for the source, i.e. the digital image."* (Kunt et al. 1985)

Information theory, introduced by Shannon (1948), provides the theoretical basis for work related to data compression by defining the amount of information conveyed in a message from a certain source. This amount of information is commonly referred to as entropy H (Strutz 2007). The term entropy originates from thermodynamics (Rao and Yip 1990, Johnson 2003).

In information theory, information can be regarded as the amount of surprise a certain message conveys. The information $I(l_i)$ contained in a single letter l_i of a message is defined by Shannon (1948) as follows: the larger the probability $P(l_i)$ of a letter l_i within a message is, the less information it conveys when it occurs.

$$I(l_i) = log_2 \frac{1}{P(l_i)} \text{ [bit]} \tag{2.5}$$

First-order entropy (H_1), the most common and simple entropy measure, is the theoretical minimum for the code length when only frequencies of letters are taken into account (Johnson 2003). The first-order entropy of a message $H_1(\mathbf{l})$ is calculated from the weighted sum of the individual letter information $I(l_i)$, where the weights are their probability of occurrence $P(l_i)$ (Shannon 1948):

$$H_1(\mathbf{l}) = \sum_{i=1}^{k} P(l_i) \cdot I(l_i) = \sum_{i=1}^{k} P(l_i) \cdot log_2 \frac{1}{P(l_i)} \text{ [bit/symbol]} \tag{2.6}$$

A system with large entropy is characterised by a large amount of uncertainty and randomness, thus a larger amount of information is required to describe it. In the case of first-order entropy, this can be derived from Equation 2.6.

Entropy coding is based on Shannon coding theory (Shannon 1948), where Shannon proved that the average code length $\overline{s}$ is always greater than or equal to the first-order entropy of the message. Additionally, Shannon proved that there is always an encoding of the message where the average code length $\overline{s}$ is smaller than $H_1 + 1$ (Strutz 2005).

The entropy of a certain source is determined using a model of the source. First-order entropy H_1 makes use of a simple model of the source; it is based on the assumption that all symbols are statistically independent and occur with a certain probability (Shannon 1948). This is not the case with most signals; hence first-order entropy is not an optimal measure to determine the amount of information for a range of everyday signals. Based on "typical" English texts, Shannon (1951) measured the first-order entropy of English to be about 4.1 bit/character. He also defined a second-order entropy by treating digrams as single symbols.

These higher order entropies use a joint density function for the data, thus the dependency among successive data values (second-order entropy) or more data values is accounted for. The joint entropy of two data values $\mathbf{l_1}$ and $\mathbf{l_2}$ is defined as (Hunt 1998):

$$H_2(\mathbf{l_1}, \mathbf{l_2}) = -\sum_{i=1}^{k}\sum_{j=1}^{k} P(l_{1,i}, l_{2,j}) log_2(P(l_{1,i}, l_{2,j})) \tag{2.7}$$

It can be shown that the second-order entropy is generally lower than, or in the case where both data values are independent, equal to, the sum of first-order entropies (Hunt 1998).

$$H(\mathbf{l_1}, \mathbf{l_2}) \leq H(\mathbf{l_1}) + H(\mathbf{l_2}) \tag{2.8}$$

This higher-order entropy leads to an entropy of 3.3 bits/character (using trigrams) for the English texts mentioned earlier (Shannon 1951). This entropy definition takes basic dependencies between different elements into account.

How efficiently data of a certain source can be compressed depends largely on the knowledge of the source and the properties of the source (Kunt et al. 1985). The better the model is and the more special statistical properties the source has, the better the compression ratios that can be achieved. While Huffman coding (Huffman 1952) leads to optimal compression of data where each letter is generated independently, it will not work optimally on data with a large dependency among neighbouring characters. Other compression techniques, like run-length encoding (RLE) for example, are based on a model that works much better for data with a strong dependency between neighbouring characters, e.g. a bi-level image. Shannon (1948) takes data correlation into account by defining the entropy as the limit of higher-order entropies as (notation based on (Sayood 2000)):

$$H(S) = \lim_{n \to \infty} \frac{1}{n} G_n \tag{2.9}$$

where

$$G_n = -\sum_{i_1=1}^{k}\sum_{i_2=1}^{k}\cdots\sum_{i_n=1}^{k} P(l_{1,i_1}\cdots l_{n,i_n}) \log P(l_{1,i_1}\cdots l_{n,i_n})$$

Two papers published by Jean-Luc Starck and Fionn Murtagh deal with the astronomical image signal and the definition of entropy (Starck et al. 2001, Starck and Murtagh 2001). They give criteria for a good "physical" definition of entropy. According to (Starck et al. 2001, Starck and Murtagh 2001) the two images shown in Figure 2.1 have the same entropy using any of the standard entropy definitions. The work developed in this thesis does not follow this argumentation, since this argument is only true if none of the higher-order entropy definitions is considered. Both images may share the same first-order entropy. Higher-order entropy definitions take the dependency of different values into account and lead to a lower entropy estimate for the left image.

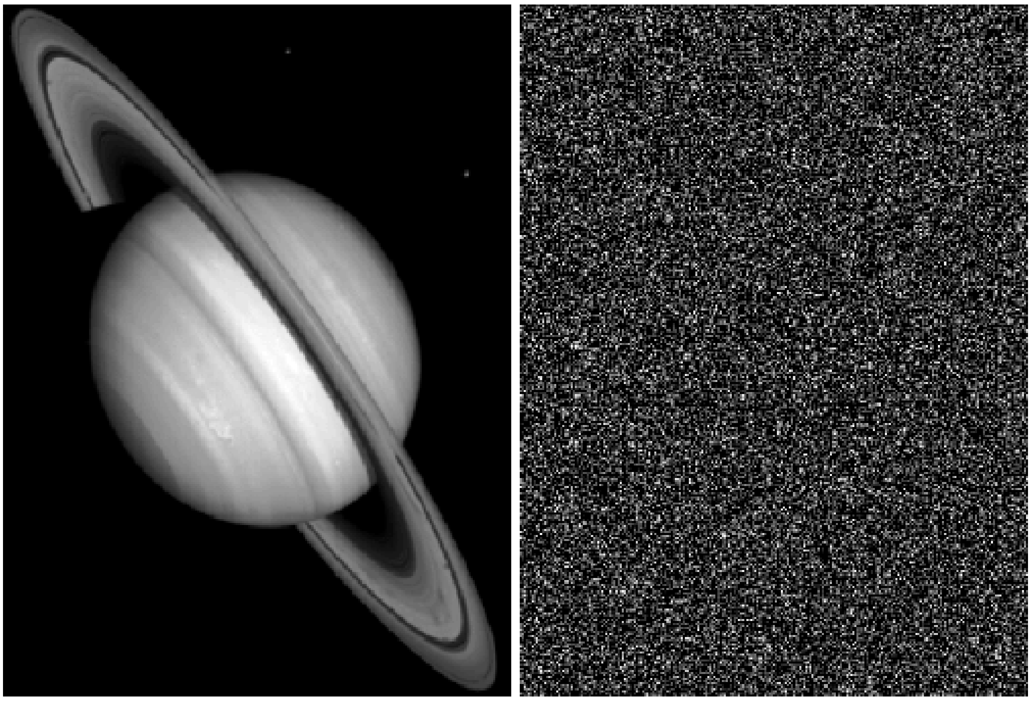


Figure 2.1: Starck and co-workers state that these two images have "the same entropy using any of the standard entropy definitions". This thesis argues that the entropy of the left image is lower than that of the right image (image from Starck et al. (2001)).

Starck et. al. present their own definition of multiscale entropy which is intended for the application in image restoration and enhancement (Starck et al. 2001, Starck and Murtagh 2001). Their entropy definition is not intended to provide information on the compressibility. According to the multiscale entropy definition of Starck et. al., the information content of the image on the left is higher than that of the image on the right (Starck et al. 2001, Starck and Murtagh 2001).

The work developed in this thesis is based on the assumption that the true information content of the image on the left indeed is lower than the first- and second-order entropy estimate. There is a high spatial coherence of intensity values within astronomical images, especially within stellar field images.

Although proposing a different entropy measure for another field of application, Starck and Murtagh (2001) are aware of this, as they mention "for someone working on image transmission, it is clear that the second image will require more bits for lossless transmission, and from this point of view, he/she will consider that the second image contains more information" (Starck et al. 2001, Starck and Murtagh 2001). This is also supported by the theoretical results presented in the work of Gupta and Virdi (1989), which leads to the results that the information content depends on the size of the field of view and the objective aperture, the mean level of star brightness, the spread of brightness values, and the average star density.

Segmentation, which is investigated in this thesis as a pre-processing step before the usage of encoders, should exploit the spatial coherence and the high differences in the dynamic range of the image regions which constitute stellar-field images.

2.1.3 Redundancy Types and Data Correlation

Data compression methods exploit redundancy which results from correlated data. This discussion aims to establish a linkage of spatial redundancy, entropy, and correlation in order to boost the discussion of Figure 2.1 in the previous section.

Compression methods for still images may exploit two types of redundancy: spatial and spectral redundancy. Spatial redundancy results from spatial correlation, some form of dependency or predictability among successive image pixels (Netravali and Limb 1980). Spectral redundancy is termed as the "correlation between different colour planes or spectral bands" by Saha (2000). In audio and video applications additionally temporal redundancy can be found between successive samples of a digital audio file or different frames of video data.

In statistics the linear correlation coefficient is used as a measure to describe the strength and direction of a linear relationship between two series X and Y of n values written as x_i and y_i with $i \in 1, 2, \ldots, n$ (Freedman et al. 1998).

$$R_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{2.10}$$

$\bar{x}$ and $\bar{y}$ are the means of the two series X and Y. s_x and s_y are their sample standard deviations. Values close to plus one are considered as strong positive correlation, values close to minus one as strong negative correlation. The correlation coefficient given in Equation 2.10 can be applied to image information in different ways. This measurement can be done for two successive or two arbitrarily chosen image rows or columns. Alternatively the dependency of intensity values of a single image data line can be examined. In (Salomon 2006) a general measure for such cases is given:

$$R^{(k)} = a_1 a_k + a_2 a_{k+1} + \ldots + a_{n-k} a_n. \tag{2.11}$$

$R^{(k)}$ gives the correlation between standardized values a_i of one series separated by a distance of k units (Salomon 2006). With $k = 2$, the measure given in Equation 2.11 can be applied to determine the linear correlation between two successive pixels. Correlation, which can be expressed at a certain degree with Equation 2.11, is considered as a measure for redundancy in the data, although this measure does not account for non-linear dependencies or higher-order dependencies.

For example, if the linear-correlation of neighbour pixel intensities of an image row is measured, no information on the dependency to intensity values of other neighbour pixels is obtained.

As the intensity difference of neighbour pixels in the left image in Figure 2.1 is often less than the intensity difference in the right image of Figure 2.1, the correlation measure $R^{(k)}$ in the scrambled right image will be much lower than in the original left version. The lower redundancy in the scrambled image is equivalent to a higher information content, which is not detected if only the first-order entropy of both images is considered.

2.2 Systematic Overview of Data Compression Methods

Starting from lossless methods which exploit basic data properties, the growing need for compression during the past twenty years has brought about many compression methods. Especially due to the amount and diversity of recently developed lossy compression methods, an overview of all methods has become more and more difficult. Many of the currently developed compression methods are application specific, especially those for image, audio and video applications. This section is an attempt to provide a short, but systematic overview of data compression methods and concepts applied.

The presentation mainly follows the taxonomy of (Strutz 2005, Strutz 2007), which is intended to be a guideline for researchers and developers of new compression systems. The main intention here is to identify and describe functional blocks of compression algorithms and understand the basic architecture of both modern lossless and lossy methods. Three functional blocks are distinguished:

- Coding: Strutz (2007) summarises techniques aiming at reducing the redundancy in the signal as coding techniques which are fully reversible. Coding techniques which follow this definition are: basic entropy coding, arithmetic coding, run-length coding, dictionary coders, and block-sorting methods like the Burrows-Wheeler Transform (BWT) (Burrows and Wheeler 1994). The amount of signal specific properties that these methods exploit is very limited.

- Data reduction: In contrast to coding and decorrelation methods, in this context data reduction refers to quantisation or sampling methods, which aim to remove non-essential parts from the signal (Strutz 2007).

- Decorrelation: Decorrelation techniques aim to concentrate the information conveyed in the signal into very few signal values (Strutz 2007). A data decorrelation step is included in many modern, special-purpose data compression algorithms. Decorrelation is achieved through techniques such as transforms, like the Karhunen-Loève Transform or the Discrete Cosine Transform (DCT), and predictive coders.

Coding, data reduction, and decorrelation methods are the main building blocks of modern compression methods. These building blocks are based on assumptions about the signal, in terms of a signal model and sets of parameters (Barthel 2003, Strutz 2005). Normally, compression is size-efficient if those assumptions hold true, which means that the compression method is well adapted to the data properties. The compression result may become worse if data parameters change (Strutz 2007). Therefore, adaptive methods have been developed to account for such property changes. Adaptation may be applied to coding, decorrelation, and data reduction. A well-known adaptive coding method is Adaptive Huffman coding, independently developed first by Faller (1973) and later by Gallager (1978), then finally enhanced by Knuth (1985). Adaptation may be as well applied to the decorrelation step in order to enhance its performance.

For example, with segmentation-based compression methods, such as the one developed in this work, model parameters and data processing are changed based on the membership of data to certain data components. Distinct pre-processing, decorrelation, and encoding steps for each data component may be better adapted to the specific data component than a single method which can either be tuned perfectly to the properties of one or the other data set. Finally, adaptation may also be applied to the data reduction step for lossy compression. For example, it was already proposed for region-adaptive medical image compression (Gokturk et al. 2001), where important regions can be compacted losslessly while a loss of some image information is tolerated in less significant image areas.

The two basic building blocks, coding and decorrelation, can be found in virtually any data compression method. With lossless methods, data is encoded and decoded without degradation, in a way that results in an image which is identical to the original image.

As lossless methods do not allow any data to be lost during compression, no data (even if considered non-essential) can be removed in a data reduction step. If any part of a compression system applies a data-reduction method, the compression algorithm is normally called lossy due to the fact that data reduction discards some digital information (Strutz 2007). But, with the astronomical image compression method Fitspress (Press 1992) and the pyramidal median transform method of Starck et al. (1995), in the lossless mode, first data reduction is applied and later the removed data is recovered by storing a difference image. It captures the image data value differences generated during the compression step.

With lossy compression schemes, redundant and non-essential information can be lost. The user sometimes is able to influence the trade-off between compression and data quality.

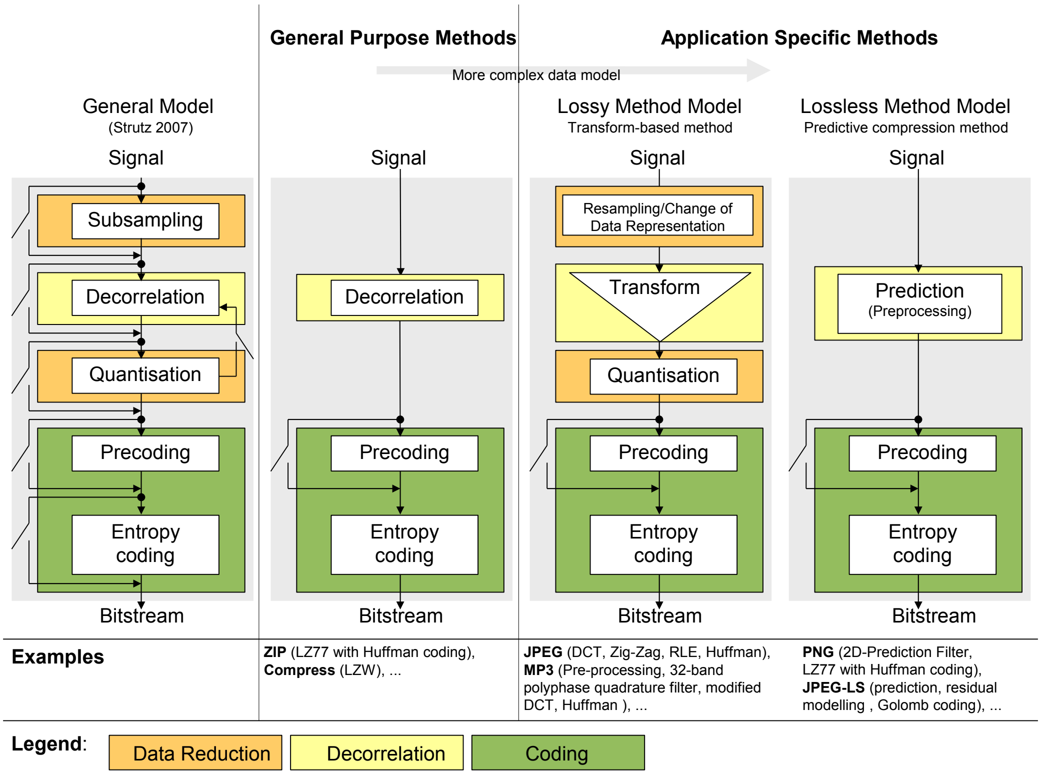


Figure 2.2: Attempt to structure the functions of different compression algorithms according to the components defined by Strutz (2007). The switch symbol next to a range of the data blocks means that these blocks can be bypassed or may not be present in some algorithms.

Figure 2.2 shows a general data flow for a compression system (in the leftmost column). It may serve as a sample for a modern compression system. Accordingly, Figure 2.2 presents a typical data flow for a general-purpose compression method, such as the ones used in Zip or Compress. Also, the figure includes a typical data flow for a transform-based application-specific compression method such as MP3 and JPEG. Finally, the rightmost column gives an outline of a modern lossless compression method. It resembles the predictive methods applied in contemporary general-purpose image compression methods, like the one used with the PNG format and the JPEG-LS compression method.

2.2.1 Encoding Methods

An elaborate review on encoding methods such as Huffman, Lempel-Ziv Welch, Arithmetic-Coding, and Run-Length coding can be found in (Vemuri et al. 2002). Detailed descriptions are found in the publications given below and in (Salomon 2000, Sayood 2000, Strutz 2005).

The most common coding methods are entropy coding methods. These methods reduce the coding redundancy by exploiting the letter distribution, using variable length codes. Variable length codes have to satisfy the Kraft inequality (Kraft 1949, Fano 1949, McMillan 1956). On the basis of the alphabet size and the codeword lengths, it provides the information whether or not a code is uniquely decodeable. The widely used Huffman algorithm can be proven to generate optimal variable length codes with an integer number of bits (Huffman 1952). Huffman coding can only be efficient under two conditions: Firstly, the frequency table of the data – or at least a good estimate – has to be known in advance. Secondly, $P(l_i)$ the probability of the occurrence of the individual letters l_i should be $P(l_i) = 2^{-j}$ where j is an integer and positive number (Strutz 2005). Arithmetic coding (Abramson 1963, Witten et al. 1987) poses a more efficient alternative to Huffman coding, if the occurrence probability of letters is not a power of 0.5. Predefined codes, such as Rice or Golomb-Rice codes (Rice and Plaunt 1971, Rice 1979, Rice 1983, Rice 1991, Golomb 1966), provide a faster alternative to arithmetic coding if the value distribution follows a geometric distribution. Exponential Golomb codes by Teuhola (1978), (punctured-) Elias codes (Elias 1975, Fenwick 1996), and the recently proposed Hybrid Golomb codes (Xue et al. 2003) are examples of predefined codes which are better adapted to other – non-geometric – distributions. But still all these methods are only efficient, regarding the achievable compression ratio, if the data follows the distribution the method is designed for.

Dictionary based compression methods, for example algorithms from the Lempel-Ziv group (Ziv and Lempel 1977, Lempel and Ziv 1978, Welch 1984), are other commonly used encoding methods with limited capability for data decorrelation.

Berg and Mikhael (1994) mention that for signal compression Lempel-Ziv-

Welch compression methods were applied with limited success only. They also state that Lempel-Ziv-Welch based techniques are unacceptable for high performance applications as the dictionary has to increase rapidly with the sample resolution to ensure frequent matches.

This is one cause, why Lempel-Ziv-Welch based methods should not be directly applied with the 16- and 32-bit-per-pixel astronomical data.

But those methods can be applied successfully, if the source data symbol set cardinality is limited for a certain data. As Lempel-Ziv-Welch exploits the correlation among different source data symbols, at a certain degree, and not only the letter distribution, it is still an option to be tested for data with limited source alphabet and correlated data. This also holds for the application of Burrows-Wheeler Transform based methods (Burrows and Wheeler 1994).

2.2.2 Data Reduction Methods

Typical data reduction methods are quantisation and sub-sampling (Strutz 2005). Quantisation decreases the number of distinct signal values (found in many lossy compression methods used in astronomy, discussed in Section 3.2.2). Sub-sampling decreases the temporal or spatial resolution of signals (Strutz 2007). Quantisation can be performed on each individual coefficient, this is known as “scalar quantisation”. Quantisation can also be performed on a group of coefficients together, this is known as “vector quantisation” (Saha 2000). Details on quantisation methods are not discussed here, as they are merely used for lossy methods and therefore the research work here cannot benefit from them. Details on quantisation methods can be found in (Gersho and Gray 1991) and (Ramakrishnan et al. 1998).

Often, with very advanced methods, the data reduction steps are highly adapted to the properties of the signal receiver to remove only unnoticeable properties of the data. Compression methods like JPEG and MP3 are highly adapted to properties of the human visual and auditory system. Therefore, a key aim when developing a new compression system is to identify data properties and select or develop new algorithms which fit well to these properties and the user requirements.

2.2.3 Decorrelation Methods

> *"Most of the lossy techniques could be adapted to become lossless since it is normally the introduction of an error stage in the process that provides the facility for compression. However, if these methods were converted to a lossless mode, compression would almost certainly not occur."* (Reid et al. 1997)

Many techniques for data decorrelation can be classified into two main categories, transform-based methods and predictive methods. Both key decorrelation method groups are discussed here with respect to their properties, especially their advantages and disadvantages regarding the usage in a lossless compression method, such as the one developed in this thesis.

In the past, unfortunately, a majority of the publications on data compression concentrated on lossy compression (Berg and Mikhael 1994). This may be due to two main reasons: Lossy methods may lead to higher compression ratios compared to lossless methods, and they are sufficient for a wide range of applications.

Transform-Based Methods

Most of the techniques which have been researched, standardised and implemented in image compression can be assigned to the family of transform-based compression methods (Ragab et al. 1998). Transform-based methods are often used for lossy compression.

Ragab et al. (1998) summarise the principle of transform-based data decorrelation as follows: "The choice of the transform matrix would produce a transformed image matrix that is sparse and with most of its large magnitude elements concentrated in a small region of the transformed image matrix". This principle is often referred to as energy compaction.

One transform-based compression method has gained popularity. JPEG compression (ISO/IEC 1994, Pennebaker and Mitchell 1992), already standardised in 1992 and used for natural digital images, is based on the Discrete Cosine Transform. JPEG is probably up to now the most common compression method for natural images, as it is applied automatically in many digital cameras.

In addition to the Discrete Cosine Transform a range of other transforms has been proposed, and compared regarding their properties. A range of them has also been tested for the special case of astronomical image compression (see Chapter 3).

Ragab et al. (1998) compare the performance of the Karhunen-Loève Transform, Discrete Cosine Transform, Discrete Hartley Transform, Discrete Gabor Transform, and Discrete Wavelet Transform. They find that, depending on the objective to achieve a low peak signal-to-noise ratio or high compression rates, no unique transform can be selected.

Saha (2000) highlights the advantages of the Discrete Wavelet Transform for mainly two reasons: firstly, the basis function (wavelet) has a limited extent; secondly, the wavelet coefficients often allow efficient quantisation and encoding strategies.

In addition to the lossy methods, lossless transform-based compression methods have been developed, like the integer wavelet-based approach of Calderbank et al. (1997) and the lossless JPEG mode.

But transform-based methods may not be optimal for computational size- and time-efficient lossless compression. “The transform alone does not lead to compression. On the contrary, the original signal is transferred into another representation, whereas in general, the data amount is even enlarged, as the transform coefficients are real-valued, while the original image is integer-valued” (Strutz 2005). This problem is also discussed by Berg and Mikhael (1994) who add that transform coefficients might even be complex-valued.

This leads to two basic problems in the context of lossless compression. Firstly, transforms based on floating-point values are inherently lossy (Barthel 2003). Secondly, the transform coefficients may, depending on their properties and their amount, not always provide the optimal basis for further compaction.

Additionally, transforms are computational time-intensive. Saha (2000) states that “this [transform-based] method provides greater data compression compared to predictive methods’, although at the expense of greater computation”.

The first problem, the inherent lossiness of transforms using floating-point values, can be reduced by the calculation and transmission of the difference between the transform compressed data and the original. Residual image creation has been applied for astronomical data compression (Press 1992, Starck et al. 1995), but much earlier for medical image compression purposes (Roos and Viergever 1988). The lossiness of the transform can as well be solved without the need of creating a residual image. According to (Strutz

2005) there are also transforms, which can be solely implemented in integer-arithmetic, but the resolution of the transform coefficients requires more bits than the original data. Therefore, the second problem even remains with integer-based transforms such as the one developed by Calderbank et al. (1997) or the one of Said and Pearlman (1996). A detailed discussion of the latter can be found in (Vemuri et al. 2002).

The second problem also appears with some of the transform compression methods proposed for astronomical image compression, such as the ones presented in (White 1992, White and Percival 1994, Louys, Starck and Murtagh 1999). Whether these methods achieve an efficient size compaction of the data depends on the properties of the transform coefficients. Lossy methods benefit from the fact that for most transforms, a vast majority of the coefficients is small or zero (Strutz 2005), and the important image features are captured in a few coefficients. Finer and finest details are encoded in the rest of the coefficients, thus the information degradation can be controlled by the coarseness, and the number, of coefficients stored. Lossless methods cannot reduce the discretisation and the number of transform coefficients, but still in some cases the transform coefficients can be encoded very efficiently.

White (1992) reported that for very smooth or constant images, transform coefficients may be zero over large regions. Unfortunately, astronomical images are typically noisy. That means the coefficients might be small but not zero. Moreover they capture important fine details and remain noise-like just as the original data. Therefore, transforms may not be the best choice for the decorrelation of astronomical image data.

Many contemporary lossless general-purpose image compression methods such as JPEG-LS and PNG do not rely on a transform-based decorrelation method, which further encourages considering other decorrelation methods when developing a lossless image compression algorithm. JPEG-LS and PNG rely on predictive coding, which is discussed in the next section.

Prediction

"Lossless compression of 2-D images has been performed very successfully using prediction-based methods" (Mielikainen 2006). Predictive coding methods aim to exploit local redundancy in the data by guessing the pixel intensity of a sample from a set of past data values. Probably the simplest possible scheme is thereby the differential pulse code modulation, where the last intensity value is used to predict the succeeding, and the difference between both samples is transmitted. Berg and Mikhael (1994) highlight that the prediction residual "generally has much lower entropy than the original data". A range of different methods for signal-prediction has been developed (Dawson-Howe 1996, Paeth 1991) and compared (Memon and Wu 1997, Weinberger et al. 2000).

Linear prediction has been researched for the compression of medical images in (Hu et al. 1997) and straightforward prediction schemes are widely used in lossless contemporary image format standards, such as the Portable Network Graphics standard (ISO/IEC 2004), and lossless JPEG. A table with the different lossless JPEG predictors can be found in (Memon and Wu 1997). Vemuri et al. (2002) discuss details of the algorithm and its different modes. With PNG, for example, four filter types use predictive coding methods to decorrelate the data. Two versions use the left, respectively the above pixel value, to predict the current pixel (ISO/IEC 2004).

The type 3 filter of PNG is an advanced higher-order linear predictor. With PNG, also the most advanced filter type, named after its inventor Paeth (1991), can be used. It uses a linear function of three neighbouring pixels on the left, above, and the upper left pixel for prediction. The Paeth filter chooses one of several possible predictor rules based on local gradients in the local surroundings of the currently processed value (Paeth 1991).

These methods are static methods which neither adapt to individual image properties nor do they selectively treat distinct image regions. Still, these more advanced prediction schemes can compete with lossless transform-based compression methods. Memon and Wu (1997) highlight that during attempts to develop a successor for lossless JPEG compression, "right from the first-round evaluations it was clear that the transform-coding-based proposals did not provide as good compression ratios" as predictive methods.

Savakis (2002) compares a range of contemporary lossless compression algorithms, including among others lossless JPEG, JPEG-LS, an S + P transform based method, and the algorithm used with the PNG standard and finds that the CALIC and the JPEG-LS outperform other methods.

Clunie (2000) compares the performance of traditional and state-of-the art lossless methods for the compression of medical images. He highlights that most modern methods involve prediction schemes with statistical modelling. He finds that new international standards like JPEG-LS or lossless JPEG 2000 perform better than existing JPEG and proprietary transform-based methods such as CREW (Zandi et al. 1995) and the S + P transform in combination with arithmetic coding (Said and Pearlman 1993).

Weinberger and co-workers (Weinberger et al. 2000) highlight that the CALIC algorithm obtains some compression gains by tuning the model more carefully to the image compression application. The observations of Weinberger et al. (2000), regarding the number of parameters of models, entropy savings and model costs, "suggest that the choice of model should be guided by the use, whenever possible, of available prior knowledge on the data to be modelled, thus avoiding unnecessary 'learning costs'". Weinberger et al. (2000) mention that "lower entropies can be achieved through higher order conditioning (larger contexts)".

With predictive coding, the predictor applied is a part of the image model. Dawson-Howe (1996) highlights that in general, "the model is based on the preceding neighbouring points in some static fashion", while for their method, the model weights are adaptively determined from the whole image using a least-squared-error solution. The adaptive determination and choice of optimal parameter sets for non-stationary signals has been discussed in (Strutz 2002, Memon and Wu 1997). These works link predictive coding with region-based compression methods discussed in Section 2.3.

Other Pre-Processing Methods

Fractal data compression (Fisher 1995, Mandelbrot 1982) uses a fractal transform to decorrelate the data prior to its encoding. In this thesis, fractal methods are treated as a separate group of compression methods. They are not assigned to transform-based methods, as the fractal transform requires search and matching operations during compression, which are not required with methods based on transforms such as the Discrete Cosine Transform or wavelets. An early review of the fractal image compression literature is given in (Saupe and Hamzaoui 1994). A more recent presentation can be found in (Zhao and Liu 2005).

With fractal compression, first the fractal transform is applied to the original data, which aims to represent the image using parts of itself by an iterated function system (Fisher 1995, Jacquin 1992). The description for generating the image is encoded. Fractal compression is often lossy as it includes a data reduction step. For a better size-efficiency, the number of iterations can be chosen to obtain a good approximation of the image

(Fisher 1995). Iterations which would be required to obtain a pixel-by-pixel identical copy of the image are omitted.

Compressing data using fractal methods is typically very time-intensive (Jacquin 1992, Saupe and Hamzaoui 1994, Hu et al. 2004).

2.3 Specific Compression Methods for Visual Data

Reid et al. (1997) state that till the 1980s image compression methods relied on encoding methods based on classical information theory (such as those discussed in Section 2.2.1) which exploit the redundancy in the images in order to achieve compression ratios of up to two on the data.

Even with lossy techniques, such as the ones based on the Discrete Cosine Transform (Wallace 1991) and a subsequent data reduction step, the higher compression ratios are typically only be achieved at the expense of image quality (Reid et al. 1997).

Therefore, in addition to these methods, a range of attempts was made to overcome the compression performance limitation of existing approaches by specialising the methods to distinct fields of application, or using adaptive methods which selectively treat distinct types of data.

2.3.1 Specialisation on Image Data Properties

One attempt to overcome the compression performance limitation is the specialisation of image compression methods to highly specific applications. That specialisation works can be seen with a simple comparison of the size of a Zip compressed BMP image, to its copy stored in PNG format. There, typically the PNG version is smaller than the Zip compressed counterpart. The predictive pre-processing step specific to image data properties is performed only with the PNG version of the image. It typically leads to a higher compression ratio.

The dependency between the tight coupling of data properties and the performance of the compression method – found as well for lossless prediction-based methods in Section 2.2.3 – can also be seen from a range of lossy special purpose methods, which have enabled the exchange of video, image, and audio information over world-wide computer networks during the last decade.

Carpentieri et al. (2000) emphasise that a compression method has to be closely coupled to the image data; in the best case, it incorporates prior knowledge about the data. This implies that in order to achieve a high compression ratio: a) image source properties have to be known, b) the compression method has to be highly adapted to those source properties. The close coupling of requirements, source properties, and compression method are probably the most outstanding research challenges for the compression of images, especially for lossless methods. In their survey of lossless compression methods, Berg and Mikhael (1994) highlight that "improvements are likely to require more complex and demanding source models". This supports the compression approach adopted in this research, aiming to adapt the compression method closely to the physical properties of the source.

2.3.2 Adaptive Methods

Another attempt to enhance the compression ratios includes adaptive methods. Adaptive methods can again be grouped into methods which adapt their prediction based on a local image area, and methods that truly divide the image into distinct areas and process them independently.

Adaptation based on the local area is applied with a range of methods which have already become widely used in some fields, for example the LOCO-I compression standard, which was standardised into the JPEG-LS standard (Strutz 2005). JPEG-LS bases its adaptation to distinct contexts on a gradient detector, which derives from a set of local gradients to which contexts the actually processed pixel belongs. JPEG-LS distinguishes between four contexts, which are encoded using different predictors (non-linear prediction), context-specific offsets, and choices of the Golomb-Rice encoder parameter (Strutz 2005).

Weinberger et al. (2000) highlight that JPEG-LS "is conceived as a low complexity projection of the universal context modelling paradigm, matching its modelling unit to a simple coding unit."

In contrast to methods that only adapt their prediction, JPEG-LS also includes a special encoding method, which is applied to flat image regions. Therefore, JPEG-LS even applies a sort of "segmentation" to the data. Although JPEG-LS is often discussed as an example for a very successful predictive lossless compression method, the performance gain of JPEG-LS compared to other predictive methods results partly from the adaptive choice of efficient encoding strategies. Figure 2.3 shows the data flow of the lossless LOCO-I/JPEG-LS compression method.

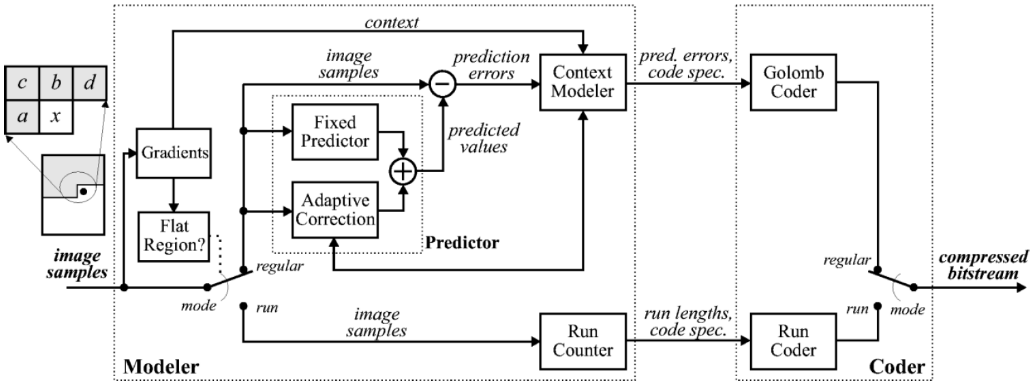


Figure 2.3: Data flow of the lossless LOCO-I/JPEG-LS compression method (image from Weinberger et al. (2000)).

2.3.3 Segmentation-Based Methods

Object-based methods or segmentation-based methods are an option to adapt a method even more closely to the data properties and the application class.

The research direction, which is based on higher-order source models and combines existing approaches with advanced methods, has been named second-generation methods by Reid et al. (1997), following Kunt et al. (1985), and Kunt (1988). Schmalz (2005) calls it “the convergence of image compression and object recognition” and highlights that “compression has progressed from entropy coding of a bit or pixel stream, to transform coding applied to rectangular encoding blocks, to feature-based compression that employs segmentation of isospectral or isotextural regions”. With JPEG-LS, the use of some kind of segmentation was already touched in the previous section.

The adaptive determination and choice of optimal parameter sets for non-stationary signals has been discussed by Strutz (2002). The author highlights that the context-based approach is a variant of composite source modelling, which treats the signal as an interleaved sequence generated by multiple sources (Strutz 2002). Each of the subsources has its own model (context) and associated parameters. The method developed in this thesis follows a similar approach, distinguishing between different regions in the image, which are encoded optimally with respect to their properties.

A range of different methods has been developed to segment the images. Reid et al. (1997) discuss region growing as a method to subdivide an image into a set of regions with a limited intensity variation. Split and merge algorithms (Reid et al. 1997, Pavlidis 1981) segment the image starting from a coarse predefined subdivision by merging adjacent regions with similar grey-level and splitting regions with large variations. K-means clustering methods minimise the sum of squared distances from all points in a cluster to its cluster centre (Reid et al. 1997). Segmentation using graphs can be achieved by using Shortest Spanning Trees (SSTs) or by applying a threshold to the fractal dimension of the image surface (Reid et al. 1997). Other recent methods make use of neural networks for the segmentation or combine segmentation with transform-based methods (Engel and Uhl 2002). A review of neural networks in image compression applications is presented in (Egmont-Petersen et al. 2002) and in Dony and Haykin (1995).

In addition to the segmentation method which is applied, segmentation-based methods can also be classified based on the content of the regions which are generated. According to the classification given in (Reid et al. 1997), visual pattern-based techniques divide the image into a set of visual patterns that contain visually important information. Reid et al. (1997) discuss the method of Biggar et al. (1988) as an example of another category of methods which are based on homogeneous regions. With these methods, the image is segmented into different regions of similar content.

The method of Ratakonda and Ahuja (1996) belongs to the category of methods based on homogeneous regions. The lossless method for natural or general-purpose grey scale images is based on a decomposition of the image into closed subregions “which are intrinsically similar and extrinsically dissimilar (with respect to adjacent regions)” (Ratakonda and Ahuja 1996). The method of Ratakonda and Ahuja (1996) can be summarized as follows: Visually important contour information is extracted with the method of Kundu et al. (1985), where detected contours are stored using contour coding. A linear prediction-based method is used to encode the residual data that remains after the edge structure removal. The authors highlight that their method gives a 15 to 20 % improvement compared to lossless JPEG.

Segmentation into different regions is also done with highly adapted compression methods for bi-level images, which incorporate features of object-recognition and image segmentation (Ono et al. 2000).

The method of Bai et al. (2004) classifies image regions based on their importance. The lossless medical image compression method is based on segmenting the image using unseeded region growing to extract diagnostically important regions of interest and encode them using the Burrows-Wheeler Transform. In the context of lossy compression, (Du et al. 2001) distinguish between content- and object-based compression. Their object-based lossy compression methods allocate more data volume to important regions than to less important regions. Also, segmentation-based hybrid compression methods have been proposed for the storage of medical images (Strom and Cosman 1997, Gokturk et al. 2001).

Regarding the application classes, segmentation, in combination with suitable encoders for decorrelating the data components generated by the decomposition, might be well suited for computer generated content such as rendered two- or three-dimensional vector graphics. Decomposition of natural images may be difficult and lead to a large number of small regions with complex textures (Jang and Rajala 1990, Jang and Rajala 1991, Reid et al. 1997). That this finding is still true can be seen from the conclusion of Schmalz (2005) that “neither the object-based compression nor image-based object recognition problem has been solved, although more progress is evident in the former case”. Strutz (2002) discusses the problem that it is difficult to find a limited number of contexts containing textures with similar properties. This thesis argues that in the special case of stellar field images this problem can be solved.

2.4 Summary

This chapter presents information theory concepts which underpin data compression, and it gives a systematic overview of compression methods. The chapter highlights that information theory states that the possible degree of compression for data depends on its entropy. Higher-order entropy, or more precise models of the data, can lead to estimates of information content which are lower than simpler models of the data.

The work of other researchers, such as in (Carpentieri et al. 2000) and (Barthel 2003), supports the argument adopted in this thesis that – in general – the better the compression method is tuned to the specific data content, the higher the possible compression ratio, especially in the case of lossless compression. This thesis adopts the position that the knowledge of data properties may be exploited to change the entropy of the data, by pre-processing the data into a form with lower entropy. This pre-processed data can be compressed into a size lower than the original data.

The data compression literature asserts that a key challenge for all compression methods is to remove the redundancy contained within the data. Berg and Mikhael (1994) noted that despite relying "on the same fundamental concepts, the implications [of lossless compression techniques] vary greatly", compared to lossy compression. While the performance of lossy methods also benefits from the removal of potentially non-essential information, lossless methods may only apply techniques which transform the input data into a representation which can be encoded with fewer symbols. In his taxonomy of data compression methods, Strutz (2007) identifies coding, data reduction and decorrelation techniques as the three main functional blocks of modern compression methods.

The overview presented in this chapter also argues that a combination of simple approaches (such as entropy coders), sensibly combined with preprocessing steps to yield a highly tuned compression method, can lead to effective compression. A model of the source can be used to describe the source properties and hold prior knowledge about the source. The review of methods for lossless compression from other fields also confirms this. It further reveals that, in order to achieve high lossless compression ratios, a central research challenge is the identification, description, and exploitation of data specific properties. This is also an outstanding research challenge for stellar-field image compression; hence results the need for research to identify the visual characteristics of stellar fields and the image properties introduced by the specific image capture devices (as investigated in Chapter 4). Therefore, a core objective of the work presented here is to adapt or develop an image data model tightly coupled to the data properties of stellar-field images.

While this chapter focused on the general architecture of compression methods and on image specific methods working on arbitrary types of image data, the focus of the next chapter is more specific. Chapter 3 gives a review of astronomy-specific image compression methods.

Chapter 3

Compression Methods for Astronomical Images

Compared to the number of general-purpose image compression methods (see also Chapter 2), astronomical image compression has been investigated much less by researchers in the field of image compression.

One cause for this could be that astronomical image compression is a specific area. This thesis argues that to maximise compression of astronomical images, standard image compression methods cannot be applied without adaptation, because of the special properties of astronomical images. How astronomical images differ from other images is discussed in detail in Chapter 4. Another cause that inhibits the direct application of a range of general-purpose image compression methods to astronomical images is the special requirement for lossless or at least near-lossless compression. While users of typical picture manipulation programmes can tolerate a loss of some information (which may be invisible), astronomers often require undegraded data. This requirement is due to the need for accurate photometric and astrometric measurements in astronomy. Lossless compression achieves less data reduction than lossy compression as only redundancy from the data may be removed in the case of lossless compression. Thus, it is especially important that the applied lossless method closely matches the properties of the data: No universal method works effectively and efficiently on all types of data.

This chapter presents a review of existing methods for compressing astronomical images; it highlights their strengths and limitations, and identifies the research gap addressed by this thesis. First, a categorisation of compression methods according to their data decorrelation method is given in Section 3.1. Then the existing methods are discussed more in detail. This presentation is split into three different categories: lossy methods (Section 3.2), methods supporting lossless as well as lossy compression (Section 3.3), and purely lossless methods (Section 3.4). Within these categories, publications are discussed according to the method applied. The latest attempt to include basic compression functionality into the FITS standard is discussed separately in Section 3.5, although it could have been included in the lossless methods section. It is highlighted in a separate section as its focus is not only on presenting a method for compressing astronomical images, but to show how compression functionality can be integrated into the FITS file standard. A performance evaluation of the astronomy specific compression methods discussed in this chapter is presented in Chapter 6.

3.1 General Categorisation

The systematic overview of compression methods, presented in Chapter 2, shows that a key feature that distinguishes compression methods is the decorrelation method which is applied. The discussion shows that, for image compression, mainly two basic decorrelation methods were used up to now: predictive methods and transform-based methods. In addition to these two, there are at least three further types of compression methods: firstly, those that make use of no data decorrelation step; secondly, those which use a limited data decorrelation process through general-purpose encoders such as Lempel-Ziv-Welch and the Burrows-Wheeler Transform, and finally fractal methods which are considered as a separate group of methods in this thesis (see also Section 2.2.3).

This section presents a categorisation of astronomy specific compression methods regarding the decorrelation method applied. The basic decorrelation categories distinguished are: no or limited decorrelation, predictive decorrelation methods, and transform-based methods. Fractal methods have not been proposed for the compression of astronomical images so far, to the knowledge of the author.

To account for the results of the discussion of methods closely coupled to properties of the visual data (see Section 2.3), which highlights that current methods often employ region-adaptive, object-based or segmentation-based methods, the decorrelation categories are further subdivided according to the level of adaptation to different image regions that the method uses.

Table 3.1: Categorisation of compression methods for astronomical images, according to their decorrelation step, their region-adaptivity, and their preservation or loss of data.

	No region adaptation	**Region adaptation**	**Segmentation**
Limited decorrelation (entropy coders)	🟦 Nieto-Santis. et al.(1999) Square root prescaling 🟦 White and Greenf. (1998) Linear quantisation 🟨 Veran and Wright (1994) Bit-plane based preprocessing 🟩 Weghorn et al. (1996) Signed Huffman coding 🟩 FITS/GZIP Deflate Algorithm 🟩 FITS/Rice Variable Length Coding		🟦 McNerney (2000) Thresholding 🟦 Huang and Bijaoui (1991) Morphological pre-processingB 🟦 Pardas (1997) Recursive segmentation, contour and texture coding 🟩 FITS/PLIO IRAF Pixel List compression algorithm
Prediction-based decorrelation		🟩 Sabbey et al. (1998) Linear prediction	
Transform-based decorrelation	🟨 Press (1992) Daubechies wavelet 🟨 White (1992) H - Transform 🟨 Starck et al. (1995) Pyramidal Median Transform 🟩 Louys, et. al (1999) Integer to Integer Transfrom		🟦 Boussalis et al. (2004) Object segm., wavelets, prediction 🟦 Dong et al. (2003) Segmentation plus wavelet

Legend:

🟦 Lossy method

🟨 Method providing both, a lossless and a lossy mode

🟩 Lossless method

The method for the classification was inspired by the taxonomy of image processing algorithms of (Egmont-Petersen et al. 2002), who distinguish between the methods by their level of abstraction of the input data, into six different categories: pixel level, local feature level, structure level, object level, object set level, and scene characterisation.

The categorisation given in Table 3.1 includes three categories for the level of adaptation to different image regions. The first category consists of methods that do not treat image regions differently. Region-adaptive methods include those that use parameters to vary the encoding and decorrelation step within the image. Furthermore, segmentation-based methods which segment images and choose different encoders and pre-processing steps for distinct areas were identified as a promising direction of image compression research in Section 2.3. These methods allow the exploitation of a very high level of redundancy with the image. Pyramidal decompositions do not really segment the image into distinct regions. Pyramidal decompositions such as those described in (Mueller 1999) and Salomon (2000) subsample the image and produce successively smaller images with fewer details.

The method of Sabbey et al. (1998), which is used to store image stamps from raw images, includes region adaptation. The method uses switched linear prediction based on a set of predefined rules. The residuals are encoded using a combination of Rice coding (for the high-order part of the residual) and binary coding (for the low-order part of the residual). It is region-adaptive insofar as the partitioning of the residual value and its encoding are varied. The change in the partitioning is performed to keep the encoding combination of binary and Rice encoding optimal. The core decorrelation method of Sabbey et al. (1998) is the prediction step.

One core statement emerging from the two-dimensional classification given in Table 3.1 is that the majority of existing methods do not employ region-adaptive processing of the data or even segmentation. As astronomical images, which are predominantly stellar-field images, consist of regions with very diverse properties, this may be an opportunity for further enhancements. Furthermore, most of the existing region-adaptive or segmentation-based methods are lossy (Huang and Bijaoui 1990, Pardas 1997, Dong et al. 2003). This applies also for the method of McNerney (2000) that applies segmentation insofar, as pixels with an intensity value below the image median are removed before the encoding. In addition to the method of Sabbey et al. (1998), another dedicated lossless method, PLIO (Pence 1994), exists that could be called region-oriented.

Most existing compression methods belong to the group of entropy coders with limited or no decorrelation. The lossy methods among them (Nieto-Santisteban et al. 1999, White and Greenfield 1998) apply different styles of quantisation, while the lossless methods mainly rely on entropy coding, like the Rice and Zip based method of White et al. (1999) and Weghorn et al. (1996).

Transform-based compression has been researched as well for astronomy applications, mainly with the aim to provide both lossless and lossy compression (Press 1992, White 1992, Starck et al. 1995), while only one attempt to develop a method based on prediction has been found in the literature (Sabbey et al. 1998). The method of Boussalis et al. (2004) employs prediction as well, but only for some image parts after segmentation, while the other parts are compressed using transform-based methods. Hence, the method of Boussalis et al. (2004) is classified between the two decorrelation methods.

3.2 Methods Providing Only Lossy Compression

Many of the multimedia applications, which evolved throughout the 1990s, are based on lossy compression. Indeed, lossy compression has made many of the revolutionary applications possible which have made computers the central entertainment platform in our homes. Lossy compression techniques have had a great impact on computer applications, for example by enabling the storage of previously large music files in small MP3 files, or large bitmap pictures in small JPEG files. This probably has encouraged many researchers to consider adapting lossy compression techniques for astronomical applications. Many of the techniques that have been reported for compressing astronomical images are lossy; few techniques allow lossless compression. Purely lossy approaches are presented in this section. Methods which are capable to work in either a lossy or a lossless mode are discussed separately in Section 3.3. Finally, methods for strict lossless compression are discussed in Section 3.4.

3.2.1 Object-Based Compression Methods

This section starts with a set of four object-based methods which use different techniques. While the first two, which have been developed in 1991 and 1997, are based on morphological pre-processing, the third method proposed by Dong et al. (2003) uses thresholding-based segmentation and wavelet-based compression to store the different image objects. The fourth, more recent method of Boussalis et al. (2004) is discussed at the end of this section.

The first lossy compression method for astronomical images, based on the morphological skeleton transform, was proposed by Huang and Bijaoui (1990). The main idea is to detect structures above the background and encode them in compact form after pre-processing the data using morphological operations (Heijmans 1994, d'Ornellas 2001). All structures at a certain level above the background are compressed, while the background is discarded using a technique called "3-sigma clipping" by Huang and Bijaoui. Huang and Bijaoui (1990) examine two different compression methods, one based on the binary morphological skeleton transform, the other based on an extension of the morphological skeleton transform to grey-scale images. This extension is called the grey-tone morphological skeleton transform. An attempt to a precise mathematical description of morphological image operators is provided in (Heijmans 1994). Algorithmic patterns for morphological image processing are discussed and presented in (d'Ornellas 2001).

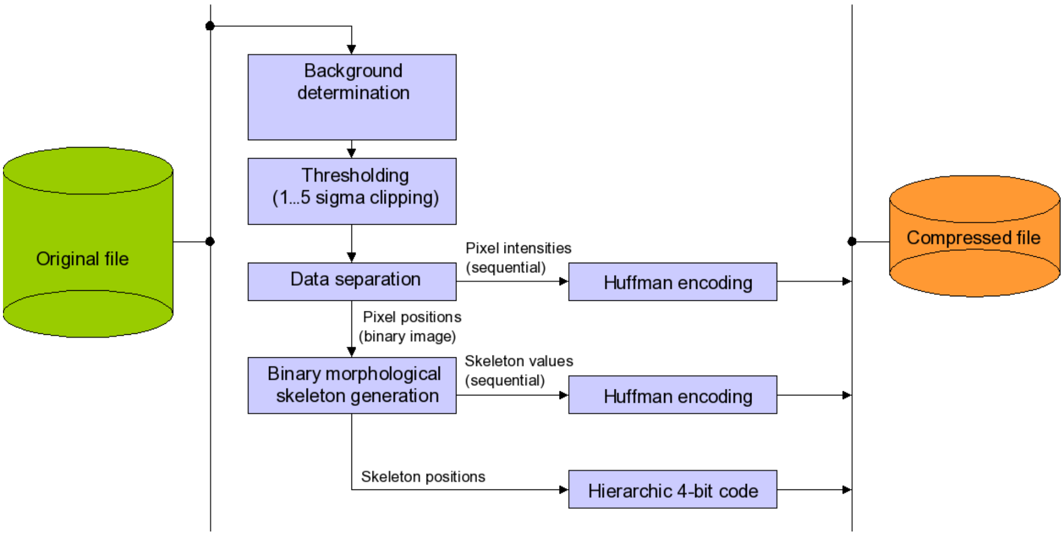

Original file
Background determination
Thresholding (1...5 sigma clipping)
Data separation
Pixel intensities (sequential)
Huffman encoding
Pixel positions (binary image)
Binary morphological skeleton generation
Skeleton values (sequential)
Huffman encoding
Skeleton positions
Hierarchic 4-bit code
Compressed file

Figure 3.1: Data flow of the binary morphological skeleton transform compression algorithm (Huang and Bijaoui 1990).

Huang and Bijaoui (1990) use the binary morphological skeleton to describe a grey-scale astronomical image as follows. First, the image is separated into two parts:

1. The intensity values of the detected foreground pixels.
2. A binary image that contains the positions of those pixels.

The intensity values are stored sequentially in an array and encoded using Huffman code. The binary image that contains the pixel positions is reduced by the binary morphological transform. Again, skeleton positions and values are separated. The positions are coded using a hierarchical 4-bit code, while the values are encoded sequentially by Huffman code. Figure 3.1 shows the data flow of the compression method in detail. A detailed description of the second algorithm using grey-scale morphological skeleton transform is presented in the original paper.

Both methods presented in the paper of Huang and Bijaoui (1990) are tested using three sample images. The authors compare their methods to the following seven simple compression methods:

1. Huffman coding: The whole image is compressed using plain Huffman encoding.

2. Successive difference encoding: First, the difference between two successive image data columns is taken, then the resulting data is encoded using Huffman coding.

3. Bit plane method: The pixel values are separated into bit-planes, which are encoded separately using the hierarchical 4-bit coding technique.

4. Run length coding: The original detected data values are stored sequentially in a one-dimensional array and Huffman coding is applied to the data. The binary image with the positions of the detected values is compressed using run-length coding.

5. Contour coding: The contour of each field is coded using the chain code of Freeman (1974).

6. Improved contour coding: Successive differences of modulo 8 are used for the contour chain code. The starting points of the contours are stored using the hierarchical 4-bit coding technique.

7. Morphological skeleton transform: The whole image is compressed using the morphological skeleton transform with the optimal structuring element for the image.

Huang and Bijaoui (1990) compare the results which the above-listed seven compression methods achieve on the thresholded images to the best results achieved by the best version of the morphological skeleton transform on the particular image. The results of Huang and Bijaoui (1990) can be summarised as follows:

For the "Coma stars" image, a stellar-field image, the compression ratio of the morphological skeleton transform is approximately three times higher than Huffman encoding. For the "Coma galaxies" image, an image containing a smaller cluster of galaxies, the compression ratio of the morphological method obtains approximately 6.5 times the compression ratio of Huffman encoding. Finally, for the third image, a large-scale elliptical galaxy, the compression ratio may be enhanced by a factor of three if the morphological skeleton transform is used instead of Huffman encoding. The comparison of the grey-scale morphological skeleton transform to the binary morphological skeleton transform leads to equal compression ratios for the "Coma star" image, slightly higher compression ratios of the binary morphological transform on the galaxy image, and by a factor of up to two improved compression ratios for the grey-scale morphological skeleton transform in case of the large-scale galaxy image (Huang and Bijaoui 1990).

One of the main advantages of the method based on morphological decomposition, compared to other lossy compression methods, is that it does not introduce a global distortion that affects each pixel of the image as may be the case with transform-based approaches (Huang and Bijaoui 1990). All structures at a certain level above the background are compressed without distortion. Nevertheless, the method is lossy because the image background is discarded completely ("3-sigma clipping"). The morphology pre-processing based compression method is compared to a method using the H-transform for pre-processing in (Bijaoui et al. 1996). The H-transform based approach is almost similar to the one of White (1992), but in order to avoid blocking artefacts which appear in the restored image (Bijaoui et al. 1996, White 1992), it uses an image restoration method based on *a priori* knowledge.

With their comparison, Bijaoui et al. (1996) come to the conclusion that both the morphological method and the H-transform based method lead to high compression ratios on classical astronomical images although the comparison relies solely on the lossy compression rates achieved on the different images. The evaluation does not include a measure for the distortion introduced to the compressed data. Hence Bijaoui et al. (1996) conclude that "no perfect solution exists for image compression, we must take into account the image texture and future use".

The second object-based compression method is proposed in (Pardas 1997). Pardas (1997) highlights that his method belongs to the family of second generation image compression methods (Reid et al. 1997). The image is described as a set of regions (partitions) and information about the region content. Therefore, the compression algorithm consists of a two-stage process. Pardas (1997) proposes a general multiresolution top-down segmentation algorithm based on morphological tools. Figure 3.2 shows the data flow of the iterative process.

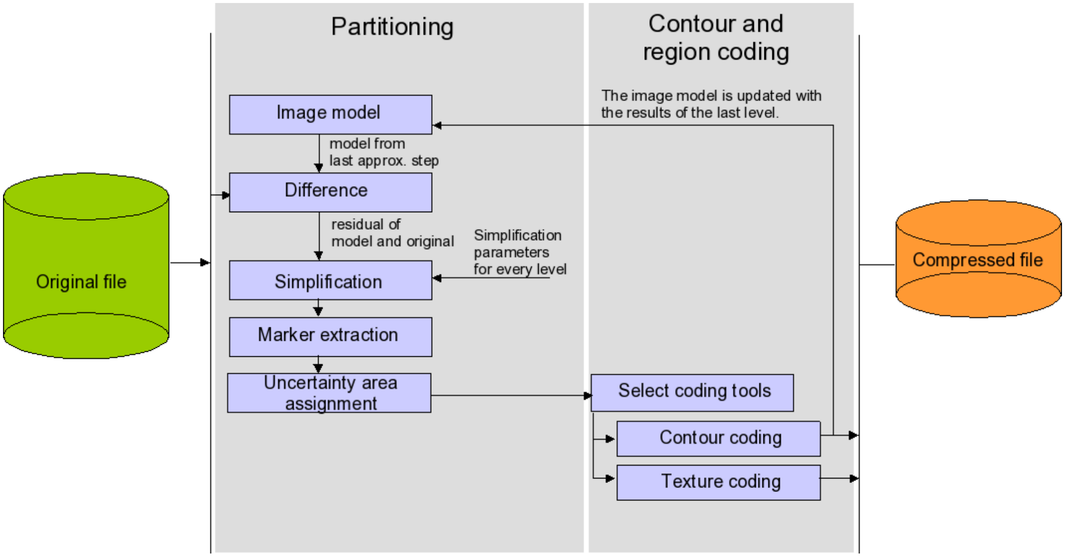

Partitioning
Contour and region coding
The image model is updated with the results of the last level.
Image model
model from last approx. step
Difference
residual of model and original
Simplification parameters for every level
Original file
Simplification
Marker extraction
Uncertainty area assignment
Select coding tools
Contour coding
Texture coding
Compressed file

Figure 3.2: Data flow of the algorithm proposed by Pardas (1997).

The method of Pardas (1997) works as follows: During each iteration step, a model of the image – based on the information gathered from previous compression steps – is generated to obtain information of the regions that are not correctly represented in the current stage. The difference of the original image and the model is calculated. The residual image is simplified by applying morphological filters. Markers that identify the segmented regions are extracted. The “uncertainty area” – area which is not yet assigned to markers – is assigned to some region with nearly similar characteristics. Finally, the region borders and textures are encoded. Pardas (1997) mentions chain coding techniques and skeleton decompositions for contour coding.

Different techniques may be applied to code the region texture. A range of techniques from simply storing the mean value of every region to using shape adaptive Discrete Cosine Transform or region-oriented wavelet transform are mentioned by Pardas (1997).

Pardas (1997) includes an example for the discussed compression method using three successive segmentation levels. If the contours are coded with the chain code method, a compression ratio of 60 is achieved when the region textures are coded using a generalized orthogonal transform.

The image compression technique presented by Pardas (1997) remains more of a proposal for a technique to consider when developing an astronomical image compression method rather than a technique that can readily be used. Before the method can be used in astronomy, an optimal texture coding technique has to be determined.

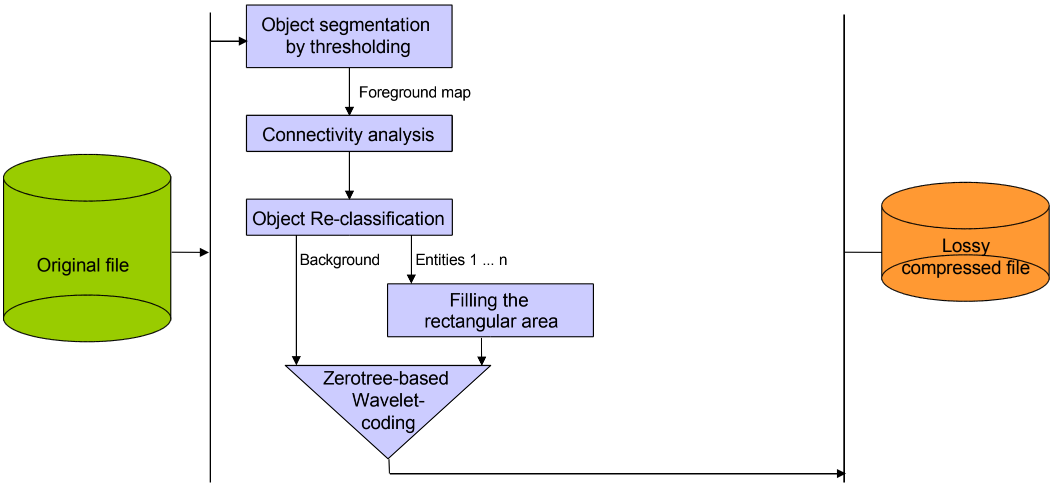


Figure 3.3: Data flow of the object-based compression algorithm (Dong et al. 2003).

A third content-based lossy method for the compression and transmission of astronomical images is presented by Dong et al. (2003) as an extension of their previous research on effective error control for video streaming. The content-based method requires the segmentation of the astronomical image into individual entities and the independent encoding of the entities using wavelets. Similar to the work presented in this thesis, Dong et al. (2003) focus on the compression of typical astronomical deep-field images – images with point like bright sources on a dark background – and adopt the classification of astronomical images also presented in (McNerney 2000). A comparison of the compression results of a large-scale object with a deep-field image shows that the SPHIT compression algorithm (Said and Pearlman 1996) has difficulties with typical astronomical deep-field images. Thus the authors conclude that a big challenge in astronomical image compression is how to compress deep-field images (Dong et al. 2003). An overview of the method proposed by (Dong et al. 2003) is given in Figure 3.3. The steps involved in the algorithm are:

1. Object segmentation: The segmentation of objects is done by thresholding. According to Dong et al. (2003) this is relatively easy due to the huge intensity contrast between foreground and background.

2. Connected components analysis: The image is scanned and the connected pixel regions are labelled.

3. Object re-classification: Objects are classified based on their area, and small ones are merged into the background. The result of this step is a list of astronomical entities and a background with insignificant entities (according to Dong et al. (2003)).

4. Wavelet coding: All entities are encoded individually using wavelet coding. The available bits are divided into the number of bits required for the background and those available for the foreground. The number of bits available for the background depends on the quality requirements. The number of foreground bits assigned to a single foreground entity depends on its area. Wavelet coding is applied to the smallest regular area that covers the whole entity, to bypass the problem of having to represent the shape of each entity accurately.

Dong et al. (2003) also present a robust packaging technique and a transmission scheme to prevent error propagation in case of packet loss on a network and to optimize the visual quality of the data in such a case.

The evaluation of the possible compression rate does not include any discussion of the influence that the compression method has on astronomy applications. Therefore it remains unclear for which application classes the method might be used. The authors evaluate their robust packaging technique, and report a good streaming quality on lossy networks in their experimental results. This is an important feature for transmission based on the user datagram protocol (UDP). But for the transmission and retrieval of astronomical images from image databases often the hypertext transfer protocol (HTTP) and file transfer protocol (FTP) are used. They are based on the transmission control protocol over internet protocol (TCP/IP), which guarantees a lossless transmission by re-transmitting lost packages.

Boussalis et al. (2004) recently proposed an object based lossy compression method, combining object segmentation with wavelet and predictive coding. The object analysis classifies objects into clear and faint, and applies predictive coding for the latter. Zero-tree wavelet-based compression is used for the clear objects. The objective of the method is to enhance the detection of faint objects in astronomy while providing a good overall visual effect (Boussalis et al. 2004).

3.2.2 Quantisation-Based Methods

Methods presented in this section are based on quantisation or thresholding. Both methods lead to the reduction of the number of bits required to encode the intensity values of every pixel. Quantisation can be applied as a first step for the lossy compression of astronomical images as it leads to a lossy approximation of the original signal. In addition to scalar and vector quantisation (see also Section 2.2.2) quantisation methods can also be characterised by their level spacing. For astronomical images, uniform and non-uniform scalar quantisation methods have been discussed. Square root prescaling has been proposed by Nieto-Santisteban et al. (1999) to reduce the size of astronomical images generated by the "Next Generation Space Telescope". Linear quantisation was first proposed by White and Greenfield (1998) for compressing floating point images as well as integer images. Watson (2002) also presents a method for quantising astronomical integer images in order to remove oversampling of the noise.

Another lossy method, which can be used for the distribution of lossy compressed images, is proposed in (McNerney 2000). The key aim for developing the method is to transmit images over a low-bit-rate link, which required 1024 by 1024 16-bit-per-pixel images to have a size of less than 30 kilobytes. Also McNerney (2000) mentions that the method targets the needs of astronomers, schools and colleges, and 'Public Understanding of the Science and Technology' (PUST) displays. Tools and services for accessing astronomical data for educational purposes are discussed in detail in Christian (2006).

As discussed earlier and in other papers, for example (White 1992), astronomers accept lossy compression only for previewing and selection purposes. Like all lossy methods, the method of McNerney (2000) might be used for schools, colleges, and planetariums. Lossy methods might not be used for scientific grade investigations, especially as the paper discusses reduction of the signal-to-noise ratio, during image generation, to meet time constraints for the data transfer to public displays.

The key idea of this method is to threshold the image data to remove all the background data below the image median. According to McNerney (2000), then run-length encoding, and a lossless encoding method are applied. The latter was developed by the same author earlier. The key idea of this method is to remove the dark and noisy background, whose presence in the images makes compression of astronomical images difficult. While this approach may lead to high compression ratios, the thresholding may also cripple faint sources. This is also highlighted by White (1992), who discusses thresholding in combination with run-length encoding as well. Also, thresholding might lead to problems if the image background is not totally plain.

The method is compared to other lossy compression methods in (McNerney 2000). The results show that faint stars are disregarded (see Figure 3.4). It is also unclear what causes the noise in the background area, given that the dark area is segmented by thresholding (see Figure 3.4).

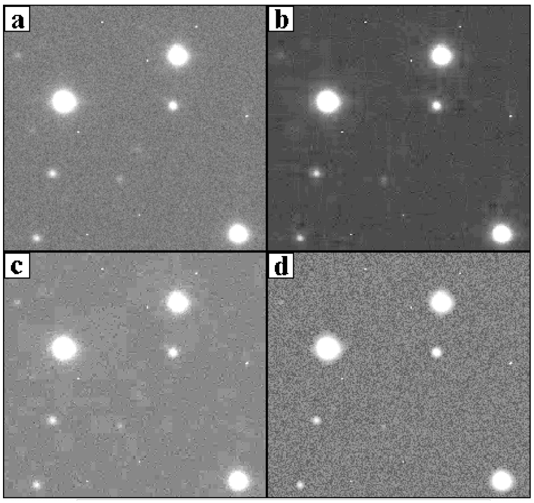


Figure 3.4: Comparison of the original image (a) and an image compressed using the method of McNerney (d). It can be seen that faint objects present in (a) are lost in (d), while Hcompress and Fitspress, images (c) and (b) keep these objects (figure from (McNerney 2000)).

The core idea of the method of McNerney (2000) is that fainter objects and the background of astronomical images are not important and can be discarded before compression. As the method of McNerney (2000) is intended for educational purposes, not for scientific grade investigations, these requirements do not match with the requirements for this investigation. For scientific applications astronomers often have to know how the sky background exactly looked like before an event, thus it cannot be considered unimportant and discarded.

Another quantisation method, square root prescaling, attempts to retain as many bits of noise of the original data as possible (Nieto-Santisteban et al. 1999). The scaled square root of the original data is calculated and truncated to reduce the dynamic range of the data. The scaling, which is derived from the readout variance R and the gain G of the imaging device, preserves a desired number of noisy bits. The formula used for prescaling and rounding is:

$$D_{quant} = \lfloor N_B \cdot \sqrt{G} \cdot \sqrt{D + \frac{R}{4G}} + 0.5 \rfloor \tag{3.1}$$

N_B is multiplied with the number of noise bits that have to be saved. Nieto-Santisteban et al. (1999) suggest to use $N_B \approx 2$; it leads to a quantisation step close to the digitisation noise. Nieto-Santisteban et al. (1999) report that following the square root prescaling, the Rice, Compress, and Gzip algorithms[1] achieve similar results, with overall compression ratios of 4 (retaining 4 noise bits), up to as much as 8 (retaining 1 noise bit).

Square root prescaling with truncation leads to a non-uniform quantisation where smaller signal values are quantised more precisely than larger signals. The prescaling that is applied with the square root quantisation can be used to preserve a portion of the noise.

The problem of compressing floating point images is addressed by White and Greenfield (1998). The authors propose a scheme for compressing this type of images by quantising the data and encoding the resulting data stream of integer values. The compression method is lossy as the original data can not be recovered exactly. The method proposed by White and Greenfield (1998) consists of three steps:

[1] Gzip is a general-purpose compression method which is widely used for compressing all kinds of binary data.

1. **Noise estimation:** First, the noise value σ is defined as 1.483 times the median absolute value of the differences of successive pixels. If the median absolute difference is zero, σ is defined as the root mean square that is computed after sigma-clipping to reject outliers.

2. **Quantization:** Secondly, the floating point values are converted to integer values by quantizing them. The quantization step size is chosen with respect to σ. The step size is $2^{-B}\sigma$. B is a user-chosen parameter that is, according to the given definition, directly related to the compressed file size. Increasing B by one increases the number of bits used per pixel by one bit.

3. **Rice coding:** Finally, the integer data generated by quantization is encoded using Rice coding.

Due to its straightforward design and the Rice coding method, the compression method is fast and as the image lines are compressed separately, not the whole image has to be decompressed when accessing only parts of it (White and Greenfield 1998).

According to the evaluation reported by the authors, compression ratios of 3 are possible using conservative settings for the permitted level of changes. The authors claim that their method has only a scientifically negligible effect on the data although the method is lossy. The compression method of White and Greenfield (1998) could also be applied to integer images, although the authors mention that quantisation may be omitted for integer images in order to losslessly encode the data. Rice encoding is also available as a standard compression method for FITS images.

A method which comprises linear quantisation is also proposed by Watson (2002). This method involves the same three steps, noise estimation, quantisation, and encoding, as the method of White and Greenfield (1998) discussed above. The lossy quantisation of the image data in the FITS file is followed by lossless compression using general-purpose compression tools such as Gzip. The key advantage of this method, compared to the one of White and Greenfield (1998), is that the quantised FITS file can easily be recovered using the same general-purpose compression/decompression tool that was used when compressing the file. The quantum used by Watson (2002) for resampling is chosen as a fraction of the estimated background noise in order to reduce the over-sampling of the noise. Also, the resampling quantum is chosen to be an integer multiple of the original encoding. Certain special values like “blank” pixels are not changed as those values are often used to flag saturated data or implicitly blank pixels.

After quantisation, the FITS image is not smaller than the original file, but the amount of different data values has been minimized. Therefore, as (Watson 2002) reports, general-purpose compression tools perform better on the quantised data.

In general, with all quantisation-based methods, it remains questionable when taking into account how data of typical raw images is distributed, if compression methods based on a look-up table could not lead to nearly equal results without introducing a distortion to the data (Weghorn 2002). A look-up table based approach is a lossless alternative to quantisation.

If lossy compression is used, non-uniform quantisation preserves small signals from darker objects better than a uniform quantisation, as the quantum sizes for smaller values are smaller than the ones for large signals. But even with non-uniform quantisation it remains uncertain in which astronomical application classes the introduced distortion is tolerable.

3.3 Hybrid Methods Providing Both Lossy and Lossless Compression

Many of the compression methods proposed for astronomy applications provide a lossless mode and a lossy one. These methods – called hybrid here – typically rely on transform-based compression methods widely used for audio and image compression. Many of the methods presented here seem to work perfectly both for fast previewing and selection of material as well as for archival storage. However, they typically suffer from the high computational complexity associated with the transform-based approach which the lossy system kernel includes.

3.3.1 Wavelet-Based Methods

One of the earliest methods for the compression of astronomical images is Fitspress (Press 1992). Fitspress is based on a two-dimensional wavelet transform, using the Daubechies wavelet with four coefficients. A description of the Daubechies wavelet transform can be found in (Salomon 2002). Compression methods based on the wavelet transform can provide both lossless and lossy compression.

Fitspress includes an interesting residual image generation method to provide both lossless and lossy compression. In one run, Fitspress outputs two result files – a lossy compressed image file and a residual file that contains the "noise" information. If both files are stored, the original image can be reconstructed exactly (lossless compression).

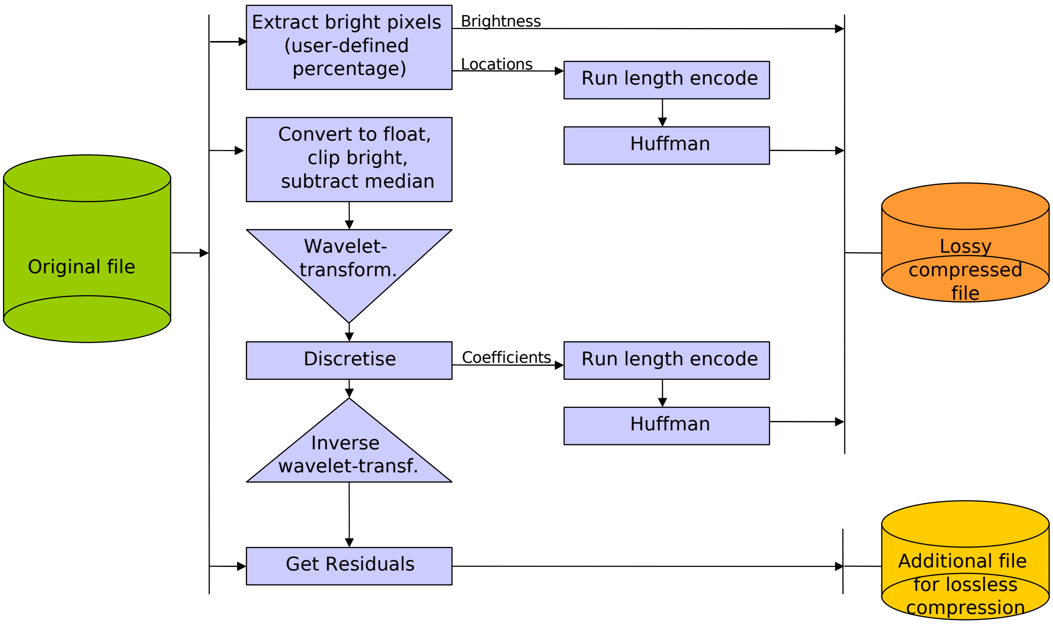


Figure 3.5: Data flow of the Fitspress compression algorithm (Press 1992).

The Fitspress algorithm attempts to store only the important information within the image through the following procedure (Press 1992): First, a user-chosen fraction of the brightest – hence regarded as the most important – pixels are extracted and stored exactly. Second, the whole image is compressed using the Daubechies wavelet transform. The wavelet coefficients are discretised again and a fraction of the wavelet coefficients is stored. Both, the extracted bright values, selection coordinates, and the wavelet coefficients from step two are compressed independently using run-length encoding followed by Huffman coding. Both encoded components are stored within the file containing the lossy compressed image. Finally, the pixel values for the residual file are determined by reversing the wavelet transform and subtracting the result from the original image values. Figure 3.5 shows the data flow.

Typical lossless compression ratios achieved range in the order of 1.5 to 5 (Press 1992). According to the measurements reported by Press (1992), Fitspress produces good results, in terms of utility for astronomical images, at compression ratios between 10 and 30. A range of example images is shown in (Press 1992). The tested release (Version 0.8) of the Fitspress software processes 2-dimensional, 16-bit-per-pixel, single-plane images. In a comparison of the Daubechies transform with the Haar transform, discussed for astronomical image compression in the following subsection, Salomon (2002) argues that the Daubechies wavelet produces a better energy compaction than the Haar transform.

Interestingly, the idea of Fitspress, which was first published for astronomical images in 1992, was re-published later as a new general-purpose image compression algorithm in (Mandyam et al. 1995). Like Fitspress, the paper presents the idea of using the Discrete Cosine Transform, discretise the coefficients, inverse the transform and calculate a residual image. Finally, the discretised coefficients and the residual should be stored.

Hcompress is an image compression package presented by R.L. White (White 1992, White and Percival 1994) which is based on the H-transform, a two-dimensional generalisation of the Haar wavelet transform. Apart from the encoding step that includes two stages, Hcompress follows the typical design of a transform-based image compression method. Like Fitspress, it provides both a lossy and a lossless mode. The H-transform based method relies completely on integer arithmetic; hence there are no losses due to rounding, which occur when floating-point arithmetic is used. Thus, during compression, the lossless mode does not require an inverse wavelet transform and a difference image to account for numeric errors.

The lossless Hcompress relies on the assumption that the coefficients of the H-transform are easier to compress than the original pixel data. As the H-transform is based on the differences between adjacent pixels, this is true for nearly noiseless data (White 1992). The author mentions that for very smooth or constant images, transform coefficients may be zero over large regions. In the lossy mode, transform coefficients are quantised as indicated in Figure 3.6, which shows the data flow. The quantisation step, as it is only applied in the lossy mode, is surrounded by a dotted line. Quantised or non-quantised wavelet coefficients finally are coded with a method first proposed by Huang and Bijaoui (1990). First, the bit-planes of the coefficients are separated and individually coded by quadtree encoding, thereafter Huffman encoding is applied to the quadtree coefficients. In case the quadtree coding causes an enlargement of the bit-plane data, the bit-plane is transcribed verbatim to the output file.

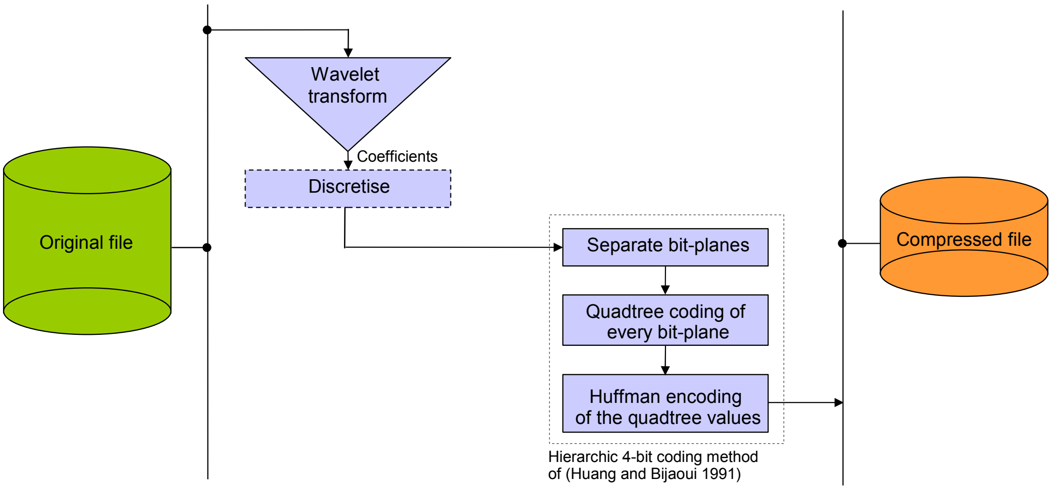

Wavelet transform
Coefficients
Discretise
Original file
Separate bit-planes
Quadtree coding of every bit-plane
Huffman encoding of the quadtree values
Hierarchic 4-bit coding method of (Huang and Bijaoui 1991)
Compressed file

Figure 3.6: Data flow of the Hcompress compression algorithm (White 1992).

According to (White 1992), Hcompress achieves lossy compression ratios of at least 10 without a noticeable loss in the astrometric and photometric properties of the image. White (1992) also includes a discussion about the properties of the H-transform for adaptively filtering the image by discarding information at certain scales. White (1992) mentions lossless compression ratios for CCD generated images of factors from 3 to 30 depending on the CCD characteristics. The only characteristic given by White (1992) is the read-out noise of the CCD.

In their second paper, White and Percival (1994) stress a possible application of Hcompress as the basis for a progressive image transmission system for remote observation and visual inspection of data from remote image archives. In an evaluation of lossy compression for astrometry applications, White and Percival (1994) come to the conclusion that astrometry is hardly affected by compression for modest compression factors of up to 20, for digitised photographic plates.

3.3.2 Other Transforms

Astronomical image compression based on the pyramidal median transform is proposed by Starck et al. (1995). The pyramidal median transform (PMT) is introduced as an alternative to a wavelet transform, for example used in the Hcompress approach. The authors argue that the PMT has several advantages over the wavelet transform:

- **Robust smoothing:** The nonlinear PMT consists of a series of smoothings of the input image, each successive smoothing with a broader kernel leads to a new resolution scale. Starck et al. (1995) argue that unlike with the wavelet transform, stellar objects are not present at each resolution scale.

- **No zero mean property:** The mean of the coefficients of the different resolution scales of the PMT is not necessarily zero, as is the case with the wavelet transform. Negative values, which can create artefacts, are not present with the PMT.

- **Integer arithmetic:** The PMT can be carried out in integer arithmetic that leads to significant computing time savings compared to floating-point arithmetic. Even today, with fast floating-point units included in standard desktop CPUs integer arithmetic is still faster on a range of embedded CPUs. These are commonly used on space-based telescope systems (Starck et al. 1995).

The proposed PMT compression method is capable of both lossy and lossless compression. As with Fitspress, the lossless mode relies on the computation of the residual between the compressed image and the original image. In the lossless mode, the compression ratio is about 3 to 4, according to the measurements of Starck et al. (1995).

Figure 3.7 shows the data flow of the PMT compression method. First, the PMT is applied to the input data. Then, the obtained coefficients are quantised to obtain a filtered image. The encoding of each resolution level relies on the Huang-Bijaoui method (Huang and Bijaoui 1990), just like in Hcompress. In the lossless mode, the residual noise between the image and the inverse transform of the quantised coefficients is calculated. The authors do not mention the method used to compress the residual noise. The intended area of application of the PMT based compression method is to provide preview functionality in large image databases, such as sky surveys.

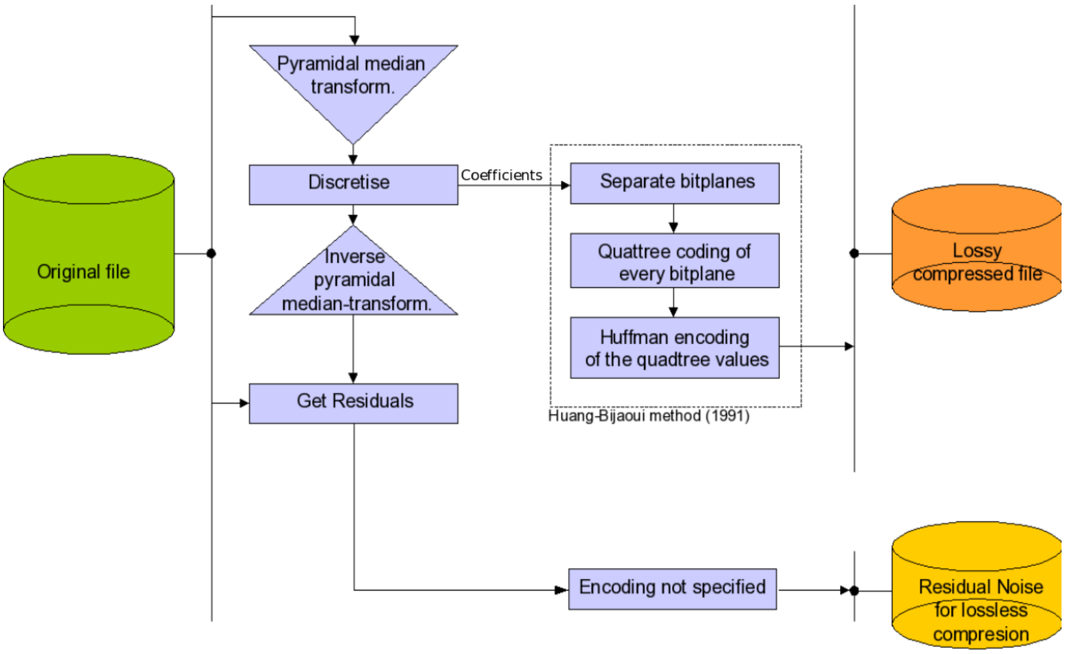

Pyramidal median transform.
Discretise
Coefficients
Inverse pyramidal median-transform.
Get Residuals
Original file
Separate bitplanes
Quattree coding of every bitplane
Huffman encoding of the quadtree values
Huang-Bijaoui method (1991)
Lossy compressed file
Encoding not specified
Residual Noise for lossless compresion

Figure 3.7: Data flow of the PMT compression algorithm (Starck et al. 1995).

3.3.3 Bit-Plane Separation With Greycode Encoding

Compfits (Véran and Wright 1994) is a utility program to analyze and preprocess astronomical images. It does not include its own compression routines but it is intended to work as a filter program to separate the compressible and less compressible bitplanes of a 16-bit-per-pixel integer image (Véran and Wright 1994). It sends the compressible part to a user-chosen external lossless compression program (e.g. Gzip, compress,...).

The main idea of Compfits is to separate compressible and incompressible bitplanes of astronomical images. While the low-order partition – the low-order bitplanes of astronomical images – is due to noise inherently incompressible and likely to grow in its overall size during "compression", the high-order partition can be compressed (Véran and Wright 1994).

Compfits works as follows. First, the input data is converted to grey code to increase the coherence of each bit-plane (see Figure 3.8). Second, the first-order entropy of every bit-plane is calculated. Compfits uses the entropy to separate the high from the low-order bit-planes. The bit-plane, for which the first-order entropy becomes slightly different from 1 bit-per-pixel defines the limit between low-order and high-order partition. This limit is taken to mean that the bit-plane does not consist of pure noise anymore. Finally, the high-order partition is padded to 16-bits-per-pixel and sent to an external compression program. The incompressible low-order partition is stored uncompressed.

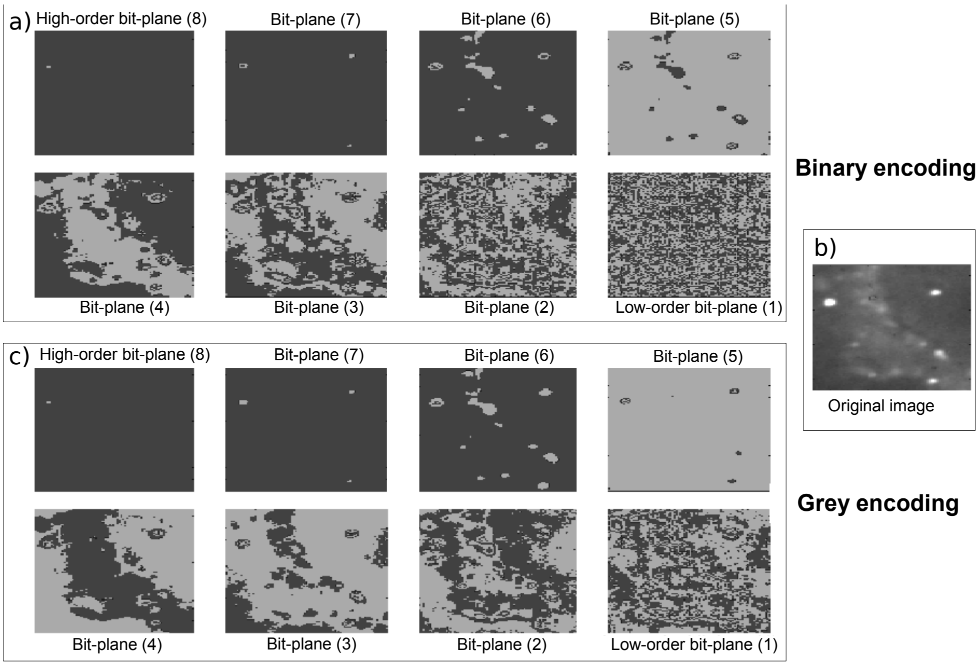


Figure 3.8: Bit-planes of a small astronomical image, encoded only with a bit depth of 8 bits. The upper eight images (a) show the original image (b) encoded in normal binary encoding. The lower eight images (c) show the image data encoded in grey code.

Figure 3.9 shows the bit-planes of an astronomical image which has been encoded with a bit depth of eight bits, in binary and grey code, to demonstrate that grey encoding leads to larger coherence of each bit-plane. While Compfits might also be used in a lossy mode, the focus lies on lossless compression.

Comparing the lossless modes, Compfits achieves nearly the same as or slightly better compression ratios than Fitspress, according to Véran and Wright (1994). Véran and Wright (1994) report that on typical images, Compfits combined with Compact 1.0 is three times faster than Fitspress.

An enhanced version of the non-lossy compression method, Compfits is presented by Gaudet et al. (2000) to address the challenge of archiving data generated by CCD mosaics. While the basic algorithm remains unchanged, the implementation is enhanced. The only change of the algorithm is a new sampling method to determine the LSB (least significant bit) - MSB (most significant bit) threshold. Compfits 2 makes use of the FITSIO library (Pence 1992) to read and write FITS files.

The comparison given in (Gaudet et al. 2000) of a variety of lossless compression programs shows that Compfits 2 (used in combination with Compact or Compress) is about ten times faster than Gzip or Bzip2 alone.

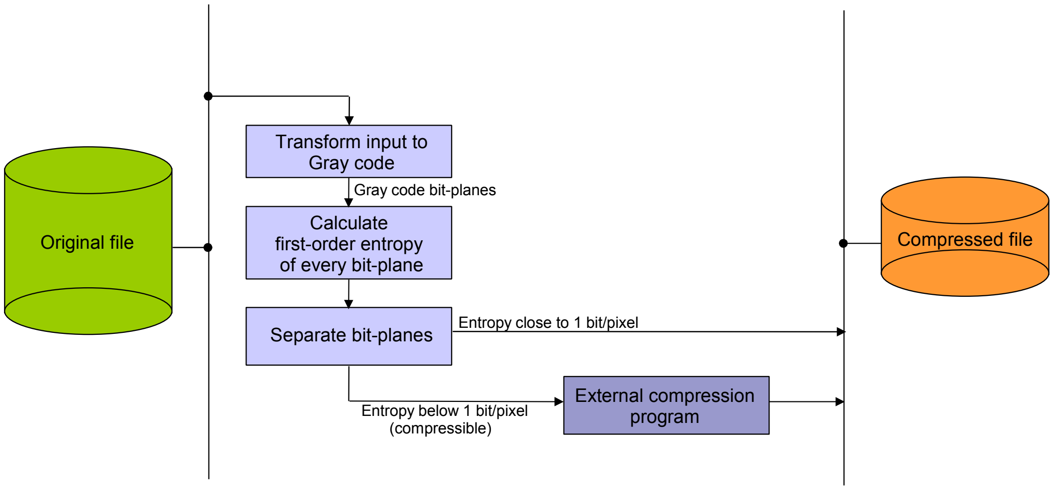

Original file
Transform input to Gray code
Gray code bit-planes
Calculate first-order entropy of every bit-plane
Separate bit-planes
Entropy close to 1 bit/pixel
Entropy below 1 bit/pixel (compressible)
External compression program
Compressed file

Figure 3.9: Data flow of the Compfits algorithm (Véran and Wright 1994).

3.4 Methods Providing Only Lossless Compression

Although, as shown in the previous section, a range of methods for the lossy compression of astronomical images has been developed and compared (Louys, Starck, Mei, Bonarel and Murtagh 1999), lossless compression methods, especially those that are efficient in terms of the computational load, are quite rare.

In general, the hybrid methods presented in the last section generate a high computational load because they follow the transform-based approach, especially if – as with the method of (Press 1992) and (Starck et al. 1995) –

both the transform and the inverse transform have to be calculated for using the lossless mode.

3.4.1 Transform-Based Methods

An integer compressor based on Swelden's lifting scheme (Swelden 1996), a framework for various wavelet transforms, is presented by Louys, Starck and Murtagh (1999). The scheme is compared to Gzip compression and lossless JPEG mode.

For the lossless compression method, Louys, Starck and Murtagh (1999) adapted the low-pass and band-pass operations used in the Haar wavelet transform, to achieve an integer-to-integer transform. The predictor and update operators use, where necessary, a floor truncation function (Louys, Starck and Murtagh 1999).

The compression method based on Swelden's lifting scheme achieves higher compression ratios than Gzip and lossless JPEG compression on a scanned plate image of size 1024 x 1024 (Louys, Starck and Murtagh 1999). Louys, Starck and Murtagh (1999) find that the lifting scheme based compression method is about four times faster than Gzip and lossless JPEG compression is about two times faster than the lifting scheme, but "lossless" JPEG suffers from rounding errors and therefore the lossless JPEG compression cannot be regarded as a truly lossless compression method. Lossless compression ratios range in the order of 1.4 (Gzip) to 1.7 (lifting scheme).

Louys, Starck and Murtagh (1999) highlight that the decompression of an image, compressed using the lifting scheme, takes more than four times longer than lossless JPEG compression and more than twice the time of Gzip.

Louys, Starck and Murtagh (1999) mention that one of the appealing properties of the Haar wavelet transform is that lower resolution versions of the image are exactly two-fold rebinned versions of the next higher resolution level. Therefore, one major advantage of this method is that lower resolution images can be generated using the transform and transmitted with low transmission time requirements. The reconstruction of the high resolution version of the image is possible using update data for the higher resolution levels, a re-transmission of the whole image data is not necessary. The integer-to-integer transform compression method is implemented in the routine "mr_lcomp" of the MR/1 package (Murtagh 2006).

3.4.2 Prediction-Based Methods

A lossless compression method which is based on switched linear prediction with Rice coding (Rice and Plaunt 1971, Rice 1979, Rice 1983, Rice 1991) is presented in (Sabbey 1999). Rice coding was discussed in Section 2.2.1. More details on the compression algorithm and the sky survey system for which it was developed can also be found in (Sabbey et al. 1998). This compression method achieves high data throughput, as it does not require computationally expensive transforms.

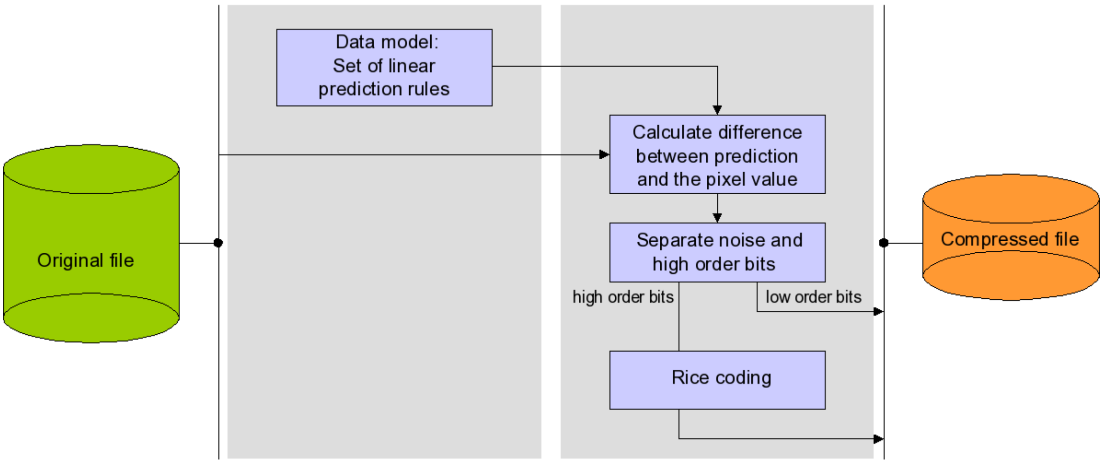


Figure 3.10: Data flow of the adaptive compression algorithm proposed by Sabbey (1999).

Figure 3.10 shows the dataflow and the steps involved in this compression method. Based on (Sabbey 1999), the method can be summarised as follows:

1. **Linear prediction:** First, the difference between each pixel value x and its prediction p is calculated. The prediction used is a linear combination of the already-transmitted consecutive pixel values. The kind of linear combination is chosen according to a set of predefined rules. A modified version of Graham's rules (Netravali and Limb 1980) is used to select one of the linear combinations based on a comparison of the local gradient approximations to a noise threshold. In flat regions, the mean of several neighbouring pixels is selected as the predictor. Otherwise the direction and order of a predictor is selected based on the local correlation (Sabbey 1999).

2. **Prediction residuals mapping:** The prediction residual $\Delta = (x - p)$ is mapped onto non-negative integer values δ:

$$\delta = \begin{cases} -2\Delta - 1 & \text{if } \Delta < 0 \\ 2\Delta & \text{if } \Delta \geq 0 \end{cases}$$

3. **Residual partitioning:** The residuals δ are then partitioned into N low-entropy (noise) and $(16 - N)$ high-entropy bits.

4. **High-order bits encoding:** The $(16-N)$ high-order bits are encoded using Rice coding, a less optimal but faster method than Huffman coding. If the Rice code exceeds a length of 12 bits, uncompressed pixels are transmitted instead. The N low-order bits are transcribed verbatim to the output stream.

5. **Adaptation of the encoding:** The partition size N that is used to separate the residual is adapted by two varying terms N_s and N_r with $N = N_r + N_s$ (Sabbey et al. 1998). The slowly varying term N_s is changed by -1, 0 or $+1$ every image block (for example of 16×16 pixels). The rapid changing component of N, N_r is typically zero but increases in bright zones, when large prediction residuals become probable.

The evaluation of the proposed algorithm included in (Sabbey 1999), using 25 test images, shows that the Rice based compression algorithm achieves, in nearly all cases, at least the same compression ratio as Hcompress, Fitspress, or Compfits + Compact. The encoding program is found to be about ten times faster, in most cases, than the other compression programs. Another approach to include the Rice compression technique exists, but has never been published[2]. It is almost similar to the method of Sabbey (1999) using Rice encoding or sending uncompressed data in bright regions when the prediction residual exceed a certain size.

3.4.3 Entropy Coding

A special lossless and computationally time-efficient coding technique for astronomical interferograms, introducing a new scheme called Signed Huffman

[2] http://stupendous.cis.rit.edu/richmond/rice/rice.html (20.02.2007)

Coding (SHC), is proposed by Weghorn et al. (1996). This coding scheme takes the probability distribution of the pixel-to-pixel variations of astronomical interferograms into account and codes negative and positive variations with the same Huffman coding sequence. An extra sign bit is used to encode the up/down information for all non zero variations. A pixel difference of zero is encoded as one single zero bit while a pixel difference other than zero is encoded with the following pattern: a leading 1, the sign bit and the Huffman coded absolute value of the pixel difference (Weghorn et al. 1996).

The developer of the method demonstrates that this coding scheme requires less bits for all variations larger than ±1. Measurements using sample interferometric data show that SHC yields compression rates of about 4.8 and therefore outperforms Huffman coding and Lempel-Ziv coding which achieved a compression rate of approximately 4.6. Weghorn et al. (1996) mention that de-biasing the images further improves the compression ratio of the SHC method up to about 10, for sample interferograms taken with a special CCD camera at the ESO Danish 1.5-m telescope in La Silla, Chile.

Weghorn et al. (1996) highlight that the computing complexity of SHC is linear with respect to the number of image pixels and report that SHC runs about five times faster than Lempel-Ziv. The Lempel-Ziv method belongs to the set of already standardised FITS file format compression methods that are discussed in the following section.

3.5 Proposed FITS Compression Standard

A FITS compression standard is proposed by White et al. (1999), presenting a general technique for storing compressed images in FITS binary tables.

FITS image compression using variable length binary tables is first proposed by Pence (1994). First, the image is divided into several rectangular sub-images or tiles. Then, each sub-image is compressed into a binary data stream. The binary data stream is stored in the FITS file specific data structure, where each of the sub-image data streams is accessible by random access (Pence 1992). The FITS compression proposal does not specify the compression algorithms to be supported, it provides a general framework how compression methods should be included within the FITS standard. Current implementations of the FITSIO library, the standard library for reading and writing FITS files, include three different compression methods, Gzip, Rice, and PLIO (PLIO stands for the IRAF Pixel List I/O compression algorithm)

(Pence 2007). Floating-point or double-precision images are quantised prior to the compression, using the scheme first proposed by White and Greenfield (1998).

One advantage of dividing the image into regular tiles is that reading programs may randomly access sub-images without having to decompress the entire image. Also, compression methods can adapt better to the data as the smaller image subset has more special properties to exploit. In principle, any kind of compression algorithm can be supported within the FITS standard.

3.6 Existing Methods and Compression Requirements

Section 1.1 of this thesis identified two research challenges for the compression of astronomical images: lossy compression of data for preview and possibly catalogue overlay, and lossless compression for data transmission and archival. With the second challenge, this thesis highlighted that a potential method should be highly adapted to the properties of the data.

With the diverse set of existing methods working in a lossy mode (11 out of 17), the first challenge was resolved satisfactorily, for example by the method of McNerney (2000) for public access to lossy compressed data, or by the method of White (1992). White (1992) explicitly suggests in (White and Percival 1994) a possible application of their method as the basis of a progressive image transmission system.

The research challenge which has not been successfully addressed yet is the second challenge, to develop a lossless highly adapted compression method tailored for data transmission and archival. Section 1.1 summarised details of the requirements for the compression of primary astronomical image data. The literature review showed that the existing methods, which work losslessly and are closely adapted to the properties of astronomical data, are few.

Many of the lossless methods are entropy coders with limited or no ability to decorrelate the data. Most of these methods, like the Rice and Zip based method of White et al. (1999) exploit only the signal value distribution of the data. The method of Weghorn et al. (1996) exploits the distribution of the pixel-to-pixel variations.

Higher-order dependency of the data pixels and from local features are exploited with conventional prediction (Sabbey et al. 1998) and transform-based methods (Press 1992, White 1992, Starck et al. 1995, Louys, Starck and Murtagh 1999). But the majority of these methods are based on potentially time-consuming transforms, and their ability to exploit data-specific properties remains limited. Features at object or region level are not directly addressed with transform-based methods.

Many astronomical images are stellar-field images, though, which consist of regions with diverse properties. Therefore a discriminative choice of compaction methods would adapt compression methods better to this type of data source. Composite source modelling, which treats the signal as an interleaved sequence generated by multiple sources (Strutz 2002), could exploit this different-regions-with-diverse-properties feature, as well as a segmentation-based or object-oriented method. Most of the existing region-adaptive, object-based or content-based approaches are lossy (Huang and Bijaoui 1990, Pardas 1997, Dong et al. 2003).

Dong et al. (2003) highlight that a big challenge in astronomical image compression is how to compress deep-field images. Existing research has not yet successfully addressed the challenge of developing a lossless method which is highly adapted to the distinct properties of astronomical stellar-field images, such as deep-field images. Existing methods, such as the lossless integer-to-integer-based compression method, do not adapt to the distinct properties of astronomical stellar-field images or the adaptation is limited to changing coding step parameters as with the Rice based method of (Sabbey et al. 1998).

Examining the compression methods which have been proposed, mainly during the past eighteen years, the literature review showed that most of them are not suited for the lossless archival of astronomical image data. Only three basic compression methods were included in the FITSIO library, the commonly used library for generating and reading astronomical image data. The data archival problem is tackled by investing huge sums of money in establishing – and even more important – continuously enlarging the storage capacity of data centres for astronomical data.

Data size and rates have increased tremendously in recent years. For example Axelrod (2006) highlights that the Large Synoptic Survey Telescope – with its 3 Gpixel imager and high observing frequency – produces roughly 17 TB per clear night, which drives the need for a method to reduce the size of the data generated. Typical spatial resolutions of single imaging devices used in astronomy nowadays are in the order of 8000 by 8000 greyscale pixels, and the resolution of imaging devices is still increasing and detectors with even higher resolutions[3] are becoming available (Howell 2000). This growth even exceeds the growth of available storage space that can be obtained with fixed or only moderately growing budgets. Also, the larger CCD mosaics that have come into use (Baltay et al. 2002, Kuijken et al. 2004) increase the data amount generated. Besides the planar size of the images, the high dynamic range of astronomical images also contributes to increase the storage requirements of astronomical data sets. Image sensors used in consumer-market digital cameras typically generate low dynamic-range colour images, using colour filters at chip level. Unlike them, astronomical imaging devices generate high dynamic-range greyscale images. An astronomical image is usually recorded using a well-defined optical colour filter to generate an image at a certain frequency band of the light spectrum.

This increasing size of the data sets drives the need for size-efficient compression methods if the data which is generated at high cost shall be archived for future use.

3.7 Summary

This chapter gave a survey of compression methods for astronomical images. The detailed discussion of existing methods for lossy and lossless compression and of methods working in both modes showed that a variety of methods has already been proposed and tested for the compression of astronomical image data.

Two basic research challenges were identified for the compression of astronomical data: firstly, the lossy compression for preview and catalogue overlay, and secondly, lossless compression for transmission and archival. The first of these challenges was successfully addressed. As shown in this survey, a wide range of methods has been examined and some of the proposed methods are used for practical applications (White and Greenfield 1998, McNerney 2000) such as the public dissemination of lossy compressed data for educational purposes (McNerney 2000).

[3] In the year 2007, the University of Bonn presented a 10-by-10 centimetres large 111 mega-pixel CCD. Details can be found on http://www.astro.uni-bonn.de/english/show_news.php?number=23 (01.04.2007).

But, with regard to the second research challenge, the review showed that the existing methods follow the design of a transform-based method such as those of Press (1992), White (1992), and Starck et al. (1995) or in one case a typical prediction-based approach (Sabbey et al. 1998). The overview given in Table 3.1 showed that the majority of existing methods does not employ region-adaptive processing of the data or even segmentation. The existing research has not yet successfully addressed the challenge of developing a lossless compression method highly adapted to the particular properties of astronomical images.

The work presented in this thesis aims at producing such a method. Thus, the next chapter discusses in detail the special properties of stellar-field images, and models which describe them. In particular, the chapter introduces the stellar-field image data model which is used at the core of the novel compression technique proposed in this thesis.

Chapter 4

Special Properties and Models of Stellar-Field Images

> *"If you have ever looked through popular astronomical publications, and at some time you must have, you will know that they are generally full of richly luminous colour photos of distant nebulae and the like – fairy-lit clouds of celestial light of the most delicate and moving splendour. Evan's working images are nothing like that. They are just blurry black-and-white photos with little points of haloed brightness."*
> Bill Bryson (2004). *A Short History of Nearly Everything.*

Lossless compression methods attempt to identify and exploit properties of the data (Salomon 2000, Salomon 2002, Barthel 2003). Using a model of the data, a priori knowledge about the data can be used to exploit the properties of the image for highly efficient compression. One of the main thrusts of this work is to identify and exploit as comprehensively as possible the signal properties specific to stellar-field images (SFI).

This chapter presents an overview of the special properties of stellar-field images. The first section of the chapter describes different categories of astronomical images which can be distinguished by their content. Typical examples of these groups are shown and their use is explained.

Based on mathematical models of astronomical image capture hardware and of stellar-field data, Section 4.2 discusses existing image data models and introduces a model of astronomical stellar-field images. Details of the stellar-field image model used in this research are given in Section 4.2.3. The stellar-field image model is the basis for the development of the compression algorithms which exploit the redundancy in the image signal. In the novel algorithms proposed in this investigation, the model of the image signal is used to segment the image signal into its components, and the image area into two different regions.

The general feasibility of the segmentation-based region-adaptive compression approach and the resulting potential data size reduction are demonstrated in Section 4.3 with a first sample calculation using a typical stellar-field image.

4.1 Categories of Astronomical Images

Astronomical images, which are used for most astronomical observations (astrometry, photometry, and spectroscopy), can be classified into two main groups: photographs and spectra.

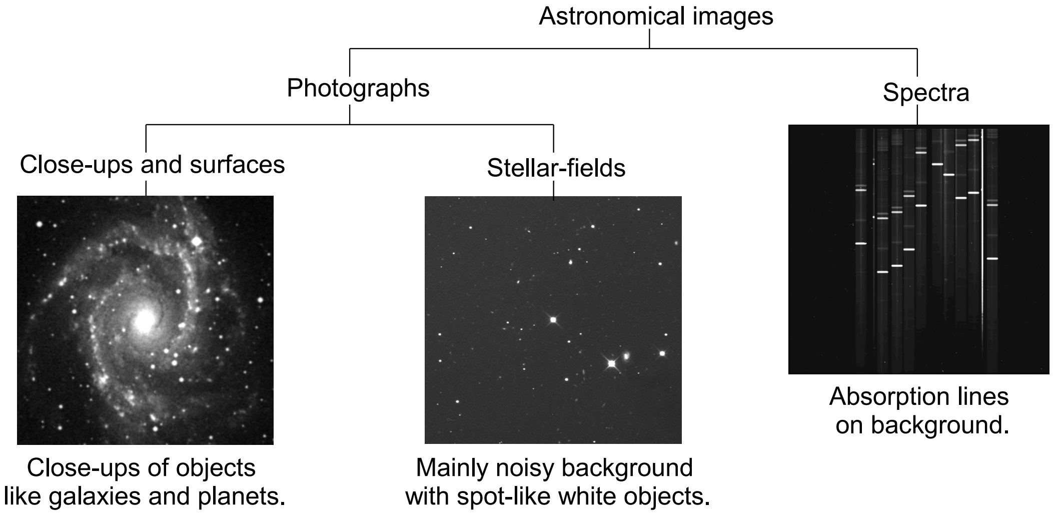


Figure 4.1: Astronomical image types: two types of astrophotographs and a spectrum.

Photographs are used for astrometry and photometry. Astrometry is the science of determining the exact location and possible movement of astronomical sources, it deals with measuring information on positions and distances, motions, and parallaxes of astronomical objects (Howell 2000). Photometry is the science of determining the amount and temporal variation of the flux emitted by an object as a function of the wavelength (Howell 2000). Narrow-

or broad-band filters are used to generate photographs in different wavebands for photometry.

Spectra are observations of the intensity distribution of the energy which is emitted by astronomical objects, sampled by the wavelength (Howell 2000). Like photometry, spectroscopy deals with the amount of flux emitted by an object. A spectrum shows the intensity distribution of the emission, but it does typically not include the spatial distribution of the energy (Howell 2000). Spectroscopy is used in astronomy to derive the physical properties of the observed objects; the absorption lines, which appear in a spectrum, can be used to determine the chemical composition of the stars, but information derived by spectroscopy includes the temperature and motion of the object as well (Howell 2000). This information is of great relevance for astrophysics, the study of physics of the universe.

In addition to the differentiation of spectra and photographs, further distinction into different categories of astronomical photographs is possible. Two important categories for the work presented in this thesis are stellar fields and close-ups:

- Stellar-field images show mainly an empty dark background with spot-like foreground objects. Stellar-field images are typically used for scientific applications.
- Close-ups are large-scale images showing close-up views with details of astronomical objects, such as surfaces and large-scale galaxies. Close-up images are often used to determine details of the exact structure of an object.

Figure 4.1 shows the main types of astronomical images. The stellar-field image which is shown in Figure 4.1 is part of the test images used for the evaluation of the compression methods developed in this research.

Astronomers often try to observe a small field of view at the highest possible magnification, and to obtain information about the furthest and oldest galaxies in order to take a look back to the origins of mankind (Villard and Williams 2006) (Figure 4.2). Although Figure 4.2 only shows a very small sample of the heavens, it can be considered representative of the typical distribution of galaxies in space. Statistically, the universe looks largely the same in all directions (Villard and Williams 2006).

A classification almost similar to the one presented here, appears in McNerney (2000), designating stellar fields and galaxies as "Type 0" images and local planets and surfaces as “Type 1” images. Compared to the classification presented by McNerney (2000), the one presented here discerns more strictly between images with large scale objects and stellar fields. McNerney (2000) assigns galaxy images to Type 0 images, while the corresponding category here only includes stellar fields without spatially extended objects.

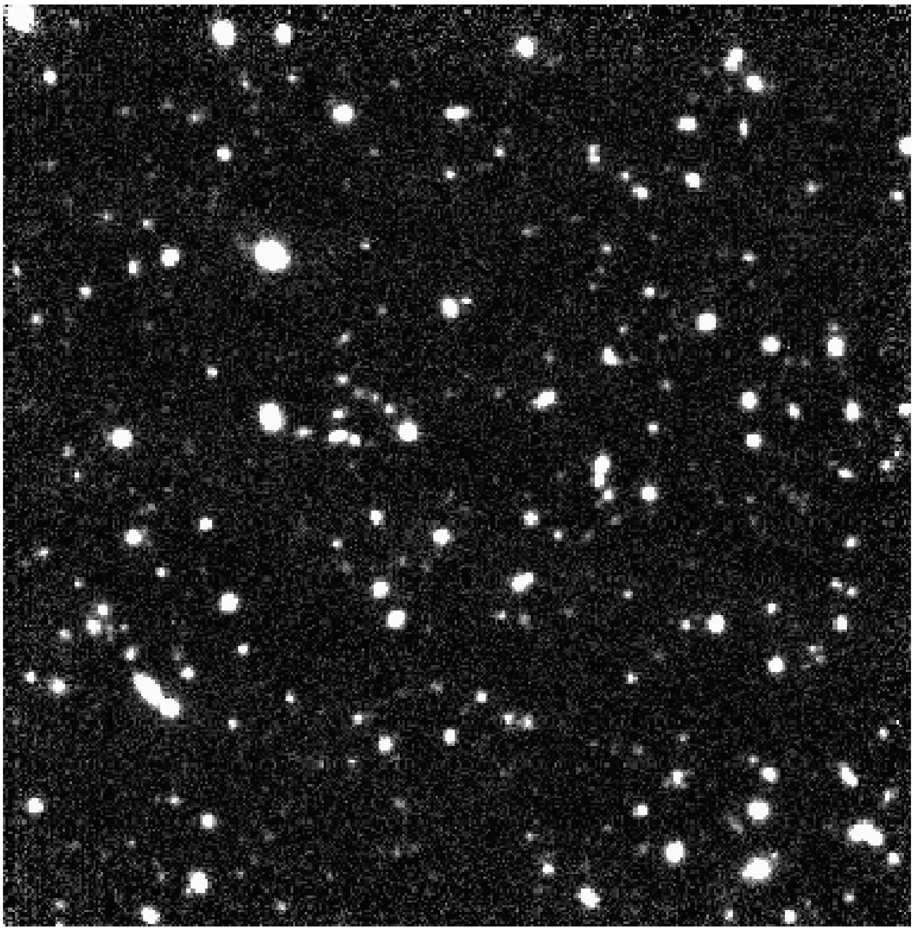


Figure 4.2: Hubble deep-field infrared image ((Villard and Williams 2006), image courtesy of NASA).

The work presented here focuses on the compression of stellar-field images like the Hubble deep-field images, which are the most common astronomical images. Close-ups are, compared to the other types, rarely used for scientific applications although these images are often shown, sometimes even coloured, on posters and in presentations.

4.2 Image Data Models

The characteristics of stellar-field images result mainly from two sources: the properties of the imaging system and the incoming radiation from stellar objects which is processed.

This section discusses models for CCDs presented by Withagen et al. (2007) and Healey and Kondepudy (1994), and their application to stellar field images. The properties of stellar-field images are also discussed. Models for stellar-field images proposed by Fusco et al. (1999) and Diolaiti et al. (2000) are also considered.

The stellar-field image model which is used in this research is derived starting from a general model of a CCD detector and incorporating features of an established model of stellar-field images.

4.2.1 Data Models for Imaging Devices

A key component of the imaging system in astronomy is a CCD or similar semiconductor device. Noise is an important component of astronomical images. Healey and Kondepudy (1994) state that "the various noise sources that corrupt digital pixel values can be quantified, by studying the operation of a CCD camera imaging system". This is done in (Withagen et al. 2007) for different camera types, "ranging from a high-end 10-bit digital camera to a consumer webcam". The model of Withagen et al. (2007) relies on the model of Healey and Kondepudy (1994). It gives the pixel intensity i_t at the time t as:

$$i_t = g_t \cdot (i_0 + \mu_{DC} + N_S + N_R) + N_Q \tag{4.1}$$

The measured pixel intensity results from six components: the true scene intensity i_0, the dark current μ_{DC}, the shot noise N_S, the readout noise N_R, the camera gain g_t and the quantisation noise N_Q which results from the digital-to-analogue converter. The quantisation noise has a uniform distribution $U(-q/2, q/2)$, where q is the smallest pixel value (Withagen et al. 2007). A schematic of a typical signal processing chain used with an astronomical CCD detector can be found in (Mackay 1986).

The different noise types given in the model of Withagen et al. (2007) and Healey and Kondepudy (1994) are important for astronomical applications. In order to conserve even the faintest objects, the sampling of the signal is typically chosen to preserve the noisiness of astronomical images. “The gain should be chosen to make the root mean square readout noise equal to a few bits of the analogue-digital converter” (Mackay 1986). While typical consumer-application images profit from a high signal-to-noise ratio, the signal-to-noise ratio in astronomical images may be low in certain image areas. The reason is that the input – which may consist of a few photons from a faint source – itself is very weak.

In (Healey and Kondepudy 1994), the authors show the contribution of different noise parts, as a fraction of the total noise variance. This is also analysed for astronomical CCDs by Mackay (1986). He shows that for low signal levels the read out noise dominates the signal. But for higher signal levels the shot noise dominates.

This result is also supported by Withagen et al. (2007), who summarise their experiments as follows: “Our Experiments show that the additive noise contributions (N_R and N_Q) are equal to or smaller than the multiplicative noise contribution for sufficiently large intensity values (larger than $10-30\%$ of the intensity range)” (Withagen et al. 2007).

In addition to the different noise types, dark current may impose an uncertainty to the signal measured with a CCD. Mackay (1986) reports that the dark current μ_{DC} of CCDs follows the well-known diode law, where A and B are constant, k is the Boltzmann's constant and T the temperature.

$$\mu_{DC} = A \cdot e^{-\frac{B}{kT}} \tag{4.2}$$

As Equation 4.2 shows, dark current can be efficiently reduced by cooling, which is done down to temperatures of -60° C to -160° C (Mackay 1986) for astronomical applications. Withagen et al. (2007) report that for the consumer cameras that they examined, “the dark current is for all cameras lower than the additive noise, so we conclude that for pixels with a sufficient large intensity we can neglect dark current”. With the weak irradiation input processed in astronomy, and tremendously longer exposure times than other CCD-based imaging systems, cooling is often necessary for astronomical applications. “In photon-counting systems at the lowest light levels, dark current is often significant in long exposure times” (Mackay 1986).

But dark current is not the only almost constant additive component which contributes to the signal in the case of astronomical CCDs. “To avoid negative numbers in the output image, CCD electronics are set up to provide a positive offset value for each accumulated image" (Howell 2000). The author Howell (2000) mentions that a typical bias level may be 400 units of the analogue-digital converter, which were especially required in the past, due to higher read-out noise, temporal drifts of the CCD, and poor stability of the electronics.

With the results of (Mackay 1986), that dark current can be almost completely removed by appropriate measures, μ_{DC} can be neglected. It can be disregarded in Equation 4.1 especially if an additional bias level b_t is introduced to the model. It may be (to a certain degree) time- and position-dependent, thus it is denoted b_t.

$$i_t = g_t \cdot (i_0 + N_S + N_R) + N_Q + b_t \tag{4.3}$$

Now, with respect to the discussion given above that at low signal levels the readout noise dominates the signal and at high levels the shot noise dominates, the pixel intensity i_t at time t is given by Equation 4.4. The quantisation noise is omitted, as it is much smaller than the read-out noise.

$$i_t = \begin{cases} g_t \cdot (i_0 + N_R) + b_t, & \text{if } i_0 \approx N_R \\ g_t \cdot (i_0 + N_S) + b_t, & \text{if } i_0 \gg N_R \end{cases} \tag{4.4}$$

Therefore, the characteristics and the amount of noise present in an image depend on the measured intensity. The shot noise N_S, which dominates in bright image regions is Poisson-distributed noise. Its standard deviation depends on the square root of the intensity. "The read out noise N_R has a Gaussian distribution" (Withagen et al. 2007). For his instrument Bolte (2006) shows that at new moon in the R-band [where the dependency to the moon is high], the read out noise is equal to just one second of integration time.

4.2.2 Data Models for Stellar-Field Images

The observed intensity i_0 (Equation 4.4) in the case of astronomical images consists of two sources, sky background and background emissions (which are apparent in all pixels), plus in some cases true irradiation from stellar objects. How dark the background intensity is depends on the sky brightness. "Sky brightness is different at different sites, in different band-passes and of course varies with the phase of the moon and distance from the moon" (Bolte 2006). Data models for stellar-field images describe the irradiation from stellar objects more precisely.

A deconvolution method for adaptive optics images of stellar fields is presented in (Fusco et al. 1999). Adaptive optics compensates, to a certain degree, for the angular resolution limits of ground-based telescopes. The deconvolution method is used to restore star parameters such as the positions and photometric properties.

Fusco et al. (1999) model the signal of stellar-field images as the convolution of the object named $o(r)$ with a point-spread function $h(r)$ which characterises the telescope and the atmospheric turbulence plus noise. The detected intensity distribution $d(r)$ of the image is given by Fusco et al. (1999):

$$d(r) = o(r) \cdot h(r) + n(r) \tag{4.5}$$

This model includes the noise $n(r)$, which is according to Fusco et al. (1999) an additive zero-mean random process resulting from photon and detector noise. That fits well with the more precise treatment of CCD detector noise by Withagen et al. (2007) and Healey and Kondepudy (1994). Fusco et al. (1999) model the observed stellar field as a sum of Dirac functions as follows:

$$o(r) = \sum_{i}^{n} \alpha_i \delta(r - a_i) \tag{4.6}$$

α_i and a_i are the intensity and the position vector of star number i, n is the number of stars. Fusco et al. (1999) state that to this model "a constant or slowly variable parameter can also be added to account for sky background and underlying emission".

A very advanced data model, for the analysis of isoplanatic high resolution stellar-field images, is presented in Diolaiti et al. (2000). This data model is used for high precision astrometry and photometry of crowded fields acquired under the assumptions of accurate point-spread function knowledge, isoplanatism, and correct sampling of the input data. Although the StarFinder image analysis algorithm aims at different goals, it uses a data model which is very similar to the one used in this work. The stellar-field is assumed to consist of a superposition of shifted and scaled point-spread functions lying on a smooth slowly varying background, which results from faint undetected stars, possible faint diffuse objects and noise.

For deblending stars Lupton (2004) uses a stellar-field model similar to the one of Diolaiti et al. (2000). "The image is made up of a set of δ-functions convolved with a known point-spread function, φ" (Lupton 2004):

$$I = S + \sum_r F_r \delta(\mathbf{x} - \mathbf{x}_r) \otimes \phi + n \tag{4.7}$$

Where I is the observed intensity, S the sky level, δ is the delta-function, F_r the flux of the star r, ϕ the point-spread function and n the noise.

4.2.3 Simplified Integrated Data Model Suitable for Segmentation-Based and Region-Adaptive Compression of Stellar-Field Images

In this section, a simplified model for stellar-field image data is derived by integrating the relevant elements of models of CCD imaging detectors and of stellar-field images which were presented in the preceding sections. Here, the intensity i_0 which is measured for a stellar-field image may be composed as follows:

$$i_0 = b_g + s_0 \tag{4.8}$$

where b_g is an almost constant component due to sky emission and s_0 is the result from the convolution of the Dirac function with the point-spread function. The insertion of Equation 4.8 into Equation 4.4 results in:

$$i_t = \begin{cases} g_t \cdot (b_g + N_R) + b_t, & \text{if } b_g \approx N_R \text{ and } s_0 = 0 \\ g_t \cdot (b_g + N_S) + b_t, & \text{if } b_g \gg N_R \text{ and } s_0 = 0 \\ g_t \cdot (b_g + s_0 + N_S) + b_t, & \text{if } b_g + s_0 \gg N_R \end{cases} \tag{4.9}$$

This can be simplified to:

$$i_t = \underbrace{g_t \cdot b_g + b_t}_{b(\mathbf{x})} + \begin{cases} g_t \cdot N_R, & \text{if } b_g \approx N_R \text{ and } s_0 = 0 \\ g_t \cdot N_S, & \text{if } b_g \gg N_R \text{ and } s_0 = 0 \\ \underbrace{g_t \cdot s_0}_{s(\mathbf{x})} + g_t \cdot N_S, & \text{if } b_g + s_0 \gg N_R \end{cases} \tag{4.10}$$

As $g_t \cdot b_g + b_t$ are almost constant throughout the whole image, from the perspective of the final image signal, they are termed as a bias level $b(\mathbf{x})$. Still, this signal component which is registered on the CCD in the absence of exposure to direct light sources, typically leads to a smooth variation of the image background, which typically manifests itself as a smooth, ramp-like variation of the background.

In addition to sensor-specific background variations mainly affecting b_t, the background region may contain light diffusion from large nearby astronomical sources or terrestrial man-made light pollution (which affect b_g). Figure 4.3 shows a section of a stellar-field image, with such a varying background. Therefore, the bias which is termed $b(\mathbf{x})$ in Equation 4.11 is position dependent. The vector $\mathbf{x}$ states the position within the two-dimensional image.

The final image signal is denoted $i(\mathbf{x})$. In any of the cases listed above, we find that the signal consists of three components: The bias $b(\mathbf{x})$ which was already discussed, some sort of noise which may be Poisson or Gaussian distributed, and a signal component $s(\mathbf{x})$ resulting from stellar object irradiation, which may be zero in certain regions.

Depending on the almost constant sky background emission b_g, the noise component, which is termed $n(\mathbf{x})$ from the perspective of the final measured signal, consists of shot noise or readout noise.

Alternatively, and this case is not explicitly listed separately in Equation 4.10, it is a combination of both. Now we can summarise the results into a simplified model of stellar-field CCD data, which is used for segmentation-based and region-adaptive compression of stellar-field images, as devised in this thesis (see Equation 4.11).

$$i(\mathbf{x}) = s(\mathbf{x}) + n(\mathbf{x}) + b(\mathbf{x}) \tag{4.11}$$

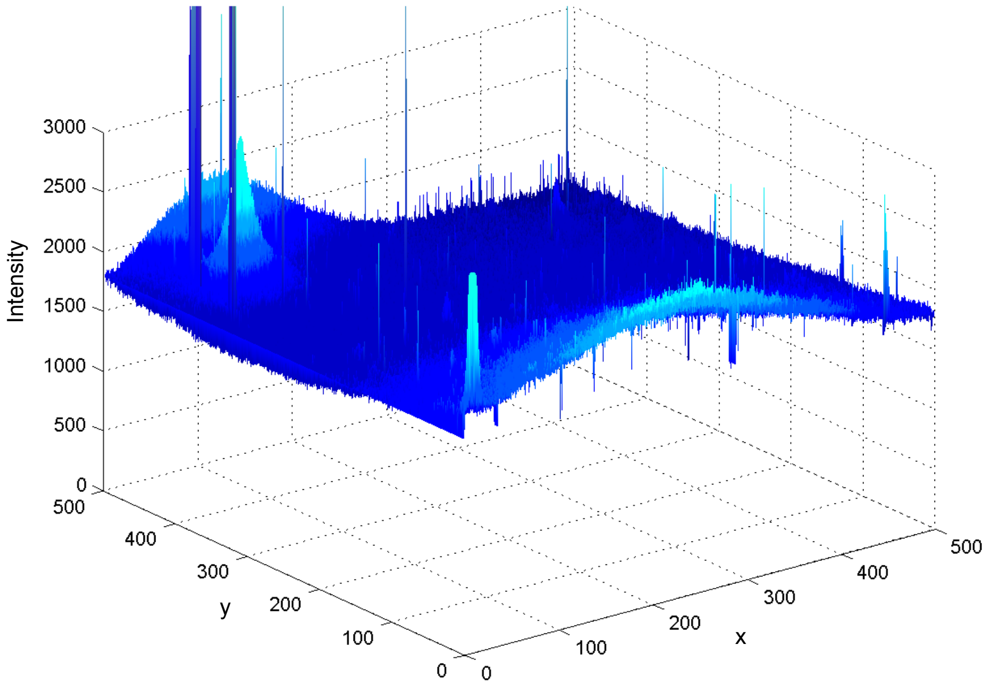


Figure 4.3: Three dimensional plot of the signal $i(\mathbf{x})$ of image "com0001" which shows a strongly varying background. This image is part of the test image set used in this investigation.

4.3 Potential Data Reduction Resulting from the Stellar Field Image Model Used in this Research

In the previous section, based on properties of charge-coupled devices and stellar-fields, a basic model of astronomical image data was formulated. In this section the focus lies on the discussion of how data conform to this model can be compressed. Unlike the last section, where the key objective was to give a definition of signal components, here the data is treated from the perspective of the final digitised measurement read out from the charge-coupled device.

4.3.1 Key Idea

The key idea behind segmentation-based and region-adaptive compression of stellar-field images, anchored on signal modelling, is that typical stellar-field images consist of a dark, in a first approximation nearly constant background, covered with white, small, bright, point-like objects such as stars or cosmic ray hits. The high dynamic range is, especially in stellar-field images, only required in specific image areas, as "the astronomical image has the characteristic of being a set of astronomical sources in the sky background whose values are not zero" (Huang and Bijaoui 1990).

The bright area, where a high dynamic range is required to encode the data, is typically very small, compared to the dark background region where only a very modest dynamic range is used (Figure 4.4). In astronomical photographs, the dark area containing mainly plain noise fills about 90 percent of the entire image (Sabbey et al. 1998). As the background noise, in professional astronomical images, typically has a low dynamic range, it requires only a low bit rate.

As seen in Section 4.2, the dark background region has a nearly constant level overlaid with noise, which can be considered as Gaussian or Poisson distributed (Howell 2000, Starck and Murtagh 2001). A conjoined encoding of both image regions (bright regions and the background regions) leads to additional encoding overhead on the large background data, where even a few bits per pixel may worsen the compression ratio.

Due to the very different properties of the two image areas and the lower dynamic range of the dark areas compared to the bright areas, algorithms based on segmentation and region-adaptive bit allocation may outperform other approaches which work well on images with other or less region-specific properties.

With region-adaptive bit allocation, pixels can be encoded using the minimum number of bits that corresponds to the dynamic range, at the cost of transferring the information about the region affiliation of the pixels as side information.

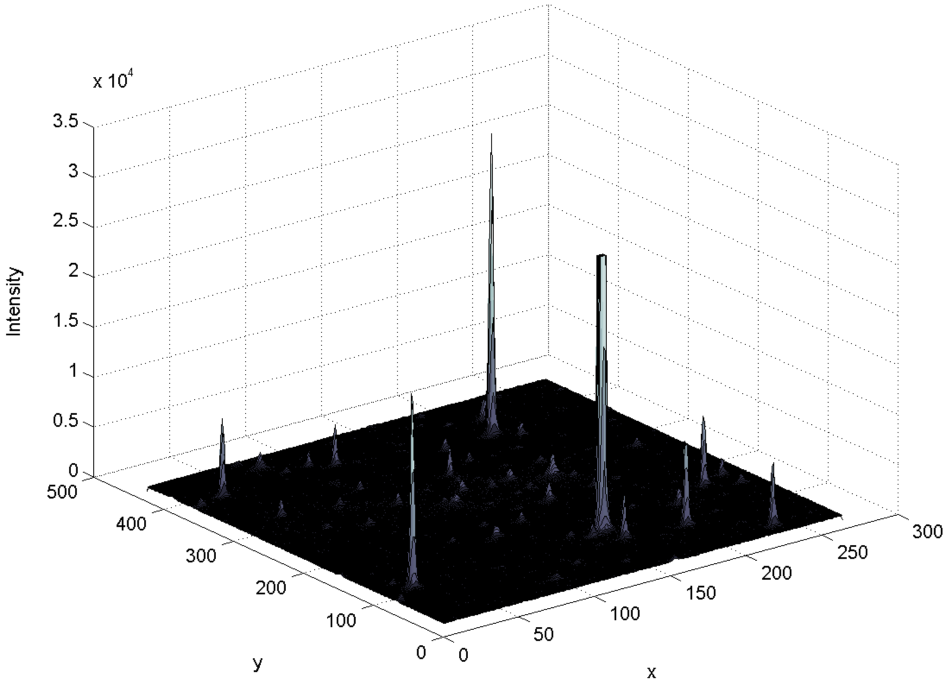


Figure 4.4: Three dimensional plot of the image signal $i(\mathbf{x})$ of image "gal0002". This image is part of the test image set used in this investigation.

The image is split into to two disjoint regions: region-of-stellar-objects and background region. The different regions are defined by the size of signal components in the model of stellar-field CCD data given in Equation 4.11.

Definition 5 (Region-of-Stellar-Objects) *In the region-of-stellar-objects, the image signal component $s(\mathbf{x})$, which results from the incident of irradiation from astronomical sources, is not zero and sufficiently large to be distin-*

guished from the background noise $n(\mathbf{x})$. All image signal components from Equation 4.11 are present.

Definition 6 (Background Region) *In the background region, the image signal component $s(\mathbf{x})$, which results from the incident of irradiation from astronomical sources, is zero or too small to be distinguished from the background noise $n(\mathbf{x})$. The observed image signal $i(\mathbf{x})$ consists only of the image background level $b(\mathbf{x})$ and noise $n(\mathbf{x})$.*

Using a sample image from the Near Earth Asteroid Tracking (NEAT) project[1], this section shows that segmentation and region-adaptive bit allocation alone may already lead to a significant reduction of the required storage space. The sample image contains relatively large objects compared to typical stellar-field images.

Section 4.3.2 includes a sample calculation which shows that high compression ratios – compared to other lossless methods – are possible through image segmentation followed by region-adaptive bit allocation. The first sample calculation to show the basic feasibility of the approach of compression-by-segmentation and signal modelling is based on the image data model given in Equation 4.11, using only a single background estimate for the whole image.

During the imaging process, parameters have to be chosen carefully by the astronomer for obtaining a good signal-to-noise ratio (SNR). In Equation 4.11 the image signal component $s(\mathbf{x})$, which is generated by the incident light, should be large compared to the noise $n(\mathbf{x})$, but the noise has to be preserved as well. The noise within professional astronomical photography typically influences only a few of the lower bits of the intensity value (Mackay 1986).

The image signal $i(\mathbf{x})$ of a sample line section (white line in Figure 4.5.a) is shown in Figure 4.5.b, in which a background level estimate and the dynamic range of the noise portion are marked. Figure 4.5.b clearly indicates that the whole dynamic range of 32 bits is required only within the bright areas of the image. Separating the background regions from regions of stellar objects, and coding both parts differently has the potential to lead to a significant reduction of the required storage space; this potential is estimated using the sample calculation presented in the next section.

[1] The Near Earth Asteroid Tracking (NEAT) project homepage can be found at: http://neat.jpl.nasa.gov/ (28.02.2005). Images from the NEAT project can be downloaded from: http://skyview.gsfc.nasa.gov/cgi-bin/skvbasic.pl (28.02.2005).

4.3.2 Sample Calculation

In this section, a sample calculation of compression ratio shows that even when only two basic properties are exploited, high compression ratios comparable to those obtained with existing methods can be achieved.

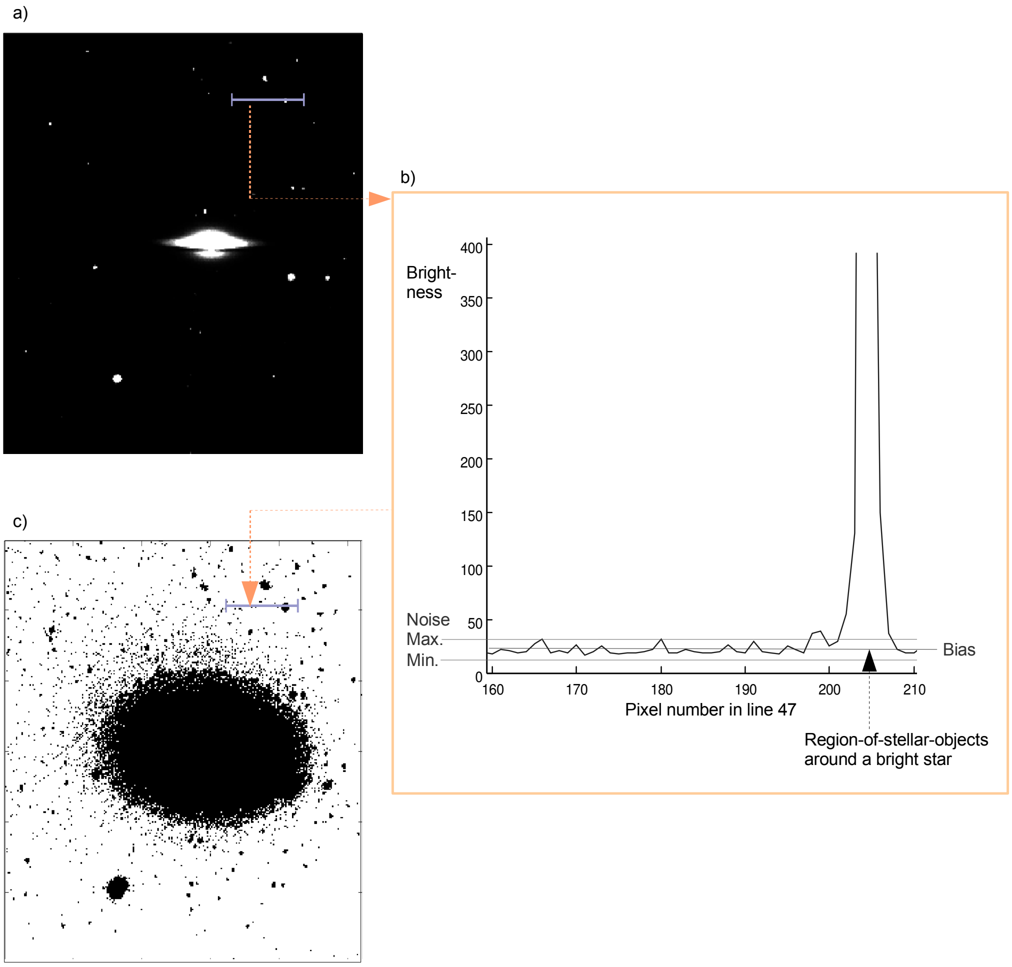


Figure 4.5: (a) Astronomical photograph sample. (b) Plot for an image scan line which is denoted by the blue line within the photograph given in (a). (c) The decision map for background and region-of-stellar-objects pixels.

An estimate of the possible compression, achievable by separating background regions from regions of stellar objects, can be obtained by analysing the number of necessary bits for the different regions. Regions of stellar objects and background regions are identified by analysing the image signal shown in Figure 4.5.b. A constant background level of 24 is used throughout the whole image, while the noise portion has got a dynamic range of 3 bits. Therefore, it can be encoded with 4 bits, including one sign bit.

In this sample measurement, 79.2 percent of the image is background region, and 20.8 percent form the so-called region-of-stellar-objects.

The calculation given in Equation 4.12 estimates the fraction of the data size which will remain if the number of bits per pixel is allocated according to the dynamic range of the region containing the pixel. The calculation is based on the assumption that background noise and region-of-stellar-objects data are stored with simple binary encoding using the maximum number of bits required for the individual data component (3+1 bits for the background noise and 32 bits for the region-of-stellar-objects data). The last component of the sum shown in Equation 4.12 corresponds to a binary region demarcation information mask, encoded as a 1 bit-per-pixel array which is included to indicate which of the two regions the individual pixel belongs to.

$$0.792 \cdot \frac{3+1}{32} + 0.208 \cdot \frac{32}{32} + \frac{1}{32} = 0.338 \tag{4.12}$$

The estimated compression ratio for the sample image is nearly 3, even without using well known compression methods. Applying appropriate compression methods to the data of both regions can lead to further compaction. Another possibility to enhance the compression is to use more efficient methods to describe the region demarcation information. Such methods could further reduce the image size by reducing the last addend in Equation 4.12.

4.4 Summary

This chapter gave an overview of common categories of astronomical images. Stellar-field images are an important category of astronomy data, which is extensively used for photometric applications, in particular. Stellar-field images show mainly an empty dark background with spot-like foreground objects.

Luminous colour photos of impressive and large astronomical objects, like clouds and nebulae, which are often shown on posters and illustrations, are mainly used for presentation purposes. The chapter identified the main relevant features of “real” stellar-field data. These are: the relatively small and localised image areas with high dynamic range, the large areas containing additive noise which is of low dynamic range and typically is Gaussian-distributed, and the spatial segregation of these two image components. These features offer the potential for appreciable lossless data compression.

The spatial segregation of the two types of image components and the spatial compactness of the high dynamic range areas underpin the idea for the algorithms, based on segmentation and region-adaptive bit allocation, which are developed in this work.

Data models for CCD sensors presented by Withagen et al. (2007) and Healey and Kondepudy (1994) were discussed, as well as data models formulated by Fusco et al. (1999) and Diolaiti et al. (2000) for astronomical stellar-field images. As the astronomical image data results from a combination of both CCD sensor properties and stellar-field data properties, a combined model was derived for the data under investigation in this thesis.

The model consists of three additive components: a component representing the spot-like stellar objects, a bias level representing the empty sky background and the dark current of the imaging device, and a noise component representing the shot noise or readout noise of the imaging device.

The model is an important part of the compression algorithms developed in this research, as detailed knowledge of data properties is required to develop algorithms closely adapted to the properties of stellar-field images. A sample calculation, applied to a stellar-field image, presented in the last section, showed that a significant reduction of the required storage space can be achieved as a result of image segmentation followed by region-adaptive bit allocation.

The next chapter uses the integrated stellar-field image model, presented in this chapter, at the core of the new compression algorithms which are proposed in this thesis. Three gradually refined compression algorithms based on image segmentation and signal modelling are devised in the next chapter to achieve increasingly higher lossless compression capability.

Chapter 5

Region-Adaptive Image Compression Based on Segmentation and Image Modelling

A relatively young idea in data compression, which is just at the moment finding its way into newer standards (see Chapter 2), is adaptation of the compression to local features, which can lead to higher compression ratios if the data shows different properties in different regions. Following the observation that stellar-field images are an outstanding example of data with properties that are highly area dependent, novel compression algorithms based on segmentation and region-adaptive bit allocation are developed in this chapter. Also, the sample calculation, given in Chapter 4, showed that segmentation-based compression, which incorporates region-adaptive bit allocation, may lead to a significant lossless compression of the data.

The basic image model discussed in Chapter 4 is used, and a set of algorithms is presented, in this chapter. Three alternative versions of the compression algorithm are presented here; they cover different algorithmic options, each with its own trade-off between complexity and optimality of compression. The proposed three algorithmic options mainly differ with respect to their choice of encoder and the choice of signal decomposition parameters. The algorithmic options are numbered sequentially, the first image compression by segmentation and signal modelling (ICSSM), called ICSSM 1, is presented in Section 5.1. The data model and most decomposition details included in ICSSM 1 apply with minor variations for the other algorithm variants as well. ICSSM is based on the image data model presented in Section 4.2.3. ICSSM 1 makes use of the data model to efficiently separate image data components, and it applies only very basic and simple encoding methods for all data components. The ICSSM 1 version mainly serves as a proof-of-concept algorithm.

The second algorithmic option (ICSSM 2), presented in Section 5.2, processes the image components differently but uses a common final encoding step. ICSSM 2 keeps the image intensity values in region-of-stellar-objects pixels verbatim, whereas noise values are encoded relative to the estimated background. For the common final encoding step for all data components, methods which are typically used in state-of-the art methods for general-purpose compression of binary data are evaluated.

Finally, for highest compression performance, ICSSM 3 selectively applies data compaction methods best matched to the properties of each data component. ICSSM 3 is discussed in Section 5.3. It is intended to be the most size-efficient version of the image compression algorithm based on segmentation and region-adaptive encoding. Evaluation of different methods for treating each data component specifically is performed for ICSSM 3 in order to determine good choices of encoders for each data component.

While this chapter presents the different algorithm versions, the performance evaluation of all algorithm versions is discussed in Chapter 6, where empirical results obtained using the standard test image set are reported.

5.1 Proof-of-Concept Compression Algorithm Using Binary and Huffman Coding (ICSSM 1)

ICSSM 1, the first variant of the compression algorithms based on segmentation and region-adaptive bit allocation, aims to provide a simple algorithm for the decomposition and encoding of the image data. It relies on the image data model presented in Chapter 4, and applies a histogram-based segmentation method – which is also used in all other ICSSM algorithm variants – for the decomposition of the image data. The segmentation includes a tile generation step and estimation of local background levels. The tile- and histogram-based image segmentation algorithm is discussed in Section 5.1.1.

Section 5.1.2 discusses the relation of the data decomposition applied in the ICSSM compression algorithms to image component extraction techniques used in tools like StarFinder (Diolaiti et al. 2000). Section 5.1.3 discusses the relation of the novel algorithm to existing methods for compressing stellar-field images. The details of the proof-of-concept algorithm and its parameters are discussed in Section 5.1.4. Finally, to ensure the losslessness and efficiency of the segmentation step, the validity of the segmentation step is discussed in Section 5.1.5.

5.1.1 Tile- and Histogram-Based Image Segmentation

In stellar-field images, such as the one shown in Figure 5.1, the noise typically has a low dynamic range; therefore it requires only a low bit-rate.

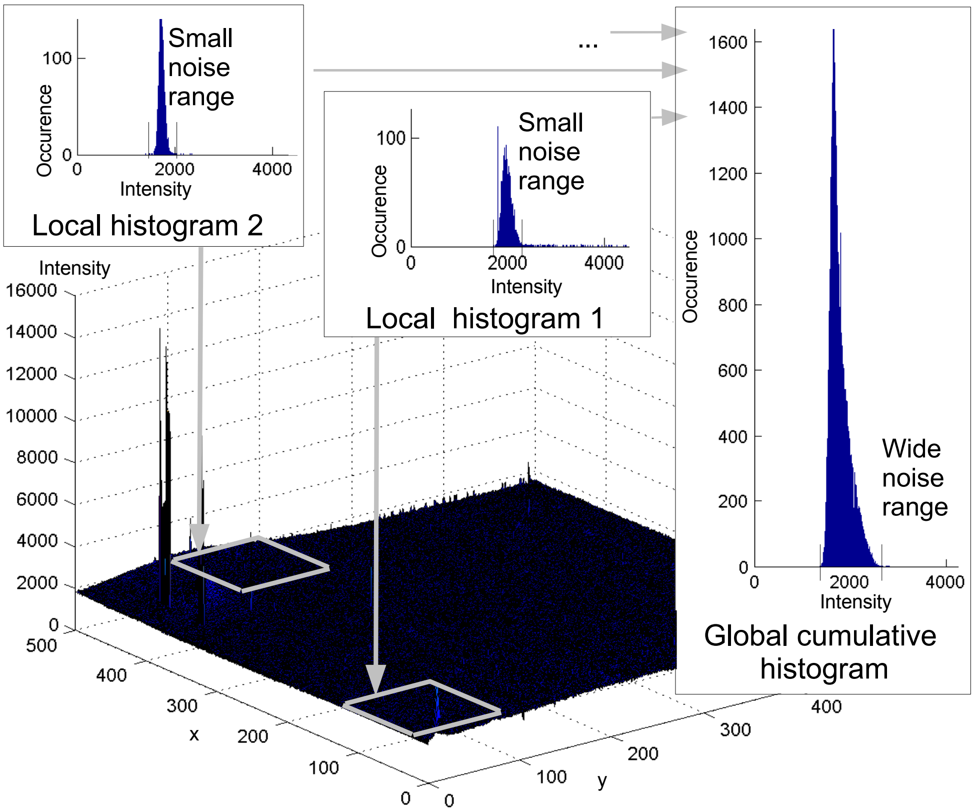


Figure 5.1: Illustrative example of pixel distributions (two local histograms and a global histogram) from the background of a stellar-field image. The global histogram has a wider noise range due to different centre-points of the local histograms.

The background variation widens the noise range in the histogram of the whole image and leads to high estimated noise bit-rates. Figure 5.1 shows two local histograms and a cumulative histogram of the image “com0001”. The background variation has a low frequency, it can thus be neglected in a local image area, and the additive noise which is superimposed on the background can still be considered as approximately Gaussian. But in a larger area, the cumulative histogram which in effect is a superposition of all “local histograms” with different centre-points, will become much wider than the local histograms.

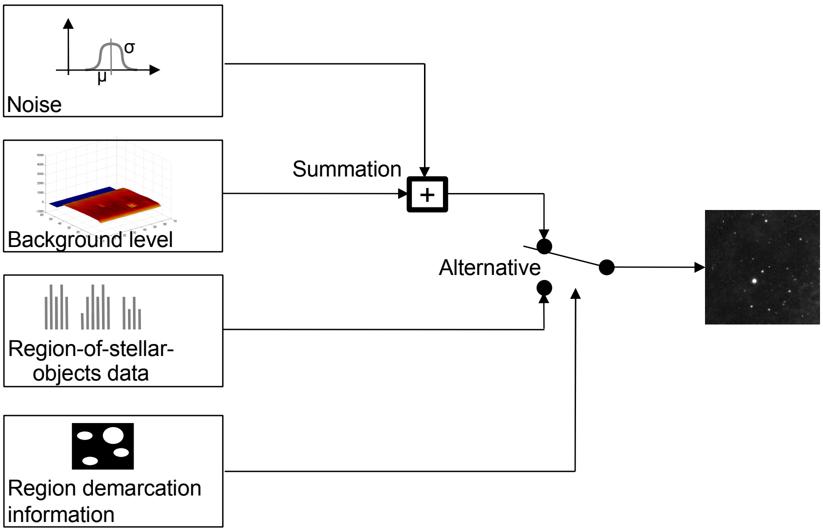

σ
μ
Noise
Summation
+
Background level
Alternative
Region-of-stellar-objects data
Region demarcation information

Figure 5.2: Data components generated and used by the image segmentation and region-adaptive bit-allocation based algorithm.

Encoding the data relative to a local background estimate would reduce the amount of data that has to be spent for encoding the noise. A histogram-based method for estimating the image background within regular image tiles (Seaman et al. 2006) is included in the segmentation and region-adaptive bit allocation-based image compression algorithm. Figure 5.2 shows the data components used with the segmentation-based method. Depending on the region demarcation information, image pixels are either associated to the region-of-stellar-objects or to the background region. Pixel data from the background region consists of a sum of the background level plus the noise.

Different methods were investigated to estimate the local background level $b(\mathbf{x})$ of the image. Straightforward techniques to estimate the background by applying smoothing operations, like median filtering, which replaces each pixel with the median computed over a suitable neighbourhood, failed. Exploratory experiments, which were conducted in early stages of the research reported in this thesis, support what Diolaiti et al. (2000) report as well. Namely, that these smoothing methods over-estimate the background level underneath strong peaks. Therefore, the method devised for segmenting the image, in the ICSSM algorithms, is based on the local histogram of the data. An estimate of the mean background noise can be obtained by means of histogram fitting techniques (Almoznino et al. 1993, Bijaoui 1980) and (Diolaiti et al. 2000). Assuming that the intensity of the sky radiation is distributed normally around a typical value, the histogram of the observed intensity levels should be quite similar to a Gaussian distribution, whose mode and standard deviation represent respectively the sky level and the associated noise (Diolaiti et al. 2000).

Therefore, in the ICSSM algorithms, the background level $b(\mathbf{x})$ of a small square image tile is determined as the mode of the image-tile histogram. The size of the tile determines the size of the neighbourhood that is used to determine the histogram; hence, it also determines the coarseness of the estimates of background levels in the image.

5.1.2 Region Segmentation Versus Detection of Point Spread Functions

As discussed in Section 4.2, the image signal of astronomical images can be modelled as the convolution of the object named $o(r)$ with a point-spread function $h(r)$ which characterises the telescope and the atmospheric turbulence. In addition to the image segmentation into region-of-stellar-objects and background region, at the first glance, the decomposition of the image into stellar object basis-functions convolved with a point-spread function seems to be a further option to simplify the region-of-stellar-objects data representation.

A decomposition into a point-spread function superposition is applied in the StarFinder image analysis tool of Diolaiti et al. (2000). This StarFinder data model is very similar to the one used for the ICSSM algorithms. Diolaiti et al. (2000) apply segmentation in a different way and with different aims to the ICSSM algorithms. The procedure of Diolaiti et al. (2000) derives first a point-spread function digital template from the brightest isolated field stars; then a catalogue of presumed objects is formed, searching for the relative intensity maxima in the image frame (Diolaiti et al. 2000). Suspected objects in the original list are accepted successively on the basis of their correlation with the point-spread function template (Diolaiti et al. 2000). A synthetic field consisting of already detected objects is created successively. The synthetic field becomes more and more similar to the observed image, and fainter and fainter remaining objects can be detected. StarFinder aims to generate high precision relative astronometric and photometric measurements automatically (Diolaiti et al. 2000).

A number of limitations argue against a decomposition of the image data into superpositions of point-spread function for the purpose of compression.

Firstly, a point-spread function based segmentation method requires a precise knowledge of the point-spread function. Very precise point-spread function models would be needed for subtracting the bright objects without or with small residuals. But a precise knowledge of the point-spread function cannot be guaranteed for a stellar-field image compression method, which should work with arbitrary stellar-field images. Precise point-spread function models are – if at all – only contrivable with prior knowledge of the optical properties of the instrumentation used or with high computational costs.

Secondly, depending on instrumentation, observation angle, and CCD positioning of the observing instrument, the point-spread function may vary extremely, depending on its position on the image. The image decomposition of Diolaiti et al. (2000) is much more difficult, if not impossible, in the case of space-variant point-spread functions.

Thirdly, the residuals from the subtraction of the point-spread function would vary in their size depending on their position on the noise data stream. Region-of-stellar-objects data includes Poisson distributed shot noise (Howell 2000, Pool et al. 1998, Cohn 2006), which results from the discrete nature of light and random photon fluctuations. Shot noise increases with the intensity of the pixel. Therefore, a potential residual data stream would be hard to compress because the resulting noise data would not be stationary.

Finally, the idea of segmenting the image into a superposition of scaled point-spread functions and background is not advisable for a stellar-field image compression algorithm, if time-efficiency is required. Identifying precisely the point-spread function parameters would render an integration of the ICSSM algorithm into the storage and archival pipeline of virtual observatories difficult due to the computational complexity of the decomposition. The iterative segmentation and calculation of the correlation coefficients is quite time consuming. Diolaiti et al. (2000) report that the StarFinder code consumes five to ten minutes, for a 386-by-386 pixel image, on a 350 MHz PC. The basic idea of the model of Diolaiti et al. (2000) is similar to the one used here for segmentation-based compression in so far as the image data is modelled as a superposition of noise and the signal component resulting from the incident of stellar object irradiation. However, unlike Diolaiti et al. (2000), the ICSSM decomposition does not decompose regions of stellar objects into point-spread function superpositions plus noise.

5.1.3 Relation of Existing Compression Methods to those Developed in this Thesis

This section examines the relation of the novel compression method, based on image segmentation and a region-adaptive bit allocation, to existing methods which were discussed in the literature review (Chapter 3).

The main thrust of the research conducted for this thesis is on the development of a lossless segmentation-based method, which primarily relies on properties of stellar-field images for transforming the data into a representation which can be further compacted efficiently. As can be seen in Table 3.1, there has not been much contribution in the area of a segmentation-based compression methods specific for astronomy images.

Existing methods are lossy (McNerney 2000, Huang and Bijaoui 1990, Boussalis et al. 2004, Pardas 1997, Dong et al. 2003) or like the PLIO method – which could as well be classified in this category – are very specific and not well suited for stellar-field image compression. Although they are lossy, two of the more recent methods (McNerney 2000, Boussalis et al. 2004) are interesting candidates, regarding a comparison. The recently proposed lossy compression method by Boussalis et al. (2004) proposes an object-based method, which classifies objects into two categories. This supports the approach adopted in the method developed in this thesis, which follows a segmentation-based approach for lossless compression.

The method proposed by McNerney (2000) differs from the ICSSM method developed in this research, as it is not intended for the archival of scientific grade data. The method of McNerney is a lossy approach, while the newly developed ICSSM method is lossless. It could be argued, that the method of McNerney is based on a simple model of astronomical stellar-field data, which is not discussed in the paper of McNerney (2000). The model of McNerney (2000) is based on the assumption that fainter objects and the background of astronomical images are not important and can be discarded before compression. But this is often not the case for scientific applications, where astronomers often have to know how the sky background exactly looked like before an event. Unlike the McNerney method, the data model of the new ICSSM method accounts for this. Developed for a different purpose, the McNerney method can be considered as a simplified version of the new method. Background levels – which solve the problem of non-planar backgrounds – and noise data are just discarded during compression and lost.

Comparing the newly developed ICSSM method with existing methods that incorporate lossless compression (see Table 3.1), it turns out that most of the methods providing a lossless mode rely on transforms as the central pre-processing step. But transform-based methods are computationally time consuming. Furthermore, transforms are excellent for lossy compression as many transform coefficients are zero or close to zero, and thus they can be efficiently compressed by entropy coders, but the noisiness of astronomical images impairs this effect.

Other existing methods, like the FITSIO methods, rely on encoders such as Rice-entropy encoder and Zip compression; the method of Sabbey (1999) uses a prediction-based approach. These methods, prediction-based decorrelation, dictionary-based, and entropy-based encoders, are examined in this investigation for encoding the data after the core pre-processing step, the segmentation. Although Lempel-Ziv-Welch based methods are not optimal to be applied directly to the 16- and 32-bit-per-pixel astronomical data (Section 2.2.1), this method is still considered for the data components generated through segmentation, because with the pre-processed data components data symbol set cardinality is limited. As Lempel-Ziv-Welch exploits the correlation among different source data symbols, at a certain degree, and not only the letter distribution, it is still an option to be tested.

In contrast to the Compfits method (Véran and Wright 1994), which only divides the image into different horizontal bit planes, the novel method presented in this thesis divides the image into distinct areas. As a result of the segmentation, the ICSSM methods remove unused bits in the background region prior to the final encoding, while Compfits keeps them and sends them to the external compression program. Compfits does not make use of the distinctiveness features in different image areas, thus it is not a segmentation-based method according to the definition used in this thesis.

Fractal methods – the last decorrelation method identified in the review of decorrelation methods in Section 2.2.3 – have not been used, to the knowledge of the author, for the compression of astronomical image data. Although at the first glance the main required property for fractal compression – self similarity – may be found in astronomical stellar field images[1], fractal methods are not examined here when developing the new method.

Carpentieri et al. (2000) argue that an image compression method should be tightly coupled to the data properties. The most distinguishing data property of stellar field images is probably the "noisy-background with bright-objects" property. Stellar-field images feature a noisy-background with bright-objects in the foreground; these images consist of noise-like regions with low correlation among adjacent pixels (Boussalis et al. 2004), sprinkled with spot-like bright and vague points. To the knowledge of the author, this special nature of stellar-field images has not been exploited by other researchers for segmentation-based lossless compression. The compression algorithms developed in this thesis differ from other existing lossless algorithms by using a decomposition of the image into distinct data areas as a core processing step, before applying other pre-processing and decorrelation methods.

5.1.4 Algorithm Outline

Figure 5.3 shows a data flow for the ICSSM 1 algorithm. The different steps (data decorrelation and encoding) which were used to classify compression algorithms in Section 2.2, are also included in the ICSSM 1 algorithm. First, segmentation is applied to obtain an image representation which would allow encoding into a smaller size. This process is followed by different encoders which apply efficient encoding to the different data components.

1. **Tile generation:** The whole image is split into equal-size square tiles of pixels using a user-chosen size parameter.

2. **Segmentation:** Each image tile is segmented into the regions of stellar objects and the background region, which is represented by a constant

[1] The self similarity within stellar field images is probably lower than one would expect at the first glance. The point-spread function differs depending on the position of the stellar object (Howell 2000). Therefore, to obtain a lossless representation of the image, with every "stellar object dot" a residual image would have to be stored after compression using fractal methods.

level $b(\mathbf{x}) = b_l$ across the tile. The segmentation[2] is based on the histogram of the tile. Assuming that the tile size (in pixels) is typically larger than the size of the bright objects, the background level b_l is approximated by the mode of the tile histogram. Under the assumption that the background noise $n(\mathbf{x})$ has a zero-mean Gaussian distribution, the noise amplitude N_l, for a tile is defined as the difference between the lowest intensity value and the histogram mode.

$$N_l = b_l - min(i(\mathbf{x})) \text{ where } \mathbf{x} \in X_l \tag{5.1}$$

X_l represents the set of all pixel positions in a tile.

Segmentation into two region types uses a threshold set to the upper range limit for the combined background and noise components, $n_{upp} = b_l + N_l$. Pixels with intensity value $i(\mathbf{x}) \geq n_{upp}$ are assumed to belong to the regions of stellar objects, the other pixels are assumed to belong to the background region. A binary image is generated to serve as region labelling and demarcation mask. The mask called region demarcation information in the following is used to indicate which of the two regions each pixel belongs to.

3. **Encoding:** The different image components are encoded as described below.

   (a) Background noise $n(\mathbf{x})$: The pixel intensity values of the background region are encoded relative to the estimated background level in order to reduce the required dynamic range. The background noise data is stored in a single data stream. Huffman encoding is adopted for coding the background noise.

   (b) Background level b_l: The assumption is that this value is encoded without compaction, using the full dynamic range of intensity values in the original image. As the background level is modelled as constant for the whole tile, only one background level has to be stored for each tile. Due to its small size, the background level data is encoded without further compaction using the same dynamic range as the original pixel data.

[2] The validity of this segmentation step is discussed in Section 5.1.5.

(c) Region-of-stellar-objects $s(\mathbf{x})$: This component is encoded using Huffman coding, like the background noise. Again, as with the background noise data, here the aim is to use a method which only exploits the statistical distribution of pixel values but does not apply further processing for redundancy reduction.

(d) The region demarcation information is encoded using a binary image without applying further compression. With this encoding, the size of the side information, which has to be stored due to the segmentation, is readily identifiable.

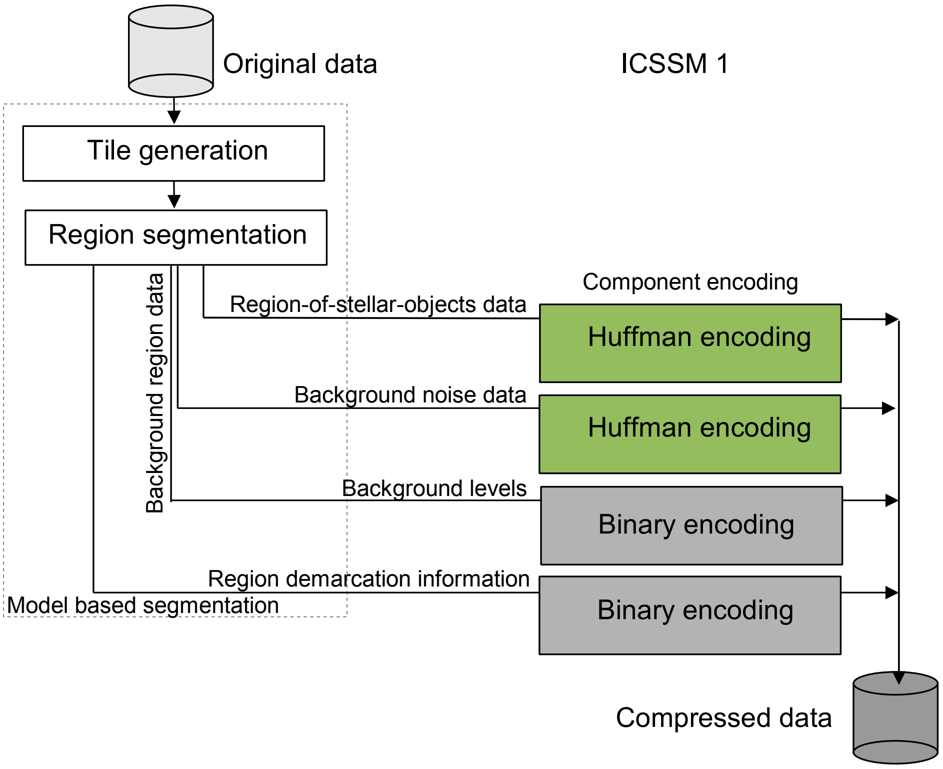


Figure 5.3: Overview of the ICSSM 1 algorithm.

For decoding the data, first the Huffman encoded of background noise data and the region-of-stellar-objects data have to be decompressed. Then the segmentation has to be inverted; according to the region demarcation information, the original value for each pixel can be restored. The data has to be processed in the same order as during compression; then for each pixel assigned to the region-of-stellar-objects, the intensity value from the corresponding data stream can be put back into place. For each background region value, the original intensity value can be recalculated by summing up the background level and the background noise data.

5.1.5 Validity of the Segmentation Step

To ensure the losslessness and efficiency of the segmentation step, possible cases which may affect it are analysed in this section. Four cases in which the segmentation may be affected are identified and discussed. The first case discussed analyses whether an object which straddles on a tile border affects the segmentation. It is called the "part of object case". In the second case, which is discussed with reference to its impact on losslessness and compression performance, a tile contains a very high amount of bright foreground pixels. This case is called the "large object case".

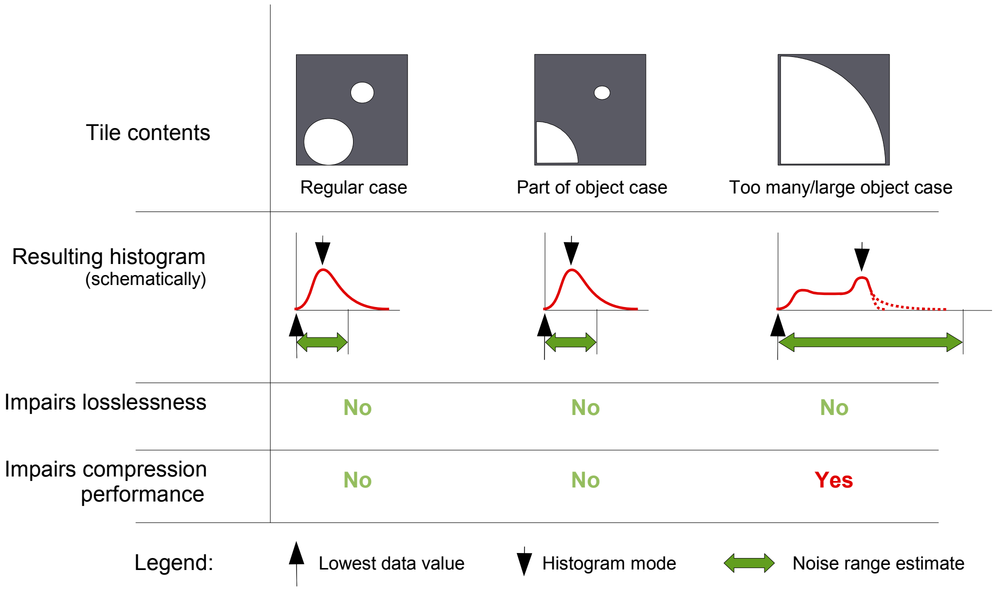


Figure 5.4: Schematic presentation of special cases which might occur during the image segmentation step: resulting histogram and its impact on compression performance and losslessness.

Figure 5.4 shows the regular case, the "part of object case" and the "large object case", in which segmentation may be affected. A third difficulty which may arise with the segmentation step, is the decreasing expressiveness of the histogram with decreasing tile size t. This issue and the fourth case, dark outlier pixels, are discussed last in this section.

With an image histogram, which shows the distribution of the single greyscale intensity values within an image tile, it is irrelevant how the stellar objects are distributed spatially on the tile. Only their contribution to the image area is important. Also, it is irrelevant for the histogram whether a tile contains a complete foreground object or not. The important feature that is used to distinguish the areas is the intensity contrast. The posture of an object on a tile border does not affect the image histogram, thus it has no effect on segmentation. Hence, the first case neither impairs the losslessness of compression nor the compression performance.

The “large object” case, in which the image tile area is dominated by bright objects, impacts the image histogram. As huge objects are typically brighter than small objects, the histogram may have an exaggerated dynamic range. This is not directly visible in the schematic black-and-white tile contents drawing in Figure 5.4. The resulting histogram below the tile contains both possible cases. The exaggerated and the unexaggerated dynamic range case are shown in dotted lines. The possibly exaggerated dynamic range in the histogram is not important for the noise range determination, as the noise range determination is based on the mode and the lowest intensity value. But with a sufficiently large amount of bright objects it is possible that the histogram mode does not represent a valid image background level estimate, although this should rarely be the case, as discussed in Chapter 4.

Even in this case, as the noise range is determined as the difference between the mode and the lowest intensity value of the image, a very high noise range is predicted for this image tile, typically above eight bits. The segmentation still remains lossless. A pixel is either assigned to the background area, encoded as difference to the (non-optimally chosen) background level, or it is encoded as region-of-stellar-objects value with its absolute value. A non-optimal segmentation may therefore occur in the case where the bright area dominates the background area within a tile. While this could affect the performance (compression ratio) of the algorithm, it would not affect its losslessness. As large noise values within the noise data stream will impair the statistics of the noise data stream and affect its compressibility, with ICSSM 2 and ICSSM 3 an upper limit for the noise range is applied before segmentation (Section 5.2).

The diminishing expressiveness of the histogram with decreasing tile size t is the third problem that might arise during segmentation. Determining the noise-amplitude and background estimate correctly becomes more and more difficult when the tile size t decreases to very low values. This occurs for two main reasons. Firstly, the significance of the intensity histogram diminishes drastically with a smaller tile size parameter. While 100 pixels contribute to the histogram in case of tile size t set to 10, only a quarter of the pixels contribute to it if t is set to 5. The number of pixels that contribute to the intensity histogram decreases as a quadratic function of the tile size.

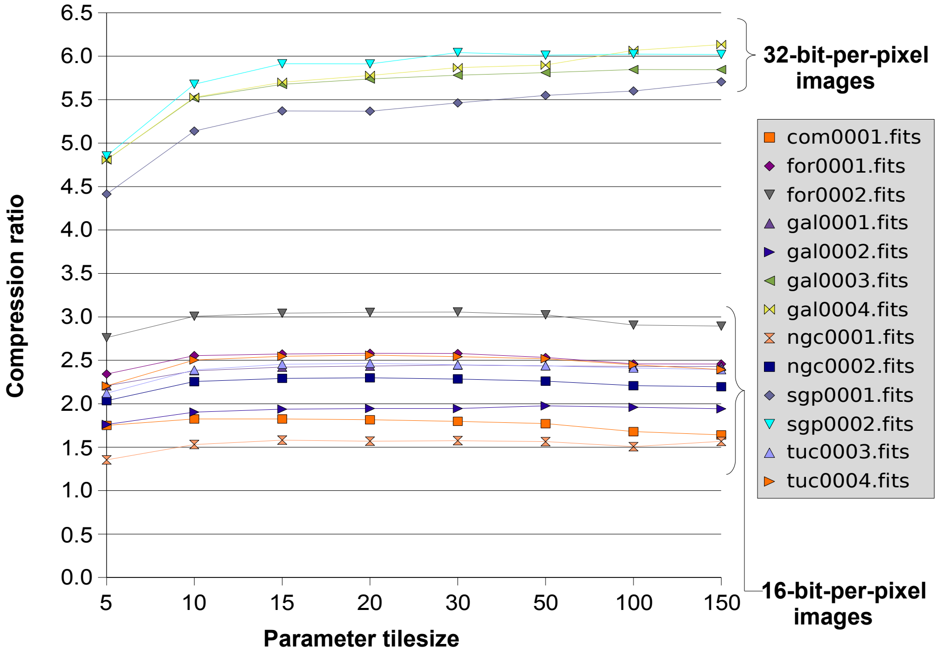


Figure 5.5: Influence of tile size on compression ratio. On the four 32-bit-per-pixel images, the compression ratio is significantly higher than on the 16-bit-per-pixel images.

This effect is directly related to the "large object" case. With the smaller image area that contributes to the histogram, bright stellar objects may affect the histogram shape more severely. If the tile size is not significantly larger than the largest object on that tile, the determination based on histogram-mode may find an excessively large noise amplitude.

Too large a noise amplitude in some image areas may lead to sequences of large “non-noise” values in the background noise data. In the case of ICSSM 1, that reduces the performance of the Huffman encoder; it increases the data rate for the noise data stream. Due to the growing Huffman tree, the increase of the codeword size does affect the size of all data values during encoding. It does not only enlarge the codeword size of the small portion of large “non-noise” values.

A solution to this problem may be to re-examine the plausibility of the noise-amplitude and background level estimates in the case of small tile sizes. This effect, in combination with the simple encoding method for the background model and the flatness of some images, may be responsible for the still increasing algorithm performance with tile sizes t larger than 100 pixels on several of the test images (Figure 5.5) with ICSSM 1.

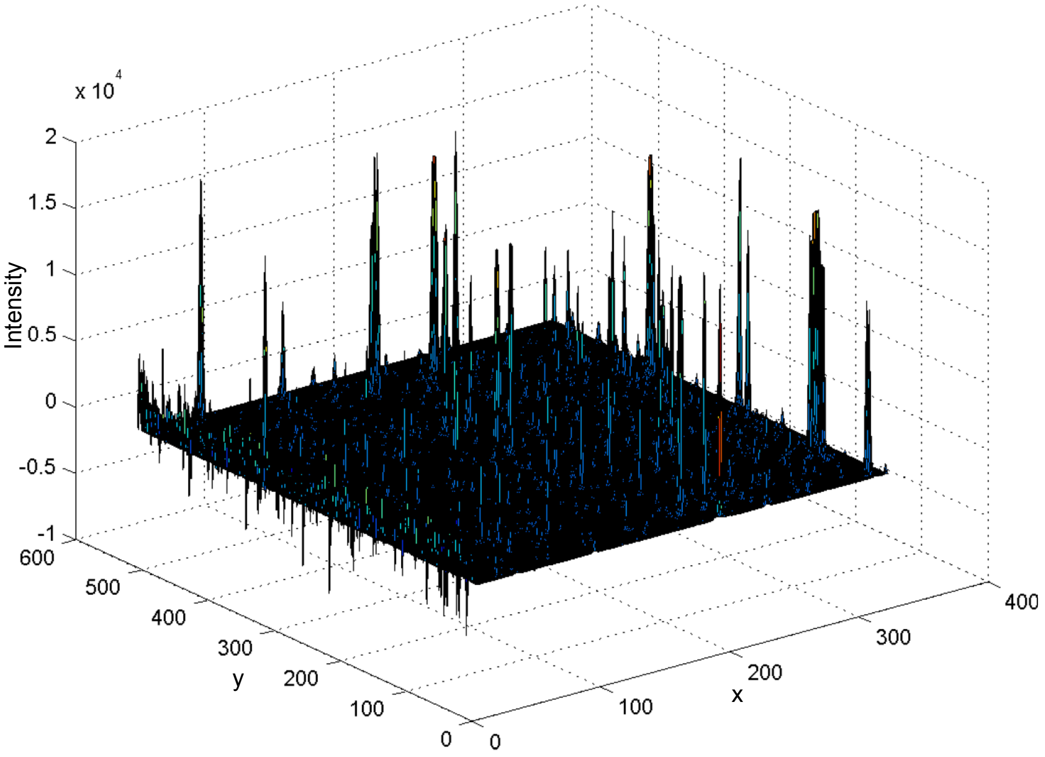


Figure 5.6: Three-dimensional plot of the image “ngc0002”, where dark outlier pixels can be observed in the left part of the image. Short, downward oriented spikes are visible at the left front border.

Finally, a fourth effect may also lead to excessive noise estimates. The method used with ICSSM to estimate the noise range N_{ij} is only optimal for typical noise distributions which are Gaussian shaped, centred around the mode and which do not contain low outlier intensity values.

The rule for estimating N_{ij} can lead to high estimated noise ranges in the case of dark outlier-pixels present in the image (Figure 5.6). In the worst case, one outlier pixel can lead to an encoding overhead of several bits for each pixel in the whole tile. More optimally, a few outlier pixels could be treated as region-of-stellar-objects pixels. One possibility to enhance the quality of the noise amplitude and background level estimations would be to use more distribution parameters for their detection. But any distribution parameter may become unreliable for small tiles. Alternatively, an upper limit for the noise range can be applied before segmentation to prevent this effect completely. This approach is chosen with ICSSM 2 and ICSSM 3.

5.1.6 Summary

This section presented ICSSM 1, the first version of the segmentation and region-adaptive compression algorithms. The algorithm decomposes the data of a stellar field, based on the statistical distribution of the pixel intensity into two distinct areas, regions of stellar objects and background regions, based on information carried by the intensity contrast (Figure 5.7). The segmentation of the data into different areas and signal components allows a region- and component-adaptive encoding of the data. Simple encoding methods which allow fast computation, although there is still potential for better compaction, are applied in ICSSM 1. The background noise component and the region-of-stellar-objects data are encoded independently using Huffman coding; background level data and the region demarcation information are encoded directly as binary information.

Finally, special cases that might arise during segmentation and their impact on losslessness and the size-efficiency of the method were analysed.

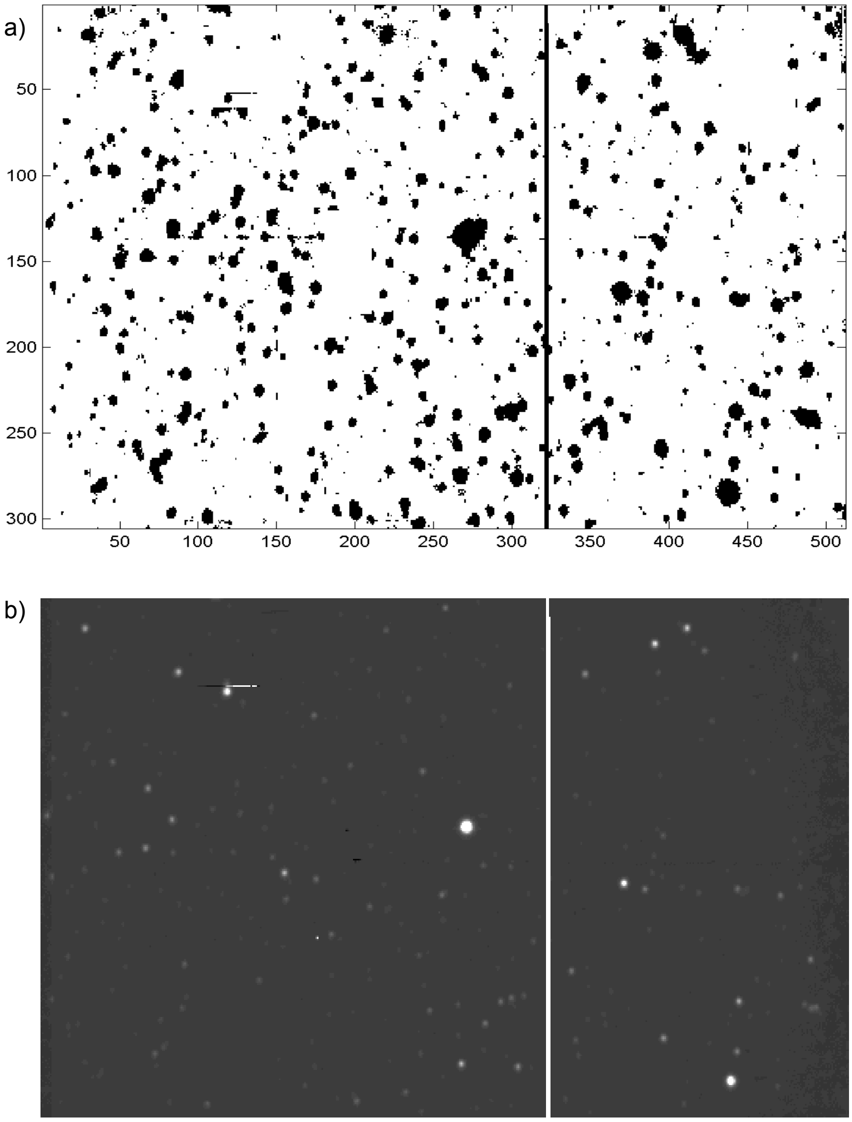


Figure 5.7: (a) Region demarcation information sample of image "for0001" segmented with a tilesize of 30. The black region is the region-of-stellar-objects. (b) Original "for0001" image: The white vertical line in the original image is due to defective CCD columns.

5.2 Compression Algorithm Using the Same Encoder for all Image Data Components (ICSSM 2)

The aim of the ICSSM 2 algorithm is to obtain another trade-off between simplicity of the data compression algorithm and coding efficiency. With ICSSM 2, only one encoder is applied to the data generated through segmentation, to keep the encoding step as simple as possible. Although the same encoder is used for all image components, the idea of region-adaptive bit-allocation is still present in this approach. This is because prior to the final encoding, region-of-stellar-objects pixels are encoded verbatim, whereas noise values are encoded relative to the estimated background.

Section 5.2.1 discusses details of the segmentation step applied with the ICSSM 2 algorithm. A range of methods are tested for the final encoding. These methods implement state-of-the art methods for general-purpose compression of binary data, such as the LZ77 variant called Deflation, a method based on the Burrows-Wheeler Transform, and an improved version of the prediction by partial matching compression algorithm. Here, general-purpose compression refers to compression applied to an image, without reference to its pictorial content. The file is processed, by the compression software, as a mere symbol stream. Implementations of the data encoding methods listed above are available as standard encoding tools such as Zip, Bzip2 or 7-Zip.

They are discussed and their compression performance is analysed using sample image data in Section 5.2.2.

Finally, Section 5.2.3 outlines the method chosen with ICSSM 2.

5.2.1 Segmentation Step

The segmentation step used with ICSSM 1 is adopted with minor changes, into ICSSM 2. As discussed in Section 5.1.5, it is possible that a large noise amplitude is produced by the histogram-based detection of image data components, if the tile size is not significantly larger than the largest object in that tile. While this effect does not affect the losslessness of the whole algorithm, it may impair the encoding performance. This detoriation could be especially severe in the case where general-purpose file compression tools are applied. These tools typically work byte-wise on the data. Hence, they can exploit the data statistics in the case where each noise value[3] is encoded within a single byte in the input stream of the external tool.

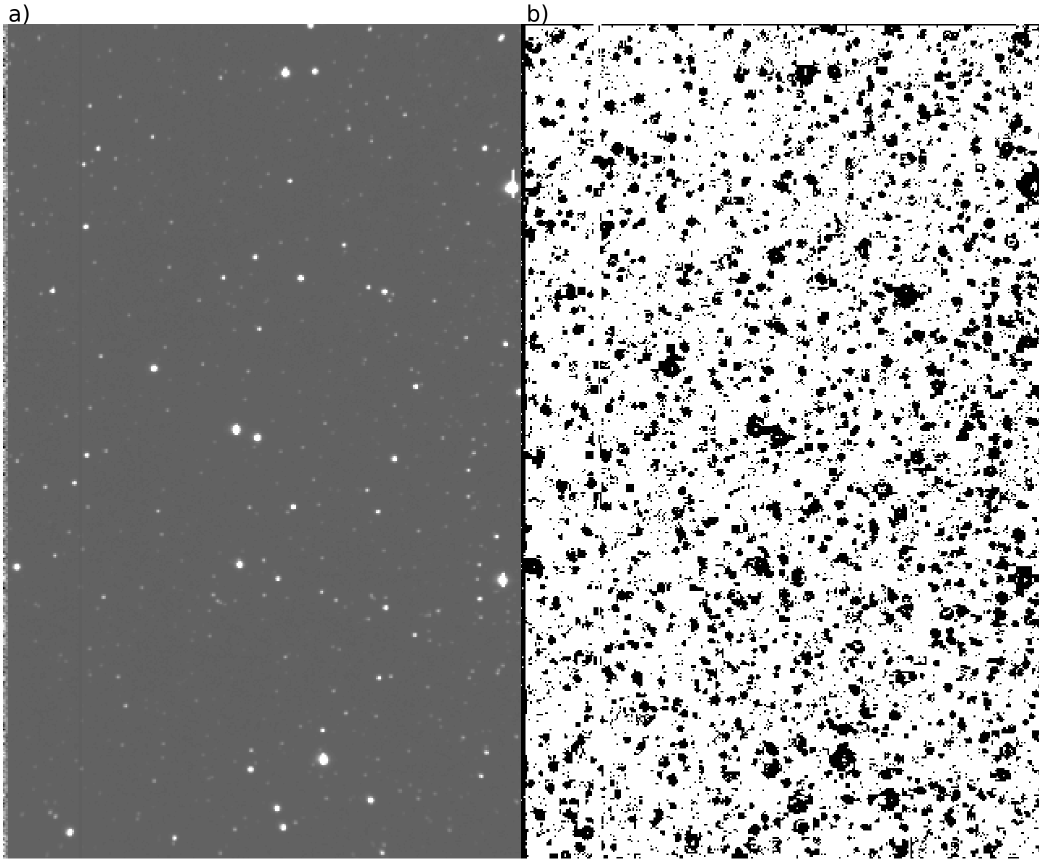


Figure 5.8: (a) Original image. (b) Region demarcation information after segmentation with ICSSM 2. The black region is the region-of-stellar-objects.

If the noise-range prediction indicates a range requiring more than eight bits, the data is no longer aligned to byte borders. This may – by changing the data statistics – impair the performance of the general-purpose compression tool.

ICSSM 2 reduces the problem of wrongly estimated noise values by applying an upper limit for the possible noise estimate. As discussed in Section 4.2, the noise of astronomical images typically influences only the last few bits. Because the general-purpose data compression tools work byte-wise on

[3] The noise value is typically much smaller than eight bits, see Chapter 4.

the data, the upper bound allowed for the noise estimate is chosen to be eight bits. With the fixed upper limit on the noise range the areas that require a larger dynamic range than one byte are treated as regions of stellar objects.

Figure 5.8 shows the region demarcation information after segmentation with a tile size parameter of five, which is the lowest tile size that was tested with the ICSSM algorithms. Inspecting the region demarcation information shown in Figure 5.8 shows that in this sample data segmentation problems are quite rare, even at low tile sizes of only five pixels.

5.2.2 Encoder Choice

ICSSM 2 uses state-of-the art methods for general-purpose compression of binary data to encode all the data streams generated through image segmentation. Efficient implementations of a range of general-purpose methods exist and are tested for ICSSM 2. In the following, tools for general-purpose compression of binary data are referred to as external file compression tools.

Experiments with the following general-purpose file compression tools were conducted to determine whether there are differences in the size-efficiency and to find out which method gives the most promising results for all four data components (region-of-stellar-objects data, background noise data, background levels and the region demarcation information).

- **Zip:** Zip is an implementation of the LZ77 variant called Deflation[4].
- **Bzip2:** Bzip2 is an implementation of the Burrows-Wheeler Transform based method[5].
- **7zip:** 7zip supports several data compression methods. It is used to test an improved version of the 1984 prediction by partial matching (PPM) compression algorithm[6].

A test image “gal0004” was segmented into its data components, which were exported into individual binary files to determine which of the methods most efficiently compacts all of them into a single archive. The results are given in Table 5.1.

[4] Source code and binaries are available from: http://www.gzip.org/ (01.02.2007).

[5] Source code and binaries are available from: http://www.bzip.org/ (01.02.2007).

[6] Source code and binaries are available from: http://www.7-zip.org/ (01.02.2007).

The compression ratio of all three tested methods lies in the range 2.02 to 2.15. 7zip and Bzip2 lead to slightly higher compression rates than Zip on the sample data. Although the results from a single sample image can not be generalized, only a small performance difference was found for the three methods. Zip compression is chosen for ICSSM 2 given its popularity and given the small performance gain obtained for the less common methods.

Table 5.1: Compression ratios of general-purpose methods for all data components generated through segmentation (Image "gal0004" with tile size parameter set to 150 pixels). The size of the uncompressed image is 1144320 byte.

Compr. method	Data size (byte)	Compr. ratio
Zip	566359	2.02
Bzip2	539959	2.11
7zip	530271	2.15

5.2.3 Algorithm Outline

As with ICSSM 1, a similar diagram for the data flow is given for ICSSM 2 in Figure 5.9. The segmentation step exports all data components in binary form as intermediate files to the encoder.

Zip, an implementation of the LZ77 variant called Deflation, is used in ICSSM 2 to further compact the data into a single output file which contains all data components.

As only the data encoding step and the determination of the noise range was changed compared to ICSSM 1, the decoding procedure is similar to the one in ICSSM 1. First, the encoding of the data components has to be reversed by "unzipping" the whole data archive. Then, the segmentation step can be reversed as with ICSSM 1. The noise range limit has no influence on the encoding and decoding procedure, it only changes the assignment of pixels to the image areas labelled as region-of-stellar-objects and background region.

5.2.4 Summary

With ICSSM 2, the same encoder is applied to the data generated through segmentation in order to keep the encoding step as simple as possible. Following the aim to derive a version of ICSSM which couples simplicity – by requiring only a single encoder – with size efficiency, choices for possible encoders were analysed. As the results obtained with the different methods differ only a little (Table 5.1), the most widely used and common method Zip is chosen for ICSSM 2 data component encoding. The implications of the segmentation step were analysed and the step was enhanced by introducing an upper limit for the possible noise range. In short, the ICSSM 2 algorithm is an ICSSM variant that combines a general-purpose encoding method with the segmentation step to achieve a straightforward and size-efficient algorithm.

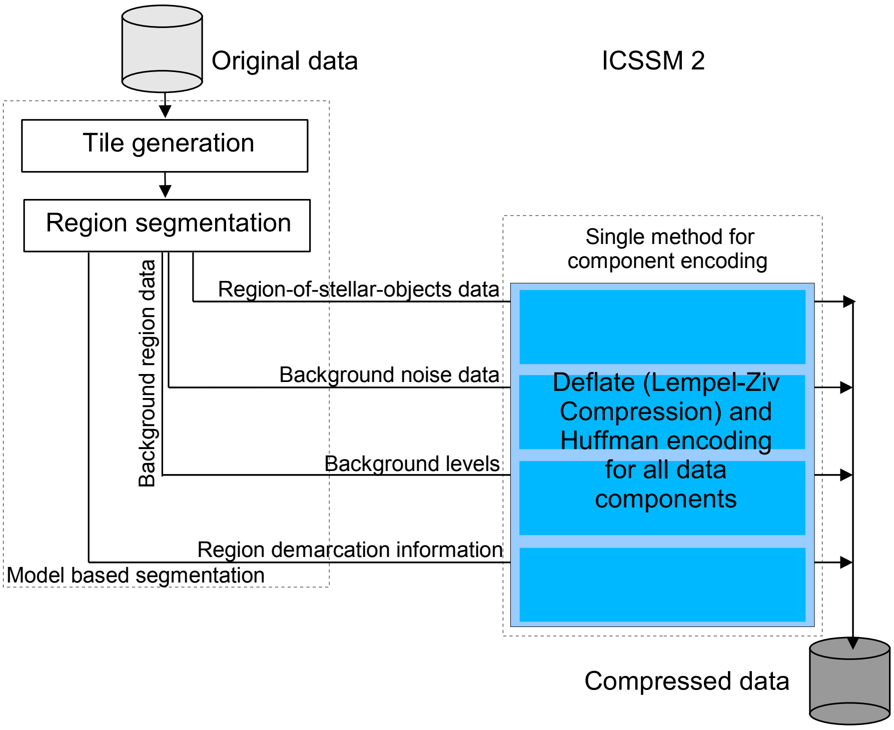

Original data
ICSSM 2
Tile generation
Region segmentation
Background region data
Region-of-stellar-objects data
Background noise data
Background levels
Region demarcation information
Model based segmentation
Single method for component encoding
Deflate (Lempel-Ziv Compression) and Huffman encoding for all data components
Compressed data

Figure 5.9: Overview of the ICSSM 2 algorithm.

5.3 Compression Algorithm Using Encoders Best Matched to Each Image Data Component (ICSSM 3)

Finally, data compaction methods best matched to the properties of each data component are applied in ICSSM 3. Possibilities for enhancing the compression ratio of the ICSSM algorithms are identified and investigated to determine the most promising data encoding approaches. With ICSSM 3, the focus lies on maximising the compression ratio of the individual data components.

Individually and optimally chosen encoding and decorrelation methods for the different data components are evaluated to enhance the compression ratio of the segmentation-based and region-adaptive compression algorithm. Using pre-segmented sample data, possible candidates for the encoding of each data component are evaluated: encoding of background noise data (Section 5.3.1), encoding of background levels (Section 5.3.2), encoding of region-of-stellar-objects data (Section 5.3.3), and finally encoding of region demarcation information (Section 5.3.4). Suitable candidates for compression ratio enhancements are identified, and using example data for each of the data components, optimized solutions for their further compaction are determined. A set of data samples (four images) was chosen for the experiments to identify optimal solutions. Experiments were conducted on all data samples from the test image set since the objective was to adapt the algorithm to stellar fields in general and not to include too many properties specific for a given test set.

Algorithm details are presented in Sections 5.3.5. Like the other ICSSM versions, the size-efficiency of the ICSSM 3 algorithm is evaluated and discussed in Chapter 6.

5.3.1 Encoding of the Background Noise

Possible methods to reduce the size of the background noise data component are analysed in this section to investigate the potential for a further size-reduction of this data component.

The properties of the background noise data are analysed in the first part of this section. It is shown that this signal component resembles Gaussian distributed white noise, as expected. Then, based on the work of Romeo et al. (1999), a theoretical limit for the compression of the background noise data – under the assumption it is pure white noise – is established. It is found that the entropy of the real background noise data is only slightly lower than that predicted for white noise.

Finally, in practical experiments, an optimal encoder is determined using real sample background noise data. Based on results of the practical experiments, the encoder is chosen for encoding the background data with the ICSSM 3 algorithm.

Background Noise Properties

If segmentation worked perfectly, the background data should be a noise data stream which is Gaussian-like distributed, stationary[7], and centred around zero. Furthermore, the data stream values should be uncorrelated and resemble white noise (Section 5.3.3). Therefore, the background noise data should have a flat spectrum. Taking a sample noise data stream, first of all, the distribution of the data values was examined. The background noise data from image "gal0004" was chosen due to its size and the image properties: It is a rather typical example of a stellar-field image. It was segmented using a tile-size parameter t of 50. This is neither a very small value nor does it lie at the upper end of the examined tile-size range.

Figure 5.10 shows that the probability density function of the background noise data resembles a normal distribution. As the data is centred on zero, the offset components from the data were removed successfully.

While the histogram given in Figure 5.10 shows that the noise data values resemble Gaussian noise, it does not show anything about whether the noise data is white noise. The fast Fourier transform function in Matlab was used for examining the frequency spectrum of the noise. The resulting spectrum is shown in Figure 5.11. Although some of the very low frequencies have a slightly increased intensity, the overall spectrum is quite flat, as is expected for white noise. Therefore, it can be concluded that the segmentation step almost successfully separated background noise data and background level data. With the minor limitations discussed, it generates a Gaussian white noise data stream.

[7] The probability distribution parameters remain fixed at different positions in the data stream.

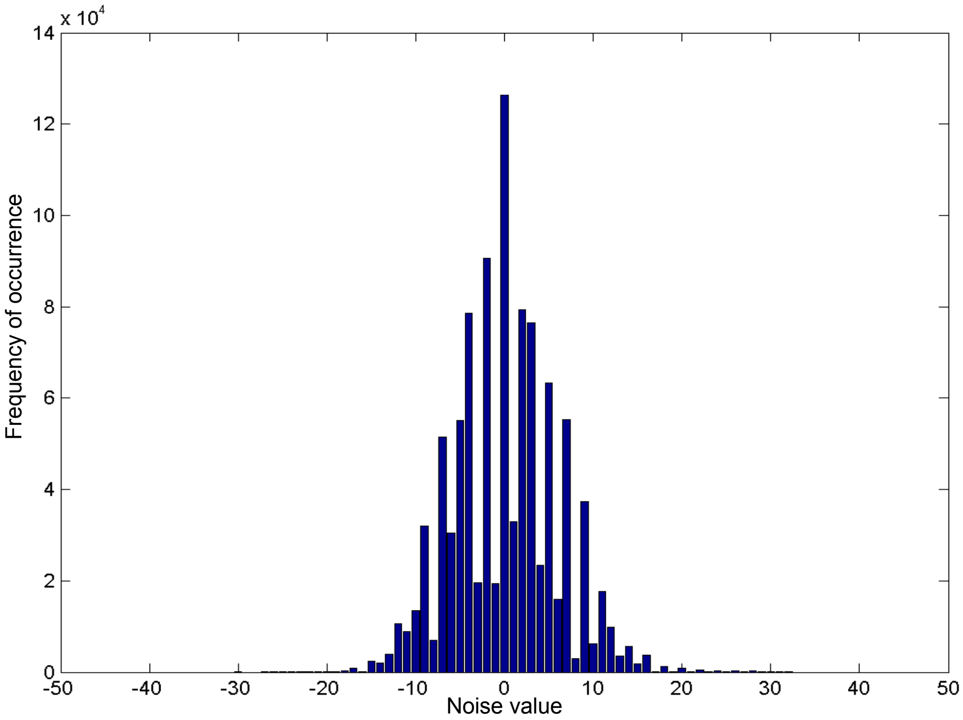


Figure 5.10: Gaussian-like distribution of data values within a noise sample stream of image “gal0004” segmented using a tile-size parameter t of 50.

Theoretical Limit for Data Compression

A theoretical approach which discusses the information content of uniformly discretised Gaussian noise is presented in (Romeo et al. 1999). Romeo et al. (1999) address different types of noise, including white noise. The possible compression ratio and the entropy are shown to depend on two factors: the shape of the noise power spectrum and the discretisation of the noise. If noise is discretised to a high resolution (as compared to its variance), the resulting distribution of amplitude of noise samples approaches a uniform distribution (Romeo et al. 1999). They find that the compression ratio decreases logarithmically with the amplitude of the frequency spectrum $P(f)$ of the noise. For zero-mean Gaussian white noise, Romeo et al. (1999) find that the entropy is given by Equation 5.2.

$$h = \log_2 \left[\sqrt{2\pi e} \cdot \frac{\sigma}{\Delta n} \right] + O\left(\frac{2\pi\sigma^2}{(\Delta n)^2} \cdot e^{-\frac{2\pi^2 \Delta^2}{(\Delta n)^2}} \right) \tag{5.2}$$

The second part of the equation, given in big O notation, provides an upper bound for the last addend of the entropy equation. The function was approximated with a series by Romeo et al. (1999).

The entropy, as given in Equation 5.2, depends on the quantisation step Δn and the variance of the probability density σ. The dimensionless quotient $\frac{\Delta n}{\sigma} = \lambda$ can be considered as a discretisation parameter. With the values $\sigma = 7$ and $\Delta n = 1$, extracted manually from Figure 5.10 Equation 5.2 predicts an entropy of 4.85 bits per data value for our noise sample data stream. This would result in a compression ratio of 1.65 compared to the raw encoding, which uses one byte per data value.

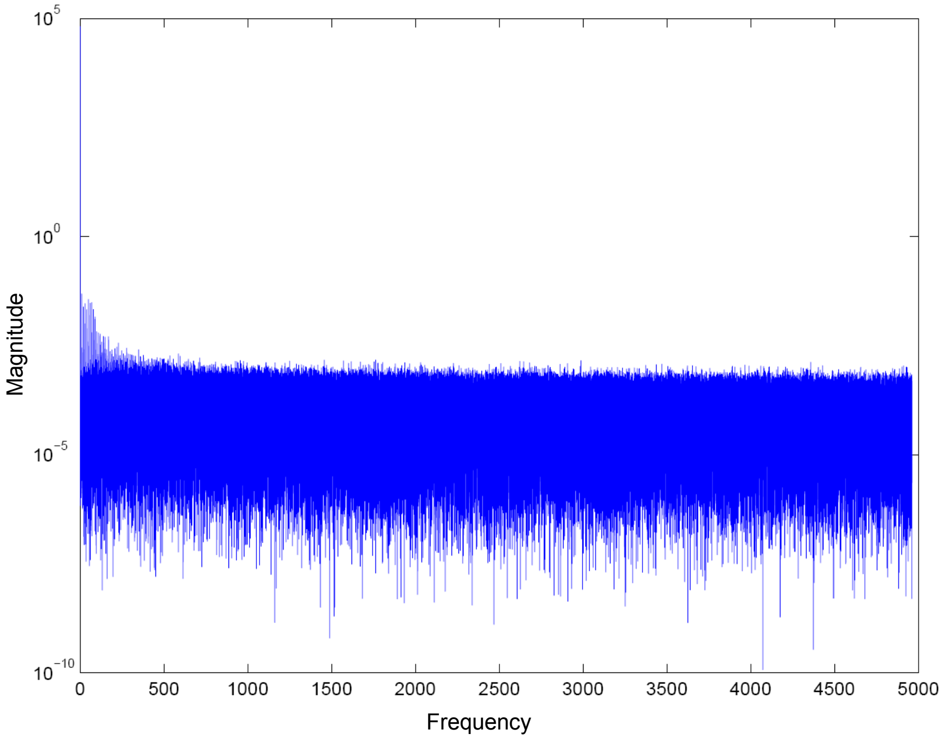


Figure 5.11: Frequency spectrum of the noise data sample from the image "gal0004".

As white noise data is uncorrelated, the only data property a compression algorithm can exploit in theory is the distribution of data values. Therefore, an optimal entropy encoder (with certain restrictions discussed in Section 2.2.1), such as the Huffman encoder, should lead to the highest possible compression ratios. That the segmentation step works well can also be seen from the fact that the first-order entropy calculated from the data (4.35) is only slightly lower than the theoretical limit for zero-mean Gaussian white noise (4.85) estimated using Equation 5.2.

As no correlation is found among the data values, data decorrelation methods such as differential encoding, or prediction-based methods, may not lead to a further size reduction of this data.

In order to verify that differential encoding of the sample data of image “gal0004” does not enhance the compression ratio, the sample noise data was encoded using differential encoding and the first-order entropy was measured. The first-order data entropy increases from 4.35 bits to 4.86 bits per data value.

Practical Experiments to Assess the Compression of Background Noise

Despite the theoretical results presented in the last section, which indicate that no significant further size reduction is possible compared to entropy encoding methods, still a practical evaluation was performed to determine how other encoders behave in practice on the background noise data.

In addition to two different versions of arithmetic coding, a set of general-purpose compression methods including Zip compression and a Burrows-Wheeler Transform based approach was tested. Compression ratios of the four tested encoders on the background noise data of the images “com0001”, “for0002”, “gal0001”, and “gal0004” were measured.

The image test set that was chosen includes images of different characteristics in order to prevent optimising the algorithm to a set of properties which is too specific. All images from the test image set are shown in Appendix C along with their pixel intensity histograms. The image “for0002” is a sample of a rather typical crowded star-field, while the image “com0001” was chosen because it includes some large scale background intensity variations. Although “gal0001” shows the typical point-like source on a dark background, it is a galaxy cluster image. Finally, image “gal0004” was chosen to include an example with a large image area compared to other test images. Furthermore, it was originally encoded with the higher 32-bit-per-pixel bit rate. For each of the images, the background noise data after segmentation with a tile size parameter t of 20 and 50 pixels was examined.

The entropy coders tested are different versions of a range-coder, which is a time-efficient arithmetic encoder implementation. A version based on order 0 probability modelling ("RangeCoder 0") and a version based on order 1 adaptive probability modelling ("RangeCoder 1") were tested on the background noise data of the image set described in the last section. The implementation of arithmetic coding is available from (Schindler 1999) under the conditions of the GNU general public license. Initial results showing the performance of this method were presented in (Schindler 1998). Additionally, Zip compression and the Burrows-Wheeler Transform based method, implemented in Bzip2, were evaluated as well. Results are summarised in Table 5.2.

Table 5.2: Compression ratios of four tested encoders on the background noise data of the images "com0001", "for0002", "gal0001", and "gal0004". Fore each of the images, the background noise data after segmentation with a tile size parameter t of 20 and 50 pixels was examined. Ar.(0) stands for the arithmetic coder order zero, Ar.(1) for arithmetic coder order one.

	Encoder				
Image	Zip	Bzip2	Ar. (0)	Ar. (1)	$\bar{c}$
com0001 (t=20)	**1.08**	1.05	**1.08**	1.04	1.06
com0001 (t=50)	1.06	1.05	**1.07**	1.03	1.05
for0002 (t=20)	2.14	2.37	2.13	**2.53**	2.29
for0002 (t=50)	2.09	2.34	2.34	**2.47**	2.31
gal0001 (t=20)	1.43	1.43	**1.49**	1.46	1.45
gal0001 (t=50)	1.43	1.43	**1.48**	1.45	1.45
gal0004 (t=20)	1.68	1.78	1.79	**1.87**	1.78
gal0004 (t=50)	1.72	1.82	1.85	**1.91**	1.83
$\bar{c}$	1.58	1.66	1.65	1.72	

The first result of the practical evaluation listed in Table 5.2, is that the theoretical results determined for an entropy encoder in Section 5.3.1 were very close, but still slightly lower than the experimentally determined results for arithmetic entropy encoder (order 0). The theoretical results determined with the formula of Romeo et al. (1999) suggested a slightly lower compression ratio of 1.65 than those experimentally determined for the arithmetic entropy (order 0) encoder (1.85).

But, as expected after the examination of the data correlation, entropy encoders proved to be the best choice for encoding the noise data. The arithmetic coder led in all tested cases to the best result values although in one case Zip reached the same compression ratio value. Both, the order 0 version and the order 1 version, led to the best result in four of the eight tested cases. The highest average compression ratio on the tested images produced by the arithmetic order 1 encoder was followed by Bzip2 compression and arithmetic order 0 encoder. Accordingly, order 1 arithmetic compression is used with ICSSM 3. Zip compression led to the lowest average compression ratio, a compression ratio enhancement can thus be expected for ICSSM 3 compared to ICSSM 2 which uses Zip as the encoder for all image data components.

5.3.2 Encoding of the Background Level

Although the background level data often contributes only little to the size of the compressed data as there are only few background levels to store, this section analyses options for a size-efficient compression of this component.

First, properties of the background level data are analysed. Then, the possibility to exploit possible redundancy in this data component by using a simple prediction is tested and rejected using an illustrative example. Finally, four promising encoders are evaluated to determine the best choice for the background level data.

A three-dimensional plot of the background levels is shown in Figure 5.12. The background level data is highly correlated across the image. Typically, the differences among neighbour background levels are very small. Only three background levels are apparent with a too high background level estimate, as discussed under the “too many/large object case” in Section 5.1.5. Depending on the tile size and the image size, the background level data component, which can be considered as a small smoothed version of the original image, reaches a size which makes its compression necessary.

The background level data can be considered as a discrete signal generated through a process. The signal $x[n]$ generated by an arbitrary process is predictable if the current signal value $x[n]$ can be determined from a linear combination of previous signal values $x[n-k]$ (Hunt 1998).

$$x[n] = \sum_{k=1}^{\infty} h_k x[n-k] \tag{5.3}$$

The question which is addressed here is whether the background level data can be successfully considered to consist of a predictable part plus an offset. Linear prediction decorrelates a signal by subtracting an estimate of the current data value from the actual value, leaving a whitened signal (Hunt 1998). The estimate of the current signal value thereby may consist of previously submitted data values $x[n-k]$ in the one-dimensional case, or already submitted data values from a two-dimensional surrounding (Hunt 1998).

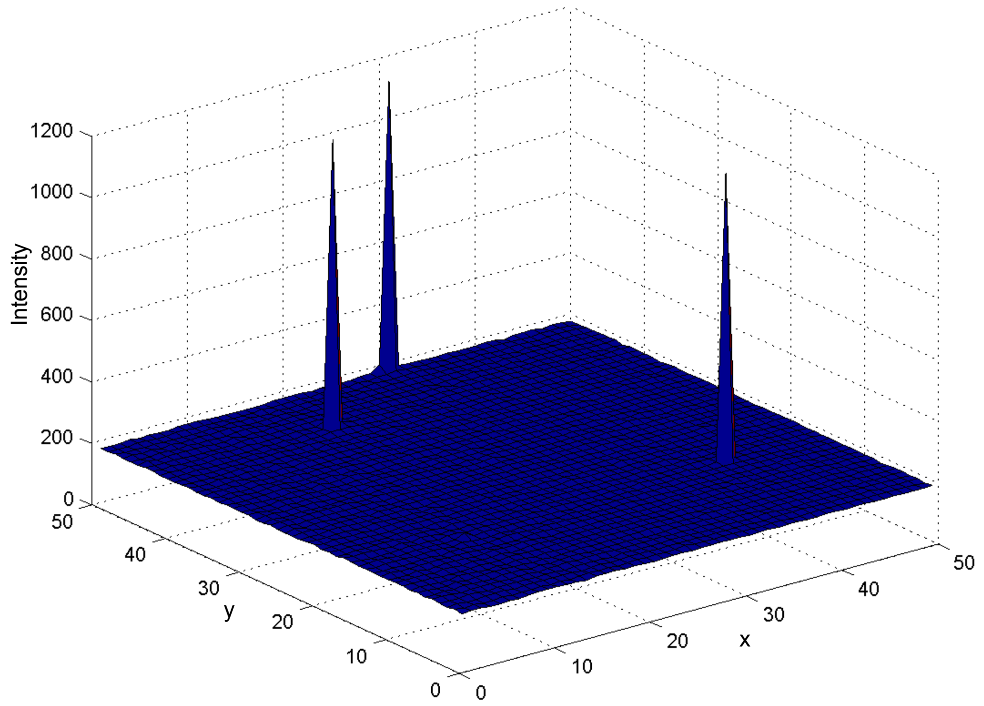


Figure 5.12: Three-dimensional plot of the estimated background levels for the image “gal0004” segmented using a tile size of 20 pixels.

Figure 5.12 suggests that there is a strong dependency among neighbour background levels. Therefore, the use of prediction-based methods to decorrelate the data, exploiting the dependency among neighbour background levels, is examined here. One possible solution to decorrelate the data is the application of a one-dimensional differential encoding, to exploit the dependency within the background level rows. The residual $r[n]$ is calculated as:

$$r[n] = x[n] - x[n-1] \tag{5.4}$$

If the differential encoding is applied row-wise, followed by differential encoding of the first column, all but one image pixel are differentially encoded.

But this simple method only exploits the dependency between successive pixels in a row, without choosing the best possible linear prediction from a larger neighbourhood. Figure 5.13 shows a residual distribution of the background levels from test image "gal0004" after linear prediction. This simple differential encoding is also applied by one of the filters used for the PNG file format (see also Section 2.2.3).

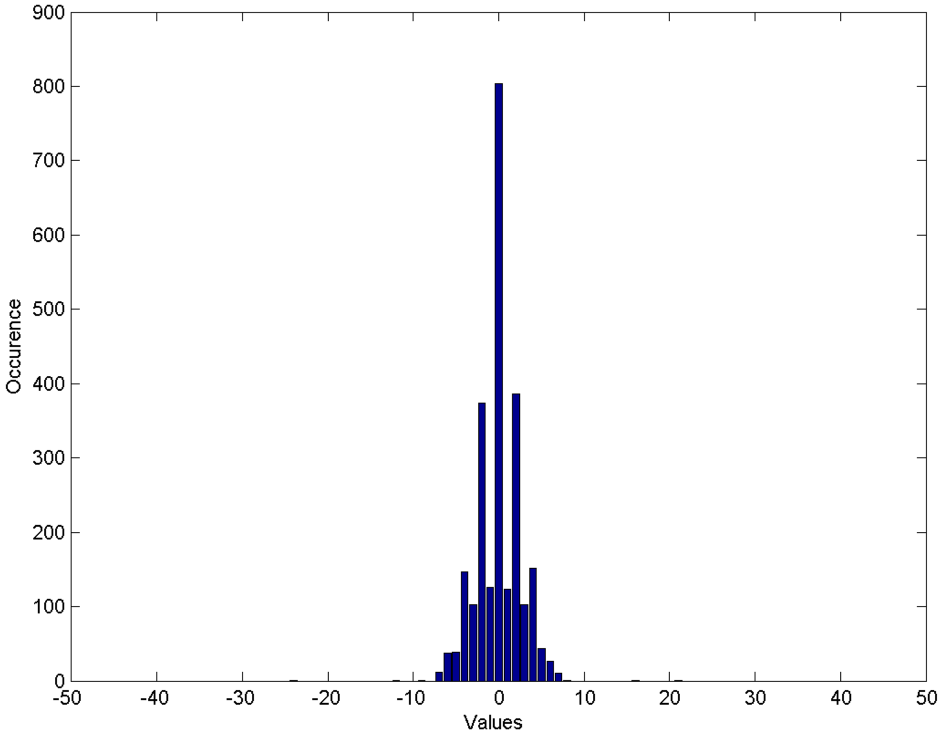


Figure 5.13: Distribution of the linear prediction residual from the background levels of test image "gal0004" segmented using a tile size of 20 pixels.

As an illustrative example, the first-order entropy of the raw background levels was measured and compared to the entropy of the data after applying difference encoding as described above.

The measurements of the first-order entropy of the raw background levels of image “gal0004” gave a value of 2.32 bits-per-pixel. The total number of different data values was increased through difference encoding of the background level data values. First-order entropy increased to 3.13 bits-per-pixel, with the application of difference encoding, despite of the estimated dependency among neighbour values.

The compression performance on the background level data stream is only important for smaller tiles as the contribution of the background levels to the whole file decreases rapidly with larger tiles. This effect can be derived mathematically. If an image with the size x by y pixels is divided into equally sized tiles of a tile size t, n tiles are generated.

$$n(t) = \left\lceil \frac{x}{t} \right\rceil \cdot \left\lceil \frac{y}{t} \right\rceil \tag{5.5}$$

The number of tiles n(t) defined in Equation 5.5 is a whole number. The special brackets used in Equation 5.5 stand for the ceiling function to account for cases where tile size does not divide the image size without a fractional part.

n(t) decreases extremely fast asymptotically to zero with a growing tile size. For example with ICSSM 1, the average size of the background level data for all images studied is less than four percent of the compressed image data, even if a small tile size of ten is chosen. Still, for optimal compression, the performance of possible background level compression methods for smaller tile size parameters is evaluated in the next section.

Practical Experiments to Assess the Compression of the Background Level

Different encoders were evaluated for the background level stream of four images, both segmented with tile sizes of 20 and 5 pixels. The two tile size parameters were chosen to evaluate both a very small tile size and a larger choice.

The tile size of 5 pixels was included in this evaluation because at small tile sizes, the background level compression becomes especially important. The background level data becomes almost unimportant with tile sizes larger than 20 pixels as the amount of stored data corresponding to the background level decreases to an almost negligible size.

The same image set which was used to evaluate the performance of background noise compression methods was used in this evaluation. Here, the results on image "com0001" would be particularly interesting as the image includes some large scale intensity variations in the background area, in contrast to the other three images "for0002", "gal0001", and "gal0004" which have a relatively flat background.

For the image "gal0004", especially high compression rates on the background levels should be possible since the background level data is – like the original image – encoded with a high bit depth.

Four different encoders were tested for ICSSM 3 on the background level data stream: the arithmetic coder using order zero and order one coding, Zip compression and Bzip2. The measured compression rates on the background level data streams are given in Table 5.3.

Table 5.3: Compression ratios of four tested encoders on the background level data of the images "com0001","for0002", "gal0001", and "gal0004". For each of the images, the background level data, after image segmentation with a tile size parameter t set to 5 and 20 pixels, was examined. Ar.(0) stands for the arithmetic coder order zero, Ar.(1) for arithmetic coder order one.

	Encoder				
Image	Zip	Bzip2	Ar.(0)	Ar.(1)	$\bar{c}$
com0001 (t=5)	1.54	**1.84**	1.43	1.11	1.48
com0001 (t=20)	1.27	**1.58**	1.30	1.03	1.30
for0002 (t=5)	2.67	**3.44**	2.55	2.68	2.84
for0002 (t=20)	1.76	**3.72**	2.49	1.71	2.42
gal0001 (t=5)	2.21	**2.85**	2.20	2.22	2.37
gal0001 (t=20)	1.51	**2.83**	2.17	1.28	1.95
gal0004 (t=5)	6.48	**9.42**	4.84	5.07	6.45
gal0004 (t=20)	7.06	**11.51**	5.19	4.61	7.09
$\bar{c}$	3.06	4.65	2.77	2.46	

In the evaluation, the Bzip2 compression method led to the highest compression ratios on all tested cases (Table 5.3). Especially with the high bit depth image "gal0004", Bzip2 achieved high compression ratios, approximately 1.5 times higher than Zip. Bzip2 is therefore used in ICSSM 3 to compact the background level information.

Typically, the optimal tile size parameter is above 20 pixels, where the background level information occupies less than one percent of the image data. Hence, compression of this information will not increase the overall performance of ICSSM 3 compared to ICSSM 2, but the identified possibility for compression was exploited.

5.3.3 Encoding of the Region of Stellar Objects

The regions of stellar objects, which typically contain elliptical images of stars, are encoded using plain Huffman encoding in ICSSM 1 and using Zip compression in ICSSM 2.

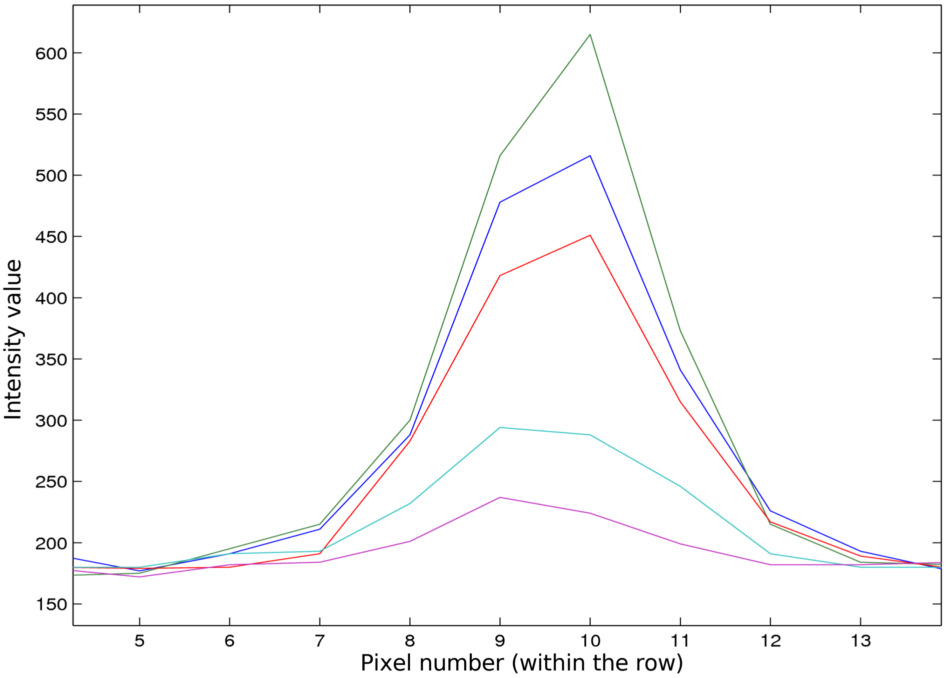


Figure 5.14: Five successive region-of-stellar-objects data rows as they are aligned in the original image (figure generated from image "gal0004").

Options that possibly offer a more size-efficient compression of this component are analysed in this section. As in previous sections, first, properties of the data component are analysed. Then, for this data component, which shows some redundancy, prediction-based redundancy reduction is examined using an illustrative example. As this example does not lead to a size-efficiency increase, the option to use a predictive decorrelation step prior to the final encoding is rejected. Finally, four promising encoders are evaluated to determine the best choice for encoding the region of stellar objects.

As the background levels, the pixels in the region of stellar objects should typically be correlated to a certain degree. The region-of-stellar-objects data consists primarily of slices through the point-spread function, with different sizes and different heights.

Figure 5.14 plots several successive region-of-stellar-objects image lines to show the dependency among them. Figure 5.15 shows a three-dimensional plot of the region-of-stellar-objects data from image "gal0004" rearranged to an almost square data block.

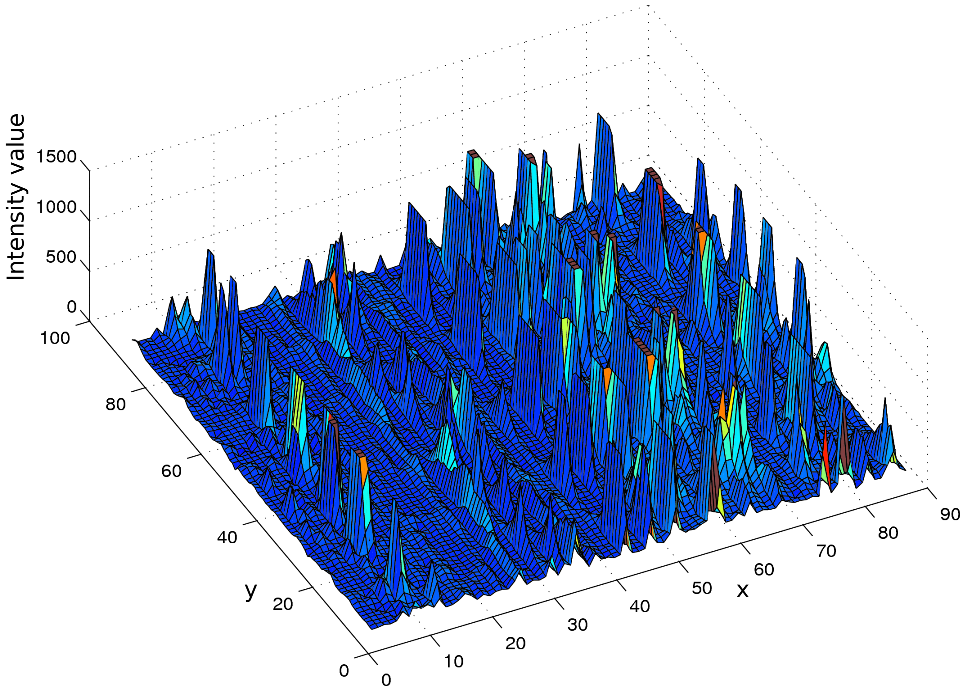


Figure 5.15: Three-dimensional plot of the region-of-stellar-objects data from image "gal0004" rearranged to an almost square data block.

For encoding the region-of-stellar-objects data, in theory it is possible to identify the parameters of the point-spread function within the region-of-stellar-objects data, subtract a model of the point-spread function and encode only the residuals. Although algorithms for the identification of such source parameters exist (Diolaiti et al. 2000), this approach is not chosen with the ICSSM algorithm as justified in Section 5.1.2.

Alternatively, compression methods which can be attributed to three main categories are examined for the compaction of the region-of-stellar-objects data:

- **Standard entropy coders:** Standard entropy coders such as arithmetic coding (order 0 and order 1 model) are examined.

- **General-purpose compression methods:** As for other data components, the Zip and Bzip2 methods, which are not adapted to specific data, are tested to examine the performance of the Deflate algorithm and the Burrows-Wheeler Transform.

- **Prediction-based methods:** Prediction-based methods may help reduce the overall size of the region-of-stellar-objects data, as there is some dependency among different data rows (Figure 5.14). The usage of a Paeth-predictior based decorrelation step is examined in the next section.

- **Existing astronomy-specific methods:** As an alternative to the options discussed above, astronomy-specific methods which were discussed in Chapter 3 could be tested for the compression of the region-of-stellar-objects data. For example, using wavelet-based approaches such as the one of Press (1992), White (1992), White and Percival (1994), Starck et al. (1995) or Louys, Starck and Murtagh (1999) can lead to a high energy compaction without losing too much run-time efficiency. The regions of stellar objects are typically small (compared to the overall size of the file), therefore such methods might not reduce the run-time efficiency of the method severely. Applying existing astronomy-specific methods would mix the approaches segmentation and image modelling with transform-based methods. Therefore, this possibility is not examined.

Paeth Predictor Sample

The Paeth predictor (Paeth 1991), which is applied in PNG (ISO/IEC 2004), was examined here for decorrelating the region-of-stellar-objects data. Paeth-based decorrelation was examined in combination with Arithmetic encoding (order 0 and order 1), Zip, and Bzip2 compression of the generated data. To keep the spatial grouping of data values, which is essential for the Paeth predictor, the predictor was applied to the complete raw image data before image segmentation.

In this experiment, after the segmentation, the Paeth predicted data was stored only for the regions of stellar objects. Although the Paeth predictor relies on upper, upper-left, and left pixel values during decompression (Paeth 1991), it could be successfully inverted during decompression, as the data surrounding the regions of stellar objects could be fully recovered from the background levels and the noise data stream.

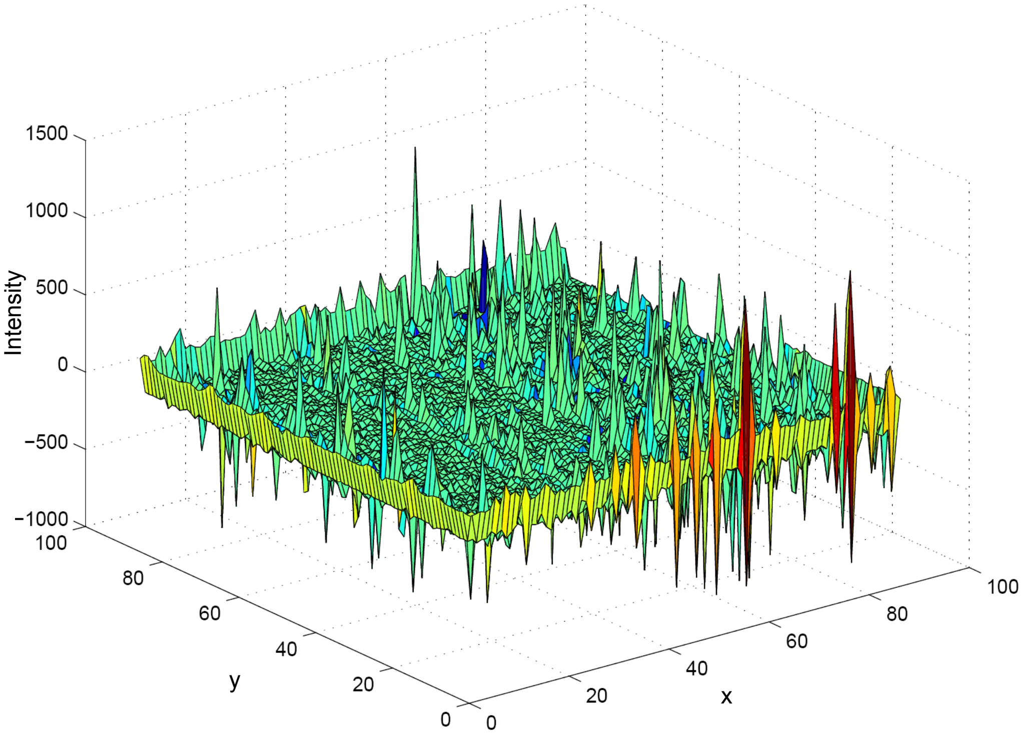


Figure 5.16: Three-dimensional plot of the region-of-stellar-objects data from Figure 5.15 after decorrelation using Paeth predictor.

Special measures had to be taken only for the first pixel row and the leftmost pixel column as they form the image border. In this case, for simplicity, the pixels outside the image border are assumed to have a value of zero.

This choice does not affect the losslessness of the encoding, but it leads to the high prediction residuals visible in Figure 5.16. Figure 5.16 shows the region-of-stellar-objects residuals from applying the Paeth predictor to the data. The data stream generated through Paeth prediction and segmentation finally has to be further compressed by other methods.

As an illustrative examination of the Paeth prediction for the region-of-stellar-objects data, here the Paeth prediction in combination with methods for further data compression was compared to the compression of the data without Paeth prediction. The same methods for compression were tested with and without Paeth prediction: arithmetic coders of order 0 and order 1 and the general-purpose compression methods Zip and Bzip2.

The measurement results obtained using the region-of-stellar-objects data stream of image "gal0004" are given in Table 5.4.

Table 5.4: Compression ratios of different encoders achieved on the region-of-stellar-objects data stream (image "gal0004", tile size parameter t set to 50 pixels).

Compr. method	Compr. ratio
Arithmetic (order 0)	2.81
Arithmetic (order 1)	2.42
Gzip	3.54
Bzip2	5.31
Paeth + Arith. (order 0)	2.42
Paeth + Arith. (order 1)	2.15
Paeth + Gzip	2.84
Paeth + Bzip2	3.63

As Table 5.4 shows, on the examined data, the performance of Paeth prediction plus the tested encoders was lower than the performance of the encoders without pre-processing. Therefore, Paeth prediction is not applied in ICSSM 3, although this merely illustrative experiment cannot guarantee that a true decorrelation using some special predictor does not lead to an enhancement of the possible compression rates. The Paeth predictor experiments should not be construed as proof that there would not be other options for better decorrelation. The compression method used with ICSSM 3 is determined in the next section, using a larger sample image set.

Practical Experiments to Assess the Compression of Stellar Objects

Again, for the choice of the region-of-stellar-objects encoder, the same four test images were chosen as for the measurements to select the encoder for the other image data components. Again, due to the higher bit depth of the original data, higher compression rates were expected for the image “gal0004”. The region-of-stellar-objects data segmented with a tile size parameter t set to 20 and 50 pixels was examined.

As with the evaluation of background noise and background level data compression, four different encoders were tested for ICSSM 3. Arithmetic coder, using order zero and order one coding, Zip compression and Bzip2 were tested on the region-of-stellar-objects data streams of the test images. The measured compression rates on the region-of-stellar-objects data streams are given in Table 5.5.

Table 5.5: Compression ratios of four tested encoders on the region-of-stellar-objects data of images “com0001”, “for0002”, “gal0001”, and “gal0004”. For each image, the region-of-stellar-objects data after segmentation with a tile size parameter t set to 20 and 50 pixels was examined. Ar.(0) stands for the arithmetic coder order zero, Ar.(1) for arithmetic coder order one.

	Encoder				
Image	Zip	Bzip2	Ar.(0)	Ar.(1)	$\overline{c}$
com0001 (t=20)	1.42	**1.64**	1.32	1.08	1.38
com0001 (t=50)	1.43	**1.70**	1.35	1.12	1.40
for0002 (t=20)	2.14	**2.92**	1.96	2.08	2.28
for0002 (t=50)	2.08	**2.82**	1.96	2.01	2.22
gal0001 (t=20)	1.45	**1.68**	1.50	1.09	1.43
gal0001 (t=50)	1.38	**1.58**	1.48	1.02	1.37
gal0004 (t=20)	3.60	**5.45**	2.93	2.58	3.64
gal0004 (t=50)	3.48	**5.31**	2.81	2.42	3.51
$\overline{c}$	2.12	2.89	1.91	1.68	

The best performing method in the evaluation given in Table 5.5 was in all cases the Bzip2 method. Surprisingly even though “com0001” includes some large scale intensity variations in the background region – which lead to a relatively large portion of region-of-stellar-objects data for a large tile size parameter t of 50 pixels – the results matched for “com0001” and the other three images. As expected, the performance of the compression on the 32-bit-per-pixel image “gal0004” was approximately twice a high as for the other images.

The region-of-stellar-objects data is, especially after segmentation, the most image specific data. It captures the data from the bright image areas. As the evaluation showed that Bzip2 compression leads to the best results on all tested image data samples, Bzip2 is used for further compacting the region-of-stellar-objects data after segmenting the image data with ICSSM 3.

5.3.4 Encoding the Region Demarcation Information

A further increase of the overall compression ratio of the ICSSM algorithm is possible by using more efficient encoding methods for the region demarcation information. A simple bitmap was used with ICSSM 1 and a Zip compressed bitmap was used with the ICSSM 2 algorithm. This section discusses possibly more size-efficient alternatives for the region demarcation information compression and reports an illustrative examination of a lossy geometrical region demarcation description. Finally, three promising encoders are evaluated to determine the best choice for the region demarcation information encoding.

The achievable compression ratio enhancement, due to more size-efficient region demarcation information encoding, remains limited because the region demarcation information contributes only about 1/16 or 1/32 of the raw image data, depending on the bit depth of the image. Nevertheless, especially for 16-bit-per-pixel images the enhancement may be worth the effort. Other methods than Zip compression, which are possibly better adapted to the data properties, include:

- **Other general-purpose compression methods:** Other general-purpose compression methods, like algorithms based on the Burrows-Wheeler Transform, can be used for compressing the region demarcation information.

- **Bi-level image compression algorithms:** Efficient methods to compress bi-level images (Hobby 1997, Fränti and Nevalainen 1995, Fränti 1994) are well-suited candidates for further compressing the region demarcation information. JBIG[8], a highly efficient compression method for bi-level images, is evaluated because of its high compression performance.

- **Vector-based description:** Vector-based methods might be used to describe the circular regions of stellar objects within the background region. Utilising the vector features in context-based compression of raster images has already been examined (Fränti et al. 2002). A method which is especially adapted to describe the elliptical regions within the data has been developed and tested to determine how far such a method may outperform the other candidates for the compaction of region demarcation information. A discussion of the algorithm and its properties is given below. Further details on the algorithm are found in Appendix A.

- **Morphological methods and tree coding:** Morphological methods, such as the ones used by Huang and Bijaoui (1990) (discussed in Section 3.2), might be another option, as well as approaches based on tree coding (Martins and Forchhammer 1996).

For the ICSSM 3 algorithm, the most promising possibilities – the first three listed above – were examined. First, the general-purpose compression methods Zip compression and the Burrows-Wheeler Transform based Bzip2 were tested. Therefore, the binary data streams of the region demarcation information of four of the test images were exported as 1 bit-per-pixel image matrix in the Portable Bitmap (PBM) format. External tools were used to measure the size-efficiency of the Bzip2 method in comparison to Zip compression.

Secondly, an implementation of the JBIG compression method was evaluated to determine the size-efficiency of this highly size-efficient state-of-the art method. To evaluate JBIG compression, the JBIG-KIT, a portable library of compression and decompression functions with a simple command line interface, was used[9]. The JBIG-KIT implements the JBIG 1 method, which is standardised in (ISO/IEC 2001).

Finally, a dedicated lossy vector-based method was developed and considered for compressing the region demarcation information. Although this

[8] http://www.jpeg.org/jbig/jbigpt2.html (25.03.2008)

[9] http://www.cl.cam.ac.uk/∼mgk25/jbigkit/ (03.04.2008)

investigation deals with lossless compression, for the region demarcation information it would be possible as well – without impinging the losslessness of the whole method – to use a lossy method, if the data is re-segmented after generating the final region demarcation map.

Geometrical Region Demarcation Information Description

Instead of using a lossless compression algorithm for the region demarcation information, one could represent the areas with an approximate geometrical description that might need far less data for the region demarcation information. But, in that case some background region pixels would be assigned to the regions of stellar objects and therefore would require more coding space. The algorithm would have to balance the data stream sizes for an optimal size reduction and it would require a re-segmentation step. Still, to evaluate a wide variety of different methods, and to get to know how compact a lossy description could be on the region demarcation information, this alternative approach was examined in an illustrative examination.

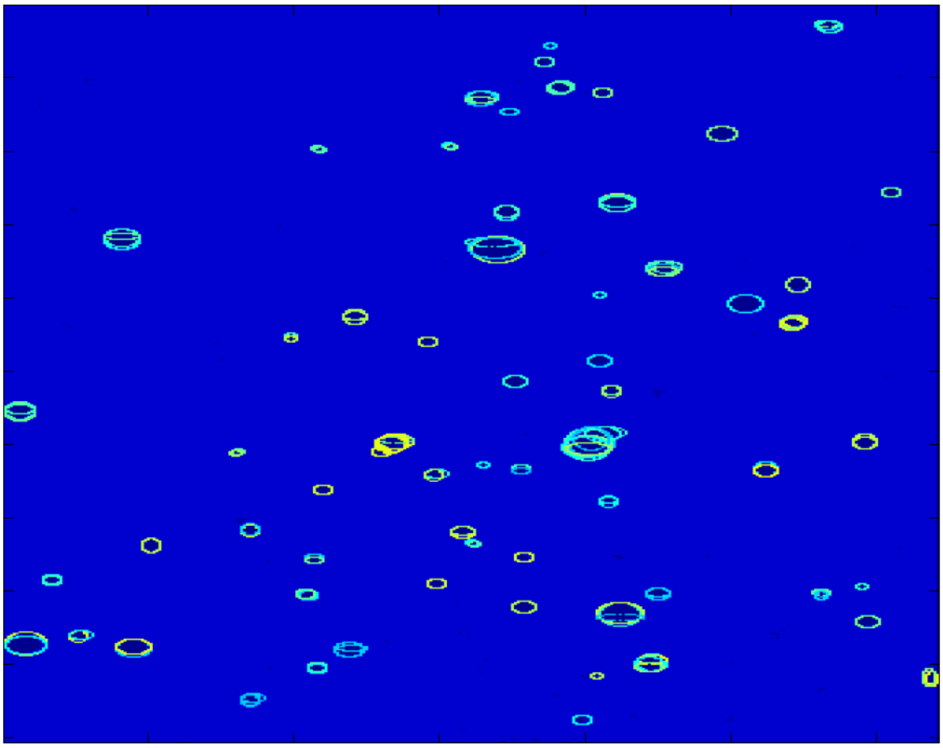


Figure 5.17: Size-efficient lossy geometrical description of the region demarcation information of image “gal0001”.

The lossy region demarcation description method segments the image into elliptical regions. Using an iterative geometrical approach, the examined method identifies potential midpoints of elliptical objects within the region demarcation information. After the removal of double and redundant hypothetical ellipsoid objects, a size-efficient lossy geometrical description of the region demarcation information is generated.

A sample of an automatically generated geometrical description of the region demarcation information of image "gal0001" is shown in Figure 5.17. This lossy approach to describe the region demarcation information data required less than five percent of the size of the original bitmap in a range of tested cases. But this positive effect on the region demarcation information would be reduced by its negative effect on the segmentation.

Unfortunately, the lossy geometrical description would require a second re-segmentation pass for a lossless compression of the whole image, without leading to a significant improvement of the overall compression ratio. Therefore, this method did not prove to be an optimal method for the region demarcation information compression and was not included in the final evaluation.

Practical Experiments to Assess the Compression of the Region Demarcation Information

Due to the disadvantages of the lossy geometrical description, the final examination to determine the encoder choice for ICSSM 3 did only include Zip compression (for comparison to ICSSM 2), Bzip2 compression, and the dedicated lossless bi-level image compression method JBIG. Results from measurements using images segmented with a tile size parameter t of 10 and of 50 pixels, and the region demarcation information encoders are shown in Table 5.6[10].

The first result shown by these measurements is that the possible compression rate typically increased tremendously for the region demarcation information with a larger tile size. The larger tiles led to higher noise estimates, that means more pixels are assigned to the background region. Hence, less spots of region-of-stellar-objects data appeared within the background

[10] The smaller tile size parameter $t = 10$ pixels was chosen for this component, as – unlike the other components where $t = 20$ was used – the region demarcation information becomes much more complex with very small tile sizes for some of the images. See also Figures 5.7 and 5.8.

region and the region demarcation information became simpler, and better compressible. This effect was particularly noticeable for flat and large images such as “gal0001” and “gal0004”, in contrast to “com0001” with its large-scale intensity variations.

Table 5.6: Compression ratios of four tested encoders on the region demarcation information data of images “com0001”, “for0002”, “gal0001”, and “gal0004”. For each of the images, the region demarcation information was examined using images segmented with a tile size parameter t set to 10 and 50 pixels.

	Encoder			
Image	Zip	Bzip2	JBIG	$\overline{c}$
com0001 (t=10)	5.22	5.70	**7.89**	6.27
com0001 (t=50)	4.97	5.28	**6.97**	5.74
for0002 (t=10)	2.17	2.15	**3.44**	2.59
for0002 (t=50)	3.40	3.59	**6.34**	4.44
gal0001 (t=10)	5.47	6.35	**8.95**	6.92
gal0001 (t=50)	12.62	16.53	**23.99**	17.71
gal0004 (t=10)	6.36	7.78	**10.18**	8.11
gal0004 (t=50)	25.88	35.89	**48.89**	36.89
$\overline{c}$	8.26	10.41	14.58	

With an average compression ratio of almost 8.26, even the general-purpose Zip compression algorithm achieved high compression ratios on the region demarcation information data sample. The usage of Bzip2 enhanceed the compression ratio only a little compared to Zip. On the test set, Bzip2 achieved an average compression ratio of 10.41. On the image “for0002”, segmented with tile size parameter t set to 10, the performance was even a bit worse than the one of Zip. The dedicated bi-level image compression algorithm JBIG outperformed the general-purpose compression tools by a factor of approximately 1.5 and higher. Therefore, JBIG is chosen for ICSSM 3. Compression ratio enhancements are especially expected for 16-bit-per-pixel images, where the region demarcation information contributes a higher proportion of data to the whole compressed image, in contrast to 32-bit-per-pixel images.

The possible size reduction through region demarcation information compression is almost completely exploited at this level, as the region demarcation information only contributes very little to the size of the compressed image. Therefore, the geometrical method is not examined further. Even if the geometrical method could further enhance the compression ratio, this would have almost no measureable effect on the overall compression ratio.

5.3.5 Algorithm Outline

The ICSSM 3 algorithm differs from ICSSM 2 (see Section 5.2.3) with regard to the encoders which are applied to the different data components. Compared to the Zip compression method which is used with ICSSM 2 for all data components, according to the measurements on the test image set in Sections 5.3.1, 5.3.2, 5.3.3, and 5.3.4, for ICSSM 3 enhancements regarding size-efficiency are possible for the individual data components.

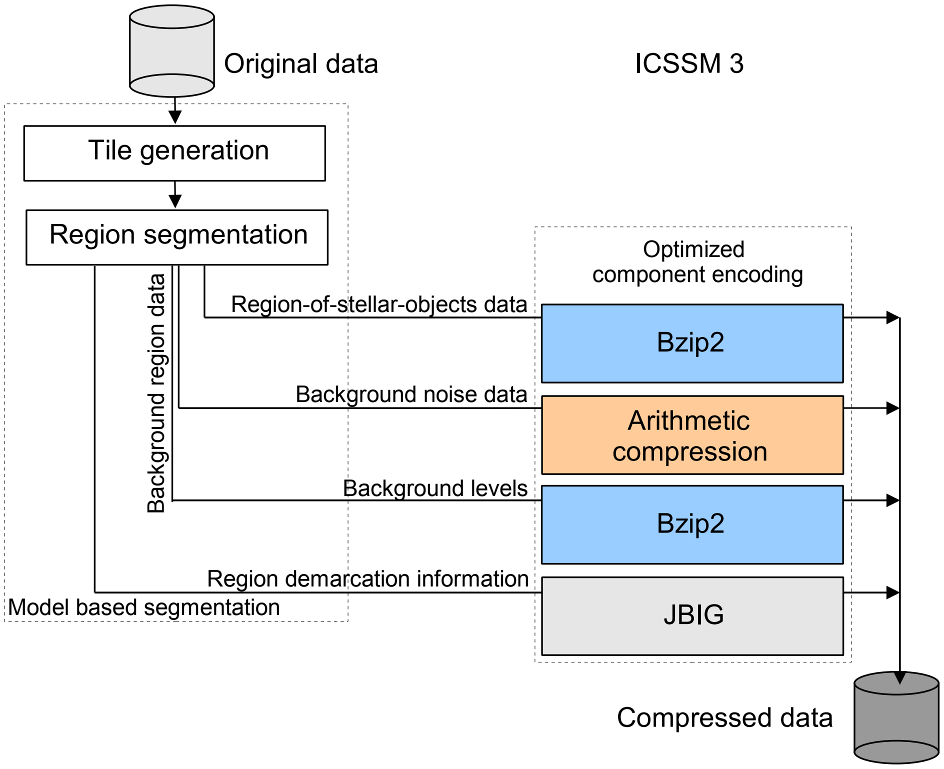


Figure 5.18: Overview of the ICSSM 3 algorithm.

As only the data encoding step was changed compared to ICSSM 2 and the segmentation method was kept unchanged, decoding works as with

ICSSM 2 or ICSSM 1. The dataflow for ICSSM 3 is given in Figure 5.18. A detailed performance evaluation of ICSSM 3 is presented in Section 6.5.

5.3.6 Summary

For optimum compression performance with the ICSSM 3 algorithm, characteristic properties such as the value distribution, noisiness and predictability of successive values of the data components, generated through segmentation, were determined. Then, compression methods adapted to the properties of each data component were evaluated in experiments and the best performing method was chosen for compressing the data component. The evaluations showed that arithmetic compression has to be used for the noise data, Bzip2 compression for both the region-of-stellar-objects data and the background levels, and JBIG for the region demarcation information mask.

ICSSM 3 can be considered as an optimised version of the ICSSM algorithms, with the following restrictions. The possibility to use other existing astronomy-specific compression methods, such as the wavelet based method of (Press 1992, White 1992, White and Percival 1994, Starck et al. 1995, Louys, Starck and Murtagh 1999) for the region-adaptive compression in combination with the segmentation method, was not examined. This might be an option for future research. Furthermore, the tile size remains a segmentation parameter which is not automatically determined.

5.4 Summary

In this chapter, three algorithm versions of the novel image compression method, ICSSM, were presented. The novel algorithms are based on image segmentation and region-adaptive bit allocation in order to exploit the differences in the dynamic range of different image regions. Different types of region-adaptive bit allocation schemes are used for the different algorithms proposed.

The decomposition of the data is based on the image model which was presented in Chapter 4. It consists of the three components background level, background noise and the signal component resulting from stellar-object irradiation, which may be zero in certain regions.

Using an estimate of the dynamic range of the Gaussian noise component, determined for a local image tile (of a predefined size), the image is split into two distinct areas: background region and regions of stellar objects. The background level is stored, and information about the spatial extent of the regions into which the image is decomposed is stored as a region demarcation information mask.

In regions of stellar objects, the intensity level is kept unchanged, prior to the final encoding. For background regions, the background noise component is determined by subtracting the background level from the detected CCD signal $i(\mathbf{x})$. The three algorithmic options of the ICSSM algorithm mainly differ with respect to the encoding applied to the data components, namely the region-of-stellar-objects data, the background level data, the background noise data, and the region demarcation information of the image.

With ICSSM 1, the background noise component and the region-of-stellar-objects data are encoded independently using Huffman coding; background level data and the region demarcation information are encoded directly in binary format. The ICSSM 1 version serves primarily as a proof-of-concept algorithm version to demonstrate what compaction can be achieved if only encoders of limited capability are applied for compacting the data components generated through segmentation.

With ICSSM 2, a trade-off between simplicity and coding efficiency is applied; all data components are compressed using Zip. This algorithm version is intended as a practically simple but useable segmentation-based compression method.

Finally, for highest compression performance, ICSSM 3 applies data compaction methods best matched to the properties of each data component. Arithmetic compression is used for the noise data, Bzip2 compression for both the region-of-stellar-objects data and background level data, and JBIG is used for the region demarcation information.

While the focus of this chapter was on introducing the versions of the ICSSM algorithm, the focus of the next chapter lies on evaluating the novel algorithms with respect to their size-reduction performance. The chapter also assesses the performance of some existing astronomy-specific lossless compression methods.

Chapter 6

Performance Evaluation

This chapter presents a performance evaluation of the newly developed lossless compression algorithms based on signal modelling, segmentation and region-adaptive bit allocation (proposed in Chapter 5), and of existing astronomy-specific lossless compression methods, discussed in the literature review (Chapter 3).

The evaluation of existing methods includes all methods from the FITSIO package, the Zip, Rice, and the PLIO method. Also, the Fitspress and Hcompress methods are included, as well as six wavelet based methods which are using the Pyramidal Median Transform, Haar wavelet, a Min-Max transform, Mallat-Daubechies, Feauveau transform, and a combination of methods. The set of evaluated existing methods is completed with a method based on mathematical morphology and five lifting scheme based methods using median prediction, integer Haar wavelet transform, integer Cohen-Daubechies-Feauveau wavelet transform, integer $(4, 2)$ interpolating transform, and an integer 7/9 wavelet transform. The newly developed algorithms and the existing approaches are compared empirically using a test image set.

Section 6.1, which presents methodological considerations regarding the performance evaluation, consists of two parts.

First, the comparability of compression rates measured with existing algorithms and the ones determined experimentally for the three new algorithms is discussed in detail in Section 6.1.1. Section 6.1.1 highlights that in order to guarantee the comparability of ICSSM results to results from the evaluation of existing methods, with all results reported for the ICSSM algorithms, the overhead for including the compressed data in a FITS file is accounted for.

Second, Section 6.1.2 presents the astronomical test image set used by other researches (Grosbol et al. 1989, Sabbey 1999) for the evaluation of astronomical image compression methods. It serves as the test set for the evaluations of existing and newly developed methods.

Section 6.2, the first evaluation section in this chapter, discusses existing state-of-the art compression methods developed by other authors. First, a brief review of some results reported by other authors is presented. Then, available implementations of some methods presented in Chapter 3 are reviewed, and the evaluation procedure is defined. Finally, results from the evaluation of these methods, conducted in this investigation, are presented and discussed in detail. In one case, where no implementation of the method was available, existing results reported by the original author of the method are included in the comparison – these were obtained with the same test image set as the one used in this research.

After comparing state-of-the-art astronomy-specific methods, the following sections focus on the newly developed segmentation and region adaptive bit-allocation based algorithms. The size-efficiency of ICSSM 1, ICSSM 2, and ICSSM 3 is examined in Sections 6.3, 6.4, and 6.5 respectively. The ICSSM 2 and ICSSM 3 algorithms are evaluated using software implementations; the implementation is in each case discussed prior to reporting the measured results.

A detailed comparison of all ICSSM algorithms and existing methods, including the discussion of application classes, is given in Section 6.6. The chapter ends with a summary in Section 6.7, highlighting and summarizing the most important findings.

6.1 Some Methodological Considerations

Methodological considerations regarding the performance evaluation are the focus of this section.

The first one of its two parts, Section 6.1.1, discusses measures taken to ensure the comparability of compression rates measured with existing algorithms to the ones determined experimentally for the three new algorithms. The second one, Section 6.1.2, presents the astronomical test image set used for the evaluations of existing and newly developed methods.

6.1.1 Comparability of File and Data Compression Measurements

If the compression ratios measured with different methods shall be compared, measurement comparability has to be assured between the methods under evaluation. For the existing methods, this topic is discussed in Section 6.2.3. It is ensured that for existing methods the file-based compression ratio, which includes the data header, has been measured. The file based compression ratio c_f is defined in Equation 6.1, where R is the size of the uncompressed file, while $C(R)$ stands for the compressed file:

$$c_f = \frac{R}{C(R)} \tag{6.1}$$

The comparability of ICSSM results to those for existing method is discussed and assured in this section; two causes that might impinge the comparability are examined. First of all, the difference between the file-based and the data-based compression ratio is discussed and accounted for. After that, another possible source of overhead that might have to be accounted for with the ICSSM methods is discussed.

For the ICSSM methods, primarily the ratio of the uncompressed and compressed image data was determined. With this primary result, the data header is not accounted for, neither with the raw nor with the compressed data. Thus, in order to ensure the comparability of the ICSSM results and the results determined for existing methods, the primary results for the ICSSM methods were corrected to include the data header overhead.

First of all, the question which arises is how the file-based compression ratio c_f, measured for existing methods, differs from the one measured for the image data only, c_d. c_f shall be calculated from c_d. c_d is defined as

$$c_d = \frac{D}{C(D)} \tag{6.2}$$

where D is the size of the raw uncompressed image data and $C(D)$ is the size of the image data after compression.

Given the data header size H, the file based compression ratio c_f can be calculated from the uncompressed data and the compressed image data as follows:

$$c_f = \frac{H + D}{H + \frac{D}{c_d}} \tag{6.3}$$

As $H \ll D$ and $H \ll C(D)$, $C(D) = \frac{D}{c_d}$, c_f and c_d should be almost similar in size. For the worst case, the image "tuc0003" with the biggest header size compared to the image data, the header size H is 1.6 KByte compared to a data size D of 203.7 KBytes. The compression ratio c_d which was measured is 2.51, with ICSSM 1 and a tile size of 50 pixels, the compression ratio c_f is:

$$c_f = \frac{1.6KB + 203.7KB}{1.6KB + \frac{203.7KB}{2.51}} = 2.48 \tag{6.4}$$

Disregarding the header would lead to an error of 0.03, for a compression ratio of 2.48.

To ensure that the adjustment factor is accounted for when comparing the ICSSM measurements to those of other available methods, the adjustment factor $\frac{c_d}{c_f}$ was measured for all thirteen test images. The file-based compression ratio c_f was calculated from the data-based one, which was measured directly in the experiments with the ICSSM methods (described in Sections 6.3, 6.4, and 6.5).

After accounting for the file header overhead with the measurements for the ICSSM algorithms, one other possible cause that might impinge on the comparability of the results of ICSSM methods and existing methods remains. For the ICSSM methods, the size of the compressed data was determined by summing up the sizes of the compressed data streams. The question remains whether no additional storage is required for embedding the data streams into the output file.

If the data is encoded into a FITS file[1], no additional overhead is required to combine the data streams into a single file if the following method is used for storing the data.

First, the region demarcation information is stored in the file, followed by region-of-stellar-objects data values, background noise data values and the background levels. Decoding works as follows: The overall size of the first data component, the region demarcation information, is already known from

[1] This overhead is already accounted for with the adjustment factor discussed earlier.

(FITS-) header information. From the region demarcation information, the decompression algorithm is able to derive the amount of region-of-stellar-objects data values and the amount of noise values to decode. After decoding these components successively to memory, the current reading position within the file arrives at the last data component, the background level data. Reading it until the end of file discriminator and decoding it reveals the background level data stored. The tile size parameter can be stored in a standard FITS header entry, which was already accounted for with the adjustment factor discussed earlier.

With the decoded and separated components, the algorithm then may restore the image by adding the background noise to the background levels, in background regions and outputting the signal values successively into the region-of-stellar-objects values. No further measures have to be taken to ensure the comparability of measurements.

With all measurement results reported for ICSSM methods, the compression ratios given in this thesis incorporate the performance loss due to the uncompressed FITS file header. Therefore, there is still an option for further improvements – not only for the ICSSM methods, but for existing methods as well.

The difference between c_f and c_d, discussed above, is small but it can be further reduced by more efficient encoding methods for the header. With its fixed length data format and the keyword-to-value assignment scheme, FITS header data compression is a specific subject which should be dealt with separately.

Even if the header is kept uncompressed after conversion, the header size may be reduced by converting it to a XML-representation instead of using the fixed-block size format. Also, using XML for the header data would be an option for the future, as parsers and compression methods for XML data (Green et al. 2004, Cheney 2001, Liefke and Suciu. 2000) already exist. The integration of the header data into an XML format may be useful for the integration of image meta-data into databases (Debray 2002). Further details about the evaluation procedure are given later in this chapter.

6.1.2 Test Images

This section discusses the astronomical test image set, which is used in this research to evaluate the existing and the newly developed compression methods. The test image set was chosen to fulfil the following criteria:

1. The test image set shall be freely available in order to allow the reproducibility of results obtained.

2. If possible, the test image set shall be accepted by other researchers on astronomical image compression techniques or other researches dealing with astronomical image properties.

3. The test image set shall contain integer stellar field image data, as the research presented here focuses on this type of data.

4. The test set shall contain images of 16-bit-per-pixel and of 32-bit-per-pixel bit depth as both data formats are used nowadays.

5. The test image data set shall contain a sufficient amount of test images.

6. The test image set shall contain a diverse range of images, regarding their properties (exposure time, content, typical imperfections, and size).

Sample sets for astronomical image processing packages were analysed if they could meet the requirements listed above.

One particular image set, which is referred to as IRAF test images (IRAF 2005), was chosen as it meets the requirements listed above. In addition to being freely available, the test image set has previously been used by other researchers to evaluate astronomical image compression methods (Grosbol et al. 1989, Sabbey 1999). Therefore, reproducibility of the experiments and the acceptance of the test image set are possible, and the first two requirements listed above are fulfilled.

Also, the test image set contains a sufficiently large number of integer stellar field images for the research discussed here. The test images are listed in Table 6.1. Thus, the second and the fourth requirements are fulfilled. The image test set also contains images encoded with a bit depth of 16-bit-per-pixel and of 32-bit-per-pixel, thus the third requirement is fulfilled as well.

Finally, the image set – despite comprising the stellar field images only and not interferograms – contains a diverse set of image properties. Different exposure times are for example found on the images “ngc0001” and "ngc0002". The long exposure version "ngc0001" even contains a typical effect called blooming, which occurs with bright sources on images with a long exposure time. Charge from very bright image areas exceeds the capacity of the respective CCD pixels and flows to surrounding pixels. Typically, pixels of the same column are affected, which leads to bright strokes through the brightest image areas. Additionally, another typical image particularity, a bright CCD column, is found in image "for0001", and the image "com0001" contains large-scale variations in the background region. The required image content is found within the data test set, as stellar field images, a crowded starfield, and galaxy clusters are present.

Table 6.1: Thirteen astronomical images used to evaluate the compression methods and their properties. The dynamic range of the image data is given in the column labelled "bpp." for bit-per-pixel.

Test image set

No.	Name	Description	bpp.	Image size
1	com0001.fits	NGC4874 coma field, strong image gradients	16	500 x 500
2	for0001.fits	Fornax dwarf in V, a crowded starfield	16	305 x 512
3	for0002.fits	Fornax dwarf in B, a crowded starfield	16	305 x 497
4	gal0001.fits	Galaxy cluster 601/0637-53/R/CL	16	320 x 503
5	gal0002.fits	Galaxy cluster coadded 727/0637-53/R/CL	16	264 x 427
6	gal0003.fits	SA 68 (MPF3869 SA68-1 T)	32	1000 x 1000
7	gal0004.fits	SA 68 (MPF3869 SA68-2 T)	32	1000 x 1000
8	ngc0001.fits	NGC 3201 in V	16	321 x 507
9	ngc0002.fits	NGC 3201 in V, shorter exposure	16	322 x 508
10	sgp0001.fits	SGP J: Composite	32	234 x 441
11	sgp0002.fits	SGP R: Composite	32	234 x 441
12	tuc0003.fits	47 Tuc off-center in V (47T/V15,F1,ROT)	16	420 x 246
13	tuc0004.fits	47 Tuc off-center in B (47T/B30,F1,ROT)	16	420 x 246

Table 6.1 shows a complete list of the images, a short description of their content and technical information related to the images. This information includes the image size and the bit depth. The description of the content is partly based on (Grosbol et al. 1989). Preview images and image histograms for the complete set of test images are included in Appendix C.

6.2 Comparison of Existing State-of-the Art Methods

An overview of all compression techniques, which have been applied in astronomy for lossy and lossless compression, was given in Chapter 3. The main aim of the comparison of astronomy-specific compression methods presented here is to evaluate what compression ratios can be expected for lossless methods. A detailed evaluation of lossy methods, in terms of achievable compression ratios and their computational complexity, is not the focus of this evaluation. Lossy methods are no option for archiving raw scientific-quality data in astronomy. Also, such a comparison would require an investigation of the photometric and astronometric properties of lossily compressed images as well as a qualitative examination of the visual quality of the results.

Lossless compression methods found in the open literature, for which implementations were available, are included in this evaluation. In a few cases, where no software was available for testing, the discussion has to rely on results reported in the literature. Interesting results from some pertinent investigations reported in the literature are summarized in Section 6.2.1.

The methods which were available for the evaluation are discussed in Section 6.2.2. Details of the evaluation procedure, which focus on the compression ratio achieved by existing methods, are given in Section 6.2.3.

Finally, the evaluation results are discussed and summarized to highlight which methods can be used for the compression of raw scientific-quality astronomical data, which is the intended application of the compression methods developed in this thesis.

6.2.1 Summary of Published Results from Some Pertinent Investigations

As summarized in Sections 3.3 and 3.4, a range of astronomy-specific hybrid and lossless compression methods has already been proposed. With experiments on FITS data, Weghorn (2002) measured the efficiency of standard compression methods applied to astronomical images. Using a small set of sample images, he measured data compression ratios of approximately two, using classical methods like Huffman or Lempel-Ziv coding. General-purpose methods like those from the Lempel-Ziv group have the advantage that implementations are available for virtually all computer platforms used today.

Weghorn (2002) also examined straightforward astronomy-specific methods, like using a look-up-table and allowing any bit depth of data. Possible compression ratios measured with those specific methods exceed those achieved using classical methods. An early comparison of lossless JPEG, a wavelet based method, and Zip compression was published by Louys, Starck and Murtagh (1999). The wavelet transform achieves slightly higher compression rates than lossless JPEG, which additionally suffers from rounding errors. Therefore, "lossless JPEG" is not truly lossless. Examining data compression for the "Next Generation Space Telescope", Nieto-Santisteban et al. come to the conclusion that it is difficult to compress astronomical data using lossless compression algorithms such as Huffman, Lempel-Ziv, run-length or arithmetic coding (Nieto-Santisteban et al. 1999). They report that this difficulty is due to the noise present in astronomical images. They report that techniques such as the Rice algorithm and its derivatives (White and Becker 1998) would probably perform better than other methods in exploiting the "almost similar value" property of adjacent pixels.

6.2.2 Methods Evaluated Experimentally in this Research

Following the aim to provide an evaluation of all fully-documented methods found in the literature, a concise overview of lossless methods for compressing astronomical images is given here.

The overview presented below summarizes methods identified in literature, which are claimed to work in a lossless mode.

- Fitspress: The method, discussed in Section 3.3.1, is one of the first astronomy-specific methods; it is based on the Daubechies wavelet transform.
- Hcompress: Hcompress is another wavelet-based method which was also discussed in Section 3.3.1.
- PMT: The Pyramidal Median Transform, introduced as an alternative to the wavelet transform by Starck et al. (1995), was discussed in Section 3.3.2.
- Integer-to-Integer Transform: An integer-based lossless compressor using Sweldens lifting scheme was proposed in (Louys, Starck and Murtagh

1999) for lossless compression. It was discussed in Section 3.4.1.

- Compfits: This method, discussed in Section 3.3.3, attempts to separate compressible and incompressible image bit-planes. Compfits is reported to achieve compression rates comparable to Fitspress (Véran and Wright 1994).
- Switched Linear Prediction with Rice Coding: This is a prediction and entropy encoding method, which was discussed in Section 3.4.2.
- Signed Huffman Coding: The entropy encoding scheme, dedicated for the compression of interferograms, exploits the distribution of pixel-to-pixel differences. It was discussed in Section 3.4.3.
- FITSIO Compression Standard: Current implementations of the FITSIO library for reading and writing FITS files include three compression methods:
  - Zip: an implementation of the Deflate algorithm, which is a combination of LZ77 with Huffman coding.
  - PLIO: an algorithm for storing compressed image masks.
  - Rice: an entropy encoding method.

  All FITSIO methods were discussed in detail in Section 3.5.

The above list of methods can be assigned to three different categories. Fitspress, Hcompress, PMT, and the lifting scheme methods are transform-based methods that follow the typical design of transform-based compression methods. Zip, a popular data compression and archival format, and Rice coding are general-purpose compression methods like Signed Huffman coding, which is an entropy coding method especially designed for astronomical interferograms. Compfits, Switched Linear Prediction, and PLIO can be assigned to model-based compression methods which exploit – to a degree – special characteristics of the data to compress.

For some of the compression methods listed above, there was no software implementation available to be used for an experimental evaluation. Fitspress and Hcompress are available as standalone software. The MR/1 software package provides a range of wavelet-based methods for image analysis and compression, which have been published and compared in (Louys, Starck and

Murtagh 1999) and (Louys, Starck, Mei, Bonarel and Murtagh 1999). The methods provided by the package include the Pyramidal Median Transform, the Haar wavelet, a Min-Max transform, the Mallat-Daubechies transform, and the Feauveau transform. The MR/1 software package also provides a range of dedicated lossless wavelet-based methods which use integer arithmetic. The MR/1 function "mr_lcomp" compresses an image using an integer wavelet transform via the lifting scheme. The coefficients are encoded losslessly for integer-valued images. Five different lifting schemes are available for lossless compression: median prediction, integer Haar wavelet transform, integer Cohen-Daubechies-Feauveau wavelet transform, integer $(4, 2)$ interpolating transform, and an integer 7/9 wavelet transform. A detailed documentation of all different transform and lifting scheme methods is given in (Murtagh 2006).

Unfortunately, the FTP sites for Compfits and for the method of Sabbey et al. (1998) are closed and the software is not available in public domain. As Compfits is reported to achieve compression ratios comparable to Fitspress, the evaluation of Fitspress gives an estimate of what can be expected from Compfits (Véran and Wright 1994). Sabbey (1999) report that their method, on 25 sample test images in nearly all cases, achieves at least the same compression ratio as Hcompress, Fitspress, or Compfits + Compact.

As (Sabbey 1999) reports that he uses the same images (Grosbol et al. 1989) as the research presented here, his results can be compared with the ones measured in this research. The results of Sabbey (1999) were extracted from his publication. Due to the non-availability of a software implementation of the method, the rigid testing for losslessness performed with all other methods could not be applied for this method. This has to be kept in mind when comparing the results to the other methods.

The Signed Huffman Coding scheme of Weghorn et al. (1996) is not included in the tests as it is intended for astronomical interferograms rather than for stellar-field images. The FITSIO library provides sample programs that use its Zip, PLIO, and Rice compression methods. All three methods are included in the evaluation. Table 6.2 lists all tested methods and the parameters which have to be used for lossless compression.

Table 6.2: Tested lossless compression methods

Method name	Function parameters used
Hcompress	./fcompress -s 1 x.fits
Fitspress	./fitspress -c x.fits
FITSIO Zip	./compress_fits x.fits outfile g 32 32
FITSIO PLIO	./compress_fits x.fits outfile p 32 32
FITSIO Rice	./compress_fits centaur outfile
PMT	./mr_comp -m 1 -r -l -f -v x.fits
Mathematical Morphology	./mr_comp -m 2 -r -l -f -v x.fits
Haar wavelet	./mr_comp -m 3 -r -l -f -v x.fits
Min-Max Transform	./mr_comp -m 4 -r -l -f -v x.fits
Mallat-Daubechies Transform	./mr_comp -m 5 -r -l -f -v x.fits
Feauveau Transform	./mr_comp -m 6 -r -l -f -v x.fits
Mixed WT and PMT Method	./mr_comp -m 7 -r -l -f -v x.fits
Lifting: Median Prediction	./mr_lcomp -m 1 -f -v x.fits
Lifting: Int. Haar wavelet	./mr_lcomp -m 2 -f -v x.fits
Lifting: Int. CDF wavelet	./mr_lcomp -m 3 -f -v x.fits
Lifting: Int. (4,2) interpol. transf.	./mr_lcomp -m 4 -f -v x.fits
Lifting: Int. 7/9 wavelet	./mr_lcomp -m 5 -f -v x.fits

6.2.3 Evaluation Procedure

A comparison of the compression ratio is not as trivial as it seems on first impression. Measures have to be taken to ensure that all compression methods treat the different data components header and image data in the same way. The comparison is based on the file compression ratio c_f, defined in Equation 6.1.

In order to allow the comparability of compression measurements, all methods under evaluation were studied carefully. First of all, it was ascertained that all methods treat the data header the same way.

Hcompress, Fitspress, and the FITSIO methods do not remove header data from the file; neither do they compress the data header. Other than these methods, the MR/1 compression method normally removes data from the header which could be meaningful to the astronomer. But using the appropriate option of the MR/1 software, it can be ensured that no header data is removed. During compression, MR/1 includes a single additional line in the header to indicate the compression method which was used. Thus, with all methods under evaluation it was ensured that the header data is kept uncompressed and complete.

To ensure that all tested algorithms really work losslessly on the data, the tool "fitsdiff" was used to compare the original FITS file with the version obtained from decompressing a compressed file. "fitsdiff" reports the number of different pixels, the maximum and mean absolute difference and the root-mean-square difference of pixels (Hsu and Hodge 2003). The evaluation was performed using three shell scripts for each compression method.

1. The first script compresses all test images using the compression method under evaluation.
2. The second one decompresses and renames the files.
3. The third one compares the original file to the processed version using "fitsdiff".

All compression methods were applied to the thirteen integer-valued test images described in Section 6.1.2.

6.2.4 Results

The compression ratios measured in the experiment to evaluate existing methods are listed in Table 6.3. For layout reasons, the images are listed as a numbered sequence corresponding to an alphabetical ordering of file names (com0001, for0001, for0002, gal0001, gal0002, gal0003, gal0004, ngc0001, ngc0002, sgp0001, sgp0002, tuc0003, and tuc0004). In the table, the '–' symbol denotes that compression was lossy according to verification with "fitsdiff", the '⋄' symbol denotes that compression was not possible due to documented software limitations. No measurement results are given for the MR/1 method labelled "mathematical morphology" (Huang and Bijaoui 1991) because it is a purely lossy method. In this section, the most apparent general outcomes are discussed first; each single method is discussed in detail later. A diagram presenting the average compression ratios achieved by individual methods is given in Figure 6.1. Hcompress compressed only one image losslessly, thus no range is given in Figure 6.1.

One very important outcome of this evaluation was that not all compression methods that claim to be "lossless" truly produce lossless compression.

The evaluation showed that there were not only differences in the achievable compression ratio on the different images, but that there were also differences in the "losslessness" of compression methods. From the 221 method-image pairs which were used for the compression ratio measurements, only 168 compression attempts succeeded and led to a truly losslessly compressed output image. One cause for this are limits in the data format that the evaluation software could process; this is normally specified in the software documentation. Some compression attempts gave an error message if an attempt to compress a wrong data type was made. For example, Fitspress could not compress some of the images, because it is limited to 16-bit-per-pixel files, while all FITSIO methods succeeded in compressing all images losslessly.

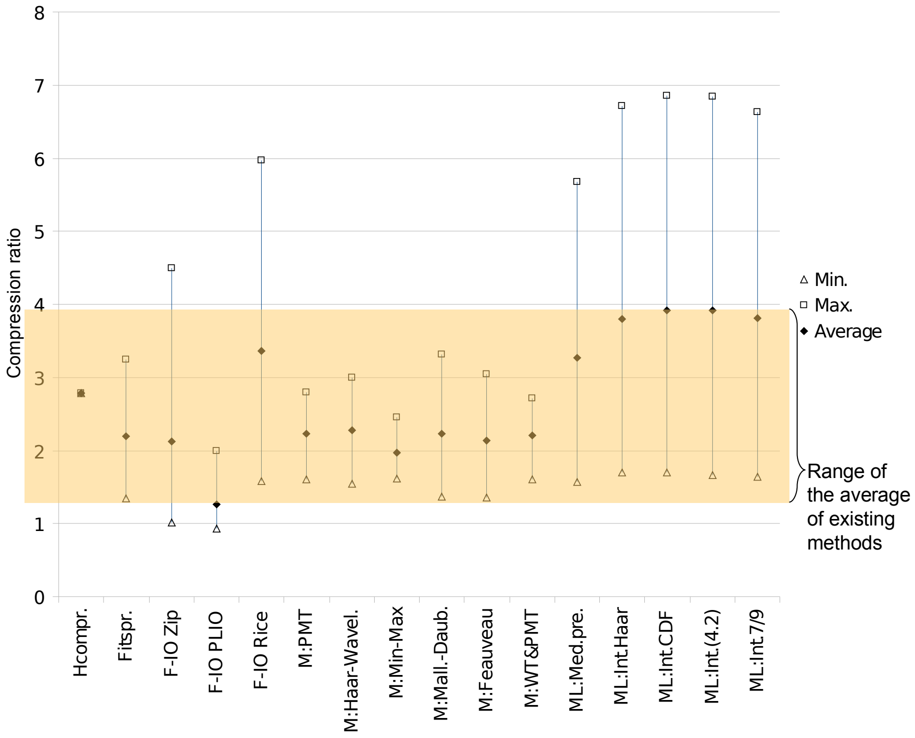


Figure 6.1: Measured average compression ratio of available, ready-to-use lossless compression tools for astronomical images.

Some methods processed the data and returned a "losslessly" compressed file which cannot be used for a truly identical reconstruction of the original data. The MR/1 standard method and Hcompress did not satisfy the losslessness criteria on a range of the thirteen test images. Hence, both methods are not an option for archiving original astronomical images of scientific grade.

The first performance evaluation candidate, Hcompress, is claimed to work losslessly on 16-bit-per-pixel integer FITS images in the software documentation. Therefore, the evaluation was only performed on these images. Despite using the program parameter setting which should lead to lossless compression, the "fitsdiff" routine reported a range of differences in nearly all images. In most cases, the measured pixel intensity differences were small – ranging from minus 2 to plus 2 intensity levels – but there were also some images where larger intensity differences were found, up to a value of nearly 0.3 times the dynamic range of the image. Hcompress reported successful compression attempts on all images, which led to compression ratios between 1 and 3.9. Despite being lossy, this range was far below the compression rates from 3 up to 30 reported by the author of the Hcompress software (White 1992). The remarkably large compression ratios White (1992) mentioned for lossless methods could not be verified in the evaluation. The resulting compression ratio, which is not truly lossless, was a bit higher than the results which were achieved by the almost similar standard MR/1 Haar transform method.

Like Hcompress, Fitspress compresses only 16-bit-per-pixel, single-plane images. Therefore, as can be seen in Table 6.3, only those images were used for the investigation. The performance evaluation supports the statement of the Fitspress manual, which reports that the size of the compressed files should be significantly (typically a factor 2.5) smaller than the original file. The average compression ratio achieved on nine test images was approximately 2.2. The reconstruction was exact in these nine cases.

The FITSIO compression methods, Zip, PLIO, and Rice, succeeded in compressing all images losslessly. PLIO is a dedicated algorithm for storing compressed image masks for an image processing package which is called Image Reduction and Analysis Facility.

PLIO achieved the worst average compression ratio. On most 16-bit-per-pixel images, it could not achieve any compression at all. The files were even enlarged a bit with compression ratios of 0.93 up to 0.97. On 32-bit-per-pixel images, compression ratios of up to approximately 2 could be achieved using PLIO.

Table 6.3: Compression ratios for existing state-of-the art methods. The '–' symbol indicates that compression was lossy according to verification with "fitsdiff", the '◇' symbol indicates that compression was not possible due to documented software limitations. The table row labelled "Pred.+Rice", lists the unverified results extracted from Sabbey (1999). They are only given for information. The last column (symbol $\overline{c}$) denotes the average compression ratio achieved on the whole image set for the respective method. For layout reasons, the test images are listed as a numbered sequence corresponding to an alphabetical ordering of file names.

Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	$\overline{\mathbf{c}}$
Bit depth	16	16	16	16	16	32	32	16	16	32	32	16	16	
Hcompr.	–	–	–	2.78	–	◇	◇	–	–	◇	◇	–	–	2.78
Fitspr.	2.03	2.62	3.24	2.57	2.05	◇	◇	1.35	1.83	◇	◇	1.97	2.08	2.19
F-IO Zip	1.14	1.51	1.75	1.43	1.18	4.48	4.50	1.02	1.35	2.96	3.30	1.43	1.49	2.12
F-IO PLIO	0.97	1.00	1.00	0.96	0.95	1.99	1.99	0.93	0.96	1.88	1.88	0.97	0.97	1.27
F-IO Rice	1.92	2.75	3.45	2.40	2.00	5.97	5.91	1.58	2.17	4.83	5.58	2.52	2.61	3.36
Pred.+Rice	2.10	3.45	4.35	2.85	2.40	4.40	3.35	1.90	2.80	3.15	3.80	3.35	3.50	3.18
M:PMT	1.85	2.44	2.80	2.42	1.95	–	–	1.61	2.20	–	–	2.35	2.46	2.23
M:Math.Mor.	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇
M:Haar-Wavel.	1.92	2.54	3.00	2.44	2.01	–	–	1.55	2.16	–	–	2.35	2.50	2.27
M:Min-Max	1.92	2.33	2.46	1.85	1.64	–	–	1.62	1.88	–	–	1.99	2.03	1.97
M:Mall.-Daub.	1.87	2.53	3.31	2.42	2.00	–	–	1.37	1.85	–	–	2.27	2.41	2.23
M:Feauveau	1.85	2.38	3.05	2.39	1.98	–	–	1.36	1.82	–	–	2.15	2.28	2.14
M:WT&PMT	1.86	2.38	2.71	2.41	1.95	–	–	1.60	2.18	–	–	2.33	2.45	2.21
ML:Med.pre.	1.91	2.57	3.16	2.45	2.02	5.63	5.58	1.57	2.13	4.88	5.67	2.37	2.48	3.26
ML:Int.Haar	2.10	3.01	3.94	2.79	2.26	6.60	6.54	1.70	2.40	5.68	6.71	2.74	2.87	3.80
ML:Int.CDF	2.10	3.17	4.55	2.79	2.30	6.53	6.47	1.70	2.46	5.81	6.86	3.00	3.17	3.92
ML:Int.(4.2)	2.09	3.17	4.68	2.78	2.30	6.50	6.43	1.66	2.41	5.81	6.84	3.03	3.21	3.92
ML:Int.7/9	2.10	3.07	4.18	2.78	2.30	6.43	6.37	1.64	2.34	5.72	6.63	2.89	3.04	3.81

The Zip and Rice compression methods led to much better average compression ratios of 2.12 and 3.36 respectively. These methods are a good, if not the best option at the moment for safely archiving astronomical images. These methods are well standardized and available free of charge. In particular, the compression ratio achieved by Rice compression is among the best of the available methods.

The MR/1 methods, which implement a range of mostly transform-based compression methods, are separated into two functions, standard and lifting scheme methods. While both functions should work in a lossless mode, the MR/1 lifting scheme methods are dedicated to lossless compression. The standard MR/1 method, labelled "mathematical morphology", is based on a morphological decomposition of the data. It was presented in a very early paper of Huang and Bijaoui (1990) and is implemented as a lossless method only. Lossless compression was not possible in this case due to documented software limitations (Murtagh 2006).

The average compression ratios of the standard MR/1 methods did not differ very much. Average compression ratios ranged from about 1.97 up to 2.27. The problem that images were not compressed truly losslessly occurred with the standard MR/1 methods applied to images with a bit depth of 32 bits per pixel. Thus, these methods should not be recommended for the archival of scientific-grade astronomical images. Even when comparing only 16-bit-per-pixel images, great performance differences between the hybrid standard MR/1 methods and the dedicated lossless MR/1 lifting scheme methods could be found. Typically, the wavelet-based methods gave higher compression ratios than standard methods. Even on the image "ngc0001", which is apparently the most difficult to compress, the best wavelet-based method – the integer Haar wavelet – led to a compression ratio of 1.7, while the best standard method gave a compression ratio of 1.62. Differences were even larger on images where both methods achieved better compression ratios. Typical average compression ratios, which can be expected with the lifting scheme MR/1 methods, were in the order of 3.75 or even above.

Although this is roughly 1.5 times the compression ratio that the MR/1 standard methods achieve, it has to be kept in mind that the standard methods fail on 32-bit-per-pixel images, where high compression ratios are possible.

6.2.5 Summary

Judging from the average compression ratios achieved in the performance evaluation for available methods, the best performing methods, which compressed all test images losslessly, are the MR/1 Lifting Scheme and the FITSIO Rice method.

Taking only the existing and tested methods into account, only the FITSIO compression methods and the MR/1 lifting scheme are an option for reliably archiving scientific-grade astronomical images in compacted format. The most efficient method, which is standardized and can be included in any software, is a straightforward entropy encoder. On average, only the lossless MR/1 methods slightly outperformed the Rice compression method on all test images. Comparing the average results measured on the test images to those from Sabbey (1999) for his own method (based on prediction and Rice coding), which averaged at 3.18, would place Sabbey's method after the MR/1 lifting scheme based methods and the FITSIO Rice method, under the supposition that the method works perfectly lossless in all cases.

For the examined existing methods, the average compression ratios of all methods for 32-bit-per-pixel images ranged from 3.81 (FITSIO Zip) up to 6.42 (the MR/1 integer Cohen-Daubechies-Feauveau wavelet transform based method). The average results on 16-bit-per-pixel images ranged from 1.37 (FITSIO Zip) up to 2.81 (the MR/1 integer $(4, 2)$ interpolating transform based method), ignoring the low performing PLIO method in both cases.

This shows that the compression results achievable for 32-bit-per-pixel stellar field images are at least twice as high as for 16-bit-per-pixel images. This finding further strengthens the argument that compression becomes even more important and successful with the currently increasing bit depths of astronomical images. Current discussions on upgrading the FITS file specification include the introduction of 64-bit-per-pixel images.

The compression ratio increase by a factor of two, between 16-bit-per-pixel and 32-bit-per-pixel images, is only surprising at first glance. As discussed in Chapter 4, the largest image area is the background region. This image area only contains background noise information, which does not differ greatly depending on the bit depth the image is encoded with. Hence, with 32-bit-per-pixel images, for the background region more than twice the amount of bits of a single background region intensity value is unaffected by noise than in the case of a 16-bit-per-pixel image. As the compression algorithms remove this overhead, the compression ratio increase by two is not surprising. Possibly, the compression ratio will almost double again with 64-bit-per-pixel images.

6.3 Evaluation of ICSSM 1

To determine the size-efficiency of the ICSSM 1 method, a software implementation of the compression method was developed according to the description given in Chapter 5. With the implementation used for ICSSM 1, the achievable compression ratio using Huffman coding was, for practical purposes, calculated rather than measured for two of the data components. This approach was chosen because the size of data compressed by Huffman coding can be easily predicted using first-order entropy. ICSSM 1 serves merely as a proof-of-concept algorithm during the development of ICSSM 2 and ICSSM 3. Hence, a full implementation of the method is not required.

The calculation considers the number of symbols, the first-order entropy of the data and the required data space for encoding the Huffman table.

6.3.1 Implementation

The segmentation step was implemented in Matlab[2]. The image was segmented automatically using the software prototype and different parameters for the tile generation step. The size of the data components after compression was measured for the binary encoded components. For Huffman encoded components, the size H_s was estimated using first-order entropy plus the required data space for encoding the Huffman table.

$$H_s = \sum_{i=1}^{k}(P(l_i) \cdot log_2 \frac{1}{P(l_i)}) + k \cdot (b + f) + 2 \cdot 8 + 16 \text{ Bit} \tag{6.5}$$

Formula 6.5 accounts for the overhead due to a required frequency table. The frequency table consists of k letters, which require as many bits as the original image bit depth b, plus the amount of bits required to store the frequency information f. f is derived from a histogram of the data. The last two addends account for the side information on the frequency table layout.

[2] Available from: http://www.mathworks.com (03.04.2008).

Three components are needed for this: the bit depth of the original letters (8 bits), the bit depth of the frequency specification (another 8 bits), and the length of the table (16 bits). This formula does not account for possible imperfections of the Huffman algorithm for certain distributions, where the optimality of Huffman encoding may be reduced in the worst case to the first-order entropy plus one bit per symbol.

The overall required size for the ICSSM 1 compressed image data was estimated by summing up the sizes of all data components after the encoding step and adding the overhead for including the data components in a FITS file. As discussed in Section 6.1.1, no additional overhead is required to combine the data streams to a single file, if the FITS file overhead is accounted for already.

The estimated results obtained by applying the ICSSM 1 compression algorithm to the test data are presented and discussed in the next section.

6.3.2 Results

Despite the relatively simple encoder used, the compression ratios estimated with ICSSM 1 are promising; results estimated for the achievable compaction on the FITS files data component are listed in Table 6.4.

The highest average compression ratio on all images was achieved with a tile size parameter of 30 pixels. The influence of the tile size on the compression ratios for the thirteen test images is discussed in detail in the next section. The average compression ratio c_f of 3.37 estimated for ICSSM 1 on all images is at par with Rice compression ratio of 3.36 (see Table 6.3). The ICSSM 1 results are only 16% lower than the best available method, the lifting scheme integer CDF wavelet method, which reaches a compression ratio of 3.92 (see Table 6.3). The ICSSM 1 compression method, based on segmentation, region-adaptive bit-allocation, and signal modelling, could compete with the existing Rice coding method, which is already included in the FITS standard.

6.3.3 Influence of Tile Size on Compression Performance

The only parameter which can be chosen with the ICSSM 1 algorithm is the tile size parameter t. The first question to be answered is whether there is an optimal tile size that leads to optimal compression ratios for each image. Therefore, the effect of tile size on the compression ratio was measured for the ICSSM 1 algorithm.

Table 6.4: Compression ratios c_f estimated for the ICSSM 1 method. The last column (symbol $\overline{c}$) denotes the mean compression ratio for the corresponding image, the last row (denoted set $\overline{c}$) gives the mean compression ratio for a certain tile size.

ICSSM 1

	Tile size								
Image	5	10	15	20	30	50	100	150	$\overline{c}$
com0001	1.75	1.82	1.82	1.82	1.80	1.77	1.68	1.64	1.78
for0001	2.34	2.55	2.57	2.58	2.58	2.53	2.46	2.46	2.52
for0002	2.76	3.01	3.04	3.05	3.06	3.02	2.91	2.89	2.99
gal0001	2.21	2.38	2.42	2.43	2.45	2.44	2.43	2.43	2.40
gal0002	1.76	1.90	1.94	1.94	1.94	1.98	1.96	1.94	1.93
gal0003	4.81	5.52	5.68	5.74	5.78	5.81	5.85	5.85	5.64
gal0004	4.81	5.53	5.70	5.78	5.87	5.90	6.07	6.13	5.72
ngc0001	1.35	1.53	1.58	1.57	1.57	1.56	1.51	1.57	1.53
ngc0002	2.04	2.26	2.29	2.30	2.28	2.26	2.21	2.20	2.24
sgp0001	4.41	5.14	5.37	5.37	5.46	5.55	5.60	5.71	5.33
sgp0002	4.85	5.68	5.91	5.91	6.04	6.01	6.02	6.02	5.82
tuc0003	2.12	2.39	2.45	2.46	2.45	2.44	2.41	2.39	2.40
tuc0004	2.21	2.50	2.55	2.56	2.54	2.52	2.45	2.39	2.48
Set $\overline{c}$	2.88	3.25	3.33	3.35	3.37	3.37	3.35	3.36	

Table 6.4 shows that for most 16-bit-per-pixel images (the compression ratios range between 1.35 and 3.06), the optimal tile size t ranges from 15 up to 50 pixels. Only for two 16-bit-per-pixel images ("gal0002" and "tuc0003"), a higher compression ratio is achieved with a much larger tile size. Other than for the 16-bit-per-pixel images, the compression ratio for 32-bit-per-pixel images ("gal0003", "gal0004", "sgp0001", and "sgp0002") still increases with tile sizes larger than 150 pixels, because the image background region of those images is almost completely flat.

Choosing an optimal image-specific tile size could only enhance the average compression ratio c_f of the ICSSM 1 algorithm on the test image set from 3.37 to 3.42. Choosing a fixed optimal tile size for each bit depth group, 16-bit-per-pixel images and for 32-bit-per-pixel images, would lead to almost no enhancement.

For reliable segmentation, each tile should include a significant amount of background region; thus, the tile size should be larger than the diameter of the largest object – otherwise the segmentation, based on the statistical distribution of pixel intensity, will fail. Therefore, as a lower bound for the tile size, the diameter of the largest object can be used. This can be done manually, for example when integrating the segmentation, modelling and region-adaptive bit allocation based compression method into the data reduction and data archiving pipeline of a certain instrument. Other possibilities are to determine the object diameter using transform-based methods or to calculate the diameter of the point-spread function from the geometrical and optical properties of telescope and sensor device.

6.3.4 Summary

Comparing the results on the whole set of 16- and 32-bit-per-pixel test images, with an average compression ratio of 3.37 (including header influence), the proof-of-concept algorithm ICSSM 1 could achieve a performance that is comparable to the best standardised FITSIO Rice compression method. It was only outperformed by four dedicated lossless wavelet-based MR/1 methods, which reached average compression ratios of 3.79 up to 3.92, depending on the tile size.

On 32-bit-per-pixel images, the average compression ratio that ICSSM 1 could achieve for a tile size t of 100 pixels was 5.93. This exceeds the average compression ratio 4.69 obtained with the existing methods on 32-bit-per-pixel images. Comparing images of this bit depth only, the compression ratio of 5.93 that the ICSSM 1 method could achieve is even closer to the values of 6.42 and 6.40 – the average compression ratio of the best performing MR/1 methods (CDF and Int. (4.2) wavelet) – than to the average (4.69) of existing and tested methods.

One major advantage of the newly developed method over the lossless wavelet based MR/1 is that no pyramidal data representation has to be generated through iterative steps, as is required by the wavelet-based methods.

Even with the limited encoding methods which were used in ICSSM 1, the influence of the tile size remains limited. The choice of the optimal tile size for an image could only enhance the average compression ratio c_f of the ICSSM method on the test image set from 3.37 to 3.42. Therefore, a fixed tile size can be used with ICSSM 1.

6.4 Evaluation of ICSSM 2

With the ICSSM 2 variant, only a single encoder is applied to the data generated through segmentation to keep the encoding step as simple as possible. The LZ77 variant called Deflate, which is used in the well known Zip compression tool, was chosen.

Again, as in the previous section, first the implementation and testing of the ICSSM 2 method is described here. Then, the measured results on the test image set are presented and analysed in detail.

6.4.1 Implementation

The image segmentation software, which was used for ICSSM 1, was modified to conduct the experiments using external compression for all data components for ICSSM 2. Segmentation is performed automatically without user interaction. In comparison to ICSSM 1, the segmentation step in ICSSM 2 is modified to limit the noise range to eight bits. In addition to enhancing the segmentation, the eight-bit valued noise stream can be efficiently processed with the external tools for data compression, which assume that each input data item is stored in a byte. The Matlab software prototype saves into files the binary data streams for each image data component generated through segmentation. For each image and tile size measured, one folder is generated, containing the image data components. The software then invokes the application of external Zip compression on this folder, leading to a single compressed output file that contains all the image data from the original image. As the external Zip compression includes some additional information such as file names and folder location in the compressed file, a small compression performance enhancement might be possible when removing such information and storing the data directly in a FITS file.

The decoding algorithm for the data was implemented as a Matlab script, in order to verify the losslessness experimentally. The decompression script reverses the encoding. First, it decompresses the data using the external Zip tool. Then, the segmentation is reversed in order to restore the original data.

For all test images, the decompressed image data was compared to the original image data in order to ensure the losslessness of the whole procedure. No difference was found between the original data and the uncompressed data recovered after compression.

The script also allows decompressing only the background level data and the region-of-stellar-objects values without adding the extracted image background noise. This is shown as an example in Figure 6.2. For fast preview and quick inspection, a transmission of the shown image parts can be sufficient. In such a case, appropriate interpolation methods, such as linear interpolation, should be applied to the background level data, instead of the box-like approach displayed to avoid visual artefacts at the tile borders.

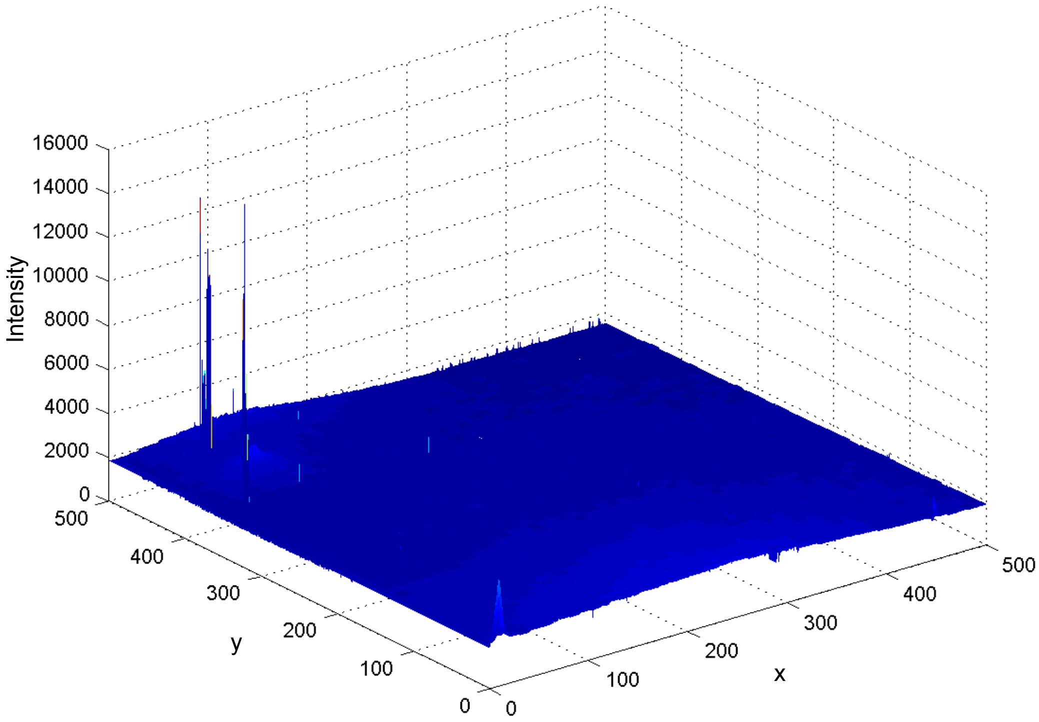


Figure 6.2: Partial decompression of the image “com0001” without adding the background noise. As the background noise component typically consumes about 66% of the compressed image data, decompression without noise may be used for quick inspection and preview purposes.

It has to be kept in mind, as with all lossy compression methods, that it cannot be guaranteed that no relevant information is disregarded by this partial decompression process. Removing the component treated as noise turns the ICSSM algorithm into a lossy compression algorithm. With the ICSSM algorithm and the preview without noise, faint sources will be disregarded, for example. Furthermore, details in very bright image areas might be omitted (compare to the "bright object" case in Section 5.1.5). In order to generate the preview image, only 16 percent of the original data were required with the 16-bit-per-pixel image "com0001". With 32-bit-per-pixel images, comparable images could be generated with less than 10 percent of the data.

6.4.2 Results

The measured results from compressing the test image set using the ICSSM 2 algorithm, for a range of tile sizes, are listed in Table 6.5.

Table 6.5: Compression ratios c_f measured for the ICSSM 2 method. The last column (symbol $\overline{c}$) denotes the mean compression ratio achieved for the corresponding image, the last row (denoted set $\overline{c}$) gives the mean compression ratio achieved for a certain tile size.

ICSSM 2

	Tile size								
Image	5	10	15	20	30	50	100	150	$\overline{c}$
com0001	1.97	2.06	2.07	2.06	2.04	2.01	1.92	1.88	2.00
for0001	2.55	2.69	2.74	2.77	2.78	2.81	2.83	2.86	2.75
for0002	2.96	3.22	3.33	3.37	3.44	3.45	3.51	3.55	3.35
gal0001	2.48	2.65	2.71	2.73	2.74	2.74	2.75	2.75	2.69
gal0002	2.09	2.20	2.24	2.24	2.25	2.26	2.26	2.27	2.23
gal0003	5.82	6.26	6.43	6.51	6.58	6.70	6.82	6.86	6.50
gal0004	5.86	6.29	6.45	6.55	6.66	6.75	6.97	7.03	6.57
ngc0001	1.69	1.77	1.81	1.78	1.81	1.82	1.82	1.85	1.80
ngc0002	2.27	2.41	2.45	2.48	2.50	2.51	2.56	2.59	2.47
sgp0001	5.06	5.40	5.60	5.52	5.72	5.84	5.93	6.01	5.63
sgp0002	5.30	5.70	5.94	5.93	6.13	6.28	6.51	6.58	6.05
tuc0003	2.35	2.51	2.56	2.58	2.59	2.62	2.66	2.66	2.57
tuc0004	2.45	2.62	2.66	2.69	2.69	2.71	2.74	2.80	2.67
Set $\overline{c}$	3.30	3.52	3.62	3.63	3.69	3.73	3.79	3.82	

The measurements show that when Zip general-purpose compression was applied, the size-efficiency was enhanced on each of the images compared to ICSSM 1. For example, using a tile size parameter t of 10 pixels, the compression ratio of ICSSM 2 reached 3.52, while ICSSM 1 could achieve an 8.31 percent lower compression ratio of 3.25 (on average over the entire test image set).

The proof-of-concept ICSSM 1 algorithm already produced the same average compression ratio as the second best existing measured method, FITSIO Rice compression; the improved results obtained with ICSSM 2 exceed those of the best standardised FITS compression methods and are close to those of the best available wavelet-based methods.

The measurements conducted with ICSSM 2 proved that segmentation-based region-adaptive compression can give results as good as wavelet-based methods. Only the lifting scheme integer CDF wavelet and the integer (4,2) wavelet methods, which both reached a compression ratio of 3.92, exceed the best average results of ICSSM 2 which is a compression ratio c_f of 3.82 a bit.

With ICSSM 2, the optimal tile size parameter (t set to 150 pixels) (see Table 6.5), observed for the average compression ratio over the test image set, was much higher than with ICSSM 1 (t set to 30 pixels) (see Table 6.4). With a larger tile size parameter, typically the noise data increased with ICSSM 1. The introduction of an upper noise limit in ICSSM 2 prevents the growth of the noise component beyond 8 bits-per-pixel. This limit on the growth of the noise component is achieved at the cost of a growth of the size of the region-of-stellar-objects data.

That ICSSM 2 achieved higher compression ratios than ICSSM 1 has to be attributed mainly to the enhanced encoder for region-of-stellar-objects data components and to the newly introduced compression of the region demarcation information. With ICSSM 1, the background level values were encoded verbatim into binary numbers, using no compression. This encoding does not exploit the correlation of background level estimates for adjacent tiles. This may be one cause why the compression performance of the ICSSM 1 algorithm still increases with tile size parameters larger than 50 pixels (the evaluation results are given in Table 6.4). ICSSM 2 showed a similar behaviour, while the performance of the Zip compression on this particular data component within the archive cannot be measured exactly and linked to the data distribution[3]. Smaller tile sizes still may lead to higher compression ratios in the case of images with strongly-varying background region intensities, for example "com0001".

Table 6.6: Compression ratios of the FITSIO Zip (c_Z) method in comparison with the ICSSM 2 results (c_{I2}). The last row gives the quotient $\frac{c_Z}{c_{I2}}$. For layout reasons, the test images are listed as a numbered sequence corresponding to an alphabetical ordering of file names.

Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	$\bar{\mathbf{c}}$
F-IO Zip	1.14	1.51	1.75	1.43	1.18	4.48	4.50	1.02	1.35	2.96	3.30	1.43	1.49	2.12
ICSSM 2 t=150	1.88	2.86	3.55	2.75	2.27	6.86	7.03	1.85	2.59	6.01	6.58	2.66	2.80	3.82
$\frac{c_Z}{c_{I2}}$	1.65	1.89	2.03	1.92	1.92	1.53	1.56	1.81	1.92	2.03	1.99	1.86	1.88	1.80

Finally, comparison of ICSSM 2 to Zip compression also showed that segmentation plus external compression methods are more effective than the application of Zip compression, which is often used for external file compression. The results from the application of Zip compression using the FITS Zip method are given in Table 6.6, in comparison to the average compression ratio that ICSSM 2 achieved on the test image set (as detailed in Table 6.5). With a tile size parameter t of 150 pixels, ICSSM 2 achieved results that are on average 80 percent better than those of the FITSIO Zip method. In the worst case, ICSSM 2 was 1.53 times as effective as the FITSIO Zip method; in the best cases ICSSM 2 was 2.03 times as effective as the FITSIO Zip method alone. This means that the application of the segmentation step prior to the Zip encoding step led to a performance increase of 1.53 up to 2.03 compared to the application of the FITSIO Zip compression alone.

6.4.3 Summary

In the evaluation reported in the preceding section, using a sample implementation of the ICSSM 2 method, the achievable compression ratio was

[3] The compression ratio that Zip achieves on a certain data component within an archive of several components may be influenced by the characteristics of the other data, which is as well included in the same archive.

measured and the losslessness of the method was ensured. With an average compression ratio c_f on the test image set of up to 3.82, ICSSM 2 yielded a compression ratio comparable to the best existing methods under evaluation.

The performance of ICSSM 2 was at par with the best four lossless wavelet based MR/1 methods for compressing astronomical images, which reach average compression ratios of 3.26 up to 3.92. Hence, ICSSM 2 proved that segmentation-based compression leads to results as good as wavelet-based methods for stellar-field images, as it allows region-adaptive allocation of bits to pixel data.

Finally, a particularly interesting result which can be shown with ICSSM 2 is that segmentation plus the general-purpose coding method lead to up to more than twice the compression ratio achieved with the external Zip compression method alone. This proves that the segmentation is a valid pre-processing step for stellar-field image data as it allows region-adaptive allocation of bits to pixel data.

6.5 Evaluation of ICSSM 3

The evaluation of ICSSM 2 discussed in the last section showed that high compression ratios, comparable even to the best existing wavelet-based methods, are possible with the segmentation-based region-adaptive approach.

With ICSSM 3, the focus is on maximising the compression ratio of the individual data components. Therefore, further possibilities to enhance the compression ratio of the ICSSM method were identified and investigated in Chapter 5 to determine the most promising compression approaches for each data component. Fitting compression methods for the individual data component were chosen (see Chapter 5.3).

6.5.1 Implementation

To measure the size-efficiency of ICSSM 3, the image segmentation software, which was developed and used for previous experiments with ICSSM 2, was adapted. As with ICSSM 2, binary data was exported and encoded using the external data compression tools, which were chosen in the experiments described in Sections 5.3.1, 5.3.2, 5.3.3, and 5.3.4. Similar to the ICSSM 2 prototype, a script invokes the external tools which compact each data component. As the segmentation step itself is lossless, and the external tools used do not introduce any change to the data, the whole ICSSM 3 method is lossless. The losslessness of the segmentation and compression was verified as with ICSSM 2.

6.5.2 Results

The investigation to determine component-specific methods for compaction (described in Section 5.3) suggested that a performance increase of ICSSM 3 compared to ICSSM 2 would be possible if the performance on the complete test image set behaved as in the experiments reported in Sections 5.3.1, 5.3.2, 5.3.3, and 5.3.4.

Table 6.7: Compression ratios c_f measured for the ICSSM 3 method. The last column (symbol $\overline{c}$) denotes the mean compression ratio achieved for the corresponding image, the last row (denoted set $\overline{c}$) gives the mean compression ratio achieved for a certain tile size.

ICSSM 3

	Tile size								
Image	5	10	15	20	30	50	100	150	$\overline{c}$
com0001	1.96	1.97	1.89	2.03	2.01	1.99	1.95	1.94	1.97
for0001	2.96	3.15	3.20	3.23	3.25	3.24	3.23	3.26	3.19
for0002	3.62	4.04	4.17	4.23	4.29	4.32	4.31	4.33	4.16
gal0001	2.65	2.77	2.81	2.82	2.83	2.83	2.83	2.83	2.80
gal0002	2.13	2.21	2.24	2.24	2.25	2.25	2.25	2.26	2.23
gal0003	6.73	7.09	7.23	7.30	7.38	7.50	7.64	7.69	7.32
gal0004	6.76	7.12	7.26	7.35	7.46	7.55	7.77	7.87	7.39
ngc0001	1.82	1.89	1.92	1.91	1.93	1.94	1.93	1.95	1.91
ngc0002	2.65	2.82	2.87	2.89	2.91	2.90	2.92	2.94	2.86
sgp0001	5.94	6.26	6.45	6.45	6.51	6.64	6.73	6.88	6.48
sgp0002	6.54	6.90	7.08	7.10	7.20	7.28	7.27	7.38	7.09
tuc0003	2.81	3.03	3.10	3.12	3.13	3.15	3.14	3.13	3.07
tuc0004	2.94	3.20	3.27	3.30	3.31	3.31	3.31	3.32	3.25
Set $\overline{c}$	3.81	4.03	4.11	4.15	4.19	4.22	4.25	4.29	

The listing of the experimental results given in Table 6.7 shows that the goal could be reached with the ICSSM 3 method. The best average result that ICSSM 3 achieved, with a fixed tile size for all images, was about 4.29. Under the same conditions, ICSSM 2 attained only a compression ratio of 3.82. This means that, compared to ICSSM 2, ICSSM 3 achieved an average performance increase by a factor of 1.12 on the test image set.

As the measured compression ratio for 32-bit-per-pixel images increased with larger tile size, the optimal choice for these images is a tile size value of 150 pixels. For 16-bit-per-pixel images the average performance reached its limit (compression ratio of 2.88) for t = 50. Given that for the 16-bit-per-pixel images the compression ratio remains nearly constant with t ranging from 30 up to 150, different choices of t for 32- and 16-bit-per-pixel images are not necessary to reach the optimal average compression ratio.

Comparing the average compression ratio on all images in the test set, with an optimal or near optimal choice of the tile size, the newly developed method was approximately 11% more efficient than the best existing method. A comparison of all ICSSM methods and the existing methods is given in the next section.

6.5.3 Summary

The size efficiency increase of ICSSM 3 compared to ICSSM 2 was approximately 12%. ICSSM 3 outperformed, in terms of average compression ratio on the test image set, all existing methods reported in research papers and tested in Section 6.2. A final comparison of the average compression ratio of available ready-to-use lossless compression tools to ICSSM 1, ICSSM 2, and ICSSM 3 is given in the next section.

6.6 Comparison of ICSSM and Existing Compression Methods

A final overall comparison of existing compression methods reported in research papers (discussed in Chapter 3) and the novel algorithms proposed in this thesis, is given in this section.

This comparison consists of six parts. First, the average compression ratio of each method on the whole test image set is discussed. Then, the influence of the image bit depth on compression performance is discussed. Additionally, each of the bit depth categories, 16- and 32-bit-per-pixel images, is analysed separately; within each category, the performance of the different methods is compared. Finally, after analysing the number of best results per compression method, an overall summary is given and practical implications of the findings are discussed.

As discussed in Section 6.1.1, the file-based compression ratio c_f, including the influence of the FITS file header on the data, was used for the comparison of performance measurements of the ICSSM algorithms and the existing methods.

6.6.1 Compression Ratio on the Whole Test Image Set

Comparing the average compression ratios of ICSSM algorithms and existing methods on the whole test image set, Figure 6.3 clearly shows that the average compression ratio of ICSSM 3 exceeded those of existing methods. The method of Sabbey et al. (1998), which could not be made available for testing, achieves an average compression ratio of 3.18[4]. Thus, ICSSM 3 also outperformed this method, on average on all images in the test set.

Even the ICSSM 2 method, which did not apply encoder specially chosen for each data component, led to average results on all images in the test set at par with the best methods under evaluation, which are based on a lifting scheme. These MR/1 lifting scheme methods are labelled 'ML' in Figure 6.3. ICSSM 1, the proof-of-concept variant of ICSSM, gave average results on all images in the test set at par with the existing Rice-based compression method included in the FITSIO library.

[4] As this result was not measured in the experiments reported in this thesis, it is only given for information and not included in Figure 6.3.

These findings validate the hypothesis that a data-specific compression method based on image segmentation can exceed the results of wavelet-based methods, in the case of astronomical images.

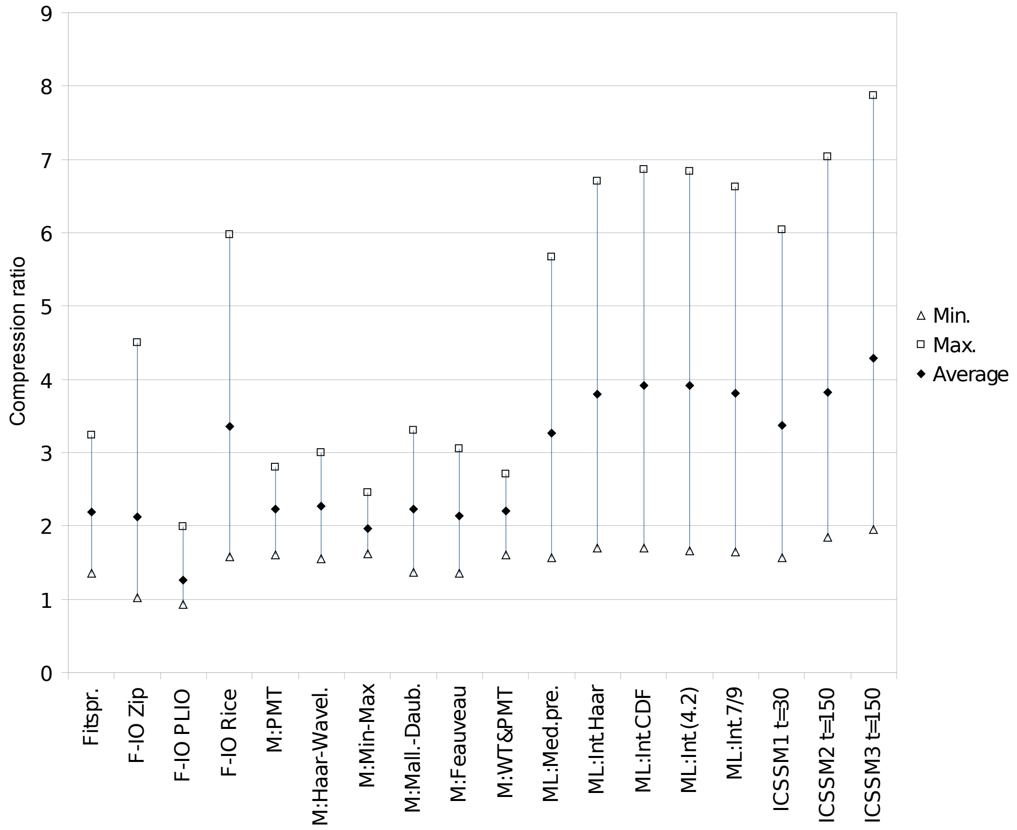


Figure 6.3: Comparison of the average, minimum, and maximum compression ratios of existing ready-to-use lossless compression tools, and ICSSM on all test images.

6.6.2 16- Versus 32-bit-per-pixel Images

In Figure 6.3, great differences in the span between the minimum and maximum compression performance can be found. This effect is especially visible when comparing the MR/1 standard methods (labelled M in the figure) and the lifting scheme-based methods (labelled ML). The lifting scheme methods are capable to compress 32-bit-per-pixel images, where high compression ratios can be achieved, while these high results are missing with methods that only worked losslessly on 16-bit-per-pixel images. Therefore, in the following both data categories 16- and 32-bit-per-pixel images are looked at separately.

Figures 6.4 and 6.5 show that the compression achievable on 32-bit-per-pixel astronomical images is at least twice higher than on 16-bit-per-pixel images.

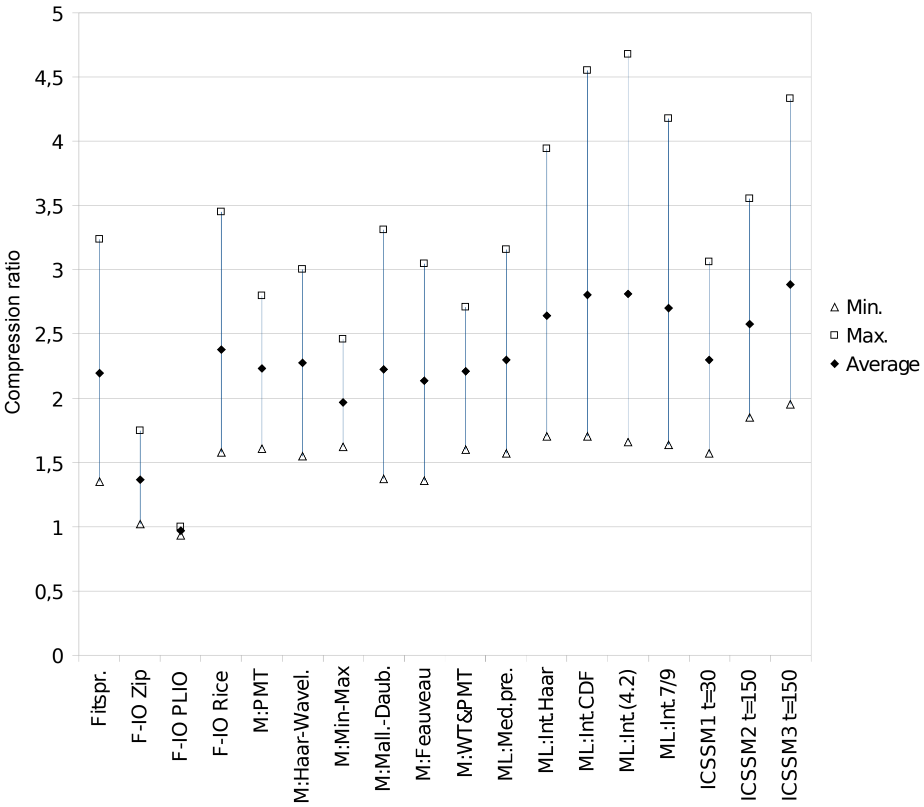


Figure 6.4: Comparison of the average, minimum, and maximum compression ratios of existing ready-to-use lossless compression tools, and ICSSM on 16-bit-per-pixel test images.

Efficient methods proved capable of reducing the size of 16-bit-per-pixel images typically by a factor of three, while typically a factor of six up to nearly eight could be achieved on the 32-bit-per-pixel images of the test set.

6.6.3 Results for 16-bit-per-pixel Images in Detail

For 16-bit-per-pixel images, a range of different methods led to almost the same average results. Ignoring PLIO, where no compression could be achieved, the average compression ratio ranged from 1.37 up to 2.88. The performance of the best method given in Figure 6.4 reaches nearly the same as the results reported for the method of Sabbey (2.96) in (Sabbey 1999), which could not be verified experimentally.

On average, the best performing method, ICSSM 3 also slightly outperformed the best method from the evaluation of existing methods, the integer (4,2) lifting scheme method from the MR/1 package (which achieved a compression ratio of 2.81 on average). The best compression ratio on a single 16-bit-per-pixel image was produced by one of the integer lifting scheme methods. Comparing the minimum compression ratio, the ICSSM 3 method also achieved the best result.

6.6.4 Results for 32-bit-per-pixel Images in Detail

For 32-bit-per-pixel images (Figure 6.5), the ICSSM 3 algorithm achieved the best average compression ratio (7.46), followed by ICSSM 2 and several methods based on a lifting scheme. The gradual performance improvement, which results from the gradually better adaptation of the compression method to the data properties, stood out with the ICSSM methods.

As with the 16-bit-per-pixel images, the performance of ICSSM 3 method was followed by some of the lifting scheme methods. Again, as found for 16-bit-per-pixel images, the integer CDF and integer (4.2) method stood out in terms of yielding a high compression ratio. The standard FITSIO/Rice method gave a higher average compression ratio than the integer lifting scheme method with the lowest average compression ratio. Therefore, using a lifting scheme method does not always guarantee better compression than using a less-advanced entropy encoder-based method. As discussed earlier, the method of Sabbey (Sabbey 1999), which was not tested experimentally, has a published result (4.88) which is lower than that of the FITSIO/Rice method on 32-bit-per-pixel images.

6.6.5 Number of Best Results per Method

A comparison of the results for individual images showed that ICSSM 3 is the method with the highest number of best results (Table 6.8). This mainly results from its high performance on 32-bit-per-pixel images, although there are only four 32-bit-per-pixel images within the data set. It is followed by the method of Sabbey, which according to (Sabbey 1999) achieves five best results on 16-bit-per-pixel images.

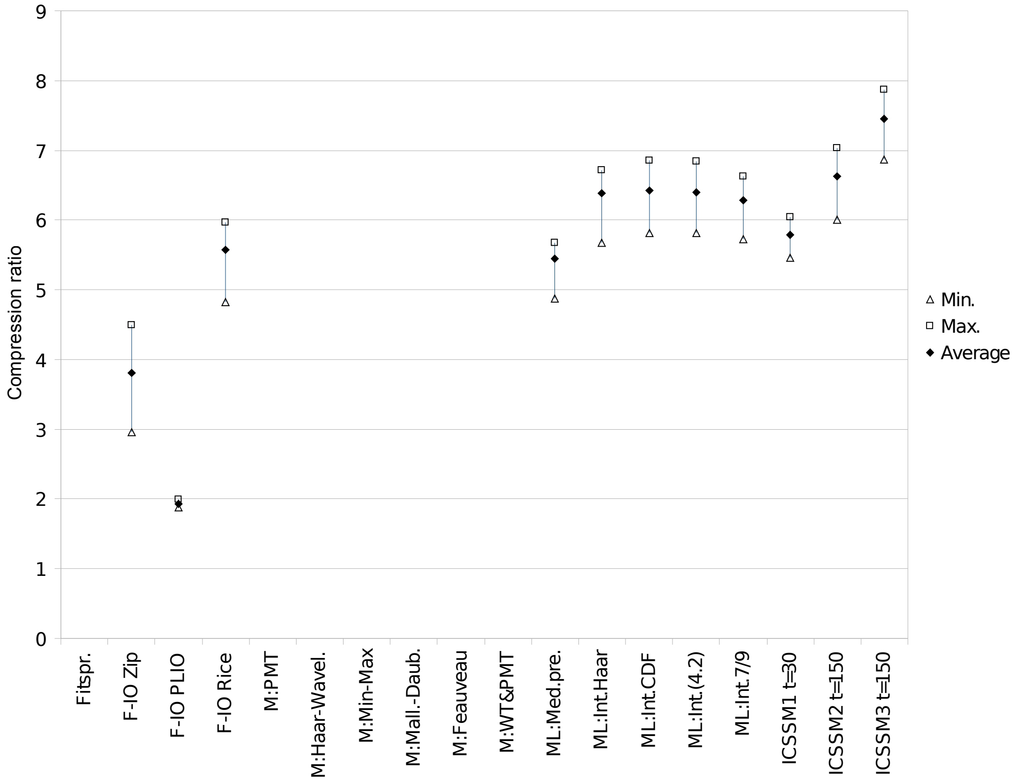


Figure 6.5: Comparison of the average, minimum, and maximum compression ratios of existing ready-to-use lossless compression tools, and ICSSM on 32-bit-per-pixel test images.

These results do only imply, at first glance, that either a predictive method or the newly developed segmentation-based method should be chosen if high compression performance is required. In many cases, the lifting scheme transform-based methods lead to the second or third best results. In two cases, the lifting scheme-based methods achieve the best results.

Table 6.8: Compression ratios c_f for state-of-the art methods for compressing astronomical images and the ICSSM algorithms. The ‘–’ symbol indicates that compression was lossy according to verification with Fitsdiff, the ‘◇’ symbol indicates that compression was not possible due to documented software limitations. The last column (symbol $\overline{c}$) denotes the mean compression ratio achieved on the whole image set for the given method. For layout reasons, the images are listed as a numbered sequence corresponding to an alphabetical ordering of file names.

Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	$\overline{c}$
Bit depth	16	16	16	16	16	32	32	16	16	32	32	16	16	
Hcompr.	–	–	–	2.78	–	◇	◇	–	–	◇	◇	–	–	2.78
Fitspr.	2.03	2.62	3.24	2.57	2.05	◇	◇	1.35	1.83	◇	◇	1.97	2.08	2.19
F-IO Zip	1.14	1.51	1.75	1.43	1.18	4.48	4.50	1.02	1.35	2.96	3.30	1.43	1.49	2.12
F-IO PLIO	0.97	1.00	1.00	0.96	0.95	1.99	1.99	0.93	0.96	1.88	1.88	0.97	0.97	1.27
F-IO Rice	1.92	2.75	3.45	2.40	2.00	5.97	5.91	1.58	2.17	4.83	5.58	2.52	2.61	3.36
Pred.+Rice	2.10	3.40	4.30	2.85	2.40	4.40	3.30	1.88	2.80	3.17	3.80	3.34	3.50	3.17
M:PMT	1.85	2.44	2.80	2.42	1.95	–	–	1.61	2.20	–	–	2.35	2.46	2.23
M:Math.Mor.	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇
M:Haar-Wavel.	1.92	2.54	3.00	2.44	2.01	–	–	1.55	2.16	–	–	2.35	2.50	2.27
M:Min-Max	1.92	2.33	2.46	1.85	1.64	–	–	1.62	1.88	–	–	1.99	2.03	1.97
M:Mall.-Daub.	1.87	2.53	3.31	2.42	2.00	–	–	1.37	1.85	–	–	2.27	2.41	2.23
M:Feauveau	1.85	2.38	3.05	2.39	1.98	–	–	1.36	1.82	–	–	2.15	2.28	2.14
M:WT&PMT	1.86	2.38	2.71	2.41	1.95	–	–	1.60	2.18	–	–	2.33	2.45	2.21
ML:Med.pre.	1.91	2.57	3.16	2.45	2.02	5.63	5.58	1.57	2.13	4.88	5.67	2.37	2.48	3.26
ML:Int.Haar	2.10	3.01	3.94	2.79	2.26	6.60	6.54	1.70	2.40	5.68	6.71	2.74	2.87	3.80
ML:Int.CDF	2.10	3.17	4.55	2.79	2.30	6.53	6.47	1.70	2.46	5.81	6.86	3.00	3.17	3.92
ML:Int.(4.2)	2.09	3.17	4.68	2.78	2.30	6.50	6.43	1.66	2.41	5.81	6.84	3.03	3.21	3.92
ML:Int.7/9	2.10	3.07	4.18	2.78	2.30	6.43	6.37	1.64	2.34	5.72	6.63	2.89	3.04	3.81
ICSSM 1 t=30	1.80	2.58	3.06	2.45	1.94	5.78	5.87	1.57	2.28	5.46	6.04	2.45	2.54	3.37
ICSSM 2 t=150	1.88	2.86	3.55	2.75	2.27	6.86	7.03	1.85	2.59	6.01	6.58	2.66	2.80	3.82
ICSSM 3 t=150	1.95	3.25	4.33	2.83	2.26	7.69	7.87	1.95	2.94	6.87	7.37	3.13	3.32	4.29

6.6.6 Implications

These results have the following implications for practical applications. Compression methods for stellar-field images may lead to a lossless compaction by a factor of three or even up to nearly eight, depending on the bit depth of the original image.

Therefore, for archiving data, compression methods may indeed save costs for storage media or data transfer time, especially if a large part of the raw images is encoded with 32 bits per pixel.

The best available and standardized method, implemented in the FITSIO file reading and writing library, is the FITSIO/Rice compression method. It can be used without additional effort by any astronomer exporting data in FITS format. Using this method, typically a 16-bit-per-pixel image may be compressed by a factor of 2 up to 2.75, while a 32-bit-per-pixel image can be compacted to about one fifth up to nearly one sixth of its original size.

For higher compression ratios on 16-bit-per-pixel images as well as on 32-bit-per-pixel images, the novel ICSSM 2 or ICSSM 3 algorithms should be used. On 16-bit-per-pixel images, the performance gain of ICSSM 3 compared to the FITSIO/Rice method is a factor of 1.21. This means that approximately 21% more image data can be stored.

For applications where high volumes of image data have to be stored, the novel ICSSM 3 algorithm should also be considered, especially if a large number of 32-bit-per-pixel images are included in the data. There, the average performance gain of ICSSM 3 relative to the FITSIO/Rice method of a factor of 1.28 is worth the effort. 28 percent more image data can be stored with the novel method. If computational time is not important, and ready availability of an implementation of the method is required, then the lifting scheme methods from the MR/1 package, especially the integer CDF wavelet and the integer (4.2) wavelet, can be applied.

6.7 Summary

The experimental comparison presented in this chapter showed what compression ratios can be expected from existing astronomy-specific lossless image compression methods.

On the test image set, reasonable existing methods[5] were shown to lead to

[5] The PLIO method is ignored here due to its low performance.

average compression ratios from 1.97 (a Min-Max transform based method) up to 3.92 (two of the lifting scheme integer wavelet based methods). For the examined existing methods, the average results of existing methods on 16-bit-per-pixel images ranged from 1.37 (FITSIO Zip) up to 2.81 (the integer (4.2) wavelet transform based method). The average compression ratios of all methods on 32-bit-per-pixel images ranged from 3.81 (FITSIO Zip) up to 6.42 (the lifting scheme integer CDF wavelet transform based method), ignoring the lower performing PLIO method in both cases.

This big difference between the higher bit depth images compared to the lower bit depth images shows the potential of data compression methods to reduce data storage problems in astronomy. As current analogue-to-digital converters can exceed a dynamic range of 16 bits, 32-bit images are a quite common data format today. A data storage space reduction by a factor of 4 and above could significantly reduce the costs for data storage and transmission. The bit depth of astronomical images is still increasing.

Currently, there are ongoing discussions about standardizing 64 bit-per-pixel images. Even higher compression ratios might thus be possible in the future on those high bit depth images.

With ICSSM 1 and ICSSM 2, no highly adapted encoders where applied for encoding the data streams after segmentation. Even without using highly adapted encoders, the model-based segmentation and region-adaptive compression methods ICSSM 1 and ICSSM 2 achieved compression ratios which are at par with contemporary methods. The average compression ratio of 3.37 on the test image set obtained with the proof-of-concept ICSSM 1 algorithm is at par with the Rice compression ratio of 3.36; Rice compression is the best freely available and standardized method. With an average compression ratio of 3.82 (on the whole test image set), the ICSSM 2 algorithm, which is intended for practical use, almost reached the compression ratio of the best available method, which is the lifting scheme integer CDF wavelet method (compression ratio of 3.92).

As the lifting scheme integer CDF wavelet method has not been standardized and included in the FITSIO library, the ICSSM 2, and ICSSM 3 algorithms are valid candidates for an enhanced FITS image compression standard. The performance evaluation results measured on the test image set showed that the optimized ICSSM 3 algorithm, developed in this research, in average on the whole sample image set outperformed all existing methods which were evaluated.

With an average compression ratio of 4.29 on the whole test image set, the ICSSM 3 algorithm reached an average which is nearly 1.1 times the average compression ratio of the best existing method included in the comparison. Looking at the bit depth categories 16-bit-per-pixel images (2.88) and 32-bit-per-pixel images (7.45) separately, in both categories ICSSM 3 achieved better results than existing evaluated methods. The performance evaluation of the ICSSM 3 algorithm showed that a method adapted to astronomical image data can outperform even the modern methods based on wavelet transform and lifting scheme.

The next chapter summarises the findings of the research reported in this thesis. It recapitulates the performance and limitations of existing compression methods and the effectiveness and drawbacks of the novel approach adapted to stellar-field images which has been developed in this investigation. Conclusions are drawn and options for further work are presented.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

Astronomy faces a high increase of data amounts, which exceeds the growth of computer data storage and transmission capacity. Therefore, the aim of the research presented in this thesis was to answer the following main question related to the specific field of astronomical image compression:

Can lossless compression, which is based on image modelling, segmentation and region-adaptive bit-allocation, be applied effectively (in terms of compression ratio) to astronomical stellar-field images?

A subsidiary question also addressed in the research is: What contribution may existing data compression methods make to reduce the size of astronomical image data?

While both research questions are timely and have practical relevance, especially for the astronomical community, which is in search of solutions to handle its data archival, storage and transmission requirements, the second research question also has implications in a broader context. In effect, the second question can be said to address the general issue of whether lossless compression methods, which are adapted to the characteristics of the primary data, may exceed the compression performance of less adapted methods.

Research, which was conducted worldwide mainly during the last ten years, showed that great performance increases are possible in the case of lossy compression regarding achievable compression-ratios at comparable visual or audible quality. An example in the area of lossy image compression is JPEG compression (Wallace 1991, Strutz 2005); similarly, MP3 and the different versions of MPEG compression are also common examples for audio and video content respectively.

These methods typically achieve a lot of their performance increase by removing parts of the data which are less noticeable, deemed irrelevant or unimportant. This approach is not possible in the case of scientific-grade astronomical data, due to the requirement of preserving the original data. But, further research has also shown that lossless methods adapted to the data may also increase performance compared to less adapted methods, within the limits of the data reduction ratio achievable by lossless compression.

Whether this is also possible in the case of astronomical stellar-field images – a very common and important data type in astronomy – is the issue addressed by the main research question of the investigation reported in this thesis.

To answer the main research question, first, special properties of astronomical images were discussed and an overview of different types of astronomical images was given. It was found that stellar-field images are an important category of astronomical images. To identify options for data redundancy reduction, the main features of stellar-field images were determined. These are: the relatively small and localised image areas with high dynamic range, the large areas containing background noise, and the spatial segregation of these two image components.

7.1.1 Performance of Existing Compression Methods

In order to answer the question of the possible contribution of existing compression methods to reduce the size of astronomical image data, methods for the compression of astronomical images were reviewed and classified into lossy, hybrid, and lossless methods.

The thesis presented a survey of existing compression algorithms for astronomical images and showed that the existing methods often follow the typical design of a transform-based method or in one case a typical prediction-based approach. The survey also showed that the majority of existing methods does not employ region-adaptive processing of the data or even segmentation. Many of the existing methods are lossy; hence, they can be used for fast previewing and quick selection of images, but not for the storage or transfer of scientific grade data. Hence, objective number one, to perform a thorough literature review of existing compression methods for astronomical images, was achieved.

Existing methods which are capable to work in a lossless mode were assessed regarding their compression size-efficiency through experiments using a standard astronomical test data set.

The evaluation showed that although only few of the existing compression methods can be used for strictly lossless compression, these methods can yield a significant reduction of the data size. Implementations of lossless methods exist, but available implementations include mainly methods that follow the typical design of transform-based methods like Hcompress, Fitspress and the lossless wavelet based MR/1 methods, or methods like Zip and Rice compression. The existing compression methods, which were – on average – found to be most effective on the test set used in this thesis, are the Rice compression method (compression ratio of 3.36) and various MR/1 lifting schemes (average compression ratios ranging from 3.26 to 3.92). They offer data reduction which may reduce data transmission times and storage requirements significantly. For the increasingly common 32-bit-per-pixel images, compression ratios are much higher than for 16-bit-per-pixel images. The average results of existing methods on 16-bit-per-pixel images range from 1.37 (FITSIO Zip) up to 2.81 (the integer (4.2) wavelet transform based method). The average compression ratios of all methods on 32-bit-per-pixel images range from 3.81 (FITSIO Zip) up to 6.42 (the lifting scheme integer CDF wavelet transform based method), ignoring the PLIO method in both cases. Therefore, also objective number two, to perform an evaluation of existing lossless methods which could be made available, was achieved.

The question, whether such methods can be included in the pipelines of archival systems, depends on several factors beyond this research, such as availability of sufficient computational power, further reduction of the cost of processing and storage hardware, system architecture and other project-specific circumstances.

However, most existing methods do not explicitly exploit the special characteristics of astronomical images in order to provide a dedicated lossless method for astronomical image storage and transmission. Most existing methods typically follow the common design of transform-based compression methods, which have conquered image compression applications in recent years. Often, the lossless mode of operation has been implemented as an additional function to lossy compression techniques. Few of the existing methods adapt to the distinct properties of stellar-field images.

To the knowledge of the author, no lossless method, which is based on region-segmentation and image modelling for exploiting the redundancy in the distribution of pixel intensity levels in stellar-field images, was published before although some region-based lossy techniques were proposed. The only astronomy-data-specific method, called PLIO, failed completely to compress the data in many of the experiments conducted in this research.

7.1.2 Effectiveness of Lossless Compression Adapted to Stellar-Field Images

In order to determine whether a dedicated compression method adapted to the properties of astronomical stellar fields may boost the compression ratio compared to existing methods, this work introduced a set of novel compression algorithms, called image compression by segmentation and signal modelling (ICSSM) algorithms.

As a foundation for the development of the ICSSM algorithms, a model describing properties of astronomical stellar-field data was established. Because the astronomical image data results from a combination of both imaging device properties and stellar-field data properties, the data model includes properties of both imaging devices and stellar fields. With this data model, objective number three – to find a suitable model that describes the properties of astronomical images – was achieved.

A set of three lossless region-adaptive compression algorithms based on image segmentation and modelling was developed. While for the proof-of-concept algorithm ICSSM 1 achievable compression ratios were estimated, the performance of the more advanced ICSSM 2 and ICSSM 3 algorithms was measured using software implementations. Hence, also objective number four – to develop region-adaptive compression algorithms based on image segmentation and modelling – was achieved.

Finally, to achieve objective five – to evaluate the compression ratios of the algorithms developed – as well, the compression ratios achieveable with the ICSSM algorithms developed in this research were measured and compared to existing methods. The experiment-based evaluation on a set of standard test images showed that a significant compaction can be achieved by segmentation and region-adaptive bit allocation anchored on a domain-specific image data model. Furthermore, the proposed algorithms achieved compression ratios comparable to – and often better than – the existing methods which were investigated.

First, the investigation showed the feasibility of compression based on image segmentation and signal modelling, using a straightforward method called ICSSM 1. It demonstrated that a simple segmentation step, coupled to region-adaptive allocation of bits encoding pixels, may lead to compression ratios comparable to existing lossless methods for astronomical images. With an average compression ratio of 3.37 on the test image set, the ICSSM 1 algorithm was for example at par with the Rice compression ratio of 3.36. Successive enhancements (ICSSM 2 and ICSSM 3) of the initial ICSSM 1 algorithm were evaluated. With an average result on the test set of 3.82, the ICSSM 2 algorithm reached almost the compression ratio of the best available existing method, the lifting scheme integer CDF wavelet (3.92).

The evaluation of ICSSM 3 showed – with an average compression ratio of 4.29 on the test image set – that it can outperform existing methods by about 10%, in terms of size reduction. On 32-bit-per-pixel images, the ICSSM 3 algorithm achieved a maximum compression ratio of 7.87, while the maximum of existing methods was 6.86 (the integer CDF wavelet transform based method).

The performance of ICSSM 3 shows that an adapted compression algorithm may outperform less adapted methods, at the cost of its adaptation to the distinct properties of the primary data. This result boosts the expectation that well adapted compression methods could also improve compression performance in other areas.

Examples of such areas are the storage of medical data, where similar adaptation approaches have been examined by some researchers, or the storage of telemetric and measurement data. An example of the latter is the compression of test bed data, such as engine test beds and data from mechanical engineering dynamometers, where higher clock speeds and finer resolutions lead to higher correlated data throughout the largest part of the measurement. Typically, within engine test data, for example, interesting data with particular properties is found only in a small time frame before or during an incident.

This research has answered the question regarding the potential size reduction and size reduction enhancement by an algorithm well-adapted to the properties of stellar-field images.

However, for a practical application of both existing methods and the newly developed algorithms, the computational complexity of the compression method should be evaluated. ICSSM 3 does not rely on potentially time-consuming transforms. The chances are high that it would additionally outperform transform-based methods, in terms of lower computational cost. Given the different types of encoders applied, however, it may fall behind the less size-efficient FITSIO Rice method. Therefore, ICSSM 3 may provide an algorithm for astronomical image compression in cases where a very high compression ratio is needed, and computational complexity is not a key consideration.

Regarding application classes for compression methods under evaluation, this thesis suggests for applications where high compression ratios are required, the application of the novel ICSSM 3 algorithm, especially if a large number of 32-bit-per-pixel images are included in the data. In cases where the ready availability of an implementation of the method is needed, then the lifting scheme methods from the MR/1 package, especially the integer CDF wavelet and the integer (4.2) wavelet, can be applied.

No computational time analysis was performed in this study because due to the long exposure times of astronomical images, this aspect is less critical for the archival of primary data than the compression ratio. For this reason the results of this study are limited to a classification regarding the size-efficiency of the methods. Further work may concentrate on adding information on the computational complexity of the method, and possibly the achievable run-time performance of the different methods on certain types of hardware or software implementations.

As some sort of fundamental limitation of the outcomes of this research, it cannot be proven that there are no better encoders, or no further or alternative decorrelation or pre-processing methods that could lead to higher compression ratios on the data under investigation.

It has to be stated further that the results achieved in this thesis are limited with respect to the selected set of compression algorithms that are applied after the segmentation step. Typically, a limited set of four methods was tested. These options were considered to provide an appropriate choice of encoders.

It cannot be proven that there are no be better choices for the encoders than those applied with ICSSM 3. Therefore, based on the segmentation-based approach, further versions of the algorithms may be developed in the future. For example, the region demarcation information could be compressed more effectively with the JBIG 2 method, that is claimed to be “the ultimate bi-level image coding standard” (Ono et al. 2000). JBIG 2 compresses typically 3 up to 5 times smaller than Fax Group 4 compression, and 2 to 4 times smaller than the original JBIG, which is used with ICSSM 3 (JBIG Committee 1999). JBIG 2 was adopted as an international standard (ISO/IEC 2001). Also, predefined codes may provide an alternative to arithmetic coding for the background noise information. Rice or Golomb-Rice codes (Rice and Plaunt 1971, Rice 1979, Rice 1983, Rice 1991, Golomb 1966) were not investigated within the new segmentation-based algorithm, as the pixel value distribution typically does not follow a geometric distribution. Predefined codes may be another option to adapt the developed algorithms to even lower computational requirements. Special predefined codes for geometric distributions exist as well, but with the key requirement of high compression ratios, these methods were not examined in this work.

Another limitation, which however does not affect the practical usability of the method developed in this research, is that no automatic detection of the tile size parameter is performed. The evaluation suggests suitable tile size choices for the different ICSSM algorithms. As the choice of an optimal tile size parameter is a potentially complex optimisation problem, it cannot be answered generically. The choice of the tile size parameter involves some trade-off between the different data components generated.

Finally, the method is – due to the special requirements it was developed for – limited to the particular image type of stellar field images. As the study presented here does not examine compression performance on astronomical interferograms, for example, it is possible that different methods may be better for other image types.

The limitations discussed here can be tackled by further work. Suggestions for future research are discussed in Section 7.2.

7.2 Further Work

The research presented in this thesis showed that a close coupling of the compression algorithm to data features can lead to an effective compression in terms of compression ratio. ICSSM 3 can be considered as an optimised version of the ICSSM algorithms, with certain restrictions. For example, the possibility to use other existing astronomy-specific compression methods, such as the wavelet-based method of (Press 1992, White 1992, White and Percival 1994, Starck et al. 1995, Louys, Starck and Murtagh 1999) in combination with the segmentation, has not been examined yet. The wavelet-based method could be applied to the regions of stellar objects to further enhance the compression-ratio on this image component.

For a wide acceptance of the work in the astronomical community, an implementation of the work into a standard software package and the integration into the FITS file standard shall be aspired to. The algorithms developed in this research could as well be extended with a near-lossless, region-adaptive version by linear or non-linear noise quantisation (such as the methods discussed in Section 3.2.2). The advantage thereby gained – compared to other proposed quantisation-based lossy compression methods for astronomical images – is that if region segmentation works perfectly, the bright image regions could still be recovered losslessly.

Additionally, the results obtained on astronomical stellar-field images encourage two future research directions to pursue and continue the track chosen in this work. Specific examples of future research work, given below, could focus on one of these directions or a combination of both.

The first direction is to follow a similar content-based approach in other areas, where distinctive image properties can be identified and exploited. This direction focuses on finding new fields of application for the method devised for astronomical stellar-field images.

The second direction is to aim at further enhancing compression ratio by exploiting redundancy types other than spatial redundancy within a certain type of data. Compression methods for still images may exploit two different types of redundancy, spatial and spectral redundancy. Spectral redundancy is termed as the "correlation between different colour planes or spectral bands" by Saha (2000). It cannot be exploited for the images considered in this research, as these are not always produced in different wavebands, at the same time and location, using the same instruments. Still, taking advantage of this type of correlation may be a future direction of research with respect to astronomical images, especially with space-based photography.

With space-based instruments, data transfer capacities are an even more important concern than the network capacities of ground-based instruments and the same instruments are often used to capture data within several wavebands. The exploitation of temporal redundancy found in captured sequences of the same image area may also be an option in possible future applications if image sequences are stored in a format which allows a reference to previously collected data.

With regard to extending the adaptation-based approach to other areas, astronomy offers another type of images to target. The research work presented here did not explicitly tackle another class of important astronomical images: interferograms. Interferograms, as the one shown in Figure 4.1, may as well be effectively compressed with the ICSSM 3 algorithm. As these images typically do not contain very small, complex shaped region-of-stellar-objects areas, a dedicated ICSSM version for interferograms could possibly enhance size-efficiency on interferograms by using a geometrical description of the region demarcation information such as the one examined experimentally for stellar-field images. Rectangular regions could be examined instead of circular regions for an optimal reduction of the data size.

Another area of future research on the segmentation-based compression lies outside the application field of astronomical images. There are recent efforts to overcome the colour and contrast limitations of "standard" image acquisition and display hardware (Debevec et al. 2004). Segmentation-based compression techniques, as the one developed here, may become more important in this field of image processing as well. With High Dynamic Range Images (HDRI), pictures captured with a dynamic scale of more than 10000:1, the bit depth of an uncompressed image (per primitive colour plane) is almost twice as high as with the currently common 8-bit-per-colour-plane images generated by consumer cameras. Segmentation of the image planes into distinct areas and the relative encoding of textures in such regions may be an option as well for such images, although the image content will have more diverse characteristics. While HDRI images will require other segmentation methods than the ones used in this work, they could also allow lossy encoding techniques for the region textures, whenever the primary purpose of the encoding is to preserve the visual impression.

Chapter 8

Glossary

adaptive compression method A compression method that changes parameters based on local image area properties
astrometry Science of determining information on positions, distances and movements of astronomical objects
astrophotography Photographs of astronomical objects acquired using telescopes. Astrophotographs are taken in different wavebands.
arithmetic coding An efficient entropy coding method in terms of achievable compression ratio
background level With the ICSSM compression algorithms, the background level captures the intensity mode within a local image area, the tile.
background level data A data component generated through the segmentation step of the ICSSM compression methods. It consists of all background levels of the image.
background noise data A data component generated through the segmentation step of the ICSSM compression methods; for pixels assigned to the background it stores the low-dynamic noise
background region With the ICSSM compression algorithms, in the background region, the image signal component, which results from the incident of irradiation from astronomical sources, is zero or too small to be distinguished from the background noise.
bit rate Average number of bits required to represent a letter. For images, the bit rate is typically given in bits-per-pixel [bpp].
blocksorting methods Coding methods such as the Burrows-Wheeler Transform that use sorting techniques to process the data
blooming An effect which may occur when images are captured using

CCDs. Charge from very bright image areas exceeds the capacity of the respective CCD pixels and flows to surrounding pixels.

Burrows-Wheeler Transform A blocksorting-based compression method

coding Reversible techniques for redundancy reduction

compression ratio The quotient of storage required before and after compression

dark current A temperature dependent component of the astronomical image signal which is almost constant throughout the whole image

data reduction methods Methods which remove information from a signal, for example by quantisation or sampling

decorrelation methods Methods that transfer data into a representation which can be more efficiently compressed

dictionary coders Encoding methods that use dictionaries and references to dictionary entries to achieve compression

differential encoding Encoding of the differences between successive signal values

entropy A measure for the amount of information in a certain message

entropy coding Encoding methods that rely on the distribution of letters within the data

external file compression tools In this thesis, tools for general-purpose compression of binary data are referred to as external file compression tools.

first-order entropy A common and simple entropy measure. It takes only the distribution of letters within the data into account.

fractal compression methods Compression methods which exploit the self-similarity within the data

general-purpose file compression tools Implementations of general-purpose encoders

general-purpose encoders In the context of this research, the term general-purpose encoders refers to encoding methods such as Lempel-Ziv-Welch and the Burrows-Wheeler Transform, which are not image data specific.

general-purpose image compression methods In the context of this research, the term general-purpose image compression methods refers to image compression methods which are not astronomy specific.

higher-order entropy Entropy definitions which do not only take the distribution of letters into account. With higher-order entropy, also dependencies to other data values are accounted for.

image compression by segmentation and signal modelling (ICSSM) The novel compression algorithm presented in this research. It is based on a model of the stellar-field image data and the segmentation of the image into two regions: region-of-stellar-objects and background region.

image tile Size of the edge of the square image section used with the ICSSM algorithms to determine the background level (in pixels). The image tile size also determines the coarseness of the estimates of background levels in the image.

image histogram A diagram presenting the amount of image pixels as a function of the brightness

interferogram An image type that shows the intensity distribution of the radiation emitted by astronomical objects.

joint entropy An entropy measure for pairs of values.

lifting scheme An – in terms of computational-complexity – efficient method for the implementation of wavelet transforms.

near-lossless compression For such a compression method, the maximum of the reconstruction error is limited to a predefined small value.

photometry Science of determining the intensity of the radiation of astronomical objects

point-spread function Function that describes how a distant, point-like object is transformed through the telescope and capture equipment into a representation within an image.

prediction-based compression method Compression methods that exploit local redundancy in the data by predicting data values from already transmitted data values.

quantisation noise A noise component which results from the analogue-to-digital conversion process due to the limited amount of quantization levels and rounding.

reconstruction error Also called residual, is the difference between the original and the reconstructed data after compression. For lossless compression methods this reconstruction error is zero.

region demarcation information With the ICSSM compression algorithms, pixels are either assigned to the region-of-stellar-objects or the background region. The region demarcation information indicates to which of the two regions a certain pixel belongs.

region demarcation information mask A 1 bit-per-pixel array which stores the region demarcation information

region-of-stellar-objects With the ICSSM compression algorithms, pixels are either assigned to the region-of-stellar-objects or to the background region. In the region-of-stellar-objects, the image signal component which results from the incident of irradiation from astronomical sources is not zero and sufficiently large to be distinguished from the background noise.

residual See reconstruction error

run-length coding Encoding methods that exploit runs of similar data values for compression

segmentation-based methods Methods that segment the image into different regions, which are encoded separately

shot noise Poisson-distributed noise component in astronomical image data, which dominates in bright image regions. It results from the discrete nature of light and random photon fluctuations.

spatial redundancy Dependency or predictability among successive image pixels

spectral redundancy Dependency or predictability among images taken in different wavebands

spectroscopy Observations of the intensity distribution of the radiation emitted by astronomical objects. It provides information on composition, movement and temperature of astronomical objects.

stellar-field image A common type of astronomical images. Stellar-field images feature a noisy-background with bright objects in the foreground.

temporal redundancy Dependency or predictability among successive images in image sequences like videos

tile size Size of an image tile used with the ICSSM algorithms to determine the background level

transform-based compression method Methods that use transforms such as the Discrete Cosine Transform in their decorrelation step to concentrate the information

wavelet-based compression methods Compression methods which are based on a special type of transform, the wavelet transform

References

Abramson, N. (1963). Robust Transmission of Unbounded Strings Using Fibonacci Representations. Technical Report CS85-14. Weizmann Institute of Science, Department of Applied Mathematics. Rehovot (Israel).

Almoznino, E., Loinger, F., and Brosch, N. (1993). A Procedure for the Calculation of Background in Images. *Royal Astronomical Society Monthly Notices*. 265(3): 641 – 648.

Axelrod, T. S. (2006). The Large Synoptic Survey Telescope. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference XV*. Vol. 351 of ASP Conference Series. Astronomical Society of the Pacific. pp. 103 – 111.

Bai, X., Jin, J. S., and Feng, D. (2004). Segmentation-Based Multilayer Diagnosis Lossless Medical Image Compression. *In Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing VIP*. Vol. 100 of ACM International Conference Proceeding Series. Australian Computer Society. pp. 9 – 14.

Baltay, C., Snyder, J., Andrews, P., Emmet, W., Schaefer, B., Sinnot, J., Bailyn, C., Coppi, P., Oemler, A., Sabbey, C., Sofia, S., van Altena, W., Vivas, A., Abad, C., Bongiovanni, A., Briceño, C., Bruzual, G., Della Prigna, F., Magris, G., Sánchez, G., Sánchez, G., Schenner, H., Stock, J., Adams, B., Gebhard, M., Honeycutt, R., Musser, J., Rensgtorff, A., Ferrín, I., Fuenmayor, F., Hernández, J., Naranjo, O., Rosenzweig, P., Harris, F., and Geary, J. (2002). A Large-Area CCD Camera for the Schmidt Telescope at the Venezuelan National Astronomical Observatory. *Publications of the Astronomical Society of the Pacific*. 114(797): 780 – 794.

Barthel, K. U. (2003). Verlustlose Bildkompression (Lossless Image Compression). *it - Information Technology.* 45(5): 247 – 255.

Berg, A. P., and Mikhael, W. B. (1994). A Survey of Techniques for Lossless Compression of Signals. *In Proceedings of the 37th Midwest Symposium on Circuits and Systems.* Vol. 2. IEEE. pp. 943 – 946.

Biggar, M., Morris, O., and Constantinides, A. (1988). Segmented-Image Coding: Performance Comparison with the Discrete Cosine Transform. *Proceedings of the IEEE.* 135(2): 121 – 132.

Bijaoui, A. (1980). Sky Background Estimation and Application. *Astronomy and Astrophysics.* 84(1 – 2): 81 – 84.

Bijaoui, A., Bobichon, Y., and Li, H. (1996). Digital Image Compression in Astronomy – Morphology or Wavelets. *Vistas in Astronomy.* 40(4): 587 – 594.

Bolte, M. (2006). Modern Observational Techniques Lecture Notes (AY 257): Signal-to-Noise Lecture. [Online] Available from: http://www.ucolick.org/∼bolte/AY257/ [Accessed: 03.03.2007].

Boussalis, H., Liu, C., Rad, K., Dong, J., Chang, W. C., Kuo, C. C., and A., V. (2004). Scaleable Object-Based Compression Algorithm for Segmented Space Telescope Images. *In Multimedia Systems and Applications.* Vol. 5600 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 231 – 237.

Burrows, M., and Wheeler, D. (1994). A Block Sorting Lossless Data Compression Algorithm. Technical Report 124. Digital Equipment Corporation. Palo Alto, CA.

Calderbank, R., Daubechies, I., Sweldens, W., and Yeo, B. L. (1997). Lossless Image Compression using Integer to Integer Wavelet Transforms. *In Proceedings of the International Conference on Image Processing (ICIP 1997).* Vol. 1. IEEE Computer Society. pp. 596 – 599.

Carpentieri, B., Weinberger, M. J., and Serossi, G. (2000). Lossless Compression of Continuous-Tone Images. *Proceedings of the IEEE.* 88(11): 1797 – 1809.

Cheney, J. (2001). Compressing XML with Multiplexed Hierarchical PPM Models. *In Proceedings of the Data Compression Conference (DCC 2001).* IEEE Computer Society. pp. 163 – 172.

Christian, C. A. (2006). Tools and Services for Education and Outreach: Accessing Real Astronomical Data. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference XV.* Vol. 351 of ASP Conference Series. Astronomical Society of the Pacific. pp. 641 – 648.

Clunie, D. A. (2000). Lossless Compression of Grayscale Medical Images: Effectiveness of Traditional and State of the Art Approaches. *In Medical Imaging.* Vol. 3980 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 74 – 84.

Cohn, J. D. (2006). Power Spectrum and Correlation Function Errors: Poisson vs. Gaussian Shot Noise. *New Astronomy.* 11(4): 226 – 239.

Dawson-Howe, K. M. (1996). Lossless Image Compression Using a Simple Prediction Method. *International Journal of Imaging Systems and Technology.* 7(3): 227 – 230.

Debevec, P., Reinhard, E., Ward, G., and Pattanaik, S. (2004). High Dynamic Range Imaging. *In SIGGRAPH 2004 Course Notes.* ACM.

Debray, B. (2002). Using XML-Schema to Model Data from Present and Future Astronomical Databases. *In Toward an International Virtual Observatory.* Vol. 2004 of Proceedings of the ESO/ESA/NASA/NSF Conference Held at Garching (Germany), 10 – 14 June 2002. Berlin; Heidelberg: Springer. pp. 1431 – 2433.

Diolaiti, E., Bendinelli, O., Bonaccini, D., Close, L., Currie, D., and Parmeggiani, G. (2000). Analysis of Isoplanatic High Resolution Stellar Fields by the StarFinder Code. *Astronomy and Astrophysics Supplement Series.* 147: 335 – 346.

Djorgovski, G. (2003). *The Roles of Small Telescopes in a Virtual Observatory Environment.* Vol. 1 - Perceptions, Productivities, and Policies of Astrophysics and Space Science Library. Dordrecht: Kluwer Academic Publishers. pp. 85 – 95.

Dong, J., Boussalis, H., Liu, C., and Rad, K. (2003). Content-based Compression and Transmission of Astronomical Images. *In Proceedings of the International Conference on Information Technology: Research and Education.* IEEE. pp. 17 – 21.

Dony, R. D., and Haykin, S. (1995). Neural Network Approaches to Image Compression. *Proceedings of the IEEE.* 83(2): 288 – 303.

d'Ornellas, M. C. (2001). *Algorithmic Patterns for Morphological Image Processing.* PhD thesis. Amsterdam: Universiteit van Amsterdam.

Du, X., Li, H., and Ahalt, S. C. (2001). Content-Based Image Compression. *In Algorithms for Synthetic Aperture Radar Imagery VIII.* Vol. 4382 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 92 – 102.

Egmont-Petersen, M., de Ridder, D., and Handels, H. (2002). Image Processing With Neural Networks – a Review. *Pattern Recognition.* 35(10): 2279 – 2301.

Elias, P. (1975). Universal Codeword Sets and Representations of the Integers. *IEEE Transactions on Information Theory.* 21(2): 194 – 203.

Engel, D., and Uhl, A. (2002). Adaptive Object-based Image Compression Using Wavelet Packets. *In Proceedings of the 4th International Symposium of Video/Image Processing and Multimedia Communications (VIPromCom 2002).* IEEE. pp. 183 – 187.

ESO (2004). ESOMIDAS: European Southern Observatory Munich Image Data Analysis System. [Online] Available from: http://www.eso.org/projects/esomidas/ [Accessed: 28.02.2005].

Faller, N. (1973). An Adaptive System for Data Compression. *In Record of the 7th Asilomar Conference on Circuits, Systems and Computers.* Naval Postgraduate School, Monterey, CA. pp. 593 – 597.

Fano, R. M. (1949). The Transmission of Information. Technical Report 65. Massachusetts Institute of Technology, Research Laboratory of Electronics. Cambridge, MA.

Fenwick, P. (1996). Punctured Elias Codes for Variable-Length Coding of the Integers. Technical Report 137. University of Auckland, Department of Computer Science. Auckland (New Zealand).

Fisher, Y. (ed.) (1995). *Fractal Image Compression: Theory and Application.* London: Springer.

Freedman, D. A., Pisani, R., and Purves, R. A. (1998). *Statistics.* 3 edn. New York, NY: W. W. Norton and Company.

Freeman, H. (1974). Computer Processing of Line-Drawing Image. *ACM Computing Surveys.* 6(1): 57 – 97.

Fränti, P. (1994). A Fast and Efficient Compression Method for Binary Images. *Signal Processing: Image Communication.* 6(1): 69 – 76.

Fränti, P., Ageenko, E., Kukkonen, S., and Kälviäinen, H. (2002). Using Hough Transform for Context-Based Image Compression in Hybrid Raster/Vector Applications. *Journal of Electronic Imaging.* 11(2): 236 – 245.

Fränti, P., and Nevalainen, O. (1995). Compression of Binary Images by Composite Methods Based on Block Coding. *Journal of Visual Communication and Image Representation.* 6(4): 366 – 377.

Fusco, T., Véran, J.-P., Conan, J.-M., and Mugnier, L. (1999). Myopic Deconvolution Method for Adaptive Optics Images of Stellar Fields. *Astronomy and Astrophysics Supplementary Series.* 134: 193 – 200.

Gallager, R. G. (1978). Variations on a Theme by Huffman. *IEEE Transactions on Information Theory.* 24(6): 668 – 674.

Gaudet, S., Véran, J.-P., Delisle, D., and Pirenne, B. (2000). Compression of Mosaic CCD Images with CompFITS2. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference IX.* Vol. 216 of ASP Conference Series. Astronomical Society of the Pacific. pp. 547 – 550.

Gersho, A., and Gray, R. (1991). *Vector Quantisation and Signal Compression.* Norwell, MA: Kluwer Academic Publisher.

Gokturk, S. B., Tomasi, C., Girod, B., and Beaulieu, C. (2001). Medical Image Compression Based on Region of Interest, with Application to Colon CT Images. *In Proceedings of the 23nd Annual International Conference on Medicine and Biomedical Engineering.* Vol. 3. IEEE. pp. 2453 – 2456.

Golomb, S. W. (1966). Run-Length Encodings. *IEEE Transactions on Information Theory.* 12(3): 399 – 401.

Golombek, D. (2004). Archives, Databases and the Emerging Virtual Observatories. *Astrophysics and Space Science.* 290(3 – 4): 449 – 456.

Green, T., Miklau, G., Onizuka, M., and Suciu, D. (2004). Processing XML Streams with Deterministic Automata. *Transactions on Database Systems.* 29(4): 173 – 189.

Grosbol, P. J., Murtagh, F., and Warmels, R. H. (eds.) (1989). *Proceedings of the First ESO/ST-ECF Data Analysis Workshop.* Vol. 31 of ESO Conference and Workshop Proceedings. Garching (Germany): European Southern Observatory.

Gupta, S., and Virdi, S. P. S. (1989). Information Content in Astronomical Images. *Astrophysics and Space Science.* 162(1): 159 – 161.

Healey, G. E., and Kondepudy, R. (1994). Radiometric CCD Camera Calibration and Noise Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 16(3): 267 – 276.

Heijmans, H. J. A. M. (1994). *Morphological Image Operators.* Advances in Electronics and Electron Physics, Supplement. 25 edn. Boston, MA: Academic Press.

Hobby, J. D. (1997). Space-Efficient Outlines from Image Data via Vertex Minimization and Grid Constraints. *Graphical Models and Image Processing.* 59(2): 73 – 88.

Howell, S. B. (2000). *Handbook of CCD Astronomy (Cambridge Observing Handbooks for Research Astronomers).* Cambridge (United Kingdom): Cambridge University Press.

Hsu, J., and Hodge, P. E. (2003). Demo of Numarray, PyFITS, and Related Software. *In Proceedings of the Astronomical Data Analysis Software* *and Systems Conference XIII.* Vol. 314 of ASP Conference Series. Astronomical Society of the Pacific. p. 828.

Hu, J., Wang, Y., and Cahill, P. (1997). Multisprectral Code Excited Linear Prediction Coding and its Application in Magnetic Resonance Images. *IEEE Transaction on Image Processing.* 6(11): 1555 – 1566.

Hu, L., Chen, Q.-A., and Zhang, D. (2004). An Image Compression Method Based on Fractal Theory. *International Conference on Computer Supported Cooperative Work on Design.* 1(26 – 28): 546 – 550.

Huang, L., and Bijaoui, A. (1990). Astronomical Image Data Compression by Morphological Skeleton Transformation. *Experimental Astronomy.* 1(5): 311 – 327.

Huffman, D. (1952). A Method for the Construction of Minimum Redundancy Codes. *Proceedings of the IRE.* 40(9): 1098–1101.

Hunt, D. (1998). White Paper on Lossless Compression. [Online] Available from: http://www.ece.uprm.edu/∼hunt/research/whitepap.htm [Accessed: 02.01.2006].

IRAF (2004). Image Reduction and Analysis Facility. [Online] Available from: http://iraf.noao.edu/iraf-hompage.html [Accessed: 28.02.2005].

IRAF (2005). IRAF Standard Test Image Set. [Online] Available from: http://iraf.anu.edu.au/iraf/ftp/iraf/extern/focas.std.tar.Z [Accessed: 28.02.2005].

ISO/IEC (1994). ISO/IEC 10918-1:1994, *Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines*: Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2001). ISO/IEC 14492:2001, *Information Technology – Lossy/Lossless Coding of Bi-Level Images*: Geneva, Switzerland: International Organization for Standardization.

ISO/IEC (2004). ISO/IEC 15948:2004, *Information Technology – Computer Graphics and Image Processing – Portable Network Graphics (PNG): Functional Specification*: Geneva, Switzerland: International Organization for Standardization.

Jacquin, A. E. (1992). Image Coding Based on a Fractal Theory of Iterative Contractive Image Transformations. *IEEE Transactions on Image Processing.* 1(1): 18 –30.

Jang, J., and Rajala, S. A. (1990). Segmentation Based Image Coding Using Fractals and the Human Visual System. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 1990.* Vol. 4 of ICASSP Proceeding Series. IEEE. pp. 1957 – 1960.

Jang, J., and Rajala, S. A. (1991). Texture Segmentation-Based Image Coder Incorporating Properties of the Human Visual System. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 1991.* Vol. 4 of ICASSP Proceeding Series. IEEE. pp. 2753 – 2756.

JBIG Committee (1999). Lossy/Lossless Coding of Bi-Level Images (14492 FCD): Final Technical Specification of the JBIG2 Standard. [Online] Available from: http://www.jpeg.org/public/fcd14492.pdf [Accessed: 22.01.2007].

Johnson, O. (2003). Information Theory (Lecture notes). [Online] Available from: http://www.statslab.cam.ac.uk/∼johnson/it.html [Accessed: 20.01.2007].

Knuth, D. E. (1985). Dynamic Huffman Coding. *Journal of Algorithms.* 6(2): 163 – 180.

Kraft, L. G. (1949). *A Device for Quantizing, Grouping and Coding Amplitude Modulated Pulses.* Master's thesis. Massachusetts Institute of Technology, Department of Electrical Engineering. Cambridge, MA.

Kuijken, K., Bender, R., Cappellaro, E., Muschielok, B., Baruffolo, A., Cascone, E., Hess, H.-J., Iwert, O., Nicklas, H., Reif, K., Valentijn, E., Baade, D., Begeman, K., Bortolussi, A., Boxhoorn, D., Christen, F., Deul, E., Greggio, L., Harke, R., Häfner, R., Hopp, U., Ilijevski, I., Klink, G., Kravcar, H., Magagna, C., Mitsch, W., Müller, P., Poschmann, H., Rengelinka, R., and Wellem, W. (2004). OmegaCAM: Wide-Field Imaging with fine Spatial Resolution. *In SPIE Conference Astronomical Telescopes and Instrumentation: Ground-based In-*

strumentation for Astronomy. Vol. 5492 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 484 – 493.

Kundu, M. K., Chaudhari, B. B., and Majumder, D. D. (1985). A Generalised Digital Contour Coding Scheme. *Computer Vision, Graphic and Image Processing.* 30(3): 269 – 278.

Kunt, M. (1988). Progress in High Compression Image Coding. *International Journal of Pattern Recognition and Artificial Intelligence.* 2(3): 387 – 405.

Kunt, M., Ikonomopoulos, A., and Kocher, M. (1985). Second-Generation Image-Coding Techniques. *Proceedings of the IEEE.* 73(4): 549 – 574.

Lempel, A., and Ziv, J. (1978). Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory.* 24(4): 530 – 536.

Liefke, H., and Suciu., D. (2000). XMill: an Efficient Compressor for XML Data. *In Proceedings of the SIGMOD International Conference on Management of Data.* Vol. 29 of Proceedings of the ACM. ACM. pp. 153 – 164.

Louys, M., Starck, J.-L., Mei, S., Bonarel, F., and Murtagh, F. (1999). Astronomical Image Compression. *Astronomy and Astrophysics Supplement Series.* 136: 579 – 590.

Louys, M., Starck, J., and Murtagh, F. (1999). Lossless Compression of Astronomical Images. *Irish Astronomical Journal.* 26(2): 119 – 122.

Lupton, R. (2004). Image Processing in the Sloan Digital Sky Survey. [Online] Available from: http://www.ipam.ucla.edu/publications/ai2004/ai2004_4539.pdf [Accessed: 03.03.2007].

Mackay, C. D. (1986). Charge-Coupled Devices in Astronomy. *Annual Review of Astronomy and Astrophysics.* 24: 255 – 283.

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature.* 1 edn. New York, NY: W. H. Freeman and Co.

Mandyam, G. D., Ahmed, N., and Magotra, N. (1995). DCT-Based Scheme for Lossless Image Compression. *In Digital Video Compression: Algorithms and Technologies.* Vol. 2419 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 474 – 478.

Martins, B., and Forchhammer, S. (1996). Bi-level Image Compression with Tree Coding. *In Proceedings of the Data Compression Conference (DCC 1996).* IEEE Computer Society. pp. 270 – 279.

McMillan, B. (1956). Two Inequalities Implied by Unique Decipherability. *IRE Transactions on Information Theory.* 2(4): 115 – 116.

McNerney, P. (2000). The Communication of Images from New Generation Astronomical Telescopes. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference IX.* Vol. 216 of ASP Conference Series. Astronomical Society of the Pacific. pp. 323 – 326.

Memon, N. D., and Wu, X. (1997). Recent Developments in Context-Based Predictive Techniques for Lossless Image Compression. *The Computer Journal.* 40(2/3): 127 – 136.

Mielikainen, J. (2006). Lossless Compression of Hyperspectral Images Using Lookup Tables. *IEEE Signal Processing Letters.* 13(3): 157 – 160.

Moffat, A., Bell, T., and Witten, I. (1997). Lossless Compression for Text and Images. *International Journal of High Speed Electronics and Systems.* 8(1): 179 – 231.

Mueller, F. (1999). *Pyramiden zur Bildkompression (Pyramids for Image Compression).* PhD thesis. Duesseldorf: Technische Hochschule Aachen.

Murtagh, F. (2006). MR/1 Documentation. [Online] Available from: http://www.multiresolution.com [Accessed: 28.02.2006].

Netravali, A. N., and Limb, J. O. (1980). Picture Coding: A Review. *IEEE Proceedings.* 68(3): 366 – 406.

Nieto-Santisteban, N. A., Hanisch, R. J., and Stockman, H. S. (1999). Data Compression for the Next Generation Space Telescope. *In Proceedings of the Data Compression Conference (DCC 1999).* IEEE Computer Society. p. 542.

Ono, F., Rucklidge, W., Arps, R., and Constantinescu, C. (2000). JBIG2 – the Ultimate Bi-level Image Coding Standard. *In Proceedings of the International Conference on Image Processing (ICIP 2000)*. Vol. 1. Vancouver, BC (Canada): IEEE Computer Society. pp. 140 – 143.

Paeth, A. W. (1991). *Image File Compression Made Easy*. Vol. 2. San Diego: Academic Press. pp. 93 – 100.

Pardas, M. (1997). Object-Based Image Coding. *Vistas in Astronomy*. 41(3): 455 – 461.

Pavlidis, T. (1981). *Algorithms for Graphics and Image Processing*. 1 edn. Rockville, IN: Computer Science Press.

Pence, W. D. (1992). FITSIO and FITS File Utility Software. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference I*. Vol. 25 of ASP Conference Series. Astronomical Society of the Pacific. p. 22.

Pence, W. D. (1994). FITS Image Compression Using Variable Length Binary Tables. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference III*. Vol. 61 of ASP Conference Series. Astronomical Society of the Pacific. p. 523.

Pence, W. D. (2007). CFITSIO User's Reference Guide. [Online] Available from: http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/c/cfitsio.pdf [Accessed: 10.01.2008].

Pennebaker, W. B., and Mitchell, J. L. (1992). *JPEG Still Image Compression Standard*. New York, NY: Van Nostrand Reinhold.

Penrose, A. J., and Dodgson, N. A. (1999). Error Resilient Lossless Image Coding. *In Proceedings of the International Conference on Image Processing (ICIP 1999)*. Vol. 1. IEEE Computer Society. pp. 426 – 429.

Pereira, F. C., and Ebrahimi, T. (2002). *The MPEG-4 Book*. Upper Saddle River, NJ: Prentice Hall PTR.

Pool, P. J., Holtom, R., and Morris, D. G. (1998). CCDs for Astronomy. *In Optical Detectors for Astronomy*. Vol. 228 of ESO 1996 Workshop Proceedings. Kluwer Academic Publisher. pp. 29 – 36.

Press, W. H. (1992). Wavelet based Compression Software for FITS Images. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference I.* Vol. 25 of ASP Conference Series. Astronomical Society of the Pacific. p. 3.

Puetter, R., Gosnell, T., and Yahil, A. (2005). Digital Image Reconstruction: Deblurring and Denoising. *Annual Review of Astronomy and Astrophysics.* 43(1): 139 – 194.

Quinn, P. J., and Gorski, K. M. (eds.) (2004). *Toward an International Virtual Observatory.* Berlin; Heidelberg: Springer.

Ragab, A. S., Mohamed, A. S. A., and Hamid, M. (1998). Efficiency of Analytical Transforms for Image Compression. *In Proceedings of the Fifteenth National Radio Science Conference (NRSC 1998).* Vol. B16 of National Radio Science Conference Series. International Union of Radio Science. pp. 1 – 10.

Ramakrishnan, S., Rose, K., and Gersho, A. (1998). Constrained-Storage Vector Quantization with a Universal Codebook. *IEEE Transactions on Image Processing.* 7(6): 785 – 793.

Rao, K. R., and Yip, P. (1990). *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Boston: Academic Press.

Ratakonda, K., and Ahuja, N. (1996). Segmentation Based Reversible Image Compression. *In Proceedings of the International Conference on Image Processing (ICIP 1996).* Vol. 1. IEEE Computer Society. pp. 81 – 84.

Reid, M. M., Millar, R. J., and Black, N. D. (1997). Second-Generation Image Coding: An Overview. *ACM Computing Surveys.* 29(1): 3 – 29.

Rice, R. F. (1979). Some Practical Universal Noiseless Coding Techniques - Part I. Technical Report JPL-79-22. Jet Propulsion Laboratory. Pasadena, CA.

Rice, R. F. (1983). Some Practical Universal Noiseless Coding Techniques - Part II. Technical Report JPL-83-17. Jet Propulsion Laboratory. Pasadena, CA.

Rice, R. F. (1991). Some Practical Universal Noiseless Coding Techniques - Part III. Technical Report JPL-91-3. Jet Propulsion Laboratory. Pasadena, CA.

Rice, R. F., and Plaunt, J. R. (1971). Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data. *IEEE Transactions on Communication Technology.* 6(19): 889 – 897.

Romeo, A., Gaztanaga, E., Barriga, J., and Elizalde, E. (1999). Information Content in Gaussian Noise: Optimal Compression Rates. *International Journal of Modern Physics C.* 10: 687 – 716.

Roos, P., and Viergever, M. A. (1988). Reversible Intraframe Compression of Medical Images. *IEEE Transactions on Medical Imaging.* 7(4): 328 – 336.

Sabbey, C. N. (1999). Adaptive, Lossless Compression of 2-D Astronomical Images. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference VIII.* Vol. 172 of ASP Conference Series. Astronomical Society of the Pacific. p. 129.

Sabbey, C. N., Coppi, P., and Oemler, A. (1998). Data Acquisition for a 16 CCD Drift-Scan Survey. *Publications of the Astronomical Society of the Pacific.* 110(751): 1067 – 1080.

Saha, S. (2000). Image Compression – From DCT to Wavelets: A Review. *ACM Crossroads.* 6(3): 12 – 21.

Said, A., and Pearlman, W. A. (1993). Reversible Image Compression via Multiresolution Representation and Predictive Coding. *In Proceeding of Visual Communications and Image Processing '93.* Vol. 2094 of Proceedings of the SPIE. Society of Photo-Optical Instrumentation Engineers (SPIE). pp. 664 – 674.

Said, A., and Pearlman, W. A. (1996). A new, Fast and Efficient Image Codec Based on set Partitioning in Hierarchical Trees. *IEEE Transactions Circuits and Systems for Video Technology.* 6(3): 243 – 249.

Salomon, D. (2000). *Data Compression: The Complete Reference.* 2 edn. New York; Berlin; Heidelberg: Springer.

Salomon, D. (2002). *A Guide to Data Compression Methods.* New York; Berlin; Heidelberg: Springer.

Salomon, D. (2006). Data Compression: The Complete Reference (Added Material: Data Correlation). [Online] Available from: http://www.ecs.csun.edu/~dsalomon/DC2advertis/Corr.pdf [Accessed: 22.12.2006].

Saupe, D., and Hamzaoui, R. (1994). A Review of the Fractal Image Compression Literature. *ACM SIGGRAPH Computer Graphics.* 28(4): 268 – 276.

Savakis, A. E. (2002). Evaluation of Algorithms for Lossless Compression of Continuous-tone Images. *Journal of Electronic Imaging.* 11(1): 75 – 86.

Sayood, K. (2000). *Introduction to Data Compression.* 2 edn. San Diego, CA: Academic Press.

Schindler, M. (1998). A Fast Renormalization for Arithmetic Coding. *In Proceedings of the Data Compression Conference (DCC 1998).* IEEE Computer Society. p. 572.

Schindler, M. (1999). Range Encoder Implementation. [Online] Available from: http://www.compressconsult.com/rangecoder/ [Accessed: 28.02.2005].

Schmalz, M. S. (2005). Visualization, Graphics, and Image Processing: On the Convergence of Image Compression and Object Recognition. *In Proceedings of the 43rd Annual Southeast Regional Conference.* Vol. 2. ACM: Association for Computing Machinery. pp. 382 – 387.

Seaman, R., Pence, W., White, R., Dickinson, M., Valdes, F., and Zarate, N. (2006). Astronomical Tiled Image Compression: How and Why. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference XVI.* Vol. 376 of ASP Conference Series. Astronomical Society of the Pacific. p. 483.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal.* 27: 379 – 423.

Shannon, C. E. (1951). Prediction and Entropy of Printed English. *Bell System Technical Journal.* 30: 50 – 64.

Sikora, T. (2003). MPEG-4 Objektbasierte Videocodierung (MPEG-4 Object-Based Video Coding). *it – Information Technology.* 45(5): 273 – 279.

Starck, J. L., and Murtagh, F. (2001). Astronomical Image and Signal Processing: Looking at Noise, Information and Scale. *IEEE Signal Processing Magazine.* 18(2): 30 – 40.

Starck, J., Murtagh, F., and Louys, M. (1995). Astronomical Image Compression Using the Pyramidal Median Transform. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference IV.* Vol. 77 of ASP Conference Series. Astronomical Society of the Pacific. pp. 268 – 271.

Starck, J., Murtagh, F., Querre, P., and Bonnarel, F. (2001). Entropy and Astronomical Data Analysis: Perspectives From Multiresolution Analysis. *Astronomy and Astrophysics.* 368: 730 – 746.

Strom, J., and Cosman, P. C. (1997). Medical Image Compression with Lossless Regions of Interest. *Signal Processing.* 59(2): 155 – 171.

Strutz, T. (2002). Context Based Adaptive Linear Prediction for Lossless Image Coding. *In Proceedings of the 4th International ITG Conference on Source and Channel Coding.* Vol. 170 of ITG - Fachbericht. VDE - Association for Electrical, Electronic and Information Technologies. pp. 105 – 109.

Strutz, T. (2005). *Bilddatenkompression (Image Compression).* Studium Technik. 6 edn. Braunschweig; Wiesbaden (Germany): Vieweg.

Strutz, T. (2007). Data Compression - Systematisation. [Online] Available from: http://navatrump.de/Technology/Datacompression/compression.html [Accessed: 27.12.2007].

Swelden, W. (1996). The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets. *Applied and Computational Harmonic Analysis.* 3(2): 186 – 200.

Szalay, A. S., Gray, J., Kunszt, P., Thakar, A., and Slutz, D. (2001). Large Databases in Astronomy. *In Proceedings of the MPA/ESO/MPE Workshop: Mining the Sky.* Vol. 2001 of ESO Astrophysics Symposia. Springer. pp. 99 – 116.

Szalay, A. S., Kunszt, P. Z., Thakar, A., Gray, J., Slutz, D., and Brunner, R. J. (2000). Designing and Mining Multi-Terabyte Astronomy Archives: the Sloan Digital Sky Survey. *ACM SIGMOD Record Archive.* 29(2): 451 – 462.

Teuhola, J. (1978). A Compression Method for Clustered bit-Vectors. *Information Processing Letters.* 10(7): 308 – 311.

Vemuri, B., Sahni, S., Chen, F., Kapoor, C., Leonard, C., and Fitzsimmons, J. (2002). *Lossless Image Compression.* Vol. 2. New York; Basel: Marcel Dekker. pp. 1177 – 1193.

Villard, R., and Williams, R. (2006). HubbleSite: Newscenter - News STScI-1996-01. [Online] Available from: http://hubblesite.org/newscenter/archive/releases/1996/01/text/ [Accessed: 12.05.2006].

Véran, J. P., and Wright, J. R. (1994). Compression Software for Astronomical Images. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference III.* Vol. 61 of ASP Conference Series. Astronomical Society of the Pacific. p. 519.

Wallace, G. (1991). The JPEG Still Picture Compression Standard. *Communications of the ACM.* 34(4): 30 – 44.

Watson, A. (2002). A Lossy Method for Compressing Raw CCD Images. *Revista Mexicana de Astronomía y Astrofísica.* 38: 233 – 249.

Weghorn, H. (2002). Applying Data Compression Methods. *In Toward an International Virtual Observatory: Proceedings of the ESO/ESA/NASA/NSF Astronomy Conference.* ESO Astrophysics Symposia. Springer. pp. 327 – 328.

Weghorn, H., Colberg, J., and Bauer, T. (1996). Real-Time Image Compression Method for Astronomical Images. *In Science with the VLT Interferometer: Proceedings of the ESO Workshop.* ESO Astrophysics Symposia. Springer. pp. 399 – 400.

Weinberger, M., Seroussi, G., and Sapino, G. (2000). The LOCO-1 Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS. *IEEE Transactions on Image Processing.* 9(8): 1309 – 1324.

Welch, T. A. (1984). A Technique for High-Performance Data Compression. *IEEE Computer.* 17(6): 8 – 19.

White, R. L. (1992). High-Performance Compression of Astronomical Images. *In Proceedings of the NASA Space and Earth Science Data Compression Workshop.* Vol. CP-3191. NASA. p. 403.

White, R. L., and Becker, I. (1998). On-Board Image Compression for the HST Advanced Camera for Surveys. *In Space Telescopes and Instruments V.* Vol. 3356 of Prcceedings of the SPIE. International Society of Optical Engineering. pp. 823 – 831.

White, R. L., and Greenfield, P. (1998). A Scheme for Compressing Floating-Point Images. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference VIII.* Vol. 172 of ASP Conference Series. Astronomical Society of the Pacific. pp. 125 – 128.

White, R. L., Greenfield, P., Pence, W., and Tody, D. (1999). Specifications for Storing Compressed Images in FITS Binary Tables. [Online] Available from: http://ledas-www.star.le.ac.uk/lheasoft/fitsio/compression/compress_image.html [Accessed: 20.03.2007].

White, R. L., and Percival, J. W. (1994). Compression and Progressive Transmission of Astronomical Images. *In Advanced Technology Optical Telescopes V.* Vol. 2199 of Proceedings of the SPIE. International Society of Optical Engineering. pp. 703 – 713.

Withagen, P. J., Groen, F. C. A., and Schutte, K. (2007). CCD Color Camera Characterization for Image Measurements. *IEEE Transactions on Instrumentation and Measurement.* 56(1): 199 – 203.

Witten, I. H., Neal, R. M., and Cleary, J. G. (1987). Arithmetic Coding for Data Compression. *Communications of the ACM.* 30(6): 520 – 540.

Xue, S., Xu, Y., and Oelmann, B. (2003). Hybrid Golomb Codes for a Group of Quantised GG Sources. *IEE Proceedings of Vision, Image and Signal Processing.* 150(4): 256 – 260.

Zandi, A., Allen, J. D., Schwartz, E. L., and Boliek, M. P. (1995). CREW: Compression With Reversible Embedded Wavelets. *In Proceedings of the Data Compression Conference (DCC 1995).* IEEE Computer Society. pp. 212 – 221.

Zhao, E., and Liu, D. (2005). Fractal Image Compression Methods: A Review. *In Proceedings of the 3rd International Conference on Information Technology an Applications (ICITA 2005).* Vol. 1. IEEE Computer Society. pp. 756 – 759.

Ziv, J., and Lempel, A. (1977). A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory.* 23(3): 337 – 343.

Appendix A

Geometrical Region Demarcation Information Description

With this algorithm, the aim is to generate an ellipse based lossy description of the region demarcation information by using an object detection-like approach. Ellipse-like structures in the bi-level image region demarcation information are described by elliptical objects represented by their parameters midpoint position and the size of their major- and minor-axis. This approach uses ellipses with major- and minor-axis aligned to x- and y-axis of the binary image containing the region demarcation information. The algorithm first generates a set of ellipse parameter hypotheses.

A.1 Midpoint Hypotheses Generation

Successively, a range of hypotheses of possible ellipse positions in the whole image is generated by applying an algorithm that estimates the midpoint and the axes size using geometrical symmetry properties. Any region-of-stellar-objects point is considered to be a potential midpoint of an elliptical stellar object region. The midpoint estimate M_i is determined iteratively from this starting point. A secant (aligned to the x-axis) is drawn through the stellar object region and the midpoint of it – relative to the stellar-object-region-to-background border – is determined. Then, a secant which is oriented orthographically to the first one is drawn and the midpoint of this second secant is used as a new midpoint estimate M_{i+1}. For a convex region, the point M_i successively converges to the midpoint of the region. The iteration is stopped if the distance between successive midpoint estimates $d_{i+1} = M_{i+1} - M_i$ falls below a predefined limit. Estimates for the size of both main axes are determined as well as from the size of the secants.

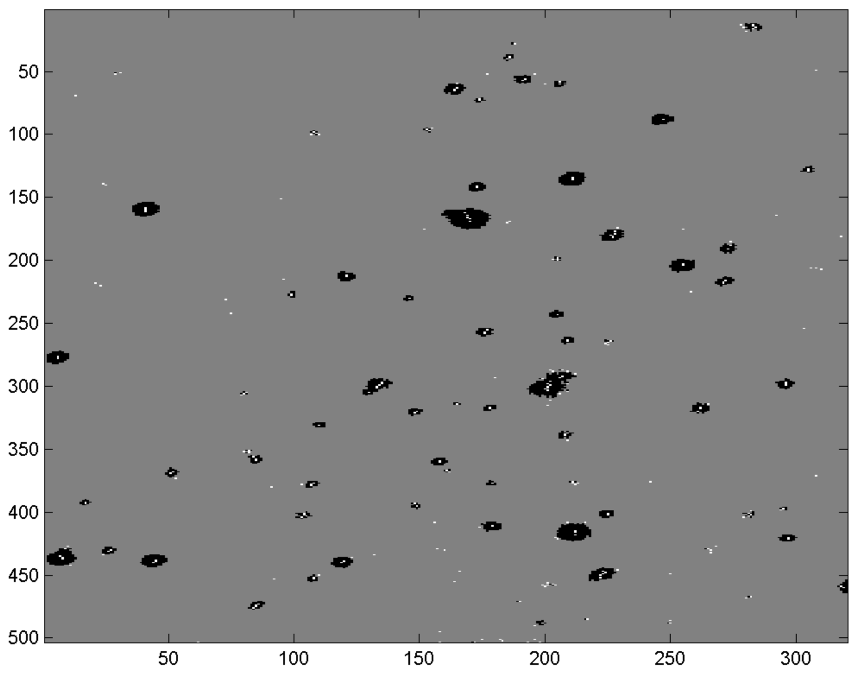


Figure A.1: Midpoint hypotheses (white dots) generated through the iterative geometrical approach.

This method determines possible midpoint estimates for all region-of-stellar-objects areas. The midpoint estimates are shown as white dots in Figure A.1. Due to the large amount of starting points, far too many midpoint estimates are found. For example, about 2800 are found in the image shown in Figure A.1. The chosen approach ensures that every ellipse is found. Redundant ellipse property estimates are removed during the next processing steps.

A.2 Removal of Replicated Ellipses

Due to the midpoint hypotheses generation algorithm used, a range of ellipse property estimates are found more than once. Doubles are removed from the list of ellipse hypotheses. For the sample image shown above, the amount of ellipse hypotheses is reduced from about 2800 to less than 300.

A.3 Single Region-of-Stellar-Objects Pixel Removal

As can also be seen from Figure A.1, a range of very simple ellipses is present in the image: elliptical objects with a minor and major axis of size one. These "ellipses" are single pixels assigned to the region-of-stellar-objects.

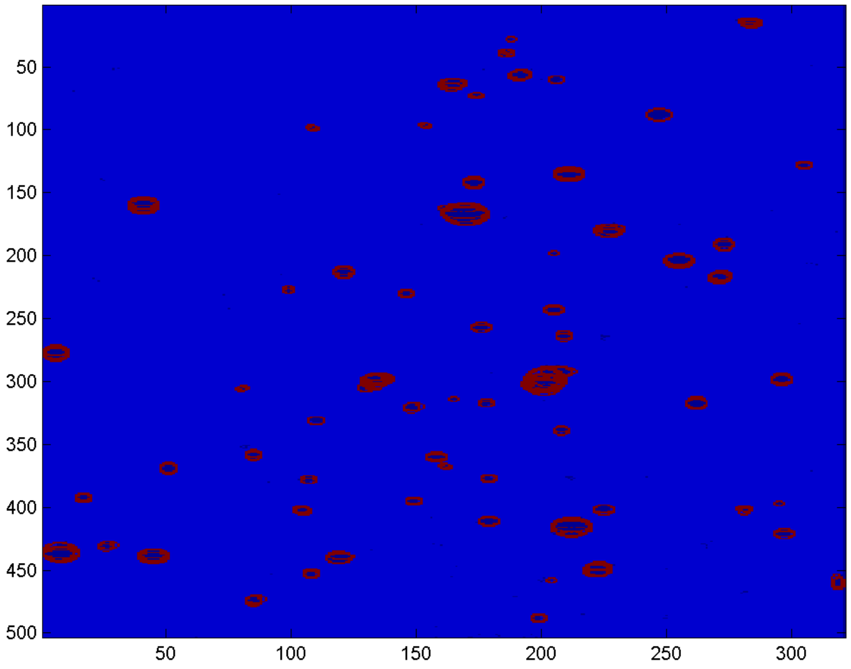


Figure A.2: Remaining identified elliptical objects after removal of single pixels assigned to the region-of-stellar-objects.

These can be stored very efficiently separately as a list of single pixel coordinates, without storing the minor- and major-axis parameters. In the examined image, 74 of such single pixel objects are present. 203 hypotheses for elliptical objects remain. These ellipses are shown in Figure A.2.

A.4 Completely Contained Ellipses Removal

Furthermore, in the sample figure, the removal of ellipses which are completely contained in other ellipses reduces the amount of objects to 125 remaining ellipse estimates (Figure A.3).

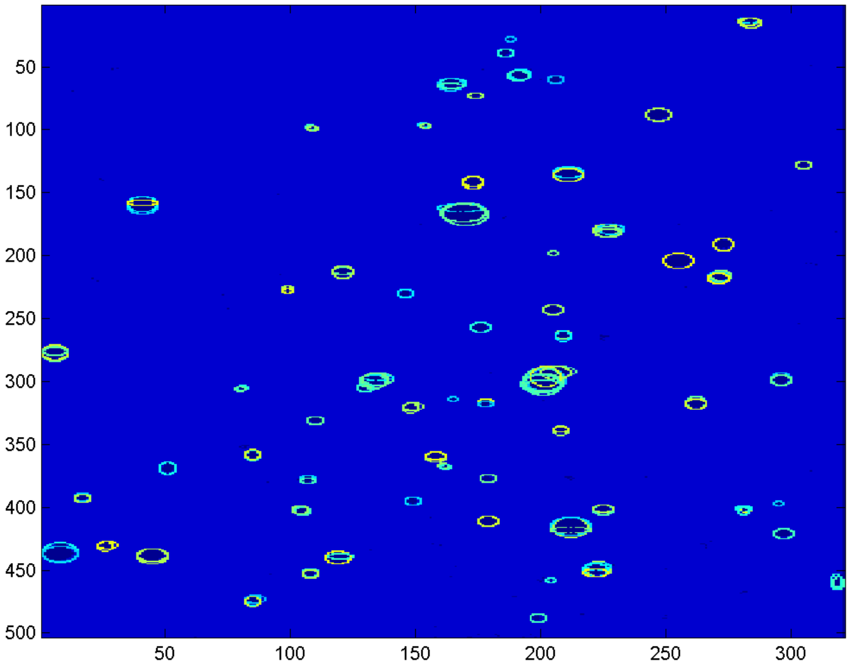


Figure A.3: Remaining identified elliptical objects after removal of single pixels assigned to the region-of-stellar-objects and ellipses completely contained in other ellipses.

After these steps, removing single pixels and redundant ellipse estimates, the region demarcation information comprises of 125 ellipse estimates (each requires 4 parameters which can be encoded with 9 bits each) and 74 single dot objects (which require two 9-bit parameters). Disregarding required additional overhead, that data can be encoded using approximately 5832 bits.

The original size of the region demarcation information (binary image) was 503 by 320 pixels, each encoded using one bit-per-pixel (160960 bits). That results in a compression ratio of almost 27.5 for the lossy compression of the region demarcation information data. Unfortunately, this does not exceed the results of other – non lossy – methods very much.

A.5 Performance of an Optimal Method for Generating a Geometrical Region Demarcation Information Description

Counting the ellipses in Figure A.3 leads to the assumption that with an optimal geometrical method, the whole data could be described with only 65 elliptical objects. These 65 elliptical objects could be described with approximately 2340 bits, leading to an overall size requirement of 3672 bits for the whole region demarcation information, including the single pixels. This result leads to a compression ratio for the region demarcation information of approximately 44.

Although this could lead to an enhancement of the compression ratio for the region demarcation information compared to other methods, further examinations in this direction are not conducted. The gain by using a lossy region information description is too small to justify an additional re-segmentation step.

Appendix B

Fitsdiff Sample Output

Python Fitsdiff is called in a Python[1] interpreter session.

```
>>> fitsdiff.fitsdiff("sgp0001.fits", "sgp0001.unc.fits")
 fitsdiff:  1.3 (22 July, 2004)
 file1 = sgp0001.fits
 file2 = sgp0001.unc.fits
 Keyword(s) not to be compared:  ['']
 Keyword(s) whose comments not to be compared:  ['']
 Column(s) not to be compared:  ['']
 Maximum number of different pixels to be reported:  10
 Data comparison level:  0.0
Primary HDU:
  Extra keyword HISTORY  in sgp0001.unc.fits
  Extra keyword CTYPE2   in sgp0001.unc.fits
  Extra keyword CTYPE1   in sgp0001.unc.fits
    Data differ at   [1, 1], file 1:  2312 file 2:  2311
    Data differ at   [2, 1], file 1:  2329 file 2:  2328
    ...
    Data differ at   [12, 1], file 1:  2513 file 2:  2512
    Data differ at   [15, 1], file 1:  2348 file 2:  2347
    There are 51425 different data points.
0
>>>
```

[1] http://www.python.org (23.03.2007)

Appendix C

Test Image Set

This section shows the test images used in performance assessment experiments, along with their histograms. Detailed information about the images and their origin can be found in Section 6.1.2.

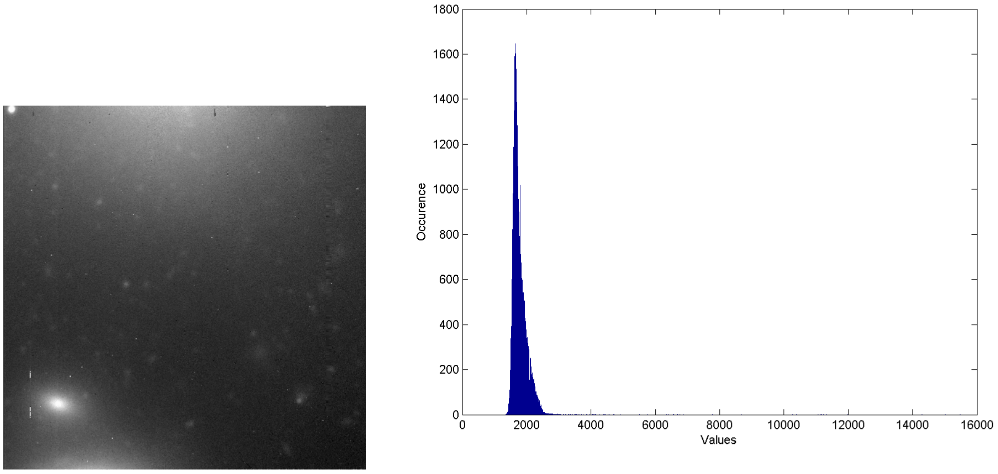


Figure C.1: Plot and histogram of image “com0001”.

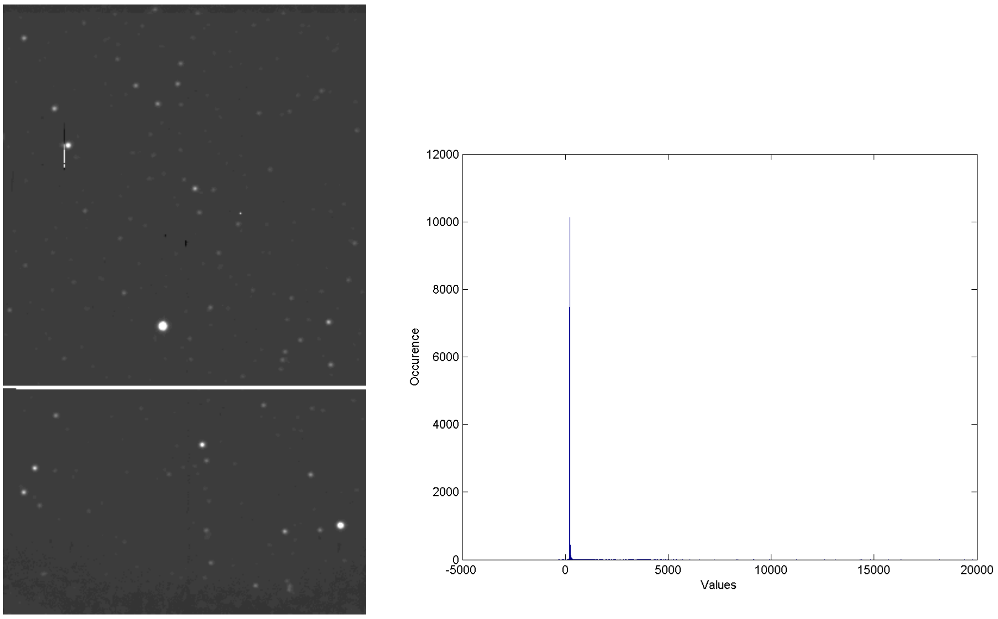


Figure C.2: Plot and histogram of image “for0001”.

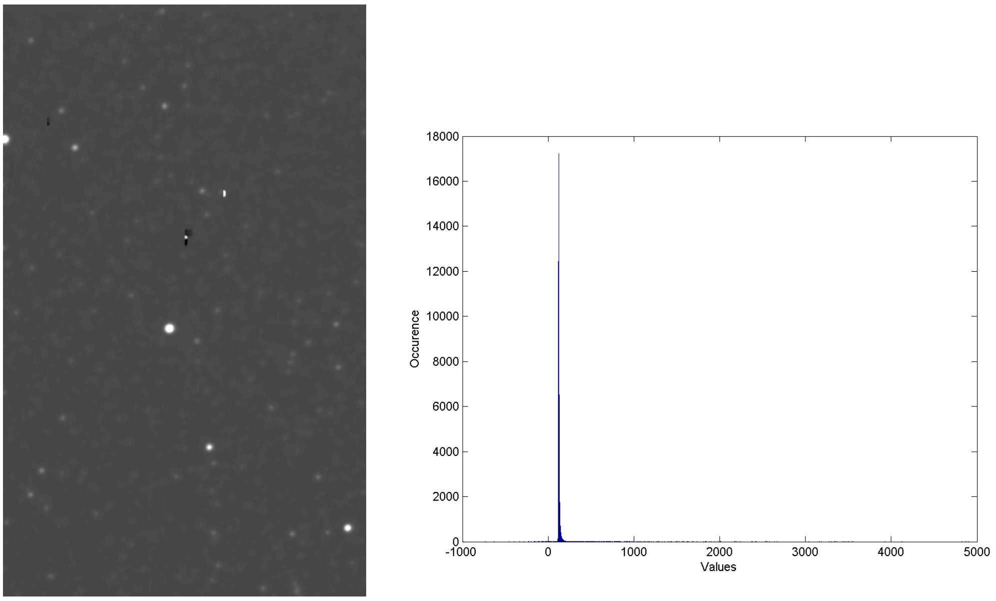


Figure C.3: Plot and histogram of image “for0002”.

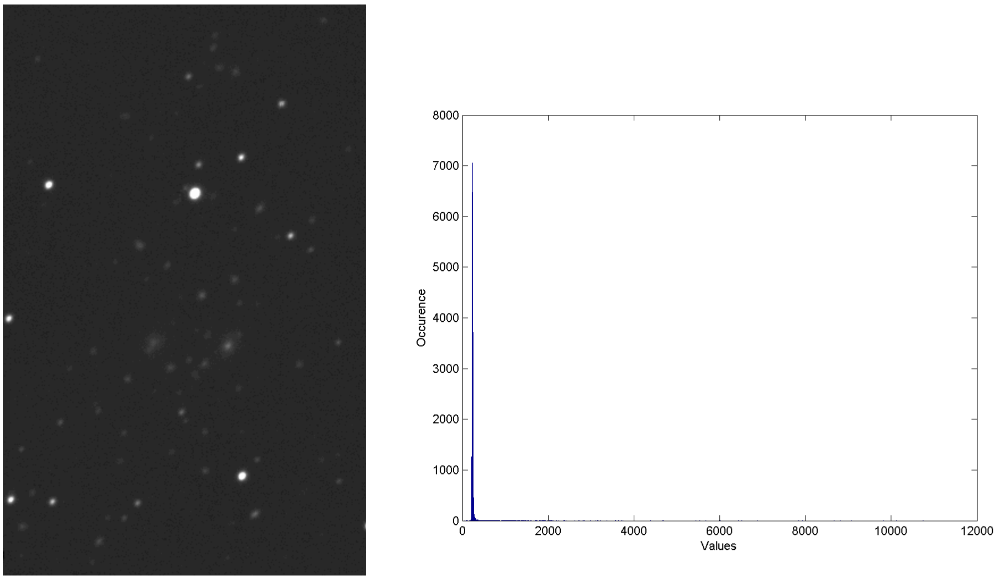


Figure C.4: Plot and histogram of image “gal0001”.

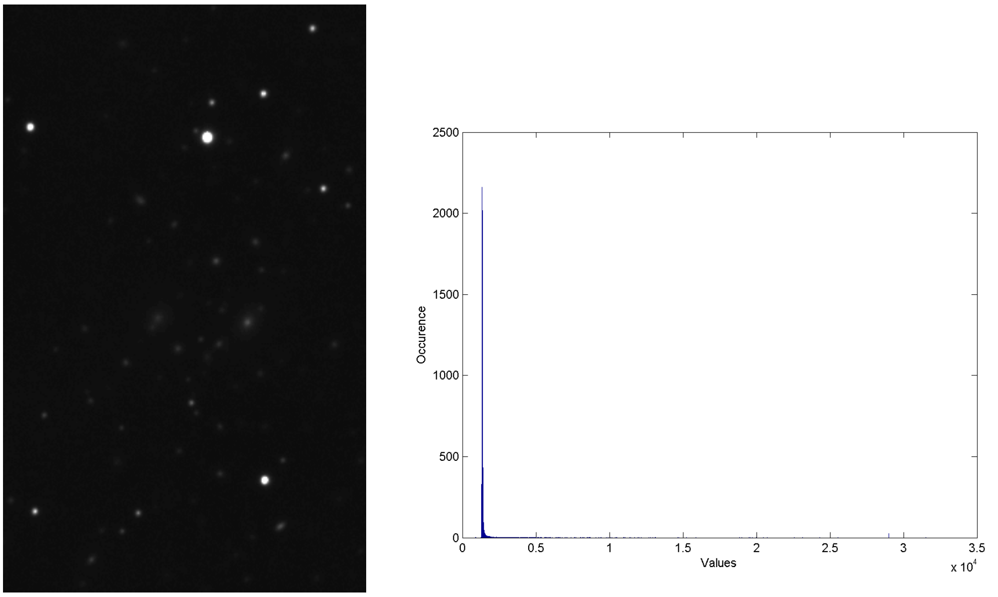


Figure C.5: Plot and histogram of image “gal0002”.

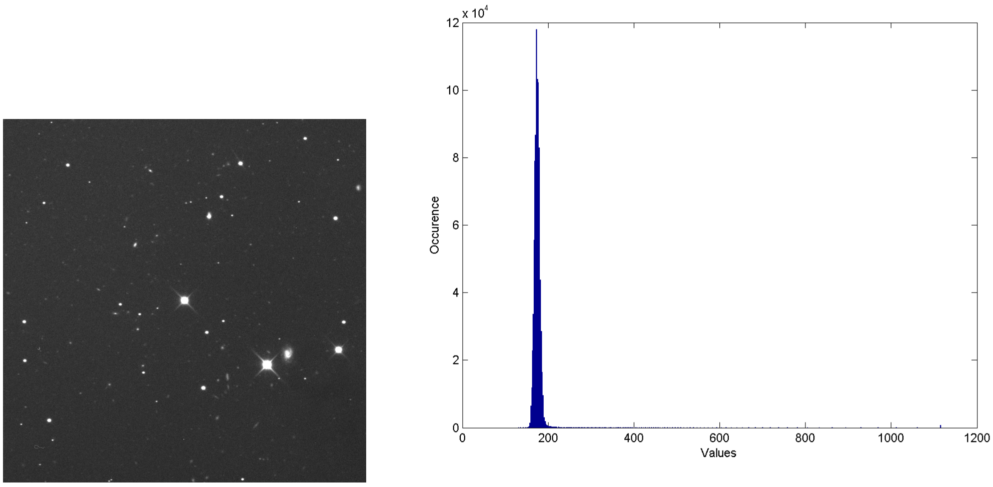


Figure C.6: Plot and histogram of image “gal0003”.

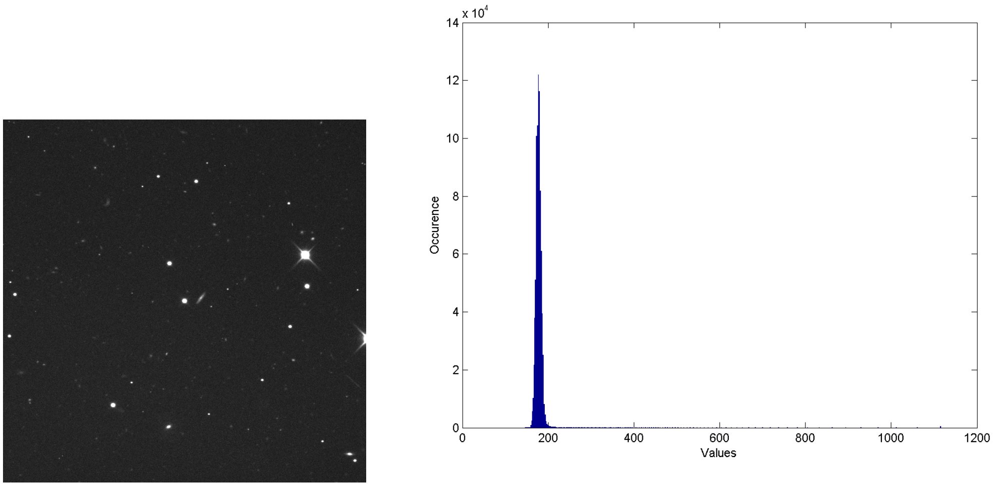


Figure C.7: Plot and histogram of image “gal0004”.

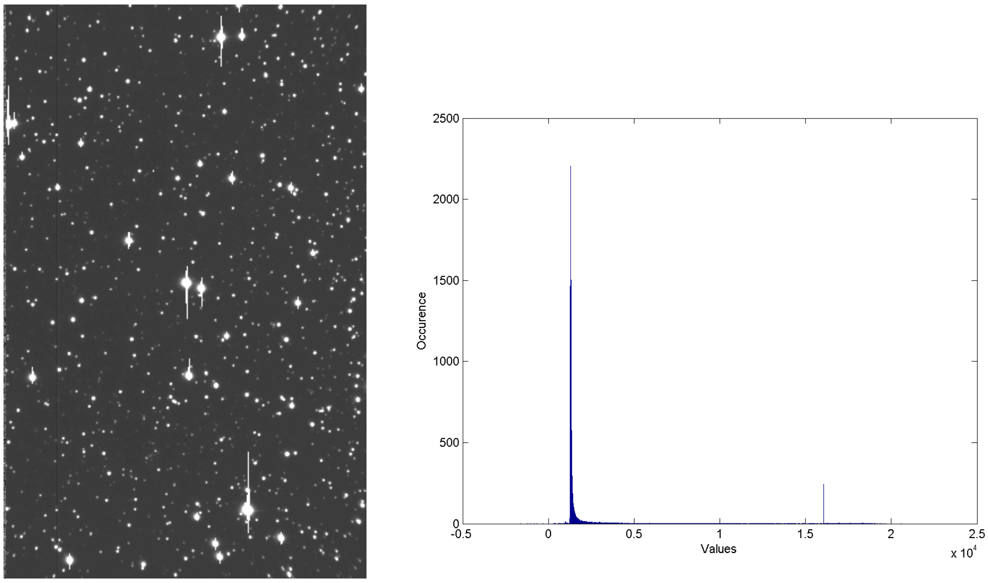


Figure C.8: Plot and histogram of image “ngc0001”.

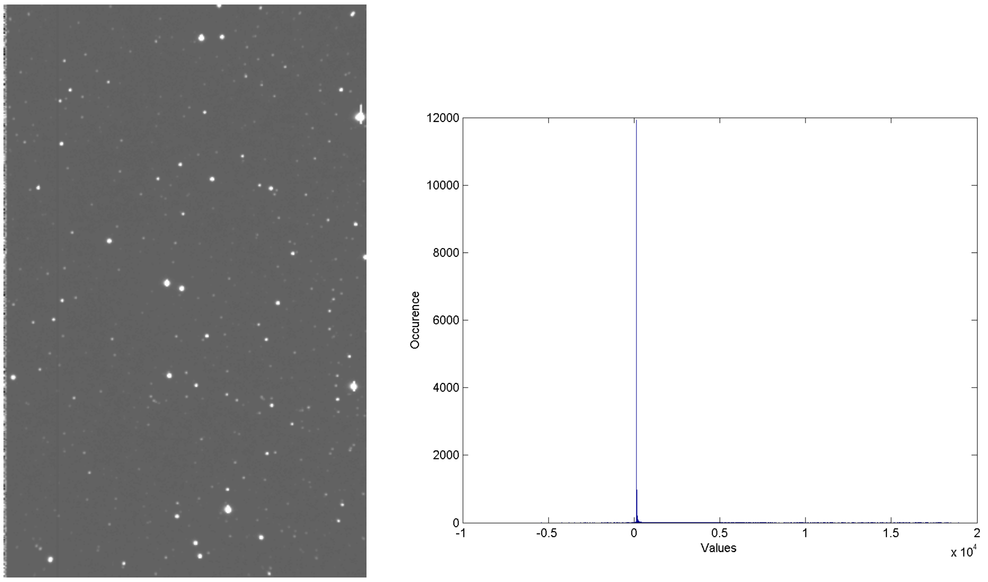


Figure C.9: Plot and histogram of image “ngc0002”.

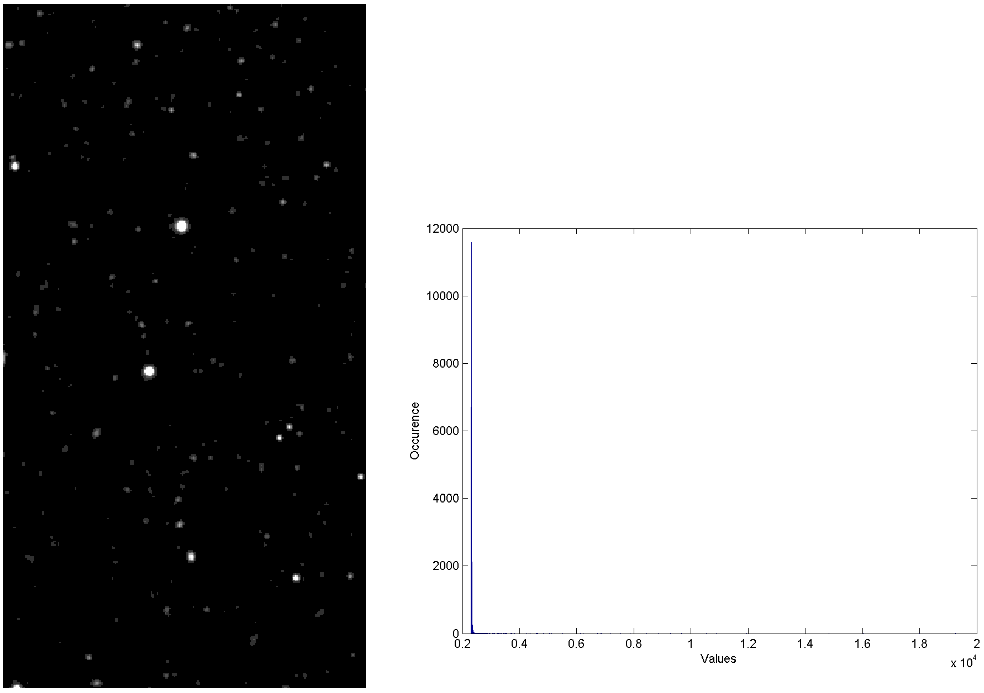


Figure C.10: Plot and histogram of image "sgp0001".

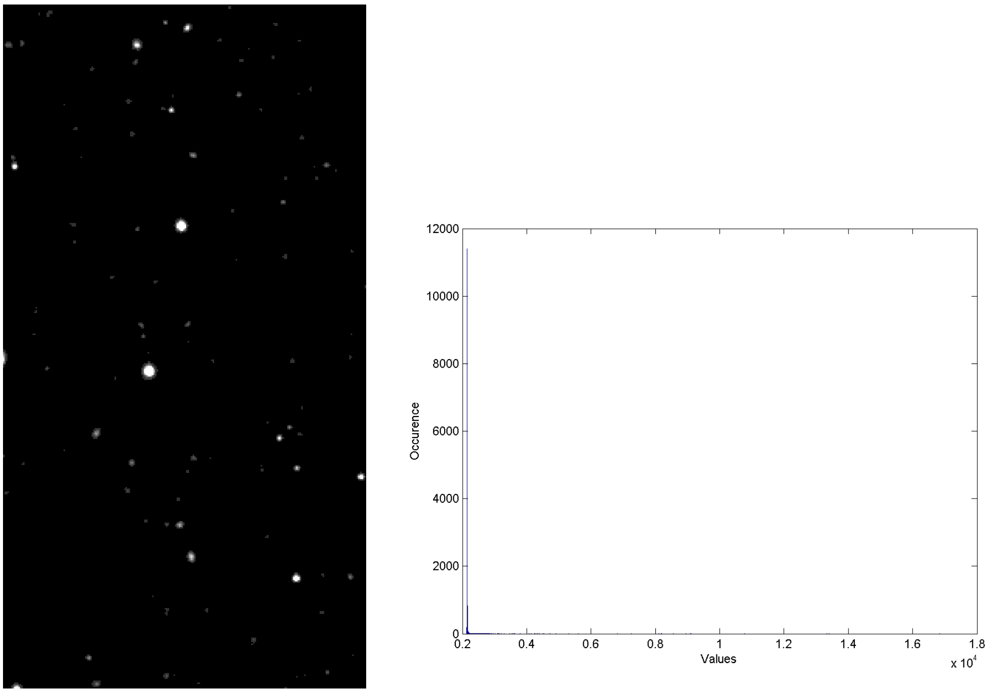


Figure C.11: Plot and histogram of image “sgp0002”.

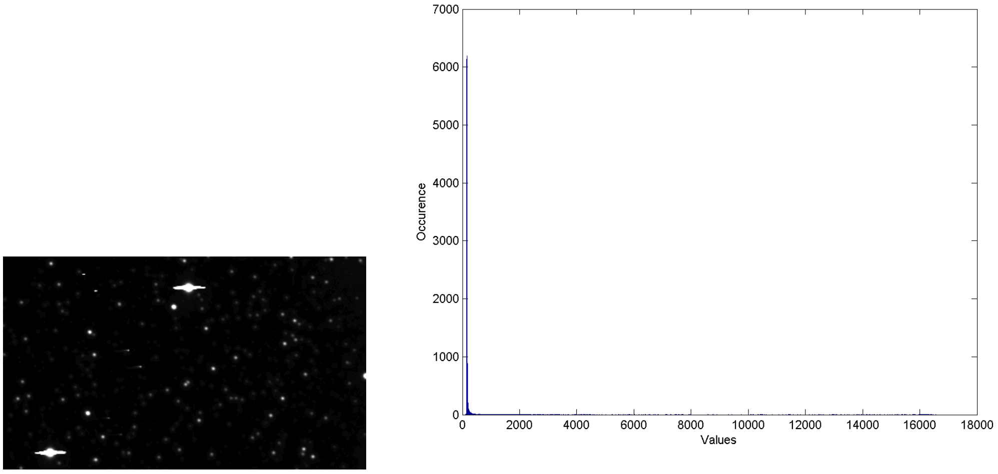
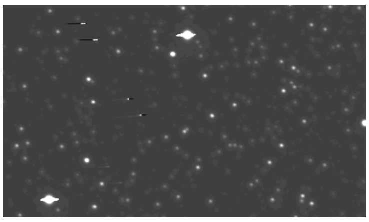


Figure C.12: Plot and histogram of image “tuc0003”.



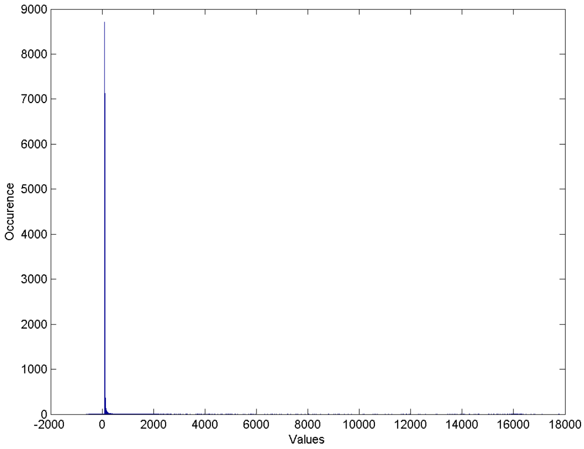


Figure C.13: Plot and histogram of image “tuc0004”.

Appendix D

Publications

Grünler, C. D., Weghorn, H., and Chibelushi, C. C. (2005). Astronomical Image Compression by Segmentation and Modelling of Image Dynamics. *In ITCS 2005 Second Annual Meeting on Information Technology and Computer Science at the BA-University of Cooperative Education.* Vol. 2 of ITCS Proceedings. Stuttgart (Germany), 14th to 17th May 2005. Stuttgart: BA-University of Cooperative Education (ISSN 1614 – 2519). pp. 14 – 18.

Grünler, C. D., Weghorn, H. and Chibelushi, C. C. (2005). Overview of Lossless Compression Methods for Astronomical Images. *In Proceedings of the Astronomical Data Analysis Software and Systems Conference XV*. Vol. 351 of ASP Conference Series. San Lorenzo de El Escoreal (Spain), 2nd to 5th October 2005. Astronomical Society of the Pacific. pp. 177 – 181.

Grünler, C. D., Weghorn, H. and Chibelushi, C. C. (2006). Compression Methods for Archiving Scientific Quality Astronomical Images. *In Proceedings of e-Society 2006.* Vol. II of IADIS Conference Proceedings. Dublin (Ireland), 13th to 16th July 2006. IADIS Press. pp. 250 – 254.

Astronomical Image Compression by Segmentation and Modeling of Image Dynamics

Christian Grünler[1,2], Hans Weghorn[1], and Claude Chibelushi[2]

[1] BA-University of Cooperative Education, Faculty of Information Technology,
Rotebühlplatz 41, Postfach 10 52 80,
D-70178 Stuttgart, Germany
[2] Staffordshire University, Faculty of Computing, Engineering and Technology,
ST18 ODF Stafford, United Kingdom
c.d.grunler@staffs.ac.uk

Abstract. Sky surveys and virtual observatories provide astronomic data at data-rates that exceed available storage capacities. Only compression of the vast data volumes will enable effective scientific exploration of the generated data. A computationally efficient method is proposed in this paper.

1 Introduction

Virtual observatories (VOs), which provide high-quality astronomy data, will make it possible for astronomers to access digital data instead of waiting until access to a telescope is granted [1]. Valuable data will be accessible to more astronomers as sources of different data sets will be linked. VOs need huge data processing and storage capacity for several reasons: the number of large telescope facilities has grown dramatically [7], the resolution of imaging devices is increasing steadily [3], and astronomical projects nowadays involve observations of several wavebands. Furthermore, survey telescopes, which are currently being planned, will image the entire sky continuously every night and therefore generate Petabytes of image data. An international VO will only become a success, if the problems of limited storage capacity and bandwidth can be solved. The research work described here targets this problem by compressing the raw images stored in astronomical archives.

2 Previous Work: Methods for Lossy and Lossless Compression

Several approaches for lossy compression of astronomical images using transform-based methods exist [13],[6],[5], but lossless compression techniques, which preserve astrometric and photometric attributes, are needed to encode precious primary data. Generally, it has to be possible to verify the results of investigations on undegraded data, especially if the results present new and revolutionary findings. Lossless compression tools, for astronomical data are quite rare. Some transform-based lossy approaches also provide a lossless mode, but still use computationally expensive transforms, which have a processor load that is too high for storing the data stream from the telescope. Three purely lossless compression methods exist: Véran and Wright [11] proposed an approach that accounts for the noise in astronomical images as it separates high-order bits from noisy - and by implication, less compressible - low order bits. Sabbey describes an adaptation of the RICE compression method [8] using linear prediction, for compressing astronomical images [9]. Finally, Weghorn et al. [12] describe a method for applying lossless compression in astronomy to interferometric data using a scheme called Signed Huffman Coding. The commonly used image format in astronomy - the FITS (Flexible Image Transport System) format [2] - currently includes only limited support for size compaction techniques. The work described here aims to develop a lossless and computationally efficient compression method by segmenting the image into regions, which are compressed adaptively by different methods. As a result, the overall-size of the stored image shall be minimized.

3 Compression by separation and region-adaptive bit allocation

Lossless compression techniques attempt to identify and exploit properties of the data for compaction [10]. The properties are described using a model of the data. One straightforward model is presented here to show the possible compression by describing the data using this model. The image-signal i(x) of an astronomical image, in general consists of three additive, independent parts: the signal s(x), which is generated by photons absorbed and detected by the light sensor, electronic or thermal noise n(x), and a constant signal offset value b, which is called bias.

$$i(x)=s(x)+n(x)+b. \tag{1}$$

The positive bias is required to avoid negative output values in the dark areas of the image, as the noise includes negative values. Ideally, the signal generated by the incident light s(x), should be large compared to the noise, for obtaining a good signal-to-noise ratio (SNR). The noise in professional astronomical photography typically influences fewer than the lower 4 bits of a 20-bit value generated by the digitizer of the imaging device. Hence, depending on the signal s(x), two image areas can be distinguished: regions of signal (ROS), where s(x) is significantly larger than n(x), and background regions, where s(x) carries values close to zero.

Pixel values along a sample line section (grey line in Fig. 1(a)) through the image are shown in Fig. 1(b), where the constant bias and the dynamic range of the noise portion are marked. Fig. 1(b) clearly indicates that the full dynamic range, provided by the FITS file format, is only required within the bright areas of the image where the signal component s(x) is large. Separating the background areas from ROS, and coding both parts differently leads to a significant reduction of the storage space. In typical astronomical photographs, the dark area, which can be coded with very few bits, fills about 90 percent of the image. A sample calculation, using the raw data presented in Fig. 1(a) shows that with a region-adaptive bit allocation, a significant reduction of the required storage space is achievable. ROS and background regions have to be identified by analysing the image signal shown in Fig. 1(b); the bias of this image is 24, while the noise portion has a dynamic range smaller than ± 8 intensity steps. Therefore, the background noise can be coded with 3 + 1 bits of the original 32 bits. Regions of signal have to be coded by the full dynamic range of 32 bits (second term in formula (2)). 79.2 percent of the image corresponds to background while 20.8 percent form the ROS. Even if a bitmap - requiring 1 bit of the original 32 bits/pixel for the whole image (third term in formula (2)) - is used to describe the different image regions, the size compaction that can be achieved in this example is nearly 3:1.

$$0.792 \cdot \left(\frac{3+1}{32}\right) + 0.208 \cdot \left(\frac{32}{32}\right) + \left(\frac{1}{32}\right) = 0.338 \tag{2}$$

The compression ratio estimated in formula (2) is achieved even without using well known compression methods. Applying compression to both regions will lead to further size reduction. Another possibility for enhancing the compression is to use more efficient methods to describe the different image regions. Such methods [4, 14, 15] could further reduce the image size by reducing the last term in formula (2). Hence, a range of standard and astronomy specific techniques, as well as techniques from second generation image compression will be examined to efficiently code the data of both areas. Therefore this method can be considered to be a version of second generation image coding.

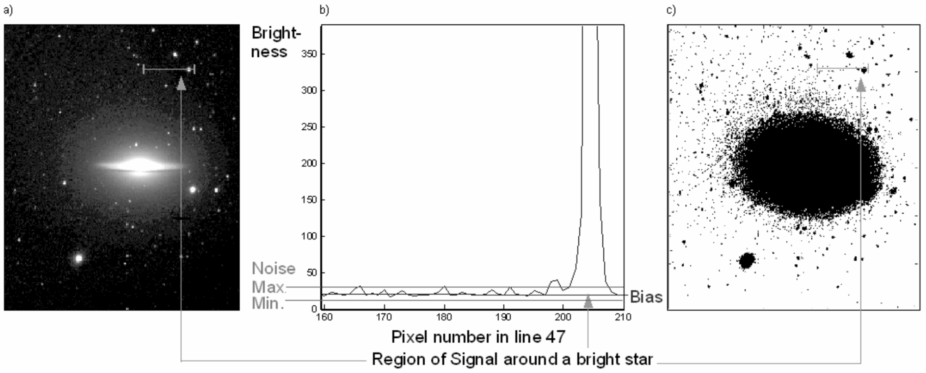


Fig. 1: a) Astronomical photograph sample. b) Signal plot of an image scan line (denoted by the grey line within the photograph given in a). c) The decision map for background noise and signal pixels.

4 Conclusion and further work

The sample calculation presented here, shows that a significant compaction can be achieved by segmentation and region-adaptive bit allocation. Future work will focus on the development of a computationally efficient compression algorithm, which determines the image bias, the regions of signal and the background area, and encodes these different parts separately. To develop a compression technique that takes advantage of image properties, first the determination of the image bias has to be automated. This work appears complicated by the fact that the bias in general has a tiled and non-uniform shape. Additionally, a range of image peculiarities have to be considered. Hence, in the first working steps, an automated compensation method for the bias will be developed before the core work on efficient compression can start.

References

1. S.G. Djorgovski, R.J. Brunner, A.A. Mahabal, S.C. Odewahn, R.R. de Carvalho, R.R. Gal, P. Stolorz, R. Granat, D. Curkendall, J. Jacob, and S. Castro, Exploration of large digital sky surveys, Mining the sky: Proceedings of the MPA/ESO/MPE Workshop, Garching, 31 July - 4 August 2000 (Heidelberg), ESO Astrophysics symposia, vol. 6, Springer, 6 2003, pp. 305 - 322.
2. W. D. Pence, FITS homepage, September 2004, http://fits.gsfc.nasa.gov/ (28.02.2005).
3. Steve B. Howell, Handbook of CCD astronomy (Cambridge observing handbooks for research astronomers), Cambridge University Press, Cambridge (UK), 2000.
4. D. Huffmann, A method for the construction of minimum redundancy codes, Proceedings of the IRE 40 (1952), no. 9, 1098–1101.
5. F. Murtagh, J. Starck, and A. Bijaoui, Multiresolution in astronomical image processing: a general framework, International Journal of Imaging Systems and Technology (1995), 332 - 338.
6. W. H. Press, Wavelet based compression software for FITS images, ASP Conf. Ser. Astronomical Data Analysis Software and Systems I (D. M., Worrall, C. Biemesderfer, and J. Barnes, eds.), vol. 25, Astronomical Society of the Pacific, January 1992, p. 3.
7. P. J. Quinn and K.M. Gorski (eds.), Toward an international virtual observatory, Springer, Berlin; Heidelberg, 2004.
8. R. F. Rice and J. R. Plaunt, Adaptive variable-length coding for efficient compression of spacecraft television data, IEEE Transactions on Communication Technology 6 (1971), no. 19, 889 - 897.
9. C. N. Sabbey, Adaptive, lossless compression of 2-D astronomical images, ASP Conference Series (San Francisco) (D. M. Mehringer, ed.), vol. 172, Astronomical Society of the Pacific, 1999, p. 129.
10. David Salomon, Data compression: the complete reference, 2 ed., Springer, New York; Berlin; Heidelberg, 2000.
11. J. P. Véran and J. R. Wright, Compression software for astronomical images, Astronomical Data Analysis Software and Systems III ASP Conference Series (D.R. Crabtree, R.J. Hanisch, and J. Barnes, eds.), vol. 61, 1994, p. 519.
12. H. Weghorn, J. Colberg, and T. Bauer, Real-time image compression method for astronomical images, Science with the VLT Interferometer ESO Proceedings (F. Paresce, ed.), Springer-Verlag, Berlin Heidelberg, 1996, pp. 399 - 400.
13. R. L. White, High-performance compression of astronomical images, Proceedings of the NASA Space and Earth Science Data Compression Workshop (Snowbird (Utah)) (James C. Tilton, ed.), March 1992, p. 403.
14. J. Ziv and A. Lempel, A universal algorithm for sequential data compression, IEEE Transactions on Information Theory 23 (1977), no. 3, 337 - 343.
15. J. Ziv and A. Lempel, Compression of individual sequences via variable-rate coding, IEEE Transactions on Information Theory 24 (1978), no. 5, 530 - 536.

Astronomical Data Analysis Software and Systems XV
ASP Conference Series, Vol. 351, 2006
C. Gabriel, C. Arviset, D. Ponz and E. Solano, eds.

Survey on the State of the Art in Lossless Compression of Astronomical Images

C. Grünler[1,2], H. Weghorn

[1]*BA-University of Cooperative Education, Faculty of Information Technology, Rotebühlplatz 41, 70178 Stuttgart, Germany*

C. C. Chibelushi

[2]*Staffordshire University, Faculty of Computing, Engineering and Technology, ST16 9DG Stafford, United Kingdom*

Abstract. Modern telescope facilities produce huge amounts of image data, but high storage costs limit possible gains from this wealth of images. Compression could help decreasing the required storage space and transmission time for an on-line distribution of images. By applying lossless compression, the costs for storage of primary data can be reduced without degrading the valuable data. An overview of existing lossless compression techniques for raw astronomical images is given here. An illustrative sample of images is used to measure the compression ratios and to verify the losslessness of selected methods that claim being lossless.

1. Introduction

With the paradigm shift from handicraft to automated mass-production during collection of observational data, extreme amounts of image data are generated and fed into the large scale database systems, in particular those of virtual observatories. Consequently, this new data-rich astronomy bears high costs for data storage and backup. For two fundamental reasons, only lossless techniques are considered for compressing primary astronomy data: Firstly, a validation of new scientific findings is only possible on undegraded data. Secondly, future and more advanced analysis methods may require high quality data, and hence the future value of such data is unknown and may be immense. The aim of this article is to summarize methods that have been proposed for the compression of astrophotographs, and which are available in terms of implemented software tools. With the latter, achievable compaction ratios are measured by applying these to a selected sample set of astronomical images.

2. Existing Lossless Astronomical Image Compression Methods

Apart from standard utilities such as "gzip", various compression methods dedicated to astronomical images have been proposed. Algortihms attuned to astronomical images could lead to higher compaction ratios than general purpose methods by exploiting as many of the special features of astronomical images as possible. An early comparison of lossless JPEG, a wavelet based method, and ZIP compression was published in (Louys, Starck & Murtagh 1999). The wavelet transform achieved slightly higher compression rates than lossless JPEG, which additionally suffered from rounding errors. With experiments on FITS data, Weghorn (2002) discussed the efficiency of standard compression methods applied to astronomical images. The overview presented below, summarizes methods identified in literature, which are claimed to work in a lossless mode.

Fitspress as one of the first astronomy specific methods is based on Daubechies wavelet transform (Press 1992). This software generates a lossy compressed image and a residual file. If both are stored, the original image can be reconstructed exactly. Fitspress stores a fraction of the brightest pixels accurately, then the whole image is compressed using the wavelet transform. The parameters are truncated and discretisized. The extracted bright pixels, their coordinates, and a fraction of the wavelet coefficients are encoded using combined run-length and Huffman coding. Finally, the residuals are computed using the inverse transform and subtracting the original image.

Hcompress, which represents another early method, is also wavelet-based (White 1992; White & Percival 1994) and provides a lossy and a lossless compression mode. The lossless mode relies on the fact that wavelet coefficients are easier to compress than the original data. For very smooth or constant image contents, wavelet coefficients may be zero over large regions (White 1992). Another transform, the *Pyramidal Median Transform* (PMT), was introduced later as an alternative (Starck 1995). Additionally, an integer based lossless compressor using *Sweldens Lifting Scheme* was proposed in (Louys et al. 1999).

Compfits is not based on any transform at all, it rather attempts to separate compressible and incompressible image bitplanes. Generally due to noise, low-order bits of any pixel intensity value are difficult to compress, and their size is likely to grow during "compression", while the high-order pixel bits can be compressed (Véran 1994). Compfits is reported to achieve compression rates comparable to Fitspress (Véran 1994). The method of Sabbey (Sabbey 1998; Sabbey 1999) is a lossless compression method, based on *Switched Linear Prediction with Rice Coding* (Rice 1971). The *Signed Huffman Coding* scheme (Weghorn 1996) exploits the distribution of the pixel-to-pixel differences in interferograms. As the focus does not lie on photographs, it is not discussd here more in detail.

A *FITS format extension* was proposed by (Pence et al. 2000) that provides a general framework of how compression methods can be included within FITS files. It is based on storing data in binary tables with variable length. Current implementations of the FITSIO library for reading and writing FITS files include three compression methods: ZIP, PLIO (pixel list I/O algorithm), and Rice. ZIP is an implementation of the Deflate algorithm, which is a combination of LZ77 (Lempel & Ziv 1977) with Huffman coding. The widely used ZIP algorithm – though not a method specifically intended for image data – is one of the best currently available general-purpose compression methods. PLIO is an algorithm for storing compressed image masks for the IRAF processing package[1]. Rice codes are a kind of entropy codes (Rice 1971).

[1] http://heasarc.gsfc.nasa.gov/docs/heasarc/fits/compress/compress_image.html

3. Evaluation Method and Results

Selected lossless compression methods are evaluated here using three sample images from the image set used by Sabbey (1999) ("ngc0001" and "sgp0001") and the MIDAS sample data ("com0001"). The evaluation is based on one objective measure, the compression ratio which is calculated as the quotient of uncompressed file and compressed file. Compression ratios have been measured for 7 methods applied to each of the 3 test images. Only those algorithms were used for the experiments, for which ready to use implementations were available.

Four software packages have been used for testing: Implementations of Fitspress and Hcompress are available in public domain, the FITSIO library provides sample programs that use its compression routines, and the MR/1 software package[2] provides wavelet based algorithms which include the PMT and Sweldens lifting scheme. As MR/1 contains a range of different methods, only the standard and lifting-based scheme methods which performed best on the sample images are discussed. Two dedicated lossless compression methods – Compfits and the linear prediction and Rice encoding method of Sabbey – could not be accessed. As Compfits is reported to achieve compression ratios comparable to Fitspress, the evaluation of Fitspress at least gives an estimate of what can be expected from Compfits (Véran 1994).

Table 1. Measured compression rates

	Hcompress	Fitspress	FITSIO GZIP	FITSIO PLIO	FITSIO Rice	MR/1 Standard	MR/1 Lifting
com0001	◇ [a]	2.03	1.14	0.97	1.92	1.92	2.11
ngc0001	◇	1.35	1.02	0.93	1.57	1.62	1.70
sgp0001	– [b]	–	2.90	1.88	4.82	◇	5.81
average	–	1.69	1.69	1.26	2.77	1.77	3.21

[a] The ◇ symbol denotes that the compression was lossy.

[b] The – symbol denotes that compression was not possible.

Table 1 shows the compression ratios, which are obtained when applying the different methods to the test images. In these experiments, the average ratios range from 1.26 to 3.21. One very important outcome of this evaluation is that not all compression methods that claim being "lossless" truly provide lossless compression, which is observed here for 3 out of 21 tested method/image pairs. The dedicated "lossless" MR/1 standard method and Hcompress did not satisfy the losslessness criteria. Hence, both methods do not represent an option for archiving original astronomical images. Fitspress could not compress all of the images, because it is limited to 16 bit/pixel files, while all FITSIO methods succeeded in compressing all images losslessly. The best performing methods, which compressed all test images correctly, are the MR/1 Lifting Scheme method and the FITSIO Rice method.

[2] www.multiresolution.com

4. Summary

At the moment, only FITSIO compression methods and the MR/1 Lifting scheme method are an option for reliably archiving astronomical images in compacted format. The most efficient method, which is practically usable, is a straightforward entropy encoder. More advanced compression methods presumably will lead to better compression ratios than this coding approach. Therefore, our current research aims to develop efficient lossless compression techniques, which exploit special characteristics of astronomical images. Such properties have already been identified. First sample calculations show that with such a dedicated method, which is based on image segmentation and adaptive coding, high compression ratios can be achieved at convenient computational costs.

Acknowledgments. We are grateful to Prof. Dr. Murtagh for his support and granting access to the MR/1 software package.

References

Lempel, A. & Ziv, J. 1977, IEEE Trans. on Inform. Theory, 3, 23, 337 – 343

Louys, M., Starck, J., & Murtagh, F. 1999, Irish Astronomical Journal, 26, 2

Pence, W., White, R. L., Greenfield, P., & Tody, D. 2000, in ASP Conf. Ser., Vol. 216, ADASS IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), 551

Press, W. H. 1991, in ASP Conf. Ser., Vol. 25, ADASS I, ed. D. M. Worrall, C. Biemesderfer, & J. Barnes (San Francisco: ASP), 3

Rice, R. F. & Plaunt, J. R. 1971, IEEE Trans. Comm. Tech., 6, 19, 889 – 897

Sabbey, C. N. Coppi, P., & Oemler, A. 1998, PASP, 110, 1067 – 1080

Sabbey, C. N. 1999, in ASP Conf. Ser., Vol. 61, ADASS III, ed. D. R. Crabtree, R. J. Hanisch, & J. Barnes (San Francisco: ASP), 8

Starck, J., Murtagh, F., & Louys M. 1995, in ASP Conf. Ser., Vol. 77, ADASS IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 4

Véran, J. P. & Wright, J. R. 1994, in ASP Conf. Ser., Vol. 61, ADASS III, ed. D. R. Crabtree, R. J. Hanisch, & J. Barnes (San Francisco: ASP), 519

Weghorn, H., Colberg, J., & Bauer, T. 1996, in ESO Proceedings: Science with the VLT Interferometer, (Springer), 399 – 400

Weghorn, H. 2002, in ESO Proceedings: Toward an International Virtual Observatory, (Springer), 327 – 328

White, R. L. 1992, in Proceedings of the NASA Space and Earth Science Data Compression Workshop, ed. Tilton J. C. 403

White, R. L. & Percival, J. W. 1994, Bulletin of the AAS, 26, 1513

COMPRESSION METHODS FOR ARCHIVING SCIENTIFIC QUALITY ASTRONOMICAL IMAGES

C. D. Grünler[1,2], H. Weghorn
[1)] Information Technology, BA-University of Cooperative Education, Stuttgart
Rotebühlplatz 41, 70178 Stuttgart, Germany
weghorn@ba-stuttgart.de

C. C. Chibelushi
[2)] Faculty of Computing, Engineering and Technology, Staffordshire University
ST18 ODF Stafford, United Kingdom
{c.d.grunler,c.c.chibelushi}@staffs.ac.uk

ABSTRACT

Most of the scientific facilities in Europe are funded from public revenue collected through taxation. For this and also due to a general interest of the broad public in outcomes of research, efficient use of these resources is required. This all applies especially to the exciting research field of astronomy. With the growing volume of astronomical data that is partially provoked by automated telescope facilities, the storage, archiving and distribution of these vast data amounts in data centres becomes more and more expensive. Image compression can reduce the data size and hence cut some of the costs. This paper introduces a lossless compression method based on image segmentation coupled to a domain-specific image data model. The proposed method exploits the distinctive characteristics of astronomical images, and aims to meet the requirements for compressing scientific-quality astronomical images. An evaluation of the method has shown that it can achieve promising compression ratios comparable to, and often better than existing methods.

KEYWORDS

Image Data Archives, Image Data Interchange, Astronomical Archives, Image Compression, Lossless Compression.

1. INTRODUCTION

Astronomy provides exciting new findings, which touch and explain fundamentals of human origin and life. Modern research in astronomy and its advance depends heavily on raw data that is collected with complex instruments like huge telescope systems or even spacecrafts. Since no single university or research institute can afford operating such facilities, independent service institutions like the European Southern Observatory (www.eso.org) and the European Space Agency (www.esa.int) were founded for operating huge telescope plants, e.g., in the Chilean desert mountains, or for performing space-based missions for observations. With expenditures in the order of a billion euro a year, these activities on the northern and southern hemisphere are financed from tax proceeds of the European Union. This all provides more than good reasons that the entire society consisting of private persons and other professional of all fields, hobby researchers, and also education institutions like schools have valid rights for having an electronic access to this valuable data. Unfortunately, astronomy faces the problem of an extreme and even accelerating growth in amounts of the collected raw data, which makes the operation of all-embracing public archives accessible by the e-society a complex problem. In particular during the recent years, a dramatic change in the generation, storage, and access of data has taken place in astronomy. Astronomical data used to be collected with a minimal degree of automation. Data of a certain object of interest was collected for a certain purpose upon demand. Now, astronomical data generation has turned into mass production. Automated survey telescopes image the entire sky every night and image data is often generated continuously in an automated process. Furthermore, data sets are available at grid-like, geographically distributed sky archives. Data for use by professional observers and the astronomical community – about half of all astronomical data in the world is at least in principle "public" (Szalay 2001) – is stored at different locations, ready to be analysed and accessed by astronomers. With this the request for certain data can be fulfilled in real-time.

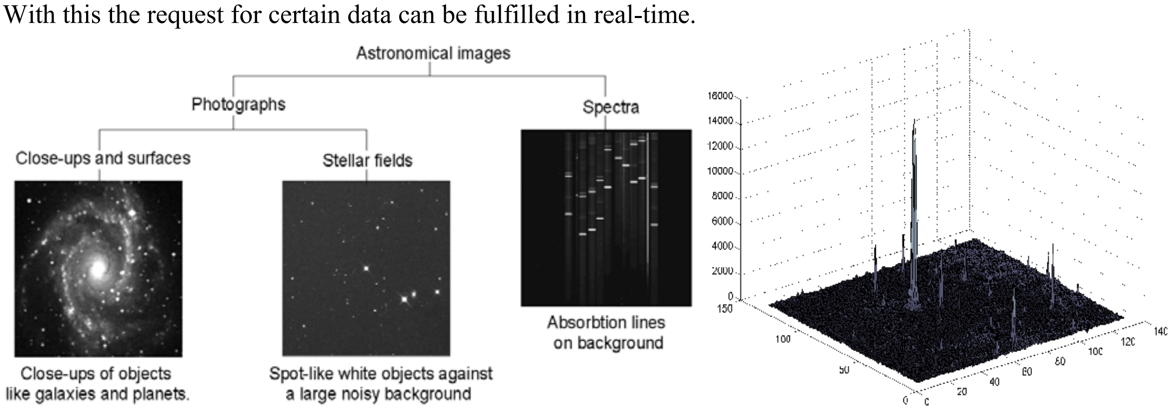


Figure 1. a) Overview on the three basic categories of astronomical images [Images courtesy of ESA]. b) A stellar field image shown as a three-dimensional plot (pixel values represented as height). [Image from the ESO sample package]

As data generation in astronomy has changed, the amount of available data is increasing dramatically. During the last 25 years, the number of camera pixels has grown over three-thousand-fold (Szalay, 2001). The data growth rate is even getting steeper, as projects nowadays involve observations of different wavebands, including gamma-rays, X-rays, optical, infrared, and radio frequencies. As the growth even exceeds Moore's law – the size of the data is doubling every year (Szalay, 2001) – despite the falling prices of disk space and network bandwidth, high annual investments for replacing and enhancing storage and network capacity have to be made. The costs for online-storage, archives and bandwidth could be reduced to a fraction of the current annual costs by using efficient compression methods.

Table 1. Overview of the differences between astronomical images and consumer market images.

	Consumer market images	**Astronomical images**
Size (pixels)	1 to 7 M pixels per CCD and camera	60 M pixels and larger for new cameras (per CCD)
Size (MB/image)	0.3 - 5 MB	Up to several 100 MB per image
Image sensor	1 Color CCD/CMOS up to 1 inch	Greyscale CCD-like image sensors, several inches
Colour	RGB(A) mode colour	Greyscale (optically band filtered)
Data type	Integer	Integer (raw images) or float
Bit depth	Typically 3 x 8 bits/pixel	Up to 1 x 64 bits/pixel
Content	Continuous tone or line art images	Stellar field photographs or spectrograms
Future use	Visual inspection	Stringent quantitative analysis
Lossy storage poss.?	Yes, even for storage of originals	Not for original images, since future value unknown
Types of practical compression methods	Rich variety of lossy and lossless transform based methods.	Some entropy coders that work on arbitrary data and transform based methods.

The requirements for compressing astronomical images differ greatly from those for the purpose of manipulating video or still images in consumer and mass market applications. Scientific data analysis usually requires high quality images. Lossless compression is required in astronomy, because the future value of images is not known and images are still obtained at high costs (Carpentieri et al., 2000). The distinctive characteristics of typical astronomical images are listed in Table 1. Astronomical images can be classified into two main groups, which are used for the three mainstays of astronomical observations: photometry, astrometry and spectroscopy (Fig. 1a). Photographs of astronomical objects collected by telescopes are used for photometry and astrometry. Spectra are used for observations of the intensity distribution of the energy emitted by astronomical objects. With these, for example extra-solar planets are going to be detected. Stellar field and spectroscopic images are suitable images for scientific explorations while close-ups are rarely used for scientific applications. A typical stellar field or galaxy image consists of a nearly flat background sprinkled with point sources and occasional extended sources (White, 1992). This is mainly caused by the fact that astronomers often try to observe a small field of view at the highest possible magnification and to obtain information about the most distant, furthest and oldest galaxies, to take a look back to the origins of Universe. Professional astronomers in general require images of high photometric accuracy, and therefore they accept only lossless compression (White, 1992). Low computational complexity of the algorithm is required because it should be possible to compress data at the same high data rate as it is collected. As the astronomy community has standardized on the FITS file format (Wells, 1981) for data interchange (Szalay, 2001), the compression method should be integrated into this standard.

2. COMPRESSION METHODS FOR ASTRONOMICAL IMAGES

A range of lossless compression methods have been published for astronomical images as well as lossy methods, which can be used for transmission and fast previewing purposes. Several comparisons of lossless compression methods have been published. Initial results on lossless astronomical image compression were published in (Louys, 1999); the performance of lossless JPEG was compared to the Haar wavelet transform and the Zip compression. It was found that the wavelet transform achieved slightly higher compression ratios than lossless JPEG, which was not truly lossless due to rounding problems (Louys, 1999). Weghorn (2002) found that general purpose compression methods lead to a significant storage space reduction for astronomical images. Nieto-Santisteban et al. (1998) found that difficulties during compression of astronomical images results from the noise. They report as well that techniques – such as Rice's algorithms (Rice, 1971) and its derivatives – probably perform better in exploiting the "almost similar value" property of adjacent pixels than other methods. A concise, partly chronological, overview and a comparison of lossless methods for com-pressing astronomical images is provided in (Grünler et al., 2005). Ready availability of implementations was used as criterion for selecting the algorithms to be compared to the newly developed method here. Hcompress (White, 1992), Fitspress (Press, 1992), the FITS IO-library methods and the MR/1 methods (Louys, 1999) were compared to the new method using a standard set of 13 test images. (http://iraf.anu.edu.au/iraf/ftp/iraf/extern/ focas.std.tar.Z). It was found that the FITS-IO Rice compression method and the MR/1 wavelet based method are the most size-efficient existing algorithms.

2.1 Image Compression by Segmentation and Signal Modelling (ICSSM)

General purpose compression programs can be used for compressing any kind of data losslessly. These tools exploit very basic properties of data, for example the probability distribution or repeating sequences of symbols. Application-specific methods exploit special properties of the data; taking advantage of them can lead to higher compression ratios – even in a lossless mode. In lossless compression, nothing is considered irrelevant and disregarded. Lossless compression techniques attempt to identify and exploit particular properties of the data for compaction (Salomon, 2000). The properties are described using a model of the data (Carpentieri, 2000). The closer the model adapts to the source, the better the redundancy resulting from signal properties can be exploited. The idea for developing a new lossless compression algorithm is to identify, and exploit as much as possible, signal properties specific to stellar fields. In contrast to general purpose data compression and general purpose image compression algorithms (typically adapted to continuous-tone grey scale images, like JPEG), a dedicated astronomy-specific method is under development by the authors.

Data de-correlation using a stellar field imaging model

In this paper, ICSSM is focussed on astronomical photographs. The underpinning idea of compression coupled to image segmentation is that stellar field images consist of a dark background, corresponding to the empty sky, covered with white stars. In general, the background of such images can be considered to be nearly constant and overlaid with electronic noise which can be modelled as Gaussian noise (Howell, 2000). Although the background is nearly constant and varies only smoothly, it is not even throughout the whole image. Due to thermal generated electrical charges (dark current) in the sensing device, the background forms an inclined plane (Fig. 1b). Stellar field photographs typically are sparse images, in which dark area, containing mainly noise, fills roughly 90 percent of the entire image (Sabbey, 1998). The background has low amplitude and the associated noise typically influences fewer than the lower 4 bits of a pixel value (Howell, 2000). With region-adaptive bit allocation, pixels can be encoded using the minimum number of bits that corresponds to the given dynamic range. A model of the image is used to segment the image into two different regions. It accounts for the image content as well as the imaging process and device, which both give rise to the special properties of astronomical images. The two-dimensional image signal i($\underline{x}$), where

$\underline{x}$ denotes the pixel coordinate, consists of three additive, independent parts. These are the signal $s(\underline{x})$, which is generated by absorbed photons emanating from stars, the imaging noise $n(\underline{x})$, and a background level $b(\underline{x})$ which is a kind of almost constant bias value. The image $i(\underline{x})$, which is read out from the imaging device, is a superposition of all these components; however the superposition is in a form that makes accurate separation into distinct components impossible. But the characteristics of the image components can still be used for an approximate estimation of the signal components, and for the segmentation of the image into different regions. The image can be segmented into two regions: The background area that contains very little or no signal, and the regions of signal, where $s(\underline{x})$ is significantly larger than $n(\underline{x})$. For a given maximal noise level N, pixels where $|i(\underline{x}) - b(\underline{x})| > N$ can be treated as signal-pixels. In background regions, only $n(\underline{x})$ and the background level $b(\underline{x})$ have to be stored.

Outline of the ICSSM compression algorithm

Using the properties identified in the last paragraph, we have developed a first implementation of a compression method based on segmentation. The segmentation step is executed automatically using a software prototype, which sequentially performs the following steps:

- Tile generation: The whole image is split into tiles of pixels for compensating first the low-frequency components in $b(\underline{x})$. With this, the background level can be considered constant $b_l = b(\underline{x})$ across each tile. The background level b_l is approximated by the mode of the tile's intensity histogram.
- Segmentation: Each image tile is segmented into region of signal $s(\underline{x})$ and background noise components. Under the assumption that the background noise $n(\underline{x})$ has zero-mean Gaussian distribution, the noise amplitude N_l for a tile, is defined as the difference between the lowest signal value and the mode of the tile's intensity histogram: $N_l = b_l - \min(i(\underline{x}))$ for all $x \in$ tile number l. The pixel assignment is derived from the comparison $i(\underline{x}) > b_l \quad + N_l$, and binary map is built up from this that indicates, to which of the two regions each tile pixel belongs.
- Encoding: The different image components can be encoded in a first approach in the following manner:

1. Background noise $n(\underline{x})$: The pixel intensity values of the background region are encoded relative to the estimated background level for reducing the required dynamic range. The background-noise data is stored in a single data stream. Huffman encoding has been adopted for the background noise.
2. Background level b_l: This scalar is encoded as binary integer value, because only one level is to be stored for each tile.
3. Region of signal $s(\underline{x})$: This component is encoded using Huffman coding.
4. The region information is encoded using a binary image.

For measuring the efficiency of the ICSSM compression approach, it is assumed that the four encoding data packages are concatenated and saved to a file, without further compaction.

2.2 Evaluation and Comparison of the ICSSM Method to Existing Methods

The main objective of the evaluation of this first implementation of the ICSSM method is to verify that – despite the basic nature of the data encoding used in ICSSM – the coupling of image segmentation and domain-specific image data model leads to compression ratios which are comparable to ratios achieved by existing approaches. The ICSSM method is evaluated here by comparing it to the best existing methods, for which implementations are readily available. To measure the compression ratio for the ICSSM algorithm, the size of the encoded data is calculated by summing up the individual sizes of all its output components. The existing methods, which have been assessed, are Hcompress, Fitspress, the FITS IO-library methods, and the MR/1 methods. The applied performance assessment measure is the image file compression ratio. For the evaluation, the set of 13 test images described in section 2 was used.

Comparing the average compression ratios, which ICSSM achieved on all test images, to the best available methods: With an average compression ratio on all images of 3.37, ICSSM comes close to the best available Rice-based methods, which achieves 3.36, and the lifting scheme integer CDF wavelet method, which reaches 3.92. On 32 bit-per-pixel images, ICSSM obtained an average compression ratio of approximately 5.8, depending on the "tilesize" parameter. This exceeds the average compression ratio of 4.69, which existing methods yield, and it comes close to the average compression ratio of 6.4, which is achieved by the best performing existing MR/1 methods.

3. CONCLUSION AND FUTURE WORK

Image compression yields a significant reduction of storage size in particular for raw astronomical data, to the benefits of producers and users of astronomical data. In addition to providing a short overview of lossless compression methods for astronomical images, this paper has introduced a compression method based on image segmentation coupled to an image data model for the compaction of stellar field images. The paper has demonstrated the feasibility of the proposed approach through experimental comparison against some existing methods. The described evaluation has shown that a significant compaction can be achieved by segmentation and region-adaptive bit allocation anchored on a domain-specific image data model. Furthermore, the proposed method obtained promising compression ratios comparable to existing methods under investigation. This result was achieved despite the fact that the here implemented, adopted data encoding is of preliminary performance only. The method presented in this paper can be combined with suitable existing compression techniques to reach even higher compression ratios. The aim of current and future research is to enhance the automated segmentation, to identify and evaluate optimal compression techniques for the individual signal components, and to extend the region-based approach to other kinds of astronomical image data such as spectroscopic data. With these enhancements, a wider range of scientific quality astronomical images can be stored efficiently, and these valuable data sets can be made available not only to professional astronomers, but also to the whole society.

REFERENCES

Carpentieri, B. et al., 2000, Lossless Compr. of Continuous-Tone Images, IEEE Proc., Vol. 88, No. 11, pp. 1797-1809.

Howell, S.B., 2000, Handbook of CCD Astronomy, Cambridge University Press, Cambridge.

Louys, M. et al., 1999, Lossless Compression of Astr. Images, Irish Astronomical Journal, Vol. 26, No. 2, pp. 119-122.

Nieto-Santisteban et al, 1998, Data Compression for NGST. ADASS Conference XV, Urbana (Illinois), USA, p. 137

Press, W. H., 1992, Wavelet based Compression Software for FITS Images. ADASS I, Tucson, USA, p. 3.

Rice, R. F., and J. R. Plaunt, J. R., 1971, Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Salomon, D., 2000, Data Compression: The Complete Reference 2nd Edition, Springer, Berlin; Heidelberg, Germany.

Szalay, A. S. et al.,2001, Large Databases in Astronomy. MPA/ESO/MPE Workshop Proc. Garching, Germany, p. 99.

Sabbey, C. N., Coppi, P., and Oemler, A. (1998), 'Data Acquisition for a 16 CCD Drift-Scan Survey', PASP, 110(751), pp. 1067 – 1080.

White R. L., 1992, High-perf. Compression of Astr. Images. NASA Compr. Workshop, Snowbird (Utah), USA, p. 403.

Wells, D.C., et al., 1981, FITS: A Flexible Image Transport System. Astr. and Astroph. Suppl. Ser, Vol. 44, pp. 363-370.

Grünler, C., Weghorn, H. and C. C. Chibelushi, 2005, Overview of Lossless Compr. Meth. for Astr. Images. ADASS XV. El Escorial, Spain, in press.

Weghorn, H., 2002, Applying Data Compression Methods in Astronomical Archives, Toward an International Virtual Observatory, Garching, Germany, pp. 327-328.