Image Reconstruction Software for Near-Field Coded Mask Instruments

Diploma thesis

Dr. Remeis-Observatory Bamberg - Astronomical Institute
Erlangen Centre for Astroparticle Physics
University of Erlangen-Nuremberg
Germany

and

Chair for Computer Science 9 - Computer Graphics
University of Erlangen-Nuremberg
Germany

and

Center for Astrophysics and Space Sciences
University of California, San Diego
United States of America

Written by: **Daniel Michalik**
born September 6 1984, Hirschau, Germany
Advisors: **Prof. Dr. Jörn Wilms**
Prof. Dr. Marc Stamminger
Dr. Richard E. Rothschild
Begin: **February 2, 2009**
End: **September 30, 2009**

Abstract

This diploma thesis is concerned with imaging of hard X-rays using coded mask cameras and the specialties of their use in near-field applications. It is focused on the experimental detector setup of the University of California, San Diego which was developed for the detection of anti-personnel mines. A general introduction of the use of coded mask cameras in far-field is followed by an analysis of the differences for near-field applications. This requires the adaption of the image reconstruction algorithms.

The second part of this thesis engages with the realization of near field algorithms and their efficient integration into a large scientific framework, the “FITS pipeline” developed by the universities of Tübingen and Erlangen-Nuremberg. Hence, a modular data flow is designed as well as some implementation work done. The thesis concludes with a summary of further steps that need to be done.

Kurzzusammenfassung

Diese Diplomarbeit beschäftigt sich mit der Abbildung von harter Röntgenstrahlung durch Coded-Mask Kameras und den Besonderheiten der Verwendung im Nahfeldbereich. Dabei steht ein Versuchsaufbau der Universität von Kalifornien in San Diego im Mittelpunkt, der zum Auffinden von Landminen benutzt werden soll. Nach einer grundlegenden Einführung in die Verwendung von Coded-Mask Kameras für die Abbildung weit entfernter Objekte werden die Unterschiede zum Verhalten der Kamera im Nahfeld aufgezeigt. Die für die Bildrekonstruktion notwendigen Algorithmen müssen dabei auf die neuen Bedürfnisse angepasst werden.

Der zweite Teil der Arbeit beschäftigt sich mit der Umsetzung einer Nahfeld-Bildrekonstruktion und der effizienten Einbindung dieser Technik in ein größeres wissenschaftliches Framework. Dazu wird die “FITS pipeline” der Universitäten Tübingen und Erlangen-Nürnberg verwendet und die notwendigen Schritte zur Integration der Bildrekonstruktionsalgorithmen untersucht. Eine Übersicht der erfolgten Implementierungsarbeiten und eine Analyse weiterer notwendiger Schritte schließen die Arbeit ab.

Contents

List of Figures

List of abbreviations

CASS Center for Astrophysics and Space Sciences

DRDC Defence Research and Development Canada

FOV Field of view

FCFOV Fully coded field of view

HEXIS Name of the coded mask detector built for MIRAX by UCSD

IAU International Astronomical Union

MIRAX Monitor e Imageador de Raios-X, planned Brazilian satellite mission

PCFOV Partially coded field of view

Declaration

This Diploma Thesis represents my own work and due acknowledgment is given in footnotes whenever information is derived from other sources. No part of this Diploma Thesis has been or is being concurrently submitted for any other qualification at any other university.

Bamberg, September 30, 2009 ______________________________

1 Introduction

X-ray examinations are a common part of everyone's daily life. A doctor will use X-ray technology for medical diagnosis of bones, weld seams of big ships and other huge metal vats are regularly checked by analyzing their X-ray images and airport security scans the traveler's baggage to search for prohibited items. All these procedures of using X-ray technology have one thing in common: They are based on a density-dependent *transmission* respectively *absorption* of the applied radiation. Hence the source will always be on one side of the examined object while a radiation-sensitive device on the opposite side is needed to detect the transmitted X-rays and their intensity.

Obviously this technology can not fit all purposes since it always needs access to two opposite sides of the object. This is not suitable for a number of tasks. Imagine a suspicious package in the corner of a room. The risk of moving it would be high since that trigger a detonation. The area would have to be evacuated and heavy armor would need to be used to ensure the safety of civilians and guards responsible for the defusing. An X-ray system working from one side only would be a good solution and improvement to current examination approaches since it would allow further inspection without moving the suspicious item. This system would have the source and the detection device on the same side and would be based on some kind of reflection.

This idea could be especially useful in the cleaning of an area where anti-personnel mines have been deployed. Most of the time the material wrapping the plastic explosives is made of plastics too. Hence common metal detectors will not be sufficient to detect them. The current approach for cleaning an area from anti-personnel mines is either to find them manually examining square-centimeter after square-centimeter or to use enforced bulldozers or heavy tanks to demolish buildings and plow up the soil, making sure no mine remains. The disadvantages are obvious: The former method takes a lot of time, while the latter leads to much unnecessary destruction. Besides the relinquishment from deploying mines at all, it would be useful to develop an X-ray detector capable of distinguishing between the ground (dirt, stones, etc.) and the plastic explosives. This idea is the main purpose behind the detector system[1] used in this thesis.

As mentioned above, regular two-sided X-ray systems are based on transmission respectively absorption, while one-sided X-ray technology has to be based on backscattering of photons, caused by Compton scattering. Since Compton scattering is dependent on the atomic number, it reveals material properties and returns a clear, photographic image of the top layers of the examined object. This is one

[1] The terms "detector", "camera" and "detector system" need to be distinguished since they are often not used in the same sense. This document refers to "detector" as a radiation- or light-sensitive area. A detector itself is not capable of real-life imaging, since that needs additional optical equipment such as lenses, apertures or, in respect to the technology used in this thesis, a multiple-hole aperture called "mask". Together they are called a "camera". A "detector system" or "X-ray system" refers to the camera and the sources used for exposing the image.

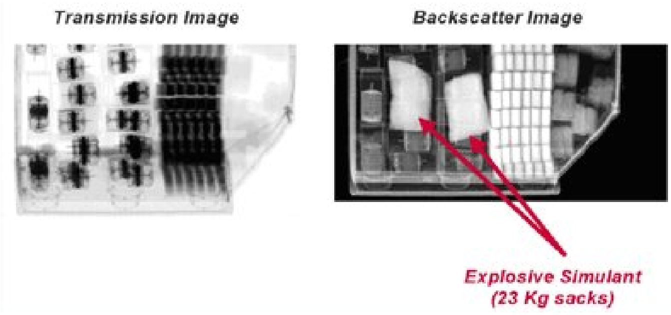

Figure 1: Improved security by backscatter imaging[2]

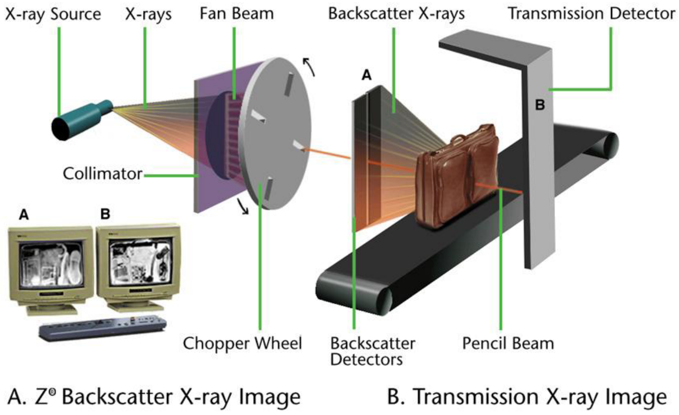

Figure 2: Imaging transmitted and backscattered radiation[3]

reason why airport security is currently testing backscattering technology as well, with astonishing results (see Figure 1).

The backscattering technology of airport scanners uses a sweeping X-ray beam to scan the baggage, illustrated in Figure 2. The object, e.g., a suitcase, is horizontally moved forward on a conveyor belt while a rotating collimator causes a vertically sweeping X-ray beam. A combination of the position of the beam and the measured intensity of both, the transmitted and the backscattered radiation is displayed and further analyzed. Information on backscatter imaging based on a rotating collimator can be found in Chalmers (2001), Chalmers (2005), Keshavmurthy et al. (1996), and Dugan et al. (2002). While this technology performs great at airports where the scanner will not be moved during operation, it is less applicable as a generic movable one-sided detector system. Under the rough requirements of in-field operations mechanical parts reduce the reliability dramatically. The time needed for the scan is another disadvantage.

The idea developed by the Center for Astrophysics and Space Sciences (CASS), a department of the University of California at San Diego (UCSD) in cooperation

[2] Courtesy American Science and Engineering Inc., `http://www.as-e.com/products_solutions/z_backscatter.asp`

[3] from Chalmers (2001).

with the Defence Research and Development Canada (DRDC), namely Dr. Anthony Faust, is to use a camera originally developed for imaging X-rays in space. It was developed for analyzing hard X-rays (above 10 keV) of celestial bodies and will fly on board of the Brazilian satellite mission "MIRAX"[4]

This camera is based on the coded mask technology and is capable of spatial resolution in hard X-rays. To use it for anti-personnel mine detection radioactive sources are distributed around the coded mask camera shining towards the object. Backscattered photons can be registered and reconstructed to an image. The final goal of this thesis is the integration of the image reconstruction into a large scientific framework called the FITS pipeline, showing an image of the examined area as quickly and accurately as possible. To achieve the best results, different deconvolution algorithms are compared in effort of implementation, usage of resources, speed, and accuracy.

- To analyze the necessary steps, the physical properties of coded mask cameras need to be understood. Section 2 explains the idea of coded mask technology by giving an overview of the historical developments from optical imaging to multi wavelength astronomy and the current technologies used for imaging in the optical, soft, and hard X-raysbands. Afterwards the mathematical background on coded mask cameras and the properties of masks are discussed. The algorithms for image reconstruction are introduced for far-field, hence showing the existing work.

- Section 3 explains the difference between far-field and near-field setups. Necessary adaptions of the reconstruction algorithms are made and improvements by background subtraction are discussed.

- Section 4 gives a quick overview of the hardware setup, followed by section 5 analyzing the software development. After comparing existing software solutions with the requirements of large scientific frameworks the data flow and integration of image reconstruction into the FITS pipeline are shown.

- Finally, the second part of section 5 focuses on the design and implementation of the software modules. Feature requirements and specific implementation details are listed and the current status of the modules is given.

- Section 6 gives a critical analysis and a final conclusion of the thesis.

[4] MIRAX (Monitor e Imageador de Raios-X) is a planned satellite mission supposed to monitor transient phenomena in the central galactic plane. It records events in soft and hard X-ray with a $\sim$ 1000 square degrees field of view and an angular resolution of $\sim$ 6 arcmin. Amongst its instruments are one soft X-ray (2–18keV) and two hard X-ray (10–200keV) coded mask cameras, with sensitivities of $\sim$ 5 respectively $\sim$ 2.6 mCrab/day. It will provide coverage of fast X-ray novae as well as long-term, in a matter of months, variable phenomena. For further information refer to Braga et al. (2005).

2 Imaging X-rays in Astronomy

2.1 Introduction

To understand the X-ray detection technology and algorithms it is useful to understand the history behind it. Hence a quick overview over observational technologies is given, with section 2.2 showing the historical development of the technology used in astrophysics, section 2.3 describing optical instruments and section 2.4 introducing the Wolter telescope for the observation of soft X-rays. Section 2.5 explains coded mask technology and gives the mathematical background as well as the image reconstruction methods.

2.2 The historical way from optical observations to multi-wavelength astronomy

This section offers a quick overview of the historical development. For more detailed information refer to Karttunen et al. (1987, pp. 47-82).

The first solution for finding out information about celestial bodies was, of course, simply observing them with the human eye. Analyzing the motion of objects in relation to each other revealed many properties: This way planets were distinguished from “fixed stars”. Astronomy based on position and proper motion of celestial objects is called astrometry and was, up to 1860, the most important form of astronomy. Improvements of the accuracy of astrometry were made in the 17th century by the development of refractor and, some decades later, reflector telescopes. To observe objects with an apparent magnitude[5] fainter than the limits of the human eye and to obtain an exact measurement of the visual movement of objects, photosensitive plates were used for long exposures. Most of the collections were preserved and are still used today, such as the plate archive of the Remeis Observatory in Bamberg (Germany) which contains over 40000 plates made between the 1920s and ∼1970. For further information on plate archives refer to Tsvetkov et al. (2005).

Until 1860 the focus of observations was only the shape, brightness and position of the object. In 1860 Kirchhof and other scientists discovered the relation between the spectrum of electromagnetic waves and the material emitting the light. From now on spectral analysis was performed in astrometrical observations to understand the molecular structure of the objects. Since those analysis all based on optical light, the telescope technology remained the same.

Only 50 years ago astronomers started to analyze other wavelengths. They quickly discovered that a complete understanding of celestial bodies requires a multi-

[5] The apparent magnitude m is an astronomical unit indicating the optical brightness of a celestial object for an observer on earth as it would have be seen without the terrestrial atmosphere. It depends on the absolute magnitude M of the star, i.e., its intrinsic brightness, and the distance d to the observer in units of Parsec: The apparent magnitude m is $M + 5(\log d - 1)$. It is given in a logarithmic reversed scale which causes a brighter object to have a smaller apparent magnitude.

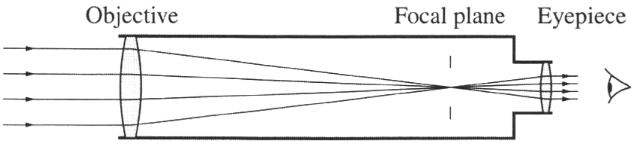

Figure 3: Illustration of a refractor telescope[7]

wavelength approach since different wavebands reveal a lot of information about the observed objects: The modern field of astrophysics was born. In a range close to the optical, conventional telescopes can be used with detectors sensitive for the wavelength to be analyzed. For increasing (respectively decreasing[6]) photon energies new technologies for focusing and imaging had to be found. To understand X-ray imaging technology fully, some basic knowledge of optical telescopes and soft X-ray technology is given before coded mask cameras are introduced.

2.3 Optical imaging

The general idea of refraction telescopes is gathering light using a lens, therefore collecting faint parallel light from a larger region than the human eye could, and focusing it. Nowadays it is of course possible to capture the image with a camera instead of the human eye. Since this does not change anything about the principle, future references will mention a human observer implying the possible use of cameras unless otherwise stated. Viewed through an eyepiece the observer gets a brighter magnified image. If the eyepiece lens is concave, the telescope is called Galilean while the Keplerian version uses two convex lenses, illustrated in Figure 3. However, refractors have two big disadvantages:

- Their chromatic and spherical aberration is relatively strong. Especially large apertures suffer from these issues but can be improved using achromatic or apochromatic objective lenses.

- The maximum aperture size is strongly limited to approximately one meter. As glass lenses will be very heavy and can only be held on their edges, gravitational effects cause the glass to sag. Above an diameter of one meter the lens will be too heavy and the gravitational sagging effects too large, therefore a different technique must be used for that aperture sizes.

To overcome those problems, curved mirrors can be used instead of lenses. Those telescopes are called reflecting telescopes. Their main idea is the collection of light using a parabolic mirror which throws the incoming light back into the direction of

[6] This thesis requires only the technological step from optical over infrared to X-rays and gamma rays while longer wavelength are not further discussed.

[7] from Karttunen et al. (1987).

[8] from Karttunen et al. (1987).

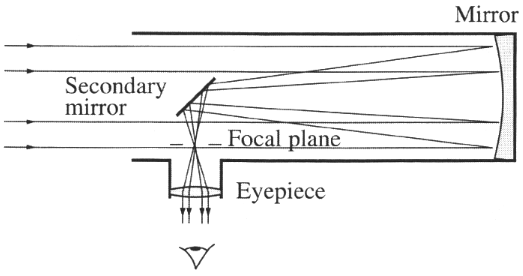

Figure 4: Illustration of a Netownian reflector telescope[8]

their origin. This is called the "folding" of the light path. A secondary mirror is used to divert the beam out of the tube and into the eyepiece of the observer or onto the detector. The shape and position of the secondary mirror as well as the exact optical design varies with the many different telescope types. Smaller telescopes for amateurs are mostly simple Newtonian, illustrated in Figure 4, or Schmidt-Cassegrain types. Other variations are used in most of the bigger telescopes.

Whatever the type of the reflector telescope is, they all have several advantages in comparison with refractor telescopes: The most obvious are structural issues. Mirrors can be held from their back and hence be very thin. Mirrors will be much lighter than a lens of the same diameter. Therefore the gravitational limits do not apply and mirror telescopes can be built with much bigger apertures. The European Southern Observatory uses mirrors with a diameter of eight meters.

Holding a mirror from its back also allows the compensation of seeing effects[9]

A second advantage is the possibility of combining smaller mirrors to form a bigger one: The Keck telescopes in Hawaii, USA uses several segmented mirrors to built up a mirror with a total diameter of ten meters.

A last and minor advantage of mirrors is the production process. While a good lens requires a completely homogeneous glass body and shaping of two surfaces, only one surface of a mirror needs to be perfect. This can be achieved by polishing it. It needs a high precession but is still much more efficient than the manufacturing of glas lenses.

2.4 Imaging X-rays below 10 keV: Wolter telescope

Mirrors are the state-of-the-art for optical imaging. If the photon energies are increased to a few keV[11], however, mirrors will eventually come to their limits. While the reflection of optical light on metallic surfaces just works, this will not be the case for shorter wavelengths. Still, reflection can be used if the angle of incidence it small enough for reflection and the material for the mirrors is well chosen. This chapter

[9] Seeing is the optical aberration caused by the atmosphere between the celestial object and the observer. It is caused by the movement of air of different temperatures. To compensate the seeing terrestrial telescopes use adaptive optics[10]. Satellites do not suffer from seeing since they fly above the earth's atmosphere.

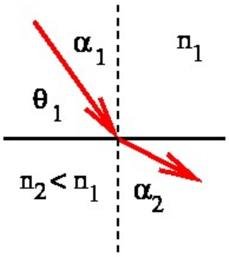

Figure 5: Snell's law

about the Wolter telescope and its physical background is based on Aschenbach (1985) unless otherwise stated.

2.4.1 Conditions of reflection

To understand the limits based on the conditions of reflection it is first necessary to understand Snell's law. Figure 5 shows a beam of light making a transition from a medium with a higher optical density to a medium with a lower optical density. The incidence angle α_1 in respect to the surface normal will be smaller than the refracting angle α_2.

$$\frac{\sin\alpha_1}{\sin\alpha_2} = \frac{n_2}{n_1} = n \tag{1}$$

If the incidence angle is above a certain limit, the refracting beam will flip to the other side of the refracting surface and become a reflecting beam. This phenomenon is called "total reflection" and is only possible for $n < 1$. The limit of the incidence angle θ_c can be derived by setting $\alpha_2 = 90°$.

$$\sin\alpha_{1,c} = n \tag{2}$$

$$\cos\theta_c = n \tag{3}$$

Note that $\cos^{-1} n$ is only solvable for $-1 \leq n \leq 1$.[12].

To eventually analyze the suitability of reflecting surfaces for shorter wavelengths it is necessary to understand the dependency of the index of refraction n from the wavelength λ which depends on the material. It is already clear that θ_c will be rather small to accomplish total reflection. To solve those equations and find the dependencies of X-rays interacting with matter, the index of refraction of the reflecting medium has to be analyzed in its complex form:

$$n = 1 - \delta - i\beta \tag{4}$$

where δ is the phase change, β describes the absorption, and $i = \sqrt{-1}$. According

[11] An electron Volt (eV) is a physical unit describing an energy. It is defined as the energy an electron gains while passing a current of one volt in vacuum, therefore causing an acceleration of the electron. (Soft) X-rays start at an energy of about 1000 eV = 1 keV.

[12] Figure 5 and derivation after Wilms (2008)

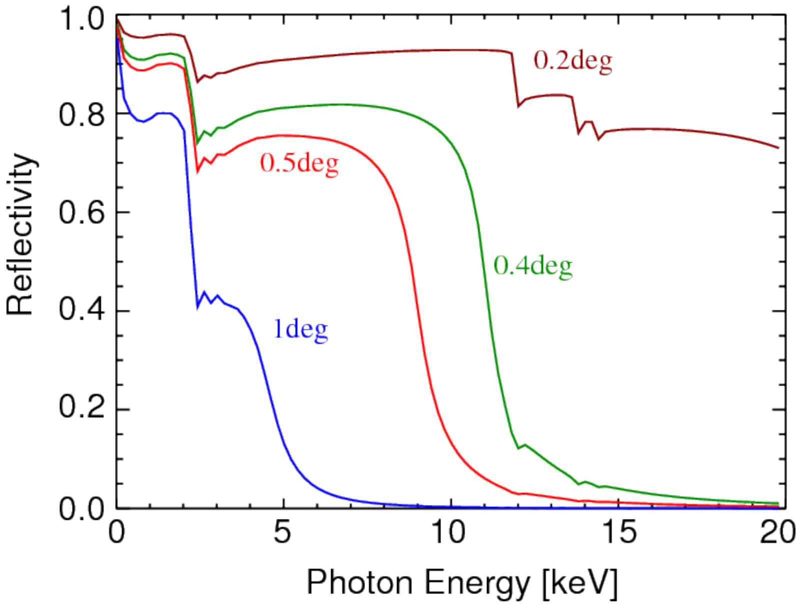

Figure 6: X-ray reflectivity of gold[13]

to Aschenbach (1985) β being negligible and combined with equation 3 results in:

$$\cos\theta_c = 1 - \delta, \text{ where } \delta = N_0 \frac{r_e Z}{2\pi A} \rho \lambda^2 \tag{5}$$

where r_e is the classical electron radius, Z is the atomic number, A is the atomic weight and ρ is the density. From $d \ll 1$, $\cos\theta_c$ simplified using a Taylor series and with $Z/A \sim 0.5$ for heavy elements follows:

$$\theta_c = \sqrt{2\delta} = 5.6\lambda\sqrt{\rho} \tag{6}$$

For wavelengths of a few Ångström this results in a critical angle of about one degree. Equation 6 shows the dependency of the critical angle from ρ which means that elements with a high density, such as gold (used for the mirror coating on the XMM-Newton satellite) or iridium (used on the Chandra satellite), have better reflection properties than others. It also shows that the angle decreases with shorter wavelength. Figure 6 illustrates the reflectivity of gold in dependency of the wavelength and the angle of incidence. It can be clearly seen that the reflection on metal surfaces such as gold works for small angles (called “grazing incidence angle”) for energies up to a certain limit. Hence a telescope design based on small angles is necessary while the focal length has to be in a reasonable range.

[13] from Wilms (2008).

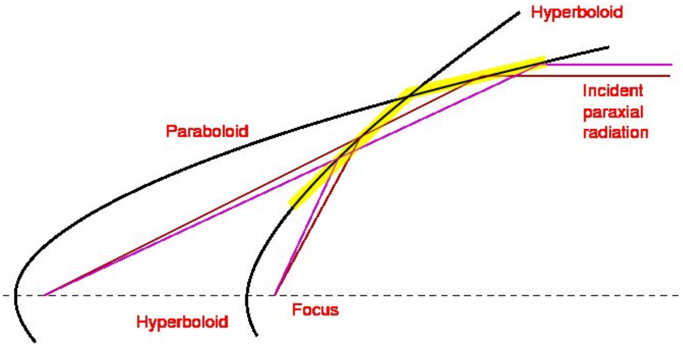

Figure 7: Illustration of the light path in a Wolter telescope[14]

2.4.2 Wolter telescope design

To achieve an imaging design for X-rays, Wolter showed that a set of hyperboloid, paraboloid or elliptical mirrors in a row can be used for focusing at a grazing incidence angle. Detailed information can be found in Kulinder Pal Singh (2005) which is the basis of this section. All three different types of Wolter telescopes use two different mirror types - the first reflection happens on a hyperboloid (type I and II) or paraboloid (type III) mirror, the second reflections happens either on a paraboloid (type I and II) or elliptical mirror. The most common type I is shown in figure 7. It can be clearly seen that although the reflection angle is very small the focal length is still reasonable, caused by the dual reflection technology. It is obvious that the light collecting area of a Wolter telescope is very limited due to the angular limits. With multiple arrangements of Wolter mirrors within each other, the collecting area can be gradually increased. This system of multiple mirrors combined to cover a bigger area is called “nested mirrors”. The manufacture and assembly requires a high precision, but the technology delivers great images and research knowledge. Hence it is the state-of-the-art for soft X-ray analysis and currently flies on the Chandra X-ray Observatory satellite (see Figure 8).

2.5 Imaging X-rays above 10 keV: Coded mask

2.5.1 Introduction

Figure 6 shows that an grazing incidence angle of 0.4 degrees or smaller would be necessary for the reflection of photons with higher energies than approximately 10 keV. This would cause a very long focal length of the instrument and therefore be totally impracticable. Hence other approaches for imaging hard X-rays must be found. Three technologies are the topics of current research: Multilayer technology, collimators and apertures with an arbitrary number of holes.

[14] from Wilms (2008, slide 3-38).

[15] Courtesy of NASA and the Chandra X-ray Center.

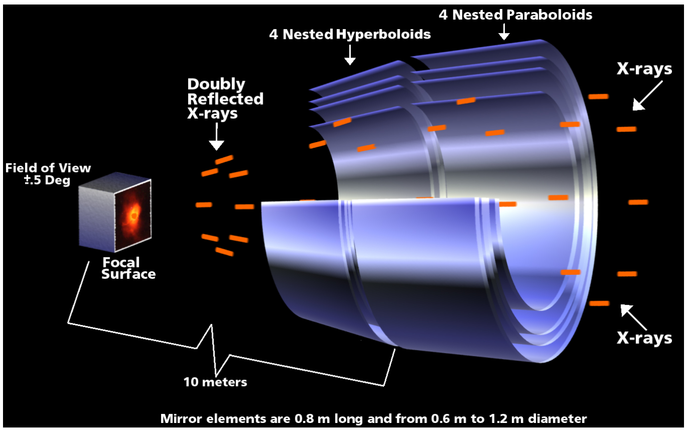

Figure 8: Wolter telescope used in the Chandra X-Ray Observatory[15]

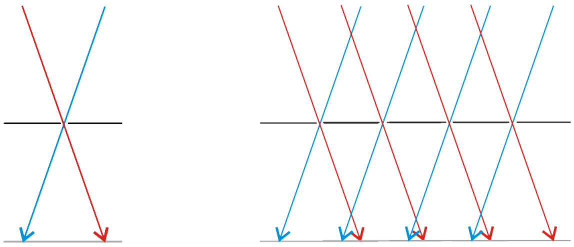

Figure 9: Pinhole and multiple hole apertures

Multilayer technologies[16] are a promising approach to manufacture mirrors capable of reflecting hard X-rays, but are not yet fully functional.

A second way is to simply not focus the incident radiation but determine the direction of the X-ray source by limiting the incident direction to a very narrow range. This is done using collimators, mostly formed like honeycombs for stability reasons. Collimators are long tubes, limiting the angle from which light can reach the detector. They have one huge disadvantage: The camera will not be capable of direct imaging. To find a source the camera has to point more or less directly into the source's direction measuring the intensity of the radiation only. Hence excessive repointing is necessary while spatial information about the source can not be gathered.

A good alternative solution are coded masks cameras, a system of a multiple hole aperture and a position sensitive detector. To understand them it is useful to understand pinhole apertures.

2.5.2 From pinhole apertures to coded mask

A pinhole aperture is a mask with a narrow transparent opening. A light source in an infinite distance in the field of view of a pinhole camera, i.e., a camera consisting

[16] Tawara et al. (2002)

of a light-sensitive detector and a pinhole aperture, will project a faint spot onto the detector. Knowing the location of the spot (i.e., the distance from the optical axis) and the distance between the aperture and the detector, it is trivial to specify the direction of the source. The narrower the pinhole is, the sharper is the spot on the detector. This makes a very accurate determination of the source's position possible, although narrowing the hole reduces the measured flux. The procedure also works for an arbitrary number of sources (or any object reflecting light). Since all photons have to pass the single hole, a laterally reversed and upside down picture is projected onto the detector, illustrated in Figure 9. Tracing them back to their origin works exactly as described for only one source. This procedure is possible with light as well as with X-rays, provided that the detector is sensitive in this waveband and the size of the hole fits the wavelength range.

Of course there is one big disadvantage of a pinhole camera: Since the sharpness of the image depends on the narrowness of the aperture hole, the intensity of the projected spot will be very small. This results in a low signal to noise ratio and very long exposure times. Furthermore only a very small part of the detector's surface is used if a single source is observed. To compensate for this problem, the aperture can have more than one hole, shown in Figure 9. Such an aperture is called a "coded mask", the whole mask-detector-system is called "coded mask camera". A single source will then project one spot for each hole, therefore making a better use of the detector area. Still, the source can be located using the shift of the projected pattern and the distance between the detector and the mask.

If more than one source is present in a detector-mask system this will result in an overlay of the projected pattern of each source on the detector. The combined projected pattern recorded on the detector is called a "shadowgram". It is also obvious that tracing the projections back to the original source is non-trivial and requires mathematical post-processing of the shadowgram. Hence coded mask cameras are a part of "computational photography". Further information can be found in Anderson et al. (2006).

The advantages of coded mask technology are obvious: Since (almost) the whole detector area is used to form the shadowgram, single inaccuracies such as a hotpixel on the detector do not take much influence on the results. Also the flux recorded for each source is much higher compared to a single hole exposure. Some further information about the energy range of the source can be determined if the detector is capable of recording the energy of single photons.

Two different forms of coded mask cameras are distinguished: Figure 10 shows a camera, where the mask area equals the detector area. This is called a "box camera". Only sources in the direction of the optical axis will project the mask pattern onto the whole detector surface (The region for sources fulfilling this statement is called the "fully coded field of view", FCFOV). Sources that are off-axis will project a part of the mask pattern that will cover the detector partially. (The region of sources is called the "partially coded field of view", PCFOV). Reconstruction for sources in

[17] from Groeneveld (1999).

[18] from Groeneveld (1999).

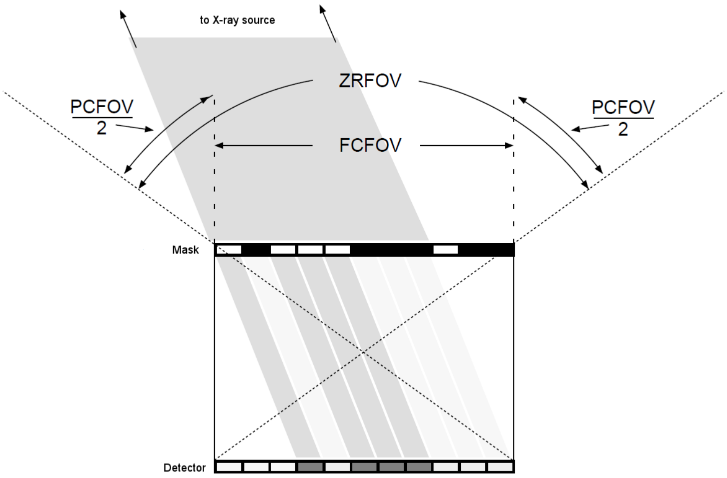

Figure 10: Illustration of a box camera[17]

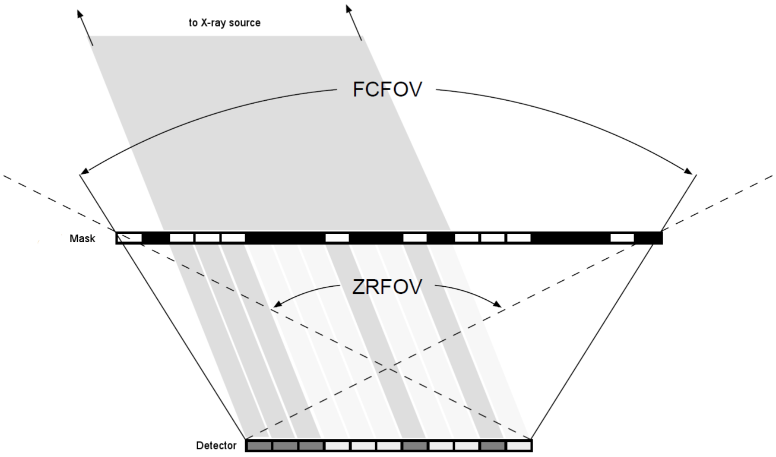

Figure 11: Illustration of a mosaic camera[18]

the PCFOV performs less than reconstruction for sources in the FCFOV. Hence, for most coded mask cameras the mask covers a larger area than the detector, forming a layout as it is shown in Figure 11.

2.5.3 Image reconstruction

To understand the basic process of reconstructing the location and intensity of sources some mathematical background is needed. This section is based on Groeneveld (1999, pp.18ff.) and Skinner et al. (1987) with some simplifications made.

Let $D(\vec{x})$ represent the number of photons that hit the detector at the pixel position $\vec{x} = (x, y)$. For now it is assumed that the camera system is perfect and every photon originated from a source, passed through the mask and hit the detector plane without creating a split event[19]. Background radiation and any kind of noises are assumed to be negligible for this simplified introduction (in reality, background plays a significant role, see, e.g. Groeneveld). $M(\vec{x})$ refers to the opacity of the mask, with pixels assumed to be either fully transparent or completely opaque and the mask itself assumed to be perfectly flat. Hence dealing with partially transmitted photons is avoided. Finally, let $S(\vec{x})$ describe the intensity distribution of the sources.

Thinking of the mask as a coding function of the intensity distribution given by $S(\vec{x})$ the detector data can be calculated using the convolution operator $*$

$$D(\vec{x}) = M(\vec{x}) * S(\vec{x}) \tag{7}$$

To reconstruct a good approximation $\hat{S}(\vec{x})$ of the original image from the shadowgram and the mask information, a decoding function $G(\vec{x})$ is introduced:

$$\hat{S}(\vec{x}) = G(\vec{x}) * D(\vec{x}) = G(\vec{x}) * (M(\vec{x}) * S(\vec{x})) \tag{8}$$

If the decoding and the coding are assoziative, then $G(\vec{x}) * M(\vec{x}) = 1$. Hence equation 8 resolves to $\hat{S}(\vec{x}) = S(\vec{x})$, which means the reconstructed image equals the original intensity distribution of the sources. For real-life applications noise and background radiation will cause the result to be less exact.

Since $M(\vec{x})$ is usually not invertible, $G(\vec{x}) = M(-\vec{x})$ is chosen instead. Inserted into equation 8 the resulting equation is

$$\hat{S}(\vec{x}) = (M(\vec{x}) \otimes M(\vec{x})) * S(\vec{x}) \tag{9}$$

where $M(\vec{x}) \otimes M(\vec{x})$ represents an auto correlation function of $M(\vec{x})$. For masks that are a finitely extended functions of independent random values, the auto correlation results in a very sharp peak where $M(-\vec{x})$ equals the convolution inverse of $M(\vec{x})$. For an infinitely extended function the result would be a delta-function with an exact peak of 1 for $\vec{x} = \vec{0}$. Since a real mask is limited to a certain area the reconstruction

[19] A split event is an event by a photon impacting at the edge of or between two pixel(s) and therefore causing simultaneous hits in two or more consecutive pixels. They have to be combined before further processing of the events can be done.

has side lobes present at $\vec{x} \neq \vec{0}$. Therefore the auto correlation of a mask can be seen as the point spread function[20] of the camera.

2.5.4 Masks

A mask is divided into pixels. Each pixel can be either open, i.e., fully transparent or closed, i.e., opaque. The ratio between open pixels and the complete number of all pixels is called the "opening fraction" (OF). To allow efficient post-processing it is important to know the exact size and shape of the mask pixels. Mostly they are either square or hexagonal with typical sizes between a tenth of a millimeter up to a few centimeters.

It is also important to understand that ideal masks only work for objects within the FCFOV. Objects in the PCFOV will project an incomplete cut-out of the mask pattern, therefore causing more noise during reconstruction. Hence all common coded mask systems are mosaic cameras and might limit the field of view to the FCFOV by using additional collimators.

In subsection 2.5.3 it was concluded that reconstructing the image from the shadowgram created by an arbitrary mask using $G(\vec{x}) = M(-\vec{x})$ will result in a main peak and a number of side lobes. The ratio between those two is a good measurement for the quality of a mask pattern. Although the experiments of Skinner (1995) showed, that imaging with bad masks, e.g., shaped like a bicycle, is possible, it is useful to understand the creation of good patterns.

A good mask pattern is given by a periodic aperture which aligns all side lobes to an equal level, therefore making it easy to distinguish a sharp peak from an even plateau. Therefore the aperiodic parts of a random mask need to be substituted by a periodic pattern. Very good results are achieved using prim numbers, e.g., Modified Uniformly Redundant Array (MURA) masks, introduced by Gottesman and Fenimore (1989). For various possible mask layouts and there quality measured by the auto-correlation functions see Skinner and Rideout (1994).

To improve the signal to noise ratio a combination of mask and anti-mask images can be used. The anti-mask is the exact opposite of the mask: Wherever the mask has an open pixel, the anti-mask has a solid one, and vice versa. If an exposure of the same object is done with the mask first and with the anti-mask afterwards, the combination of both images lead to significant improvements. The summation of the two images reduces noise and artifacts while doubling the strength of the signal. For further information about the physical background refer to Faust et al. (2003) and Accorsi (2001).

2.5.5 Image formation and reconstruction methods

Various methods for image reconstruction are possible. Most often Fast Fourier Transformation or back projection is used. These two methods are described in this

[20] The "point spread function" describes how an imaging system responds to a point source. It is a measurement of the blurring of a perfect source in the captured image.

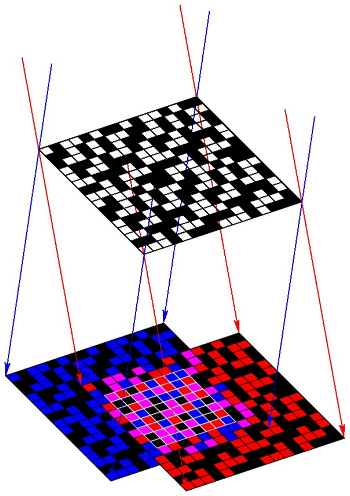

Figure 12: Shadowgram formed by two sources[21]

section. For further general information on reconstruction techniques see Skinner et al. (1987).

2.5.5.1 Convolution using Fast Fourier Transform

From equation 8 it can be seen that the convolution of the recorded shadowgram $D(\vec{x})$ with the reversed mask $M(-\vec{x})$ solves the reconstruction for the strongest present source. The convolution itself can be done efficiently using the Fast Fourier Transform with the kernel being the convolution of $M(-\vec{x})$. The Fast Fourier Transform will then result in a matrix showing the quality of the overlay of the mask and the shadowgram with a peak for the direction of the source.

If more than one source is present, as illustrated in Figure 12, the described procedure still works. The shadowgram of the sources will overlay, still the stronger source can be found using FFT. After finding the most likely position of one source the projected mask pattern is subtracted from the shadowgram data. Assuming that at least some pixels were only hit by that one source the minimum intensity found is the intensity to be removed therefore leaving flux coming from other sources untouched. After the removal of one source the describe procedure can be applied again to find the second-most intense source. This is called the “Iterative Removal of Sources”[22].

2.5.5.2 Image reconstruction using back-projection

Another approach to image reconstruction is to revert the accumulation of intensity on the detector plane. Every photon counted must have come through an open mask

[21] Courtesy: Image gallery of the ISDC Data Centre for Astrophysics, `http://isdc.unige.ch/bin/images.cgi?ALL`.

[22] Hammersley et al. (1991, pp. 589ff.)

pixel from a source far away. Therefore a virtual sky covering the FCFOV can be introduced and rays originating from each detector pixel can be traced back onto this plane. Obviously there is no information to determine which mask opening the photon passed on its way to the detector. Hence it is useful to implement a sub-photon and throw it back through all mask openings. It is useful to set its initial intensity according to equation 10 to ensure that the total energy of the system will not be increased through the reconstruction algorithm.

$$I_{\text{sub-photon}} = I_{\text{photon}}/\text{number of(mask openings)} \tag{10}$$

(Side note: Another possible approach is to project the photons from the detector through each mask opening using the full photon energy. After iterating through all photons an even background is subtracted from the virtual sky.) Once the ray hits the virtual sky plane the struck sky plane pixel needs to be determined. This can be simply done by intersecting the plane and the rays. The sky plane pixel accumulates the energy, its value will be increased by the incident photon's energy.

To obtain a satisfactory result a few more issues have to be considered:

- For most cameras, the detector will be smaller than the mask, usually about half the size. This is necessary to obtain a larger FCFOV, otherwise even sources only slightly off the optical axis will only partially expose the detector. Since the virtual sky should cover at least the whole FCFOV and needs to be much further away than the distance between mask and detector the plane should be curved. This can be done by three steps: The ray is created as before using two points, the originating detector pixel and the mask opening. Then the three-dimensional Cartesian coordinate is calculated using the distance of the spherical cutout depicting the virtual sky. These coordinates are transformed by a spherical projection into a pixel position on an equidistant grid.

- If a photon is always assigned to one detector pixel, information about the exact position of the hit is lost and cut down to integer accuracy. Especially split events are not taken into account well enough. This requires the back-projection algorithm to either throw all photons back originating from the center of the pixel, hence neglecting particular possible source positions. On the other hand the area of the whole pixel could be projected through the mask opening, causing a widening of the source position but giving a better probability to reach the correct source position on the virtual sky.

- The same problem applies to mask pixels as well. The exact transition point within the mask pixel is not known. If the center is assumed the reconstruction might miss the relevant portion of the virtual sky. Again the whole area can be used for the projection with the same disadvantage: Sources are further widened. An illustration is given in Figure 13. From this discussion it can be seen, that a large number of photon hits is needed to achieve satisfying results with coded mask imaging.

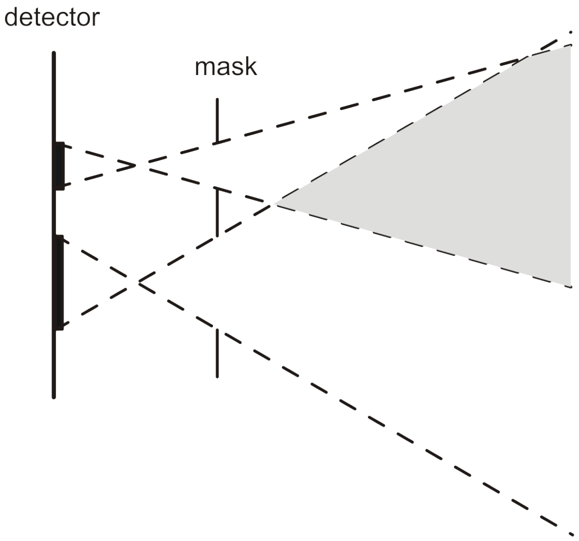

Figure 13: Widening caused by back-projection in far-field. The grey area illustrates the segment wherein the source is located

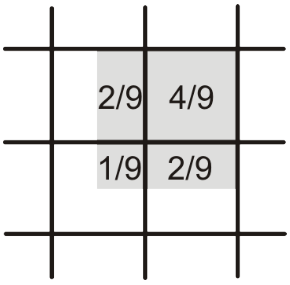

Figure 14: Coverage of multiple detector pixels and their area ratio

- Taking both areas into account results in a wide area projected onto the virtual sky. Since it is likely that more than one sky pixel is hit the intensity has to be further distributed. For a good solution the coverage ratio should be used as the factor for the intensity of each sky pixel, illustrated in Figure 14. The consideration of all those areas leads to a high programming complexity and mistakes can be easily made.

The runtime complexity adds up as follows: All photons of one detector pixel can be projected onto the virtual sky at once with an energy integrated over all photons that hit this particular detector pixel:

$$I_{\text{super-photon}} = \int I_{\text{photon}} \tag{11}$$

$$I_{\text{sub-super-photon}} = \frac{I_{\text{super-photon}}}{\text{numberof(mask openings)}} \tag{12}$$

With both areas (of the detector and the mask pixel) taken into account four rays per pixel pair are needed - one originating of each corner of the detector pixel and going through the diagonally opposite corner of the mask pixel, assuming square pixels. The cost of each ray leads to a high constant factor, still the complexity of a back-projection algorithm is only dependent on the number of detector pixels d and open mask pixels m resulting in $O(d \cdot m)$.

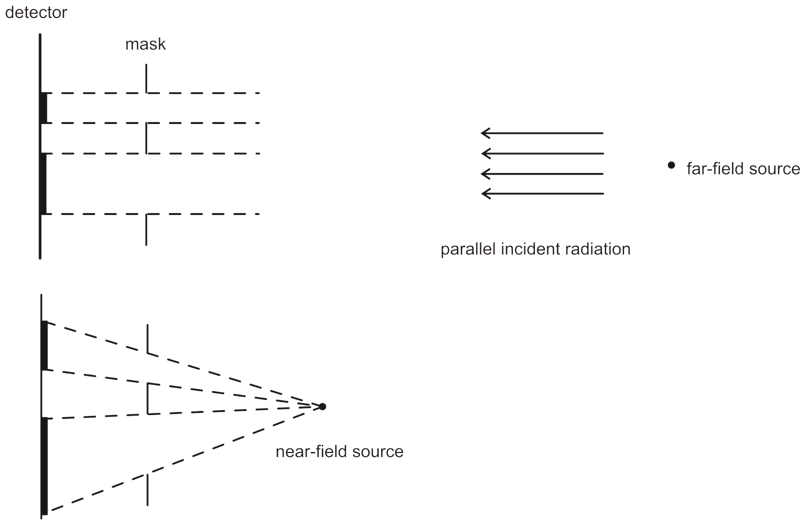

Figure 15: Parallel vs. radial emitted light

3 Near-field X-ray imaging

3.1 Introduction

After the far-field reconstruction was understood, section 1 introduces the basic motivation for using coded mask. Section 3.3 and 3.4 explain the algorithmic adaptions, followed by the idea of background subtraction in section 3.5.

3.2 Magnification

To understand the difference of imaging in far- and near-field very simple physics are needed: Light emitted by objects which are very far away is considered to be parallel. Hence an exact copy of the mask pattern is projected onto the detector plane.

Imaging in near-field works different: A point source close to the camera will emit light radially and therefore cause a magnification of the projected mask pattern. Since the image reconstruction algorithms described earlier in this document are not scale-invariant they have to be extended to take this into account.

The magnification factor M depends on the distance of the object from the mask $d_{\text{mask,object}}$ and the distance between mask and detector $d_{\text{detector,mask}}$:

$$M = \frac{d_{\text{mask,object}}}{d_{\text{detector,mask}} + d_{\text{mask, object}}} \tag{13}$$

3.3 Fast Fourier Transform

Even in near-field measurements FFT based methods can be used to reconstruct an image. To take the magnification into account the inverted mask pattern used for the convolution has to be magnified. To determine the magnification factor it is necessary to either know or guess the approximate distance of the object.

The easiest solution is to just determine a valid range of distances, e.g, a minimum distance of the object from the mask which can be increased up to infinity and step through the magnification factors between the limits. This requires three parameter to be set: The minimum distance of the object from the mask, the maximum distance and the delta for one step in the magnification factors. For each magnification factor the fourier transform has to be done and the best match is chosen. It is obvious that the usual complexity of a Fast Fourier Transform of $O(n \cdot \log(n))$ will be increased by the number m of magnification factors to be taken into account. Hence the complexity increases to $O(m \cdot n \cdot \log(n))$.

The complexity issue can be solved by using binary search. Binary search is an algorithm to locate an object in a sorted list. Further information can be found in Cormen et al. (2001) and other algorithmic compendiums. Binary search starts by comparing the middle element of the list with the object to be searched. If the object is smaller than the middle element, the search continues by doing the same step again using only the bottom half of the list. Otherwise the recursion continues with the upper half. This is done until the element is found or it is determined that the element does not exist. The complexity for binary search is $O(\log(m))$ for the worst case scenario with m being the number of elements in the list.

A similar idea can be used for finding the optimal distance without calculating FFT for all of them. The Fast Fourier Transform of the increasing distance between object and mask is seen as the sorted list. The optimum is the Fast Fourier Transform for the one distance where the best match, e.g., the highest amount of one frequency was found. This is equivalent to the determination of the contrast of the image represented by the results of the convolution. For a description of contrast evaluation methods refer to Peli (1990).

Since it is unclear whether the optimal result will be in the right or left half of the search domain, binary search has to be adapted to use two middle elements instead. Finding out the direction in which the results improve tells which half needs to be pruned. The complexity of the whole algorithm including the Fast Fourier Transform adds up to $O(n \cdot \log(m) \cdot \log(n))$.

3.4 Back-projection algorithm

Another approach is the adaption of back-projection. The main idea is identical to the back-projection reconstruction method for far-field coded mask imaging (see section 2.5.5.2): Starting from every hit pixel's corner, rays are traced back through the diagonally opposite corners of each open mask pixel. At the intersection of each ray with the plane symbolizing the virtual sky, the intensity will be increased by an appropriate fraction dependent on the total number of open pixels in the mask.

At first the magnification factor in near-field situations seems to be a big problem for the back-projection algorithm. The radiation projected by a close-by source through a mask pixel is widened by the magnification factor of the system. Hence

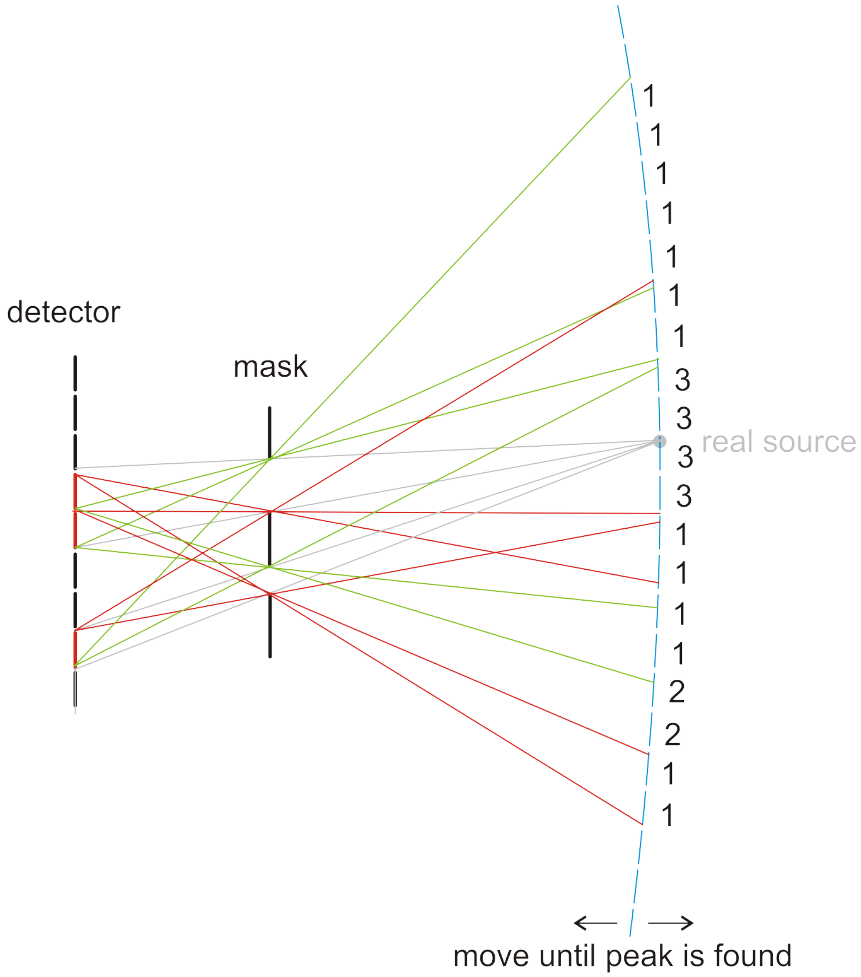

Figure 16: Illustration of back-projection in near-field

the back-projected area will be even more widened. To understand that this issue has no problematic effect, it has to be kept in mind that coded mask cameras only work if a large number of photons are captured by the detector. Therefore the overlay of the widened back-projected areas will still give a correct spot of the source within the limits of the coded mask cameras' resolution. A one-dimensional illustration for a mask with two holes is shown in Figure 16. This mask is sufficient to locate the source to an area of four pixels.

As a conclusion the back-projected algorithm can be mainly taken without any changes except that additionally the distance of the virtual sky has to be determined. This can be done analogue to the possibilities described for the determination of the distance of the virtual sky in the case of the FFT algorithm: The correct distance can be set manually or chosen automatically by contrast detection.

3.5 Background subtraction

There was also the concern that near-field imaging suffers from further artifacts caused by the surroundings of the detector setup. Photons are scattered on the shielding or might "reflect" on more than one surface before hitting the detector. Two approaches are follow to minimize the influence of those effects:

1. The exposures can be done without the lead bricks in front of the detector. Photons that would normally be scattered back by the shielding bricks are distributed into the air in front of the detector, therefore causing radiation in the laboratory instead of adding noise to the final image.

2. A background could be recorded and subtracted from the image. Therefore a total of four exposures is made: Two (mask and anti-mask) without an object in front of the detector-system and two (mask and anti-mask) regular exposures of the object. After combining both background images and both object images the final image is composited by the subtraction of the background from the image. This was not done so far, since the problem does not apply to space. This thesis proposes to take this idea into further research, hence considering this case in the software design.

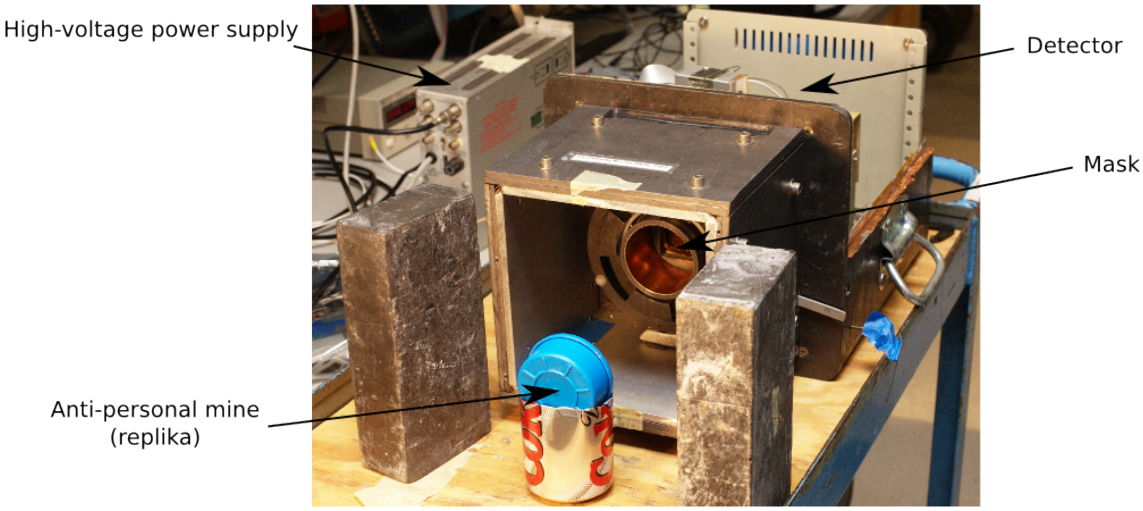

Figure 17: Complete hardware setup at UCSD

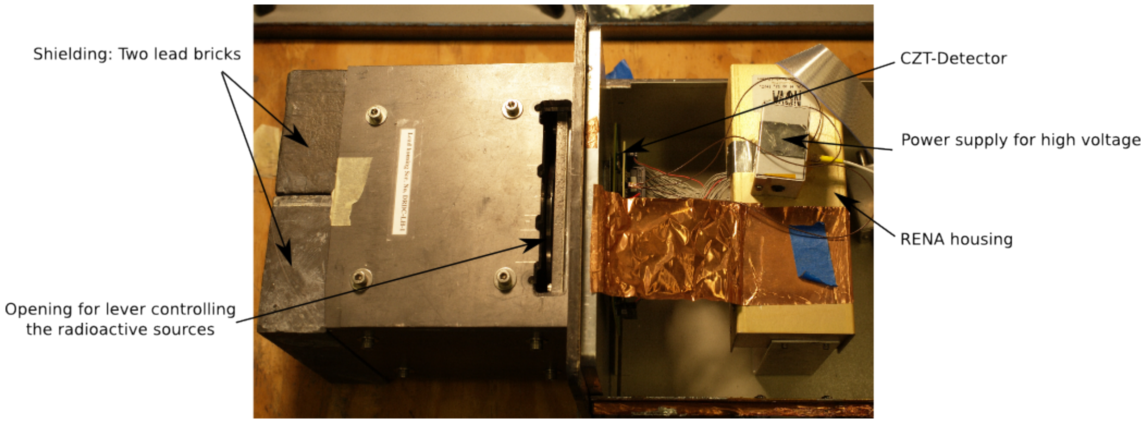
Shielding: Two lead bricks
CZT-Detector
Power supply for high voltage
RENA housing
Opening for lever controlling
the radioactive sources

Figure 18: Top view: Detector and RENA modules

4 Experimental setup - hardware

4.1 Introduction

The laboratory setup at the UCSD consists of a lead box housing the radioactive sources, the mask, the detector and the RENA3 hardware to control and read-out the detector. Figure 17 shows the complete hardware setup including an inert anti-personnel mine in front of it. The exposure is made with the object close to the detector housing's opening with the shielding bricks removed. Figure 18 shows a top view of the detector and its power supply as well as its RENA modules forwarding the gathered data to a computer for further processing. The side view shown in Figure 19 gives an idea of the detector width of approximately 1/4 inch. The detector used is a CZT detector capable of recording single photon hits and their energy. This is done using an anode-cathode-design. The mask can be seen in Figure 20,

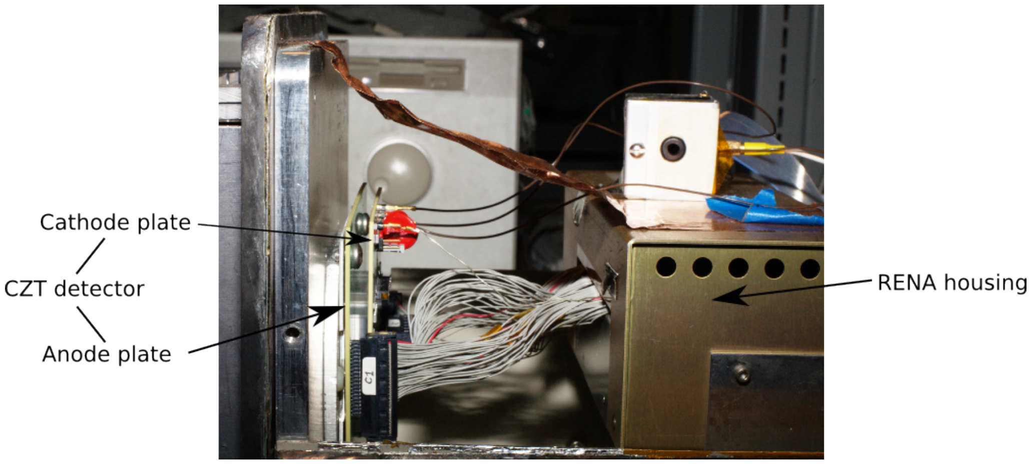

Figure 19: Side view: Layered structure of the CZT detector

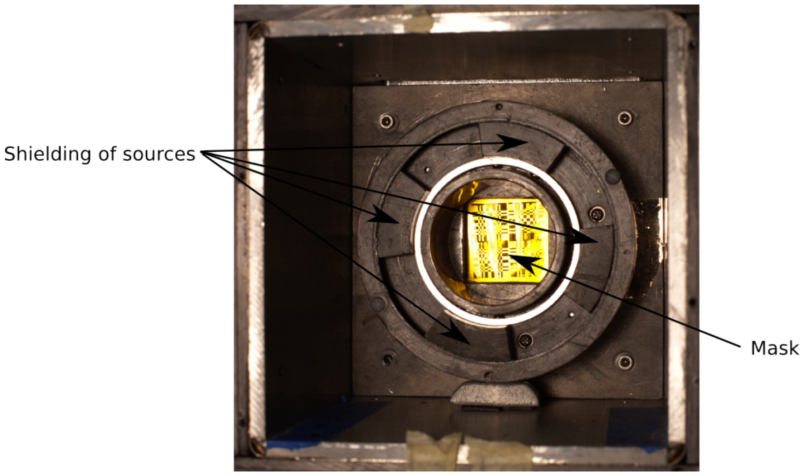

Figure 20: Front view: Mask and sources

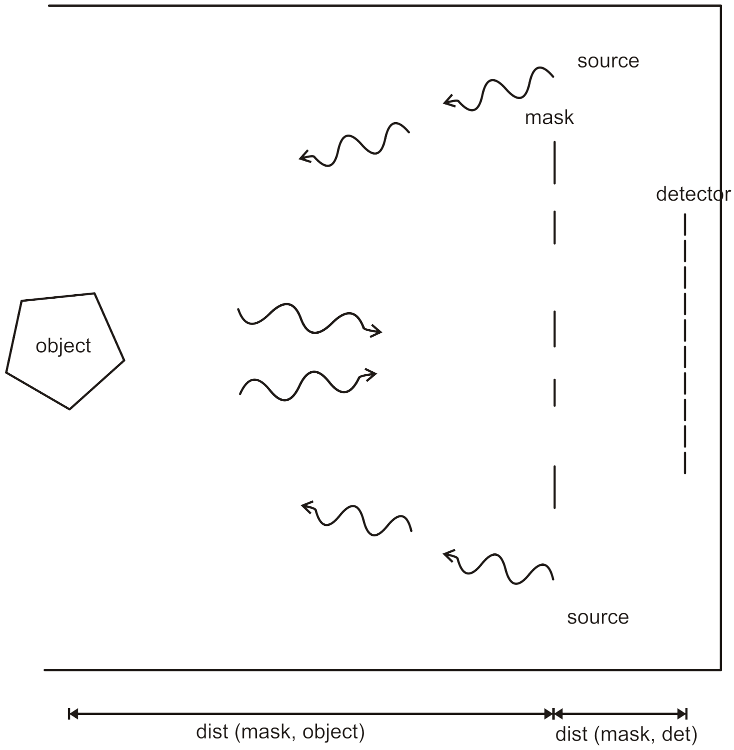

Figure 21: Illustration of the coded mask setup at UCSD

with the two times four sources distributed around the mask and covered by lead. Radiation from those sources will hit an object in front of the camera. Three forms of the interaction of a high-energetic photon with matter are possible: *Absorption*, *transmission* and *backscattering*. The two former ones are trivial to understand. The third one can be understood as a kind of reflection and is the one of interest for the UCSD detector: Photons from the X-ray sources hit the object. Some of them are being backscatter in the direction of the camera. Those are recorded and used to reconstruct an image of the object in front of the camera. A illustration of the structure and the principal setup of hardware for backscatter imaging is shown in Figure 21.

4.2 Physical background of backscattering: Compton

The backscattering of photons is based on the Compton effect which is the is the interaction between a free electron and a photon. It is useful to have a basic understanding of it: Once a photon hits a free electron, the photon's energy E_{ph} is partially transfered into kinetic energy E_k of the electron. The photon changes its direction and continues with a lower energy. This is illustrated in Figure 22. Before

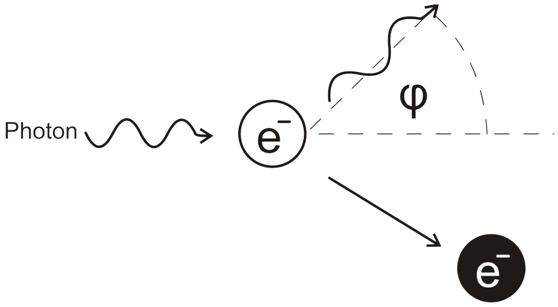

Figure 22: Compton effect: Interaction of a photon with a free electron

the photon and the electron collide, their energies E_{ph} respectively $E_{e,0}$ are

$$E_{ph} = hf, E_{e,0} = mc^2 \tag{14}$$
$$\tag{15}$$

where h is Planck's Constant, f is the frequency of the photon, m is the mass of the electron and c is the speed of light. The impulses p_{ph} and p_e are

$$p_{ph} = \frac{h}{\lambda}, p_e = mv \tag{16}$$
$$\tag{17}$$

where v is the speed of the electron.

The sum of all energies and the sum of all impulses are conserved in a perfect system, resulting in

$$E_{ph_0} = E_{ph,1} + E_{e,cin}, p_{ph,0} = p_{ph,1} + p_{e,1} \tag{18}$$
$$\tag{19}$$

where an index of 0 denotes the particle before the collision and an index of 1 denotes the particle afterwards.

The combination of the above equations shows that the wavelength change of the backscattered photon depends on the backscattering angle φ only.

$$\Delta\lambda = \frac{h}{mc}(1 - \cos(\varphi)) \tag{20}$$

4.3 The detector system

To complete the view on the hardware side of the UCSD setup, some facts about the detector system are collected. They will be used as parameters for the simulation and reconstruction software.

- Detector:

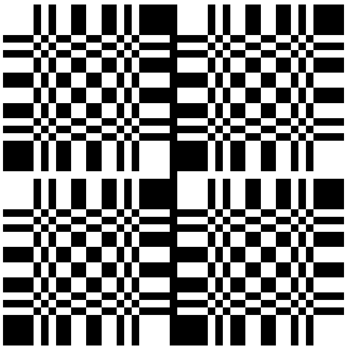

Figure 23: 2 × 2 MURA mask with N = 23.

  - Number of cathode stripes: 32
  - Number of anode stripes: 32
  - Grid size of detector: 0.05 cm
  - Sources available: 4× Americium 241 and 4× Cobalt 57

- Mask:
  - 2 × 2 array of a MURA mask
  - $N = 23$ (see Figure 23)
  - Pitch: 0.0865 cm
  - Mask can be rotated to anti-mask

- Distance between mask and detector: 3.34 cm

- Distance between mask and end of lead box (object position A): 6.5 inch $\sim$ 17cm

For further details on the detector system refer to Rothschild et al. (2006), Suchy et al. (2006), and Faust et al. (2003).

5 Experimental setup - data processing

5.1 Introduction

So far the physical background and the hardware setup were discussed. This chapter analyzes how an efficient reconstruction software can be designed and integrated into a bigger scientific framework. The chosen framework is the "FITS pipeline" developed by the University of Tübingen (Schwarzburg, 2005) and the University of Erlangen-Nuremberg (Wilms et al., 2006). It will be used for the IXO satellite mission as well as its use is considered for eROSITA. One important advantage of the integration into this particular software project is its modular design and the use of the (astronomy-wise) widely used and open file format FITS. Before the FITS format and the FITS pipeline are introduced in section 5.3 and 5.4, a quick overview of the existing software is given in next section. Section 5.5 analyzes the required data flow for an efficient integration of image reconstruction into the FITS pipeline. Section 5.6 shows details about the implementation itself.

5.2 Existing image reconstruction software at UCSD

The motivation for setting up the FITS pipeline came from the need to build the analysis software within a coherent software framework instead of the monolithic software engineering approach used in the existing software. These latter software modules include a simple image reconstruction software in IDL as well as a similar software written in the root framework.

The IDL solution performs real time imaging and tracks a radioactive source put in front of the detector. It was used as a proof of concept for demonstrations for the Department of Homeland Security. It is a rather simple standalone solution.

There is also a analysis software by Dr. Anthony Faust of DRDC, which was written in ROOT[23]. ROOT seems to make small interface changes between minor version updates therefore leading to a strong dependency on the exact same ROOT version. Since this caused significant problems due to conflicts this software was not running during the time this thesis was written and could therefore not be tested.

Both solutions are rather simple monolithic solutions. For more sophisticated image reconstruction the use of a larger framework based on the FITS file standard is required.

5.3 Flexible Image Transportation System (FITS)

FITS is a development endorsed by the NASA and the International Astronomical Union (IAU) and is an important file standard in astrophysics. It was first introduced by Wells et al. (1982). The purpose of the FITS format is the handling of large

[23] ROOT is an Object-Oriented Data Analysis Framework with a syntax similar to C++. It is developed at CERN and especially useful for the processing and mining of large amounts of data. Further information can be found at `http://root.cern.ch/`

quantities of data in multiple dimensions, such as images, spectra or generic tables respectively data cubes of arbitrary values. It is the format of choice for data coming from all professional observatories, such as the European Southern Observatory in South America, the Hubble Space Telescope or the XMM-Newton satellite.

Scientific data needs additional information for the correct interpretation, e.g., coordinate systems for rectification, calibration parameters or meta data describing how the data were gathered. Most image formats, such as jpeg or png, do not offer of combine image data and arbitrary additional information. FITS, on the other hand, fulfills this requirement and makes additional parameters in the header of the FITS file possible. Each FITS file consists of one or more header and data units ("HDUs"). The first HDU is called the "primary HDU" or "primary array", while the additional HDUs are called "extensions".

The header of each unit is formatted as ASCII. This has the advantage of being human readable and editable without additional software. Each parameter is a pair of a keyword (up to 8 characters of uppercase letters, digits, hyphens and underscores) and its value, separated by an equal sign and a single space. Due to historical reasons of block size restrictions on tape devices the length of each keyword/value pair including its comment must count exactly 80 characters and has to be padded to fulfill this requirement. There are few reserved and mandatory keywords (such as the number of dimensions and the size of each dimension). Keywords for other purposes can be introduced by simply using them (and documenting there meaning) which is very useful for new developments.

The header is followed by the data itself. It consists of arbitrary data in multiple dimensions with the sizes and value type defined in the header.

Standard extensions are "image", "table" (filled with ASCII characters) and "bintable". Furthermore various conformed extensions exist for standard tasks. User-defined extensions are possible. Backwards compatibility is assured, hence every FITS file that was valid once is granted to be valid in all further FITS versions. With these properties FITS qualifies as the standard file format for data in astrophysics. It is used in almost all setups and a sophisticated library for accessing FITS files as well as a large collection of tools for FITS file processing exists.

The library is called "cfitsio" and is developed by NASA[24]. It offers efficient routines for reading and changing data and accessing parts of FITS files by providing the "Extended Filename Syntax"[25]. This syntax extends filenames by square brackets containing parameters giving the number of the HDU to read, the specification of certain columns, or a range/region to read values from. This makes writing tools for the processing of FITS files much easier.

As mentioned before a large collection of utility tools exists, written for the creation, analysis and manipulation of FITS data. These tools form the core of the

[24] Information on the cfitsio library can be found at `http://heasarc.gsfc.nasa.gov/fitsio/`.

[25] A complete list of filter parameters is given on the webpage of cfitsio: `http://heasarc.nasa.gov/docs/software/fitsio/filters.html`.

HEASARC[26] system for the reduction and analysis of astrophysical FITS data.[27]

5.4 The fpipe project

The following paragraphs give an overview why the FITS pipeline is a good choice to serve as a framework for the image reconstruction of the HEXIS detector. The data processing software for most astrophysical detector projects in the past tends to be specialized monolithic software. Still many projects use similar algorithms and techniques. Therefore development resources are wasted and programming or design errors are more likely.

A better solution would be a modular design aiming for reusability. Every project should reuse the existing modules and contribute new ones that do not exist yet. This would lead to a collection of well tested and efficiently coded modules and therefore be a huge benefit for astrophysical projects.

Since the University of Tübingen was in need for a data processing software for a laboratory detector project they decided to develop it under these criteria. A framework to perform data processing with a combination of FITS tools as well as a first set of FITS tools for this project was designed and implemented. This project is called the “FITS pipeline” or “fpipe” and bases, according to Wilms et al. (2006) and Schwarzburg (2005), on the following requirements:

- The steps of data processing are separated, each step is performed by one module specialized for this particular task. This is basically the UNIX philosophy of “one job - one tool”: Each software program should do exactly one thing and be good at doing that one. Therefore the bloating of software is avoided and efficiency is promoted.
- The pipeline itself is defined by a XML file describing the data flow through the pipeline and the way the modules should be connected to each other. This is done by the pipeline driver and can be controlled with a GUI. The driver is responsible for triggering the individual modules once input data is available for them. Communication between the driver and the modules or between two modules is done using XPA Messaging, first introduced by Mandel et al. (1995). Additional information about XPA can be found in Schwarzburg (2005).
- Data are read or written from respectively to FITS files or streams. Exceptions are the tools that convert input data into FITS files or that export processed data into other formats. For the data exchange between modules, the FITS format is used exclusively in either regular files or shared memory segments.
- Basic support for existing ftools is implemented, therefore not only ftools developed specifically but all ftools within certain standards can be used within the pipeline.

[26] NASA’s High Energy Astrophysics Science Archive Research Center

[27] see `http://heasarc.gsfc.nasa.gov/ftools/ftools_intro.html`

- The fpipe works near realtime. This is done by two properties: Firstly, every single module needs to use efficient algorithms that can handle the incoming data stream without causing further delays. Secondly, the pipeline driver is designed to continuously feed data from one module to the next one as soon as the predecessor offers it. It is not necessary that all data were already processed.

This design enables us to reuse software. If a different task is to be performed, it can be done by simply changing the pipeline description file, aka the way of combining a certain choice of modules. Many projects will require a common subset, hence those projects can use the existing modules. The FITS pipeline qualifies for its use in a lot of similar projects, for both, laboratory/offline analysis as well as reallife/online usage. The fpipe is used in testing parts of the IXO setup in Tübingen and at FAU Erlangen-Nuremberg and considered to be used in eROSITA data processings as well. For further information on the properties of the FITS pipeline please refer to Schwarzburg (2005).

5.5 Data flow

To use the fpipe for the image reconstruction of data from a near-field coded mask system the necessary data flow has to be analyzed. The following steps have to be performed:

- Read the hits from the detector and transform them to events[28]. In this stage, calibration is performed and data can be dumped to files for processing at a future time.

- Image reconstruction is performed on the events. For FFT this happens after a transformation step of the events into a shadowgram. Back-projection works directly on the eventlist.

- Both ways of image reconstruction return a FITS file with an image of the reconstructed virtual sky. A module is needed to display this image and export it into other file formats.

- To be independent from the detector system a module simulating a coded mask camera is required.

Figure 24 shows how the image reconstruction of data gathered from the detector system is done in the pipeline. From section 2.5.4 it is known that significantly improvements of the image quality can be made by combining mask and anti-mask

[28] The term “event” refers to a photon being registered by the detector. A “hit” refers to the occurrence of energy transmitted by a photon in a detector pixel. One photon can cause several hits in adjacent detector pixels. They have to be combined by a cleaning stage to form an event.

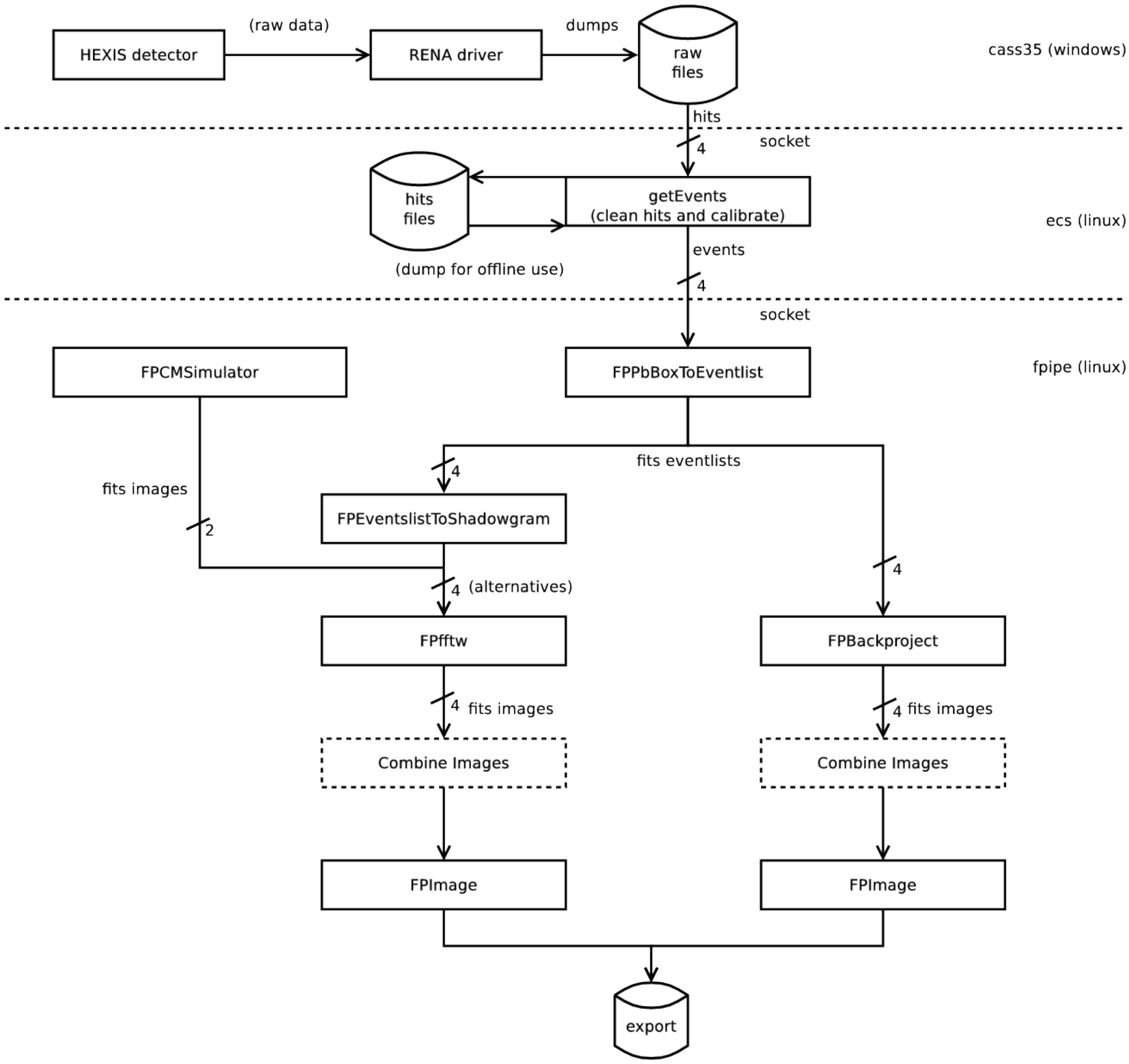

Figure 24: Data flow

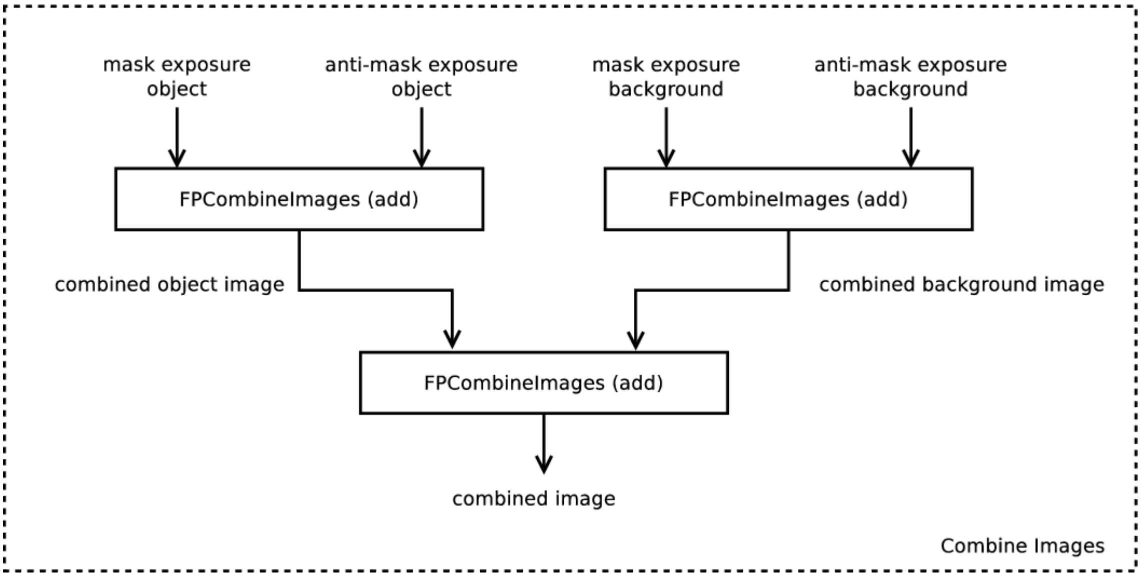

Figure 25: The combination of images in the pipeline

images. Furthermore section 3.5 suggested the subtraction of the combined background image. Hence the data flow described above was extended to process up to four image (represented by `/4` in the data flow diagram) respectively up to two image (represented by `/2`, applies for the simulation which is currently not designed to simulate any noise) which are combined before being displayed or exported. The way of combining the images in the pipeline is shown in Figure 25. Further explanations why this is done in the illustrated way can be found in section 5.6.8.

5.6 Implementation

5.6.1 Overview

The section describes the design of the modules listed in the data flow diagram. The parameters, feature requirements, implementation and status is given for each of the modules.

5.6.2 `getEvents`: transform hits to events

Purpose

The hits read by the detector system and forwarded to this module by a socket need to be cleaned and calibration needs to be performed. The final set of events is streamed into the fpipe through another socket.

Parameters

- `hits:` list of filenames to read the hits from (optional, only in the offline version of the software)
- `host:port` the socket where the hits are received (optional, only in the online version of the software)
- `calibration:` filename of calibration information (mandatory)

Features

Two versions of this module exist: One works directly on live detector data. In this case, data are read from a socket connection to the windows machine controlling the detector system. After transforming the hits to events and calibrating them (this step is called pre-processing) the events are stored in a ringbuffer and streamed to the fpipe as soon as a connection is established. The offline version consists of two parts: Part A receives the hits from the detector through the formerly described socket and dumps the data into (a) file(s). These files can be parsed at a future time by part B. Part B is responsible for reading the hits from file, performing the same pre-processing as the online version and streaming the events to the fpipe as described before.

Details about the implementation

Hits are cleaned by the following criteria: The hits of split events are defined by an identical timestamp and up to three adjacent cathode and anode hits. This means a maximum of nine hits can be combined to one event. If more than three or non-adjacent hits occur at the same time, they are removed from the list as invalid hits. The event position is truncated down to integers representing the pixel number in (x, y) direction. Afterwards a detector calibration matrix is applied. The cleaned events are stored in a buffer for streaming them to the fpipe. The modules listens on a socket and provides the data as soon as a connection was established.

The protocol for the stream needs to separate up to four different sets of events:

1. Background recorded with mask
2. Object recorded with mask
3. Object recorded with anti-mask
4. Background recorded with anti-mask

The assumption is made that putting the object in front of the camera and removing it again does not change the background while a change from mask to anti-mask back to mask could cause little inaccuracies in the mask alignment. Hence this order is a good pick and therefore a mandatory property of this protocol. The sets of events are separated by special events, therefore empty sets can be used if some of the four exposures were not made. For the offline version the implementation of this feature is easy, in case of an omitted filename or an empty file only the special event is sent. For streaming with the online version further configuration is needed to determine which of the sets the Windows machine controlling the detector system sends in which order.

Each event is a quadruple of values: Two integers describing the (x, y) position in terms of the anode- and cathode-index, a float representing the energy of the event and an integer giving a timestamp. Since (x, y) can not be negative for valid events, the special event sets $x = -1, y = -1$ thus indicating the end of the set of events.

Status

This module works as described and credits for help with the implementation go to Tom Gasaway of UCSD. Improvements could be made by interpolating split events into floating point numbers therefore getting a better resolution than the detector pixels would allow, assuming that most events will influence adjacent pixels, too.

5.6.3 `FPPbBoxToEventlist:` Convert data from lead box to eventlist

Purpose

This module is the first stage in the FITS pipeline. It reads data and dumps them in FITS files. This process is called fitsifying.

Parameters

- `host:port` specifies the server delivering the event data. (mandatory)
- `background-mask:` filename output (mandatory)
- `object-mask:` filename output (mandatory)
- `object-antimask:` filename output (mandatory)
- `background-antimask:` filename output (mandatory)

Features and implementation details

This module connects to the server socket. The output files are opened in any case and the events are written into them. If not all four sets of events are provided by the server, the corresponding output eventlists will be empty. This gives subsequent modules in the data flow the chance to detect this and handle it appropriately.

Current status

Module is in draft status. If filtering is needed, e.g., considering only certain time frame or energy band, this should be done after this step. Modules for filtering eventlists are already part of the FITS pipeline project.

5.6.4 `FPEventsToShadowgram`: Accumulate the events to a shadowgram

Purpose

This module reads an eventlist and transfers it to a shadowgram.

Parameters

- `inputfile`: filename of the FITS file containing the eventlist (mandatory)
- `outputfile`: filename of the FITS file to write the final shadowgram to (mandatory)
- `x-size`: size of the shadowgram/number of detector pixels in x (mandatory)
- `y-size`: size of the shadowgram/number of detector pixels in y (mandatory)

Features and implementation details

First, memory of the given size is allocated and all pixel values are initialized to 0. For each event in the list the corresponding pixel's value is incremented. It is obvious that the timestamp can be discarded. Because further processing algorithms, that base on the shadowgram as an greyscale image, have no need for it, this applies for the energy as well.

Current status

Module is in draft status. Improvements considered are the automatic detection of the (x, y) size. This could be either done by adding the detector pixel number to the header of the eventlist files (which requires adding them to the socket protocol as well) or by an additional pass through the eventlist before accumulating the values.

5.6.5 `FPCMSimulator`: Coded mask Monte Carlo simulation

Purpose

To simulate an coded mask camera used in near-field a Monte Carlo simulation was written. This has the advantage that real-life properties such as calibration issues, noise, artifacts and background scattering do not occur. The image reconstruction can be evaluated in a perfect environment and results of the real-life setup and the simulation can be compared. Furthermore the simulator makes an examination of detector failures possible: Broken pixels or stripes can be simulated in software (instead of manipulating the laboratory setup) and their influence on the image reconstruction can be evaluated.

For a Monte Carlo simulation in far-field applications refer to Vadawale et al. (2005).

Parameters

- `source-position`: Determines the three-dimensional Cartesian coordinate of a source (one mandatory, more than one optional)
- `source-intensity`: If more than one source is present, this value is used to determine a ratio between the sources (optional, defaults to 1)
- `det-pixel-num`: Number of pixels in x- and y-direction, assuming a square detector (mandatory)
- `det-pixel-size`: Length of the edge of one detector pixel (mandatory)
- `distmaskdet`: Distance between mask and detector (mandatory)
- `mask`: FITS file specifying the mask design (mandatory)
- `mask-pixel-size`: Length of the edge of one mask pixel (mandatory)
- `photons`: number of photons for the Monte Carlo simulation. (optional, defaults to 1 000 000)
- `shadowgram-mask`: filename to write the shadowgram of the mask simulation into (mandatory)
- `shadowgram-antimask`: filename to write the shadowgram of the anti-mask simulation into (mandatory)

Features

To simulate the detector a Monte Carlo simulation is used. This means that a large number of random photons are created. Those photons are traced and a virtual detector pixel is incremented if the photon reaches it. This way of simulating requires the number of photons to be very large.

The described procedure is performed for both, mask and antimask. After all photons were simulated, the shadowgram of the mask exposure and the shadowgram of the anti-mask exposure is saved in a FITS file.

Details about the implementation

Before the simulation process itself starts, the source intensities are normalized. They are required to add up to 1, therefore the normalized intensity is transformed to a value between 0 and 1 giving the likelihood that a photon originates from this source. These values are used for the photon distribution. The next step is the randomized generation of moving photons according to the calculated distribution. These photons are then traced onto the coded mask camera.

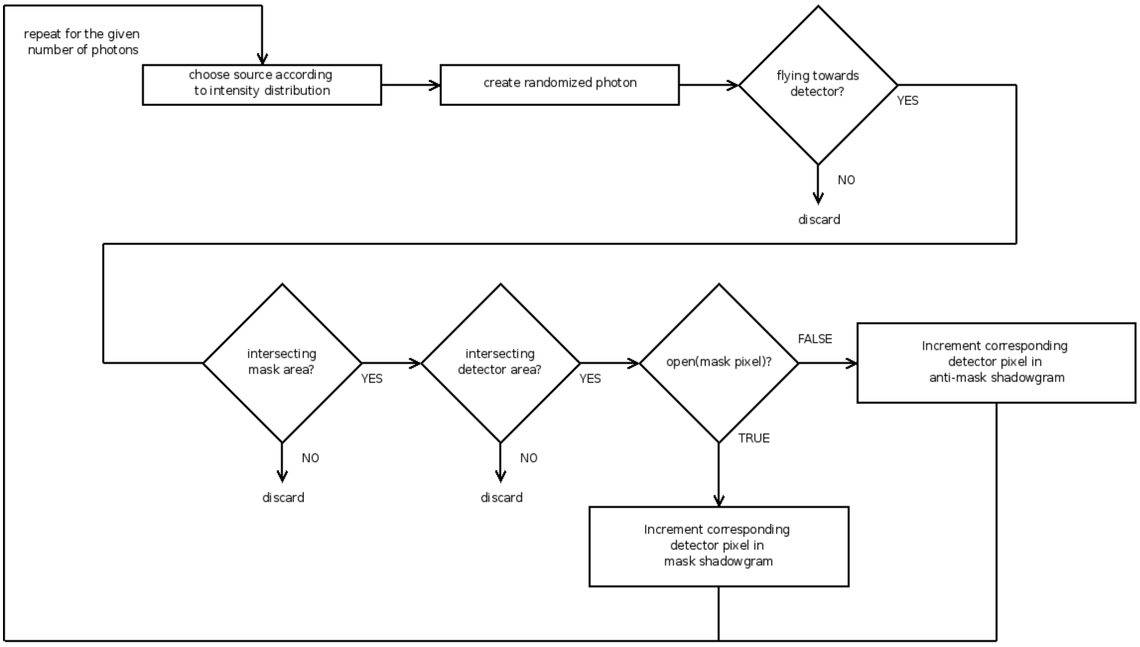
repeat for the given number of photons
choose source according to intensity distribution
create randomized photon
flying towards detector?
YES
NO
discard
intersecting mask area?
YES
NO
discard
intersecting detector area?
YES
NO
discard
open(mask pixel)?
FALSE
TRUE
Increment corresponding detector pixel in anti-mask shadowgram
Increment corresponding detector pixel in mask shadowgram

Figure 26: Data flow of photon processing

Current status

The camera is defined in the following way: The optical axis equals the x-axis, with the mask in the origin of the coordinate system and the detector being at $x = -1$. If the randomized photon is moving perpendicular to or away from the camera ($x \geq 0$) it is discarded. Otherwise the intersection between the photon path and the mask plane is calculated using simple geometry. If the intersection point is out of the mask bounds, the photon is discarded. Otherwise the intersection between the photon and the detector plane is calculated. Again the photon is discarded if it does not hit the detector area. At this point only photons remain that passed through the area of the mask and hit a detector pixel. To calculate both, mask and anti-mask shadowgrams, at once a knack is used: If the photon passed the mask at a open pixel, the hit detector pixel of the mask shadowgram is incremented. On the other hand the corresponding anti-mask shadowgram pixel is incremented if the photon passed the mask at a close pixel. Since all open anti-mask pixels are close mask pixels and vice versa this is a good strategy to simulate both exposure at the same time. For a better understanding the processing of each photon has been illustrated in Figure 26.

Current status

The basic functionality was implemented as described above. The implementation is considered to be at a alpha state since many details could be solved in a more sophisticated way, e.g. an arbitrary number of sources should be allowed. An important improvement and additional feature should make photons originating

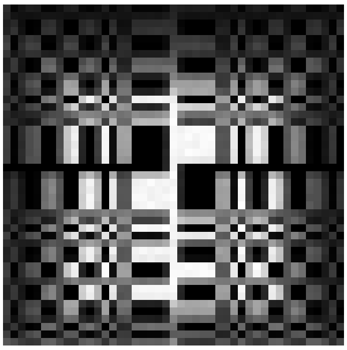

Figure 27: Shadowgram, exposed with the HEXIS mask

from area sources possible. Figure 27 shows the shadowgram of a simulated source at $(10, 3, 0)$.

5.6.6 `FPfftw`: Reconstruction using FFT

Purpose

This module takes the shadowgram and recalculates the sources and background of the area in front of the detector. To perform that step it needs the exact layout of the mask as well as the detector parameters together with the distance between the two. The image reconstruction uses Fast Fourier Transform.

Parameters

- `shadowgram`: FITS file containing the image data of the shadowgram (mandatory)
- `det-pixel-num`: Number of pixels in x- and y-direction, assuming a square detector (mandatory)
- `det-pixel-size`: Length of the edge of one detector pixel (mandatory)
- `mask`: FITS file specifying the mask design (mandatory)
- `mask-pixel-size`: Length of the edge of one mask pixel (mandatory)
- `distmaskdet`: Distance between mask and detector (mandatory)
- `distmaskobject`: Distance between mask and assumed object plane (mandatory)
- `outputfile`: FITS file to write the image of the virtual sky to (mandatory)
- `sky-pixel-num`: Number of pixels of the virtual sky in x- and y-direction, assuming a square virtual sky (mandatory)

- `sky-pixel-size`: Length of the edge of one virtual sky pixel (mandatory)
- `sampling-factor`: Number of subpixels forming one regular pixel (mandatory)

Features

This module performs a Fast Fourier Transform, following the algorithm that was described in section 3.3. Different mask and detector pixels sizes and number have to be treated. This is done using a sampling factor which increases the resolution and therefore the runtime complexity of the algorithm.

Details about the implementation

First, shadowgram and mask are read from the given FITS files. The FFT is performed by the fftw3 library[29]. This approach requires both images to have the same pixel size and number. For most setups, the size of projected mask pixels (which are magnified in near-field setups) and detector pixels differs, hence subsampling is used. The following steps are performed (assuming that the mask area is bigger or equal the detector area):

- Memory for the subsampled mask image and the subsampled shadowgram is allocated with a dimension of `mask-pixel-num * sampling-factor` values for each direction.
- The mask image values are copied into the subsampled mask image, each regular pixel is represented by `sampling-factor` × `sampling-factor` subpixels.
- The corrected sampling-factor for detector-pixels is calculated by

$$\frac{\texttt{sampling-factor}}{\texttt{mask-pixel-size}} \cdot \texttt{det-pixel-size} \tag{21}$$

- The shadowgram values are copied into the subsampled shadowgram using the corrected sampling-factor. Since the shadowgram will be smaller than the available area, it is centered. Undefined values surrounding the subsampled shadowgram are filled with zero (this is done by initializing the array with zeros at the time of creation).

Afterwards, a FFT plan is calculated for the shadowgram and the projected mask pattern and FFT is performed on both planes. In the next step, the corresponding values of the two complex results are multiplied. The multiplied values are then transformed by the Inverse Fast Fourier Transform. The result is interpreted as the virtual sky and written to the FITS output file.

[29] A comprehensive documentation on the fftw3 library can be found on the fftw3 webpage: `http://www.fftw.org/fftw3_doc/`.

Current status

This module is in alpha state. It is currently limited to perform Fast Fourier Transform on equally sized detector and mask image. Furthermore, the distance to the object is not automatically determined as suggested in section 3.3.

5.6.7 `FPBackproject`: Reconstruction using back-projection

Purpose

This module takes the eventlist of the detector and recalculates the sources and background of the area in front of the detector using the back-projection algorithm described in section 3.4. The back-projection algorithm takes the energy of the photons into account. Therefore please note that this module differs from the `FPfftw` interface by using the eventlist instead of the shadowgram as its input.

Parameters

- `eventlist`: FITS file containing the list of the photon events (mandatory)
- `det-pixel-num`: Number of pixels in x- and y-direction, assuming a square detector (mandatory)
- `det-pixel-size`: Length of the edge of one detector pixel (mandatory)
- `mask`: FITS file specifying the mask design (mandatory)
- `mask-pixel-size`: Length of the edge of one mask pixel (mandatory)
- `distmaskdet`: Distance between mask and detector (mandatory)
- `distmaskobject`: Distance between mask and assumed object plane (mandatory)
- `outputfile`: FITS file to write the image of the virtual sky to (mandatory)
- `sky-pixel-num`: Number of pixels of the virtual sky in x- and y-direction, assuming a square virtual sky (mandatory)
- `sky-pixel-size`: Length of the edge of one virtual sky pixel (mandatory)

Features and Implementation

The module needs to iterate through all detector pixels. If a pixel had at least one photon event, basic geometry is used to calculate four rays for each open mask pixel, originating from the four corners of the pixel. To determine the origin of each ray, the (x, y) coordinate has to be transformed into a Cartesian coordinate by taking the size of the detector pixels into account. Rays are defined by the corner of the pixel and

1. the bottom left corner of the mask pixel for rays originating from the upper right corner of the detector pixel.
2. the bottom right corner of the mask pixel for rays originating from the upper left corner of the detector pixel.
3. the upper right corner of the mask pixel for rays originating from the bottom left corner of the detector pixel.
4. the upper left corner of the mask pixel for rays originating from the bottom right corner of the detector pixel.

Each ray is then intersected with the plane representing the virtual sky. The Cartesian intersection point has to be transformed back into a pixel coordinate of the virtual sky array. If the four rays of one detector pixel do not hit the same virtual sky pixel, a double nested loop is used to increment all pixels in between the four corners on the virtual sky.

In all steps the photon energy needs to be distributed according to the number of possibilities: The energy of all photons in one detector pixel is added up and then divided by the number of open mask pixels multiplied with the number of hit virtual sky pixels.

Current status

This module is in draft state. Improvements can be made by the automatic determination of the distance between mask and object. Furthermore, the accuracy can be improved by distributing the energy by the area ratio of the virtual sky pixels that have been hit.

5.6.8 `FPCombineImages`: Combine mask/anti-mask and background images

Purpose

This module opens two images of the same size and combines the values of corresponding pixels according to the given mathematical operation.

Parameters

- `infile1`: filename of FITS file containing source image one (mandatory)
- `infile2`: filename of FITS file containing source image two (mandatory)
- `outfile`: filename of FITS file to write the resulting image to (mandatory)
- `operator`: string specifying the mathematical operation “add”, “sub”, “mul” or “div” (mandatory)

Features and Implementation

Before the actual implementation is discussed, the interface and idea should be clarified. First approaches of implementing this modules combined all four images in one run with a fixed order of operations. This has the advantage of producing less overhead for reading and writing files but is less flexible. Since the overhead should not be to high for the image sizes used in this setup, a decision for a more flexible solution was made. Therefore this module offers the combination of two images with an arbitrary operation which makes the pipeline driver responsible to apply this module in the correct way.

The combination works as follows: The two FITS input files have to be opened and read. The first step is to ensure that both image planes have the same size. Otherwise this module terminates with an error message and an negative exit code. An image plane for the resulting image is allocated. Afterwards the values of the two input images are traversed and each two corresponding pixels are combined according to the operator. As a final step the resulting image is written to the output file.

The data flow for coded mask image reconstruction in near-field requires addition and subtraction only – still, multiplication and division are offered for further use in other projects. While addition and multiplication are commutative, subtraction and division require to take the order into account. The first input file is taken as the subtrahend respectively the dividend, the second input file represents the minuend respectively divisor. Note that pixel values of the divisor could equal zero. In that case the result of the division is not defined. The corresponding output pixel is set to NaN.

Current status

Works as described above.

5.6.9 `FPImage`: 2D imager

Purpose

This ftool reads a two-dimensional image from a FITS file and displays it. The GUI gives the user the ability to interact with the view or export the image to another file format. There are some reasons to write a new image viewer instead of using existing work: Only a few of the widely spread image viewers are capable of reading FITS files. Those that are are usually bloated and/or can not be easily integrate into the pipeline. Furthermore, none of the available FITS viewers would have ensured the tight interoperability with fpipe required.[30]

[30] The only option of a third-party reader considered for this project was the astronomical data visualization application “SAOImage DS9” (`http://hea-www.harvard.edu/RD/ds9/`). It is capable of reading FITS files and supports XPA, hence it could have been integrated into fpipe. It is a rather big and powerful application. We decided for a small and lightweight imager and therefore discarded DS9.

Parameter

- `infile`: Filename of the FITS image to be shown. The number of the extension can be specified in square brackets. (mandatory)
- `log`: If set to 1 the image values are interpreted on a logarithmic scale. (optional, defaults to 0)
- `zoom`: Defines the scaling factor used to display the image. This parameter has no influence on the size of the exported image file. A positive factor states a magnification (e.g., 2 will double the edge length of each pixel) while a negative factor will shrink the displayed image. (optional, default to 3)
- `color1`: Integer giving the BGR-color into which the minimum value should be transformed (optional, defaults to black)
- `color2`: Integer giving the BGR-color into which the maximum value should be transformed (optional, defaults to white)

Feature requirements

- FPImage uses the cfitsio functions to efficiently read the necessary header data and the image from the FITS file.
- The intensity of the image pixels has to be transformed into color values. Different color scales should be taken into account, e.g., at least linear and logarithmic scales should be supported. The transformation should allow different color assignments and a contrast regulation.
- The color values are displayed according to the given parameters.
- A area displays the processed image. Manipulation of the contrast can be done using the mouse.
- The displayed image can be exported into various file formats, e.g., png, jpeg and pdf. The exported image pixels are colored according to the current settings. The zoom factor has no influence on the exported image.

Details about the implementation

The Implementation of the GUI is based on Gtk[31], therefore doing things analogue to other graphical ftools such as `FPPlot2`. Three important choices had to be made: The way how to read the data from the FITS file (which is obviously done best by the cfitsio library), the steps and data structures to transform the intensity values into colors and the way to display the image itself.

[31] "GTK+ is a highly usable, feature rich toolkit for creating graphical user interfaces which boasts cross platform compatibility and an easy to use API." (`http://www.gtk.org/`

The intensity values are first cropped to the region of interest determined by `contrast-width` and `contrast-offset`. The initial values for these settings are:

```
contrast-width = max(pixels) - min(pixels);
contrast-offset = min(pixels)}
```

Afterwards the values are transformed to a double between 0 and 1 using a linear or logarithmic scale. An additional scale can be applied to the transformed values in future versions. Since two color values can be set for the minimum and maximum value to be displayed, the resulting double array is linear transformed into color values and saved in a GdkPixbuf.

This GdkPixbuf then needs to be display according to certain parameters. In the first version of the implementation this was done manually: An own widget, inheriting from GdkDrawingArea, was written. It first magnified the value array according to the zoom factor and painted it into the DrawingArea using `cairo`. Export was done using a cairo surface inherited as PDF, PS, or SVG surface.

Once the implementation started to take interpolation methods for arbitrary zooming into account it was decided that this implements a lot functionality already done by existing GtkWidgets. After further research it was decided that GtkImageView[32] should be used. It is available in most modern distributions and its code appears to be in a good stable state. GtkImageView takes a GdkPixbuf and displays it according to a zoom parameter and callbacks that can be used to extend the functionality of GtkImageView. Callbacks are currently used to grab the mouse movement and status to use it for contrast detection.

Current status

The basic functionality was implemented with most of described features. Two screenshots showing two images with different color scales are shown in Figure 28 and 29. Many details need to be improved and some advanced features added: Callbacks should read the pointer position and display the corresponding value of the pixel. Furthermore a 3×3 pixel block should be added, showing the surroundings of the current pointer position.

[32] For further information please refer to `http://trac.bjourne.webfactional.com/chrome/common/gtkimageview-docs/` or to the documentation of GtkImageView provided by your distribution. Note that there is another GtkWidget named GtkImageViewer. Both were compared during the development of the imager and the decision for GtkImageView was based on the code basis, interface, features and popularity. Hence GtkImageViewer was *not* used.

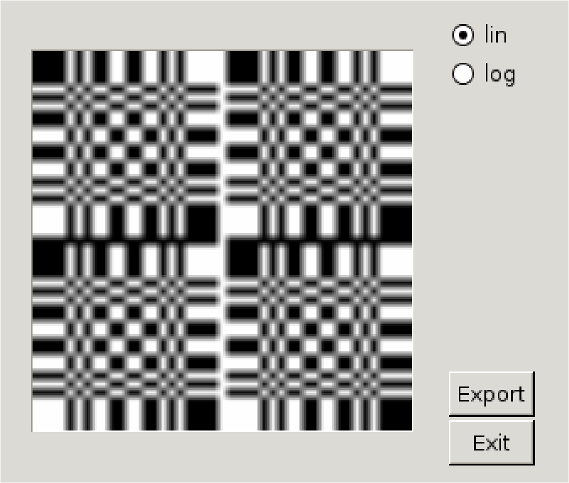

Figure 28: Mask displayed in black and white

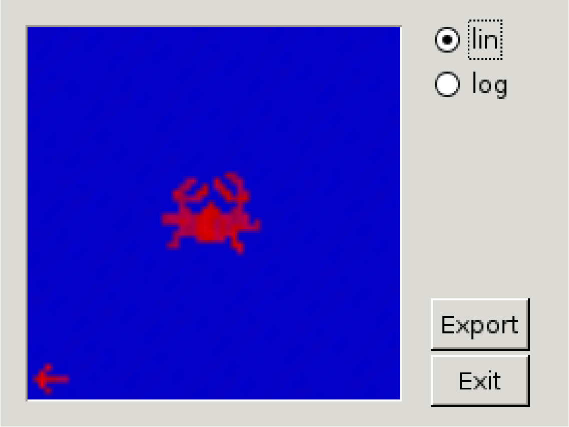

Figure 29: Testimage, values transformed into a blue and red color scheme

6 Summary and outlook

This document describes how the author analyzed the integration of image reconstruction software for coded mask cameras in near-field use into a scientific software framework. A major part of this work was understanding the physics behind coded mask cameras.

Only two algorithms for image reconstruction were discussed: Fast Fourier Transform and the back-projection of photons. They are the most commonly used algorithms so far. Both have been analyzed in far-field setups, followed by the theoretical discussion of necessary adaptions for using the algorithms in near-field applications.

A third possibility of an algorithm performing in near- and far-field, not discussed here for time reasons, is the "Pixon algorithm" introduced by Piña and Puetter (1993). It is a sophisticated approach for distinguishing between hits caused by photons from the source and background/noise, hence, cleaning the image. The algorithm can be considered to be a image restoration method and works, with some pre-processing, for coded mask image reconstruction as well. Also, Pixon looks promising on retrieving three-dimensional properties of the object exposed, e.g. the distance between object and camera. Future work improving the efforts made in this thesis should have a closer look into this matter. A good start is Puetter (2005) as well as the Pixon webpage `http://www.pixon.com`.

For further improvements of the two conventional algorithms (FFT and back-projection), an additional idea was introduced, following other imaging problems: Two exposures are made, one including the object, the other one taking an image of the noise caused by the detector's surroundings. This is similar to taking and subtracting a darkframe with conventional (optical) cameras. This "background" can be subtracted from the object exposure and is expected to eliminate noise and artifacts. As soon as the whole image reconstruction process works as intended, the background subtraction for coded mask cameras should be evaluated.

Some software progress has been made: The integration into the fits pipeline was designed and a list of modules was created. Each module description brings a detailed list of the basic parameters, the basic feature requirements and implementation details. The status of each module is given and improvements are suggested. Still, many modules remain in draft status or in a unfinished alpha stadium due to time reasons. Hence, much implementation work has to be done to finish the image reconstruction pipeline. Since it was not possible to perform a real-time evaluation of the adapted algorithms this thesis should be seen as the basis for further work done in this area.

References

Accorsi, R. (2001). *Design of a near-field coded aperture cameras for high-resolution medical and industrial gamma-ray imaging.* PhD thesis, Massachusetts Institute of Technology. Dept. of Nuclear Engineering.

Anderson, D. N. et al. (2006). Detection and location of gamma-ray sources with a modulating coded mask. *Technometrics*, 48(2):252–261.

Aschenbach, B. (1985). X-ray telescopes. *Rep. Prog. Phys, vol. 48*, pages 579–629.

Braga, J. et al. (2005). The MIRAX X-ray Transient Mission: Recent Developments. In *The X-ray Universe 2005*, page 245 et. sqq. ESA Publications Division.

Chalmers, A. (2001). Transmission Enhanced Backscatter X-ray Images. *Hard X-Ray and Gamma-Ray Detector Physics III, Proceedings of SPIE*, 4507:125–131.

Chalmers, A. (2005). Three applications of backscatter x-ray imaging technology to Homeland Defense. *Sensors, and C3I Technologies for Homeland Security and Homeland Defense IV*, 5778:989–993.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms, Second Edition.* The MIT Press, Cambridge.

Dugan, E. T., Jacobs, A. M., Su, Z., Houssay, L., Ekdahl, D., and Brygoo, S. (2002). Development and field testing of a mobile backscatter x-ray lateral migration radiography land mine detection system. *Detection and Remediation Technologies for Mines and Minelike Targets VII, Proceedings of SPIE*, 4742:120–131.

Faust, A. A., Rothschild, R. E., and Heindl, W. A. (2003). Development of a Coded Aperture Backscatter Imager using the UC San Diego HEXIS detector. In *Detection and Remediation Technologies for Mines and Minelike Targets VIII, Proceedings of SPIE*, volume 5089, pages 95–106.

Gottesman, S. R. and Fenimore, E. E. (1989). New family of binary arrays for coded aperture imaging. *Appl. Opt.*, 28(20):4344–4352.

Groeneveld, H. A. (1999). *Design, Simulation und Optimierung kodierter Aperturen.* PhD thesis, Eberhard-Karls-Universität, Tübingen, Germany.

Hammersley, A. et al. (1991). Reconstruction of images from a coded-aperture box camera. *Nuclear Instruments and Methods in Physics Research A311*, pages 585–594.

Karttunen, H., Kröger, P., Oja, H., Poutanen, M., and Donner, K. J., editors (1987). *Fundamental Astronomy.* Springer, Berlin, Heidelberg.

Keshavmurthy, S. P., Dugan, E. T., Wehlburg, J. C., and Jacobs, A. M. (1996). Analytical studies of a backscatter x-ray imaging landmine detection system. *Proceedings of SPIE*, 2765:512–523.

Kulinder Pal Singh (2005). Techniques in X-ray Astronomy Part 1 and 2. In *Resonance Vol. 10 No. 6 and 7*, pages 15–23, 8–20. Springer India.

Mandel, E., Swick, R., and Tody, D. (1995). The X Public Access Mechanism. *Proceedings of the 9th X Technical Conference.*

Peli, E. (1990). Contrast in complex images. *Journal of the Optical Sociecty of America*, 7(10):2032–2040.

Piña, R. and Puetter, R. (1993). Bayesian image reconstruction: The pixon and optimal image modeling. *Publications of the Astronomical Society of the Pacific*, 105:630–637.

Puetter, R. (2005). Investigation into the utility of Pixon image reconstruction technique for the DRDC Coded Aperture X-ray Backscatter Imager. Contract report.

Rothschild, R. E., Tomsick, J. A., Matteson, J. L., Pelling, M. R., and Suchy, S. (2006). CZT Detector and HXI Development at CASS/UCSD. In F. d'Amico, J. Braga, R. R., editor, *The Transient Milky Way*, pages 107–111. AIP Conf. Proc. 840.

Schwarzburg, S. (2005). Eine Software zur Echtzeitanalyse von experimentellen Daten im Flexible Image Transport System (FITS). Master's thesis, Eberhard Karls Universität, Tübingen, Germany.

Skinner, G. (1995). Coding (and decoding) coded mask telescopes. In Bassani, L. and Cocco, G. D., editors, *Imaging in high energy astronomy.* Kluwer Academic Publishers.

Skinner, G. and Rideout, R. M. (1994). A compendium of coded mask designs. In Bassani, L. and Cocco, G. D., editors, *Imaging in high energy astronomy.* Kluwer Academic Publishers.

Skinner, G. K., Ponman, T. J., Hammersley, A. P., and Eyles, C. J. (1987). Techniques for the analysis of data from coded-mask X-ray telescopes. *Astrophysics and Space Science*, 136:337–349.

Suchy, S., Pelling, M. R., Tomsick, J. A., Matteson, J. L., and Rothschild, R. E. (2006). Laboratory testing of the HEXIS Hard X-Ray Imager Balloon Telescope. In *Space Telescopes and Instrumentation II: Ultraviolet to Gamma Ray, Proceedings of SPIE*, volume 6266, page 626628 et sqq.

Tawara, Y. et al. (2002). A hard x-ray telescope with multilayered supermirrors for balloon observations of cosmic x-ray sources. *Advances in Space Research*, 30(5):1313–1319.

Tsvetkov, M., Tsvetkova, K., Borisova, A., Kalaglarsky, D., Bogdanovski, R., Heber, U., Bues, I., Drechsel, H., and Knigge, R. (2005). Bamberg southern photographic patrol survey: Incorporation in the WFPDB. *Proc. IV Serbian-Bulgarian Astronomical Conference*, pages 303–308.

Vadawale, S. V., Hong, J., Grindlay, J. E., and Skinner, G. (2005). Monte-Carlo simulations of the expected imaging performance of the EXIST high-energy telescope. *Optics for EUV, X-Ray, and Gamma-Ray Astronomy II, Proceedings of SPIE*, 5900:590014 et. sqq.

Wells, D. C., Greisen, E. W., and Harten, R. H. (1982). FITS: A Flexible Image Transport System. *Astronomy and Astrophysics Supplement Series 44*, pages 363–370.

Wilms, J. (2008). Galactic x-ray astronomy. Class at University of Erlangen-Nuremberg: `http://pulsar.sternwarte.uni-erlangen.de/wilms/teach/xray1/`.

Wilms, J., Schwarzburg, S., Remillard, R., et al. (2006). MIRAX Software Aspects. In F. d'Amico, J. Braga, R. R., editor, *The Transient Milky Way, AIP Conf. Proc. 840*, pages 122–126.