UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

NIELS BOHR INSTITUTE

MSc in Physics

Applied Super Resolution for X-Ray Imaging

Virtual Potatoes And How To X-Ray Them

Simon Nyrup

Supervised by Prof. Brian Vinter and Assistant Prof. Kenneth Skovhede

Handed in: September 30, 2020

NIELS BOHR INSTITUTE

UNIVERSITY OF COPENHAGEN

Master's Thesis in Physics
submitted by

SIMON NYRUP

2020

Simon Nyrup: Applied Super Resolution for X-Ray Imaging - Virtual Potatoes and How to X-Ray Them

This Master's Thesis has been carried out at

THE NIELS BOHR INSTITUTE, COPENHAGEN

under the supervision of

PROF. BRIAN VINTER
(Niels Bohr Institute, Copenhagen)

and

ASSISTANT PROF. KENNETH SKOVHEDE
(Niels Bohr Institute, Copenhagen)

ACKNOWLEDGMENTS

In the process of learning, writing and implementing the various concepts relevant for the application of super resolution (SR), several people have guided me.

First of all, I would like to thank Brian Vinter for providing the idea of applied SR - his creative and curious mind always gave me new directions to seek. I would like to thank Kenneth Skovhede for stepping in in the last minute, giving me the guidance needed for a final product. A big thanks should be given to Carl Johnsen, working as a secondary supervisor and carrying me through this whole project. His help, motivation and clever inputs were essential for the work in this thesis. Furthermore, I would like to thank Aleksandar Topic for the fruitful discussions and wider perspective of the underlying theories relevant for this subject.

The above is just a small list of the people I met throughout my time in the eScience group - a group I as a whole would like to thank for their warm and welcoming nature. Their introduction to the combination of cold beers and science has been truly joyful. At last, I would like to thank my friends and family for motivation, grammar corrections and appreciated pats on the back - every cliché about friendship must definitely be true.

ABSTRACT

In the field of food inspection, the implementation of X-ray scanners allows for a non-destructive analysis of the interior of objects. The scanner is a part of a larger pipeline that performs automatic classification by sending objects through the system on a conveyor-belt while doing real-time sorting. Such systems require a low latency with high precision, that allows for a high throughput of objects. The precision of such systems is dependent on the contrast and resolution of the X-ray images, but the various physical processes in the image formation degrade the final representation of the object and remove important features, that are essential to the classification task.

This project provides an investigation of applying multi-frame super resolution (MFSR) as a preprocessing tool for such a classification pipeline. MFSR is the concept of using latent information in a sequence of low resolution (LR) frames to produce a single high resolution (HR) image. The iterative re-weighted super resolution (IRWSR) algorithm, developed by [1], was implemented and applied to sequences of images acquired from an X-ray system. The resulting output shows no clear signs of feature enhancement. Furthermore, the runtime seems to be far from what is required for a high throughput system e. g. a ×2 magnification of an image with size 200×200 is processed in the order of 1000 s.

In the examination of the different limitations of X-ray setups, the simulation tool Xsim was created, capable of generating X-ray images of user-defined 3D objects. The versatile Xsim allows the user to simulate images for different X-ray source spectra, object compositions and geometrical distances. The generated images are perceptually comparable to real X-ray images, but lack effects such as scattering or beam hardening. The IRWSR algorithm was applied to a sequence of test images generated with Xsim. The resulting images contain no perceivable new information compared to the original LR frames.

This project gives an insight into the difficulties of applying SR in real settings. Through the analysis and discussion of the various theories needed, a direction for further use is given. A further investigation should especially address the point-spread-function (PSF) of the X-ray imaging system in combination with a conveyor-belt, and the output should be validated with respect to the larger classification scheme.

CONTENTS

INTRODUCTION

SUPER RESOLUTION

X-RAY SIMULATOR

COMBINING THE TWO

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF CODE LISTINGS

ACRONYMS

ADC	analog-to-digital converter
LSI	linear shift-invariant
PSF	point-spread-function
SR	super resolution
HR	high resolution
LR	low resolution
MFSR	multi-frame super resolution
SISR	single image super resolution
IRWSR	iterative re-weighted super resolution
CCD	charge-coupled device
CMOS	complementary metal-oxide-semiconductor
MAP	maximum a posteriori
ML	maximum likelihood
MAD	median absolute deviation
FOV	field of view
MSE	mean squared error
PSNR	point signal-to-noise ratio
SSIM	structural similarity index
CT	computed tomography
GPU	graphical processing unit
CNN	convolutional neural network
CG	conjugate gradient
SNR	signal-to-noise ratio
DFT	discrete Fourier transform
CDF	cumulative distribution function

Part I

INTRODUCTION

1 INTRODUCTION

In the process of imaging a real world scene to a digitalized format, a loss of information is inevitable. Degradation happens by quantization of a continuous scene, diffraction and the relative motion between the scene and the camera. This results in a loss of features such as edges, shapes and placement, important information in any image classification task. Degradation of this type exists not only in the world of optical light, but also in the field of X-ray imaging, which has important applications in the medical industry, security-surveillance and food inspection. Here especially automatic classification is a growing field of interest and development.

A common X-ray setup in food inspection starts with an X-ray source emitting X-rays through the desired object moving on a conveyor belt and then a detector measuring the attenuation of the rays. An X-ray setup is a tool for examining the interior of an object and is part of a bigger pipeline. This pipeline is based on hardware like cameras, mechanical hinges and conveyor belts, with the purpose of doing a real-time sorting of the objects based on physical quantities such as size, weight and appearance. From an industry perspective, the speed of the overall pipeline is of great importance, as it limits the amount of objects to be analyzed in a given time interval. This leave a high demand of efficiency in the individual hardware and software parts. Considering the connection between X-ray imaging and the relative speed of the objects i. e., the speed of the conveyor belt, there is a trade off between the image contrast and the amount of motion blur. A higher contrast demands a longer exposure time, which will be more affected by motion blur. A lower contrast will give less blur, but show less detail of the given object. Both at the cost of image features, which are essential for the classification.

A promising approach for overcoming this compromise is that of computational super resolution (SR), where the goal is to "recover" the unknown high resolution image from the low resolution image given. A problem of this type is in an inverse problem sense an ill-posed problem, as no unique higher resolution solutions exist. In the literature there are two main approaches; single image super resolution (SISR) and multi-frame super resolution (MFSR). In MFSR the information is gathered over a sequence of low resolution frames to create a single high resolution image - a process that might be well suited for the image sequence outputted from the camera in the pipeline.

The progress of SR methods in the last decade has followed the evolution of computer power and computer vision. It has further divided it into subcategories such as Dictionary Learning[2], Bayesian modeling [1] and Deep learning[3]. The quality of a method is characterized by computational time, complexity, robustness to different kinds of degradation and the final high resolution image. A common way to quantify the quality of a SR output is the artificial degradation of a high resolution image to a low resolution image mimicking the natural degradation of the "high resolution" real world. The low resolution image is then upscaled with a given algorithm and compared to the original image using the point signal-to-noise ratio (PSNR). In the SR community the PSNR is a general optimization benchmark, that is purely based on simulated data and only works as an approximation to real images. This gives a performance gap between simulation and real data, and it has been shown that there is a small correlation between behavior of SR algorithms on simulated and real data [4].

Only recently, the application of SR in real scenarios have started to become feasible. Some SR methods have been implemented in image editing software as a post-processing tool, to enhance the visual perception of natural optical images[5][6]. In recent research combining SR with medical imaging has been applied to images of the human eye, enabling mobile low-cost cameras to obtain comparable quality to that of stationary high-cost cameras [7]. SR research in Magnetic Resonance Imaging (MRI) makes it possible to shorten the scan time while keeping the same image quality [8].

In automatic classification, using online X-ray imaging systems, SR methods could function as powerful preprocessing tools. Here the goal of SR would not be to upscale the image, but to enhance and highlight important features inherent in the target objects. The results of an SR algorithm are depending on a proper analysis of the imaging system, as the underlying structures, such as the point-spread-function (PSF), can be applied directly to the algorithm. *In this work a developed SR scheme should account for the linear motion of the online system and be robust to small random rotations, photometric variance and system noise as it will corrupt the individual frames.*

In order to obtain a better intuition and understanding of the imaging system, an X-ray simulator has been developed in this work. Such a simulator needs to account for all the physical processes included in the imaging system and create images comparable to real ones. The simulation tool can furthermore be used to examine the limitations of a given X-ray imaging system, as it gives full control over source energy, object composition, conveyor belt speed, exposure time and noise. With a basis in this system one could quickly examine the possibilities of classifying different anomalies in objects. For example, it would be easy to interchange a small aluminum fragment in soft tissue with a small iron fragment. However, such a system is only an approximation to the physics of the world, the consequence being that it will never replace the importance of examining real world scenarios. As such it can be viewed as a playground, giving indications of what is feasible for real setups. It can be used as a baseline for SR, as the behavior of the individual frames can be easily isolated. This can help uncover if SR is even feasible and assist in finding the optimal configurations.

The primary purpose of this work is to examine and simulate the structure of X-ray imaging systems in food inspection. We seek to utilize the inherent structures in the system by applying MFSR algorithms on the simulated and real images and quantifying the performance of the MFSR algorithms by examination of the image features.

1.1 AXIS

The work in this thesis is a contribution to the AXIS project, which aims to develop an automatic classification system for X-ray scanners in food inspection. AXIS is a collaboration between 4 different facilities; NewTec in charge of the machinery and integration, Qtechnology developing cameras, MagnaTek producing the X-ray source and the eScience department at the Niels Bohr Institute providing the software.

The demands for the softwares developed for this purpose is that it is versatile, efficient and precise, because it is intended to be integrated as a real-time analysis on an online sorting system. In the process of achieving this goal, the eScience group has developed and examined various image filters, contrast enchantments and classification schemes. The work of this thesis lies in the extension of the previous and draws parallels to the concept of an automatic classification scheme. For this, two novel ideas are researched; the implementing of SR as a preprocessing scheme for the classification pipeline, and the concept of virtual X-ray machine able to test various X-ray setups and provide data for an image processing algorithm.

1.2 DOCUMENTATION

The python scripts written as a part of the work in this thesis is divided into two main sections:

- The implementation of the presented iterative re-weighted super resolution (IRWSR) algorithm by [1] is found at https://github.com/SimonNy/IRWSR.
- The implementation of the X-ray simulator **Xsim** is found at https://github.com/SimonNy/Xsim.

Within each link, the corresponding documentation is provided with examples for generating the results showed in this work. A list of hardware and system specifics for the runtimes provided in this thesis is given in § A.1.

Part II

SUPER RESOLUTION

2 SIGNAL AND IMAGE PROCESSING

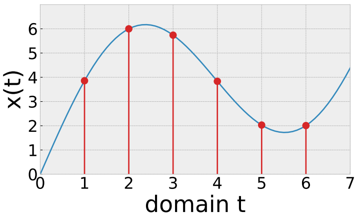


Figure 2.1: A simple discretization of a continuous-signal.

Acquiring data from the physical world and applying it to a computational framework, roughly said, can be accomplished by two components; a sensor that converts the physical signal to an electrical current or voltage, and an analog-to-digital converter (ADC), which digitalizes the current or voltage by sampling. Depending on the desired physical signal one needs a specific sensor. A membrane in a microphone can record sound waves, and a thermistor can be used to measure the temperature. Two common ways to digitize visible light is by the use of a charge-coupled device (CCD) or a complementary metal-oxide-semiconductor (CMOS). Both are integrated circuits containing arrays of coupled capacitors, able to convert optical photons to electrical charges, creating a discrete-space two-dimensional representation of the continuous-space physical world, which is commonly known as a digital image.

Discretization of a continuous signal is bound by degradation through different factors e. g. is the original signal affected by background noise, to what degree is the sensors prone to statistical or systematic errors, is the signal measured linearly e.t.c. In images such degradations provide blurred, and noise corrupted representations of the scene. One framework created to counter such effects is that of multi-frame super resolution (MFSR), where the goal is to reconstruct a high resolution (HR) image i. e., an image with high informational content, from a sequence of low resolution (LR) images.

The subjects of signal processing and imaging are interwoven, and as such, the following section iterates between the two. The elaborating of the theoretical framework is shallowly described, making room for an applied and more illustrative understanding, with a focus on the convolution operation. In § 2.1.1, a short introduction to signals and linear shift-invariant (LSI) systems is given, moving on to how it relates to image processing. In § 2.2 the image formation process is elaborated with the definition of resolution. The whole thing is then related to sampling and MFSR in § 2.3. At last, in § 2.4, additional image processing algorithms relevant to the work in this thesis is described.

2.1 CONVOLUTION OF IMAGES

An essential part of image processing is the convolution operation. In the following section a, short introduction to the theoretical framework is given, followed by applied examples.

2.1.1 *Linear Shift-Invariant Systems*

If we represent an image as a 2-dimensional signal $f(x,y)$ carrying intensity information related to a spatial domain (x,y). For now $f(x,y)$ denotes a real-valued continuous input signal and $g(x,y)$ a real-valued continuous signal, which is the output from a system described by an abstract operator $\mathcal{H}\{\cdot\}$. Illustrated in Fig. 2.2 and described as:

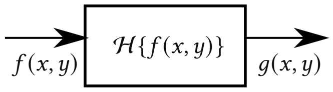


Figure 2.2: The input $f(x,y)$ converted to the output $g(x,y)$ through the system $\mathcal{H}\{\cdot\}$.

$$g(x,y) = \mathcal{H}\{f(x,y)\}.$$

An important type of system that relates to that of imaging, is the LSI systems. Linearity allows for superposition of signals e.g. given two signals $f_1(x,y)$ and $f_2(x,y)$:

$$\begin{aligned} \mathcal{H}\{\alpha \cdot f(x,y)\} &= \alpha \cdot \mathcal{H}\{f(x,y)\} \\ \mathcal{H}\{f_1(x,y) + f_2(x,y)\} &= \mathcal{H}\{f_1(x,y)\} + \mathcal{H}(\{f_2(x,y)\} \end{aligned}$$

Shift-invariance guarantees that a shift in the input domain will provide a shift in the output by the same amount:

$$g(x - x_0, y - y_0) = \mathcal{H}\{f(x - x_0, y - y_0)\},$$

where x_0 and y_0 are arbitrary shifts of the domain[9].

A fundamental feature of an LSI system is that the system operator $\mathcal{H}\{\dot{}\}$ can be described by the impulse response $h(x,y)$ and the output of the system is described by the convolution of the impulse response with the input signal[1]. Convolution for a continuous signal is described as:

$$g(t) = (h * f)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x - x', y - y')f(x', y')dx'dy',$$

where $*$ denotes the convolution operation. For the integral to be well-defined the functions h and f must decay towards infinity. An interpretation of the convolution is that the output $g(x,y)$ is a weighted average of $f(x,y)$ with weighting function $h(x,y)$. The impulse response $h(x,y)$ of the system describes the response of a system when the input signal is the Dirac delta

[1] The term impulse response comes from signal theory and goes under different names such as filters or kernels. For imaging it is often called the point-spread-function (PSF), and is elaborated in § 2.2

function i.e. $f(x, y) = \delta(x, y)$. The Dirac delta function is defined as being infinitely narrow and infinitely high:

$$\delta(x, y) := \begin{cases} 1 & \text{if } x = y = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The Dirac delta can be used to pick certain points of a given function, as the Dirac delta function can be shifted by a value (x_0, y_0):

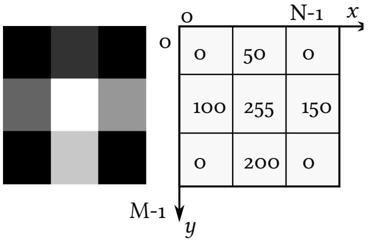
$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\delta(x - x_0, y - y_0)dxdy = g(x_0, y_0).$$

As known by the property of superposition, a signal can be represented as a linear combination of a indefinite amount of Dirac functions. This implies that the output $g(x, y)$ of any system can be described by the convolution of $h(x, y)$ with the input signal $f(x, y)$.[9]

We now apply the concept of convolution to discrete two-dimensional signals i. e., digital images. Throughout this chapter [] *notes a discrete signals and* () *a continuous signal.*

2.1.2 Images as discrete signals

A digital image, is a discrete two-dimensional signal $f[m, n]$ given on the integer locations $[m, n]$. The connection between the discrete and the continuous signal is given through sampling, which will be elaborated in § 2.3.



0	50	0
100	255	150
0	200	0

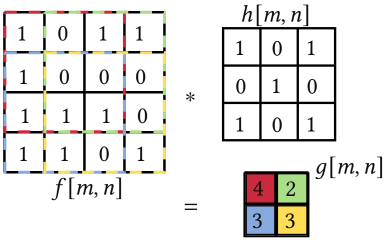
Figure 2.3: Pixel values and the corresponding gray map.

For the digital image, the different elements are named pixels which is an abbreviation of the word picture element. The pixels represent the intensity value at the specific coordinate and are in this thesis given as 8bit values i. e., the values are in the range of 0 to 255. They are commonly given as single channel(grayscale image) or three-channel(color image). The size of the image is the digital resolution and is denoted $M \times N$ i. e., the number of pixels along the rows and columns. A small illustration of a 3×3 grayscale image is given in Fig. 2.3.

The impulse response $h[m, n]$ is in the following used as a filter. The convolution with such a filter provides an output $g[m, n]$, often called the feature map. The 2D discrete convolution of an image f with the given kernel is $h[m, n]$ is defined as:

$$g[m, n] = (h * f)[m, n] = \sum_{m'=-a}^{a} \sum_{n'=-b}^{b} h[m', n']f[m - m', n - n'], \qquad (2.1)$$

where a and b is the supported range of the kernel i. e., the area it spans over the image. The convolution is the process of adding the supported neighborhood of the filter, where the filter values functions as weights. It is often illustrated as a kernel scanning over the input image producing an output at its center. An illustration of a 4×4 binary image convolved with a 3×3 kernel is given in Fig. 2.4. Given an image of size $N \times N$ and a symmetric kernel of size K, for Equation (2.1) each pixel in the image needs K^2 computations, which gives a runtime of $O(N^2K^2)$. The O is known as the big O notation and states how the runtime grows with the input size[10].



$$\underset{f[m,n]}{\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}} * \overset{h[m,n]}{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}} = \overset{g[m,n]}{\begin{bmatrix} 4 & 2 \\ 3 & 3 \end{bmatrix}}$$

Figure 2.4: Convolution of a 4×4 binary image with a 3×3 kernel. Colors indicate the supported area of the kernel contributing to the matching color in the output.

With a foundation in the discrete convolution, we proceed by presenting common filters used for image processing.

2.1.3 Image filters

The application of image filters are essential parts of image processing and is widely used for altering of desired image qualities. In the following, a demonstration of some of the typically used image filters are shown with reference to the cameraman image, Fig. 2.5.

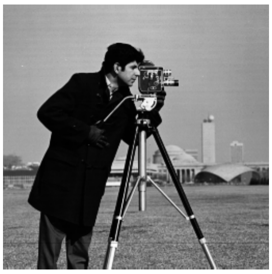


Figure 2.5: The cameraman.

Mean Filter

The mean filter is a basic filter with every element in the kernel carrying the same value. To prevent changes of overall intensities the filter is normalized over all its values.

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The cameraman convolved with the mean filter is given in Fig. 2.6

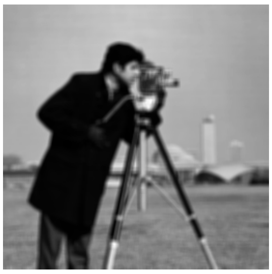


Figure 2.6: Convolution of cameraman with a mean filter of size $N = 5$.

Gaussian Filter

A common type of filter is the Gaussian, which is the standard model for blur. The kernel values are given by an isotropic zero mean 2-D gaussian:

$$h[m,n] = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-(m^2+n^2)}{2\sigma^2}},$$

where σ is the standard deviation and defines the total blur of the image. A 1D Gaussian is shown in Fig. 2.7 for various σ values. The Gaussian distribution is not only important as a common blur model but will also be used in the

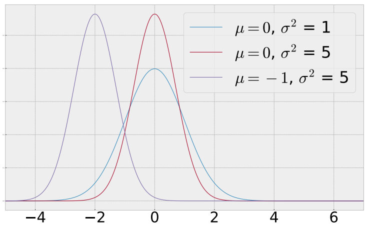


Figure 2.7: 1D Gaussian for various values of μ and σ.

later considerations of noise in an image. One simple example of a 3 × 3 kernel is:

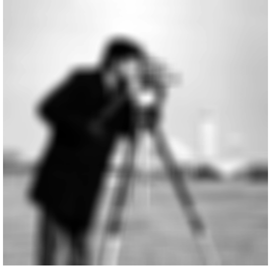


Figure 2.8: Convolution of cameraman with a gauss filter $N = 15$ and $\sigma = 1$.

$$G = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

The convolution of a 15×15 Gaussian kernel with $\sigma = 1$ and the cameraman is given in Fig. 2.8.

Sobel filter

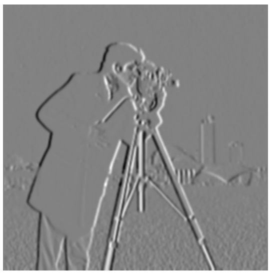


Figure 2.9: Convolution of cameraman with S_x

The Sobel filter is designed to emphasize the edges of an image. The convolution of an image with the Sobel filter produces a discrete differentiation, providing an approximation to the gradient of the image intensity. These types of filters are usually used for edge detection. The Sobel filter in the x direction is:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix},$$

The convolution with the cameraman is given in Fig. 2.9
The Sobel filter in the y-direction is given as:

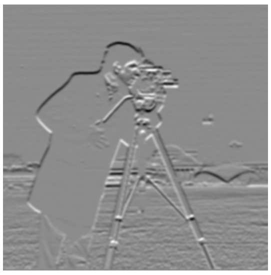


Figure 2.10: Convolution of cameraman with S_y

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix},$$

The convolution with the cameraman is given in Fig. 2.10[9].

The above filters are essential parts of central image processing tasks. The Gaussian filter will later be implemented as a critical part of the MFSR algorithm, together with the properties of image gradients, as given by the Sobel filters. We then proceed with a short explanation of the frequency of images and connects it with the Fourier transform.

2.1.4 *Fourier transform of images*

The Fourier transform of an image is a representation in the frequency domain. Frequencies in images relate to the rate of intensity changes and are decomposed to sine and cosine components. A high-frequency part of an image is characterized by rapid changes in intensity and is shown as sharp edges. The features that change gradually over large areas contain the low frequencies. An example of high and low frequency is shown for Barbara in Fig. 2.11. For a digital images of size $M \times N$ the discrete Fourier transform (DFT) is given as.

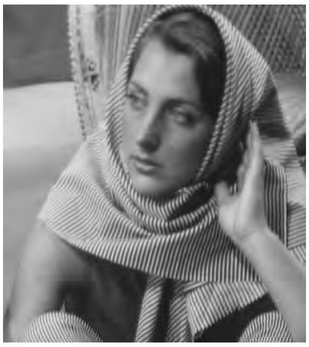


Figure 2.11: Crop of the Barbara image. High frequency parts are the stripes of the clothes. Low frequency is the background.

$$\mathcal{F}(f[m,n]) = \hat{f}[u,v] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]\exp(-\frac{2\pi imu}{M})\exp(-\frac{2\pi inv}{N}),$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform of the given signal and $\hat{f}[u,v]$ denotes the Fourier transform of $f[m,n]$. The runtime for the discrete Fourier transform is $O((MN)^2)$, as there for every index in the output of size $M \times N$ will be a sum over N and M.

Each point in the frequency domain $[u,v]$ corresponds to frequencies contained in the corresponding spatial domain. The Fourier transform of an image is illustrated in Fig. 2.12. For illustrative purposes, the absolute logarithm is taken, and by convention, the image is shifted such that high-frequency parts are placed in the center. When working with convolution the Fourier transform has an important property through the convolution theorem. The convolution theorem states that the convolution of two signals in space is identical to the point-wise multiplication in the Fourier domain:

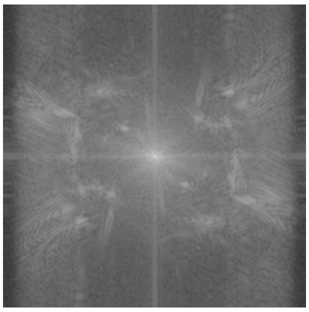


Figure 2.12: The absolute log of the Fourier transform of the Barbara image in Fig. 2.11.

$$\mathcal{F}(f * g) = \hat{f} \cdot \hat{g}$$
$$\mathcal{F}(f \cdot g) = \hat{f} * \hat{g}.$$

The above provides an alternative to Equation (2.1). As the convolution with a filter can be calculated as:

$$(h * f)[m,n] = \mathcal{F}^{-1}(\mathcal{F}(h) \cdot \mathcal{F}(f)),$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform[10]. In combination with the fast Fourier transform (FFT), the run time for an image of size $N \times N$ convolved with a symmetrical filter of size K, scales in the order of $O(N^2 \log(N))$. This property makes it more efficient, than common convolution when working with large filters.

The Fourier transform has many properties in imaging, such as filtering, reconstruction and compression. As the Fourier transform is not directly applied in this work, it is mainly mentioned for completion and comparison.
The basic concepts of the LSI and convolution can be used in the definition of imaging formation.

2.2 THE IMAGING SYSTEM

We are considering a digital image as a discrete two-dimensional signal, representing a three-dimensional continuous optical scene. From everyday use of digital cameras and mobile phones, the image is acquired through the following schematic; A light source emits light, the light reflects on an object, the reflected light passes through an optical component focusing the light rays and is finally converted to a digital image through a CCD or a CMOS. The individual pieces are essential factors in the final quality of the image, as known by a high-end camera, which produce HR images, with small blur and noise artefacts. In the following we proceed by elaborating the reason for blur in an optical system.

2.2.1 Optical system

There are two main reasons for optical blurring: Diffraction and aberrations. Diffraction is the spreading of light as it passes through a slit or an aperture. The amount of diffraction is related to the size of the slit and the wavelength of the light, which gives a physical lower limit for the possible resolution. Aberrations are due to manufacturing faults in the optical components, giving rise to uncertainties of the representation of the object. The blur can be described mathematically by the point-spread-function (PSF). The PSF correspond to the impulse response elaborated in § 2.1.1 and is a measure of how a point source from the object plane is smeared out in the image plane. In Fig. 2.13, a schematic is given of the smearing of point sources in the image plane.

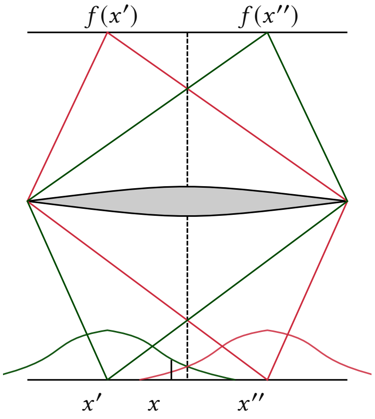


Figure 2.13: Schematic showing how a scene in the object plane is transferred to the image plane through an optical system. The optical system defines the PSF. Here the point x, takes an intensity value as a mixture of the two points $f(x')$ and $f(x'')$ related to the PSF, due to the optical system.

An ideal imaging process would have a PSF in the shape of a delta function, constructing a one-to-one copy of the original image. In real-world scenarios, the PSF is often approximated to the form of a 2-dimensional Gaussian or another parametric kernel, resulting in the PSF being a low-pass filter removing high-frequency components of the image. The consequence is that the PSF limits the given resolution of the acquired image. An illustration of how a delta and Gaussian PSF influences a simple image is given in Fig. 2.14. Notice how the four different pixels in the bottom left corner are smeared together due to the Gaussian PSF, while still being separable in the bottom right corner.

The term resolution is widely used in this thesis, and the exact meaning is interwoven between two different definitions: The optical and digital resolution.

2.2.2 Resolution

When working with digital images, the resolution of the image has two separate definitions. The optical resolution relates to which degree, different objects in a scene are separable. A question often answered by a subjective observer. A more quantitative definition is the Rayleigh resolution criterion which states the following; *Two equally bright points are barely resolved if the first zero of the curve for the PSF from the image of the first point, coincides with the central maximum of the image of the second points PSF[11].* A demonstration is given in the bottom of (c2) in Fig. 2.14, where pixel values of equal intensity overlap when convolved with the Gaussian kernel.

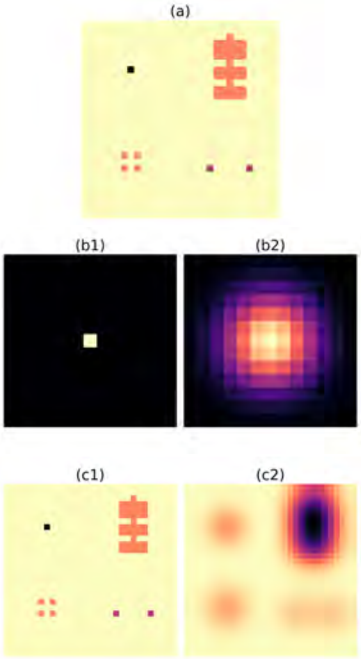


Figure 2.14: (a) An image. (b) A delta PSF and a Gaussian PSF. (c) The convolution between the image and the above PSF

The digital resolution refers to the number of pixels in an image. A pixel is a small square in a 2-dimensional grid, representing the intensity value from the light in a given scene. The value is in this thesis given in a 8bit format i. e., a value between 0 and 255 and either in a single channel(grayscale image) or three-channel(color image). The resolution is noted as the pixels per rows times the pixel per column. A LR image is visually experienced as unpleasant, as the discretization of the scene often stands clear, with every pixel being separable. A HR image is visually pleasing for the viewer, as it seems continuous and the separate pixels blend together[10]. An demonstration of a HR/LR pair is given in Fig. 2.15.

The two resolution definitions are interwoven for a digital image i. e., a high pixel resolution is superfluous if the independent objects are inseparable.

2.2.3 Image Detectors

The digital resolution is referring to the grid size of the pixel array, which is physically connected to an image sensor like the CCD or the CMOS. The sensors are constructed as a grid of photodetectors, where the incoming light hits an active pixel area. The incident photons are converted to electrons through the photoelectric effect, where they accumulate over a time interval. The amount of electrons gathered in the time interval, known as the exposure time, is proportional to the pixel intensity[10].

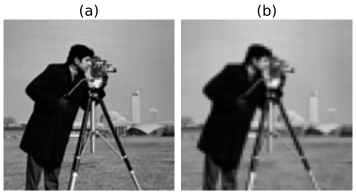


Figure 2.15: (a) a HR image (b) its LR counterpart

Large exposure time will generally in low light conditions, provide images of higher contrast, as more light is gathered. However, high exposure times are also more likely to induce motion blur in the final image, an effect due to the relative motion between the camera and the scene. When the exposure time is increased the amount of light hitting the same parts of the scene hits multiple pixels over time, which can be seen as a stretched blur over the

Figure 2.16: An X-ray image of a homogeneous rabbit under motion. Image generated with xSim elaborated in § 5.

digital image.
Another property depending on the exposure time is the saturation of pixels, where the irradiance of the incoming light exceeds the maximum electron capacity of the photodetector. A saturated pixel in a digital image carries the maximum intensity value, degrading the final quality of an image.

The quality of an image can often be described through the signal-to-noise ratio (SNR) ratio, which is a metric describing the noise corruption of a given image[2] A high super resolution (SR) corresponds to an image of higher contrast i. e., the degree to which the separate objects are distinguishable. The nature of noise in images is based on the various stochastic phenomena inherent in the different components of the detector. It is viewed as random fluctuations of the image intensity. In Fig. 2.17, the cameraman is influenced by what is known as Gaussian noise i. e., fluctuations in the intensity, following a Gaussian distribution, with a given standard deviation[10]. In image processing various tools exist, where the key is to reduce noise while preserving the inherent features of the image, in § 2.4.3 we will review two such methods.

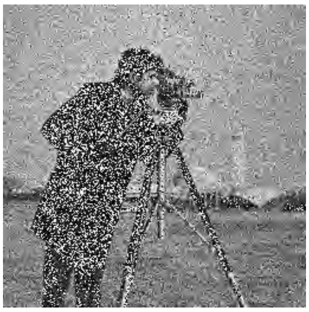


Figure 2.17: The cameraman influenced by Gaussian noise. The form of Gaussian noise will later be important for our implemented IRWSR scheme.

The two significant parameters defining the pixel resolution is the pixel pitch and the active pixel area, see Fig. 2.18 for an illustration. The pixel pitch is the distance between the centres of two neighbouring pixels and can be interpreted as the density of pixels. The pixel pitch defines the granularity of the sampling grid, which can be directly related to the spatial sampling frequency. The active pixel area also named the fill factor, is the percentage of the sensor array pixels, which are occupied by the photodetector i. e., what area of the incoming light is converted to a digital pixel. This reduced spatial responsibility of a sensor is due to electrical components occupying space in the pixel area. In general a large fill factor desirable, as it provides as higher SNR[7].

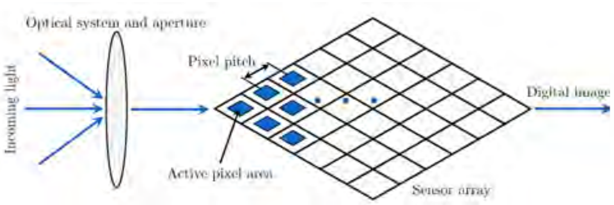


Figure 2.18: Rough schematic showing how a pixel grid gives a lacks in information due to the active pixel area and the pixel pitch. Image from [7],

[2] For now we will only consider SNR conceptually. We will in § 3.1.1 provide a more formal description.

2.3 SAMPLING

The above theory of signals and image formation can now be combined in that of sampling, where the goal is to form a representation of a continuous signal by discretization. To avoid confusion between images and frequencies, we will from now on denote an input signal/image as x and output as y.

2.3.1 *The Nyquist-Shannon sampling theorem*

Obtaining a discrete representation $y[n]$ of a continuous signal $x(t)$ requires sampling. The discrete points n is the sampling index and is related to $t = \frac{T_s}{n}$, where T_s is the sampling pitch i.e the equidistant steps for sampling. For ideal sampling the sampling operator is described by a Dirac delta comb;

$$y(t) = \sum_{m=-\infty}^{\infty} x(t)\delta(t - mT_s), \tag{2.2}$$

where $y(t)$ is the continuous representation of $y[n]$. In the Fourier domain the sampling is described by the sampling frequency $f_s = \frac{1}{T_s}$ and the sampling is given as

$$\hat{y}(k) = f_s \sum_{m=-\infty}^{\infty} \hat{x}(f - mf_s).$$

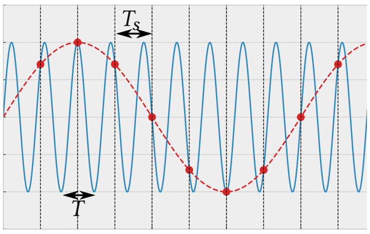


Figure 2.19: In blue a continuous-signal is shown. In red a discrete version of the signal sampled at a frequency $f_s = 1/T_s$ below the Nyquist-Shannon limit, giving rise to aliasing.

Using this definition of sampling the limits of discretization can be defined with relation to a band-limited signal.
A signal $x(t)$ is band-limited if the Fourier representation is upper-bounded by a frequency f_0. In order to get an error-free representation of a signal i. e., obtaining $y[n]$ from which it is feasible to reconstruct $x(t)$, the sampling frequency f_s should follow the Nyquist-Shannon sampling theorem. The theorem states that f_s should be twice the bandwidth f_0 of $x(t)$. If violated, frequencies higher than $f_n \equiv \frac{f_s}{2}$ are sampled as if they were of a lower frequency by the ADC[12][7]. This is known as undersampling and gives rise to the aliasing effect shown in Fig. 2.19 and the moiré pattern shown in Fig. 2.20.

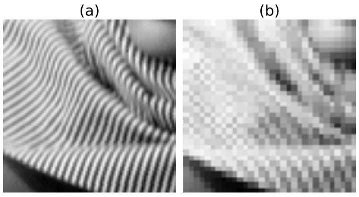


Figure 2.20: Example of spatial aliasing in the form of moiré pattern. The stripes on the image (a) changes alignment in the undersampled image (b).

In general sampling is non-ideal and the Dirac delta function in Eq. 2.2, would be replaced by the more complex impulse response $H(\cdot)$. For imaging $H(\cdot)$ is the blur operator(PSF) which functions as a low-pass filter with cut-off frequency f_h. This further limits how $x(t)$ can be reconstructed from $y[n]$,[7]. The role of blur in imaging is further described in § 2.2.

2.3.2 *Multi-Channel Sampling*

The conclusion from the previous section is that the sampling criterion and the blurring of a signal both limit an error-free sampling of an optical scene. These quantities are for real setups directly related to the quality of the measuring equipment. A natural conclusion is that a higher sampling rate leads to a better quality e. g. a CCD/CMOS with a higher pixel-density would lead to greater resolution. High-quality equipment often comes at a higher financial cost A natural follow up would then be, what is the highest possible resolution obtainable by a given system?

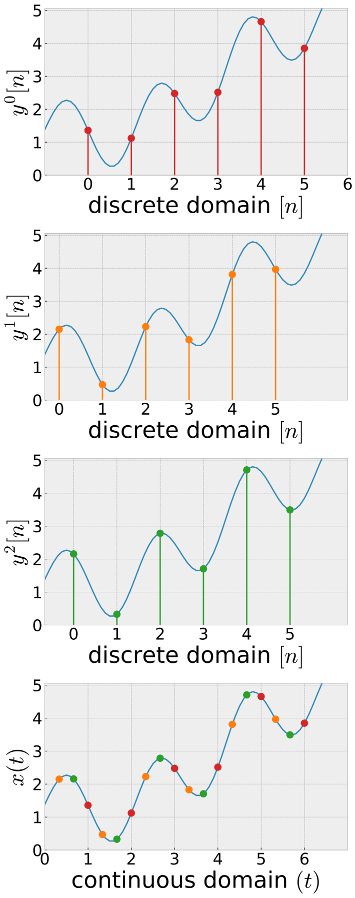


Figure 2.21: A continuous signal sampled by the same sampling frequency into 3 separate discrete channels by different offsets.

Taking an arbitrary system which discretized a one dimensional signal $x(t)$, into K multiple channels $y^k[n]$ by sampling frequency $f_s < f_0$, providing an undersampled representation. If each channel is sampled by N instances, the total amount of sampled points will be KN. In Fig. 2.21 this kind of sampling is shown, where the separate channels are complementary i. e., sampled by different offsets smaller than f_s, such that they carry different information about the original signal. It is possible to construct a discrete signal $x'[n]$, with an apparent sampling frequency f_s' consisting of sN sampling points. The magnification factor s is given as

$$s = \frac{f_s'}{f_s}.$$

Extending this to image theory, it forms a foundation for multi-frame super resolution (MFSR). Here the signal is two-dimensional, and the offsets are acquired through subpixel translations between the scene and the camera. The amount of magnification possible is bounded by both the band-limit of the original signal due to the Nyquist-Shannon sampling theorem and the cut-off frequency f_h due to blurring. This gives an effective magnification:

$$s^* = \begin{cases} \frac{1}{f_s}\min(f_s, f_h) & \text{if } f_s \geq 2f_0 \\ \frac{1}{f_s}\min(2f_0, f_h) & \text{otherwise} \end{cases}$$

The first criteria state that the signal needs to be under-sampled to enhance the resolution. If $f_h < f_0$ the blurring applies anti-aliasing, removing the essential components for the SR scheme. The lower of the two then defines the effective maximum magnification[7].

The conclusion from the above is that for digital imaging one needs to account for the PSF related to the system, as the SR cannot provide reliable results beyond this limit. Before moving on to SR as a concept, we will elaborate some of the image processing techniques considered in this thesis.

2.4 ADDITIONAL IMAGE PROCESSING

Before moving on to the MFSR framework, short reviews of important computer processing algorithms, used throughout this work is given. The algorithms are put in context of the later examined X-ray images of a circuit board, illustrated in Fig. 2.22. All processing techniques are implemented through the open-source computer vision library openCV[13].

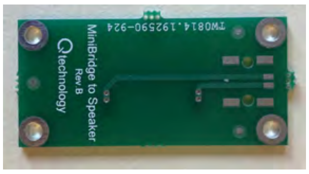


Figure 2.22: Optical image of the examined circuit board.

2.4.1 *Histogram*

An useful alternative to the representation of images in 2D pixel grids is the histogram. A typical histogram of an 8bit image is divided into the range of 256 bins showing the pixel count for the given intensity. This spectrum gives a visualization of the dynamic range of the image. Where the high end of the spectrum indicates an bright image and a low end indicates dark images. A very narrow region in the intensity range represents poor contrast, carrying low detail and as such little information. A very broad histogram indicates that all intensity values are well represented. This carries a lot of information. The use of histograms to represent images can be a strong tool, as the color mappings of images tend to misguide the viewer as they provide arbitrary interpretations of the details.

Histogram Equalization

A tool for altering the contrast of images is histogram equalization. This is the process of changing the dynamic range of the image to be equally distributed across the whole intensity scale. First we define the probability density function for the intensity range of a given image i. e., the normalization of the histogram counts.

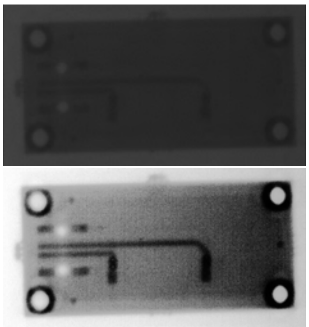


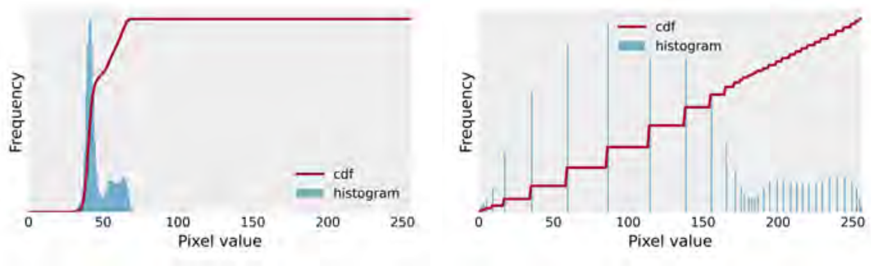
Figure 2.23: (top) Underexposed X-ray image of a circuit board. (bottom) Histogram Equalization

$$p(i) = \frac{n_i}{M \cdot N} \tag{2.3}$$

where n_i is the count of intensities i in the image of size $M \times N$. The cumulative distribution function (CDF) is then given as:

$$cdf(i) = \sum_{j=0}^{i} p(j),$$

which is the cumulative stacking of the counts, as $cdf(i)$ is the sum of all pixel elements up to the i'th intensity value. The histogram equalization process then maps the image to a linear distribution of the cdf.
The process is illustrated in Fig. 2.24 where the histogram of the circuit



(a) Normal Histogram

(b) Histogram Equalization

Figure 2.24: (a) the narrow range histogram of the circuit board in Fig. 2.23. (b) the histogram after performing histogram equalization. The CDF changes shape to an approximative straight line.

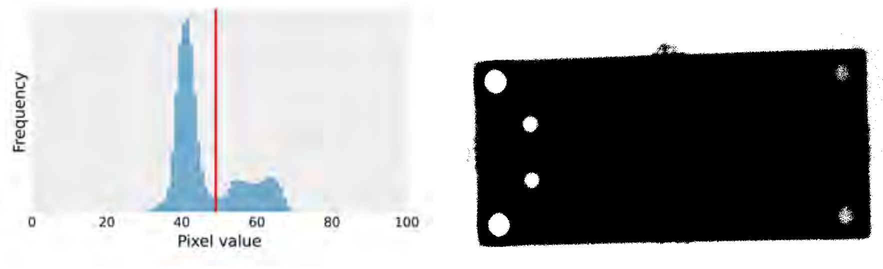
board shown in Fig. 2.23 is equalized. For the two peaks in Fig. 2.24 the CDF approximates an linear slope, and as so the intensities of the image are spread over the total intensity range. The histogram equalization considered here, works on a global scale which for some situations tend to enhance the background of images. For these cases adaptive histogram methods exists, which we will not elaborate[9].

2.4.2 *Segmentation*

A typical task in computer vision is the segmentation of various objects in a scene. The process works by dividing the different intensity values to a new range of intensities, where the levels partition the image to various regions. The most basic segmentation algorithms work through thresholding. A simple implementation is the choice of a thresholding value θ which assign the image to two classes depending on whether a pixel is larger or smaller than a chosen threshold value. For an image $f(x, y)$ the output image $g(x, y)$ is given as

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

The choice of θ can be picked by hand, with a simple estimation, based on the corresponding histogram i. e., one can look at the shape of the intensity curves, and decide where to split them. Another approach, which is in the spirit of automation, is the use of Otsu's method, which is based on a statistical analysis of the image's histogram. The image is spilt into a foreground and background class, and the aim is to maximize the inter-class variance σ_b^2. For a given θ, the probability of a pixel being in the background or foreground is



(a) $\theta = 49$ (b) Segmented Image

Figure 2.25: (a) The histogram of Fig. 2.23 with $\theta = 49$ found through Otsus method. (b) The corresponding segmented image.

defined as

$$\underbrace{P_1(\theta)}_{background} = \sum_{i=0}^{\theta} p_x(i), \quad \underbrace{P_2(\theta)}_{foreground} = 1 - P_1(\theta) = \sum_{i=\theta}^{255} p_x(i),$$

where p_x is the probability density function defined in Equation (2.3). The mean values of each class is then defined as

$$\mu_1(\theta) = \frac{1}{P_1(\theta)} \sum_{i=0}^{\theta} p_x(i)i, \quad \mu_2(\theta) = \frac{1}{P_2(\theta)} \sum_{i=\theta}^{255} p_x(i)i$$

The inter-class variance can then be expressed as

$$\sigma_b^2(\theta) = P_1(\theta)P_2(\theta)(\mu_1(\theta) - \mu_2(\theta))^2$$

As the variance is a indication of the spread of the two probability distributions, the maximization will give the maximum distance between the means of the two distributions, which is equivalent to minimizing intra-class variance. The optimal value is then found as

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \sigma_b^2(\theta), \;\; \theta \in [0, 255],$$

which can be found by iterating over all possible intensity values of the spectrum [9]. The Otsu thresholding method is used as a preprocessing algorithm for the image registration done in § 2.4.4.

2.4.3 Non-linear filters

Some applied filters in image processing are not LSI and can therefore not be performed through convolution, which is a linear operation. Here we consider two cases of importance for the later developed MFSR scheme. For demonstration we will see how they remove noise from the noise corrupted cameraman in Fig. 2.26

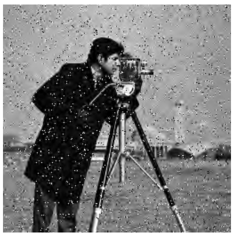


Figure 2.26: The cameraman corrupted by salt-and-pepper noise

Median filter

Applying the median to a set of sorted numbers will return the center value. A median filter uses this principle, where the set is small portions of the kernel size. An simple illustration is give in Fig. 2.27 The median filter then places the median value in the center as its output. As with other types of filters an odd size is used for a well defined center. The median works well for noise removal as the median value removes outliers without giving a contribution to the output. Contrary to the average where strong outliers shifts the output. This makes the median especially good at removing salt-and-pepper noise (dead or saturated pixels with intensity values of 0 and 255 respectively), see Fig. 2.28. Furthermore, it preserves edges better than the Gaussian filter, which as mentioned blurs the image.

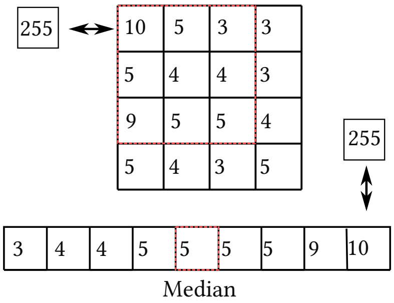


Figure 2.27: A 3 × 3 median filter applied on a 4 × 4 image. The robustness of the median is illustrated by the value being constant when changing one pixel to a saturated intensity value. In contrary, the average would change from 5.5 to 32.8.

The concept of the median is important for the robustness of the MFSR elaborated in § 3.

Bilateral filter

Another popular edge-preserving denoising filter is the bilateral filter. The concept is a weighted average of the given neighborhood of a pixel for noise removal. The process then takes the variation of the intensities into account to keep the edges. That is only similar intensity values in the neighborhood should influence the output. In some sense it is an extension to the Gaussian filter given in § 2.1.3, with the addition of the variation. The filter is given as

$$BF[I]_p = \frac{1}{W_p}\sum_{q\in S} \underbrace{G_{\sigma_s}(||p-q||)}_{spatial}\,\underbrace{G_{\sigma_r}(|I_p-I_q|)}_{intensity\,range}\, I_q,$$

where the normalization weight W_p is

$$W_p = \sum_{q\in S} G_{\sigma_s}(||p-q||)G_{\sigma_r}(|I_p-I_q|).$$

Breaking up the equation to smaller parts: $BF[I]_p$ is the output intensity of the target pixel p, and the sum runs over all q pixels in the supported neighborhood S i. e., the kernel size. The G_{σ_s} is a Gaussian with standard deviation σ_s. The weight of the Gaussian decreases the influence of q with

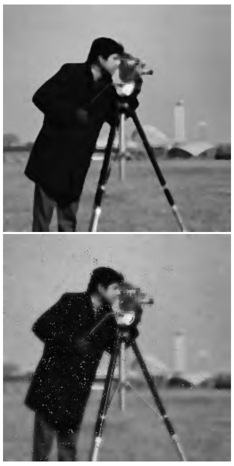


Figure 2.28: (top) noise removal by median filter of size $N = 5$. (bottom) bilateral filter of size $N = 9, \sigma_s = \sigma_r = 100$.

the distance. G_{σ_r} is a Gaussian with standard deviation σ_r. The weight of the Gaussian decreases the influence of q if the intensity value differ from I_p.

When σ_r increases it widens and becomes nearly constant over the intensity range of the image, neglecting its effect and approximation the Gaussian convolution [14].

A concept analogous to that of the bilateral filter, is implemented as a key part for the coming MFSR model.

2.4.4 *Image Registration*

A part of the coming SR scheme is the translation of a scene to separate frames in an image sequence. The goal for MFSR is to reverse the image degradation process and as such the translations should be found. This is done through image registration, which defines how to transform the separate frames into a single coordinate system. For the model examined later it is assumed that all frames are linear transformation of each other. Furthermore, we focus our attention to translations caused by a running conveyor-belt. A sufficient implementation provided by *openCV* is that of optical flow combined with Shi-Tomasi corner detection. The following gives a short descriptions of the principles behind these concepts and a demonstration of the corresponding equations. Throughout this section examples of the outputs from the implemented image registration scheme, is given in the margin. The input is two frames from an image sequence of the circuit board seen in Fig. 2.29. The first step is the preprocessing through Otsu's thresholding, see Fig. 2.30.

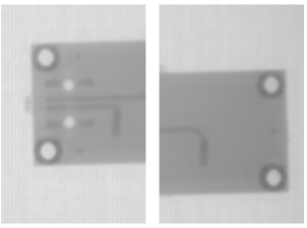


Figure 2.29: Two frames from an image sequence where the circuit board is translated. The frames are cut in half to illustrate their misalignment.

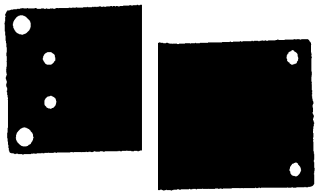


Figure 2.30: Binary images of the circuit board generated by otsu-thresholding.

Shi-Tomasi corner detector

As an edge in a image can be viewed as a sudden change in intensity values i. e., a high frequency area. A corner is then defined as the junction between two edges. The idea of Shi-Tomasi corner detection is to find the corners by looking for large gradients in the image. Mathematically it can be expressed as:

$$M(x,y) = \sum_{x,y} \underbrace{w(x,y)}_{window function} \cdot \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

where $w(x,y)$ is a weighting function with respect to the neighborhood. I_x and I_y is the image derivatives, which can be found through the Sobel filters defined in § 2.1.3. The eigenvalues of M, λ_1 and λ_2 can then be given to a scoring function.

$$R = \min(\lambda_1, \lambda_2)$$

The value of this score R is then set in context of a threshold. If they lie above the threshold, the given window is a corner. The algorithm is in openCV implemented as *goodFeaturesToTrack*.

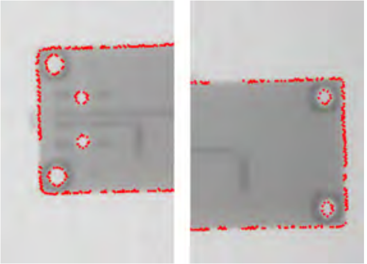


Figure 2.31: An illustration of the features detected by Shi-Tomasi Corner detection

Shi-Tomasis corner detection algorithm is an expansion of the well known Harris detection algorithm, following the same approach, but with a different scoring. The Shi-Tomasi corners and edges found for the circuit board sequence is illustrated in Fig. 2.31.

Optical Flow

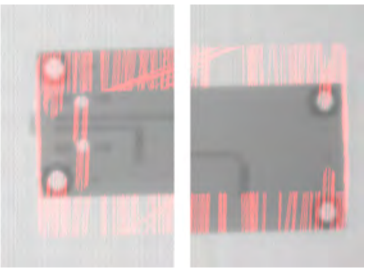


Figure 2.32: A visualization of the optical flow between the two frames.

Optical flow is a method to track apparent motion of objects in a video. The method aims to track objects between consecutive frames. Two main assumptions lies as the foundation; The intensity level between two consequent frames does not change rapidly. And that groups of pixels move together.

If a singe pixel with intensity I moves between two consecutive frames, the intensity can be expressed as a function of the spatial and temporal coordinates and as we assume the intensity is constant

$$I(x, y, t) = I(x + \partial x, y + y, t + \partial t)$$

The above equation can through a Taylor expansion be converted to what is known as the optical flow constraint equation:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{2.4}$$

with $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial x}{\partial t}$.
The problem of optical flow is to solve for the parameters u and v as they define the movement of the object over time. The equation for it self is under-determined, as only one equation is provided for two constraints. This problem can in connection with the edges found, be converted to a set of equations. This is named the Lucas-Kanade method. The method takes a small window of size $N \times N$ and places it around the points found by the edge detection algorithm. Assuming that all points carries the same motion. This provides a set of N^2 equation on the form of Equation (2.4), which is an over-determined problem. The problem is then converted to two equations through least squares fitting, which when solved gives the translation vector between the two frames[15].

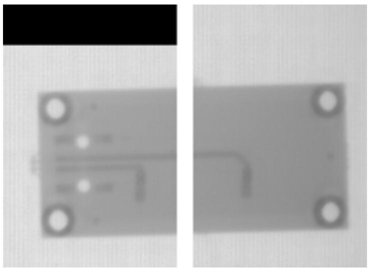


Figure 2.33: The left frame translated with the motion estimate found through optical flow.

In our work the sparse optical flow is implemented, focusing on defined features - such as edges or corners. A contrary to this, is dense sparse flow which gives the flow of all pixels in the frames. Giving rise to a higher accuracy at higher computational cost . A result from the implemented algorithm is demonstrated in Fig. 2.32 and the corresponding translation of the left frame is viewed in Fig. 2.33.

The process of image registration can be done in various ways and depends on the specific situation. When applying image registration to an automatic online process i. e., processing while data is acquired, the most important property is that it efficient and robust. However, in our implementation this has not been given much thought, as the bottleneck for a sufficient runtime is the implementation of the MFSR scheme implemented. In regards to the robustness, the method works well in general, but fails with increased motion-blur, this will be elaborated in § 6.3.

3 SUPER RESOLUTION

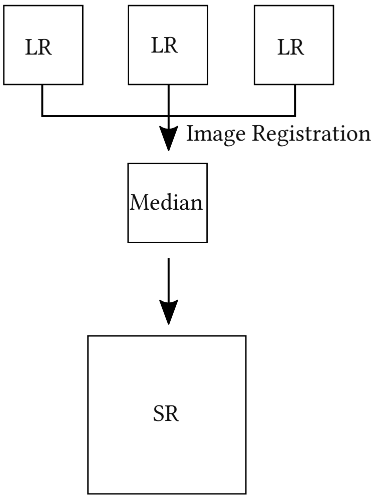


Figure 3.1: Simple schematic of MFSR.

Super Resolution (SR) is the process of recovering a high resolution (HR) digital image from either a single or a set of low resolution (LR) digital images. It can be viewed as complex upscaling aiming to construct HR images with a higher informational content. Creating more visually pleasing results than standard upscaling methods such as the bi-cubic interpolation. In general, the methods are a computational compromise to expensive hardware, enabling low-cost solutions to high-quality imaging.

The process of super resolution (SR) is itself an inverse problem, where one seeks to find the casual factors that created them observations. The observations are the provided LR frames constructed from a unknown HR image. The connection between the two is the image degradation process, described mathematically through different operations of translation, blur and downscaling. The blur often works as an low-pass filter removing essential high-frequency parts of the image. The shape of the degradation is impossible to identify completely and as such SR is an ill-posed problem. The ill-posed problem is defined through its counterpart, the well-posed problem. This is a problem carrying three qualities; (1) Its solution exists, (2) it is an unique solution and (3) the solution is stable under perturbation. A common way to solve an ill-posed problem is through Bayesian regularization, where prior information is induced to make the task well-posed.[16]
The iterative re-weighted super resolution (IRWSR) algorithm we will consider is a Bayesian approach to MFSR. A rough graphical outline of the algorithm is given in Fig. 3.1

This chapter provides an analysis of the SR schemes considered in this thesis, comparing and contiuning the discussion about the theory of image processing found in the previous chapter. Furthermore, it elaborates on the implemented model of IRWSR by [1], as for this reason the next chapter is based on the original article, which proposed the scheme, and the later published P.hD thesis [7]. In § 3.1, an overview of SR is given with a short discussion of considered approaches. In § 3.2, the image degration process is explained as the foundation for SR. In § 3.3, the Baysian framework for the IRWSR is elaborated and at last in § 3.4, an outline of the IRWSR scheme is given, with examples of the intermediate outputs.

3.1 SUPER RESOLUTION - AN OVERVIEW

In the last three decades several SR algorithms have been developed. In fact, the amount of proposed solutions for SR seems as ill-posed as the problem itself. The SR methods can be divided into two main categories: single image super resolution (SISR) or multi-frame super resolution (MFSR).

A common approach to SISR is the statistical self-similarity method employed by the use of small image patches such as in [17]. Here a single image is downscaled to a set of low-resolution patches constructing a set of HR/LR pairs, forming a set of linear constraints used to interpolate the SR image from the original HR image. Another similar HR/LR approach is given in [18], where a convolutional neural network (CNN) is trained to interpolate the HR image from the artificially degraded LR image. In [2] the LR/HR patches are decomposed into a sparse framework and used to train a set of dictionaries, which forms the basis of the SR interpolation. A general assumption in these types of algorithms is that the unknown point-spread-function (PSF) is a low-pass filter such as a Gaussian. In [19] this problem is addressed by developing a scheme for blind estimation of a non-parametric PSF, found by a maximum a posteriori (MAP)[1] of the HR/LR set.

In MFSR the multiple LR frames carrying complementary information due to subpixel movement are fused together through motion compensation. The first proposed attempt is given in [20] and is iterative analog to the back-projection schemes used in tomography. The original approach for the implementation in this thesis, was to use the kernel estimation from [19] in combination with the sparse representation in [2]. This approach was abandoned due to the lack of documentation from [19], and that [2] was to cumbersome to have a realistic implementation on an online system. A short review of [19] is provided inn § A.1.1.

The MFSR method presented by Köhler in [1] is especially interesting. Here a MFSR algorithm based on regularization by bilateral total variation, is used in an IRWSR. The proposed method is evaluated and compared to other SR algorithms in [4], where it is quantitatively outperforming other methods in matters of robustness i. e., performing high quality SR across different scales of degradataion such as poor motion estimates. Showing that it is a promising candidate for the further developed scheme is this thesis.

For comparison of SR algorithms, different metrics have been implemented, two of the most popular being the point signal-to-noise ratio (PSNR) and the structural similarity index (SSIM).

[1] The MAP will be elaborated in § 3.3

3.1.1 Quantifying SR algorithms

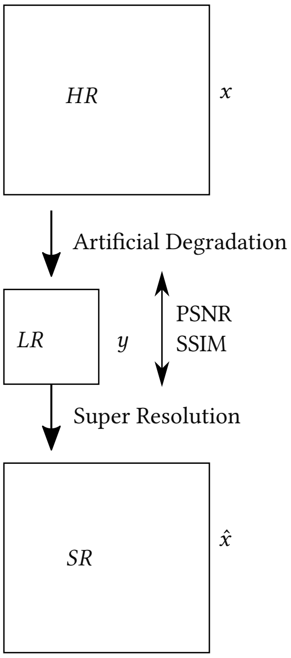


Figure 3.2: An HR image artificially degraded and then upscaled to a SR image. The two images are then compared through the two metrics PSNR and the SSIM.

There exists a vast amount of SR algorithms, based on choices of regularization prior, estimation method, optimization and image registration. All these algorithms differ in qualities such as computational time, SR image quality and robustness to variations in noise, radiometric and translation model. A natural consequence of generating SR images from either a single or multiple LR images is the lack of a HR reference. The task of quality assessment is then left with respect to an HR image x, which is artificially degraded to its LR counterparts y and then upscaled with SR to a new HR image $\hat{x}$. Two common quantitative measures used for measuring different features in the SR literature is the PSNR based on intensities and the SSIM based on structural information. A simple schematic of the comparison process is given in Fig. 3.2.

PSNR: The signal-to-noise ratio is the ratio of desired signal over background noise in terms of power, which due to a wide dynamic range is expressed in a logarithmic decibel scale:

$$\mathrm{SNR}_{dB} = 10\log_{10}(\frac{P_{signal}}{P_{noise}}).$$

the more convenient PSNR is used in digtial imaging, which is pixel based. The PSNR is based on the mean squared error (MSE), which is the L_2 norm of the intensity difference between the separate pixels in a reference image x of size $m \times n$ and the noise corrupted image $\hat{x}$ of identical size.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[x(i,j) - \hat{x}(i,j)]^2.$$

The PSNR is then defined as:

$$\mathrm{PSNR} = 10\log_{10}(\frac{L}{MSE}),$$

where L is the dynamic range of the intensity values for the image - in a 8 bit representation being 255.

SSIM: The Structural Similarity index was developed as a method for quantifying the degradation of human perception when compressing an image. The definition of the SSIM is a weighted combination of the luminance, contrast and structure between the two samples x and $\hat{x}$. Giving a final expression as:

$$\mathrm{SSIM}(x,\hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + C_1)(2\sigma_{x\hat{x}} + C_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2)}$$

where μ denotes the mean, σ the standard deviation, $\sigma_{x\hat{x}}$ the covariance and $C_1 = (k_1L)^2$, $C_2 = (k_2L)^2$ are stabilizing constants with $k_1 = 0.01$ and $k_2 = 0.03$. As a contrast to the PSNR, the SSIM is not estimating absolute errors, but instead perceived changes in the structural information[21].

The above two metrics will be applied to our SR outputs when feasible i. e., for real data, where no reference is given. We now proceed with the foundation for the implemented SR algorithm - the image degradation model.

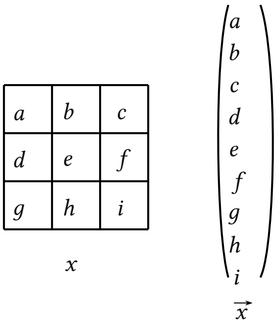


Figure 3.3: Illustration of the vector representation of an image, sorted in rows.

3.2 IMAGE DEGRADATION MODEL

Given a sequence of low resolution frames, the MFSR task is to recover the unknown high resolution image. That is we seek to reverse the degradation from a high resolution digital image $\vec{x}$ to a low resolution frame $\vec{y}^{(k)}$. Throughout this chapter the notation of an image is given as a vector $\vec{x}$. This allows an extension of image processing such as convolution, to that of a dot product between the image and a matrix representation of the desired kernel. In Fig. 3.3, the representation of an image as a vector is showed. The degradation process consists of a translation, blurring, down-sampling and additive noise. Where each frame k is acquired through different translations, giving a total of K individual low resolution frames. Mathematically each component can be expressed as a matrix operator which gives the following relation:

$$\begin{aligned}\vec{y}^{(k)} &= \boldsymbol{DHM^{(k)}}\vec{x} + \vec{\epsilon}^{(k)} \\ &= \boldsymbol{W}^{(k)}\vec{x} + \vec{\epsilon}^{(k)},\end{aligned} \tag{3.1}$$

where $\boldsymbol{D}$ is the decimation(down-sampling) operator, $\boldsymbol{H}$ is the blurring operator, $\boldsymbol{M^{(k)}}$ is the translation of the k'th frame and $\vec{\epsilon}^{k}$ is the noise in the k'th frame[4].

The three operators are gathered together is the degradation matrix $\boldsymbol{W}^{(k)} = \boldsymbol{DHM}^{(k)}$. In a computational framework $\boldsymbol{W}^{(k)}$ carries the whole degradation process except the additive noise $\vec{\epsilon}^{(k)}$. In Fig. 3.4, an illustration of the image degradation process is given.

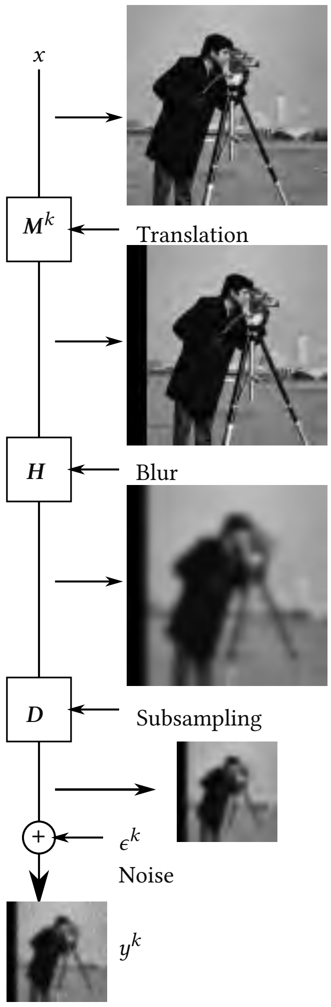


Figure 3.4: The image degradation process for the cameraman image.

In the following two computationally different approaches for image degradation is presented. The one in focus is $\boldsymbol{W}^{(k)}$ expressed as a sparse matrix and the LR frames are found through the dot-product with $\vec{x}$. An alternative approach is the degradation of the HR image through standard image processing methods, such as filtering.

3.2.1 *Matrix representation of degradation*

Considering a HR frame $\vec{x}$ of length N, the given LR frame $\vec{y}$ will be of size $M = N/s$, where s is the magnification factor. The whole degradation system $\boldsymbol{W}^{(k)}$ is then a matrix of size $M \times N$, where each row indicates the pixels from x which contribute to the value of a pixel u_m in y. Building $\boldsymbol{W}$ depends on the data at hand and is illustrated in Fig. 3.5.

Using a motion estimate for the k'th frame $\boldsymbol{t}^{(k)} = (t_h^{(k)}, t_v^{(k)})$, where $t_h^{(k)}$ and $t_v^{(k)}$, is the translation in the horizontal and vertical direction respectively. The relation between a pixel at position $\boldsymbol{u_m}$ in the low-resolution frame and the counterpart $\boldsymbol{u'_m}$ in the high-resolution frame is $\boldsymbol{u'_m} = \boldsymbol{u_m} + \boldsymbol{t}^{(k)}s$.

The blurring $h(\cdot)$ is related to the PSF of the system and is supported within a radius defined by the standard deviation ω_{PSF}. Assuming that the contribution from a pixel v_m in the high resolution frame is only contributing to u_m, if it lies within the area defined by $\omega_{PSF}(\boldsymbol{u'_m})$. The above can be summarized in the definition of each component in $\boldsymbol{W}$:

$$W_{mn} = \begin{cases} \frac{1}{\sum_{i=1}^{N} h(u'_m - v_n)} h(u'_m - v_n) & v_n \in \omega_{PSF}(u'_m) \\ 0 & \text{otherwise,} \end{cases}$$

where each component is normalized with respect to the supported area. The benefit of this is that a continuous version of the PSF can be used in the $\boldsymbol{W}$ scheme. A common approximation for an unknown systems, is the PSF in the shape of a zero mean 2-dimensional Gaussian with standard deviation σ_{PSF}. In combination with the Gaussian a model for the supported area in the high-resolution frame is created:

$$h(\boldsymbol{u}) = \exp(-\frac{1}{2}||\boldsymbol{u}||_2^2/(s^2\sigma_{PSF}^2)). \tag{3.2}$$

Setting $\omega_{PSF} = 3\sigma_{PSF}$ ensures 99.7% of the Gaussian is within the supported area. The resulting matrix $\boldsymbol{W}$ would be best represented by a sparse matrix[2], as the amount of non-zero inputs will be $O(KMN_{psf})$, where K is the amount of LR frames and N_{psf} is the radius of the supported part of the PSF[7].

In the applied scheme $\boldsymbol{W}$ is created by iterating over every pixel in all of the low resolution frames. An simplified version of the implemented code is given in Code Listing A.2. The motion estimates are given through the image registration process described in § 2.4.4.

An alternative to the previous approach is to separate each degradation step into individual image operations. The motion translation $\boldsymbol{M}^{(k)}$ is calculated through direct translation using the motion estimates. The blur $\boldsymbol{H}$ is calculated through a discrete convolution, see § 2.1.2. The down-sampling can be implemented through nearest neighbor interpolation.[3]

[2] A sparse matrix is, as the name suggest, a matrix where most of the elements are zero. Memory wise only the non-zero elements are stored in a list referring to their indices.

[3] The degradation through filtering is used for generating artificial LR frames for examination of the HR process.

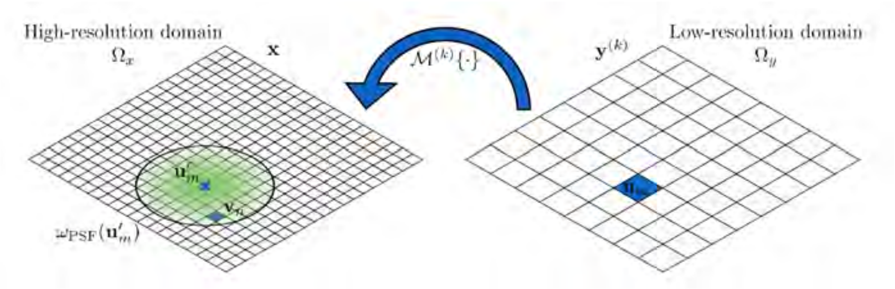


Figure 3.5: Illustration of the relation between a pixel in the low-resolution frame and the pixel values contributing from the high-resolution image. Image from [7]

This approach of filtering avoids the nuisance of creating and storing a sparse matrix. However, this approach is not straightforward to implement in the MAP given in § 3.3 and as such the sparse implementation is preferred. Another advantage of the matrix representation of $\boldsymbol{W}$ is that it can be precomputed. For a system repeating the same structural degradation, the whole matrix is only calculated once and then repeatedly used. Furthermore, the matrix calculation can be evaluated in a parallel manner, as each point is independent.

Depending of the implementation of the image degradation, several approximations are made. The above motion translations assume a rigid motion of the overall frame and as so, does not include the independent motion of individual objects in the scene. The overall scheme should only work for multiple frames gathered by camera translations on a still object, or possibly in the case of a conveyor-belt where the entire scope of the frame is moved simultaneously. The problem of independent movements is also apparent in the approximation of the PSF being space-invariant. The independent motion blur will cause a PSF of spatial dependencies.

With a foundation in the degradation model, we proceed with another important step of our implemented scheme. The Bayesian framework used to stabilize the problem through the method of regularization.

3.3 BAYESIAN FRAMEWORK FOR MFSR

Based on conditional probability, the Bayesian theorem forms a foundation for the examined MFSR model. In this framework, the high resolution image and low resolution frames are modeled as random variables and related through probability density functions $p(\cdot)$. Bayes theorem is given as

$$\underbrace{p(x|y)}_{\text{posterior}} = \frac{p(y|x)p(x)}{p(y)} \propto \underbrace{p(y|x)}_{\text{conditional probability}} \underbrace{p(x)}_{\text{prior}},$$

where the denominator is neglected, as the contribution is a positive constant. The posterior $p(x|y)$ is the condition probability of obtaining x given data y. $p(y^{(k)}|x)$ is the conditional probability of obtaining a low-resolution frame $y^{(k)}$ from x, through the image degradation process given in Equation (3.1). The prior $p(x)$ describes the statistical properties of x. The goal of the SR process is to estimate the image x which best explains the low-resolution frames. Commonly done through the maximum a posteriori (MAP) method, finding the x which maximizes the posterior:

$$x_{MAP} = \underset{x}{\operatorname{argmax}}\, p(y|x)p(x), \tag{3.3}$$

by taking the negative logarithm, the problem is converted to a minimization problem:

$$x_{MAP} = \underset{x}{\operatorname{argmin}} \big\{ -\log(p(y|x)) - \log(p(x)) \big\} = \underset{x}{\operatorname{argmin}} \big\{ L(x) + \lambda R(x) \big\},$$

where $L(x)$ is the negative log-likelihood function[4], λ is the regularization weight and $R(x)$ is the chosen prior distribution. The minimization is noted as an overall minimization of an energy function.

$$x_{MAP} = \underset{x}{\operatorname{argmin}}\, F(\boldsymbol{x}) \tag{3.4}$$

The energy function $F(\boldsymbol{x})$ is divided into two essential parts; The Likelihood term $L(x)$ and the regularization term $R(x)$.

The shape of $L(x)$ and $R(x)$ can take many shapes, depending on the statistical assumptions made for the problem. In the SR literature various suggestions with different qualities has been proposed. A complete analysis of all these methods are out of the scope for this thesis. We proceed by elaborating the choices of [1], which is based on the image residuals and a Gaussian assumption of the image noise.

3.3.1 The Likelihood term

The first term of the energy function is the negative log-likelihood term $L(x)$, which is described through a observation model. Giving the probability of

[4] The negative log-likelihood comes from maximum likelihood (ML) estimation, an estimation technique not accounting the prior. The regularization by a prior is needed for a conversion to a stable solution, for an ill-posed problem[7]

obtaining a low-resolution frame $y^{(k)}$ as a conditioned probability $p(y^{(k)}|x)$. The distribution $p(y^{(k)})$ follows the image degradation process in equation 3.1, and is dependent of the image noise $\epsilon^{(k)}$. As such the whole structure of $p(y^{(k)}|x)$ is dependent on an assumption of the noise distribution. The authors of the IRWSR proposes a model with a spatial weighted Gaussian. This allows the standard deviation of the noise distribution to change with the spatial location. Giving:

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\beta}) \propto \exp\left\{-\frac{(y-\boldsymbol{W}x)^T\boldsymbol{B}(y-\boldsymbol{W}x)}{2\sigma^2_{noise}}\right\}. \tag{3.5}$$

Providing a negative log-likelihood function as

$$L(x) = \frac{1}{2\sigma^2_{noise}}(y-\boldsymbol{W}x)^T\boldsymbol{B}(y-\boldsymbol{W}x), \tag{3.6}$$

where σ_{noise} is a scale parameter and $\boldsymbol{B}$ is a diagonal matrix containing the confidence weights $\beta_1, \dots, \beta_{KM}$. The confidence weights β_m is defined in the interval $[0,1]$. A value of 0 indicates an outlier regarding the distribution and 1 indicates an inlier. The estimation of $\boldsymbol{B}$ is given in § 3.4.2. The part $y-\boldsymbol{W}x$ is later denoted as the residual between the given LR frames and the estimated guess for an SR image x.

With the first part of the MAP defined, we proceed with the elaboration of the chosen prior.

3.3.2 The Regularization term

The second part of the energy function relates to the prior. The prior $p(x)$ carries statistical properties of the HR image x, and is dependent on the regularization term $R(x)$. The $R(x)$ can be chosen to best solve the task at hand, and different choices will suppress and enhance various qualities in the image. A common choice is the Gaussian prior, which considers x to be spatial smooth. The authors of the IRWSR argues that the use of a bilateral total variation (**BTV**) prior. A generalization of the total variation prior, which is common for denoising of images. The total variation prior seeks to minimize the absolute gradients in the vertical and horizontal directions of the image. The bilateral part is an extension to gradients in multiple directions, and is inspired by bilateral filtering, see § 2.4.3. This is done through different combinations of shifts in the horizontal and vertical direction:

$$R_{BTV}(x) = \sum_{m=-P}^{P}\sum_{n=-P}^{P} \alpha_{BTV}^{|m|+|n|}||S^{m,n}\mathbf{x}||_1,$$

$$S^{m,n} = ||\mathbf{I}_{N\times N} - \mathbf{S}_v^m \mathbf{S}_h^n||_1,$$

S_v^m and S_h^n denotes shifts of x by m and n pixels in the vertical and horizontal direction respectively. P defines the window size and α_{BTV} weights the difference. The regularization can be interpreted as a multi-scale analysis of the image gradients. A matrix S assembling the different components of the operator is defined as:

$$S = (S^{-P,-P} S^{-P+1,-P} \dots S^{P,P}). \tag{3.7}$$

For implementation this is created as a diagonal block matrix, where each block in the diagonal is the gradient in the given direction. The value of P defines the amount of different combinations of the horizontal and vertical shift. The implementation of S as a matrix is given in Code Listing A.1.

The authors of the IRWSR argues that the edge preserving of the above prior is limited as it is not spatial adaptive. They proceed by proposing a spatial weighted version, given as a zero-mean weighted Laplacian distribution[5]:

$$p(x|\alpha) := \frac{1}{Z(\sigma_{prior}, \alpha)} exp\{-\frac{||\boldsymbol{AS}||_1}{\sigma_{prior}}\},$$

where σ_{prior} is the distribution scale parameter, α is the confidence weights set in a diagonal matrix $\mathbf{A} = diag(\alpha_i, ..., \alpha_{N'})$ and $Z(\sigma_{prior}, \alpha)$ is a normalization constant. In combination with the above we get the weighted bilateral total variation prior:

$$R(x|\alpha) = ||\boldsymbol{AS}x||_1,$$

where the weight ensures that the regularization of discontinuities are reduced when $\alpha_n = 0$ and performs smoothing in homogeneous regions where $\alpha_n = 1$. The estimation of $\boldsymbol{A}$ is given in § 3.4.2.

The goal of the SR algorithm is to estimate the unknown HR image $\hat{x}$, which is calculated trough the minimization of the MAP given in Equation 3.3. As both the prior term and the negative log-likelihood is spatially weighted the equation takes the form:

$$\hat{x} = \underset{x}{\operatorname{argmax}}\{p(x|\alpha) \prod_{k=1}^{K} p(y^{(k)}|x, \beta^{(k)})\}, \tag{3.8}$$

where it is assumed that each frame is drawn independently from its probability distribution. In the above equation both α and β works as latent variables i. e., they are random variables we do not know a priori. The authors of the IRWSR suggests the estimation of the weights through a majorization-minimization (MM) scheme, as the distribution of $p(\alpha)$ and $p(\beta)$ is difficult to

[5] The argument of the Laplacian is motivated through the field of natural image statistics, which is out of the scope of this thesis. We refer to a analysis made in[7].

model for real-world problems. The (MM) scheme is a way to iterate between the process of estimating α and β and the estimation of $\hat{x}$.

Combining the above with the degradation model $\boldsymbol{W}$, the following section provides an outline of the IRWSR scheme.

3.4 ITERATIVELY RE-WEIGHTED SUPER RESOLUTION

The method developed by [1] is based on a vast amount of statistics and inverse problem theory. As we are mainly focusing on a reliable implementation, main parts of the analysis is left out with references to the authors original work. The MAP estimate in Equation 3.8 forms the starting point for the iterative re-weighted super resolution (IRWSR). The whole scheme alters between estimating $\hat{x}$ and the latent weights α, β and λ. As a whole the iterations t is set in a coarse-to-fine scheme, which gradually increases the magnification of the image by steps Δs. To follow the progress, we will in the margin provide examples of a LR sequence being super-resolved. In Fig. 3.6 a reference image of Barbara is shown and in Fig. 3.7 its artificially degraded counterparts. The choice of Barbara is based on the high-frequency stripes disappearing in the degradation process, as such a reconstruction of those features is a good indication for whether or not our implementation works. The degradation is done through the filtering process described in § 3.2.

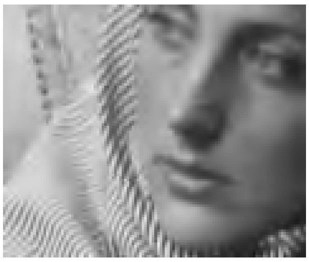


Figure 3.6: Reference image of Barbara.

3.4.1 *Outline of the algorithm*

The overall process is the following for the t'th iteration:

1. The reconstructed image x^{t-1} found at the previous step is magnified by Δs, through bi-cubic interpolation. The weights are estimated as: $\alpha^t := \alpha(x^{t-1}|\sigma^t_{prior})$, $\beta^t := \beta(x^{t-1}|\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(K)}, \sigma^t_{noise})$.
2. The scale parameters σ^t_{noise} and σ^t_{prior} together with the regularization weight λ is estimated.
3. Using the previous estimated weights x^t is gathered through a MAP estimation.

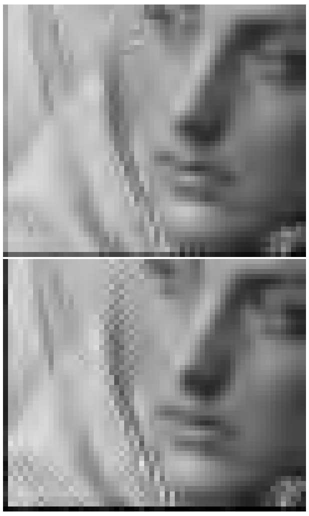


Figure 3.7: Two frames from a sequence of artificially degraded images of Barbara.

For an initialization of the scheme the initial weights α^0 and β^0 is set to all one vectors of their respective lengths. The first estimate of x^0 is set to the motion-compensated median of the low-resolution frames i. e., the LR frames are aligned and the median value taken over all K frames. The motion-compensated median for the Barbara image sequence is shown in Fig. 3.8. The estimation of x^t is based on the energy function found in Equation 3.4, which with our choice of prior and observation model takes the shape:

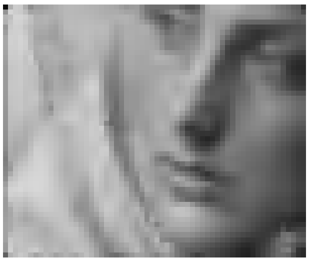


Figure 3.8: The motion-compensated temporal median of the Barbara image sequence.

$$F^t(\boldsymbol{x}) = (\boldsymbol{y} - \boldsymbol{Wx})^T \boldsymbol{B}^t (\boldsymbol{y} - \boldsymbol{Wx}) + \lambda^t ||\boldsymbol{A}^t \boldsymbol{Sx}||_1.$$

With the corresponding gradient:

$$\nabla_x F^t(\boldsymbol{x}) = -2\boldsymbol{W}^T \boldsymbol{B}^t (\boldsymbol{y} - \boldsymbol{W}x) + \lambda^t \boldsymbol{S}^T \boldsymbol{A}^t \operatorname{sign}(\boldsymbol{A}^t \boldsymbol{S}x) \qquad (3.9)$$

Where the regularization term is approximated through a smooth and differentiable function[6]:

$$R(x|\alpha) = ||\boldsymbol{ASx}||_1 \approx \sum_{i=1}^{N'} \alpha_i \sqrt{[\boldsymbol{Sx}]_{\boldsymbol{i}}^2 + \tau},$$

which for $\tau = 10^{-4}$ is a reliable approximation[1].

For our implementation x^t is estimated through the conjugate gradients (**CG**) method[22] through the scipy library[23].

3.4.2 *Estimating weights*

The observation weights β are determined through the residuals $r(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{Wx}$, where y is a vector of all the given low-resolution frames of length KM and x is the estimated super resolved image. Large residuals will then be downweighted, lowering the contribution to the MAP estimate. In the proposed scheme there will be a β value for each point. The β weights are decomposed into two weights. One which accounts for frame-wise outliers and one which accounts for pixel-wise outliers:

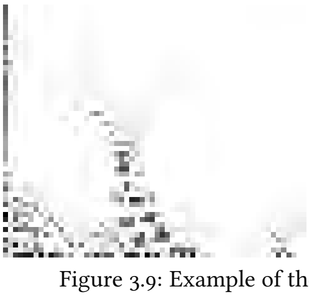


Figure 3.9: Example of the β map. Calculated from the residual between the temporal median and a single LR frame from the degraded Barbara sequence.

$$\beta_i(\boldsymbol{r}|\sigma_{noise}) := \beta_{i,bias}(\boldsymbol{r})\dot{\beta}_{i,local}(\boldsymbol{r}|\sigma_{noise}).$$

The bias estimations goal is to find global outliers and suppress them i.e. frames with wrong motion estimations and degraded by motion blur. Using the assumption given in Equation (3.5), that is the noise is given as a zero-mean Gaussian, the residual for an individual frame should have zero-mean. The bias weight for a single frame is set as the overall median being below a given threshold:

$$\beta_{i,bias} = \begin{cases} 1 & \text{if } |\text{MED}(r^{(k)})| \leq r_{max} \\ 0 & \text{otherwise,} \end{cases},$$

[6] The proof can be found in [7].

where MED denotes the median, $r^{(k)}$ is the residual related to the k'th frame and r_{max} is chosen to be 2% of the maximum intensity[7]. The weighting for local outliers is performed on a per-pixel basis and is defined as:

$$\beta_{i,local} = \begin{cases} 1 & \text{if } |r_i| \leq c\sigma_{noise} \\ \frac{c\sigma_{noise}}{|r_i|} & \text{otherwise,} \end{cases},$$

where σ_{noise} is the estimate of the standard deviation for the weighted normal distribution and $c = 2$ is a tuning constant. The inverse weighting of outliers gradually down weights pixels, which are related to non-Gaussian noise. An example of the β map for the residual between a single LR frame and the temporal median is given in Fig. 3.9

Determining the prior weights is related to a value defined as $z = Sx$, where S is the matrix defined in Equation (3.7). The weights are then defined as:

$$\alpha_i(z|\sigma_{prior}) = \begin{cases} 1 & \text{if } |[Q(z)]_i| \leq c_{prior}\sigma_{prior} \\ p\frac{(c_{prior}\sigma_{prior})^{1-p}}{|[Q(z)]_i|^{1-p}} & \text{otherwise,} \end{cases},$$

where $p \in [0, 1]$ is the sparsity parameter and σ_{prior} is a scaling parameter. The filter $\boldsymbol{Q}$ reduces noise and is implemented as a median filter, elaborated in § 2.4.3. As S is promoting edges in the image, discontinuities in the image will be down weighted accordingly. An example of the α map for the temporal median is given in Fig. 3.10.

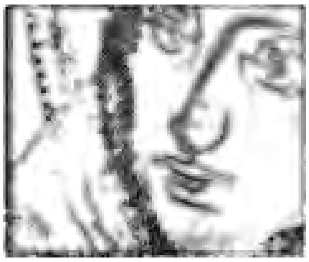


Figure 3.10: Example of the α map, represented by the sum of the gradients found in all directions for the temporal median of the degraded Barbara sequence

3.4.3 Estimating parameters

The two scale parameters σ^t_{noise} and σ^t_{prior} are estimated at each iteration through a maximum likelihood (ML) estimation. Here the conditional probabilities are based on previous estimated values for β^{t-1}, α^{t-1} and $\hat{x}^{t-1}$. For σ^t_{noise} giving:

$$\sigma^t_{noise} = \underset{\sigma_{noise}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{x}^{t-1}, \beta^{t-1}, \sigma_{noise}).$$

For a robust ML estimation the authors propose the median absolute deviation (MAD) of the residual. Using the estimated β^{t-1} as weights, σ_{noise} is found as

$$\begin{aligned} \sigma_{noise} &= \sigma_0 \cdot \text{MAD}(\boldsymbol{r}^{t-1}|\beta^{t-1}) \\ &= \sigma_0 \cdot \underset{i=1,\ldots,KM}{\text{MED}}(|r_i^{t-1} - \text{MED}(r^{t-1}|\beta^{t-1})|\beta^{t-1})' \end{aligned}$$

where $\sigma_0 = 1.4826$ is based on the Gaussian assumption for inliers[24].

[7] The value originally used by the author

Following the same procedure for σ_{prior}, a robust estimate is given as

$$\sigma_{prior}^{t} = \sigma_0 \cdot \text{MAD}(\boldsymbol{S}\boldsymbol{x}^{t-1}|\boldsymbol{\alpha}^{t-1}),$$

where $\sigma_0 = 1$ is relating it to a n Laplacian[7].

A correct estimation of the regularization weight λ is key in obtaining a well reconstructed image. Essentially. it works as a balance between the residual based object model and the gradient based prior model. When λ is close to 0 the likelihood term will dominate the reconstruction and the solution will be ill-posed. If λ is large the prior term will dominate, and provide an over smoothed and blurred image. Instead of relying on methods such as the "Grad student descent"[8] the λ^t should be estimated at each iteration through a method analogue to two-fold cross-validation[9].

The authors of the IRWSR propose to split the data of the low-resolution frames and the temporary high-resolution image in two sets $\boldsymbol{I}_\delta$ and $\bar{\boldsymbol{I}}_\delta$, where the first is used to estimate $x(\lambda)$ as proposed earlier. The estimated x is then validated through a score given by the likelihood function in Equation 3.6. To ensure that λ^t avoids local minima, the scheme is set in an adaptive grid search. In the range from $[\log(\lambda_l), \log(\lambda_u)]$, split into steps of T_c, where the values λ_l and λ_u is the lower and upper initial regularization weights. For every iteration a new range is set with $\lambda_{u,l} = \log(\lambda^{t-1}) \pm 1/t$ and the amount of steps divided by 2. In our implementation starting values $\lambda_l = -5$, $\lambda_u = 0$ and $T_{cv} = 10$ was sufficient, as λ seemed to always land is this range. The choice of cross-validation greatly enhances the runtime of the overall scheme, as the minimization of the energy function is done multiple times. For a well known system one could propose the use of a constant λ, as a way to achieve a lower runtime at the cost of lower precision.

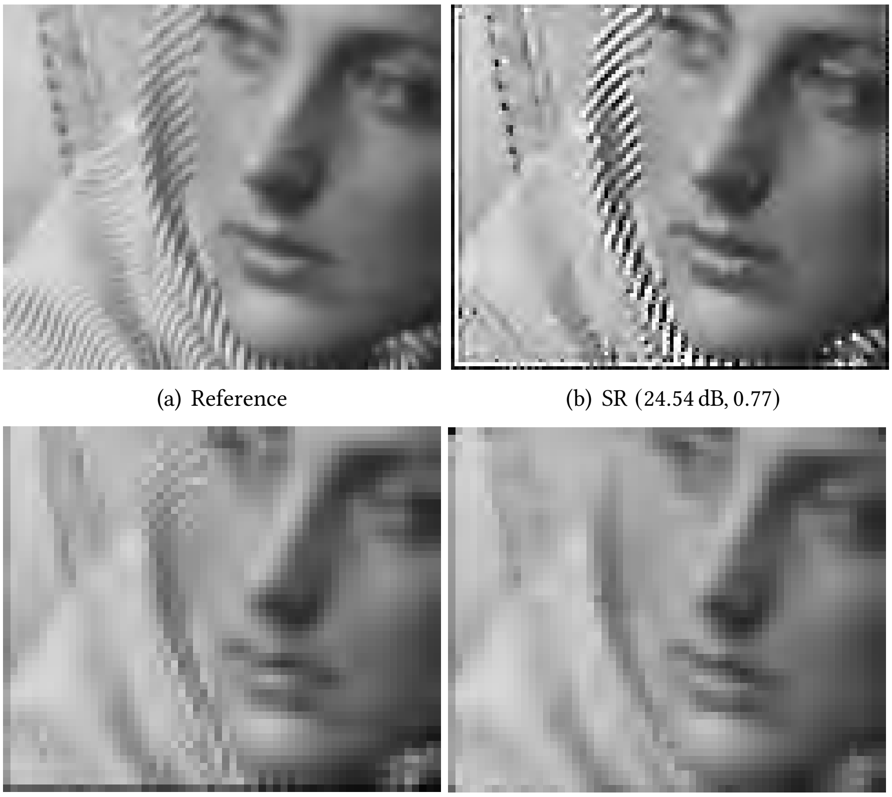
3.4.4 Intermediate results

With the above outline of the SR scheme the degraded images sequence in Fig. 3.7 is upscaled in Fig. 3.11. One should notice how some of the high frequency information stands sharper than in the LR counterpart, and the images is perceptually more pleasing. The PSNR is best for the median, which is due to the reduced noise and that the intensity levels are more true to the original. The values of the SSIM is highest for the SR image, indicating that the enhanced features are close to the original.

Further results for the implementation of the IRWSR scheme is shown in § 6, where it is applied to X-ray images. In connection with that, a discussion of the possibilities of the IRWSR scheme is given. In the coming chapters, an introduction to X-ray physics and our simulator, defines the foundation for the data we will use the IRWSR on.

[8] A common phrasing in the machine learning community for manual tunning of the parameter space.

[9] A method where the data is split into training and test sets. The parameters for the given model is then estimated on the training sets and validated on the test sets. This prevents the model for over fitting.



(a) Reference

(b) SR (24.54 dB, 0.77)

(c) Degraded LR frame (11.37 dB, 0.73)

(d) Median (32.19 dB, 0.65)

Figure 3.11: MFSR image of Barbara. Magnified by a factor of $s = 2$ in steps of $\Delta s = 0.1$, from a size of 60×50 to 120×100. The LR frames were created by random subpixel translations in the range of 5 pixels, blurred by a Gaussian blur of $\sigma = 0.5$ and downscaled to half the original size by nearest-neighbour interpolation. The regularization parameter $\lambda = 0.0008$ was found through cross-validation and the total runtime was 212 s. The median and one of the degraded LR frames upscaled by bicubic interpolation to that of the original image. The first measure is the PSNR and the second measure is the SSIM.

Part III

X-RAY SIMULATOR

4 X-RAY IMAGING

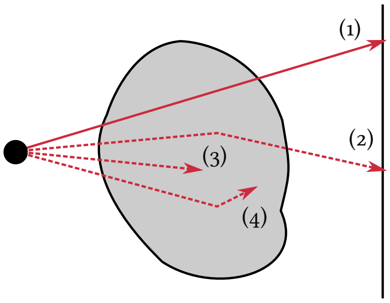


Figure 4.1: X-rays interacting with matter and hitting a detector. (1) direct transmission of a photon to the detector. (2) scattered photon adding noise to the image. (3) and (4) absorption of photons in matter.

For electromagnetic radiation, the defining quantum is the photon. A massless particle, holding an energy of $E = h\nu = \frac{hc}{\lambda}$, where h is Planck's constant, ν is the frequency, c is the speed of light and λ is the wavelength. X-rays are essentially photons with wavelengths shorter than 10^{-9} m corresponding to energies greater than 1 eV and frequencies larger than 10^{8} Hz. The photon interacts with matter either by absorption or ionization, with the probability of absorption increasing with the atomic number Z. X-ray allows for a non-destructive examination of the internal structure of objects, such as organic materials, as the penetration rate will be larger in soft tissue than that of metals or bones. A typical X-ray imaging setup consists of the following; an X-ray source, a target object and a detector [25]. This type of X-ray imaging is widely used in the fields such as medical imaging for diagnostic purposes[9], security for luggage examination and in our case food inspection for quality assessments.

In the understanding of the X-ray imaging concept for food inspection, we need to form a basis in the different physical theories relevant for the image formation. In § 2, we described the origin of optical imaging, and on a detector level, there are many parallels to be drawn. In the following, we seek to explain the different physical processes relevant to make this connection, providing us with a two-dimensional image representation of the interior of a target object.

The field of X-ray physics is vast, and the full derivations of the different physical processes are quite complicated - whole books are dedicated to all the theory behind this chapter. This section goes through the basic theory behind X-ray imaging, dividing it into its main components. In § 4.1, the X-ray source is elaborated with the X-ray emission spectrum. In § 4.2, different types of interactions with matter is elaborated, forming the attenuation spectrum. In § 4.3, the detection of X-rays is described together with an analysis of the noise. Finally, in § 4.4, the theory is put in the context of a real-world setup to provide an understanding of the possibilities and limitations of X-rays in food inspection.

4.1 X-RAY EMISSION SPECTRUM

Artificially X-rays can be produced by particle accelerators, where charged particles are accelerated to considerable kinetic energies and thereby emit radiation. Particle accelerators can be divided into two main categories;

Electrostatic and Cyclic. In an electrostatic accelerator, a static electric field accelerates the charged particles, limiting the maximum kinetic energy to the voltage difference. The cyclic accelerator, on the other hand, carries a non-static electric field, allowing the kinetic energy of the particles to exceed the voltage difference of the system. Possible by the fact that the field is non-conservative and the particle passes it in a cyclic system. A cyclic accelerator typically allows for a narrow energy spectrum, but at a higher financial cost and use of space. In this work, the focus is on the electrostatic X-ray tube typically used in medical imaging and food inspection[26].

X-ray tube

The X-ray tube, as illustrated in Fig. 4.2 is a vacuum tube consisting of a negatively charged cathode made of tungsten wire. The wire is heated to approximately 2200 C° providing the electrons with enough energy to escape the surface. The cathode is surrounded by a negatively charged focusing cup, which focuses the beam unto an anode. The anode is a large metal target(typically made of tungsten) applied with a positive voltage accelerating the electrons. The potential difference between the anode and the cathode is between 25 kV and 140 kV and is known as the accelerating voltage. The accelerated electrons interact with the atoms in the target, and most of these interactions are collisions, where the kinetic energy of the electrons are converted to heat. The X-ray radiation is a result of two processes; Bremsstrahlung and X-ray fluorescence[27].

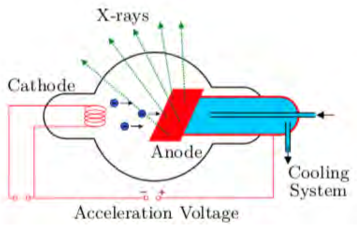


Figure 4.2: Schematic drawing of an X-ray tube. Image from [25]

4.1.1 *Bremsstrahlung*

When a negatively charged electron is within the proximity of the nucleus of an absorber, it will interact with the positive electric field through Coulomb interactions. The interactions result in a loss in kinetic energy for the electron, therefore, decelerating it. It is quantified by combining Newtons and Coulombs laws.

$$m_e a = \frac{eZe}{4\pi\epsilon_0 r^2}$$

giving,

$$a \propto \frac{Ze^2}{m_e r^2},$$

where m_e is the mass, e the charge and a acceleration of an electron, Z is the atomic number of the absorber, r is the distance between the particle and the nuclei and ϵ_0 is the vacuum permittivity. This shows that the deceleration is proportional to the size of the absorber and decreases as the inverse square with the distance. An accelerated or decelerated charged particle, emits parts of its kinetic energy as photons. An effect is known as bremsstrahlung radiation and is described by the Larmor relationship

$$P = \frac{1}{6\pi\epsilon_0}\frac{e^2a^2}{c^3},$$

where c is the speed of light. The Larmor relationship shows the power of radiation is proportional to the square of both the charge and the acceleration, indicating that the energy of the emitted X-rays increases with atom size and decreases with the distance to the nuclei. The range of bremsstrahlung photon energies goes from zero to the kinetic energy of the electron. The probability of production decreases almost linearly with photon energy $h\nu$. At low photon energies, there is a greater absorption of photons in the housing of the X-ray tube, making up the broad part of the spectrum shown in Fig. 4.3 [26][28].

Radiation Fluorescence

When an atom interacts with a photon or a particle, there is a probability of creating a vacancy in one of its atomic shells, creating an excited state. The excited state decays when an electron from an outer shell, transits into the vacancy, emitting a photon with an energy $h\nu$ corresponding to the difference in binding energy between the shells. The number of possible photons, a given atom can emit, is based on its atomic number Z and the quantum numbers of the shells involved in the transition. The radiated photons are characteristic for a given element, and the energies are known as the line spectrum. X-rays are mainly generated in inner shell transitions of elements with large Z. As an example, a K-shell(the innermost shell) electron of tungsten carries a binding energy of about 70 keV. A transition from the L-shell to the K-shell(A transition between the two inner shells) generates photons with energies 57.98 and 59.32 keV. In Fig. 4.3 the characteristic X-rays can be seen as sharp peaks in the spectrum[26].

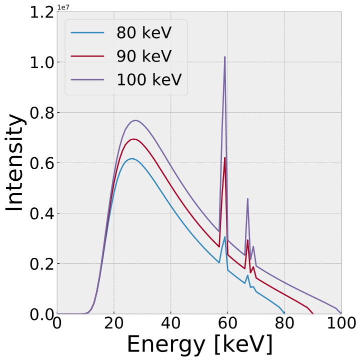


Figure 4.3: Simulated data showing the distribution of photon energies emitted by a tungsten X-ray. Voltages are set to 80, 90 and 100 KeV. Data generated with SpekCalc [29]

With a baseline in the processes behind the generation of X-rays, we can proceed with the emission spectrum.

4.1.2 X-ray Emission Spectrum

A type of X-ray tube used in food inspection is based on the element tungsten, having a high atomic number of $Z = 73$, melting point at 3370° C, high thermal conductivity and low vapor pressure. This makes it ideal for the high temperatures in the tube and allows a high vacuum to be obtained. In Fig.

4.3, the energy spectrum of a tungsten X-ray tube is simulated at different voltages. The spectrum is polychromatic and broad, which complicates the analysis of target objects, as we will discuss in the following section. By the use of filters absorbing undesired energy levels, one could be tempted to try and narrow the bandwidth of the source. However, this is often not feasible in a real setup as less than 1% of the electron energy is converted to X-rays, the rest is converted to heat or UV rays[27].

When the photons are emitted from the source, they will in an imaging system proceed to interact with the objects, of which interior we seek to examine. This is done through photon-matter interactions.

4.2 PHOTON-MATTER INTERACTIONS

Nuclear physics describe several processes of photon-matter interactions, which generally are dependent on the incident photon energy $h\nu$ and the atomic number Z of the absorber. In X-ray imaging, the most critical aspect of these interactions is how the matter attenuates and scatters the incoming X-ray photons.[1] The most critical parameter here is the linear attenuation coefficient μ, which describes the probability, per unit path length, that an incident photon will interact with the matter. For a narrow collimated beam of mono-energetic photons the exponential attenuation is described by Beer-Lamberts law:

$$I(x) = I_0 e^{-\int \mu(z) dz} \tag{4.1}$$

where I is the intensity of the emerging beam, I_0 is the original intensity of the beam and z notes the penetration distance. Accounting for the object density ρ the mass attenuation coefficients can be defined $\mu_m = \frac{\mu}{\rho}$. The values of μ_m are based on the theoretical values of the entire atomic cross section σ_{tot}, which describes the probability of a specific interaction taking place. The relation is given as:

$$\mu_m = \frac{\mu}{\rho} = \frac{\sigma_{tot}}{uA}, \tag{4.2}$$

where u is the atomic mass unit and A is the relative atomic mass of the absorber. The total atomic cross section is dependent on the photon energy and the atomic number Z and is governed by the following contributions:

$$\sigma_{tot} = \sigma_{pe} + \sigma_r + \sigma_C + \sigma_{pair},$$

[1] In medical physics, radiation dosimetry is another essential aspect of the interactions, which is the energy transfer from photons to charged particles and the absorptions of the energy in the irradiated tissues.[26]

where σ_{pe}, σ_r, σ_C and σ_{pair} are the atomic cross section for the photoelectric effect, Rayleigh scattering, Compton scattering and Pair production, respectively[30].

In the coming, we will go through the different photon-matter interactions one by one, and then combine them in the total mass attenuation spectrum.

Photoelectric Effect

The photoelectric effect is the interaction between a photon of energy $h\nu$ and a tightly bound electron in the orbital of the absorber atom[2], where the incident photon will be fully absorbed and the electron known as the photoelectron, ejected from the orbital. The process is illustrated in Fig. 4.4. The photoelectron carries a kinetic energy of $E_k = h\nu - E_B$, where E_B is the binding energy of the electron in the given orbital shell. Due to the emission of the photoelectron, there will be a vacancy in the given shell. An electron will fill this vacancy from a higher shell, and the transmitting energy will be emitted as a characteristic photon. The attenuation spectrum takes a sawtooth shape at the absorption edges due to the binding energy of the different orbital shells of the element[26]. Here the K-edge exists for all atoms but lies below the energy levels emitted from a common source for low Z materials. For high Z values, the L and M absorption edges become apparent. In general, contribution to the attenuation from the photoelectric absorption is in the range below 100 KeV and grows approximately as $\propto Z^4$ [11].

[2] A tightly bound electron has a binding energy comparable to the photon energy $h\nu$. Interactions only occur when the binding energy is slightly smaller than the photon energy.

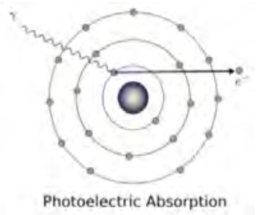


Figure 4.4: Image from [25]

Rayleigh Scattering

Rayleigh scattering occurs when an incident photon of energy $h\nu$ interacts with a bound electron of the absorber with a binding energy greater than the photons. This energy difference prevents the photon of ionizing the absorber, leaving a negligible change in energy and only changes the angle of the exciting photon. The atomic cross section for Rayleigh Scattering σ_R is largest at low energies, but negligible compared to that of the photoelectric effect. In all energy ranges σ_R is secondary to the other main contributions of the cross section.

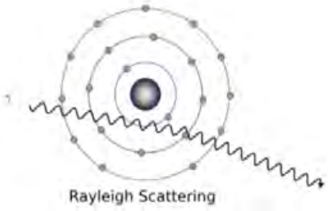


Figure 4.5: Image from [25]

The Compton Effect

The Compton Effect describes the interaction between a photon of energy $h\nu$ and a loosely bound electron in the outer shell of the absorber. A loosely bound electron has a binding energy E_B smaller than the photon energy $h\nu$ that is $E_B \ll h\nu$. In the process, a photon known as the scattering photon is produced with an energy $h\nu'$ smaller than the incident photon, which is annihilated. The electron is ejected from the atom with a kinetic energy E_k. As a consequence of energy and momentum conversation, the scattering photon changes angle with respect to the incident photon. The scattered photons proceed to hit different pixels in the detector, adding noise to the final images and reduces the overall contrast.

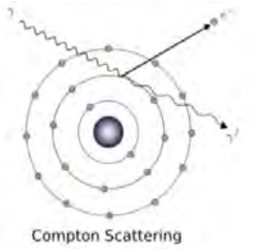


Figure 4.6: Image from [25]

Pair Production

When a high energy photon $(h\nu) > 2m_ec^2$ interacts with the Coulomb field of an absorber, it can lead to the annihilation of the incident photon and the creation of an electron and positron pair, which is known as the pair production. In this process, the total momentum cannot be transferred to the electron-positron pair, and the leftover momentum is transferred to the absorber. The atomic cross section for the pair production has a lower limit given by the threshold energy needed for producing the pair $2m_ec^2 = 1.022\,\text{MeV}$. Above the threshold the atomic cross section is proportional to Z^2.inspection.[26].

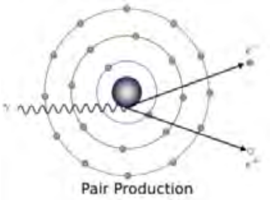


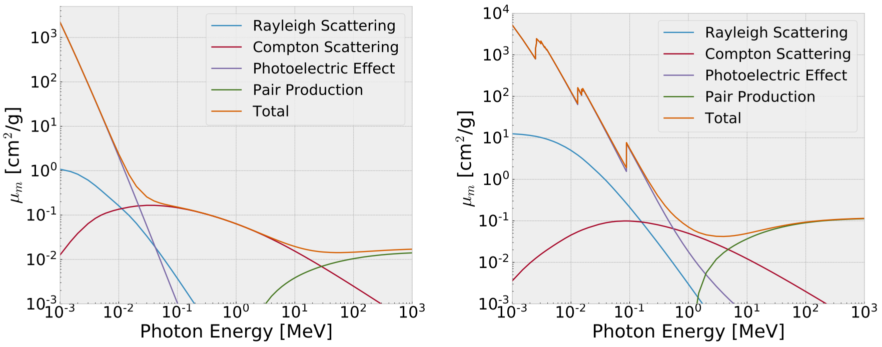
Figure 4.7: Image from [25]

The different processes can then be combined in the total mass attenuation coefficients.

4.2.1 Total mass attenuation coefficient

As shown in Equation (4.2), the total mass attenuation coefficient is proportional to the total atomic cross section. In Fig. 4.8 μ is plotted as a function of photon energy for carbon and lead. At low photon energies $h\nu < 100k$ eV and high atomic number Z the photoelectric effect has the biggest contribution. The contribution from the pair production dominates in the region $h\nu > 10M$ eV and for high atomic numbers Z, which makes it unimportant in food inspection, where the typical energy range is $h\nu < 100$. The Compton Scattering contribution is most apparent in the intermediate region and dominates wider at low atomic numbers Z. The Rayleigh scattering contribution is secondary to the other interactions and largest in the lower photon energy region[26].

The total mass attenuation describes how the X-ray beam attenuates when passing through the object of interest. The object interior can then be imaged, through the means of a detector in combination with a scintillator.



(a) Carbon $Z = 6$

(b) Lead $Z = 82$

Figure 4.8: The total mass attenuation coefficients μ_m as a function of photon energy. The mass attenuation coefficients related to the various interactions described in the main text, together with a total. The data is shown for (a) carbon and (b) lead. Data from [31]

4.3 DETECTOR AND NOISE

From above, we know that when matter is exposed to X-rays, the photons will either penetrate the target object or be attenuated. This process leaves an X-ray beam, where the reduced intensity is related to the interior structure of the target object. Detection of the attenuated X-rays on a two-dimensional grid can then provide what we know as an X-ray image.

4.3.1 Scintillator and detetor

As the high energy X-ray photons have a low probability of interaction directly with detectors such as charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) chips[3]. The indirect detection method of converting the X-ray photons to optical light by placing a scintillator between object and sensor can be utilized. The scintillator absorbs the incident X-rays, exciting the electron states, which relaxes by emitting optical light. When a photon interacts with the interior of the scintillator, it emits photon isotropically, sending half of the photons in the detector direction. The spreading of the light induces blur in the final image, which increases with the thickness of the scintillator. The overall efficiency of the scintillator can be described by the variable η_{scin} i. e., the percentage of X-rays converted to optical photons registered by the corresponding pixel[11]. The scintillator is popular in an industry setting as it is a cheap, non-toxic material with a high conversion rate from X-rays to light.

The optical light from the scintillator can then be registered in a detector.

[3] In this thesis, we only consider digital cameras in combination with scintillators as a detector. Other systems exist like phosphor storage system, digital flat-panel detectors and film-based methods

In an X-ray setting, the final image quality is dependent on the stochastic processes inherent in all of the above.

4.3.2 Noise as a statistical distribution

In an X-ray imaging setup there is an accumulation of the following different stochastic processes.

- The sources emitting photons as a Poisson process.
- The number of photons passing through the object is a Binomial process.
- The number of X-rays converted to light photons in the scintillator is a Binomial process
- The number of light photons captured by the detector is a Binomial process

The noise in the system can be analyzed by going through the processes systematically. As seen in the following it leads to the thinning of a Poisson process by a Binomial process.
We start by considering a single ray path between the source and the detector, where the X-ray penetrates a homogeneous object. In a given exposure interval there will be emitted $\mathcal{N}$ X-ray photons of a given energy, following a Poisson distribution with a mean of $E[N] = N_0$. N_0 is dependent of source energy and exposure time. The distribution is then

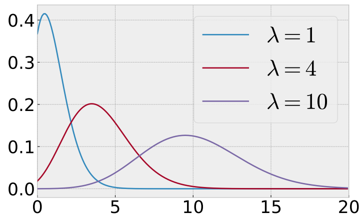


Figure 4.9: Example of the Poisson distribution for different Poisson variables λ.

$$\text{Poisson}(N_0) = P\{\mathcal{N} = n\} = \frac{N_0^n}{n!} e^{-N_0},$$

assuming $\mathcal{M}$ photons will pass unaffected through the object. This process can be viewed as a Bernoulli trial, where the success probability p is the probability of leaving the target object unaffected.

$$P\{\mathcal{M} = m | \mathcal{N} = n\} = \binom{n}{m} p^m (1-p)^{n-m} = \frac{n!}{m!(n-m)!} p^n (1-p)^{n-m},$$

where $p = e^{-\int \mu(z)dz}$ is from the Beer-Lambert law given in Equation (4.1). Using the total probability theorem[4] $\mathcal{M}$ can be determined

$$\begin{aligned} P\{\mathcal{M} = m\} &= \sum_{n=m}^{\infty} P\{\mathcal{M} = m|\mathcal{N} = n\}P\{\mathcal{N} = n\} \\ &= \sum_{n=m}^{\infty} \binom{n}{m} p^m (1-p)^{n-m} \frac{1}{n!} e^{-N_0} (N_0)^n \\ &= \frac{(N_0 p)^m}{m!} e^{N_0 p}. \end{aligned}$$

As this yields a new Poisson distribution[5], the result for the expected value[6]. of photons passing through the object is the Beer-Lambert law

$$E[\mathcal{M}] = N_0 p = N_0 e^{-\int \mu(z)dz}.$$

As all other following processes are viewed as Binomial, the resulting photon count distribution can be estimated as a Poisson thinned by several Binomial distributions. Giving the quantum efficiency of the detector η i. e., the percentage of incoming photons contributing to the final output of the detector, the resulting distribution is then

$$P\{\mathcal{Y} = k\} = \frac{1}{n!} e^{-N_0 \eta p} (N_0 \eta p)^k,$$

where $\mathcal{Y}$ is the amount of photons represented in the final image. The efficiency of the scintillator η_{scin} is included in η[11].

4.3.3 *Signal-to-Noise Ratio*

For measuring the quality of the final image, the signal-to-noise ratio (SNR) is used. Contrary to the prior definition of the point signal-to-noise ratio (PSNR), we use a statistically based definition, defined as the mean over the standard deviation

$$SNR(\mathcal{Y}) = \frac{\langle k \rangle}{\sigma} = \frac{E(\mathcal{Y})}{\sqrt{E((\mathcal{Y} - \langle k \rangle)^2)}}.$$

We can then use the general result, that the expected value and variance of a Poisson distribution is the Poisson number $E[\mathcal{Y}] = \sigma^2 = N_0 p \eta$.

$$\begin{aligned} SNR(\mathcal{N}) &= \frac{N_0 p \eta}{\sqrt{N_0 p \eta}} = \sqrt{N_0 p \eta} \\ &= \sqrt{N_0 \eta} e^{-\frac{1}{2} \int \mu(z)dz}. \end{aligned}$$

[4] Given as $P(A) = \sum_n P(A|B_n)P(B_n)$, where n denotes the independent events.

[5] A full proof can be found in [32].

[6] The expected value and variance of a Poisson-distribution are both equal to the Poisson number

The above shows that the relative noise can be reduced by a higher input of photons. A higher SNR provides an increased image contrast as the noise has a relatively smaller influence on the image. In an X-ray generator, the amount of photons emitted in the system is proportional to the current i. e., the signal. From the above expression, it can be stated that by doubling the signal, the $SNR(\mathcal{Y})$ is only increased by a factor of $\sqrt{2}$,[9].

4.4 APPLIED X-RAY IMAGING

Following the previous introduction to the physics behind X-ray imaging, we proceed by explaining some of the technical details relevant for taking X-ray images. In Fig. 4.4, a test setup for examining X-ray scanning procedures is shown. The machine is located at NewTec, and the specifications are given in Table 4.1.

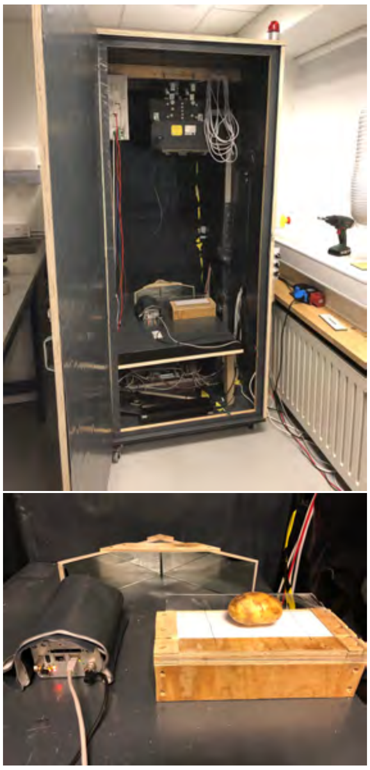


Figure 4.10: X-ray imaging system located at NewTec (top) full view (bottom) close-up. The setup consisting of an X-ray generator, object of interest(potato) lying on a scintillator sheet and a camera. Image from the master thesis by Aleksandar Topic

The X-ray generator is mounted at the top of a lead shielded cage protecting the users of unwanted radiation. The X-ray is sent in the direction of an object(in this case a potato) placed on a sheet of the scintillator. The optical photons from the scintillator are reflected on a mirror towards a CMOS detector.

When working with this X-ray source, two parameters are controllable: The accelerating voltage controlling the upper energy limit of the X-ray spectrum and the current controlling the intensity(photon count) changing the SNR. An optimal setting for these parameters varies with the object and should be set to maximize the contrast of the internal structure i. e., maximizing the intensity variance providing higher structural information.

NewTec Scanner parameters	
X-ray generator	Spellman XRBHR100
Voltage range	(35-105) kV
Current range (35-70) kV	(0.35 -7.5) mA
Current range (71-105) kV	(0.35 - 5) mA
Maximum power	525W
Camera	CMV8000
Resolution	8.4 MP
Pixel Size	5.5 μm
Frame Rate	103 fps
Conveyor-belt speed	15.5 cm/s

Table 4.1: The specifications of the NewTec scanner.

In the setup Fig. A.1, an area detector(CMOS camera) is used for taking high resolution (HR) images of still objects, allowing an optimal setting for investigating the possibilities of detection of various anomalies e. g. what are the limits for finding holes in various pieces of food, or how small foreign objects can we detect.

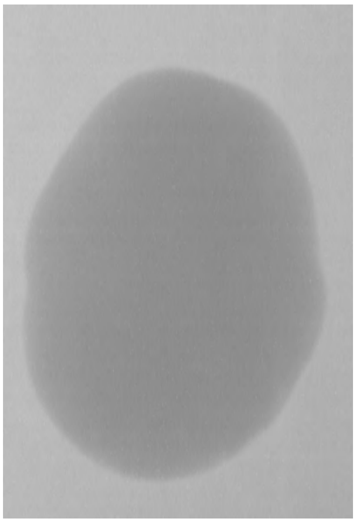


Figure 4.11: X-ray image of a potato taken by the NewTec scanner.

However, this scenario of still images is not realistic in the context of a food inspection system. In real-time sorters, high throughput of objects is desired, which, as a consequence, demands the objects to be in constant motion. In the NewTec scanner, a conveyor belt can be placed above the scintillator, allowing the test of images under movement. In combination with an area scanner, there is a fine balance between the exposure time and the induced motion blur. A low exposure time gives a higher frame rate, which means less motion per frame. A high exposure time gives a low frame rate, which increases the motion blur.

An alternative to this, and more common in online systems, is the line scanner. A line scanner captures single rows of pixels and interpolates them to form the full image. The line rate (analogous to the frame rate), is synchronized with the speed of the conveyor belt, providing HR images without motion blur. In this work, we have not considered the influence of line-scanners, as we had no easy access to raw data.

The focus on resolution enhancement in this work is based on the Area Detector, as a possible way to counter the motion blur induced. An implemented super resolution (SR) algorithm would provide an alternative to the line detectors, allowing real-time output frames containing whole objects, giving new possibilities in regards to the segmentation and classification algorithms developed for the AXIS project.

In § 6.3, we examine the effect of the iterative re-weighted super resolution (IRWSR) algorithm on different image sequences from the NewTec scanner. Before this, a demonstration of a simulated alternative to setups like the NewTec scanner is given.

X-RAY SIMULATOR 5

The X-ray simulator **Xsim** developed in this thesis is a computational tool for generating X-ray planar radiography images. The target objects are virtual and in movement. Xsim consists of the same essential elements as a traditional X-ray imaging system; A source, an object and a camera, as shown in Fig. 5.1. The different components are related through a geometrical analysis of the individual ray paths emitted from the source and hitting the camera pixels. The final image is generated by using the physical properties assigned to the virtual object in combination with Beer-Lambert Law. Xsim provides an easy and efficient way of testing different combinations of materials and setup configurations. This allows analysis of both realistic and unrealistic setups, given the assumptions made in constructing the system. The code for **Xsim** is publicly available at [33], together with examples generating some of the images showcased in this work.

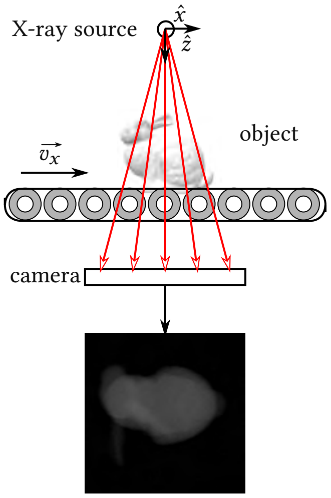


Figure 5.1: X-ray image of a virtual homogeneous bunny in motion.

In § 5.1, the geometric description of the rays between the different system components is described, providing a framework for the simulator. Afterwards, the different components are related to the previous X-ray physics section in § 5.2. Different approaches to optimization are discussed in § 5.3. The § 5.4 system is showcased with different examples of generated images and descriptions of the user-defined variables. At last in § 5.5.1 Xsim is put in the context of related work with a discussion of its limitations.

In the following a right-handed coordinate system is used as a reference system, see Fig. 5.2. In general the analysis is done in the $\hat{x}$ direction, but is completely analogues in the $\hat{y}$ direction as the the system is symmetric to the x and y-axis. The notation is as:

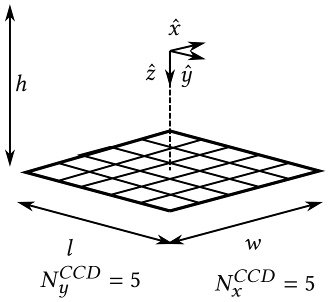


Figure 5.2: The reference system with respect to the camera object.

- The xy plane is parallel with the camera object.
- The $\hat{z}$ is going from the source to the center of the camera plane.
- The notation N_x^{CCD} and N_y^{CCD} refers to the grid size of the charge-coupled device (CCD) object in the x and y direction respectively.
- The n'th element refers to the different elements in the defined camera object and the corresponding ray path.
- The width w^{CCD} and l^{CCD} are measures related to the physical size of the CCD object and is aligned with aligned with $\hat{x}$ and $\hat{y}$ respectively.
- The height h is the physical height of the system aligned with $\hat{z}$

- The physical sizes of the components is used in the geometrical considerations.

- The grid size of the different components is used in describing how quantized the system is. A finer grid gives less pixelated images.

5.1 GEOMETRICS

The framework for Xsim is the geometrical considerations of the ray paths between a point source and a camera. Each ray path is connected to a pixel in the camera object, and the attenuation for the path through the object is used in generating the final image.

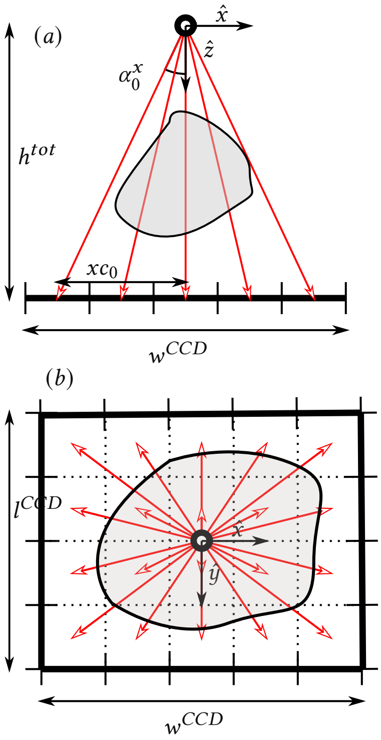


Figure 5.3: X-rays emitted from a point source through an object towards a camera. In (a) side view and in (b) top view.

In Fig. 5.3, a simple side and top view illustration of the system is given. The X-ray source is defined as an isotropic point source at a height h^{tot} above a camera object. The camera object denoted CCD is a 2-dimensional grid of size $N_x^{CCD} \times N_y^{CCD}$ having a physical size of width w^{CCD} and length l^{CCD}. The angle in the x and y direction for each ray hitting the n'th pixel is given by

$$\alpha_n^x = \arctan(\frac{xc_n}{h^{tot}}), \quad \alpha_n^y = \arctan(\frac{yc_n}{h^{tot}}), \tag{5.1}$$

where xc_n and yc_n is the distance in the x and y direction to the center of the n'th CCD element. The direction of the n'th ray going from the source to the CCD, is then fully described by the angles α_n^x and α_n^y.

5.1.1 *Field of View*

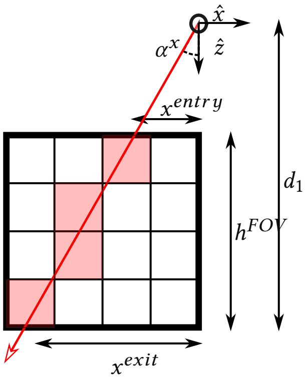


Figure 5.4: Example of an approximated line in a 2D FOV

The next step is to relate the rays to an object, done through a reference field of view (FOV), where the objects can be projected onto. The objects are based on volume elements and as such the FOV is quantized to a 3-dimensional grid. A side view of the system with the FOV is given in Fig. 5.4. The FOV is a volume describing the connection between the emitted rays and the object of interest. It is defined as having a physical size $h^{FOV} \times l^{FOV} \times w^{FOV}$ containing volume elements of size $v_x \times v_y \times v_z$. Furthermore, the distance d_1 between the point source and the bottom of the FOV is user-defined. The combination of the given information is used in calculating the entry and output coordinates, for each of the paths.

```python
def FOV(grid_D, N_ccd, voxelSizeD, d1, FOV_size, alpha):
  # Finds the x and y entry/exit coordinates
  entry_x, exit_x, entry_y, exit_y =
    findIO(voxelSizeD, d1, FOV_size, alpha)

  for n in range(N_ccd):
    ray_index_list[i,:,:] = Bresenham3D(
      entry_x, entry_y,
      exit_x, exit_y, grid_D[0])

  return ray_index_list
```

Code Listing 5.1: Use of Bresenham algorithm through the FOV.

$$
\begin{aligned}
x_n^{\text{entry}} &= (d_1 - h^{FOV}) \cdot \tan(\alpha_n^x), \\
x_n^{\text{exit}} &= d_1 \cdot \tan(\alpha_n^x).
\end{aligned}
$$

Combining this with the expressions given in Equation (5.1)

$$
\begin{aligned}
x_{\text{entry}} &= (d_1 - h^{FOV}) \cdot \frac{xc_n}{h}, \\
x_{\text{exit}} &= d_1 \cdot \frac{xc_n}{h}.
\end{aligned}
$$

Since a straight line is ambiguous to define in a quantized 3D-grid, Bresenham's line algorithm is used in estimating a line from the entry to the exit point[1]. The algorithm returns the indices for each volume element that the ray passes and is implemented to iterate over every index n. The indices are then stored in a list for each of the N^{CCD} paths. A simplified python snippet is given in Code Listing 5.1.

An important property for the later calculations is the path traveled within each volume element for the nth ray. This distance is approximated to be the same for each layer in the FOV and is given by the euclidean norm

$$
d_n = \sqrt{(\tan(\alpha_n^x) \cdot v_x)^2 + (\tan(\alpha_n^y) \cdot v_y)^2 + v_z^2}, \tag{5.2}
$$

5.1.2 *Projection of the Object*

Having the indices relating the volume elements in the FOV with the traces between the point source and the CCD array. The object[2] of interest is

[1] Bresenham's line algorithm finds a close approximation to a straight line, between two points in a N-dimensional grid. The algorithm is explained in § B.2

[2] The object is described further in § 5.2.2

projected onto the FOV. The object in the framework called the density tensor D and has a size of $N_x^D \times N_y^D \times N_z^D$. An example for the geometrical considerations in the x direction is shown in Fig. 5.5. Here the centre of the two systems is shown as a thick line and the relative translation defined between the centres as $x_{translation}$. For optimization, only the relevant part of the object is projected i. e., the part which will contribute to the final image. The width of the relevant part of **D** is called the broadcasting width B_x and is defined as the overlapping parts of the D and the FOV. The reasoning for this division between an object grid D and the FOV, is that the whole system with the FOV can be preprocessed once. The different objects of interest can then be projected onto the system multiple times under different translations. A simplified version of the python code used for projection is showed in Code Listing 5.2 and the method is further elaborated in § B.4.

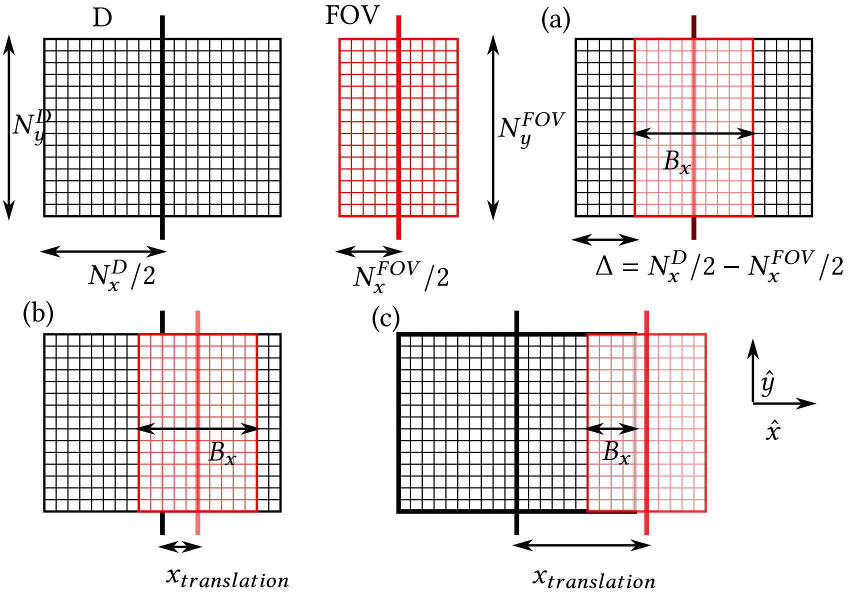


Figure 5.5: Top view showing the projections of the object(Density tensor D) onto the FOV for different translations. The B_x denotes the overlapping parts of the object and the FOV. In (a) the centres of the two are aligned. In (b) there is a relative movement between the two, still with the whole of the FOV being within the scope of D. In (c) there is a relative translation between the two making the area of interest smaller than the area of the FOV.

```python
def projectDtoFOV(grid_FOV, grid_D, D, move):
  # Finds the smallest/largest axis in each direction
  min_axis = np.min([grid_FOV, grid_D], axis=0)
  max_axis = np.max([grid_FOV, grid_D], axis=0)

  # length and width of broadcasting b_l
  # Set to be either the minimum axis size
  # or as the difference between the minimum
  # and maximum axis
  b_l = np.min([min_axis[1:], min_axis[1:]//2 -
    (np.abs(move)-max_axis[1:]//2)], axis=0)

  # Finds the difference in grid size between
  # object and FOV
  delta_fov = np.subtract(grid_FOV, grid_D)//2

  # Use the defined constants to find the index
  # range in x and y
  ind_x = FindIndexX(b_l, delta_FOV, grid_FOV, grid_D)
  ind_y = FindIndexY(b_l, delta_FOV, grid_FOV, grid_D)

  # Defines an empty field of view
  ref_fov = np.zeros(grid_fov)
  # Broadcast D to FOV
  ref_fov[0:D.shape[0], ind_x[0]:ind_x[1],
    ind_y[0]:ind_y[1]]
    = D[:, ind_x[2]:ind_x[3], ind_y[2]:ind_y[3]]

  return ref_fov.astype(int)
```

Code Listing 5.2: Projecting the object D onto the FOV. The projection accounts for the scenarios of either FOV or D being largest in the x and y direction. The variables are $grid_FOV = (N_z^{FOV}, N_x^{FOV}, N_y^{FOV})$, $grid_D = (N_z^D, N_x^D, N_y^D)$ and $move = (transaltion_x, translation_y)$. The projection is explained more throughly for one direction in § B.4

5.2 SIMULATING PHYSICS

As explained in § 2, a digital image is an array whose values represent the brightness of each pixel. In Xsim this is noted as I_n, referring to the intensity of the n'th pixel. The final values for I_n are computed using Beer-Lambert law. The equation connects the input and output intensity of X-rays penetrating an object and is previously described in § 4.2 and found in Equation (4.1). In order to make the law computationally feasible a discretization is needed, the discretization is done with respect to 1) the sampled photon energies from the source explained in § 5.2.1 and 2) the grid definition of the object defined in § 5.2.2.

5.2.1 Simulating Source

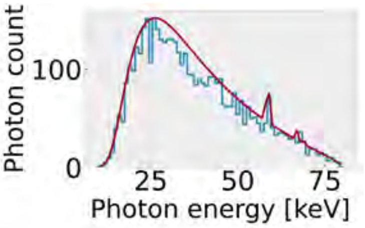


Figure 5.6: Sampled energy spectrum with a histogram showing counts. Sampled in the range $[10\,\text{keV} : 1\,\text{keV} : 80\,\text{keV}]$ for $N^{photons} = 5000$.

To simplify the system, the X-ray source is assumed to be an isotropic point source. As described in § 4.1, a common X-ray source emits photons as a spectrum of energies, dependent on target material and acceleration voltage. To make a sufficiently accurate simulation of the energy distribution, a spectrum like the one shown in Fig. 4.3 is used as a probability distribution. The distribution is sampled in a user-set energy range $[E_{min} : E_{step} : E_{max}]$ to a user-set number of photons $N^{(photons)}$. For sampling, the hit-and-miss method is used, drawing points uniformly in the desired interval. The points are counted as a hit if their positions lie under the desired curve and a miss otherwise. An example of the sampling is showed as a histogram in Fig. 5.6. The python code for making the sampling is given in Code Listing 5.3.
The number of photons in the system is an indirect measure of the intensity of the rays, as the final intensity measure in the CCD is dependent of the count of photons and not the photon energy as explained in § 2.2.3. In a real setup, the number of photons emitted would be directly related to the power of the source. This sampling scheme allows the system to be flexible enough to implement various x-ray sources by specifying the relevant spectrum.

5.2.2 Simulating Object

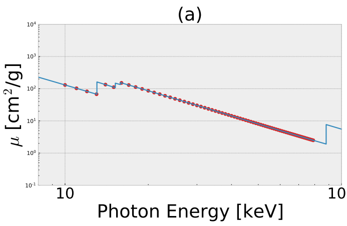
Value	Elemental/Compound
0	Undefined/Vacuum
1	Carbon
2	Aluminum
3	Iron
4	Air
5	Water
6	Soft Tissue

Table 5.1: Example of a dictionary for referencing voxel values with different elements.

In Xsim the object of interest, is defined in a three dimensional tensor object D of size $N_x^D \times N_y^D \times N_z^D$, where each index is referred to as a voxel. Each voxel carries a string of the element or compound it represents. When running the system, each string in converted to a value referring to a dictionary of all the elements. See Table 5.1 for an example of the correspondence between different voxel values and compounds. For each element/compound a corresponding array of the attenuation coefficients

μ_i as a function of photons energy denoted by i is given. The spectrum's are based on data from [31] and are as shown in Fig. 5.6. The attenuation interpolated to the energy range given in the initial definition

of the source distribution. This gives a direct relation between each voxel element in the object tensor and the attenuation. In Fig. 5.7(b) a simple 2D object is shown with values related to the dictionary in table Table 5.1. For the geometrical considerations of the system, the object is also assigned a physical size by defining height h^D, length l^D and width w^D. These physical considerations are used to automatically define the physical size of the voxel representations of the FOV. Guaranteeing that the desired size of the object holds when projected onto the FOV. The physical size of the object and FOV is used relative to the physical size of the CCD array and the distances defined from the source to the camera. All physical distances are defined in units of cm as μ : [cm$^2/g$], providing a proper attenuation in every voxel.



(*b*)

0	1	1	0
1	3	3	1
1	3	3	1
0	1	1	0

Figure 5.7: (a) Example of the values of mu_i for lead sampled in the interval [10 keV : 1 keV : 80 keV]. (b) A simple 2D object with values referring to Table 5.1. This object would be interpreted as a piece of carbon with an iron core.

```python
def produceXray(spectrum, photonsRay, Emin, Emax, Estep):
  # normalize
  spec = spec / np.sum(spec)
  # defines the bins for the histogram
  N_bins = len(np.arange(Emin, Emax, Estep))+1
  # Empty arrays to be filled with values
  xhit = np.zeros(photonsRay)
  for i in range(photonsRay):
    while True:
      # Sample an x and y value from the spectrum
      x = np.random.uniform(Emin, Emax)
      y = np.random.uniform(0, np.max(spec))
      # if the values lies within the spectrum save
      if (y < spec[x]):
        break
    xhit[i] = x
    # counts the number of photons at each energy level
    counts, bin_edges = np.histogram(xhit)
    return counts
```

Code Listing 5.3: The method draws X-ray photons related to the given energy spectrum, defined in the variable spec. The photon are drawn in the range $[E_{min} : E_{step} : E_{max}]$.

5.2.3 *Simulating Camera*

The geometrical framework, the emission of the source and attenuation of the object can now be combined. The connection is established through a discrete version of the Beer-Lambert law introduced in Equation (4.1).

$$\underbrace{I_n}_{camera} = \underbrace{\sum_{i=E_{min}}^{E_{max}} N_i^{(photons)}}_{source} \underbrace{e^{-\mu_i^n d_n}}_{object}. \tag{5.3}$$

$$\mu_i^n = \underbrace{\sum_{k}^{N_z^{FOV}} -\mu_i^{n\{k\}}}_{object},$$

where k refers to the layers of the FOV in the z-direction, i refers to the different energy levels of the photons and n refers to the different pixels/ray paths. The $N_i^{photons}$ is the amount of photons at the i'th energy level, μ_i^n is the total attenuation for the $n'th$ path at the $i'th$ energy level and d_n is the path length of the ray through the FOV. At last the $\mu_i^{n\{k\}}$ is the attenuation coefficient related to the k'th layer in the path of the n'th ray.

With Equation (5.3), each intensity value of a pixel, is the sum of the attenuation of all the photons emitted from the source. It is assumed that the point source is an isotropic emitter i. e., photons are emitted in equal amounts uniformly through all angles. The algorithm has to iterate through the sum for each pixel for both the energies and all the layers in the FOV. The most cumbersome calculation is the μ_i^n, scaling with the height of the FOV grid[3]. To simplify the sum over the layers in the FOV, the path lengths $d_n^{(k)}$ is set to d_n. That is the path length through each layer is approximated to be the same for every voxel for the nth ray as given in Equation (5.2). The final run time for producing a single frame is then in the order of $O(N_x^{CCD} N_y^{CCD} N_z^{FOV} E^{(steps)})$.

For an intuitive understanding of the algorithm. We imagine the object defined in Fig. 5.7(b) projected onto the FOV showed in Fig. 5.4. For an independent ray, there will, for each layer in the FOV, be a reference to the dictionary of elements showed in Table 5.1, which in the first layer would be carbon. The absorption coefficients of the carbon spectrum will then be interpolated to the energies of interest as seen in Fig. 5.6 and be placed in an array. For the next layer, the corresponding values of iron will then be added. The above procedure is then repeated for every single ray defined in the system.

As stated in § 4.3.2 the final intensity values of the system should be pulled

[3] The calculation is scaling with the height as the driving axis of Bresenham's line algorithm is defined in the z-direction, see § B.2.

from a Poisson distribution[4].:

$$\underbrace{I_n}_{camera} = \text{Possion}(\underbrace{\sum_{i=E_{min}}^{E_{max}} N_i^{(photons)}}_{source} \underbrace{e^{-\mu_i^n d_n}}_{object}), \tag{5.4}$$

where the quantum efficiency is defined $0 \leq \eta \leq 1$, which is a measure of the percentage of recorded photons in the pixels. A final notion of the simulating of the camera is the variable Nsubpixel. The variable determines how many subpixels a single pixel in the CCD array consist of and is illustrated in Fig. 5.8. The photons per ray decrease with an inverse square giving a constant intensity with changes of the variable. The different photon counts per subpixel is then added before setting the saturation value. The reasoning for this implementation is the probability of missing small objects in the FOV, when doing the Bresenham line algorithm. An alternative to this, could be objects defined in a finer spatial grid.

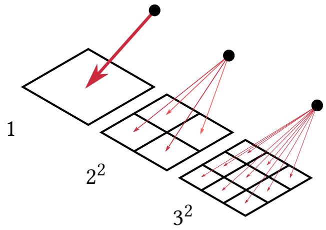


Figure 5.8: Illustration of the Nsubpixel variable related to a single pixel. The variable is set to 1, 2 and 3.

5.2.4 *Simulating movement*

With the possibility of creating single frames using Equation (5.4) the movement of a conveyor belt can be simulated. The user-defined variable $move_{step} = (movement_x, movement_y)$ decides the stepwise translation of the object between subframes. The subframes are temporary frames in the system, added together to form a single frame. The reasoning is to simulate the exposure time of a camera as discussed in § 2.2.3. With this approach the object is able to move under a user-defined number of subsamples. A low number of subframes gives a final frame carrying little motion blur and vice versa. The object is projected onto the FOV at every iteration through the method described in § 5.1.2.

A common phenomena happening in imaging, is the saturation of pixels i. e., a pixel all white as it photon threshold is exceeded. In order to make the exposure time better reflect physical real-world limitations the pixel carries the possibility of saturation. In the system this is defined as an user-defined intensity threshold, setting all values above to max intensity.

Putting all this together the main loop of Xsim is given in Code Listing 5.4.

[4] It holds that the sum of independent Poisson random variables is a Poisson random variable with a parameter being the sum of the independent parameters[34]

```
ray_index_list = FOV(grid_D, N_ccd, voxSizeD, d1,
                 FOV_size, alpha):

for i in range(N_frames):
    for j in range(N_sub):
      # pulls photons from energy distribution
      ray_energy, ray_counts = produceXray(
          spectrum, photonsRay,
          Emin, Emax, Estep)

      # Projects the object to the desired position
      # in the FOV
      ref_FOV = projectDtoFOV(grid_FOV, grid_D,
                D, move)

      for n in range(N_CCD)
        mu_d[n] = calc_mu(mu_dict, ref_FOV, v_d)

        I_photon_temp = np.sum(np.random.poisson(
            ray_counts * np.exp(-mu_d)), axis=2)
    I_photon += I_photon_temp
I_photon[I_photon] > saturation] = saturation
# normalize to make 8bit grayscale
I_ccd = int(normalize(I_photon)*255)
```

Code Listing 5.4: The main loop of Xsim, where all the above mentioned processes have been implemented. ray_index_list is the list of paths through the FOV. The ray_counts are the amount of photons corresponding to the different energies. The variable ref_FOV carries the object projected onto the FOV. The array mu_d carries the complete value of linear attenuation coefficients multiplied by the path length. I_ccd is then the final intensity values converted to an image.

5.3 OPTIMIZATION

In order to improve the run time and memory footprint used by the system, different optimization approaches are used. An optimized system allows for a faster examination of different system setups and can generate a large set of data more efficient.

In general, the run time of the whole system is dependent on the amount of traces going through the FOV i. e., it scales with the size of the CCD array $N_x^{CCD} \times N_y^{CCD}$. In a simple evaluation, the system would run through the Bresenham line algorithm, Beer-Lambert law and all other individual components of the system for each index in the CCD array. Giving a system that scales as $\mathcal{O}(N_x^{CCD} N_y^{CCD})$. Using that the FOV is defined symmetrical at the x and y axis with the source in the centre, the amount of iterations used for Bresenham's line algorithm can approximately be reduced by a factor of 8. Dividing the CCD array into 4 quadrants as in Fig. 5.9, it can be seen that all the paths in Q_1 are a mirror of the paths in Q_2. This indicates that the number of iterations can be reduced by a fourth, as calculating all paths in one quadrant should be sufficient. Furthermore, some of the paths in one quadrant can be interchanged by switching the x and y coordinates. Reducing the number of iterations in one quadrant from $N_x^{CCD} N_y^{CCD}/4$ to

$$\frac{N_x^{CCD} N_y^{CCD}}{4} - ((N_y^{CCD})^2/8 - N_y^{CCD}/4), \; N_x^{CCD} \leq N_y^{CCD}.$$

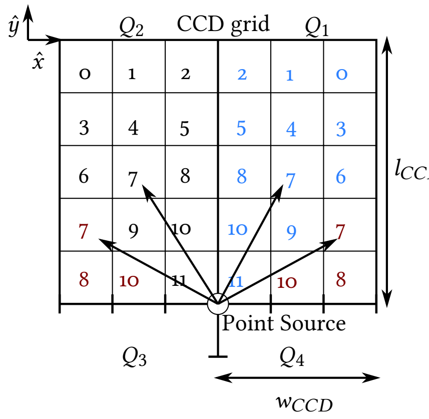


Figure 5.9: Top view of the X-ray simulation system showing the traces from the point source to the CDD array. The indices in the CCD array indicates unique paths for the rays. The path to all the entities labelled 7 is a transformation of each other, and therefore Bresenham's line algorithm only needs to run for all the numbers in black. The two indices labelled 7 in the left quadrant are interchangeable by switching the x and the y coordinates. The indices mirrored over the y axis are interchangeable by adding the width to the indices.

If the array is symmetrical i. e., $N = N_x^{CCD} = N_y^{CCD}$ the amount of unique paths simplifies to $(\frac{(N)^2}{8} + \frac{N}{4})$. Taking the relation between the simple and the symmetrical approach

$$\frac{\frac{(N)^2}{8} + \frac{N}{4}}{N^2} = \frac{1}{8} - \frac{1}{4N},$$

which for big N gives $1/8$. It should be noted that the expression only makes sense for $N > 2$, as the quadrant split would not be defined. Furthermore, this optimization only directly reduces the number of paths calculated by Bresenham's line algorithm. Beer-Lambert law for the object in the path still needs to be evaluated for the separate traces. An example of the optimized iteration scheme is given in Code Listing 5.5

```
delta_qc = grid_qc[0] - grid_qc[1]
count_start = np.arange(0, grid_qc[1]*(delta_qc+1),
                        grid_qc[1])
count_end = np.cumsum(np.arange(grid_qc[1], 1, -1))+
                      grid_qc[1]*delta_qc
counter = np.insert(count_end, 0, count_start)
#Loops over nonreversible  part
for i in range(delta_qc):
  for j in range(grid_qc[1]):
    ind = counter[i]+j
#Loops over the reversible part
for i in range(delta_qc, grid_qc[0]):
  for j in range(i-delta_qc, grid_qc[1]):
    ind = counter[i]+j-i+delta_qc
```

Code Listing 5.5: Optimization of iterations through symmetry. grid_qc is the quadrant of grid_C i. e., half the size of grid_C in each direction.

5.4 SHOWCASE OF XSIM

With a description of the system, we can now proceed with a showcase of some of the possible outputs. A list of all variables controllable in the system is given in § B.1

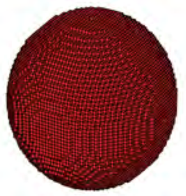


Figure 5.10: Example of a sphere consisting of voxels

5.4.1 *3D graphics figures*

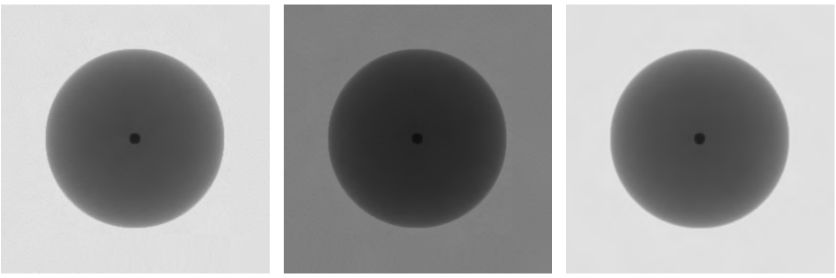
The object of interest can be user-defined by creating a 3D array as purposed in § 5.2.2. An example of such an object is shown in Fig. 5.10, where a homogeneous approximation to a sphere is represented with voxels. More complex structures can also be created as long as they fit in a 3D voxel grid structure. Another approach for finding objects is to use a 3D object defined in a meshgrid, such as shown in Fig. 5.11. As such a file is not directly applicable by xSim, the tool binvox[35] is used. Binvox slices the mesh into voxels at a user defined size. Converting the 3D potato to the the same format as given in Fig. 5.10.

Figure 5.11: Example of a 3D potato used as an object in the simulator.

5.4.2 *A sphere embedded in a sphere*

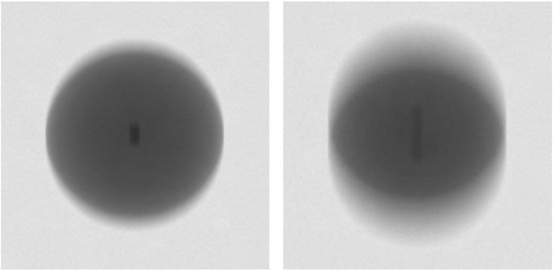
As an example of generated objects, we show a sphere embedded in a sphere. The sphere is defined in a $200 \times 200 \times 200$ grid and is defined to be made of aluminium carrying a smaller sphere of copper inside of it. The sphere is set to have a physical size of 1 cm in radius. For all images the distances are set to $d_1 = 1$ m and $d_2 = 1$ m. In Fig. 5.12, all images only carry one subframe and as so no movement. In (a) and (b) the images have the same camera settings. Set to 200 × 200 pixels with a pixel size of 10 μm. The camera is set to saturation of maximum 4000 photons, normalizing the pixels from 0 to 255 in this range. In (a) there are 3500 sampled photons for every pixel and for (b) there are 2000 photons. This illustrates how the system can simulate different exposure times, as the image (b) is much darker and carry less contrast than (a). In (c) the photon count is 3500, and the camera grid is set to be $400x400$ pixels at a size of 5 μm, giving what corresponds to twice the resolution of (a).

In Fig. 5.13 multiple subframes are used in creating the frames. The camera settings are the same as in Fig. 5.12 (a). To make the photon counts equivalent the 3500 photons are divided by the number of subframes used in the creation of the frames. That is the "exposure time" is equivalent to that of (a), but



(a) High exposure, low resolution (b) Low exposure, low resolution (c) High exposure, High resolution

Figure 5.12: Simulated X-ray images of a iron sphere embedded in a carbon sphere. The images differ in camera options described in the main text. (c) is rescaled for comparison. Runtime: (a) FOV: 34.8 s, Frame 39.34 s (b) FOV: 26.98 s, Frame 25.13 s (c) FOV 106.39 s, Frame 101.04 s.



(a) 0.1 cm speed (b) 0.3 cm speed

Figure 5.13: Simulated X-ray images of a iron sphere embedded in a carbon sphere. The images differ in movement options described in the main text. Runtime: (a) FOV: 35.22 s, frame 746.61 s (b) FOV: 34.26 s, Frame: 747.37 s

the sphere is under movement inducing motion blur. For both images, 20 subframes are used per frame i. e., the object is translated 20 times and added together. In Fig. 5.13 (a) the change in movement is one voxel in the FOV per subframe and in (b) it is three voxels. Corresponding to a movement of 0.1 mm and 0.3 mm in the image capturing time.

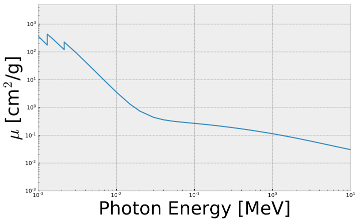


Figure 5.14: Attenuaction coefficents as a function of energy for a potato consisting of 80 % water and 20% starh. Data from **XCOM**[30], it can calculate the total attenuation coefficients for any mixture of elements ($Z \leq 100$).

5.4.3 *A potato containing anomalies*

In the above we use standard chemical components is Xsim. Xsim can be extended to the specific knowledge of chemical objects with the help of the tool **XCOM**[31]. If one where to generate an X-ray image of a common German White potato[36] one could use the chemical composition of it is roughly 80% water and 20% starch. White potato starch is on average consisting of approximately 24% phosphorus, **P**, 26% magnesium, **Mg** and 50% carbon, **C**[36]. This would give the attenuation spectrum as showed in Fig. 5.14, which in combination with Fig. 5.11 can generate realistic X-ray images of a potato.

In Fig. 5.16, two X-ray images of different potatoes containing water and starch are showed, generated under the same conditions: $d_1 = d_2 = 1$ m, camera resolution is 400×400 with a pixel area of size 15μ m. In (a) the length of the potato is 1.3 mm and in (b) it is 2.0 mm. In Fig. 5.16 (b) an iron needle, a aluminum paperclip and a heart shaped hole is added, to demonstrate different types of anomalies. The image is perceptual comparable to the potato X-rayed in Fig. 4.11.

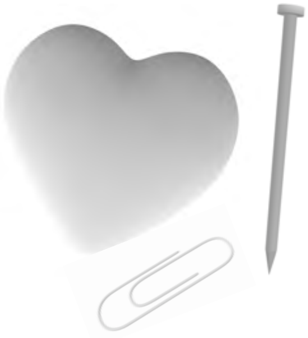
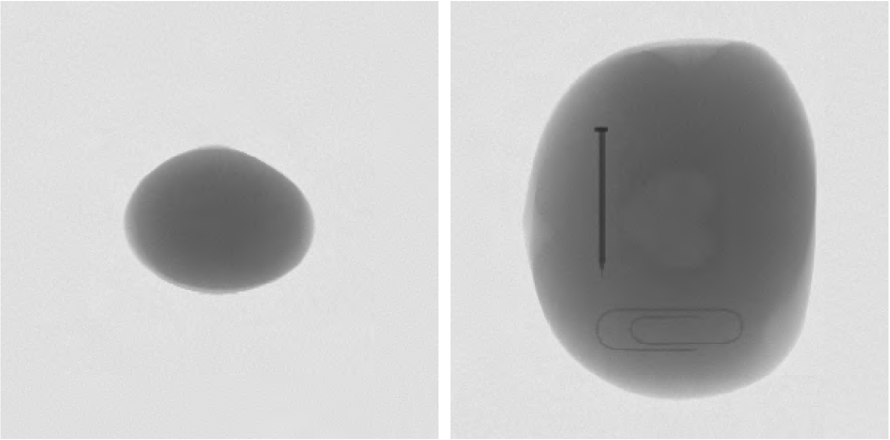


Figure 5.15: Example of different 3D objects applicable for the simulator. Files from [37].



(a) Created in 94.9 s

(b) Created in 104.1 s

Figure 5.16: Example of two different potatoes X-rayed under the same conditions. In (b) different anomalies has been inserted. Runtime: (a) FOV: 34.69 s Frame: 30.16 s (b) *FOV* : 39.04 s, Frame: 30.45 s.

5.5 DISCUSSION

With an outline of the possibilities of Xsim, we will now put it in the context of other simulators and discuss Xsims limitations.

5.5.1 *Related Work*

The idea of an X-ray simulator has in the literature been examined widely, especially with a focus on medical imaging and synchrotrons. The work in [38] provides an excellent overview of different approaches, dividing the methods into two main categories: The Monte Carlo simulations such as GATE[39] and the Ray tracing approach such as gVirtualXRay [40]. These two approaches differ with the Monte Carlo simulations being more truthful concerning scattering, refraction and noise, but at the cost of high computational complexity, in some scenarios taking days for creation of images. The ray tracing method is optimized through a GPU framework making it efficient and able to produce images in a matter of seconds. However, it assumes effects such as scattering is an added noise applied as post-processing. Xsim, in this thesis, relates to the method of ray tracing and are in someways analogues to the work of gVirtualXRay. The two methods differ mainly in the definition of objects, as gVirtualXRay uses a mesh-based approach in contrast to the voxel approach in Xsim. Furthermore, Xsim allows for image capturing of objects under motion, which is of more interest when relating X-ray imaging to inspection on a conveyor belt.

The concept of a precomputed FOV in combination with Bresenhams line algorithm is, to the best of our knowledge, an unique approach to the X-ray imaging scheme.

5.5.2 *Limitations and further work*

In application, Xsim was created as a tool to examine the properties of X-ray images generated under motion. For further analysis of these images, the synthetic images must be as truthful as possible. In the following, we will consider the different approximations made and elaborate on their limits.

Ray Tracing: A direct reflection for how well Xsim represents the natural world lies in the created framework. In the considerations of the individual ray paths from the source to camera pixels, optical phenomena such as scattering are completely neglected i. e., it is indirectly assumed to have no contribution to the final images. This approximation may seem harsh and does limit the system. Scattering is of importance, as it commonly degrades image quality. An effect, seen when considering high-contrast interfaces such as metal versus organic materials the refraction becomes essential. When there is low-contrast between materials, the scattering will mostly contribute as additional noise in the final images. One way to overcome this problem with scattering could be to implement a path-tracing framework[41], which is a Monte Carlo expansion of ray tracing. The effects of absorption and reflection should be modelled sufficiently by the Poisson distribution of Equation (5.4), as they are not registered in the final image and only contribute through there stochastic behaviour.

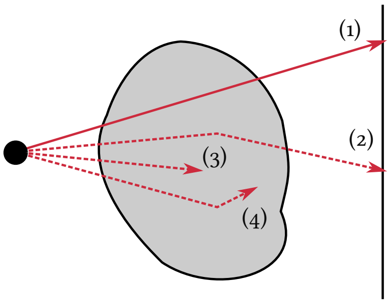


Figure 5.17: Different photon/matter interactions. (1) direct transmission as simulated in Xsim. (2) scattered photon adding noise to image. (3) and (4) absorption of photons in matter.

Point source: In Xsim the source has been modeled as a single point, which is a sufficient approximation at large distances as the pixel sizes would be bigger than the geometric unsharpness[42]. However, this is quite rare in real scenarios and should be taken into account, as it leads to degradations such as added blur. To make a more adjustable and realistic implementation [40] implements a source consisting of multiple point sources. The points sources are spread out in a 2D area, where different shapes can be chosen, to simulate different kinds of real emitters.

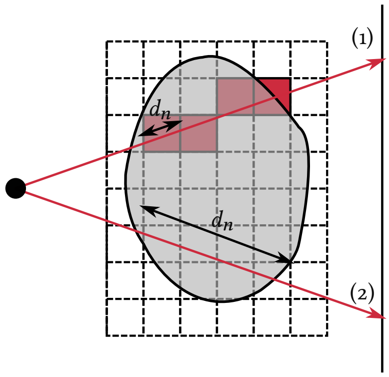


Figure 5.18: Comparison between a sparse and a dense object representation. (1) Dense voxel representation, distance constant per voxel in path length, but iterates over every voxel in the object. (2) Sparse voxel representation finding distances in homogeneous chuncks of the material.

Representation of objects: The voxel representations of objects in Xsim gives an intuitive way of constructing complex objects, as they are build of small bricks representing different materials. Furthermore, the voxel representation works well with the idea of a preprocessed FOV, as the objects can be projected directly into the desired position. Computationally this representation is more inefficient than that of triangular meshes such as used in [40], having a sparse representation of the objects. Here only the surface of the objects is considered, making it sufficient to only iterate over the number of material changes in the ray path, in contrast to the iterations over every layer in the FOV. Xsim might overcome this inefficiency by segmenting out homogeneous parts of the desired object i. e., changing object representations from dense voxels to a sparse surface representation of voxels. However, this is not a guaranteed optimization as the system would need to keep track of the different path lengths in every part of the object. The quantization of mesh 3D objects to voxels provides a challenge for the overall representation, as some object details need to be represented by a fine grid. For the 3D models considered in this work, a quantization of 200 voxels on the longest axis seems sufficient.

Beam Hardening: A general problem in X-ray imaging systems is that of Beam hardening. The phenomena describe how the low energy photons from the polychromatic X-ray beam has a higher absorption rate, changing the spectrum with penetration distance of the object. Essentially it works as a high pass filter and changes the mean energy of the beam to a higher level. This is especially a problem in computed tomography (CT) as it leaves image artefacts such as streaks and cupping[9]. The inclusion of beam hardening in Xsim could be done in two ways. Firstly filters could be put after the source which changes the initial spectrum used for generating photons i. e., high-pass the initial spectrum. Secondly, the photon energy counts could be changed for the different absorption coefficients associated with the different elements. The last contribution would increase the complexity of the system as there for each individual ray path would be a separate photon count.

Rotations: A valuable feature lacking in Xsim is the possibility of rotating the objects. This would allow the simulation of objects under a motion to be more realistic, as rotating for some object shapes would be unavoidable. Furthermore, it could be extended to CT scans of the object. Rotations could be implemented by either changing the position of the source with respect to the FOV, which would make the system more general, but also cumbersome to implement as the main framework should be redefined. Another possibility would be to rotate the object inside the D grid, this will however be limited by the quantization of D and a interpolation scheme should be made to allow angles which would interfere with the defined grid.

Detector: When considering the detector properties of Xsim, we assume that the photons are directly converted to pixel values. This is done by a normalization over the resulting photon counts with relation to a given saturation value. In doing this, we omit the physical properties of the scintillator, optics of the camera, and the analog-to-digital converter (ADC) in the CCDs. The work of [43] provides detailed modelling of these, designed to be implemented in existing X-ray imaging schemes. Here they model the contributions to the point-spread-function (PSF) when considering the scattering in the scintillator, Compton scattering and electron scattering inside the detector. Inclusion of the above should be one of the first steps taken in creating more realistic images. Furthermore, it is essential for proper evaluations of SR algorithms on simulated data, as they are designed to counter it.

Validation of images: A key feature of a good simulator is that it is a valid model. As mentioned, there are several limitations in the implementation of Xsim as the model is limited to quantization, and some physical properties have been omitted. If one were to go further with the work of Xsim, the output should be validated in comparison with real data. An approach could be the creation of a physical phantom object, where all shapes, sizes and materials are well described with high precision. Such a phantom could then be implemented in the Xsim framework, and artificial images can then be compared to X-ray images of the physical phantom. Another approach would be a comparison between different simulators, such as the high complexity images generated from methods such as [39].

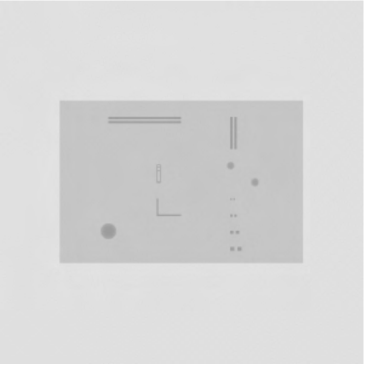


Figure 5.19: Simulated X-ray image of a phantom object containing well defined marks.

Parallelization and GPU: The framework of Xsim is constructed such that each pixel carries an individual calculation of its intensity representation. Furthermore, each frame is computed independently of each other. These two features indicate that parts of Xsim could be parallelized, increasing its general runtime by running on multiple cores. Another approach for optimization of Xsim could be the conversion to a graphical processing unit (GPU) framework, which as it names suggest is optimized for 3D rendering. The GPU is specially optimized for the process of running the same instruction on multiple data, and new types of ray tracing dedicated hardware could especially be of interest.

5.5.3 *The Ideal Xsim*

With a basis in the problem of validation, we propose a scheme for further work on Xsim. Inspired by the work of [44]. Several of the above mentioned lacking features would greatly increase the computational complexity of the system. In Fig. 5.20 we propose a validation model. In this setting Xsim, would be a flexible system, where one can implement the desired components related to the problem at hand i. e., should beam hardening be implemented if no high contrast materials, or is the point source valid at long distances. The system specifications can then be defined in a simulator of higher complexity, giving the same $3D$ phantom as input, the two outputs can be compared with a measure such as cross-correlation. If deemed significant, a justification to the various approximations could be given. Furthermore, the 3D phantom could represent a real phantom with respect to physical properties, allowing Xsim to be validated against a real X-ray imaging setup. If the different outputs are deemed sufficient, **Xsim** can be used as a flexible tool for changing small attributes of the system, such as exposure time or movement speed or examinations of new types of setups in X-ray imaging. The process of comparing with an existing simulator, as the 3D models already exist, should be the first step in the validation of Xsim.

An exciting possibility could be to make statistical representations of the

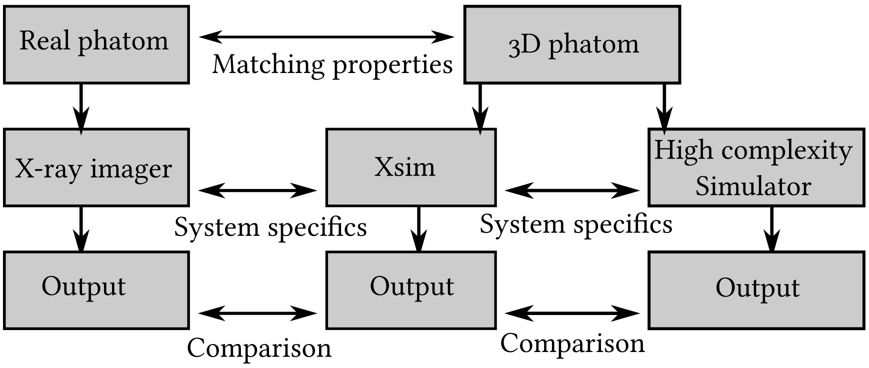


Figure 5.20: Validation scheme for Xsim. An 3D phantom with well described properties is a input in Xsim and a high complexity simulator. The two system variable should be the same. The two outputs can then be compared with respect to there energy maps

desired objects i. e., analyze a small sample and create mathematical models for shape and compositions. If feasible, this would enable the production of a vast amount of 3D models which in combination with Xsim would give a database of X-ray images, usable for testing a classification scheme. This idea is further elaborated in § 6.4.3 along with a discussion of applying the given super resolution (SR).

Xsim is a flexible simulator, allowing the X-ray imaging of various objects under motion. The efficiency of the algorithm in the CPU format allows quick examination of small and coarse objects - images created in a matter of seconds, to finer detailed images generated in a couple of minutes. The voxel description of the objects gives an intuitive way of creating a test object and allows even small details to be tested. However, Xsim has some major and minor flaws and should be properly validated. Especially the lack of scattering and detector properties might be problematic in testing schemes like SR - elaborated in § 6.2.

Part IV

COMBINING THE TWO

6 APPLICATION OF SUPER RESOLUTION

In § 3 we demonstrated the iterative re-weighted super resolution (IRWSR) algorithm developed by [1]. In § 4 the process of acquiring X-ray image was described and in § 5 an X-ray simulator was given. The previous work allows three types of test images of the SR scheme. Artificial degradation of known images, simulated and real X-ray images. As mentioned, the artificial degraded image is a common procedure for comparing different super resolution (SR) schemes in the literature. A standing problem is a gap in performance between the artificially degraded images and real low resolution (LR) images. That is algorithms providing the best results in regards to point signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are not guaranteed to give the most visually pleasing images on real data. This gap is somewhat due to the missing references for quantifying the real images[4]. With Xsim, a possible middle step is given, as one can take control of the imaging process and produce both the multiple LR images and the desired ideal high resolution (HR) images, providing access to a reference image. In Fig. 6.1 we illustrate the three types of images available for the multi-frame super resolution (MFSR) and indicate how they differ.

In the following section, different results of the IRWSR algorithm explained in § 3.4 are showcased. Firstly, in § 6.1, other results for the artificial degradation of test images are shown, indicating the drawbacks of SR. Secondly, in § 6.2, SR images are created from a series of images generated by Xsim. Thirdly, in § 6.3, the motion estimates and SR results are shown for sequences of real X-ray images of a circuit board. At last, in § 6.4, the process is evaluated, and the possibilities of applying SR to a real-time X-ray imaging classification setup is discussed.

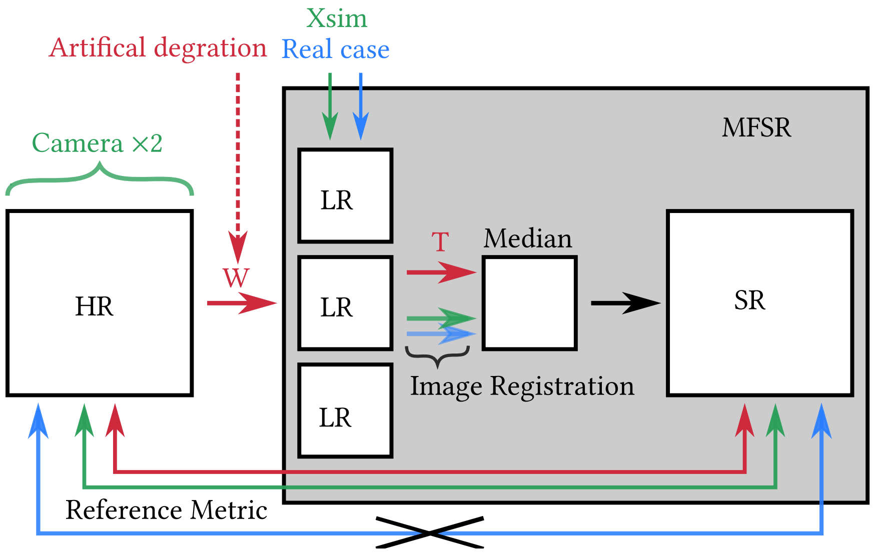
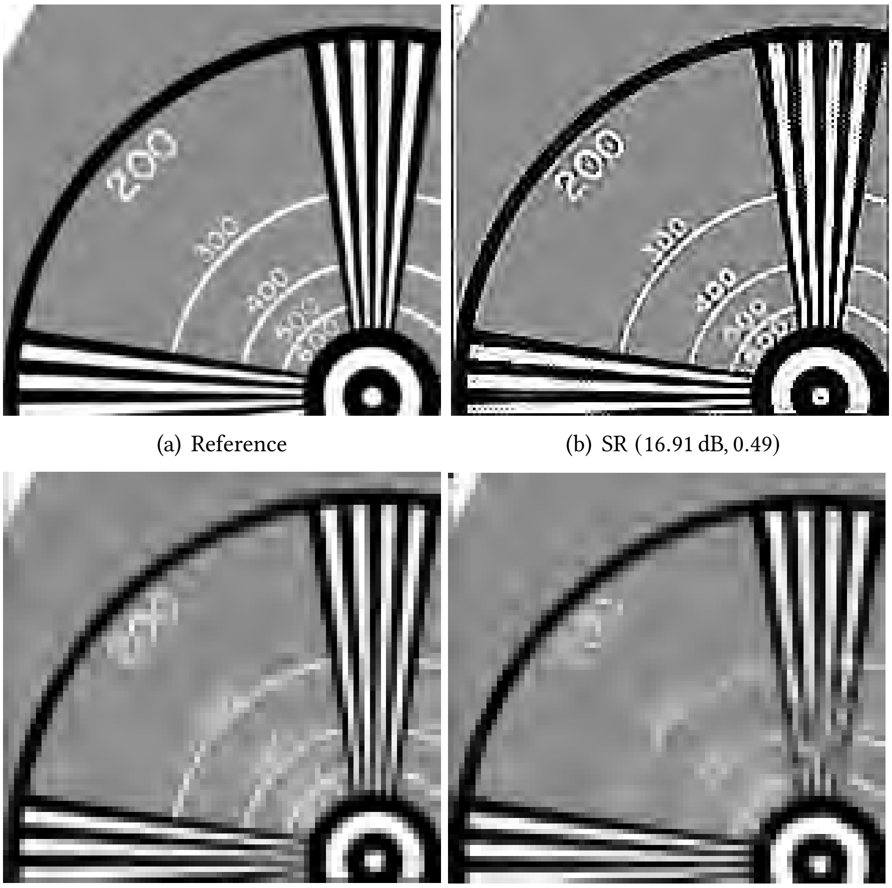


Figure 6.1: The three different data inputs for the MFSR scheme and how they differ. The artificial degradation scheme using a reference high resolution image degraded by W as an input to the MFSR. The translations T for image registration are given by the chosen degradation process. The simulated and real images follow the same process of image image registration in the MFSR scheme, but only the simulated images are given a reference. This is due to the possibility of changing camera resolution with Xsim.

6.1 ARTIFICIAL DEGRADATION

In Fig. 6.2 the upscaling of an artificially degraded image is demonstrated. The SR shows clear signs of a perceptually increased experience compared to the shown LR frame, easing the readability of the numbers. The LR and median are upscaled for comparison. The algorithm introduces artifacts not inherent in the original image. Because the SR is intended for image classification, this could lead to erroneous results and could problematize the classification task. This is not a guaranteed problem, but should be investigated further. The LR frame has a low PSNR compared to the other two. This is due to the noise in the image, as it is a L_2 norm of the distance between the seperate pixels. The SSIM is highest for the LR image, which could be due to it being a measurement created for validation of compression.



(a) Reference

(b) SR (16.91 dB, 0.49)

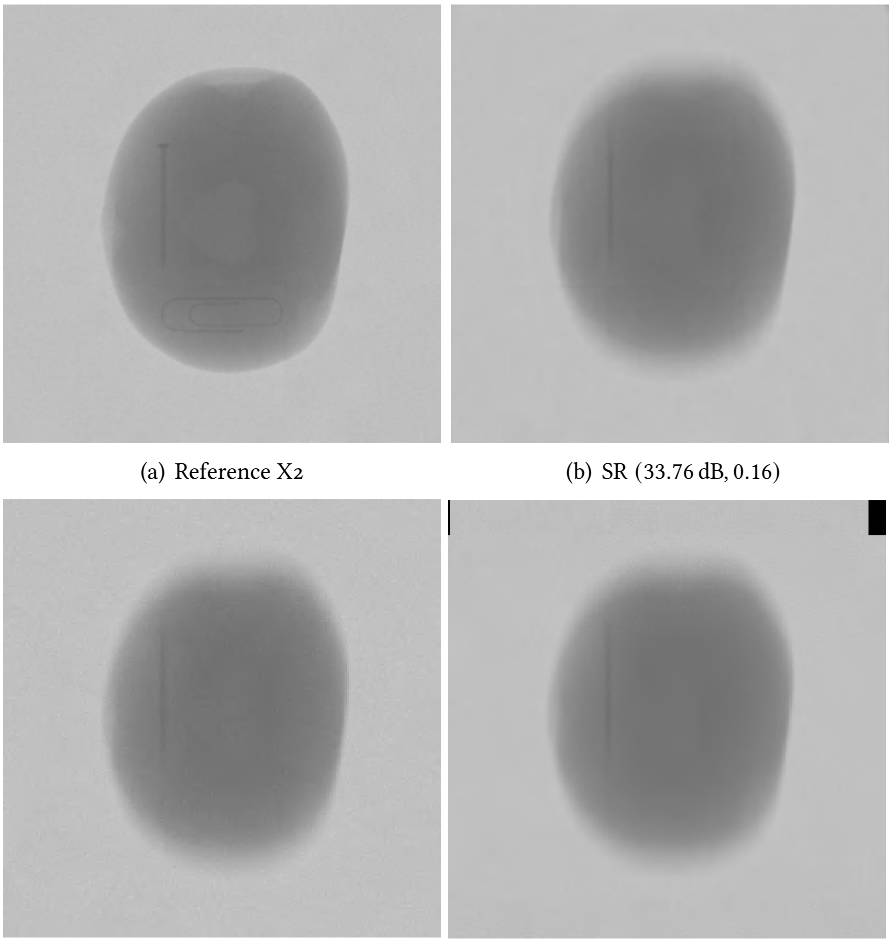
(c) Degraded LR frame (8.81 dB, 0.74)

(d) Median (20.14 dB, 0.40)

Figure 6.2: MFSR image showing clear signs of artifacts. The image is magnified by $s = 2$ from 80×75 to 160×150 in steps of $\Delta s = 0.1$. The LR frames were created by random subpixel translations in the range of 5 pixels, blurred by a Gaussian blur of $\sigma = 0.5$ and downscaled to half the original size by nearest-neighbour interpolation. The regularization parameter $\lambda = 0.0028$ was found through cross-validation and the total runtime was 508 s. The first measure is the PSNR and the second the SSIM. The median and one of the degraded LR frames upscaled by bicubic interpolation for comparison.

6.2 SUPER RESOLUTION ON SIMULATIONS

In Fig. 6.3 SR is demonstrated for the potato containing anomalies. The LR frame belongs to a sequence of 7 images with a speed of 0.3 mm per frame. In the LR image the needle characteristics are converted to a line, the heart barely visible and the paper clip gone. Neither the median or the SR image provides any assistance in recovering the objects, and mostly function as a computationally expensive noise removal filter.



(a) Reference X2

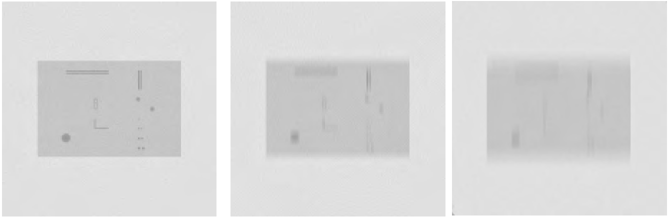
(b) SR (33.76 dB, 0.16)

(c) Original LR frame (0.06 dB, 0.91)

(d) Median (22.68 dB, 0.15)

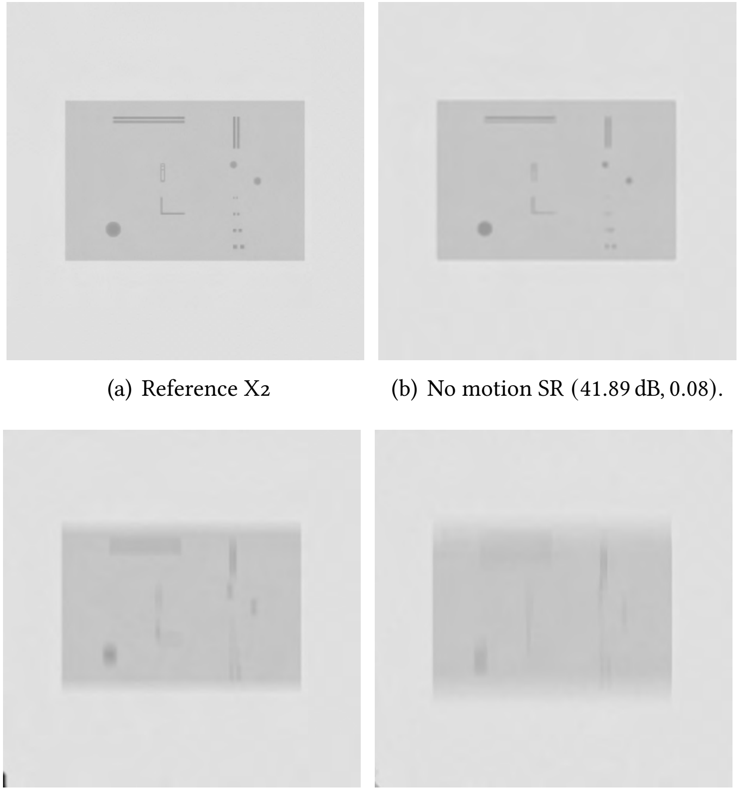
Figure 6.3: MFSR image of the potato containing anomalies given in § 5.4.3. The potato is set in a bounding box of $2.5 \times 2 \times 3$ mm, $d_1 = 2$ cm and $d_2 = 1$ cm The camera is of size 200×200 with a pixel size of 0.03 mm. The reference image is generated with a camera of size 400×400 and pixel size 0.015 mm. In the IRWSR the image is magnified by $s = 2$ with steps of $\delta_s = 0.25$. The regularization parameter $\lambda = 0.056$ found through cross-validation. Process took 1909.7 s.

In Fig. 6.4 an X-rayed board with marks of different shapes, sizes and alignments is demonstrated. The images are generated for different movements and are a part of a sequence of 7 images used for MFSR illustrated in Fig. 6.5. For the LR frame there is increasing loss of detail with increasing speeds. However, it would not seem that the SR scheme is able to reconstruct the details for any speed.



(a) No motion (b) Low motion (c) High motion

Figure 6.4: Generated LR images of a carbon board with marks at different speeds. The height of the carbon object is 5 cm with a width of 45 cm. The iron engraving is 0.5 cm thick. Distances are $d_1 = 1m$ and $d_2 = 1m$. All frames belong to a sequence of 7 images. In (a) the object is translated such there is no motion blur. In (b) and (c) the board carries a speed of 0.1 mm per frame and 0.2 mm per frame. The camera size is 150 × 150 with a pixel area of 30 μm



(a) Reference X2 (b) No motion SR (41.89 dB, 0.08).

(c) Low speed SR (34.92 dB, 0.06). (d) High speed SR (35.43 dB, 0.04).

Figure 6.5: MFSR image of an carbon board with iron marks sequences given in Fig. 6.4. The LR frames where magnified by $s = 2$ in steps of $\Delta s = 0.25$. To reduce runtime the regularization parameter $\lambda = 0.038$ was constant. The reference image is generated with a camera set to 300 × 300 and pixel size 15 μm. The run times are (b) 637.29 s, (c) 717.86 s and (d) 2121.8 s

6.3 AN X-RAYED CIRCUIT BOARD

In the following the IRWSR algorithm is applied to real X-ray data. We provide two types of datasets: One where there is a manual translation of the object in the individual frames i. e., no motion in the frames and one where the object is placed on a running conveyor-belt. The X-ray images are taken at different exposure times, providing a dataset where we can test the various effects of motion blur and contrast. The object of interest is the circuit board shown in Fig. 6.6.

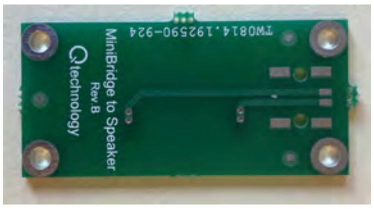
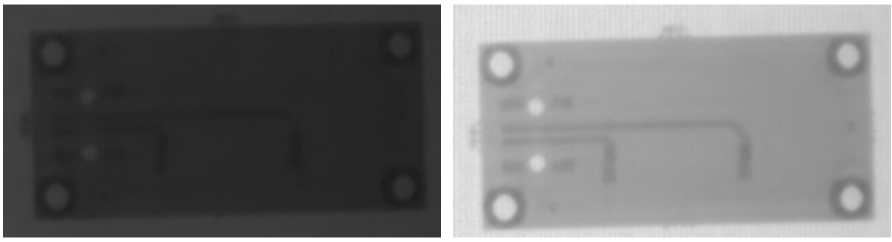


Figure 6.6: Optical image of the examined circuit board.

The images are all taken with the setup given in Fig. A.1, with the X-ray generator set to a voltage of 50 kV and a current of 8 mA. All images are manually cropped to fit the object, with as little background as possible and all images has been passed raw to the IRWSR scheme. The motion estimates were found through sparse optical flow explained in § 2.4.4. For illustrative purpose the SR images are normalized to the whole 8bit range. In this series of images the metrics of PSNR and SSIM cannot be applied as no HR reference exist. In the following a constant regularization parameter $\lambda = 0.038$ is used for all runs of the IRWSR. The choice of a constant value is based on the cross-validation not converging to an image representing the LR frame. The regularization value is chosen based on a manual scan of various lambda values.

Manual Translation

First, we look at the manually translated image sequence. The frames has been gathered through the process of placing the circuit board in the X-ray scanner. An image was the taken for the three different exposure times 10 ms, 50 ms and 200 ms. The circuit board was then moved approximately 1 mm and three new images where taken for each exposure time. In Fig. 6.7 a single frame from the 50 ms and 200 ms is shown.



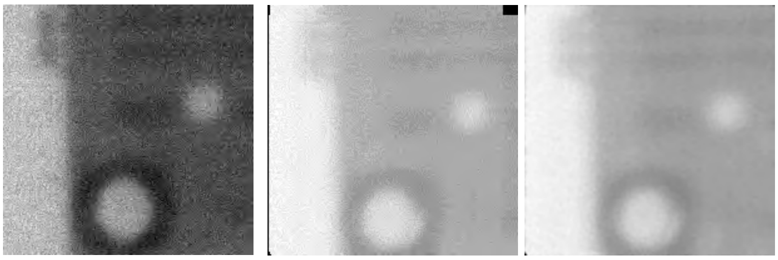
(a) 50ms frame

(b) 200ms frame

Figure 6.7: Raw image from the 50 ms and 200 ms exposure time for the translation series.

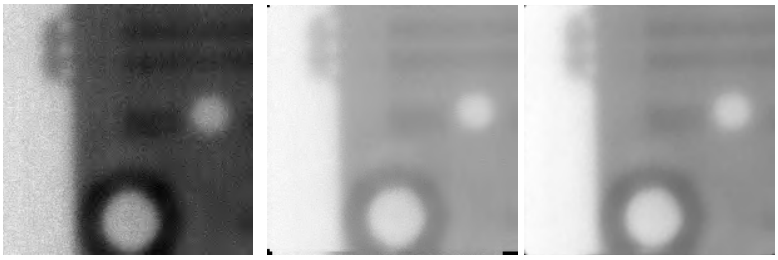
In Fig. 6.8 and Fig. 6.9 examples of the IRWSR algorithm applied to the 10 ms and the 200 ms sequence is shown. The images has been cropped to a size of 200×200, due to the large run time of the algorithm. The frames are magnified by $s = 1.3$ to 260×260 with coarse-to-fine steps of $\Delta s = 0.1$. The regularization parameter is set to the constant $\lambda = 0.038$. All images normalized to a range of 255 to optimize visual perception. The LR frame and median is interpolated through bicubic interpolation to the size of the SR.

The specific features of interest are the holes of three different sizes and the lines. Especially the small holes in the top left corner are unclear for the unprocessed image in (a). The application of the SR algorithm does not perceptually enhance the features of the unprocessed image and the amount of variance seems to be insignificant in connection with classification.



(a) Original LR frame (b) Median (c) SR

Figure 6.8: SR for 10 ms of a 200×200 crop of the original frames. The frames where magnified by $s = 2$ in steps of $\Delta s = 0.25$. To reduce runtime the regularization parameter $\lambda = 0.038$ was constant. Processing time 1033.2 s. All images normalized to a range of 255 to optimize visual perception.

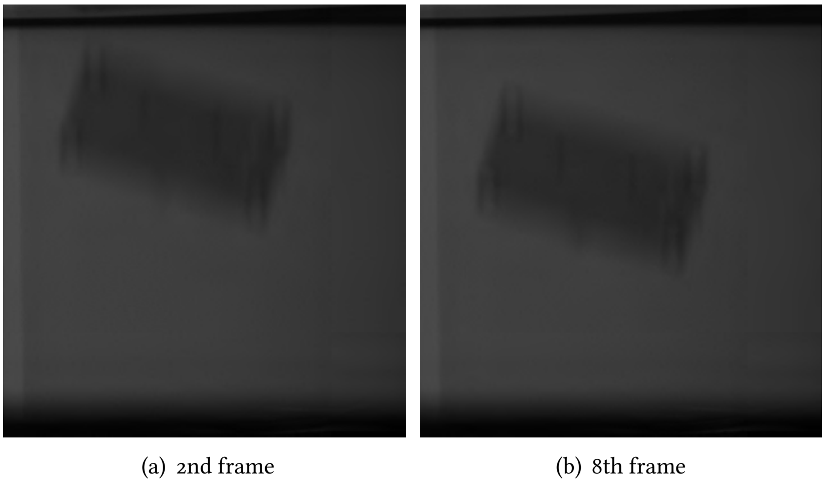


(a) Original LR frame (b) Median (c) SR

Figure 6.9: SR for 50 ms of a 200×200 crop of the original frames. The frames where magnified by $s = 2$ in steps of $\Delta s = 0.25$. To reduce runtime the regularization parameter $\lambda = 0.038$ was constant. Processing time 1718.5 s. All images normalized to a range of 255 to optimize visual perception.

Conveyor translation

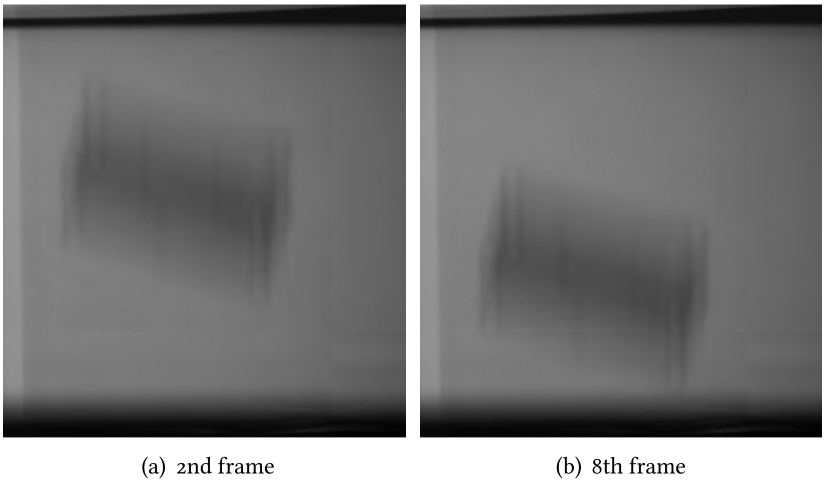
A series of frames of the circuit board under movement has been generated. The movement is induced by a conveyor-belt moving at 15.5 cm/s and the different exposure times are 13.6 ms, 16 ms, 25 ms, 50 ms and 100 ms. As a natural consequence of different exposure times, the sampling rate varies from 10 fps for the highest exposure time to 70 fps for the shortest. In Fig. 6.10 and Fig. 6.11 two sequential frames are shown for the exposure times of 50 ms and 100 ms. There is a notable difference in distances and motion blur as the speed exposure time increases.



(a) 2nd frame

(b) 8th frame

Figure 6.10: Two sequential frames for the conveyor series. Exposure time is 50 ms corresponding to a frame rate of 20 fps.

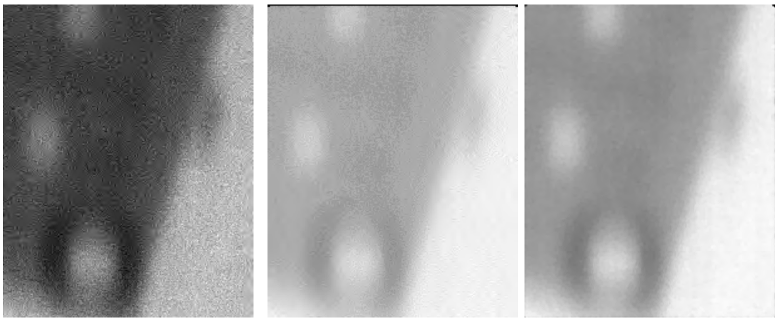


(a) 2nd frame

(b) 8th frame

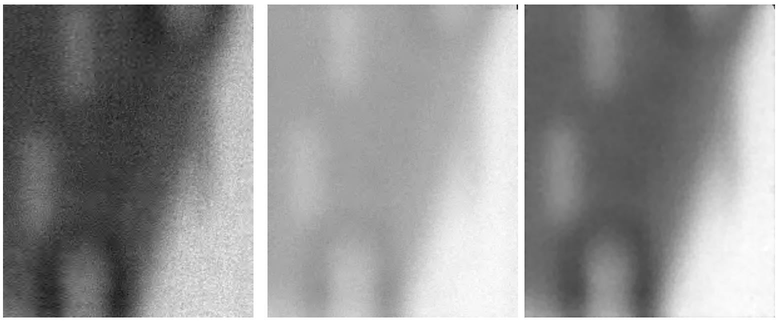
Figure 6.11: Two sequential frames for the conveyor series. Exposure time is 100 ms corresponding to a frame rate of 10 fps.

In Fig. 6.12 and Fig. 6.13 examples of the IRWSR algorithm applied to the 13 ms and the 25 ms sequence is shown. Like with the manual translation a cropping has been made of 250 × 200 to increase the runtime. The frames are magnified by $s = 1.3$ to 325×260 in steps of $\Delta s = 0.1$. The regularization where for both sequences set to the constant $\lambda = 0.038$. The LR frame and median is interpolated through bicubic interpolation to the size of the SR. The features of interest are the same holes and stripes described before. No notable enchantment other than noise removal is noticed.



(a) First frame (b) Median (c) SR

Figure 6.12: SR for 13 ms of a 250×200 crop of the original frames. The frames where magnified by $s = 2$ in steps of $\Delta s = 0.25$. To reduce runtime the regularization parameter $\lambda = 0.038$ was constant. Processing time 360.7 s. All images normalized to a range of 255 to optimize visual perception.

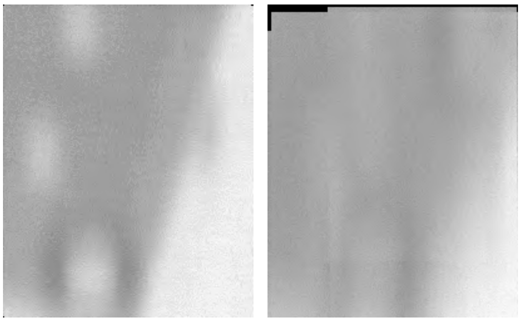


(a) First frame (b) Median (c) SR

Figure 6.13: SR for 25 ms of a 250×200 crop of the original frames. The frames where magnified by $s = 2$ in steps of $\Delta s = 0.25$. To reduce runtime the regularization parameter $\lambda = 0.038$ was constant. Processing time 434.0 s. All images normalized to a range of 255 to optimize visual perception.

6.3.1 Motion estimates

A vital part of the IRWSR scheme is the motion estimates used in the degradation model W and the median giving the initial guess for the MFSR method. The algorithm is intended to be robust at subpixel misalignment, guaranteed by the median of the residuals in the calculations of the β maps. However, when the misalignment is at pixel level and above, the scheme seems to fall apart, which is apparent when applying the segmentation/optical flow scheme described in § 2.4.4 to the motion blurred images of the conveyor-belt. With an increased blur, the motion estimates seem to misalign the frames as shown in Fig. 6.14, where a poor motion estimate apparent in the median severely decreases the output of the SR scheme. However, as the implemented SR does not enhance the images with less motion blur, the optimization of the image registration for high motion blur is not our biggest concern.



(a) 16ms, median (b) 50ms, median

Figure 6.14: (a) well aligned frames in median and (b) misaligned frames in median.

6.4 LIMITS FOR APPLICATION OF SR

In the following we discuss our findings of the implemented IRWSR schemes. The Algorithm is put in context of machine learning algorithms, such as neural networks, dominating the present evolution in the field, as an elaboration of neural networks is out of the scope of this thesis. They are mainly considered on a conceptual level, as a method to analyze large amounts of data and finding inherent patterns in the images.

6.4.1 Baysian vs. ML

To the author's knowledge there are no widely applied SR algorithm based on the Bayesian framework presented in § 3. In the applied SR review [45] from 2016[1] the SR advancements are divided into three stages. Frequency based methods, regularized MFSR and single image super resolution (SISR) based on example patches. The analysis from [45] seems to fit with the present overload of SISR machine learning algorithms, as seen in this list of 88 open source projects algorithms[3]. At the time of writing this trend seems to

[1] The same year as the IRWSR was published[1].

continue with new network architectures being developed, surpassing its priors in terms of PSNR, SSIM and other quality measures. The work of [4] from 2019 sought to make a publicly available benchmarking system for evaluation of SR algorithms. Here, the robustness of the IRWSR scheme seemed to outperform the compared machine learning methods on several quantities such as PSNR and SSIM. These evaluations were performed on scales of different image degradation, i. e., a range of varying PSFs and motion translations. Sadly, it seems that the newer algorithms has not been compared to this dataset. Thus, with the lack of a common benchmark it is difficult to quantify the robustness of the machine learning algorithms with respect to the IRWSR.

Another problem stated in [4] is the gap in performance between SR algorithms on test data and real data. Since the beginning of this project a growing amount of cases of commercial available implementations of SR based on machine learning[5][6]. However, these are made for increasing visual perceptions, where the goal of an SR algorithm in this thesis is the enhancement of edges and features as a preprocessing tool for a classification algorithm. A phenomena with mixed results and not widely researched in the literature. In [46] SR is used as a preprocessing tool for a classifying convolutional neural network (CNN) but concluded it unnecessary. In [47] the architecture of an SR is combined with a CNN, reporting an increased performance in classification.

6.4.2 Optimization of IRWSR

In our opinion it seems that the choices of a SR implementation lies between a Bayesian multi-frame model and a neural network performing on single frames. As demonstrated above, the IRWSR lacks the computational efficiency to perform in a online system; Even the ×2 magnification of a 200×200 sequence of frames takes more than 1000 s on a single core CPU. Especially the computation of the W matrix and the conjugate gradient (CG) are limiting the performance of the system. Some approaches can be taken to the optimization of the computation of W. A precomputation of W can be done if the system in mind is well described, i. e., the translation are known without image registration methods. Another more robust approach based on the discussion in § 3.2 would be to change the degradation model to conventional filters W. This would skip the computation of W as a matrix and only leave an evaluation. However, in the CG method the gradient calculations are essential and the translation of W in Equation (3.9) would be difficult to define[2]. A third approach could be a method taken from the CNN community; the im2col representation[48]. Here, the image is changed to a column matrix representation and the convolution is a simple dot product with the flattened kernel. This seems like a promising approach to decreasing the runtime and

[2] A quick implementation of this method provided no real results.

keeping the properties of the dot product in the equations. Furthermore, the blurring part of the degradation kernel is assumed Gaussian and lacks the features of motion blur. This would demand a throughout analysis of the point-spread-function (PSF) for the given system, as this might cause a nonlinear PSF dependent on spatial properties. The estimated PSF should be implemented in Equation (3.2), as an alternative to the Gaussian approach.

6.4.3 Validation of SR

The IRWSR has slow computational performance and the possibility of induced artifacts, as shown in Fig. 6.2. Furthermore, it showed no clear visual signs of finding latent information in the LR frames of the simulator or the real images of the circuit board. This indicates that the IRWSR would be a poor choice for preprocessing images for a real-time classification algorithm. If the runtime of the classification pipeline were unimportant, it would be valid to test if IRWSR increased the performance. This could be done with a greater dataset of simulated or real images, with a well defined classification task. Such as detection anomalies in a potato or the like.
Another promising approach to SR in this setting, would be the implementation of newer SR machine learning algorithms. As some trained networks can process a SR image in a matter of seconds[5], indicating that they possess the speed needed in a pipeline. However, as networks are trained for a specific setting, they are not guaranteed robust. Summing up, a validation of a SR algorithm(or any preprocessing algorithm) in a classification scheme needs to be related through the input and output of the larger pipeline, a process outlined in Fig. 6.15. The objects of interest are imaged in an X-ray setup.

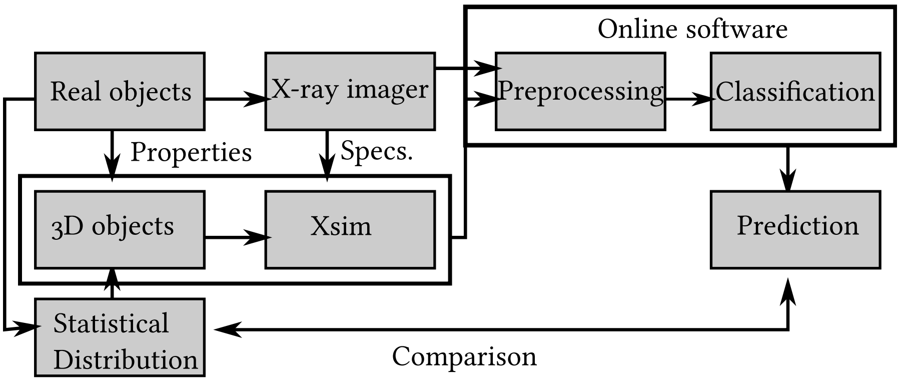


Figure 6.15: Suggested flowchart for validation of different preprocessing algorithms in a classification pipeline. Objects are send through an X-rap image an preprocessed in a online system. The prediction of the classification should then be related to the statistical distribution of the given input. Furthermore, we suggest that Xsim can be an addition to this pipeline, providing simulated data based on the real input.

The images are then processed before being classified. The predictions of the classifier are hereafter related to the input object, i. e., assessment of whether the known anomalies are found. The preprocessing can in this case be advanced filters, such as super resolution, or more common, such as gamma correction and histogram equalization. This provides a flexible method of trying out different combinations of filters analogous to the cross validation used in § 3 and common for parameter optimization in machine learning.

Furthermore, we suggest the addition of Xsim to the data acquisition. If possible, the real objects can be converted to statistical 3D models, used in the generation of various realistic objects. The 3D objects can then be induced different anomalies pulled from a distribution, such as holes or foreign objects. This would be analogous to that of data argumentation used for increased robustness in machine learning models. If followed by validation checks of the simulator, as suggested in § 5.5.3, it would become a powerful tool in optimization of classification algorithms in X-ray imaging.

7 CONCLUSION

In this thesis we have proposed the application of super resolution (SR) in the context of X-ray scanners for food inspection. The goal were to enhance image features, such that it could support the classification task in the overall pipeline. As SR can be viewed as high-level computer processing algorithms, we started by providing a short theoretical and applied foundation for the subject. Demonstrating the filters and computer vision algorithms necessary for implementation of the desired scheme.

The applied iterative re-weighted super resolution (IRWSR) was chosen on the basis of a review of several SR algorithms given in the literature, because it showed promising results in with regards to robustness and output, in comparison with its competitors. The content of the algorithm was elaborated and results from our implementation were shown for artificial degraded data. These results indicated that the implementation were able to convert a sequence of low resolution images to a single high resolution image, showing the reappearance and sharping of features lost in the image degradation process.

Regarding X-ray scanners, the physical theories behind X-ray images were elaborated. Relating the theories of X-ray sources, object attenuation and detectors to the practical considerations behind using such a setup. A short demonstration of a real X-ray scanner used for generating the images analyzed in the context of SR were given.

For examination of the possibilities of X-ray scanners in general, we created the tool **Xsim**. Xsim allows simulation of the imaging process of objects X-rayed under motion. The creation of the algorithm were explained with regards to the geometrical and physical considerations. The algorithm outputs were showcased providing perceivably realistic results of an X-rayed potato. Xsim was then put in the context of related work, together with an examination of its limitations. However, a validation by comparison to a more complex simulator could specify the range of these limits.

At last, we applied the implemented SR scheme on the different types of data acquired. For artificial data degradation a positive perceptual result is given, however, the inducement of image artifacts could indicate problematic behavior in a larger pipeline. With the simulator, images were generated on two different magnification levels for the camera, providing a reference for the SR output. The SR algorithm was then applied to sequences of data with and without induced motion, providing no signs of positive results in regards to the perception. A possible reason could be the lack of a proper implementation of the point-spread-function (PSF) for Xsim, as it is the low-pass behavior of such a filter, multi-frame super resolution (MFSR) seeks to overcome. In the examination of the application for real world data, different sequences of exposure time and motion blur were provided. No perceivable, positive result were present with the application of SR, and the algorithm functioned as a noise removal filter. Furthermore, the overall computational time of the IRWSR scheme seems insufficient for a real-time application. As an example the magnification of an image of size 200×200 by $\times 2$ is done at the scale of 1000 s. Because the data quantitiy is small, and no standardized validation scheme exists, the work of this thesis lays the foundation on how to proceed with the application of SR in the context of an X-ray scanner. For significant results the implemented tools should be validated through the two proposed schemes.

7.0.1 Future work

The process of applied SR is still a widely researched area for which there are little significant results in the context of a classification pipeline. As the amount of available algorithms only grows, one could try the application of newer models which seem to outperform the others. Before doing so, one should however have a clear idea of the performance benchmark. For applied SR with classification algorithms, the benchmark could be the prediction rate from a known input distribution.

Related to the IRWSR, different improvements could be made. First of all, an optimization of the run time for the degradation process should be made. This could be achieved by implementations of convolutions by filters, instead of the cumbersome creation of W as a sparse matrix. Trial implementations indicated an increased run time for the degradation process, which could be feasible for proper online implementations. However, the output images were not representative of the input. A reason could be the gradient of the energy function for the IRWSR not being well defined in this matter. As such, this minimization with the conjugate gradient (CG) was not evaluated properly, and a more throughout investigation should be made.

A main topic of this work is the motion blur induced by a moving conveyor-belt. However, a flaw is the lacking analysis of the contributions to the PSF. Such a PSF could be implemented as a part of the degradation process, which should provide more truthful residuals between the degraded image and the given low resolution image, and as such provide a better estimate for a non-blurred image.

The implemented IRWSR algorithm could further be used as the backbone for a vast type of image reconstruction processes. A versatile reimplementation could allow for easy changes of different parts of the algorithm. That could be the trial of different priors or the steps in the degradation process e. g.

a reconstruction could be to omit the downscaling. The examination of different minimization methods could also be examined.

Regarding the future of Xsim, it is the foundation for a powerful simulation tool. There is a prominent lack of considerations to a range of physical processes, such as scattering, the spatial extension of the source, beam hardening and the PSF contribution from the scintillator-detector combination. The degree of error from these approximations should be examined through a combination of mathematical estimates and the proposed validation scheme. The validation scheme relates the X-ray image of an input phantom to the output of a real-setup or a more complex simulation. As some physical effects are more prominent than others for a given setup, this validation should be done at different setup limits, e. g. investigating what is the effect of an spatially extended source related to the distances, how does different material compositions influence the scattering, etc.

Furthermore, the different physics should be implemented. In comparison with the SR scheme, the inclusion of the PSF is an important first step, as it is the degradation from these low-pass effects it is designed to counter. The acquired PSF could also be implemented as a part of the IRWSR degradation, analogous to the proposed implementation of a motion-blur kernel.

For a runtime optimization, the inherently separable structure of the framework could be utilized by parallelization. An implementation on a GPU would also be a low-hanging fruit, as they are optimized for 3d processing. Newer models even feature dedicated ray-tracing hardware, which sounds like a perfect fit for the simulator.

Using Xsim in relation to the AXIS-project, whose final goal is a fully automatic classification system for X-ray scanners, the examination of the statistical properties of shape, density and anomalies of target objects could be utilized to produce 3D-graphical representations, allowing tests on a large, and vast, amount of generated data. Such an investigation should start of by looking into simple and approximately homogeneous objects, such as potatoes, and can then be expanded to more complex elements.

APPENDICES

A ADDITIONAL SR

A.1 SYSTEM SPECIFICS

All software and runtimes in this thesis are run on the following system. ERDA - DAG hardware:

- CPU: 1st Generation AMD EPYC @ 2 GHz, 8 threads time sliced.
- RAM: 8 GB DDR4 2666 MHz

The following packages with corresponding versions has been used:

- Phyton 3.7.6
- numpy 1.18.0
- openCV 4.2.0
- scipy 1.4.6

A.1.1 *Nonparametric Blind Super-Resolution*

The following provides a short review of two additional super resolution (SR) algorithms examined in this work. The idea behind the nonparametric blind SR[19] is to address the blur kernel used in the image degradation process. Addressing that the use of low-pass filter such as the gaussian in § 3.2, leads to low quality SR as it neglects effects such as zoom, focus and camera shake. They propose a blind estimation method based on the maximum a posteriori (MAP) for automatically estimating the kernel between high resolution (HR) and low resolution (LR) patches of the desired image. Using a property from natural image statistic that different patterns of images recur across resolution scales of the same scene. The nonparametric part of the kernel is that it follows no known distribution and allows it to take negative values.

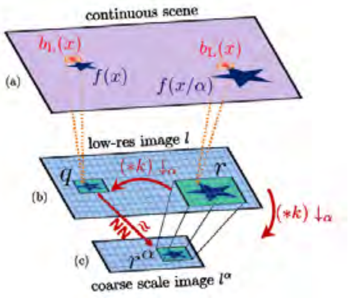


Figure A.1: The exploiting of patches recurring across different image scales. Image from [19]

The kernel estimation is independent of the bigger SR scheme and can be implemented as a substitute to the blur H, in the degradation model $\boldsymbol{W}$. The source code for the kernel estimation is not public available, and our implementation seemed flawed as it always converge to a gaussian like kernel. The authors addresses some of the faults of the original work and has recently provided a blind estimation scheme based on a generative adversarial network (GAN), with public source code[49]. Future work on the iterative re-weighted super resolution (IRWSR) algorithm should examine the possibilities of including such kernels.

A.1.2 Showcase of data used in artificial degradation of images

Showing a sequence of 8 artificial degraded LR frames used in the multi-frame super resolution (MFSR) scheme.

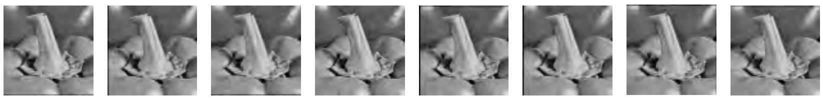
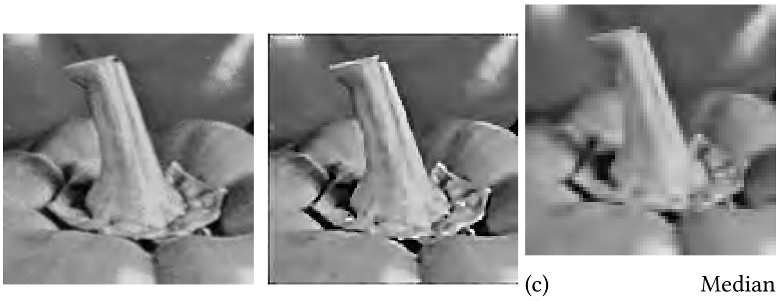

(a) (b) (c) (d) (e) (f) (g) (h)

Figure A.2: An example of a sequence of 8 artificial degraded LR frames used in the MFSR scheme.


(a) Reference

(b) SR (30.83 dB, 0.73)

(c) Median (31.72 dB, 0.65)

Figure A.3: An example of the peppers image and super resolution. The first measure is the PSNR and the second is the SSIM. The median is upscaled by bicubicinterpolation to its original size.

A.1.3 Extra data of circuit board

A.1.4 Code snippets related to the IRWSR

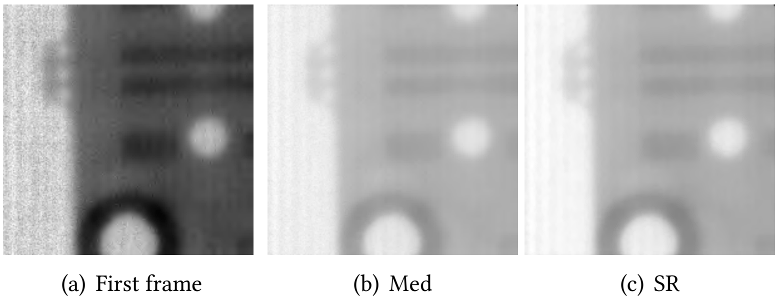

(a) First frame (b) Med (c) SR

Figure A.4: SR for 200 ms of a 200×200 crop of the original translated frames. Image magnified by $s = 1.3$ to 260×260 with coarse-to-fine steps of $\delta s = 0.1$. The regularization parameter set to the constant $\lambda = 0.038$. Took 385 s. All images normalized to a range of 255 to optimize visual perception.

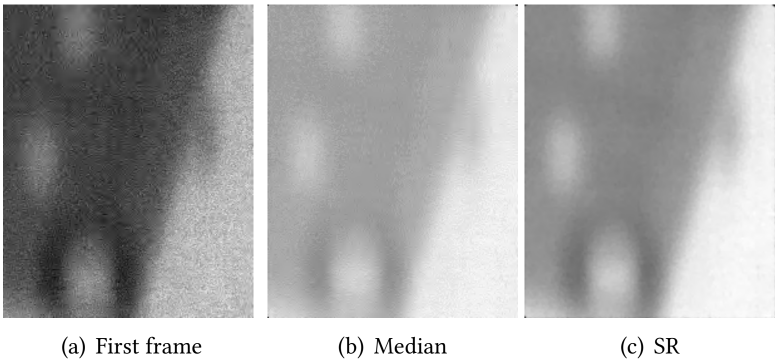

(a) First frame (b) Median (c) SR

Figure A.5: SR for 16 ms of a 250×200 crop of the original conveyor frames. Image magnified by $s = 1.3$ to 325×260 with coarse-to-fine steps of $\delta s = 0.1$. The regularization parameter set to the constant $\lambda = 0.038$. Took 410.7 s. All images normalized to a range of 255 to optimize visual perception

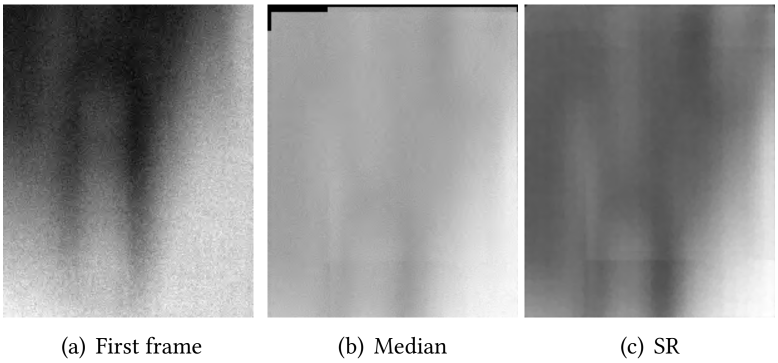

(a) First frame (b) Median (c) SR

Figure A.6: SR for 50 ms of a 250×200 crop of the original conveyor frames. Image magnified by $s = 1.3$ to 325×260 with coarse-to-fine steps of $\delta s = 0.1$. The regularization parameter set to the constant $\lambda = 0.38$. Took 458.2 s. All images normalized to a range of 255 to optimize visual perception.

```python
def createSh(img_size, n):
  N1, N2 = img_size
  # creates one block
  block = eye(N2, N2, -n)
  # string with number of blocks in diagonal
  num_blocks = "block, " * N1
  # creates whole block matrix
  blockMatrix = block_diag((eval(num_blocks)), format = "csr")
  return blockMatrix

def createSv(img_size, m):
  N1, N2 = img_size
  # creates block matrix
  N = N1 * N2
  blockMatrix = eye(N, N, -m*N2)
  return blockMatrix

def createSmn(img_size, m, n, alphaBTV):
  N1, N2 = img_size
  alpha = alphaBTV**(np.abs(m)+np.abs(n))
  S = createSv((N1,N2), m).dot(createSh((N1,N2), n))
  return alpha*(identity(N1*N2)-S)

def createSStack(img_size, P, alphaBTV):
  N1, N2 = img_size
  # meshgrid of all posible m, n values
  pRange = np.arange(P*2+1)-P
  pGrid = np.meshgrid(pRange, pRange)
  # create string with all the diffrent Smn matrices
  SmnStr = "["
  for i in range((P*2+1)**2):
    SmnStr += f"createSmn((N1, N2), {pGrid[0].ravel()[i]},"\
      +f"{pGrid[1].ravel()[i]}, alphaBTV), "
  SmnStr += "]"
  # Creates s as a stack of sparse matrices
  S = vstack(eval(SmnStr))
  return S
```

Code Listing A.1: The generation of the sparsifying matrix. Implemented with numpy and the scipy sparse library.

```
def composeSystemMatrix(lrshape, magFactor, psfWidth,
    motionEstimate):
# LR shape
M1, M2 = lrshape
M = M1 * M2
# HR shape
N1, N2 = int(round(M1 * magFactor)),
    int(round(M2 * magFactor))
N = N1 * N2
# Define all the pixels in LR
uX, uY = np.arange(0, M2), np.arange(0, M1)
# Define all the pixels in HR
vX, vY = np.arange(0, N2), np.arange(0, N1)
# max distance of the supported part of the psf.
# guarantees that it is atleast 1 pixel wide
maxPsfRange = max(3 * psfWidth * magFactor, 1)
# Find subpixel positions of the given LR frame in HR
uPrimeX, uPrimeY = (uX + motionEstimate[0])*magFactor,
    (uY + motionEstimate[1])*magFactor
# initiate the matrix W
W = lil_matrix((M, N))

# iterate over every pixel in LR.
for i_y in range(M1):
  for i_x in range(M2):
    # finds distance between all pixels in HR and u'
    distX, distY = np.abs(uPrimeX[i_x] - vX),
      np.abs(uPrimeY[i_y] - vY)
    # mask defining the squared support of the PSF
    maskX= distX <= maxPsfRange
    maskY =distY <= maxPsfRange
    dist = np.meshgrid(distX[maskX],distY[maskY])
    # Finds all euclidean distances within supported square
    dist = np.sqrt(dist[0]**2 + dist[1]**2)
    # mask defining the  radial support of the PSF
    mask = dist <= maxPsfRange
    # calc exponents in gaussianblur
    weights = np.exp(- dist /
      (2 * magFactor**2 * psfWidth**2))
    weights[mask != True] = 0
    # normalize
    weights = weights/np.sum(weights)
    # Avoid nans
    weights[np.isnan(weights)] = 0
    idx_u = i_x + i_y*M2
    # Every row in W corresponds to a u in y,
    # put corresponding weights at the right column
    idx_v = np.meshgrid(vX[maskX], vY[maskY]*N2)
    idx_v = np.ravel(idx_v[0] + idx_v[1])
    W[idx_u, idx_v] = np.ravel(weights)
return W
```

Code Listing A.2: The generation of the image degradation matrix. Imple-

B XSIM EXTENDED NOTES

The following sections provides an elaborating of some of the concepts used in the development of Xsim.

B.1 CONTROLLABLE PARAMETERS

The following is a overview of the different user-defined variables used for Xsim. All distance and sizes are given in [cm].

d1: *(float)* Distance from point source to bottom of the field of view (FOV).

d2: *(float)* Distance from bottom of the FOV to the camera.

voxelSizeD: *(float, float, float)* The physical size of one element in the D and FOV array

pixelArea: *(float, float)* The physical area of one pixel in the charge-coupled device (CCD) array.

gridCCD: *(int, int)* The camera resolution i. e., the amount of pixels.

Nsubpixels *(int)* The amount of different rays hitting one pixel. Goes as $(Nsubpixelsh)^2$, so if $Nsubpixels = 3$ - 9 subpixels are used in the summation of a single pixel

photonsRay: *(int)* The amount of photons emitted in each ray path.

saturation: *(int)* The photon count setting the saturation level of a pixel.

eta: *(float)* Quantum efficiency $0 \leq \eta \leq 1$ the percentage of photons registered by the camera.

spectrum: *(2D array)* X-ray source spectrum given probability distribution. First row energy [KeV]. Second row probability.

in:Estep:Emax): *(int, int, int)* The energy range for the sampled spectrum [KeV].

Nframes: *(int)* The amount of images to generate.

Nsubframes: *(int)* The amount of subframes contributing to a single frame.

moveStep: *(int, int)* The translation of D in the FOV per subframe.

moveStart: *(int, int)* The placement of D in the first frame.

B.2 BRESENHAMS LINE ALGORITHM

Bresenhams line algorithm is used for drawing lines in an N-dimensional raster grid between two points, returning a list of integer indices creating the line in the grid. In a 2D grid case we go from point $P_{start}(x_1, y_1)$ to $P_{end}(x_2, y_2)$. We start by limiting the system to the positive direction i.e. $x_1 < x_2$ and $y_1 < y_2$. Furthermore the slope of the linen is defined to be in the range $0 \leq m \leq 1$, which is found by:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta_y}{\Delta_x}$$

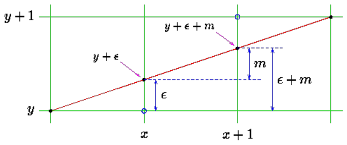


Figure B.1: The slope error for Bresenhams line algorithm. The error is found with respect to the integer values and the true values of the red line. Image from [50]

The algorithm works iteratively by going stepwise from point to point on the line, through a driving axis. At the step k the point$P_k(x_k, y_k)$ is found. If x is chosen as the driving axis the point at step $k+1$ is either $P_{k+1}(x_k + 1, y_k)$ or $P_{k+1}(x_k + 1, y_k + 1)$. This choice is it determined with respect to a slope error ϵ between the integer value of y_k and the "true" value of the line point which should be $y_k + \epsilon_k$. The error is illustrated in Fig. B.1 For every step the "true" value of the line point is increased by m and if this is above 0.5 the integer value of y is incremented. So the next point is given as:

$$P_{k+1} = \begin{cases} (x_k, y_k) & \text{if } y_k + \epsilon_k + m < y_k + 0.5 \\ (x_k, y_k + 1) & \text{otherwise.} \end{cases} \tag{B.1}$$

The error is then changed with respect to the choice of the next point

$$\epsilon_k + 1 = \begin{cases} (y_k + \epsilon_k + m) - y_k & \text{if } P_{k+1} = (x_k + 1, y_k) \\ (y_k + \epsilon_k + m) - (y_k + 1) & \text{if } P_{k+1} = (x_k + 1, y_k + 1). \end{cases}$$

To avoid floating points in $m = \Delta_y/\Delta_x$ the different expressions are multiplied by Δ_x. The condition in (B.1) is further multiplied by 2 and simplified by removing y_k. This gives

$$P_{k+1} = \begin{cases} (x_k, y_k) & \text{if } 2\epsilon_k\Delta_x + \Delta_y < \Delta_x \\ (x_k, y_k + 1) & \text{otherwise.} \end{cases}$$

The error is then changed with respect to the choice of the next point

$$\epsilon_k + 1 = \begin{cases} \epsilon_k + \Delta_y & \text{if } P_{k+1} = (x_k + 1, y_k) \\ \epsilon_k + \Delta_y - \Delta_x & \text{if } P_{k+1} = (x_k + 1, y_k + 1). \end{cases}$$

Which should be calculated for each iteration[50].

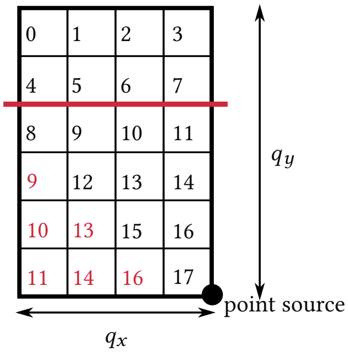
```python
def Bresenham3D(x1, y1, x2, y2, fov_height):
  dx = abs(x2 - x1)
  dy = abs(y2 - y1)
  # goes from top to bottom of in z
  dz = fov_height
  z1 = 0
  ListOfPoints = np.zeros((2, dz))
  # Positive or negative slope
  if (x2 > x1):
    xs = 1
  else:
    xs = -1
  if (y2 > y1):
    ys = 1
  else:
    ys = -1

  # Driving axis is Z-axis"
  eps_x = 2 * dx - dz
  eps_y = 2 * dy - dz
  # while (z1 != z2):
  for i in range(dz):
    z1 += 1
    if (eps_y >= 0):
      y1 += ys
      eps_y -= 2 * dz
    if (eps_x >= 0):
      x1 += xs
      eps_x -= 2 * dz
    eps_y += 2 * dy
    eps_x += 2 * dx
    ListOfPoints[:, i] = (x1, y1)
  return ListOfPoints
```

Code Listing B.1: Bresenhams line drawing algorithm with driving axis in z. Inspired by [**Bresenham**]

In using Bresenhams line algorithm for Xsim the above is expanded to a 3-Dimensional grid. This is done by choosing z as a driving axis and finding the slope errors with respect to the x and y direction. In the cases of a negative slope, the incrementation is set to be -1 if the endpoint is smaller than the start point for that axis.[51]

The python function for evaluating Bresenhams algorithm in Xsim is given in Code Listing 5.5.

B.3 ITERATION OPTIMIZATION IN THE FOV



0	1	2	3
4	5	6	7
8	9	10	11
9	12	13	14
10	13	15	16
11	14	16	17

Figure B.2: Top viev of one quadrant of the CCD array. Illustration the equivalent ray paths as colored indices. The quadrant sides are $q_x = 6$ and $q_y = 4$ and the red line indicates the two terms in the derivation.

This section provides an elaboration of the optimization scheme used in reducing the amount of iterations it the preprocessing of the ray paths through FOV. As stated in § 5.3 the system is symmetrical in the x and y axis and the ray paths can be split into 4 equal quadrants when related to the CCD array of size $N_x^{CCD} \times N_y^{CCD}$. For simplification of the equations, we denote the size of a quadrant in the CCD array as $q_x \times q_y$ i.e $q_x = N_x^{CCD}/4$. The simplified system is shown in Fig. B.2, where the indices in black is sufficient for describing all unique paths. The number of unique paths can be described by the following

$$N_{paths} = \underbrace{q_x(1 + q_y - q_x)}_{above\ line} + \underbrace{\frac{q_x(q_x - 1)}{2}}_{under\ line}, \text{ for } q_y > q_x.$$

The two terms are related to the indices above and under the red line in Fig. B.2. The equation can be simplified further:

$$\begin{aligned} N_{paths} &= q_x \frac{1 + 2q_y - q_x}{2} \\ &= q_x q_y + \frac{q_x - q_x^2}{2}, \text{ for } q_y > q_x. \end{aligned}$$

B.4 PROJECTION OF OBJECT

In § 5.1.2 a simplified description of the projection of objects denoted D onto the FOV is given. As explained the two are defined as 3-dimensional arrays consisting of volume blocks. The system is defined such that the volume-blocks of the FOV and the D is of the same size and cubic, such that geometrical considerations holds for the projections. Furthermore, the amount of blocks in the z direction is for the FOV defined to be the same size as of the object i.e $N_z^{FOV} = N_z^D$ giving a one to one projection in the z direction. For the x and y plane parallel with the camera, the same considerations holds for both directions and the further descriptions is only given for the x direction. For projection in the x-direction there are three relevant variables the grid sizes N_x^D, N_x^{FOV} and the desired translation from the center x_{trans}. The projection is dependent on whether the D or the FOV is smallest as it defines the maximum width of the broadcasting B_x i.e. the relevant part of D for projection. The smallest and largest of the two is defined as $x_{min} = min(N_x^D, N_x^{FOV}$ and $x_{max} = max(N_x^D, N_x^{FOV}$p The broadcasting is further described by the translation, as the object might be out of scope for the FOV. Giving $B_x = min(x_{min}, x_{min}/2 - (|x_{trans}| - x_{max}/2)$. The projection is further defined by the difference in grid size as $\Delta_x = |N_x^D - N_x^{FOV}$. added with the translation $x_{proj} = \Delta_x + x_{trans}$.

With the different variables defined the projection can be done with respect to four different scenarios.
Positive translation with $N_x^D > N_x^{FOV}$

$$FOV[0 : B_x] = D[x_{proj} : x_{proj} + B_x]$$

Positive translation with $N_x^D < N_x^{FOV}$

$$FOV[x_{proj} : x_{proj} + B_x] = D[0 : B_x]$$

Negative translation with $N_1^D > N_1^{FOV}$ $ind_D = min(x_{proj}, 0)$

$$FOV[N_x^{FOV} - B_x : N_x^{FOV}] = D[ind_D : ind_D + B_x]$$

Negative translation with $N_1^D < N_1^{FOV}$ $ind_D = min(x_{proj}, 0)$

$$FOV[ind_D : ind_D + B_x] = D[N_x^D - B_x : N_x^D]$$

```
def projectInX(Nx_FOV, Nx_D, xTrans, D):
  min_x = min(Nx_FOV, Nx_D)
  max_x = max(Nx_FOV, Nx_D)
  delta_x = (max_x - min_x)/2 + xTrans
  xB = min(min_x, min_x/2 - (abs(xTrans)-max_x/2))
  xProj = delta_x + xTrans
  if xTrans >= 0:
    if Nx_FOV <= Nx_D:
      FOV[0:xB] = D[xProj:xProj+xB]
    elif Nx_FOV > Nx_D:
      FOV[xProj:xProj+xB] = D[0:xB]
  if xTrans < 0:
    # Guarantees that xProj isn't negative
    xProj = min(0, xProj)
    if Nx_FOV <= Nx_D:
      FOV[Nx_FOV-xB:Nx_FOV] = D[xProj:xProj+xB]
    elif Nx_FOV > Nx_D:
      FOV[xProj:xProj+xB] = D[Nx_D - xB:Nx_D]
```

Code Listing B.2: Example of object projection in x-direction.

BIBLIOGRAPHY

[1] Thomas Kohler et al. "Robust Multiframe Super-Resolution Employing Iteratively Re-Weighted Minimization." en. In: *IEEE Transactions on Computational Imaging* 2.1 (Mar. 2016), pp. 42–58. ISSN: 2333-9403, 2334-0118. DOI: 10.1109/TCI.2016.2516909 (Cited on pp. ii, 2, 4, 24, 25, 30, 33, 34, 72, 81).

[2] Roman Zeyde, Michael Elad, and Matan Protter. *On Single Image Scale-up Using Sparse-Representations," Curves and Surfaces.* 2012 (Cited on pp. 2, 25).

[3] *Image Super Resolution.* https://awesomeopensource.com/project/idealo/image-super-resolution (Cited on pp. 2, 81).

[4] Thomas Köhler et al. "Toward Bridging the Simulated-to-Real Gap: Benchmarking Super-Resolution on Real Data." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (May 2019), pp. 1–1. DOI: 10.1109/TPAMI.2019.2917037 (Cited on pp. 3, 25, 27, 72, 82).

[5] *All about the New ML Super Resolution Feature in Pixelmator Pro - Pixelmator Blog.* https://www.pixelmator.com/blog/2019/12/17/all-about-the-new-ml-super-resolution-feature-in-pixelmator-pro/ (Cited on pp. 3, 82, 83).

[6] *Gigapixel AI – Topaz Labs.* https://topazlabs.com/gigapixel-ai/ (Cited on pp. 3, 82).

[7] Thomas Köhler. "Multi-Frame Super-Resolution Reconstruction with Applications to Medical Imaging." In: *arXiv:1812.09375 [cs]* (Dec. 2018). arXiv: 1812.09375 [cs] (Cited on pp. 3, 14–16, 24, 28–30, 32, 34, 36).

[8] Yuhua Chen et al. "Efficient and Accurate MRI Super-Resolution Using a Generative Adversarial Network and 3D Multi-Level Densely Connected Network." In: *arXiv:1803.01417 [cs, eess]* (June 2018). arXiv: 1803.01417 [cs, eess] (Cited on p. 3).

[9] Andreas Maier et al., eds. *Medical Imaging Systems: An Introductory Guide.* en. Vol. 11111. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-96519-2 978-3-319-96520-8. DOI: 10.1007/978-3-319-96520-8 (Cited on pp. 7, 8, 10, 18, 19, 40, 49, 66).

[10] Bernd Jähne. *Digital Image Processing.* en. 6th rev. and ext. ed. Berlin ; New York: Springer, 2005. ISBN: 978-3-540-24035-8 (Cited on pp. 9, 11, 13, 14).

[11] J. Fessler. *EECS 516 Lecture Notes.* https://web.eecs.umich.edu/~fessler/course/516/l/ (Cited on pp. 13, 44, 46, 48).

[12] John Bechhoefer. "Feedback for Physicists: A Tutorial Essay on Control." In: *Reviews of Modern Physics* 77.3 (Aug. 2005), pp. 783–836. doi: 10.1103/RevModPhys.77.783 (Cited on p. 15).

[13] *OpenCV.* https://opencv.org/ (Cited on p. 17).

[14] Sylvain Paris et al. "Bilateral Filtering: Theory and Applications." en. In: *Foundations and Trends® in Computer Graphics and Vision* 4.1 (2008), pp. 1–75. issn: 1572-2740, 1572-2759. doi: 10.1561/0600000020 (Cited on p. 21).

[15] *Introduction to Motion Estimation with Optical Flow.* https://nanonets.com/blog/optical-flow/ (Cited on p. 22).

[16] Subhasis Chaudhuri, Rajbabu Velmurugan, and Renu Rameshan. *Blind Image Deconvolution: Methods and Convergence.* en. Springer, Oct. 2014. isbn: 978-3-319-10486-7 (Cited on p. 24).

[17] Daniel Glasner, Shai Bagon, and Michal Irani. "Super-Resolution from a Single Image." en. In: *2009 IEEE 12th International Conference on Computer Vision.* Kyoto: IEEE, Sept. 2009, pp. 349–356. isbn: 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459271 (Cited on p. 25).

[18] Assaf Shocher, Nadav Cohen, and Michal Irani. ""Zero-Shot" Super-Resolution Using Deep Internal Learning." In: *arXiv:1712.06087 [cs, eess]* (Dec. 2017). arXiv: 1712.06087 `[cs, eess]` (Cited on p. 25).

[19] Tomer Michaeli and Michal Irani. "Nonparametric Blind Super-Resolution." en. In: *2013 IEEE International Conference on Computer Vision.* Sydney, Australia: IEEE, Dec. 2013, pp. 945–952. isbn: 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.121 (Cited on pp. 25, 90).

[20] Michal Irani and Shmuel Peleg. "Improving Resolution by Image Registration." en. In: *CVGIP: Graphical Models and Image Processing* 53.3 (May 1991), pp. 231–239. issn: 1049-9652. doi: 10.1016/1049-9652(91)90045-L (Cited on p. 25).

[21] Z. Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity." en. In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. issn: 1057-7149. doi: 10.1109/TIP.2003.819861 (Cited on p. 27).

[22] Jonathan R Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.* Technical Report. USA: Carnegie Mellon University, 1994 (Cited on p. 34).

[23] *SciPy — SciPy v1.5.2 Reference Guide.* https://docs.scipy.org/doc/scipy/reference/index.html (Cited on p. 34).

[24] Peter J. Rousseeuw and Christophe Croux. "Alternatives to the Median Absolute Deviation." In: *Journal of the American Statistical Association* 88.424 (Dec. 1993), pp. 1273–1283. ISSN: 0162-1459. DOI: 10.1080/01621459.1993.10476408 (Cited on p. 35).

[25] Martin Berger, Qiao Yang, and Andreas Maier. "X-Ray Imaging." en. In: *Medical Imaging Systems: An Introductory Guide*. Ed. by Andreas Maier et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 119–145. ISBN: 978-3-319-96520-8. DOI: 10.1007/978-3-319-96520-8_7 (Cited on pp. 40, 41, 44, 45).

[26] Ervin B. Podgorsak. *Radiation Physics for Medical Physicists*. en. Second. Biological and Medical Physics, Biomedical Engineering. Berlin Heidelberg: Springer-Verlag, 2010. ISBN: 978-3-642-00875-7. DOI: 10.1007/978-3-642-00875-7 (Cited on pp. 41–45).

[27] Nadine Barrie Smith and Andrew Webb. *Introduction to Medical Imaging: Physics, Engineering and Clinical Applications*. Cambridge Texts in Biomedical Engineering. Cambridge: Cambridge University Press, 2010. ISBN: 978-0-521-19065-7. DOI: 10.1017/CBO9780511760976 (Cited on pp. 41, 43).

[28] Jerrold T. Bushberg and John M. Boone. *The Essential Physics of Medical Imaging*. en. Lippincott Williams & Wilkins, Dec. 2011. ISBN: 978-0-7817-8057-5 (Cited on p. 42).

[29] *SpekCalc*. en. http://spekcalc.weebly.com/ (Cited on p. 42).

[30] curtis.suplee@nist.gov. *X-Ray Mass Attenuation Coefficients*. en. https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients. Text. Sept. 2009 (Cited on pp. 44, 64).

[31] curtis.suplee@nist.gov. *XCOM: Photon Cross Sections Database*. en. https://www.nist.gov/pml/xcom-photon-cross-sections-database. Text. Sept. 2009 (Cited on pp. 46, 57, 64).

[32] Peter Harremoes, Oliver Johnson, and Ioannis Kontoyiannis. "Thinning and the Law of Small Numbers." en. In: *2007 IEEE International Symposium on Information Theory*. Nice: IEEE, June 2007, pp. 1491–1495. ISBN: 978-1-4244-1397-3. DOI: 10.1109/ISIT.2007.4557433 (Cited on p. 48).

[33] Simon Nyrup. *SimonNy/Xsim*. Aug. 2020 (Cited on p. 51).

[34] Dr Krishna Jagannathan and R Ravi Kiran. "Lecture 15: Sums of Random Variables." en. In: (), p. 5 (Cited on p. 59).

[35] *Binvox 3D Mesh Voxelizer, Keywords: Voxelization, Voxelisation, 3D Model*. https://www.patrickmin.com/binvox/ (Cited on p. 62).

[36] C. A. Brautlecht and A. S. Getchell. "The Chemical Composition of White Potatoes." en. In: *American Potato Journal* 28.3 (Mar. 1951),

pp. 531–550. ISSN: 0003-0589, 1874-9380. DOI: 10.1007/BF02850251 (Cited on p. 64).

[37] *Obj Free 3D Models - .Obj Download - Free3D.* https://free3d.com/3d-models/obj (Cited on p. 64).

[38] Elena Gallio et al. "A GPU Simulation Tool for Training and Optimisation in 2D Digital X-Ray Imaging." en. In: *PLOS ONE* 10.11 (June 2015), e0141497. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0141497 (Cited on p. 65).

[39] *Forewords | GATE.* http://www.opengatecollaboration.org/ (Cited on pp. 65, 67).

[40] *gVirtualXRay: Gate vs gVirtualXRay, Point Source Case.* http://gvirtualxray.sourceforge.net/06/gate_vs_gvirtualxray_point_source.php (Cited on pp. 65, 66).

[41] *Smallpt: Global Illumination in 99 Lines of C++.* http://www.kevinbeason.com/smallpt/ (Cited on p. 66).

[42] Philippe Duvauchelle et al. "A Computer Code to Simulate X-Ray Imaging Techniques." en. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 170.1 (Sept. 2000), pp. 245–258. ISSN: 0168-583X. DOI: 10.1016/S0168-583X(00)00185-3 (Cited on p. 66).

[43] Marco Bontempi, Matteo Bettuzzi, and Andrea Visani. "Semiempirical Simulation of X-Ray Detectors for Imaging Applications." en. In: *Medical Physics* 39.12 (2012), pp. 7677–7685. ISSN: 2473-4209. DOI: 10.1118/1.4767768 (Cited on p. 67).

[44] Franck P. Vidal and Pierre-Frédéric Villard. "Development and Validation of Real-Time Simulation of X-Ray Imaging with Respiratory Motion." en. In: *Computerized Medical Imaging and Graphics* 49 (Apr. 2016), pp. 1–15. ISSN: 08956111. DOI: 10.1016/j.compmedimag.2015.12.002 (Cited on p. 68).

[45] Linwei Yue et al. "Image Super-Resolution: The Techniques, Applications, and Future." en. In: *Signal Processing* 128 (Nov. 2016), pp. 389–408. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2016.05.002 (Cited on p. 81).

[46] Luke Jaffe, Shiv Sundram, and Christian Martinez-Nieves. "Super-Resolution to Improve Classification Accuracy of Low-Resolution Images." en. In: (), p. 10 (Cited on p. 82).

[47] Dingding Cai et al. "Convolutional Low-Resolution Fine-Grained Classification." In: *arXiv:1703.05393 [cs]* (Oct. 2017). arXiv: 1703.05393 [cs] (Cited on p. 82).

[48] Anirudh Shenoy. *How Are Convolutions Actually Performed Under the Hood?* en. https://towardsdatascience.com/how-are-convolutions-

actually-performed-under-the-hood-226523ce7fbf. Dec. 2019 (Cited on p. 82).

[49] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. "Blind Super-Resolution Kernel Estimation Using an Internal-GAN." In: *arXiv:1909.06581 [cs]* (Jan. 2020). arXiv: 1909.06581 [cs] (Cited on p. 91).

[50] *The Bresenham Line-Drawing Algorithm.* https://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html (Cited on p. 96).

[51] *Bresenham's Algorithm for 3-D Line Drawing - GeeksforGeeks.* https://www.geeksforgeeks.org/bresenhams-algorithm-for-3-d-line-drawing/ (Cited on p. 97).

COLOPHON

This document was typeset using the custom $\LaTeX\ 2_{\varepsilon}$ document class `dionsthesis`, developed by Dion Haofner. The document class is based on `uiothesis` developed by Eivind Uggedal. It uses Minion Pro, developed at Adobe Systems, and Fira Sans, developed by the Mozilla Foundation, as body fonts. `dionsthesis` is available at:

https://github.com/dionhaefner/dionsthesis/

The style of `uiothesis` was inspired by Robert Bringhurst's seminal book on typography The Elements of Typographic Style. Typographic, structural and graphical decisions in this document follow the ideas presented in Jean-Luc Doumont's book Trees, Maps, and Theorems.

DECLARATION

I declare that this thesis is my own personal effort. I have not already obtained a degree on the basis of this work. Furthermore, I took reasonable care to ensure that the work is original, and, to the best of my knowledge, does not breach copyright law, and has not been taken from other sources except where such work has been cited and acknowledged within the text.

Copenhagen, September 30, 2020

Simon Nyrup