



TECHNISCHE
UNIVERSITÄT
WIEN



MEDIZINISCHE
UNIVERSITÄT WIEN

DIPLOMARBEIT

Optimizing Functional PET Image Reconstruction

Towards High-Temporal Resolution on Long-Axial Field of View PET Systems

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Physikalische Energie- und Messtechnik

eingereicht von

Florian Dellekart

Matrikelnummer: 51841316

ausgeführt am

Atominstitut der Fakultät für Physik der Technischen Universität Wien

und an den

Neuroimaging Labs, Universitätsklinik für Psychiatrie und Psychotherapie der Medizinischen Universität Wien

Betreuung:

Assistant Prof. DI Dr.techn. Albert Hirtl (TU Wien / Atominstitut)

Dr. scient. med. Murray Bruce Reed, MSc (MedUni Wien / Neuroimaging Labs)

Wien, 6. März 2025

(Unterschrift Verfasserin)

(Unterschrift Betreuer)

Acknowledgements

This thesis was completed at the Neuroimaging Labs, a research group based at the Department of Psychiatry and Psychotherapy, Medical University of Vienna under the lead of Prof. Rupert Lanzenberger.

My special appreciation goes to Murray Reed, who was always available without hesitation to provide answers and clarifications. His support was responsible for that all obstacles encountered during working on this thesis turned into great learning experiences. Besides all other team members of the Neuroimaging Labs, I would especially like to thank Prof. Andreas Hahn, Christian Milz, Magdalena Ponce de León, Pia Falb, Matej Murgaš and Giulia Vallini for their support, interesting discussions and for making the months at the lab a fun and enriching experience. Furthermore, I want to thank Prof. Lanzenberger for the opportunity and his support.

Special thanks go to Prof. Albert Hirtl for enabling a cooperation between technical and medical universities. I am deeply convinced that this format is one of the most valuable ways to conduct a thesis.

Additionally, I want to express my deep gratitude to all friends and colleagues who I was allowed to share the last years with. Although all of you would without doubt deserve it, the list of inspiring people would be too long to mention everybody personally.

Finally, I want to thank my family and my girlfriend, for always supporting me in my goals but also for providing the environment to disconnect when it was necessary.

Abstract

Functional Positron Emission Tomography (fPET) is a medical imaging method allowing for the assessment of task specific changes of molecular processes. Via frame by frame reconstruction, also referred to as dynamic reconstruction, molecular processes inside the human brain or other organs can be observed. It is expected that temporal resolutions of one second and below will improve our understanding of these molecular processes. However, reducing the frame lengths also reduces the signal per frame. Therefore, scanners with a higher sensitivity must be used for an accurate detection of fast molecular processes. Long axial field of view (LAFOV) scanners have many detector rings and can therefore capture emissions which would remain undetected in other systems. Using a system of this type is one, compared to others, readily available possibility to increase the acquisition sensitivity.

The computational complexity of the reconstruction mainly depends on the number of detectors. Furthermore, in the case of dynamic PET, the reconstruction algorithm is executed once per frame. Therefore, the processing of data from LAFOV systems requires very performant hard- and software. In clinical practice, software from scanner manufacturers is used. Additionally, other openly available implementations do exist for research. However, the literature does not provide thorough comparisons. To alleviate the uncertainties in selecting a toolbox, this thesis conducted a detailed review of available solutions in the context of fPET. A reconstruction pipeline, using data acquired by a Siemens Biograph mMR, was implemented with two open toolboxes, NiftyPET and SIRF. They were evaluated in terms of processing time and hardware utilization. Furthermore, their performance was compared to the Siemens solution, e7-tools.

It was found that the e7-tools are the best option amongst the tested candidates, being more than ten times faster compared to the openly available solutions. Generally, they are highly optimized. It will therefore remain challenging to compete with their performance. Especially, when calculations are executed on the GPU, processing is significantly faster. However, this is only the case when appropriate hardware is used. Furthermore, GPU processing is unfortunately not supported for all scanner models. Especially for LAFOV systems such as the Siemens Vision Quadra, improving the e7-tools to also support GPU accelerated reconstruction for this scanner type would therefore be highly favorable. Additionally, it was found that certain design decisions in histogramming and scatter simulation can lead to increased processing times which are further amplified in the case of fPET.

To obtain a better understanding of the challenges of dynamic reconstruction on LAFOV systems, the general dependency of processing requirements on the number of detector rings was evaluated. It was found that the number of computations grows with the fourth power of the number of rings in the scanner. This illustrates the challenge which comes with a transition to larger scanner systems. The effect could be mitigated using a restricted field of view (FOV). Therefore, substantial potential for performance improvements are still possible for fPET reconstruction by a combination of GPU processing with a limitation of the FOV.

Zusammenfassung

Funktionelle Positronen Emissions Tomografie (fPET) ist ein medizinisches Bildgebungsverfahren zur Untersuchung aufgabeninduzierter Änderungen molekularer Prozesse. Über die Rekonstruktion einer Bildsequenz, auch dynamische Rekonstruktion genannt, können Prozesse im Gehirn untersucht werden. Es wird erwartet, dass zeitliche Auflösungen von einem Bild pro Sekunde und darunter das Verständnis dieser Prozesse verbessern werden. Kürzere Aufnahmedauern pro Bild verringern jedoch die Signalstärke. Daher sollten zur Abbildung schneller molekularer Prozesse hochsensitive Systeme verwendet werden. Long axial field of view (LAFOV) Scanner bestehen aus einer großen Anzahl an Detektorringen und sind daher imstande Signale zu messen, welche das System ansonsten undetektiert verlassen würden.

Da der, zur Rekonstruktion der Bilder aus den Emissionsdaten, nötige Rechenaufwand stark von der Anzahl der Detektoren abhängt, erhöht sich dieser durch die Verwendung von LAFOV Systemen signifikant. Zusätzlich besteht bei dynamischer Rekonstruktion die Notwendigkeit, eine Vielzahl von Einzelbildern zu rekonstruieren. Daher, und insbesondere für hohe zeitliche Auflösungen, bedarf fPET an LAFOV Systemen äußerst performanter Hard- und Software, um die Bilder in akzeptablen Zeiträumen errechnen zu können. In der klinischen Praxis werden meist von Scannerherstellern zur Verfügung gestellte Programme verwendet. Zusätzlich gibt es frei verfügbare Lösungen, welche von Wissenschaftlern entwickelt und öffentlich zugänglich gemacht wurden. Durch das Fehlen von Vergleichen in der Literatur gestaltet sich eine Auswahl im Kontext der dynamischen Rekonstruktion für fPET schwierig. Um diese Unsicherheiten zu verringern wurde im Zuge dieser Arbeit eine detaillierte Recherche der verfügbaren Optionen durchgeführt. Mit einem Siemens Biograph mMR Scanner aufgenommene Daten wurden mithilfe der zwei vielversprechendsten Programme, NiftyPET und SIRF, rekonstruiert. Die Programme wurden in Hinblick auf Laufzeit und Hardwareauslastung evaluiert und mit der Siemens Lösung, e7-tools, verglichen. Zusätzlich wurden die e7-tools auf verschiedenen Hardwarekonfigurationen getestet.

Die Ergebnisse zeigen, dass die e7-tools die beste getestete Lösung darstellen. Sie konnten die Rekonstruktion der Bildsequenz um teils mehr als das zehnfache schneller ausführen als andere Programme. Besonders bei Ausführung der Berechnungen mithilfe einer GPU waren die e7-tools im Vorteil. Die Verwendung dieser Bedarf jedoch leistungsfähiger Hardware. Außerdem werden GPU-optimierte Berechnungen nicht für alle Scannersysteme unterstützt. Besonders im Fall von LAFOV Systemen wie dem Siemens Biograph Vision Quadra wäre eine Weiterentwicklung der e7-tools hin zur GPU also wünschenswert. Die Gründe für die bessere Leistung wurden untersucht und erläutert. Sie liegen hauptsächlich in fPET-spezifischen Punkten in der Sinogrammerstellung und in Streuungssimulationen, welche zugunsten höherer Genauigkeit auf gewisse Optimierungen verzichten. Zusätzlich sind die e7-tools eindeutig hocheffizient und konkurrenzfähige Software würde einer aus allen Perspektiven optimierten Implementierung bedürfen.

Um die Herausforderungen rund um die Verwendung von LAFOV Systemen gepaart mit dynamischer Rekonstruktion besser zu verstehen, wurde die Abhängigkeit des Rechenaufwands von der Anzahl der Detektorringe untersucht. Dies hat gezeigt, dass die Anzahl der nötigen Berechnungen mit der vierten Potenz der Anzahl der Ringe steigt. Hierdurch wird klar, worin die signifikante Steigerung der Komplexität des Problems begründet liegt. Eine substantielle Reduktion des Aufwands wäre jedoch durch eine Einschränkung des Bildes auf das Gehirn möglich. Allgemein lässt sich daher schlussfolgern, dass Verbesserungen durch eine Kombination aus GPU-Prozessierung und anwendungsspezifischer Rekonstruktion immer noch möglich sind.

Contents

Abstract	iv
Zusammenfassung	vi
1 Introduction	1
2 Physical and Technological Background	3
2.1 Working Principle	3
2.2 Scatter and Attenuation	4
2.3 Data Acquisition	5
3 PET Image Reconstruction	11
3.1 Filtered Back Projection	12
3.2 Maximum Likelihood Expectation Maximization	14
3.3 Ordered Subset Expectation Maximization	16
3.4 3D Reconstruction	17
3.5 Randoms, Scatter and Attenuation Correction	19
4 Materials and Methods	23
4.1 Siemens Biograph mMR	23
4.2 Siemens Biograph Vision Quadra	24
4.3 Execution Environment	24
4.4 Input Data	26
4.5 Reconstruction Toolboxes	27
4.6 Toolbox Comparison	31
5 Results	33
5.1 e7-tools	35
5.2 NiftyPET	39
5.3 SIRF	41
5.4 Image Quality	43
5.5 Computational Implications of LAFOV Reconstruction	46
6 Discussion	49
6.1 Efficient fPET Histogramming	51
6.2 Effects of Image Resolution in Scatter Simulation	51
6.3 Parallelization and Hardware Utilization	52
6.4 Memory Leaks	53
6.5 Disregarded Toolboxes	54
6.6 Image Quality	55
6.7 Future Improvements in fPET Reconstruction	55
7 Conclusion and Outlook	57
References	61

1 Introduction

Positron emission tomography (PET) has evolved to be an important medical imaging modality for clinical and scientific applications. Its working principle is illustrated in Figure 1.1. It builds on the fact that a positron emitted during a β^+ decay, under the usual conditions encountered in tissue, annihilates with a nearby electron. In this process two γ rays traveling in almost exactly opposite directions are emitted. If two photons are detected in opposing detectors in a very narrow timeframe, they are said to coincide. A coincidence event suggests that an annihilation took place on the line connecting the detectors. Thus, acquiring a large number of such coincidences allows to reconstruct the underlying activity distribution.

Physiologically active substances can be marked with isotopes undergoing a β^+ decay. Introduced into the bloodstream, these substances participate in metabolic processes in the same way as their non-radioactive counterpart. In some cases, their metabolites, the products of these processes, accumulate in regions of metabolic activity. The activity distribution measured by PET therefore allows to observe the dynamics of biological processes inside living organisms. One application of this is, for example, the localization of cancerous cells, as they show a high metabolic activity.

The kinetics of those processes can be studied using frame by frame reconstruction, a method referred to as dynamic PET. When combined with a constant tracer infusion it can be used to compare task induced changes to baseline measurements within a single measurement instead of requiring multiple measurements. In this case it is referred to as functional PET (fPET). In neuroscience, fPET can serve as a tool to observe task induced changes in the uptake of glucose inside the brain or the synthesis of certain neurotransmitters. It was first independently explored by Villien *et al.* [1] as well as Hahn *et al.* [2] at the group around the Neuroimaging Labs of the Medical University of Vienna, at which this thesis was executed. Furthermore, it was later successfully applied in studies on the effects of various stimuli on the human brain [2–5].

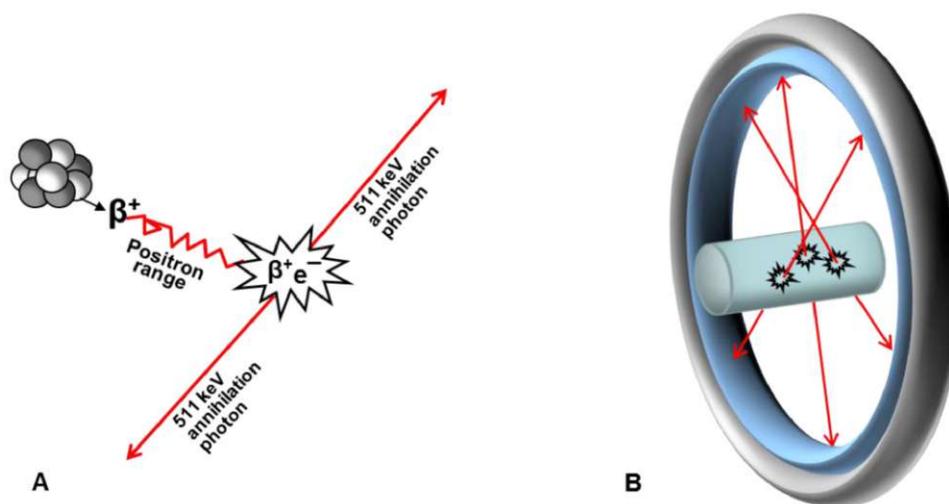


Figure 1.1: Schematic of the primary principle underlying PET imaging from [6]. **A** Shows how a positron emitted by a decay travels a certain range ($\sim 2\text{mm}$) until it annihilates with a nearby electron. **B** Shows how the γ rays traveling in opposite directions are detected to form a coincidence event describing a line of response (LOR).

Continuous efforts are being made towards reducing frame durations. Previously, studies have used increasingly shorter frame durations of 30 [3], 16 [7] and down to 3 [5] seconds. It is expected that single seconds and below will enable further advances of studies, leading to a better understanding of the human brain. Naturally, higher temporal resolution will lead to a reduced number of detected coincidences per frame. As a consequence, the resulting image quality will be reduced. To compensate for this, acquisition systems with higher sensitivities can be used. An example to improve it, can be an increased number of possible detector pairs to measure coincidences. This can be achieved by the use of scanners with a particularly LAFOV, and therefore many detector rings. These systems provide a higher sensitivity due to the fact that a larger share of the signal does not escape the scanner. However, the increased sensitivity comes at the expense of a drastically higher number of possible detector pairs. Thus, the reconstruction of images is increasingly computationally expensive with a growing number of rings.

A variety of software toolboxes exist to reconstruct the underlying activity distribution from the emission data. Clinically, the most used solutions are provided by scanner manufacturers, others were published by researchers and are openly available. Currently, the literature does not provide thorough benchmarks or comparisons of toolboxes. This is especially the case with regard to the high performance requirements necessary for reconstructing fPET images from LAFOV systems. fPET is a recent development and reconstruction toolboxes tend to be designed for clinical applications with low performance requirements. Reconstructing a series of frames with high-temporal resolution is therefore a fairly novel problem for reconstruction software. In general, it is no trivial task to obtain an overview about the options and make a sound decision on the toolbox to use. Selecting a toolbox currently requires to invest a significant amount of time into research and testing. The various toolboxes rely on different computing paradigms and differ in their implementation.

To alleviate the uncertainties, this thesis conducts a thorough overview and comparison of the best options currently available. Furthermore, an outlook towards possible further developments is provided. As a first step, the openly available options are carefully screened and evaluated for compatibility. After identifying the most promising options, a reconstruction pipeline is set up for each of them using list mode data acquired by a Siemens Biograph 3T mMR system. The results are compared to the released research solution e7-tools by Siemens. Finally, the requirements for larger systems are estimated through an analysis of processing complexity of the involved algorithms. The specifics and challenges of high-temporal resolution reconstruction of LAFOV images in the context of fPET are laid out. As an example of a LAFOV system, the Siemens Vision Quadra PET/CT scanner is used.

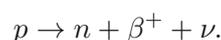
2 Physical and Technological Background

From radiotracer injection to a reconstructed PET image, many physical phenomena, a significant amount of engineering and sophisticated algorithms have to work together to obtain a meaningful result. The following subsections discuss the most important physical phenomena and methods. Where not indicated otherwise, all provided information was extracted from the books on medical imaging and PET by Cherry [8], Hendee [9], Wernick [10] and Bailey [11].

2.1 Working Principle

Positrons are the antimatter counterpart to electrons. While they possess in general equivalent properties, they differ with respect to their charge, which is positive for the positron.

Particular unstable nuclei with excess protons can decay in a process referred to as β^+ decay. During this transmutation of the nucleus a proton decays into a neutron, a positron and a neutrino



Whilst the neutron remains in the nucleus and the neutrino does not interact, the positron is emitted. When traveling through the surrounding matter it loses its kinetic energy through scattering processes with electrons and nuclei. As soon as it has lost almost all of its kinetic energy an annihilation with an electron can occur. Through this process a pair of 511 keV gamma photons is emitted. The travelled distance prior to the annihilation, referred to as the positron range, depends on the initial kinetic energy and therefore on the used radionuclide. It negatively impacts the image quality, typically resulting in a resolution loss of approximately 2 mm.

The fact that the annihilation occurs almost at rest is important as the positrons' kinetic energy determines the angle of the emitted photons. At rest the γ - rays will be emitted collinearly in opposite directions. Therefore, if two photons are detected in opposing detectors within close temporal proximity, it can be concluded that the annihilation likely happened on the line connecting the two detectors. This line is commonly referred to as a line of response (LOR).

A list of radioactive isotopes decaying via a β^+ process and with a half life appropriate for PET is given by Bailey in [11]. To produce a tracer, the isotopes are bound to metabolically active molecules. Although various isotopes can be used to produce tracers for different use cases, Fluorine-18 ($[^{18}F]$) is the predominantly employed isotope with the radiotracer fluorodeoxy-glucose (FDG). $[^{18}F]$ FDG is especially useful because it is trapped within cells when metabolized and therefore accumulates in regions of metabolic activity. This behavior can be beneficial for diagnosis in oncology as well as for measurements of neurological activity.

2.2 Scatter and Attenuation

Given that the gamma rays must pass through the subject tissue, bones and scanner material, many of them are scattered. Scattered emissions lead to incorrect LOR formations and can severely degrade the image quality. Therefore, the ability to appropriately correct for scatter is a prerequisite for high quality image reconstruction. To perform the correction, an accurate physical model of the underlying processes is essential. In general, a variety of effects can lead to photon scattering. With the energy of annihilation photons in combination with the atomic number of elements usually found in tissue, the process of particular relevance is incoherent, also referred to as inelastic, scattering. Although coherent scattering also occurs, its impact on photon energy and direction can be neglected [12]. Similarly, absorption through the photoelectric effect is a secondary process for elements with a mass number <20 as they are relevant for PET.

During an incoherent scatter process, a photon interacts with a loosely bound electron in one of the outer electronic orbitals. A recoil electron is produced, to which a portion of the photons' energy is transferred. The remaining energy of the photon E'_γ depends on the incoming energy E_γ , electron mass m_0 and the scatter angle θ_c . It is described by the Compton equation

$$E'_\gamma = \frac{E_\gamma}{1 + \frac{E_\gamma}{m_0 c^2} (1 - \cos \theta_c)}. \quad (2.1)$$

To obtain an accurate description of the effect of incoherent scattering on a PET measurement, a model for the scatter probabilities and angles is required. This model is given by the Klein-Nishina equation. With the fact that the energy of annihilation photons is equivalent to the mass of an electron, the equation can be written as

$$\frac{d\sigma}{d\Omega} = \left(\frac{1}{2 - \cos \theta_c} \right)^2 \left(1 + \frac{(1 - \cos \theta_c)^2}{(2 - \cos \theta_c)(1 + \cos^2 \theta_c)} \right), \quad (2.2)$$

where $\frac{d\sigma}{d\Omega}$ represents the differential cross-section.

The energy of scattered photons is decreasing with increasing scatter angle, reaching a minimum for direct back-scattering at an angle of 180° . Similarly, the scatter probability is decreasing with higher deflection angles.

Scattered photons do not only cause an increase in signal in undesired places, but their signal contribution is also lost for the detectors which they would have reached without an interaction. Both degrading effects combined lead to a loss in contrast.

The extent to which the gamma radiation is attenuated before reaching the detectors depends on the material it has to pass through. To describe the properties of various materials in this regard, the linear attenuation coefficient μ is used. It describes the relative decrease in beam intensity per unit of penetration depth

$$\frac{\Delta I}{I} = -\mu \times \Delta x, \quad (2.3)$$

where I is the beam intensity and x represents position.

The attenuation coefficient

$$\mu(E) = \sigma(E)n \quad (2.4)$$

is directly related to the cross-section of interaction through the molar density n . Solving Equation 2.3, the intensity of a beam at x_2 with initial intensity $I(x_1)$ is

$$I(x_2) = I(x_1)e^{-\int_{x_1}^{x_2} \mu(x)dx}, \quad (2.5)$$

where the exponential term can be interpreted as the probability that a single particle will pass through the distance between x_1 and x_2 unscattered.

In practice, the three-dimensional map of the attenuation coefficient $\mu(x)$ (μ -map) is obtained through one of the following processes, depending on the availability in the used system:

- **Transmission Scan:** A radioactive source inside the scanner is measured with and without the subject. Through comparison of the measurements a μ -map can be calculated.
- **CT Scan:** CT images are formed through attenuation and therefore represent a direct measurement of the μ -map. However, to account for the difference in photon energies, corrections have to be applied before it can be used for PET.
- **MRI Scan:** MRI images can be used to obtain a μ -map. Due to the different signal characteristics in comparison to other modalities, special scan sequences and algorithms are used to obtain an estimate of the μ -map [13, 14].

2.3 Data Acquisition

After passing through the patient tissue and scanner material, the emitted photons reach the detector where they are converted into a measurable signal. When photons interact with matter they can ionize or excite electrons. In many cases the absorbed energy dissipates as heat. However, in certain materials the recombination or de-excitation of electrons emits visible light. Materials with this property are referred to as scintillators and they come in a variety of forms. Whilst organic scintillators are typically dissolved in a liquid, inorganic ones are solid crystals. Due to constraints regarding sensitivity and handling, especially in the energy range of PET gamma rays, organic scintillators are not commonly adapted for PET detectors.

The scintillating properties of inorganic crystals can be attributed to the electronic structure formed by their crystalline nature. Individual atoms of an inorganic scintillator do therefore not show scintillating behavior and effects can only be observed when the atoms are arranged into a lattice. Furthermore, most materials, including the ones used for scintillators, fully lack scintillation capabilities in their pure form. Some materials, for example sodium-iodide (NaI), do scintillate in their pure form, however, they only emit light at certain conditions such as for example very low temperatures. To enhance or activate the beneficial properties of the materials the majority of available scintillators is impurity activated. This means that the desired properties are enhanced or induced via the addition of impurities. Examples for scintillators relevant for various applications in nuclear medicine with some important parameters are given in Table 2.1.

Table 2.1: Properties of scintillator materials used in nuclear medicine. Extracted from [8]. Impurity elements are given in parentheses.

Property	NaI(Tl)	BGO	LSO(Ce)	GSO(Ce)	CsI(Tl)	<i>LaBr₃(Ce)</i>
Density (g/cm^3)	3.67	7.13	7.40	6.71	4.51	5.3
Effective atomic number	50	73	66	59	54	46
Decay time (nsec)	230	300	40	60	1000	35
Photon yield (per keV)	38	8	20-30	12-15	52	61

Abbreviations: *BGO*, $Bi_3Ge_4O_{12}$; *GSO(Ce)*, $Gd_2SiO_5(Ce)$; *LSO(Ce)*, $Lu_2SiO_5(Ce)$;

Many factors need to be taken into account when selecting an appropriate scintillator. Foremost, the scintillator must be capable of stopping high energy γ photons in the relevant energy window. Additionally, this stopping power must be sufficient for fairly small detector crystals which allow precise positional detection. Photoelectric as well as Compton effects are highly dependent on the mass number of elements, therefore the effective atomic number given in Table 2.1 is an indicator for the stopping power of mixtures of elements.

In addition to a small stopping range, the time between excitation and light emission has to be sufficiently short. This time is referred to as the decay time of the excited state. Firstly, it determines the possible timing precision, which is relevant for PET due to the short time window in which coincidences have to be detected. Secondly, longer decay times limit the rate of events at which the detector is able to operate, due to dead time effects. To detect two consecutive interactions without an overlap in signal the time between them must be at least 2-3 times the decay time. If this requirement is not fulfilled, the dead time of detectors can significantly increase and coincidences will go undetected.

The last parameter given in Table 2.1, the photon yield, is a measure of the efficiency of scintillators and describes the amount of light produced per unit energy of the incoming radiation. In part, it determines the requirements on the readout of the scintillator. In addition, it can be utilized to distinguish between events in terms of their energy as they interact with the detector.

Although the aforementioned scintillator NaI, which has many applications in nuclear medicine, was initially also used for PET, modern scanners commonly rely on materials with a higher effective atomic number such as LSO or BGO, as they provide shorter decay times. To form a detector, the crystals are placed on a readout device. This device is responsible for transforming the light emitted by the crystal into an electrical signal which can be processed by subsequent electronics. Whilst older versions used a single readout per scintillator position, this approach soon became complex and costly due to the physical limitations in producing smaller detectors and the high costs of implementing a one-to-one coupling. Therefore, newer designs increase the positional resolution of the detector by implementing a block design.

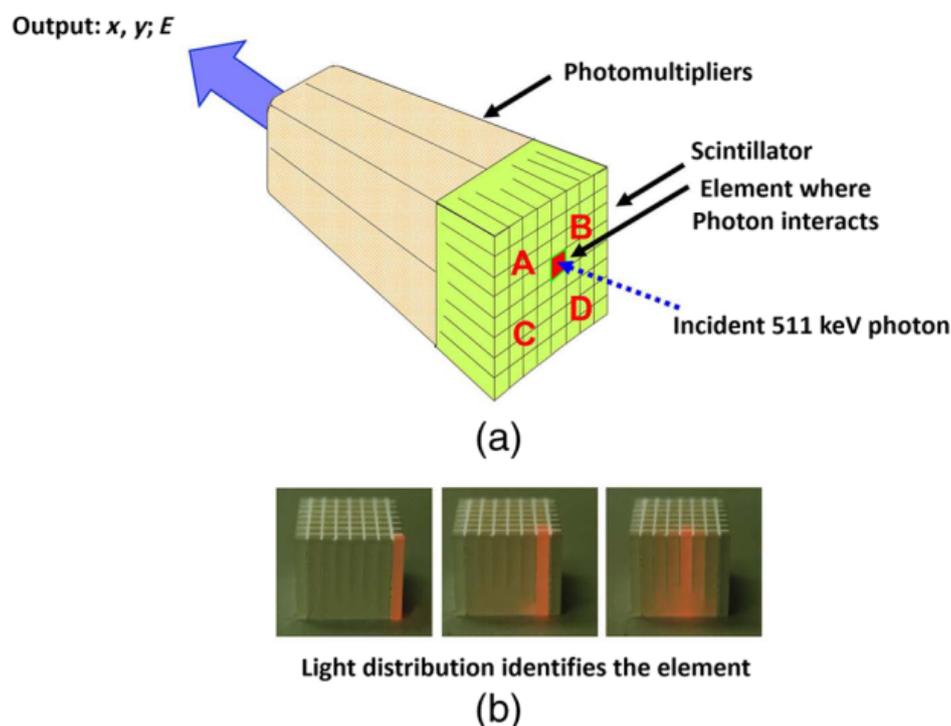


Figure 2.1: Structure and working principle of a block-design PET detector to increase positional resolution with a limited number of readout devices from [15]. (a) shows the structure with four photomultipliers and a scintillating crystal. The crystal is cut with varying depths. This steers the distribution of light and allows to draw conclusions about the position of an interaction as shown in (b) without requiring an individual readout for each position.

A light sharing technology overcomes the problems encountered with a single readout per crystal. In this approach crystals are cut in a rectangular shape to separate the interaction positions on the surface but still allow photons to propagate into neighboring regions further inside the crystal. Through this special shape it is possible to determine the position of an interaction in a, for example, 8x8 grid with just four readout devices, compared to 64 if crystals were coupled one-to-one. The overall structure of the design as well as an illustration of the light distribution within the crystal are shown in Figure 2.1 (a) and (b).

The readout devices on the back of the scintillator are an important part of the detector assembly and their characteristics impact the overall system performance. Its purpose is to convert the very low intensity light emitted by the crystal into an electrical signal of sufficient magnitude to be measured. A commonly used device is the photomultiplier tube (PMT). It amplifies the input by the interaction of photons with a photoemissive substance and the subsequent production of secondary electrons. The secondary electrons are produced by repeated acceleration in an electric field and collision with dynodes. Typically, the electron flux is amplified by a factor of three or six per dynode, which results in a very significant cumulative signal increase by a factor of commonly around 10^7 .

An alternative to PMTs are avalanche photodiodes (APDs). They rely on an operating principle similar to common photodiodes, but the emitted electrons are accelerated by a strong internal electric field to produce secondary charges. Using this method, amplification gains of 10^2 to 10^3 can be achieved. While this represents a significant improvement to conventional photodiodes, the requirements on readout electronics are still significantly higher than for PMTs. Silicon-based readouts are used in applications where the use of PMTs is not feasible due to space limitations, such as in small animal PET, or for combined PET-MRI, as PMTs cannot be operated within a magnetic field.

The charges emitted by the readout are first captured by a capacitor from which the output voltage can be measured. Through a parallel resistor this capacitor is discharged again. The impulse response of this system is a sharp rise followed by an exponential decay with a time constant of around $50 \mu\text{sec}$. As a consequence, for count rates above a certain threshold there will not be enough time for the output voltage to fully decay back to baseline. As a result, the output will be composed of overlapping impulse responses as shown in the top graph of Figure 2.2.

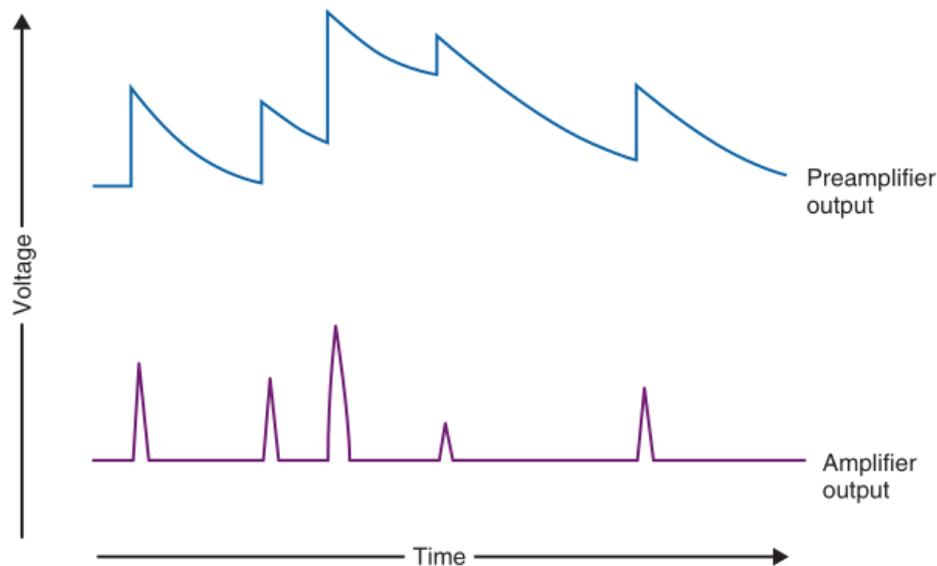


Figure 2.2: Signal in different steps of the output amplification of a radiation detector from [8]. The top graph shows the overlapping impulse responses of individual interactions. The bottom graph shows the amplifier output after pulse shaping.

To further enhance the signal for subsequent processing, the amplified output passes through a pulse shaping circuit which transforms the overlapping impulse responses into individual peaks at the rising edges as shown in the lower graph of Figure 2.2. Up until here the described methods of detection and signal transformation suit a wide range of applications in nuclear medicine. As already described in previous sections the distinctive feature of PET is its use of coincidence events.

Coincidences between pairs of detectors are identified by feeding the pulse-shaped signals into a so-called coincidence unit. In its simplest form, working on just two input signals, this device only outputs a signal if both of its inputs are high within a predefined coincidence window which

is commonly around 10 ns. In a PET scanner the coincidence unit implements this connection for each pair of detectors for which coincidences should be detected. The output of the coincidence unit are individual coincidence events. They are saved for later processing in combination with timing information. This form of storage for emission data is referred to as list mode and serves as the basis for image reconstruction.

3 PET Image Reconstruction

A clear definition of the problem to be solved by image reconstruction is given by Lalush and Wernick in [10] as

"(finding) the object distribution f , given a set of projection measurements g , information (in the form of a matrix H) about the imaging system that produced the measurements, and, possibly, a statistical description of the data and a statistical description of the object."

For simplicity, we will firstly only consider the reconstruction of a slice from the image distribution. The emission data therefore consists of coincidences within one ring of detectors. To reconstruct a volume, this process can be repeated for each slice along the FOV, a method which is known as 2D reconstruction. The details of 3D reconstruction will be discussed in Section 3.4. However, the methods used for it are mostly a generalization of the 2D methods.

We consider an activity distribution to be reconstructed as $f(x, y)$ where x and y are Cartesian coordinates in the imaging plane. Projection measurements are denoted as $g(r, \phi)$ with a view angle ϕ and r as the distance from the scanner center. An illustration of the formation of a projection is illustrated on the left of Figure 3.1.

The set of projections for all angles $0 \leq \phi < \pi$ and positions r is referred to as a sinogram. A single element of a discretized sinogram is referred to as a bin. When performing reconstruction based on list mode data, the coincidence events must first be transformed into a sinogram by histogramming. In this process events are counted for each of the bins and timeframes.

Commonly the formation of projections from the image is described as a linear process where the activity in each voxel contributes an amount of counts to each bin. The counts are proportional to the activity contained in the voxel f_i . The total signal in a bin equals the sum of the contributions of all voxels. This can be expressed as the matrix multiplication

$$g = Hf, \quad (3.1)$$

where H is the system matrix which describes the imaging system. It is of dimension $n_g \times n_f$, where n_g is the number of bins and n_f is the number of voxels in the image. Each element represents the contribution of a unit source in a voxel to a bin.

Due to the statistical nature of emission and acquisition, the actual values of g are realizations of a random process, therefore, the acquisition process is more realistically described as

$$E[g] = Hf, \quad (3.2)$$

where $E[g]$ denotes the expectation values of the projections. The main problem to be solved in image reconstruction can therefore be defined as an inverse problem of H , however, H is commonly not trivially invertible. Analytical solutions do most often rely on a model of the acquisition based on Equation 3.1 as incorporating the statistics is computationally not feasible. The most important analytical approach to note here is filtered backprojection (FBP). To overcome the shortcomings of analytic methods, iterative algorithms based on statistical models of the involved processes were developed. Most applications rely on the ordered subset expectation maximization (OSEM) algorithm, which builds on the maximum likelihood expectation maximization (ML-EM) algorithm but with certain modifications for improved performance.

3.1 Filtered Back Projection

In general, back projection, the inverse operation to forward projection, is a mathematical operation which assigns each voxel in the imaging plane the sum of the projection values of all LORs passing through a given voxel. In mathematical terms this can be described as

$$b(x, y) = \int_0^\pi g(r, \phi) d\phi = \int_0^\pi b^1(x, y; \phi) d\phi, \quad (3.3)$$

where r is $x \cos \phi + y \sin \phi$, b denotes the backprojected image and b^1 is the backprojection of a single projection in Cartesian coordinates. This operation by itself can serve as one of the simplest forms of analytic image reconstruction. However, it is not capable of restoring the true activity distribution due to the fact that the bin values are smeared out across the image.

It can be shown that

$$G(\nu_r, \phi) = F(\nu_x, \nu_y)|_{\nu_r=0}, \quad (3.4)$$

where capital letters denote the Fourier transforms of the respective functions and ν_r is the Fourier conjugate variable of r . This relationship is known as the central section theorem as it states that the Fourier transform of a projection equals a section through the center of the Fourier transformed image at the equivalent angle ϕ . An illustration of the theorem is given in Figure 3.1. It is of central importance to various imaging modalities.

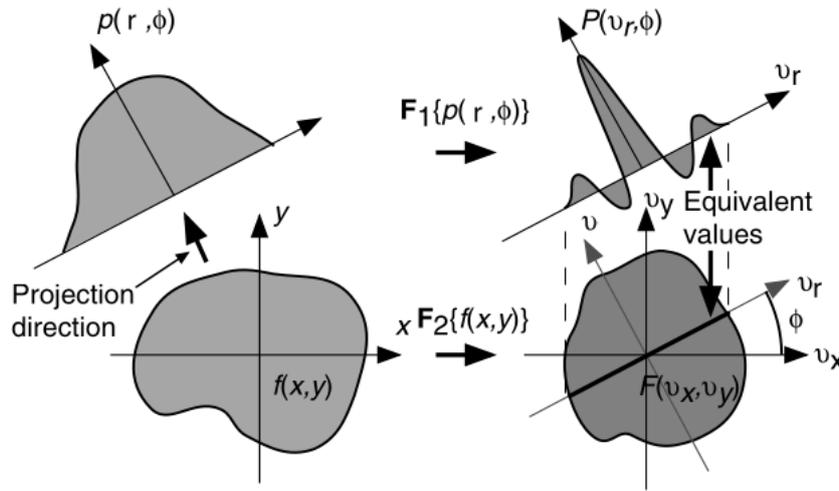


Figure 3.1: Object in image and frequency space (bottom left and right) and a projection formed by parallel line integrals in the projection direction with its one-dimensional Fourier transform (top left and right) from [10]. The central section theorem states that the one-dimensional Fourier transform of a projection equals the section with the same angle through the origin of the two-dimensional Fourier transform of the image. Projection and its Fourier transform are denoted as p and P .

Viewing the central section theorem from the perspective of a single projection, it becomes clear that the Fourier transform of an image with constant values along the projection direction (the backprojection of a projection with one distinct ϕ) is non-zero only on a line through the origin in frequency space

$$B^1(\nu_x, \nu_y; \phi) = G(\nu_r; \phi) \delta(\nu_r; \phi), \quad (3.5)$$

where δ is the delta function, a proof of which is given in [10].

As the Fourier transform of the image can also be obtained by integration of all B^1 over the range of ϕ , it can be shown that lower frequencies will be overrepresented due to oversampling. Visually this can be explained by the fact that the lines representing individual B^1 are further spread out at a greater distance from the center. It can be shown that

$$B(\nu_x, \nu_y) = \frac{F(\nu_x, \nu_y)}{\nu}, \quad (3.6)$$

where $\nu = \sqrt{\nu_x^2 + \nu_y^2}$. This means that the Fourier transform of the back projected sinograms is the Fourier transform of the image divided by the distance to the origin in the frequency domain. It represents an amplification of lower, and attenuation of higher frequencies and mathematically explains the smear caused by assigning a uniform value along the full length of a LOR. As a consequence, simple back projection is not suitable for obtaining quantitatively usable reconstruction results. The blurring caused by the described effects must be compensated for.

The simplest solution to compensate for the $1/\nu$ term in Equation 3.6 is to perform a Fourier transform on the backprojection, multiply the function with $\nu = \sqrt{\nu_x^2 + \nu_y^2}$, and then apply the inverse transform to obtain a filtered image. This method is known as backprojection filtering (BPF) as it applies a ramp filter to the backprojection. This approach requires big matrix sizes for calculations, which makes it computationally inefficient.

A method which is equivalent to BPF is filtered backprojection (FBP). The difference is in the application of the filter, which happens prior to backprojection. This minor change solves the aforementioned computational problems. FBP is therefore the most widely adapted analytic reconstruction algorithm.

The filtered projection which the FBP algorithm operates on is

$$g^F(r, \phi) = F^{-1}\{|\nu_r|F\{g(r, \phi)\}\}, \quad (3.7)$$

where F in this case denotes the Fourier transform. g^F can be considered a pre-corrected version of the projection data which yields the same result as if a ramp filter would have been applied later

Although FBP represents an exact solution of the described inverse problem, and was therefore used as a computationally efficient solution in many applications, it does not in any way account for the statistical nature inherent to the decay and acquisition process. Therefore, images reconstructed with this method have inaccuracies and artifacts. To better resemble the physics behind the involved process, iterative statistical methods were developed and are described in the following sections.

3.2 Maximum Likelihood Expectation Maximization

Iterative algorithms for solving the reconstruction problem can differ in many aspects, but usually consist of a common procedure. As a first step an initial estimate of the image distribution is set up. The estimate is then forward projected and the projection data is compared to the measurement. Projection space errors are calculated and back projected to obtain errors in image space which are used to update the image estimate accordingly. This process is repeated until a satisfactory estimate of the image is obtained. Termination usually happens after a fixed number of iterations. The strength of this strategy lies in the feedback loop provided by the comparison which is not part of other methods such as FBP.

A method to derive a reconstruction algorithm can be to specify a criterion which describes the goodness of fit of a given image estimate and subsequently maximize this value. A popular comparison criterion is the maximum likelihood (ML) criterion. A likelihood is the conditional probability $p(g|f)$ of observations given an estimate of a parameter function. For the given case, the observations are the measured projections $g(r, \phi)$ and the parameter function is the current image estimate $f(x, y)$. Radioactive decay and the detection processes within the acquisition system are known to be statistically described by a Poisson process. The likelihood of the measurement obtained by the PET scanner is therefore

$$p(g|f) = \prod_{i=1}^{n_g} \frac{\bar{g}_i e^{-\bar{g}_i}}{g_i!}, \quad (3.8)$$

where \bar{g}_i is an element of the forward projected image $E[g]$ given by Equation 3.2 and g_i denotes the value of the actual measurement. In terms of the ML criterion, the most probable reconstructed image estimate maximizes the likelihood and reconstruction can therefore be framed as an optimization problem

$$\hat{f} = \arg \max_f p(g|f), \quad (3.9)$$

where \hat{f} denotes the reconstruction result.

An approach to solve various types of ML problems was proposed by Dempster et al. in 1977 as the ML-EM algorithm. It is applicable for estimation problems with unknown parameters independent of their underlying statistical distribution and consists of two steps. In the expectation step the likelihood of the observations depending on the unknown true image distribution is derived and maximized in the second step. It can be shown that this procedure increases the likelihood of the estimate in each iteration. For measurements with the aforementioned statistical properties this leads to

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}} \quad (3.10)$$

as the update rule for the image, where h are elements of the system matrix. In terms of the general procedure of forward projection, comparison, back projection and update, as it was outlined in the beginning of this section, the ML-EM algorithm is illustrated in Figure 3.2.

The benefits of the ML-EM algorithm are that it has a simple form which can be explained with the general model shown in Figure 3.2, and it has well understood convergence behavior. Also, it can be proven that, with a positive initial estimate of the image, all subsequent iterations

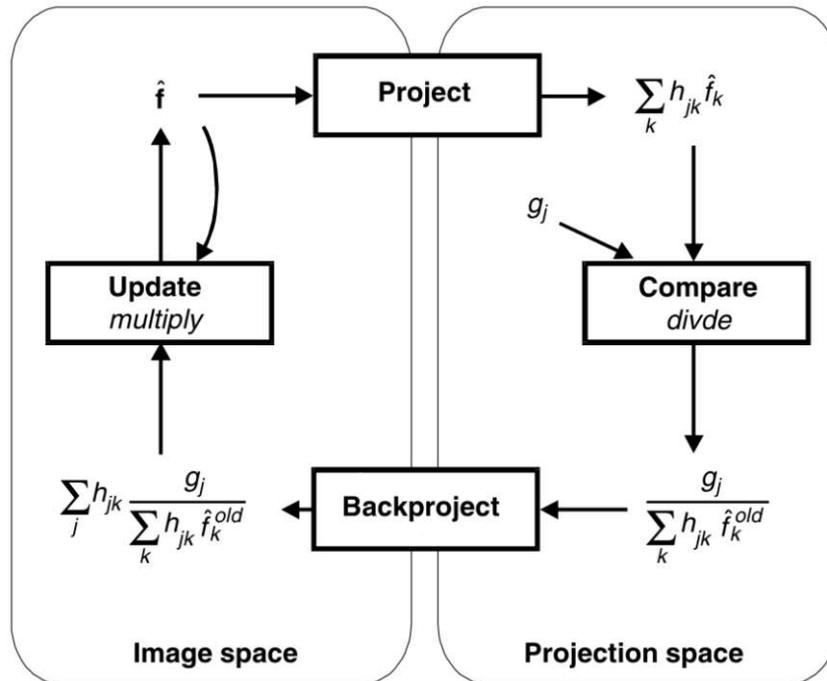


Figure 3.2: The ML-EM algorithm in terms of the general structure in which many iterative algorithms can be described, modified from [10]. For the case of this algorithm the errors are computed by division of the measurements with the projected estimate. The update is performed by multiplication with the backprojected errors.

will yield positive results, which is a convenient way of imposing a non-negativity constraint on the reconstruction [16].

However, it also comes with certain disadvantages. The main drawback is the rather slow convergence. An appropriate reconstruction usually requires around 30-50 iterations. As a single iteration is composed of a forward and a backprojection step, the algorithm usually requires two orders of magnitude more processing time compared to FBP.

Furthermore, the optimal solution in terms of the ML criterion tends to be noisy. Various methods and other algorithms which are out of the scope of this work were presented in the literature to improve on the ML-EM algorithm. A thorough overview of them can be found in [10]. Usually, noise is reduced by prematurely stopping the algorithm or by applying post-reconstruction smoothing.

3.3 Ordered Subset Expectation Maximization

As outlined in the previous section, ML-EM has well understood convergence behavior, however, it requires a high number of iterations to arrive at a result. Therefore, the processing time of the algorithm will be unacceptably high for most applications. An improved algorithm with similar benefits to ML-EM, but more performant, is the ordered subset expectation maximization (OSEM) algorithm. It is the currently most widely adapted algorithm in practice due to its relatively simple form paired with acceptable performance.

It works equivalent to ML-EM, however, instead of using the whole set of projections to update the image, the projections are divided into subsets and each iteration only uses a portion of the data to update the image. Mathematically this can be expressed equivalent to Equation 3.10, with the only difference being that sums over projections are constrained to subsets S_n

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i' \in S_n} h_{i'j}} \sum_{i \in S_n} h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}. \quad (3.11)$$

The subsets are iterated in an ordered manner where one update is referred to as a subiteration and a full pass through the dataset as an iteration. A reduction to a single subset would be equivalent to ML-EM. The performance improvement strongly depends on the choice of subsets, however, as a general rule it can be said that the decrease in processing time is of the order of the number of subsets.

Many strategies to divide the data into subsets can be proposed. Commonly, projection bins are grouped by their angular position around the scanner ring into what is referred to as a projection view. Projections are then organized such that the angular distance within a subset is maximized. For the simple case of four views divided into two subsets (a) and (b) the views in the two subsets are shown in Figure 3.3. The update rule defined by Equation 3.11 is then applied with consecutive subset until the dataset is exhausted. This process is repeated a predefined number of times.

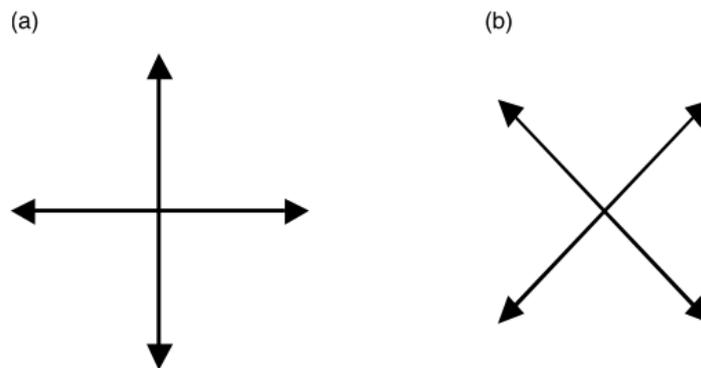


Figure 3.3: Simple example of dividing four views into two subsets with a maximum angular difference between view within a subset from [10].

3.4 3D Reconstruction

To limit their complexity, preceding explanations have focused on a slice by slice reconstruction method of the volume. This can be achieved by only accounting for coincidences within rings and is referred to as 2D reconstruction.

Emitted photons do not show any preference in direction. As a result, coincidences within rings are only a minor fraction and the majority of events is formed between distinct rings. The projections formed by those are referred to as oblique and the information carried by them is inherently redundant. However, used appropriately, the supplementary data can improve the signal-to-noise ratio (SNR) of the result. Algorithms which include them into the reconstruction are referred to as 3D. The most important aspects of the transition from 2D to 3D iterative reconstruction are outlined in the following paragraphs.

The 3D OSEM algorithm is mathematically identical to the 2D case and the update rule from Equation 3.11 is applicable without modification. The only difference is that the system matrix needs to be enlarged to account for the increased number of projections.

The benefits of 3D reconstruction do come at the expense of significantly higher data volumes to store the list mode data, sinograms and system matrix. Additionally, the larger sizes imply longer processing times to perform forward- and backprojection and scatter simulation. Fully 3D reconstruction, individually accounting for all possible ring pairs, is therefore computationally not feasible in many cases.

Methods to reduce the number of projections whilst sustaining the are widely adapted. Ideally, these should have a minimal impact on the image quality. They should therefore compress the information carried by the coincidences with, ideally, no loss. In reality, however, a reduction in computational requirements will always be a trade-off with image quality.

Axial compression, also referred to as axial undersampling, can be used to reduce the amount of data without disposing a subset of projections. It aggregates the events acquired by detector pairs with different ring differences but the same central position into one projection by averaging them.

Furthermore, axial undersampling artificially increases the resolution of the image in axial direction. When averaging projections from an even number of consecutive rings, the average position lies between two rings. Therefore, the resulting projection forms an artificial ring in their middle. The projections formed by compression are depicted as red lines in Figure 3.4 and it can be seen that every second artificial LOR lies between two detectors.

A convenient tool to visualize axial compression, the Michelogram, is shown in Figure 3.4 (A). Each division of the axes represents one ring of detectors. A mark on the grid indicates that coincidences between the corresponding rings are allowed to form. Fully 3D reconstruction would therefore be represented as a completely filled diagram without any other features. As a result, the number of projections in this case would be n_r^2 , where n_r is the number of rings.

The parameter which defines the extent of axial compression is referred to as the span. It is odd numbered and equals the number of LORs that form a projection at a ring, plus the number that form one between rings. Figure 3.4 (A) shows a Michelogram for 16 rings with a span of 7.

The projections then form segments, separated by dashed lines in Figure 3.4 (A), corresponding to different angles within the system. The diagonal segment in the Michelogram forms direct projections as shown in Figure 3.4 (B). The original LORs in black are averaged to form artificial projections indicated in red. Segments not lying on the diagonal of the Michelogram are tilted. The two segments next to the diagonal segment are shown in Figure 3.4 (C) and (D).

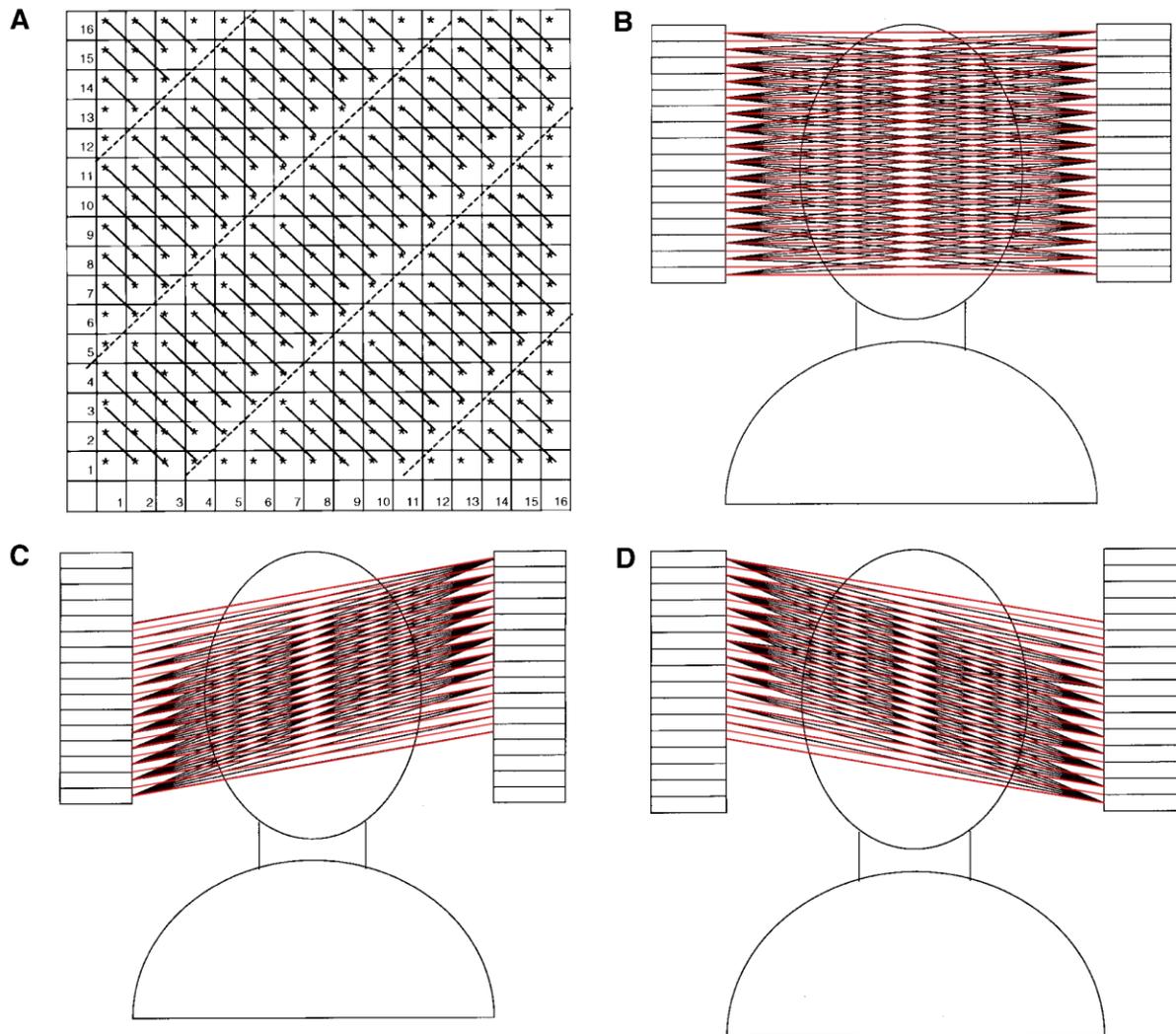


Figure 3.4: Effects of axial compression modified from [17]. **A** shows a Michelogram for a 16 ring scanner with span 7. **B** shows the projections of the diagonal segment in the Michelogram. **C** and **D** show the projections of the segments below and above the diagonal segment in the Michelogram. True LORs are indicated as black lines. The red lines illustrate the artificial LORs formed by axial compression. The LORs of the two segments farthest from the diagonal are not shown.

In addition, many applications also limit the maximum allowable ring difference for a projection to be included. In the Michelogram this leads to an exclusion of the top left and bottom right corners of coincidences. However, this obviously also decreases the sensitivity gains from the 3D acquisition.

3.5 Randoms, Scatter and Attenuation Correction

Coincidence events in a non-ideal imaging system can be divided into three groups as shown in Figure 3.5.

Events which originate from the same annihilation and reach the opposing detectors without any interaction are denoted as true coincidences. They are the only useful type for reconstruction and ideally all detected events would be true coincidences.

However, two effects can form undesired LORs. Firstly, as illustrated in the second image of Figure 3.5, it is possible for two interactions to coincide merely by chance. In parts, those random coincidences can be reduced using shorter coincidence time windows. However, it must be assured that true coincidences are not excluded due to a difference in time of arrival or statistical variations.

Secondly, the gamma rays can be scattered before reaching the detectors. The physics governing this were discussed in Section 2.1. On the right of Figure 3.5 the impact on the resulting LOR is shown. It will significantly deviate from their true origin. Apart from this, scatter also leads to an attenuation of photon flux towards the detector pairs which the photons would have reached without interaction. In 3D acquisition scattered events can make up for more than 40% of detected coincidences.

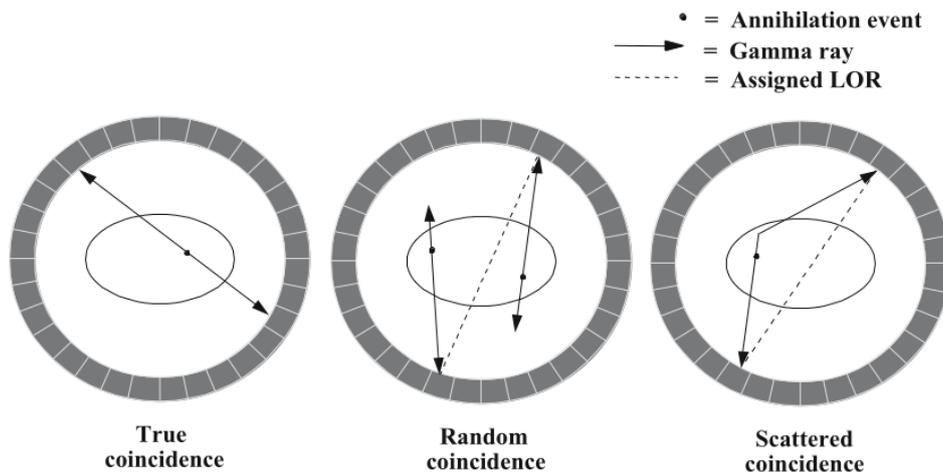


Figure 3.5: Different types of coincidence events as they are detected in a non-ideal acquisition system from [11]. **True coincidences** originate from the same annihilation and reach the detectors without any interaction. **Random coincidences** form by chance through the detection of two unrelated photons within the coincidence time window. **Scattered coincidences** yield ill formed LORs due to an interaction of one or both of the photons before reaching the detector.

Random and scattered events degrade the image quality. For low noise and quantitatively accurate images, reconstruction thus needs to correct for them.

The contribution of random coincidences is commonly estimated using a delayed coincidence time window. The detector signal is compared with a signal delayed by a time significantly longer than the coincidence window. This ensures that none of the detected coincidences originate from a true event and the output consists of randoms only. To correct for them, the delayed coincidences

are histogrammed and subtracted from the prompt sinograms before reconstruction.

While many of the methods used to estimate scatter rely on dedicated measurements or analytical models, contemporary software commonly uses simulation models. As a full Monte Carlo transport simulation is computationally not feasible for most practical applications, reconstruction toolboxes usually implement single scatter simulation (SSS), limiting the model to events where at most one of the two photons has been scattered once. Although multiple scatter does occur, it is responsible only for a minor fraction of the total scattered events.

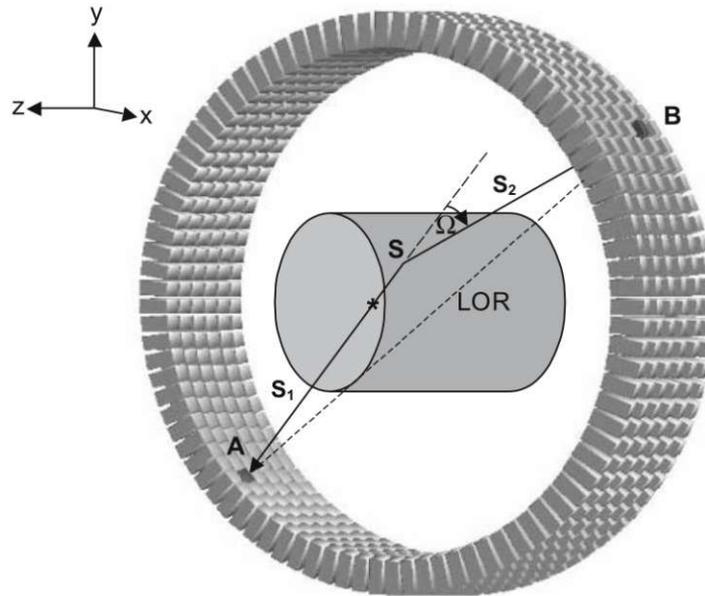


Figure 3.6: Geometrical scatter model for single scatter simulation from [11]. One of the photons is scattered in S and a wrong LOR is formed between detectors A and B . s_1 and s_2 are the paths of the unscattered and scattered photons. The scattered photon has a decreased energy.

A detailed view of the geometry of single scatter is depicted in Figure 3.6. The photons reaching the detectors were emitted at some position within the image volume V_S and exactly one of them was scattered in S . To model the rate of scattered coincidences detected for the LOR formed by A and B , five factors need to be accounted for:

1. Attenuation of the photon along the unscattered path s_1
2. The integrated emitter intensity along the path s_1
3. The probability of scattering towards detector B at point S
4. Attenuation of the scattered photon along s_2
5. Detection efficiencies of both detectors

Intensity reduction due to attenuation is estimated using Equation 2.5. The attenuation coefficient depends on the photons' energy and therefore must be corrected to accurately model the behavior after scatter. The corrected attenuation coefficients are denoted as μ' .

The emitter intensity distribution f is unknown prior to reconstruction. Therefore, it is required to perform initial scatter correction on a preliminary estimate obtained through unscattered reconstruction. Scatter estimation can then be repeated with each consecutive iteration of the reconstruction to further improve the scatter estimate. Depending on the implementation, this iterative process can be performed during the main reconstruction process [18] or prior to it with an image estimate of lower resolution [19].

The cross-section σ_c of scatter towards detector B is defined by Equation 2.2.

The detector efficiencies ϵ account for many factors involved in the probability of detection for a photon reaching the detector such as the solid angle of the detector at the scatter position, stopping power, energy resolution and specifics of the detection circuitry. Those are also dependent on the photon energy and need to be corrected for this fact.

Combining all the described factors leads to

$$P_{scat}^{AB} = \int_{V_s} dV_s \left(\frac{\sigma_{AS}\sigma_{BS}}{4\pi R_{AS}^2 R_{BS}^2} \right) n \frac{d\sigma_c}{d\Omega} (I_A + I_B) \quad (3.12)$$

as the model for the rate of scattered coincidences between a pair of detectors P_{scat}^{AB} , which was first published by Watson et al. in [20] and where the intensities I_A and I_B are defined as

$$I_A = \epsilon_{AS}\epsilon'_{BS} e^{-\left(\int_S^A \mu ds + \int_S^B \mu' ds\right)} \int_S^A f ds \quad (3.13)$$

and

$$I_B = \epsilon'_{AS}\epsilon_{BS} e^{-\left(\int_S^A \mu' ds + \int_S^B \mu ds\right)} \int_S^B f ds, \quad (3.14)$$

where R_{SA} and R_{SB} are the distances between S and the detectors, and $n = \mu/\sigma_c$ denotes the density of scatter centers.

In short, the integrand of Equation 3.12 calculates the rate at which an annihilation somewhere on s_1 or s_2 is scattered in S towards the detectors through single scatter and both events are actually detected. Through integration of this across the whole image volume the total rate of scattered coincidences for the detector pair formed by A and B is obtained. The integral needs to be evaluated for each detector pair. As a consequence, scatter estimation is a computationally demanding task and can be responsible for a significant amount of the processing time of reconstruction. Therefore, estimating scatter on a reduced image resolution can be beneficial. After an estimate of the scatter contribution for all bins is calculated it is subtracted from the projections to obtain a corrected version of the emission sinograms.

Lastly, scattered photons must not only be accounted for in terms of wrongly formed LORs, but also in terms of signal attenuation. Using Equation 2.5 the probability that two photons sent out from X both reach the correct detectors is

$$p_{AB} = e^{-\int_X^A \mu dx - \int_X^B \mu dx} = e^{-\int_A^B \mu dx}, \quad (3.15)$$

which conveniently does not depend on the position of the annihilation on the LOR. Therefore, projection data can be corrected for attenuation by division with p_{AB} .

4 Materials and Methods

For this thesis, data from an fPET study using a Siemens Biograph molecular magnetic resonance (mMR) system was reconstructed using two openly available toolboxes as well as the vendor's, called e7-tools. The vendor toolbox served as the reference implementation and the results from other toolboxes were compared to it in terms of image quality and processing efficiency.

The following chapter briefly describes the mMR system, the execution environment in which the processing was performed, the input data, preprocessing and analysis steps. Furthermore, the used toolboxes as well as the methods used to analyze and compare individual results are introduced. All programs written for the execution of the reconstructions, analysis and evaluations as well as the Dockerfiles defining the environments of the open toolboxes are published on GitHub [21].

4.1 Siemens Biograph mMR

The Biograph mMR is a combined PET-MRI system developed and distributed by Siemens. An image of it is shown in Figure 4.1. As the specifics of the MRI system are of only minor relevance to the present work they will not be described in further detail. Results of benchmark measurements as well as further details on both the PET and MRI system are available in the performance evaluation publication of the system [22]. The PET scanner in the system consists of 8 rings with 56 detector blocks per ring. A detector block consists of 8x8 detectable crystal positions read out by an 3x3 APD array. The individual detectors are 4x4x20 mm LSO crystals [23].



Figure 4.1: Siemens Biograph mMR from [23].

4.2 Siemens Biograph Vision Quadra

The Biograph Vision Quadra is shown in Figure 4.2. It is a combined PET-CT LAFOV system which internally concatenates four PET systems originally developed for the Biograph Vision PET-CT system. Throughout this work it will serve as an example LAFOV system to extrapolate the results from the Biograph mMR to a larger system. Performance measurements and detailed information about the integrated PET and CT systems are available in [24]. The PET system has 32 rings of 38 detector blocks, where on block contains 4x2 mini blocks which consist of 5x5 detector crystals each. This results in a total axial span of 320 crystals, a total length of the FOV of 106 cm and 760 angular detector positions within a single ring. The individual detectors are 3.2x3.2x20mm LSO crystals [25, 26].



Figure 4.2: Siemens Biograph Vision Quadra from [26].

4.3 Execution Environment

Due to restrictions in the operating system and processor compatibility of the toolboxes, independent workstations were used for the e7-tools and other toolboxes. Additionally, the e7-tools were executed on two machines with the second one having more performant hardware to analyze performance differences. The specifications of the three used computers are listed in Table 4.2.

For reproducibility and portability to other systems the reconstructions running on the Linux workstation were set up and executed within Docker [27] containers. Utilities provided by Docker were also used to monitor the resource consumption of the processes as described in Section 4.6. Data exchange between host system and containers happened through two directories, one for data input and one for data output, which were mounted into the respective containers as

volumes. The reconstruction pipelines then output their results in a predefined format and the analysis described in Section 4.6 is executed to calculate the parameters to compare the toolboxes with regard to image quality and processing time in a standardized manner.

The e7-tools reconstruction was executed directly on the Windows workstations without any virtualization.

Table 4.2: Specifications of computers used for the project. Floating point operations per second (flops) refer to 32 bit values.

	Linux Workstation
CPU	AMD Ryzen Threadripper PRO 5975X 3.6 GHz (32 Cores)
Memory	1024 GB (8 x 128 GB) DDR4 3200 MHz
GPU	NVIDIA T400 1.1 Tflops 4GB GDDR6 SDRAM (80 GB/s) PCIe 3.0
Disk	Kingston Fury Renegade PCIe 4.0 NVMe M.2 SSD
Operating System	Ubuntu 24.04.1 LTS
	Low-End Windows Workstation
CPU	Intel Xeon E5-1650 v2 3.5 GHz (6 Cores)
Memory	32 GB (2 x 16 GB) DDR3 1600 MHz
GPU	NVIDIA Quadro K600 336 Gflops 1 GB DDR3 (29 GB/s) PCIe 2.0
Disk	WD Red 4TB SATA SSD
Operating System	Windows 10.0.17763 Enterprise LTSC
	High-End Windows Workstation
CPU	Intel Xeon Gold 64384 3.7 GHz (64 Cores)
Memory	1024 GB (16 x 64 GB) DDR4 3200 MHz
GPU	NVIDIA L40S 91.6 Tflops 48 GB GDDR6 + ECC (864 GB/s) PCIe 4.0
Disk	Micron 7450 PRO PCIe 4.0 NVMe U.3 SSD
Operating System	Windows 10.0.17763 Enterprise LTSC

4.4 Input Data

The data used for the reconstruction was from the measurement of a single subject from an fPET study by Hahn *et al.* [3]. The task design of the study is depicted in Figure 4.3. For the fPET measurement, a bolus plus constant infusion protocol is used. During this protocol, a certain amount of $[^{18}\text{F}]FDG$ is applied right at the start of the measurement. After this application the radioligand uptake will be fairly constant after a short amount of time. Therefore, the baseline glucose metabolism as well as task induced changes can be quantified. After application of the bolus, the infusion continues through the whole measurement with a lower constant rate. This ensures that free radioligand is available in the bloodstream throughout the whole measurement.

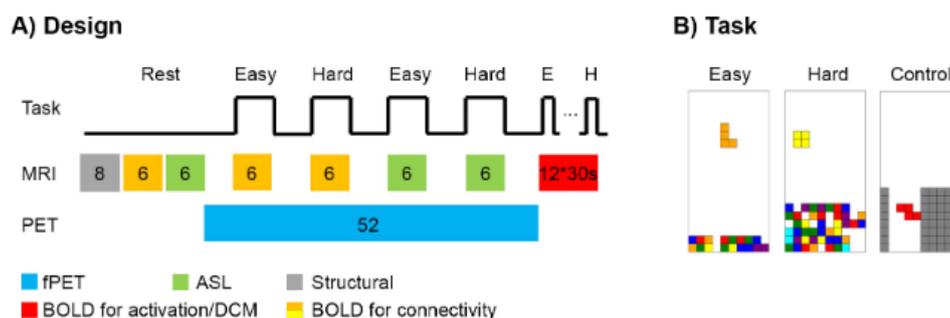


Figure 4.3: Study design of the reconstructed acquisition. Image copied from [3]. Initial MRI measurements were executed at rest. During the fPET block subjects were presented with alternating blocks of cognitively easy and hard tasks separated by short breaks. During the tasks blocks blood-oxygenation-level dependent (BOLD) and arterial spin labelling (ASL) MRI measurements were performed concurrently to the fPET acquisition.

For reconstruction, three types of data output from the scanner are used:

1. List mode data of coincidences (~ 9.5 GB)
2. Normalization scan of detector efficiencies
3. T1 weighted structural MRI image

List mode data is exported in form of a header following an interfile-like format by Siemens in combination with a binary file containing timing information and prompt as well as delayed coincidences. The binary file comes in a 32-bit PETLINK [28] format. The prompt and delayed coincidence events per frame over the acquisition time are shown in Figure 4.4. The number of events sharply increases at the beginning of acquisition because of the bolus. During the time of constant infusion the number of events increased approximately linearly. The code used to extract the number of prompt and delayed events from the binary data provided by the scanner is given in Appendix A.

In addition to the main PET data, normalization sinograms are acquired on a daily basis prior to the main measurements using a germanium source. Furthermore, synthetic computed tomography (CT) scans for attenuation correction are generated from a T1 weighted structural MRI image using the methods described by Burgos *et al.* in [14]. The used list mode data contained an extra minute of acquisition and was therefore reconstructed as 106 frames of 30 seconds.

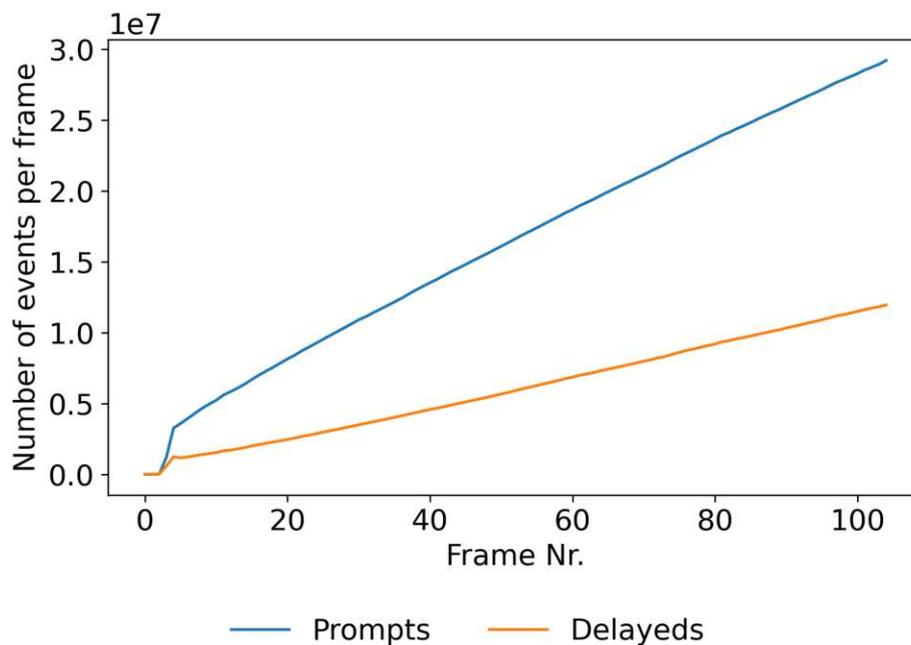


Figure 4.4: Prompt and delayed events in the list mode data per frame of 30 seconds.

4.5 Reconstruction Toolboxes

In the initial phase of this project a thorough internet and literature review was conducted to identify toolboxes for dynamic PET reconstruction that are openly available. For a toolbox to be included into the analysis it must

- support reading the list mode data in the provided format, or
- provide a working data converter from the scanner format, and
- support dynamic reconstruction, or
- be adaptable to support it with reasonable effort, and
- be sufficiently documented so that a clear path for setting it up is visible, and
- be compatible with the Siemens Biograph mMR.

Although many candidates were found, in the end only two of the toolboxes were able to meet the requirements. The reasons for excluding the others were due to compatibility issues or because the toolboxes were simply not available. A brief description of the disregarded toolboxes and the reasons behind the decisions are given in Section 4.5.4. The two toolboxes that led to successful reconstructions were SIRF [29] and NiftyPET [18]. Their results and performance characteristics were compared to the e7-tools, a research software released by Siemens.

4.5.1 Siemens e7-Tools

The Siemens e7-Tools are a set of programs provided to scientists for experimentation and research with reconstruction parameters. At the research group this project was executed, the toolbox is the standard software for reconstruction. They serve as the reference implementation to which results from other toolboxes will be compared to. Although other algorithms are available in the toolbox and parameters can be adapted, the aim of the reconstruction for this project was not to compare different parameters of the reconstruction algorithm. Therefore, a set of parameters which has yielded successful results for research was used without modification.

The following parameters were used for the reconstruction:

- 106 frames of 30 seconds
- 24 subsets
- 3 iterations
- 60 maximum ring difference
- Span 11
- Ordinary Poisson - Ordered Subset Expectation Maximization (OP-OSEM)

The frame length was chosen deliberately longer to limit the processing time necessary for the analysis. It was initially planned to adjust the reconstruction to span-1, however, this was not possible in the given version of the e7-tools and the reconstruction therefore had to be executed with span-11. Apart from this, these parameters represent the toolbox defaults. The reconstruction was executed with them on both the high- and low-end Windows machine. In each case computations were executed on the CPU once and another time with GPU processing was activated. As this toolbox is proprietary and distributed in binary form, it is not possible to provide further details on the methods used for reconstruction apart from the aforementioned parameters.

Although GPU processing is not fully supported by the e7-tools, no problems were encountered for reconstructions of data from the Biograph mMR system with the described configuration. However, other tests have shown that enabling the GPU for other scanner systems, such as the Siemens Vision Quadra, did not utilize the GPU.

4.5.2 SIRF

The Synergistic Image Reconstruction Framework (SIRF) [29] is a collection of software interfaces aimed for combining PET and MRI reconstruction. It is developed by the Collaborative Computational Project in Synergistic Reconstruction for Biomedical Imaging (CCP-SyneRBI), a collective of multiple institutions. For PET reconstruction it relies on the "Software for Tomographic Image Reconstruction" (STIR) [19] toolbox.

SIRF was used in favor of directly accessing STIR as it provides a Docker image with all the necessary dependencies, including STIR, preinstalled and using it is recommended by the STIR user guide [30], if the toolbox is not compiled from source. With the image, the reconstruction setup becomes significantly more approachable. Furthermore, SIRF provides high level Python interfaces and example scripts, which further reduced the complexity of the setup.

To make the list mode data and normalization files compatible with SIRF, they first need to be converted into the appropriate STIR interfile format. The conversion was performed using `nm_extract` provided by the `pet-rd-tools` [31], which is a collection of utilities for different tasks related to working with emission data.

Furthermore, the pseudo CT images were converted to a subject attenuation map using `nm_mrac2mu` and merged with a hardware μ -map using NiBabel [32]. For this, the subject μ -map was resampled to the resolution of the hardware μ -map using NiLearn [33].

The individual steps of the reconstruction were combined into a script based on examples provided by SIRF. STIR classes were accessed via the `sirf.STIR` Python module, which is installed in the aforementioned Docker image. All classes mentioned in the following paragraphs are from this module if not noted differently.

As a first step, the list mode data is histogrammed using the `ListmodeToSinograms` class, from which a subsequent estimation of randoms sinograms from delayed coincidences can also be obtained. Next, the an `AcquisitionSensitivityModel` object is defined. It represents the system matrix and loads detector efficiencies from normalization files to correct for efficiency differences between detectors.

Scatter sinograms are estimated using the `ScatterEstimator` class. Internally this class relies on SSS, as it was described in Section 3.5. Scatter estimation is performed prior to the main reconstruction routine on a lower resolution image estimate and down-scaled scanner model. To update the image estimate the OSEM algorithm with 3 iterations and 21 subsets is used.

Using the previously mentioned `AcquisitionSensitivityModel` object, together with the output of scatter and randoms estimation, an `AcquisitionModel` was defined. This object is used by STIR to forward- and backproject acquisition data including all necessary corrections.

The reconstruction object was set up as an `OSMAPOSLReconstructor` with a logarithmic Poisson log-likelihood as the objective function. This reconstruction object can be used for more advanced algorithms, however, if there is no prior or penalty defined, the implemented algorithm is equivalent to OSEM.

As the reconstruction is only executed for a single frame, all the described steps were repeated on consecutive frames over the full acquisition duration. The reconstructed images were saved to intermediate files using the `NiftiImageData` class from the `sirf.Reg` module. Finally, the resulting single-frame images were merged into a single 4D file using the `fslmerge` utility provided by the FMRIB Software Library (FSL) [34].

4.5.3 NiftyPET

NiftyPET [18] is an image reconstruction toolbox developed at the University College London (UCL). It is distributed as a Python package and all main processing routines are parallelized on a GPU using CUDA C. Limiting the processing to the CPU is not supported.

The toolbox supports reading the list mode and normalization data in the format provided directly by the scanner and also internally generates the attenuation images from the synthetic CT scans. Thus, no format conversions or similar steps are necessary prior to reconstruction.

Additionally, all the steps of the reconstruction are abstracted into the `nipet.mmrchain` function, therefore it is not necessary to define the individual steps explicitly. Also, frame by frame reconstruction is supported by default and outputs the 4D data directly into a single file.

Scatter is estimated using a voxel based scatter model (VSM). This approach is based on and generally similar to SSS as it was described in Section 3.5, however, it computes a full scatter

sinogram for the activity in each individual voxel. According to the toolbox authors this is beneficial in terms of quantitative reconstruction results and better suited for parallelization of computations [18].

Because the number of subsets is set inside the source code of the toolbox, adjusting it to the value used with other solutions was not possible. As a consequence, the default of 14 subsets was used. Reconstruction was performed on those subsets with 7 iterations over the full dataset to arrive at a similar number of total iterations as other solutions.

During first tests with the toolbox it was found that frames without activity within the FOV lead to an error. Due to the study protocol of the measurements used for this project tracer is only applied shortly after PET acquisition start and initial frames without activity are present. Therefore, the first five minutes of acquisition were excluded from reconstruction with this toolbox. The decision to exclude this fairly long timeframe from the beginning was arbitrary and certainly cropped some frames with activity from the reconstruction.

4.5.4 Disregarded Toolboxes

The toolboxes which were excluded from further analysis are summarized in the following paragraphs with some reasoning behind the decision. They are ordered by increasing estimated severity of the obstacles which were encountered.

Customizable and Advanced Software for Tomographic Reconstruction - CASToR [35] is an open-source toolbox for emission and transmission reconstruction developed by a collaboration of mainly french scientists. It does not offer any GPU acceleration, however, the processing on the CPU is highly parallelized. Reconstruction can be performed starting with list mode data as well as with sinograms. For this, a dedicated file format is necessary in both cases. For the Biograph mMR, only a converter from sinograms is available. Thus, the data was initially histogrammed using the e7-tools. It was then attempted to convert them into the CASToR data format using the converter provided on the project website. However, processing the file did not seem to finish and was cancelled after more than twelve hours of processing. The converter used a single CPU core during the conversion. Possibly, it has internal inefficiencies which are no problem when converting clinical datasets with shorter acquisition times but become problematic on the extended duration of the data used for this study.

Quantitative Emission Tomography Iterative Reconstruction - QETIR [36] is developed by the MEDISIP group at Ghent University, Belgium. Researchers interested in the software can contact MEDISIP, which was done. However, the request was declined.

PyTomography [37] is a Python library which relies on the use of PyTorch tensors for GPU accelerated calculations. As this higher level approach has significant benefits with regard to developer experience, compared to lower level implementations using CUDA C such as NiftyPET, a comparison of this toolbox would have been particularly interesting. The software cannot directly read the list mode data provided by the Biograph mMR, however, it supports the fairly new PETSIRD standard format.

Although a converter directly from the scanner format to PETSIRD was not available a, at the time of writing unfinished but usable, software to convert from the STIR format to PETSIRD does exist [38]. First `nm_extract` from `pet-rd-tools` [31] was used to convert to the STIR format. Subsequently, this data was fed into the other converter. Which was very inefficient but worked.

The data loading process from the PETSIRD list mode data file consumed large amounts of

memory of more than 200 GB, compared to the 9.5 GB of raw list mode data volume. It is suspected that this is caused by the fact that in the current version of the software individual coincidences are loaded from the PETSIRD file into an array of Python objects, which are known for their memory inefficiencies compared to types used in lower level languages.

Furthermore, the toolbox does not support frame by frame reconstruction and also specifying a limited timeframe is not possible. Although we managed to change the source code of the toolbox in a way that we could reconstruct an image from a timeframe with unknown start and end times, implementing dynamic reconstruction with strictly defined frame lengths and the necessary output utilities was beyond the scope of this thesis.

TomoLab - Occiput - NiftyRec [39] The toolbox is publically available and looked promising. However, we were not able to find usable examples or documentation on how to set up a reconstruction. The toolbox is not actively maintained anymore.

OMEGA [40] is a set of routines written in C++, including CUDA C for GPU support, with additional higher level interfaces in Matlab and Python. There is no converter available to transform the scanner data into the required input format, therefore, a reconstruction using this toolbox could not be implemented. The toolbox was initially designed for PET-CT systems.

EMRecon [41] is not developed any more and therefore not available.

For **PRESTO** [42] only the publication could be found but without any information about how to obtain the toolbox. The original publication is from 2008 and development was discontinued since then.

4.6 Toolbox Comparison

The reconstruction pipelines established using the aforementioned toolboxes were evaluated in terms of processing time, computing resource consumption and image quality.

Depending on the toolbox, different steps were necessary to instrument the reconstruction with timing measurements. In the case of SIRF, the individual steps of the pipeline are defined in the main reconstruction script. Consequently, the reconstruction steps were timed directly there. In comparison, the NiftyPET toolbox has the full pipeline abstracted. Therefore, a copy of the main NiftyPET repository was set up and modified to have timing measurements in all the relevant places. The updated version of the Python package was then installed into the reconstruction container, it is available on GitHub [43].

To compare the toolboxes in terms of their computational resource usage, in parallel to the reconstructions the following parameters were monitored as a time series over the full reconstruction duration:

- Number of utilized CPU cores
- Memory usage
- GPU utilization
- GPU memory usage
- Data read from disk per time interval
- Data written to disk per time interval

For reconstructions running on the Linux workstation, the parameters were acquired for the container they were running in using `docker stats` and `iotop`. Additionally, GPU parameters were acquired using the NVIDIA System Management Interface (`nvidia-smi`). For the e7-tools running on the Windows workstations, the parameters were measured for the process using the `Get-Counter` powershell cmdlet [44] on the Windows Workstation. Only for the GPU parameters on the high-end machine this cmdlet did not return any values. Therefore, the GPU parameters were also obtained using `nvidia-smi`.

To calculate image quality metrics for certain regions within the brain, the reconstructed images were transformed into Montreal Neurological Institute (MNI) space [45]. This is a standardized frame of reference used by neuroscientists to factor out the varying anatomy of subjects. The transformation is commonly also referred to as normalization. For this, the 4D PET data was motion corrected to account for subject movement. Next, a T1 weighted structural MRI acquisition was coregistered to the mean PET image and then transformed into MNI space. The obtained transformation parameters were subsequently applied to the PET image frame by frame. All these steps were performed using tools provided by FSL [34].

By applying this transformation the reconstructed volume can be segmented into brain regions using the automated anatomical labeling (AAL) atlas [46]. This is necessary as different regions of the brain show varying tracer uptake and the image quality metrics were thus compared by region. Furthermore, it ensures that only volumes which lie within the brain are used for the calculations. Including data from outside the brain would underestimate the standard deviation of activity as none is present outside the body.

Two indicators of image quality from [47] were used to compare image qualities. To ensure that the noise levels in the images are similar, the signal-to-noise ratio (SNR) was calculated per brain region. It is defined as

$$SNR(f) = \frac{\bar{f}}{\sigma_f}, \quad (4.1)$$

where f is the reconstructed activity concentration in the ROI. \bar{f} and σ_f denote the expectation value and standard deviation of f within the brain region of interest.

To additionally ensure a similar contrast of the images, the contrast-to-noise ratio (CNR) was calculated as

$$CNR(f) = \frac{\bar{f} - \bar{f}_B}{\sigma_B}, \quad (4.2)$$

where \bar{f}_B is the mean of the background and σ_B is the standard deviation of the background. The activity in the cerebrospinal fluid was used as the background region. It was segmented using a tissue probability map generated with SPM12 [48] and a threshold of 0.8.

5 Results

A slice of one frame of the series of reconstructed images is shown in Figure 5.1. It is clearly visible that it was not possible to achieve equivalent results with all toolboxes. The time required to reconstruct the full acquisition into a series of images with the individual setups is listed in Table 5.3. In total, all openly available toolboxes required a significantly longer processing time compared to the reference implementation. Furthermore, an acceleration of the e7-tools reconstruction was observed on the high-end setup. The following chapter will describe the measurements and their differences in further detail and elaborate on variations within steps of the reconstruction.

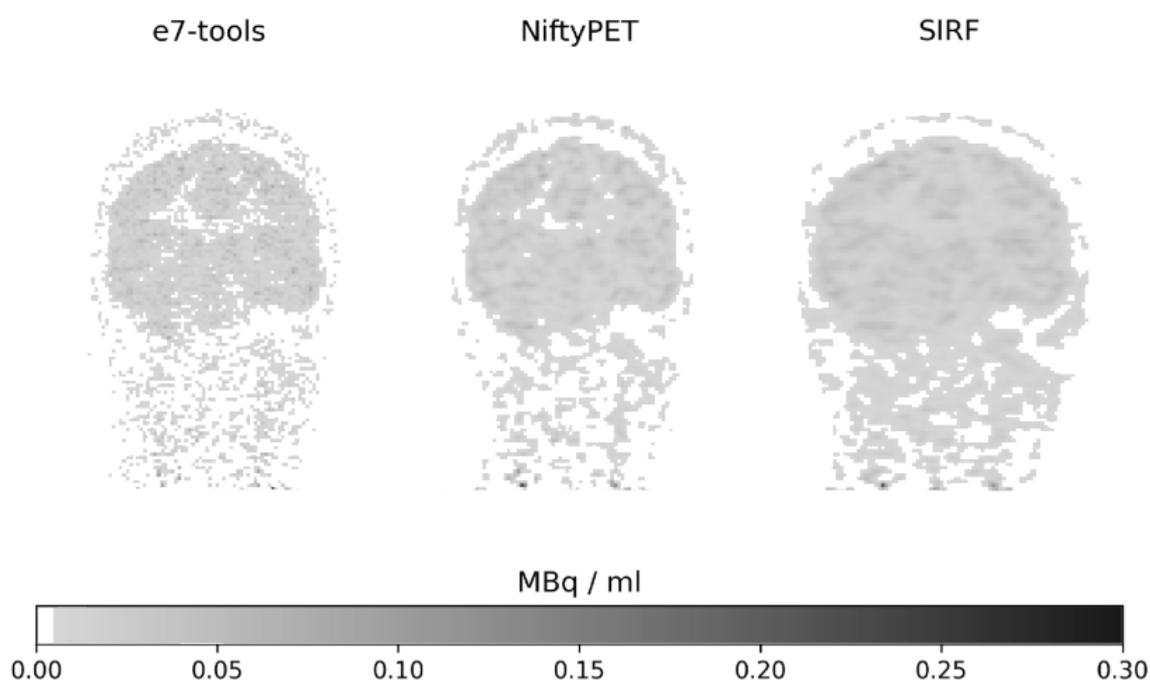


Figure 5.1: Slice of the reconstruction result of all toolboxes for frame 75.

For histogramming, the fastest option was the e7-tools on the stronger hardware. NiftyPET was in a similar magnitude with seven minutes instead of two. On the less powerful hardware the e7-tools are slower by an order of magnitude and SIRF was the overall slowest option taking almost four hours for histogramming.

For scatter simulation, NiftyPET is the slowest solution. The most performant implementation for this step is provided by the e7-tools with a total of 21 to 25 minutes spent on scatter estimation on all systems. Compared to this, SIRF is slower by a factor of 10 and NiftyPET by 25.

For the reconstruction, the by far most performant solution are the e7-tools using the powerful GPU with a nearly sevenfold acceleration compared to the implementation solely executing computations on the CPU. NiftyPET took longer than the reference, with an increase of about 60 %. SIRF's reconstruction took around 250 % longer. Enabling the GPU on the low-end setup lead to an increase of processing time by a factor of nearly two.

The routines for measuring resource usage described in Section 4.6 required more time than

expected on all setups. Therefore, the inter-frame time-series presented in the following subsections cannot properly represent fast changes. As a consequence, fast and short changes in the values can be suppressed. This can lead to inconsistencies throughout frames. Observations where this is of relevance are indicated in the text.

Table 5.3: Processing times of all tested reconstruction variants for 106 frames of 30 seconds. Durations are given in hh:mm format. Percentages in parentheses show the increase in total processing time compared to the reference and the share of individual steps. LE and HE indicate the low- and high-end systems used for the e7-tools reconstruction. The best results per column are highlighted.

	Total	Histogramming	Scatter	Reconstruction	Unassigned
e7 - tools					
LE CPU	03:36	00:37 (17 %)	00:21 (10 %)	01:51 (51 %)	00:47 (22 %)
HE CPU	02:29 (-31%)	00:01 (1 %)	00:21 (14 %)	01:45 (70%)	00:20 (14 %)
LE GPU	06:03 (+68%)	00:34 (9 %)	00:21 (6 %)	03:43 (62 %)	01:24 (23 %)
HE GPU	01:02 (-71%)	00:03 (4 %)	00:25 (40 %)	00:16 (25%)	00:18 (29 %)
NiftyPET	12:31 (+248%)	00:07 (1 %)	08:44 (70 %)	03:05 (25 %)	00:32 (4 %)
SIRF	14:16 (+296%)	03:44 (26 %)	03:33 (25 %)	06:34 (46 %)	00:25 (4 %)

5.1 e7-tools

Figures 5.2 and 5.3 show the CPU and RAM utilization of the e7-tools on the low-end setup for reconstruction of an example frame without and with GPU enabled. In addition, the GPU utilization and memory usage for the GPU variant is shown in Figure 5.4. Due to the aforementioned problems with `Get-Counter`, statements about the intraframe resource usage can only be verified for the low-end setup. Furthermore, histogramming takes place prior to reconstruction using an individual executable. Therefore, the metrics for one frame were concatenated in the plots. The vertical dashed line indicates the end of the histogramming process. The resulting sinograms are written to files and read from disk by the reconstruction.

Histogramming is not parallelized and mostly uses less than one CPU. A continuously high disk utilization could be observed during this time on the low-end setup. The constant memory usage of around 2 GB could be observed consistently for all frames. Additionally, the histogramming took a highly variable but constant time of $\sim 20.5 \pm 5.5$ seconds per frame. On the high-end setup this was significantly lower and less variable with $\sim 0.8 \pm 0.1$ seconds.

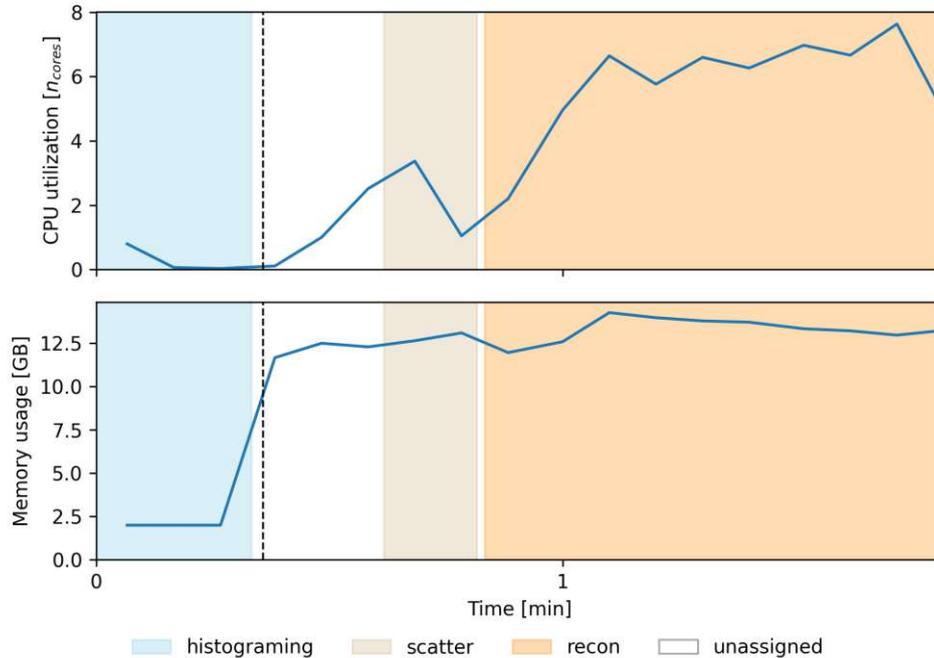


Figure 5.2: e7-tools without GPU on low-end system: CPU and RAM utilization for frame 75.

After histogramming, a period which couldn't be assigned to any concrete task was observed. This period is primarily responsible for the high unassigned percentage presented in Table 5.3. It required a constant time per frame of $\sim 17.5 \pm 0.8$ seconds on the low-end CPU setup, $\sim 10.2 \pm 0.4$ on the high-end CPU, $\sim 38.9 \pm 0.8$ on the low-end GPU and $\sim 9.5 \pm 0.4$ on the high-end GPU setup. The tasks during this time show a small amount of parallelization on the CPU towards the end with the non-GPU version. With the GPU enabled, a sharp increase in CPU utilization could be observed in the beginning, followed by a decline to single-threaded processing. Likely due to the long sampling intervals, this sharp increase was not present for all frames. Furthermore, when enabled, the GPU is active during this time. It is fully utilized shortly after the start.

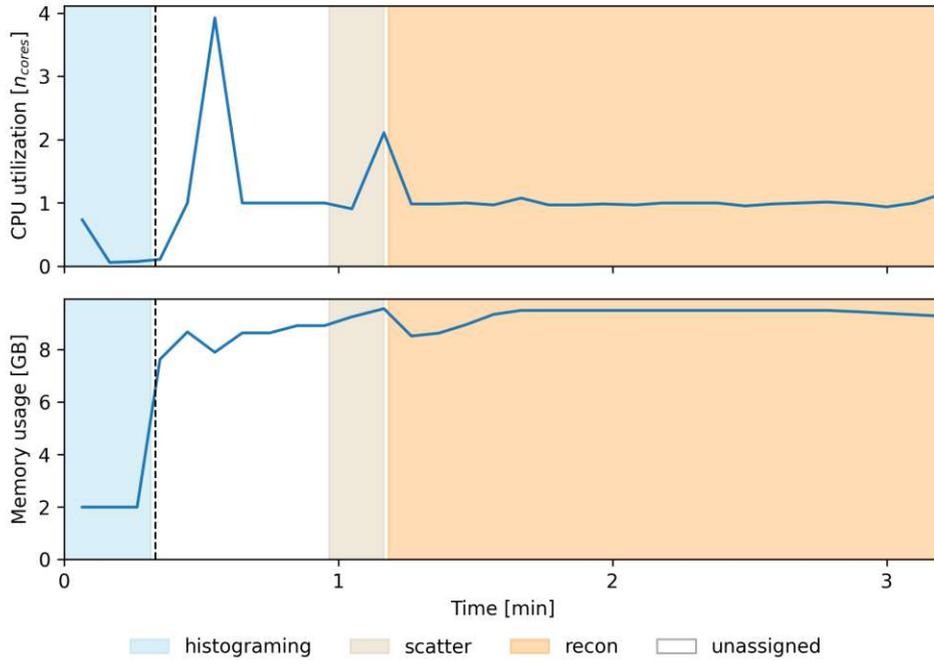


Figure 5.3: e7-tools with GPU on low-end system: CPU and RAM utilization for frame 75.

On the low-end setup GPU memory usage increases to 600 MB during this time and goes back to zero before scatter estimation starts. This behavior was observed consistently for all frames. Due to inaccuracies in assigning certain stages in processing to specific subtasks through the log output of the e7-tools, the possibility remains that this unassigned portion of processing is in parts responsible for preparation tasks of the scatter simulation.

During scatter estimation, which takes $\sim 11.6 \pm 0.7$ seconds per frame, the memory consumption stays at the original level and the CPU utilization lies between one and four cores. The duration of the scatter simulation is similar on all tested setups. Due to the short duration in comparison to the sampling interval, values for the CPU utilization within this portion of the process tend to be highly variable. The GPU was not active during scatter estimation even if it is enabled.

The transition to the main reconstruction process does only slightly decrease the host memory usage. Shortly after the start it again increases to previous levels. On the GPU, allocated memory is reduced to approximately 150 MB. However, CPU utilization is significantly higher compared to previous steps if GPU processing is disabled. If it is enabled, then the CPU is continuously running with a single active core and the GPU is operating with a utilization of around 90%. In terms of GPU memory a similar behavior was observed on the high-end setup with a higher demand in memory of 2 GB during scatter which also dropped for reconstruction to around 500 MB. The GPU memory usage was therefore approximately increased by a factor of four.

As mentioned, most of the parameters not vary between frames. However, starting from 3 GB, a linear increase in RAM usage of ~ 180 MB per frame was observed. When GPU usage was disabled, this increase continued uninterrupted until reaching a maximum of approximately

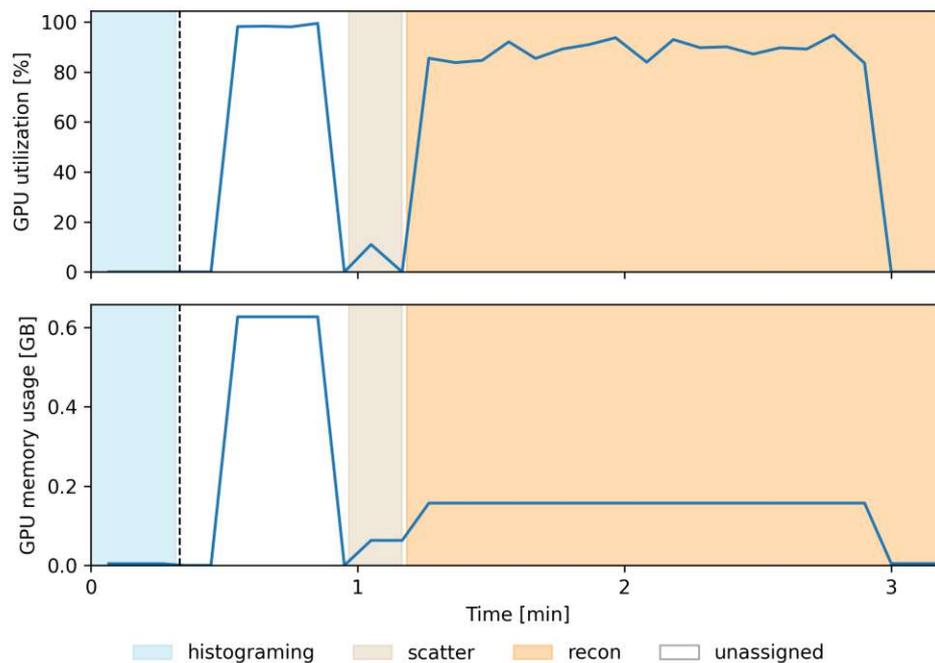


Figure 5.4: e7-tools with GPU on low-end system: GPU utilization and memory for frame 75.

18 GB before the end of processing. When GPU was enabled on the lower power system the increase was also present, however, it sharply decreased to less than 6 GB at frames 47 and 77. Therefore, the maximum used host memory was ~ 9 GB when GPU processing was enabled.

On the high-end system the memory was increasing in the same way as on the low-end system with and without the GPU activated as shown in Figure 5.5 and 5.6. The reduction of the memory usage during processing was not observed in the high-end system. Furthermore, it can be seen in Figure 5.5 that, although the high-end system has a higher number of CPU cores available, processing is only intermittently parallelized to more cores. In general, most of the processing only uses around six cores. This is mostly equivalent to what the low-end system used.

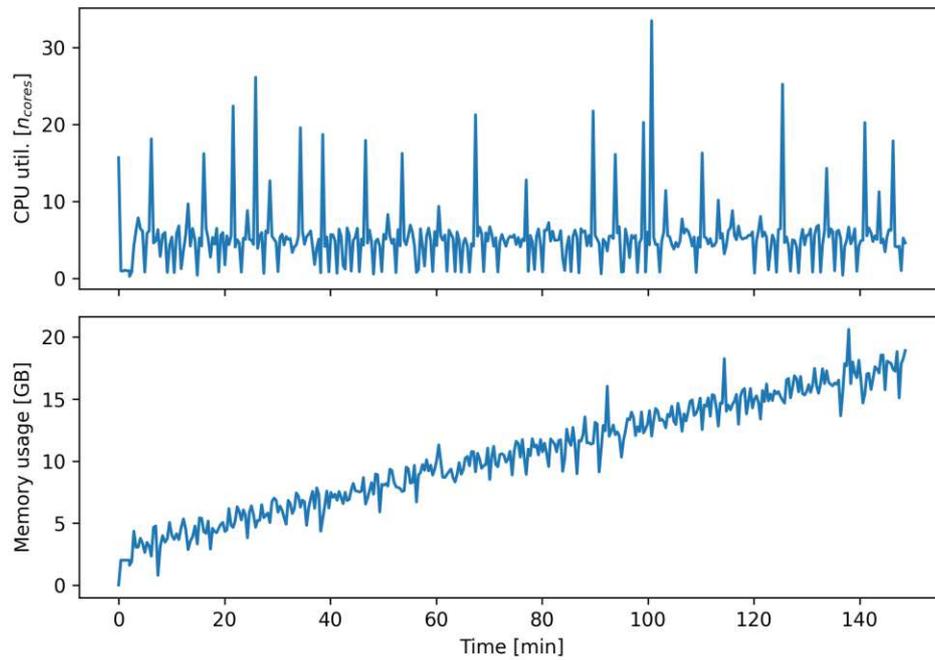


Figure 5.5: e7-tools without GPU on high-end system: CPU utilization and memory usage for the full reconstruction.

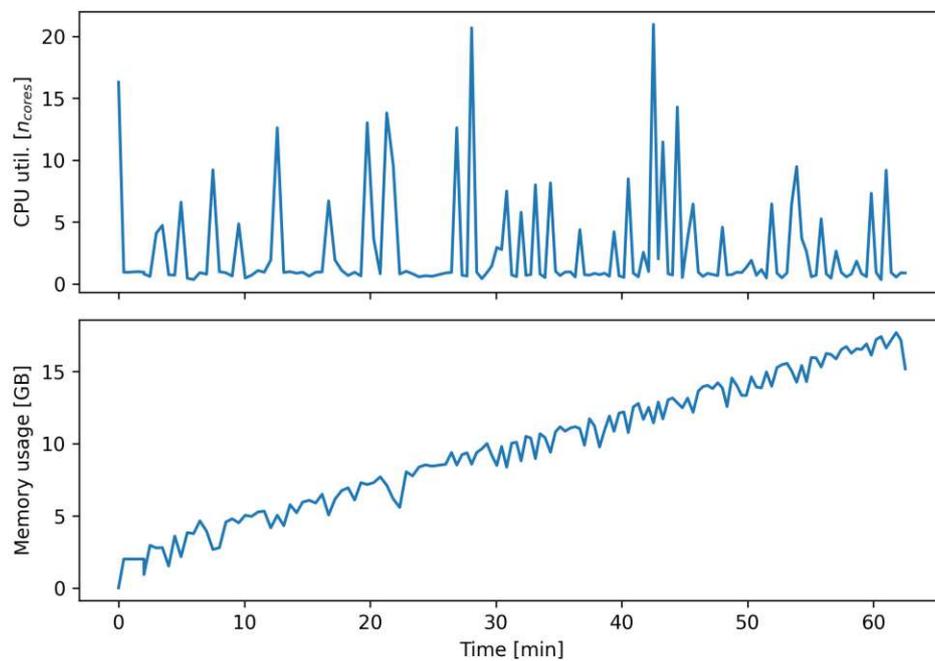


Figure 5.6: e7-tools with GPU on high-end system: CPU utilization and memory usage for the full reconstruction.

5.2 NiftyPET

CPU and RAM usage as well as GPU utilization and memory usage for a single frame in the reconstruction with NiftyPET are shown in Figure 5.7 and 5.8.

The histogramming involves the calculation of sinograms for both prompt and delayed coincidences and took a constant time per frame of $\sim 4.2 \pm 0.13$ seconds. Before the main reconstruction starts the randoms are estimated from delayed sinograms. A constant unassigned period of $\sim 5 \pm 0.02$ seconds was observed between histogramming and randoms estimation during which attenuation factors are forward projected. The randoms estimation also took a constant time of ~ 1.6 seconds. Between randoms and the main reconstruction, another $\sim 9.8 \pm 0.04$ seconds are unassigned.

The intermediate reconstruction blocks iterate the full dataset once with 14 subsets, which on average takes ~ 16.5 seconds per block or ~ 115.5 seconds per frame. Compared to the reference of the e7-tools on the low-end CPU system this resembles an increase of 60 %. Compared to the configuration of the e7-tools running on a significantly more powerful GPU, the processing time is increased by more than a factor of ten.

Contrary to all other toolboxes which were part of this analysis, NiftyPET does not simulate scatter prior to the main reconstruction using an image with lower resolution. Instead, it uses the current estimate of the resulting image as the activity distribution for the scatter estimation. This can be seen in Figures 5.7 and 5.8 as alternating blocks of reconstruction and scatter simulation. The scatter estimation step within one iteration takes $\sim 45.7 \pm 1.3$ seconds which results in a total of $\sim 274.2 \pm 7.8$ seconds per frame. In comparison to the e7-tools this is an increase by a factor of almost 25. This implementation is therefore by far the slowest in simulating scatter.

During processing of a frame one CPU core was used continuously. Host memory was fairly constant within a frame, showing an increase after histogramming and small spikes at the end of scatter estimation. However, as shown in Figure 5.9 and starting at approximately 4 GB, the allocated memory grew linearly towards later frames with 330 MB per frame, reaching 35 GB at the last frame.

The GPU is fully utilized for most of the time, with short interruptions at the transition between substeps. The GPU memory used was approximately 2 GB during reconstruction iterations and 600 MB during scatter, with a short spike to around 1.8 GB shortly before finishing an iteration of the scatter estimation.

Compared to the reference, NiftyPET required twice the amount of host memory.

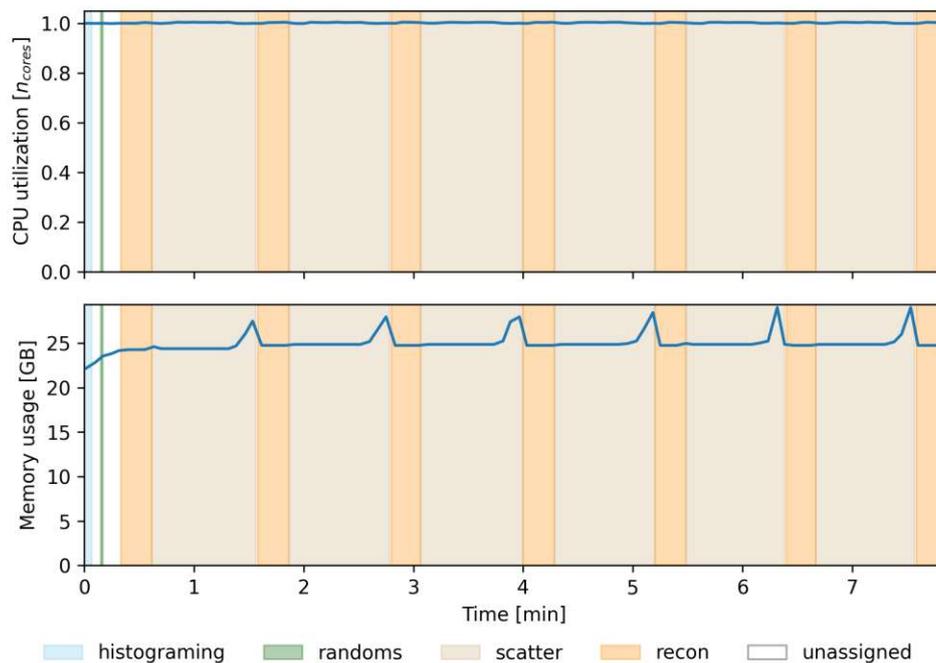


Figure 5.7: NiftyPET: CPU and RAM utilization for frame 75.

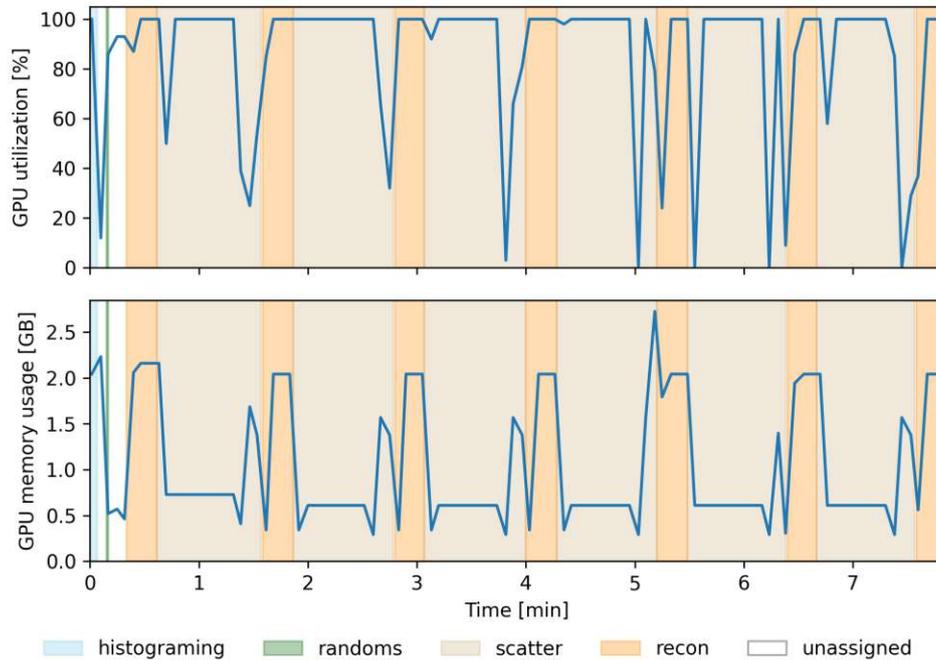


Figure 5.8: NiftyPET: GPU utilization and memory for frame 75.

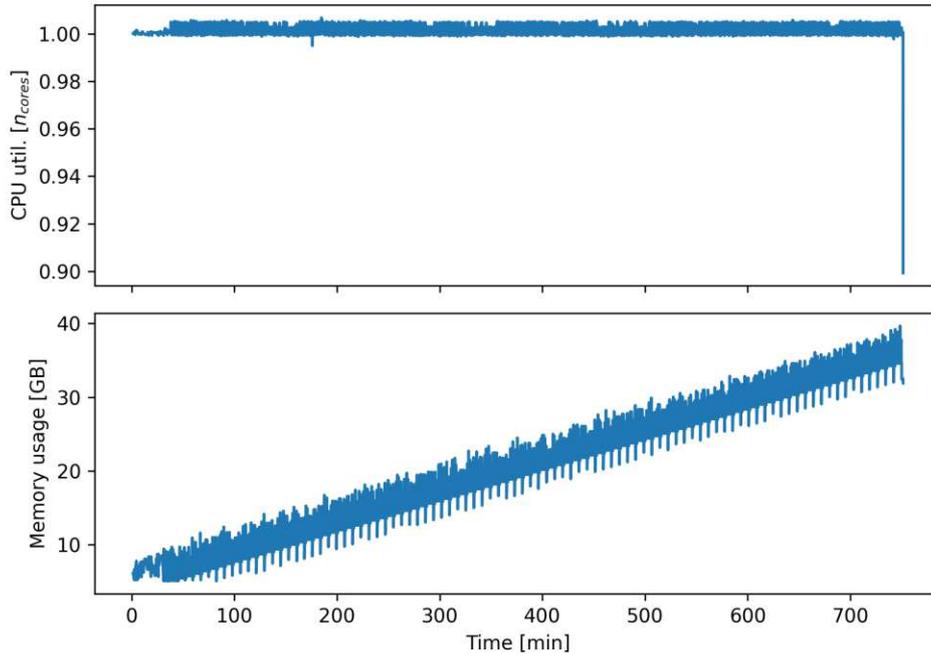


Figure 5.9: NiftyPET: CPU utilization and memory usage for the full reconstruction.

5.3 SIRF

The CPU and RAM used by SIRF to process one example frame is shown in Figure 5.10. Histogramming and randoms estimation are both executed on one CPU core. Their duration increases with later frames. The curves of the dependence of both processing times are of similar shape with an approximately constant offset of around 20 seconds as shown in Figure 5.11. Whilst the initially required time for both processes is fairly moderate, for later frames histogramming and randoms estimation contribute a very significant part to the overall processing time.

Scatter estimation is parallelized and increases the CPU usage in comparison to the preceding histogramming steps. Utilization fluctuates between twenty and slightly above fifty cores. $\sim 120.7 \pm 1.7$ seconds are spent on scatter simulation per frame. This represents an increase of more than a factor of ten compared to the duration measured for the e7-tools. However, it is approximately twice as fast as it was in the reconstruction using NiftyPET.

The reconstruction itself required $\sim 223.3 \pm 6.9$ seconds per frame. This are around 250 % more than the reference and an increase of a factor of a bit less than 25 compared to the fastest solution. During the reconstruction the CPU utilization decreased in comparison to scatter estimation and resided around $\sim 17.1 \pm 0.6$ cores.

As shown in Figure 5.10, the memory used by the reconstruction was nearly constant throughout all steps involved in the reconstruction of one frame. However, Figure 5.12 shows that it linearly increased from 6.2 GB towards later frames with 60 MB per frame, reaching a maximum of 12.2 GB at the last frame.

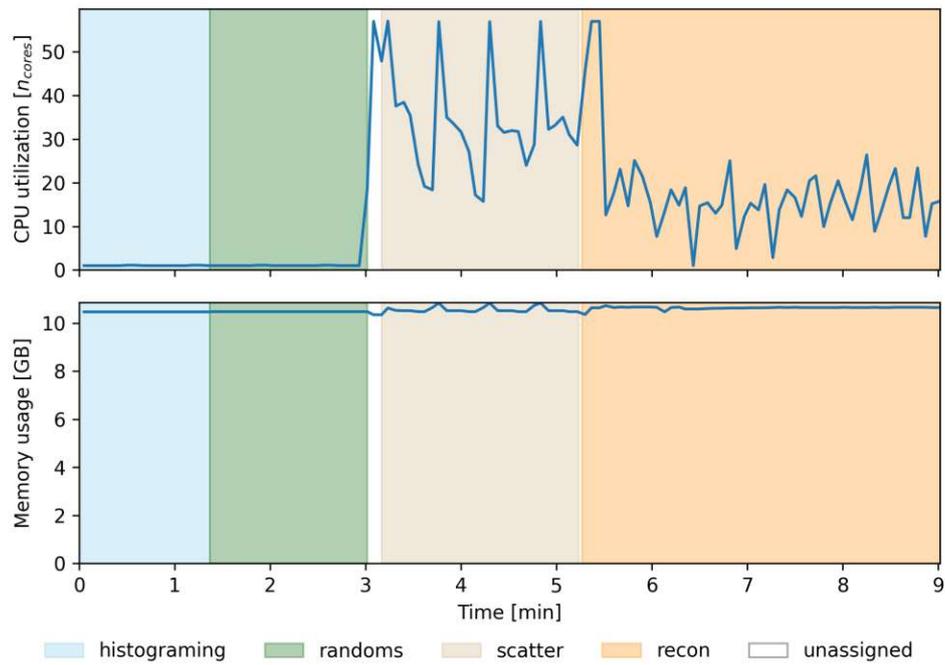


Figure 5.10: SIRF: CPU and RAM utilization for frame 75.

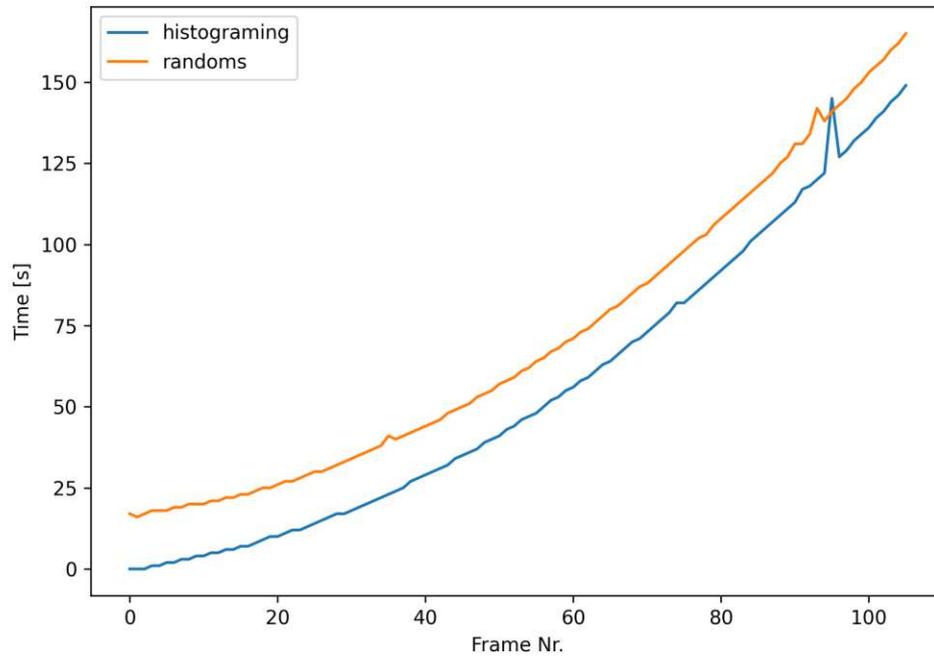


Figure 5.11: Processing time of SIRF histogramming and randoms estimation over frame number.

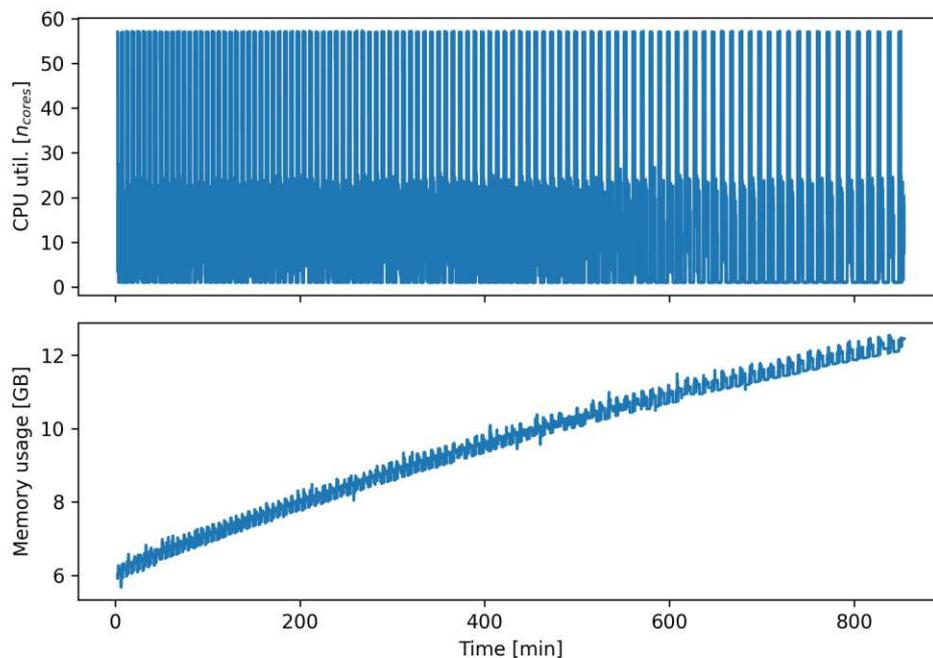


Figure 5.12: SIRF: CPU utilization and memory usage for the full reconstruction.

5.4 Image Quality

The distribution of the SNR and CNR as calculated for each brain region using Equations 4.1 and 4.2 is shown for an example frame of the series of images reconstructed with NiftyPET and the e7-tools in Figures 5.13 and 5.14. The mean SNR of NiftyPET lies slightly below the e7-tools. On the other hand the CNR of NiftyPET is slightly higher than the one produced by the e7-tools. However, in general it can be stated that the toolboxes produce similar image qualities with regard to the observed parameters. However, a visual inspection of the images, an example of which is shown in Figure 5.1, shows that NiftyPET produces different results. The images appear more blurred in comparison to the results obtained by the e7-tools.

The image quality of SIRF was excluded from the analysis because an inspection of the results has shown that the motion correction performed prior to normalization did not work as expected. This resulted in some segmented brain regions lying outside reconstructed active volume. Therefore, the SNR was overestimated in those regions.

Further investigation has shown that the affine matrix, responsible for transformation between image and scanner coordinates, written to the `.nii` file differs from the results obtained from e7-tools and NiftyPET. The tools provided by SIRF for saving the results to a file do not require any further inputs. Therefore, this indicates a problem in the `SIRF.Reg.NiftyImageData` class used for outputting the result. As the results presented in Section 5.3 do not suggest that SIRF will be a promising solution in the context of this thesis, these problems were not further investigated. A visual inspection of the results from NiftyPET showed no problems during the preprocessing for SNR and CNR calculations.

The SNR over the acquisition is shown in Figure 5.15. Again it can be seen that NiftyPET shows a very similar behavior compared to the e7-tools, both in terms of mean SNR and variance.

All toolboxes approach a maximum value of slightly above three towards the end.

Over the course of the acquisition, the CNR shown in Figure 5.16 behaves similar to the SNR. It increases sharply in the beginning and then approaches a fairly constant value, which in this case lies close to 1.5 for both the e7-tools and NiftyPET. SIRF is not shown in Figure 5.16, as the values calculated for it were not meaningful.

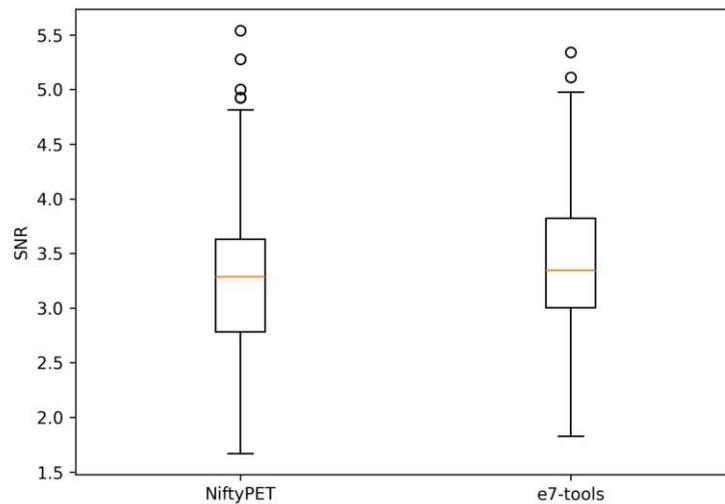


Figure 5.13: SNR across brain regions per toolbox. e7-tools are equivalent with and without GPU.

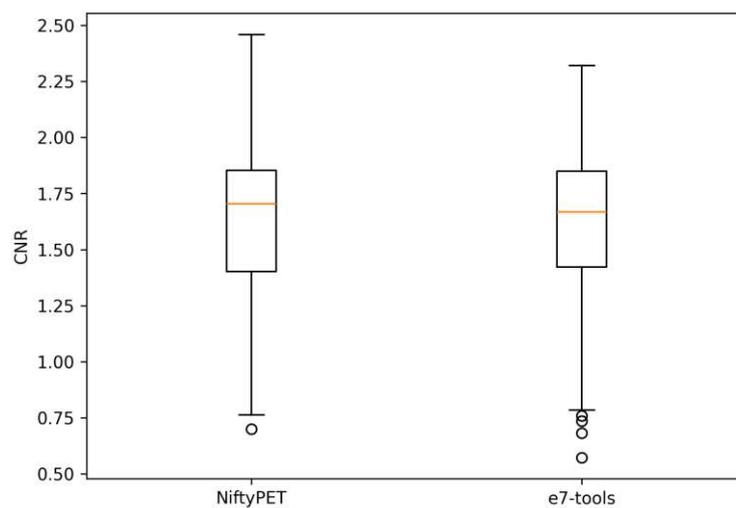


Figure 5.14: CNR across brain regions per toolbox. e7-tools are equivalent with and without GPU.

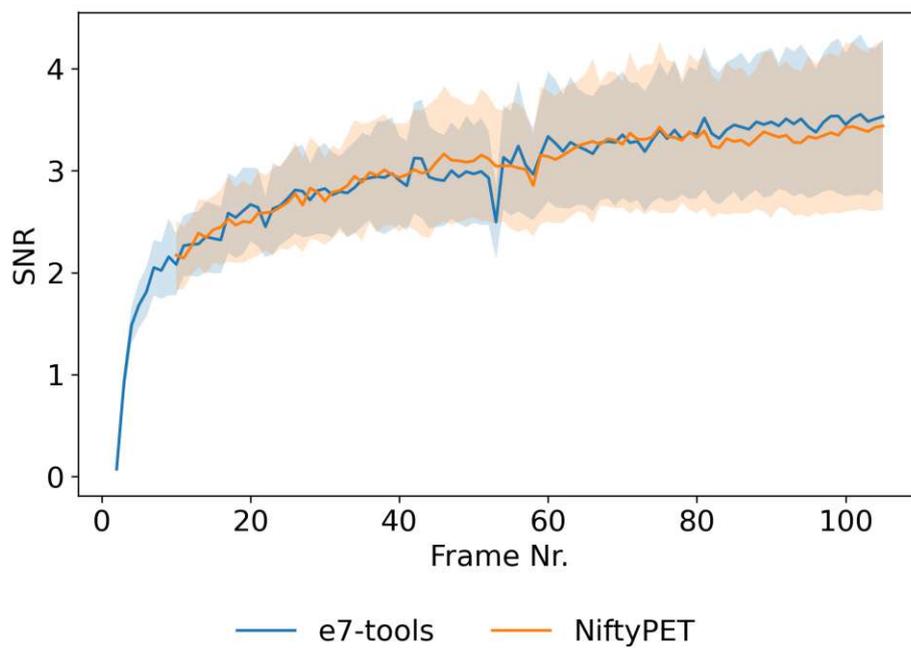


Figure 5.15: SNR over the acquisition time. Line graphs show the mean value of the SNR over brain regions. Overlays indicate a standard deviation above and below the mean.

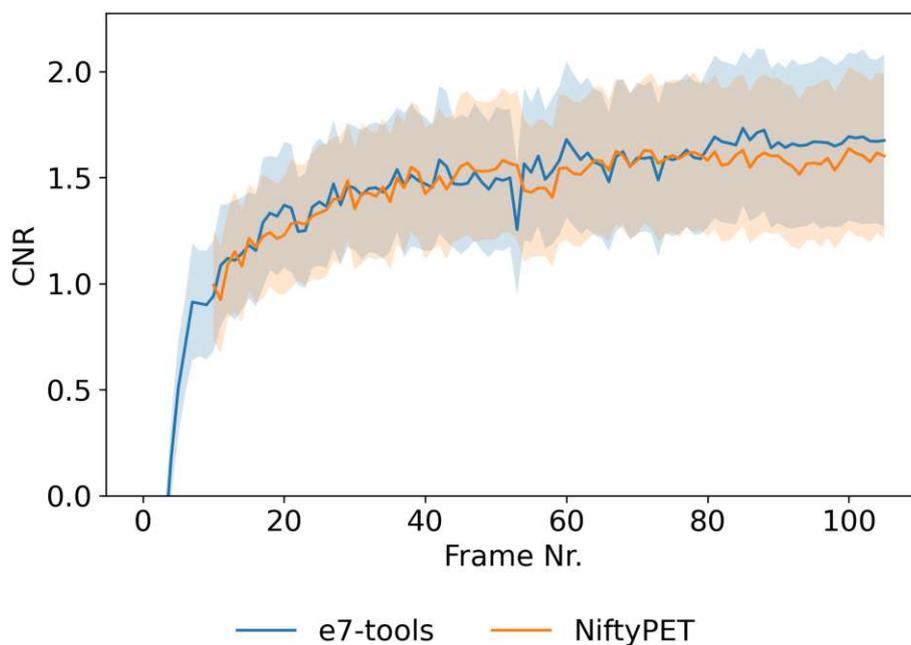


Figure 5.16: CNR over the acquisition time. Line graphs show the mean value of the CNR over brain regions. Overlays indicate a standard deviation above and below the mean.

5.5 Computational Implications of LAFOV Reconstruction

To understand the requirements and challenges of soft- and hardware used for reconstruction, a detailed understanding of the involved data volumes is necessary. This information can aid the estimation of both storage requirements and computational complexity. Furthermore, it should be possible to calculate those requirements for different scanner sizes and acquisition parameters, so that results obtained for smaller systems can be used to estimate the applicability of the used tools for LAFOV systems.

An analysis of the Michelogram shown in Figure 3.4 (A) with regard to the number of projections yields

$$n_{pd} = 2n_r - 1 \quad (5.1)$$

and

$$n_{pi} = 2(n_r - (S + 1)/2 - Si) - 1 = 2n_r - 2 - S(1 + 2i), \quad (5.2)$$

for the number of direct projections n_{pd} and oblique projections n_{pi} for spans $S > 1$ and

$$n_{pd} = n_r \quad (5.3)$$

and

$$n_{pi} = n_r - i - 1 \quad (5.4)$$

for $S = 1$, where i is the segment number to one side of the diagonal starting with 0 at the first oblique segment. The total number of axially compressed projections is therefore

$$n_p = n_{pd} + 2 \sum_{i=0}^{m-1} n_{pi}, \quad (5.5)$$

where m is the number of segments on one side of the diagonal segment in the Michelogram. When the last segment ends exactly where the maximum ring difference Δr_{max} truncates the corners of the Michelogram, so for $2\Delta r_{max} + 1 \pmod{S} = 0$, m is

$$m = \frac{2\Delta r_{max} + 1}{2S} - \frac{1}{2}. \quad (5.6)$$

For $S > 1$ Equation 5.5 expands to

$$n_p = 2n_r + 4m(n_r - 1) - 2Sm^2 - 1 \quad (5.7)$$

and to

$$n_p = n_r(2\Delta r_{max} + 1) - \Delta r_{max}(\Delta r_{max} + 1) \quad (5.8)$$

for $S = 1$, in which case $m = \Delta r_{max}$.

From this, the total number of projection bins n_g can be calculated with

$$n_g = n_p \times \frac{1}{2} n_d \times n_t, \quad (5.9)$$

where n_d is the number of detectors per ring and n_t is the number of transaxial subdivisions made in the direction of r from Figure 3.1.

To achieve an increased sensitivity with additional detector rings, the maximum ring difference needs to be close to and increased with the number of rings. When this is the case, Equations 5.7 or 5.8 show, that the number of projections grows with $O(n_r^2)$. As a consequence, also the required memory to store the projection data from histogramming grows in the same manner. In addition to the projections, memory to store the image is necessary. If the full FOV is reconstructed, the size n_f of the image grows linearly with the number of rings, due to the additional slices in axial direction.

For comparison, without axial compression, no limits on the maximum ring difference and using the specifications presented in Sections 4.1 and 4.2, Equation 5.9 yields a total of $\sim 3.15 \times 10^8$ projections for the Biograph mMR and $\sim 1.32 \times 10^{10}$ for the Biograph Vision Quadra. This assumes a constant transaxial image dimension of 344 voxels from the Biograph mMR, which could be potentially doubled in the reconstruction of the Vision Quadra [49]. Furthermore, it incorporates also a factor of almost two for the increased number of detectors per ring in the Vision Quadra. Still, the number of projections increases by two orders of magnitude and the greatest share of this can be attributed to the additional rings.

As described in Equation 3.10 and Figure 3.2 the reconstruction requires a system matrix to model the acquisition process. To achieve optimal computational performance, the best solution would be to precompute and store this matrix in memory. However, with dimensions $n_g \times n_f$ and the aforementioned storage complexities of both components, the system matrix size would grow with $O(n_r^3)$. It would need to be stored in addition to, on its own potentially large, data of projections and images. The required memory for storing a precomputed system matrix can quickly get unmanageable. Therefore, its values are commonly computed on the fly. This happens in all the used reconstruction toolboxes. SIRT and NiftyPET rely on Siddons ray tracing algorithm [50] to implement the computation.

The OSEM algorithm shown in Equation 3.11 is composed of summations over subsets of projections nested with summations over the image space. Assuming a constant number of subsets and subiterations, those operations need to be repeated for each voxel to obtain an update of the image. As the number of projections grows with $O(n_r^2)$ and the number of voxels in the image with $O(n_r)$, the number of operations necessary for an image update grows with $O(n_r^4)$.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6 Discussion

Dynamic PET refers to the process of reconstructing PET data into a series of images. When used in neuroscience to observe changes of metabolic activity in the brain upon the application of a stimulus it is referred to as fPET. It is expected that fPET with increased temporal resolution will allow researchers to better understand various processes inside the human brain. As a higher temporal resolution also leads to shorter acquisition times per frame, it reduces the number of annihilation events within a frame. Therefore, scanner systems with a higher sensitivity are necessary to ensure a sufficient image quality. LAFOV scanners provide a higher number of detector rings. As a consequence, they capture signal which would be lost otherwise and therefore provide a higher sensitivity. Moving towards these systems for fPET studies could be beneficial, however, the combination of frame by frame reconstruction with a high number of detectors comes with its own challenges on image reconstruction.

Image reconstruction software commonly relies on the OSEM algorithm to reconstruct the tracer distribution from emission measurements. As shown in Section 5.5, the number of operations required for the reconstruction of a PET image using the OSEM algorithm grows with the fourth power of the number of detector rings. For data acquired with LAFOV systems this process is executed for a particularly high number of rings. Furthermore, with increasing temporal resolution it must be repeated more often for the growing number of frames. Therefore, software and hardware used for fPET reconstruction need to be extraordinarily efficient, so that studies with many subjects can be executed in a timely manner and with reasonable resources.

The presented results provide a thorough evaluation of the current state of reconstruction software for fPET imaging. A review of the available literature for openly available reconstruction toolboxes was conducted. Although more than ten options were found, the majority of them had to be disregarded. Mostly due to compatibility issues, however, some were also outdated or insufficiently documented. The two candidates with which a reconstruction could successfully be implemented, NiftyPET and SIRF, were compared to the vendor solution e7-tools by Siemens. The comparison was conducted by reconstructing list mode data acquired with a Siemens Biograph mMR system for an fPET study [3]. All toolboxes were evaluated in terms of processing time of individual steps in the reconstruction process, such as histogramming, scatter simulation, randoms estimation and reconstruction. Furthermore, their hardware utilization and image quality was evaluated and compared. The performance parameters of the pipelines were measured over the whole course of the reconstruction process on different types of hardware. To compare the results in terms of image quality, the images were normalized, coregistered and segmented into brain regions. The SNR and CNR were computed for each region and compared.

Finally, the process of image reconstruction and its mathematical properties were outlined in the introductory chapters. To extrapolate the results towards LAFOV systems, the OSEM algorithm was analyzed in terms of its inherent computational complexity. Through this the reasons for the demanding nature of the transition to LAFOV systems were shown. As mentioned previously, the number of operations necessary for reconstruction grows with the fourth power of the number of rings. Using a LAFOV system can therefore increase the number of operations by two orders of magnitude. However, this is only the case if the full FOV is reconstructed, as it is the case currently for all tested implementations. If the reconstructed volume is restricted to the ROI of neuroscience studies, significant improvements are still achievable.

The main findings to be discussed in the following were that

- at the time of writing, the e7-tools on high-end hardware are the best available option for fPET.
- reconstruction software must be designed specifically for fPET in certain aspects.
- scatter simulation on lower image resolutions has a significant impact.
- high performance hardware has a major impact on processing time, but support is limited.
- major improvements are still possible if toolboxes would be specific to brain imaging.

Without doubt, none of the evaluated publicly available toolboxes is able to outperform the e7-tools. A big share of this advantage can be attributed to the highly optimized nature of the e7-tools. Although SIRF is implemented in C++, which is low-level and fast, the e7-tools are significantly faster in certain subtasks which operate on similar data and logic. Additionally, due to the adaptations which were necessary to allow fPET reconstruction using SIRF, it spent high amounts of time searching through the list mode data during histogramming. This emphasizes that an appropriate fPET reconstruction software must be designed for this purpose from the very beginning at least in terms of histogramming. Simply adapting an existing reconstruction routine to be executed iteratively will lead to major inefficiencies. The details of this are discussed in Section 6.1.

Significant performance differences were observed in terms of the quality of the used hardware. Most importantly, using a high performance GPU resulted in major improvements of processing time for the e7-tools. This is due to the benefits provided by the parallelization of processing. Unfortunately, GPU processing is only supported by the e7-tools for some scanner models. Although no problems were encountered enabling the GPU for a reconstruction for the Biograph mMR, a test run with data from the Vision Quadra showed that GPU accelerated reconstruction is not supported for this scanner model. Therefore, the fPET community must rely on the vendor to implement GPU support for LAFOV scanners, which would be highly favorable from a performance perspective.

Insufficient bandwidth towards the systems hard drive has shown to be a potential bottleneck if files are written to disk during processing. Reconstruction facilities should therefore in all cases rely on modern Solid State Drives (SSDs), which are connected to the system directly via the Peripheral Component Interconnect (PCI) bus, instead of older storage devices connected via Serial - ATA (SATA). Additionally, just providing better and more hardware resources is only useful if the software is also utilizing it to its best. It was found that the e7-tools do only in small parts utilize additional CPU cores on hardware with a high number of cores. The effects of different types of hardware are discussed in Section 6.3.

The results show that, if acceleration using GPUs is not sufficient, major performance improvements for fPET are still possible. Especially if implementations were to focus on the neuroscientific subdomain, improvements could be achieved. All the tested toolboxes perform a reconstruction of the full FOV of the scanner. This includes a big number of LORs which do not intersect the brain. Limiting the processing to only the LORs which are relevant could result in an acceleration of one order of magnitude for LAFOV scanners. Therefore, an optimal solution for fPET reconstruction with high temporal resolution of neuroimaging data from LAFOV systems would be very specific to this domain.

6.1 Efficient fPET Histogramming

The fact that list mode data is used for dynamic reconstruction must be kept in mind during histogramming. The PETLINK format [28] in which data is exported from the Biograph mMR, stores timing and event information as consecutive, fixed size data elements. This is typical also for other forms of storage for this kind of data. The position of a certain point in time inside the file can therefore not be known prior to reconstruction without some form of search through the file.

SIRF does support to specify a certain timeframe to use for reconstruction. However, dynamic reconstruction is not supported. Therefore, the reconstruction was executed in a loop, independently for each frame. A closer investigation of the long and increasing histogramming times shown in Figure 5.11 shows that SIRF's histogramming in combination with the aforementioned loop is inefficient for fPET. To find the starting point of the frame in the file, SIRF iterates through the events up until the specified timeframe to find the events of interest. As the number of events prior to the frame increases with later frames, also the time required to find the frame increases. Together with the fact that the activity per frame increases a bit less than linearly, the almost quadratic increase of the processing time shown in Figure 5.11 can be explained. Towards the end of the acquisition the iteration through the data consumes up to approximately 2.5 minutes per frame, which is an unacceptable amount of time. As histogramming is necessary for both prompt and delayed coincidences, the effect of this is further amplified.

In LAFOV systems and with high temporal resolution the severity of this inefficiency would be further amplified. Firstly, the higher number of rings would result in more coincidences and increase the amount of events which must be iterated. Secondly, the higher number of reconstructed frames would increase the number of times the software would iterate through the data to find a frame. It can therefore be said that histogramming for high temporal resolution LAFOV fPET must persist the position of the current frame in the data to make sure iteration can be continued from there for the next frame.

6.2 Effects of Image Resolution in Scatter Simulation

For scatter estimation the e7-tools were the by far most performant solution. SIRF required an increased processing time for scatter simulation by a factor of 10 compared to the e7-tools and NiftyPET by a factor of 25. For SIRF the scatter estimation happens prior to reconstruction on a lower resolution image. NiftyPET on the other hand uses the full resolution image for scatter simulation. As described by Equation 3.12, to estimate the scatter rate of a single pair of detectors an integration over the whole image volume is necessary. This process can be rather costly and most reconstruction implementations perform the scatter correction on a lower resolution image prior to reconstruction. Although this approach is less accurate, the lower resolution allows for significantly faster processing.

The results show that using a lower resolution image has the expected effect and can dramatically reduce the processing time. Other solutions were faster by an order of magnitude compared to the NiftyPET implementation. The GPU of the used Linux workstation is a rather small device and more performant hardware could possibly improve the performance. Still, it is not expected that the approach used by NiftyPET will have a major advantage on other hardware. Furthermore, the parameters used for image quality evaluation do not indicate that the more sophisticated method leads to significant improvements. However, the quantitative accuracy of the images cannot be evaluated in this case as this would require measurements of phantoms

with known activities. Quantitative accuracy is a major design goal of NiftyPET. Therefore, the possibility remains that the image is better from this perspective. However, in fPET task induced changes are observed instead of absolute values. As a consequence, the quantitative accuracy of results has a secondary role in this case.

It must be noted that the possibility remains that a portion of the unassigned time before the scatter estimation could be part of it due to inaccuracies in determining the durations from logs. Furthermore, individual preprocessing steps of the attenuation maps or other components could be included into the times which are labeled as scatter for other toolboxes but not for the e7-tools. Still, the advantage of the vendor implementation would be very significant also in those cases.

In conclusion, the presented results suggest that using a lower resolution image can be highly recommended for fPET reconstruction. In terms of SNR and CNR, no significant differences in image quality between the two approaches were found. However, if high requirements on quantitative accuracy are necessary, then the trade-off between this and shorter processing times should be reevaluated.

6.3 Parallelization and Hardware Utilization

A comparison between the high-end GPU version of the e7-tools reconstruction and others is an excellent example of the acceleration factor which can be achieved using parallelization. As shown in Table 5.3, the main reconstruction on the high-end GPU overall only requires 16 minutes and is therefore almost seven times faster than the reference running only on CPUs. This significant acceleration is due to the high level of parallelization which is achieved on the GPU used in the high performance Windows machine. In contrast, when the reconstruction is executed on the CPU, it can be seen in Figure 5.5 that the e7-tools do not fully expand processing to fit the increased number of available cores on the high-end hardware. Therefore, the transition to more performant hardware for reconstructions with the e7-tools running only on the CPU did only result in a minor performance improvement.

Just executing the reconstruction on any GPU, however, is not sufficient and high-performance hardware should be used. A comparison between the low-performance setup with and without GPU shows that in this case enabling GPU processing increased the time required to reconstruct the images. The comparably small GPU of this hardware configuration is not able to accelerate the processing compared to an execution on the CPU. Likely, this is due to additional operations involved in copying data from the host memory to the GPU and similar operations which are necessary for the computations. The acceleration achieved by the specialized hardware needs to compensate for the additional operations required for moving data to the device.

An analysis of the processing times leads to the observation that parallelization has to work together with optimized software to fully utilize its advantages. Adding parallelization to inefficient software can alleviate performance issues. However, optimization must happen from both perspectives for optimal results. For example, SIRF had many more CPU cores available than the reference implementation and also used them to a greater extent. The hardware showed particularly high utilization during scatter simulation and also during reconstruction the number of used cores was above the reference. Nevertheless, SIRF was not faster than the e7-tools in neither scatter simulation, nor the main reconstruction. Therefore, it can be concluded that the implementation and code behind SIRF have inefficiencies which are not present in the e7-tools. As mentioned, dynamic reconstruction is not supported by default and the major aim of the

toolbox is to support many algorithms, scanner geometries and to be portable to many systems [19]. As a consequence, it seems likely that a focus on performance was not relevant for the project.

A bottleneck which was encountered in the low performance Windows machine was a SSD with insufficient bandwidth. As it was connected via SATA instead of NVMe PCI, the low-end setup took almost forty minutes for histogramming. In comparison, on other systems this process required one minute. Due to the limited bandwidth of the SATA connection, writing sinograms to disk took significantly longer. To truly optimize the reconstruction, it must therefore be ensured that all components with major contact to processing provide an optimal performance.

From the described findings it can be concluded, that parallelization certainly has to play a major role for efficient dynamic PET reconstruction. Especially towards reconstruction of data from LAFOV systems with high temporal resolution, the significant acceleration of a high-end GPU will be necessary. However, the parallelization has to go hand in hand with an optimization of the underlying algorithms and other hardware to avoid bottlenecks. Additionally, using just any GPU can even have a detrimental effect on processing time and care should be taken in selecting the appropriate hardware. High performance hardware can have major effects. However, it must be ensured that the used software properly utilizes the available resources.

6.4 Memory Leaks

An inspection of the memory utilization of the toolboxes has uncovered increases in memory consumption over the course of reconstruction. While SIRF showed a moderate increase of 60 MB per frame, the memory used by the e7-tools grew with 180 MB and for NiftyPET even with 330 MB per frame.

The amount of data to store sinograms and the image is equivalent for each frame. Therefore, if other data which can accumulate, such as for example logs, is cleaned and stored to disk periodically, the only reason for the memory to increase is the higher number of counts for the emission data. In the case of the e7-tools this data is not necessary in the main reconstruction, as histogramming happened earlier, and the increase in memory therefore suggests that certain portions of memory are not deallocated properly and produce a memory leak. For SIRF this possibility can be excluded as the reconstruction is restarted after each frame. In the case of NiftyPET it is possible that the increase in memory utilization over time is due to the increased amount of emission data. However, it seems fairly high and the memory consumption should be closely monitored if the toolbox is adapted for higher temporal resolutions.

Overall it is clear that a significant memory leak can badly impact the reconstruction. Although it may stay unnoticed at first, problems could be caused when the temporal resolution is increased. Especially, if memory allocated for the processing of individual frames is not deallocated, the consequences of the leak will worsen with a higher number of frames. Therefore, to avoid any surprises, the memory utilization of reconstruction toolboxes should be closely monitored to avoid any surprises when transitioning to more demanding jobs with high temporal resolution and on scanners with a LAFOV. Especially on this type of scanners the problems could be further amplified due to the higher volumes of data.

6.5 Disregarded Toolboxes

A major reason for the decisions to disregard toolboxes were compatibility issues. Often, they require a specific file format and are therefore only able to operate using an appropriate format converter. Those converters are in many cases not available. The ones tested throughout this project are the conversion utilities from the `pet-rd-tools` [31], the `STIR2PETSIRD` [38] converter and the Castor datafile converter to convert sinograms created by the `e7-tools`. Although the first two yielded usable results, their processing times were very significant. The Castor toolbox was excluded from the analysis due to the unacceptably long processing time of the converter. Although the time required for the conversion is not part of the presented results, it can contribute a significant amount to the processing time. Therefore, even if the reconstruction itself would be performant, the converter would render the toolbox unusable without further efforts to optimize the file conversion.

On the contrary, NiftyPET was easier to use as it did not require a format conversion prior to reconstruction and operated directly on the provided PETLINK data. Although this is convenient, looking into the source code of the toolbox shows that it is highly coupled to the Biograph mMR. As an adaptation of hard-coded parameters, such as the number of subsets, has shown to be a highly complex task, it is expected to be close to impossible to adapt the code to a different scanner system. In general, a shared data format and toolboxes flexible to the scanner system would therefore be highly favorable. The fairly new PETSIRD standard could therefore be a promising option to improve the situation and standardize the format. However, as it can be seen in the used converters, such as [38], the converters still need time to mature. Generally, we would advise toolbox developers to put more emphasis into the design and development of data formats and conversion tools. It seems like this has usually been dealt with as a secondary topic and the solutions are therefore not fit for use with larger volumes of data.

Apart from compatibility issues, many toolboxes are primarily employed in clinical settings where a frame by frame reconstruction is rarely necessary. In this study this was for example the case for PyTomography. The toolbox made a promising impression, and it was possible to set up a basic reconstruction. However, the toolbox is currently not designed to support the reconstruction of subsets of the data. Also, as described in Section 4.5.4, the toolbox had very demanding memory characteristics. Although those would likely not be a major issue in a clinical setting with shorter acquisition times, they will certainly be in case of LAFOV list mode data with measurement durations of one hour and more. Also, many other toolboxes were advertised for mainly clinical purposes.

Generally speaking, the requirements for clinical reconstruction which are relevant to most available toolboxes are different to fPET. Therefore, to achieve truly optimal results in the context of this domain, it seems that efforts towards a custom solution from within the community using it would be the fastest path to improvements.

Lastly, sufficient documentation is essential for a toolbox to be applicable for users outside the vicinity of the development team. It must provide a clear path forward for setting up reconstruction pipelines. Although it might be a straight-forward task for toolbox developers to use their software, users will not be willing to engage into finding out how to use it. Clear and understandable documentation facilitates the reuse of software and therefore avoids unnecessary work throughout the community. Furthermore, if not used directly, well documented software can serve as a learning resource for researchers and developers, to understand the actual logic involved in PET image reconstruction.

6.6 Image Quality

As it can be seen in Figure 5.1, the reconstructed images differ significantly between the toolboxes. While the result provided by the e7-tools shows the most granular details, NiftyPET provides fewer details and the image is blurred even more for the result obtained with SIRF. The differences are caused due to two different reasons for the two toolboxes.

In the case of NiftyPET, it was not possible to configure the toolbox with the exactly same parameters as the e7-tools. The number of subsets is hard-coded into the source code of the toolbox and an attempt to modify it was unsuccessful. Therefore, although we tried to match the number of iterations in a way that results in the same number of total iterations, it cannot be guaranteed that this results in the exact same image.

For SIRF, during the first test runs of the reconstruction, problems with the scatter data were encountered. The scatter sinograms had very high values, which led to strongly malformed reconstructions. It was first wrongly concluded that the problem are the units of the scatter sinograms. They were therefore divided by 1000, which led to seemingly correct reconstructions. However, the actual underlying reason was a misunderstanding of the data used to generate attenuation maps. To generate them, a pseudo-CT computed from a T1 weighted MRI acquisition as described in [14] was used. When configuring the reconstruction with SIRF this pseudo CT was mistakenly taken to be equivalent to a special magnetic resonance attenuation correction (MRAC) acquisition. It was converted to a μ -map using the `nm_mrac2mu` utility provided by the `pet-rd-tools` [31]. As those are in fact two different approaches on generating attenuation factors from MRI measurements, the resulting values for the μ -map were incorrect. A review of the possibilities for calculating attenuation factors from MRI images was published by Wagenknecht *et al.* in [13] and gives further details on the differences of the two methods. The scatter sinograms used for the reconstruction with SIRF are therefore incorrect. Furthermore, SIRF had problems with the image geometry when exporting the results to a file. As a consequence, the image on the right side of Figure 5.1 appears stretched.

Both of the described problems lead to differences in the resulting image. However, they do not impact the general toolbox performance. Furthermore, it was found that both toolboxes are not able to compete with the e7-tools in terms of performance. Therefore, no further efforts were invested into resolving the issues.

6.7 Future Improvements in fPET Reconstruction

The clear advantages of the e7-tools shown in this thesis lead to the conclusion that this toolbox is currently the best option for dynamic PET reconstruction. Although SIRF and NiftyPET were able to reconstruct the data from the Biograph mMR, the measured performance characteristics are far from what would be necessary for efficient high temporal resolution LAFOV fPET.

That said, it is certainly not the case that the current state of the e7-tools is optimal. It was found that the GPU acceleration of the e7-tools does work for the Biograph mMR but not for the Vision Quadra. For this example of a LAFOV system, where the GPU acceleration could have the best impact, researchers will have to wait for improvements in a future release of the e7-tools. Alternatively, an attempt to implement a new and more performant toolbox would be possible. However, due to the highly optimized nature of the e7-tools, achieving similar performance will be a challenging task. Furthermore, to achieve this the choices in histogramming, scatter simulation and other aspects of the algorithms as they were discussed previously need to be evaluated closely for optimal performance and conscious about the involved trade-offs.

Beyond the mentioned factors influencing processing time, a reconstruction implementation specific to brain imaging could result in further improvements in case the GPU acceleration is not sufficient by itself. A common property of all the analyzed toolboxes is, that the full FOV is reconstructed and also considered as a potential source of emissions during scatter simulation. However, in neuroscience, the main motivation of using LAFOV scanners is the increased sensitivity. Most of the FOV is therefore not of interest and could be excluded from the reconstruction. This would result in a dramatic reduction of the number of voxels in the reconstruction and subsequently the necessary number of operations in the computation.

The growth in computational complexity of $O(n_r^4)$, as it resulted from considerations in Section 5.5, assumed that the number of voxels in the FOV grows linearly with n_r . This is only the case if the full FOV is reconstructed. Holding the number of reconstructed voxels in axial direction constant while increasing n_r would result in a reduction of the exponent of the computational complexity by two as the number of operations in Equation 3.11 would only increase in terms of the number of projections. For example, moving from the Biograph mMR to the Vision Quadra Scanner would then only increase the number of operations for reconstruction by a factor of 25 instead of 625. The number of voxels included into the reconstruction could further be reduced by excluding voxels in the transaxial dimensions through the use of anatomical information from MRI, CT or simple positional information about the patient table.

The described points of improvement are certainly very specific to the domain of neuroscience. Furthermore, utilizing them for improved performance will likely only be necessary in the case of fPET on LAFOV systems, as clinical applications and other less demanding types of studies do not have as high requirements on performance and relatively longer processing times are usually not an issue. Additionally, the effort towards an exclusion of parts of the FOV should only be made in case the acceleration using a GPU is not sufficient.

7 Conclusion and Outlook

Dynamic PET is an imaging method where a series of images is reconstructed from a PET acquisition. When this method is used with special tracer administration protocols, to observe task induced changes in molecular activity, it is referred to as fPET [1, 2]. It is expected that higher temporal resolutions will lead to a better understanding of processes inside the human brain. Although current studies use frame lengths between 3 and 30 seconds [3, 5, 7], possibilities to go below one second are investigated. Shorter frame durations decrease the signal magnitude. Therefore, PET scanners with higher sensitivity must be used. LAFOV scanners are clinically of interest due to their extended range of vision. In neuroscience, their high number of detectors could provide the necessary additional sensitivity. The increased number of detectors captures signal which would be lost otherwise. However, it also drastically increases the amount of calculations required to derive an image from the emission data. Furthermore, in the case of dynamic reconstruction, these calculations need to be repeated once per frame. Reconstructions of images acquired with LAFOV systems and high temporal resolutions are therefore computationally very demanding.

Most modern reconstruction software is designed to be used for clinical applications which rely on static images. In most cases, the software toolboxes provided by the scanner vendor are used. However, also research software is openly available. Prior to this thesis it was generally not known if other solutions are possibly better suited for performant frame by frame reconstruction. Especially in the case of LAFOV systems, the requirements on soft- and hardware are demanding and only an optimal setup will result in reconstruction times applicable for modern research. Therefore, this thesis conducted a thorough review of the available literature and identified reconstruction toolboxes which could be useful for reconstructing fPET data from LAFOV systems.

After an initial screening of all toolboxes, the most promising ones were used to set up a reconstruction pipeline. Although many potential candidates of open software for PET reconstruction were found, most of them had to be excluded. The reason for this mostly were compatibility problems with the input data formats. Additionally, most toolboxes are designed for clinical applications which usually do not require dynamic reconstruction. Finally, two toolboxes, NiftyPET [18] and SIRF [29], were used to set up a reconstruction pipeline. The performance of the toolboxes was compared to a reconstruction executed using the e7-tools by Siemens. Additionally, the reconstruction using the e7-tools was run on varying hardware configurations to analyze the improvements provided by a higher number of available CPU cores and GPUs with varying sizes.

The results show that neither NiftyPET nor SIRF is able to outperform the e7-tools in terms of processing efficiency when used for dynamic reconstruction. The reasons behind this are a combination of various differences. Generally, we can conclude that the e7-tools are a highly optimized software toolbox which will remain very challenging to surpass. For example, SIRF is implemented in C++ and highly parallelized. In comparison, the e7-tools utilized less CPU cores and were still able to perform the reconstruction almost four times faster. This indicates a very thorough optimization of the processing inside this software. Furthermore, the e7-tools performed even better on a high-end GPU, which accelerated processing by a factor of two to three and was even more than ten times faster than NiftyPET and SIRF.

It was found that especially the process of transforming emission data into sinograms, histogramming, must be designed with dynamic PET in mind. Software which was not originally built for this use-case can be adapted to support it by executing it in a loop. However, im-

portant information about the position of frames inside the list mode data is lost after each iteration. As a consequence, the starting position of later frames must be found by searching the data before processing each frame, which is time-intensive and cumulatively makes up for a significant amount of processing time. For LAFOV systems and high temporal resolutions, this accumulated effect would be dramatically worse due to the higher number of events which would be iterated more often for a higher number of frames. For this work, SIRF was adapted in the described way, which led to increased time requirements for histogramming. Especially in later frames the effect led to unnecessary processing times of two minutes per frame and more.

Another factor with important implications on processing efficiency is scatter correction. Modern reconstruction routines commonly rely on SSS. This model simulates scatter under the assumption that at most one of the photons forming a LOR is scattered once. For each detector pair, the simulation requires an integration over the whole image volume in addition to an intermediate reconstruction. The reconstruction and simulation are executed alternately to iteratively improve the image. It seems logical to use the main image for the intermediate reconstructions and simulations. However, the aforementioned integration over the image volume tends to make this process rather costly. Therefore, most implementations perform scatter simulation prior to the main reconstruction on an image with lower resolution. Although this leads to a loss in accuracy, the processing will require significantly less time.

The effect of operating on a downscaled image for scatter simulation can clearly be observed by the results of this work. In comparison to the other two toolboxes, NiftyPET operates on the final image resolution in all cases. Overall, it spent 08:44 on scatter simulation. In comparison to SIRF, this is more than two times the processing time and compared to the e7-tools it took longer by as much as a factor of twenty. A clear advantage of the higher resolution used by NiftyPET in terms of image quality could not be observed. However, the methods used for the evaluation do not allow any conclusions about the quantitative accuracy of the results, which could be different. The results presented herein suggest that the use of a lower resolution image is beneficial for fPET imaging, as it requires significantly less time to process.

To obtain a better understanding of the implications of a transition towards LAFOV systems, an analysis of the theoretical computational complexity of the involved calculations was conducted. Especially, the dependency of this complexity on the number of detector rings was considered. For an update of a voxel value through the OSEM algorithm, nested summations over the image and projections are necessary. Assuming equal ring and detector dimensions, the number of voxels in the image increases linearly, and the number of projections quadratically, with the number of detector rings. As this nested summation is executed for each voxel in the image, the number of operations necessary for an image update increases with $O(n_r^4)$, where n_r is the number of rings in the scanner. This explains why a step towards LAFOV systems comes with such a sharp increase in processing time or computing resources. For example, a transition from the Siemens Biograph mMR with 64 rings to the Siemens Biograph Vision Quadra with 320 rings increases the number of involved operations by a factor of 625, assuming that all other image dimensions stay the same. In reality this factor is even higher due to the smaller detectors used in the Vision Quadra. The actual increase in processing time caused by this additionally depends on other factors specific to the processing of those operations, such as parallelization. The reconstruction of a high number of frames from a system with many rings therefore requires the high performance which in this work could only be achieved by the e7-tools.

Although the e7-tools are highly optimized, it seems possible to further improve the performance of reconstruction on LAFOV systems for neuroimaging. However, only if certain anatom-

ical information of the brain is included into the process. The discussed dependencies of the computational complexity on the number of detector rings assume that the axial image dimension grows with the number of rings. However, due to the long extent of the axial FOV, the brain only occupies a small portion of the reconstructed volume. If the image size is held constant when the number of rings is increased, the growth of operations could be limited to $O(n_r^2)$. Currently, all used toolboxes reconstruct the full FOV and therefore cannot take advantage of this. As this would require major implementation changes inside the toolboxes the implementation of this would require significant efforts.

In conclusion, it can be stated that the e7-tools are currently the only available solution for reconstruction of data acquired on Siemens LAFOV systems. Although many other solutions are available publicly, most of them come with major compatibility problems and the ones which could be set up require a significantly longer time for processing. One reason for this is inefficient histogramming, which must be implemented specific to dynamic PET. Furthermore, different approaches to scatter simulation, which require an integration over the full image volume and can be optimized by operating on a lower resolution image. Apart from this, it was found that the e7-tools are inherently better optimized than other tools and will therefore remain challenging to surpass in terms of performance. Although these results do not suggest that the performance of reconstruction will further improve, it was found that advances are still possible if the FOV is limited to the brain. Although this would require major efforts in implementation, it can be expected to bring the cost and effort involved in neuroscience using LAFOV PET with high temporal resolution down considerably.

References

- [1] M. Villien, H.-Y. Wey, J. B. Mandeville, C. Catana, J. R. Polimeni, C. Y. Sander, N. R. Zürcher, D. B. Chonde, J. S. Fowler, B. R. Rosen, and J. M. Hooker. “Dynamic functional imaging of brain glucose utilization using fPET-FDG”. In: *NeuroImage* 100 (2014), pp. 192–199. DOI: 10.1016/j.neuroimage.2014.06.025.
- [2] A. Hahn, G. Gryglewski, L. Nics, M. Hienert, L. Rischka, C. Vranka, H. Sigurdardottir, T. Vanicek, G. M. James, R. Seiger, A. Kautzky, L. Silberbauer, W. Wadsak, M. Mitterhauser, M. Hacker, S. Kasper, and R. Lanzenberger. “Quantification of Task-Specific Glucose Metabolism with Constant Infusion of 18F-FDG”. In: *Journal of Nuclear Medicine* 57 (2016), pp. 1933–1940. DOI: 10.2967/jnumed.116.176156. eprint: <https://jnm.snmjournals.org/content/57/12/1933.full.pdf>.
- [3] A. Hahn, M. Breakspear, L. Rischka, W. Wadsak, G. M. Godbersen, V. Pichler, P. Michenthaler, T. Vanicek, M. Hacker, S. Kasper, R. Lanzenberger, and L. Cocchi. “Reconfiguration of functional brain networks and metabolic cost converge during task performance”. In: *eLife* 9 (2020). Ed. by T. Yeo, T. E. Behrens, and J. D. Bijsterbosch, e52443. DOI: 10.7554/eLife.52443.
- [4] A. Hahn, M. B. Reed, V. Pichler, P. Michenthaler, L. Rischka, G. M. Godbersen, W. Wadsak, M. Hacker, and R. Lanzenberger. “Functional dynamics of dopamine synthesis during monetary reward and punishment processing”. In: *Journal of Cerebral Blood Flow & Metabolism* 41 (2021). PMID: 34053336, pp. 2973–2985. DOI: 10.1177/0271678X211019827. eprint: <https://doi.org/10.1177/0271678X211019827>.
- [5] A. Hahn, M. B. Reed, C. Vranka, G. M. Godbersen, S. Klug, A. Komorowski, P. Falb, L. Nics, T. Traub-Weidinger, M. Hacker, and R. Lanzenberger. “High-temporal resolution functional PET/MRI reveals coupling between human metabolic and hemodynamic brain response”. In: *European Journal of Nuclear Medicine and Molecular Imaging* 51 (2023). DOI: 10.1007/s00259-023-06542-4.
- [6] I. Velikyan. “Prospective of ⁶⁸Ga-Radiopharmaceutical Development”. In: *Theranostics* (4 2014), pp. 47–80. DOI: 10.7150/thno.7447.
- [7] S. D. Jamadar, P. G. D. Ward, E. X. Liang, E. R. Orchard, Z. Chen, and G. F. Egan. “Metabolic and Hemodynamic Resting-State Connectivity of the Human Brain: A High-Temporal Resolution Simultaneous BOLD-fMRI and FDG-fPET Multimodality Study”. In: *Cerebral Cortex* 31 (2021), pp. 2855–2867. DOI: 10.1093/cercor/bhaa393. eprint: <https://academic.oup.com/cercor/article-pdf/31/6/2855/38555181/bhaa393.pdf>.
- [8] S. R. Cherry, J. A. Sorenson, and M. E. Phelps. *Physics in Nuclear Medicine - 4th edition*. Elsevier Saunders, Philadelphia, 2012.
- [9] E. Samei and D. J. Peck. *Hendee’s Physics of Medical Imaging*. John Wiley & Sons Inc., 2019.
- [10] M. N. Wernick, J. N. Aarsvold, and et al. *Emission Tomography, the Fundamentals of PET and SPECT*. Elsevier Academic Press, 2004.
- [11] D. L. Bailey, D. W. Townsend, P. E. Valk, and M. N. Maisey. *Positron Emission Tomography - Basic Sciences*. Springer-Verlag London Limited, 2004.

- [12] H. Zaidi and K. F. Koral. “Scatter modelling and compensation in emission tomography”. In: *European Journal of Nuclear Medicine and Molecular Imaging* (31 2004). DOI: 10.1007/s00259-004-1495-z.
- [13] G. Wagenknecht, H.-J. Kaiser, F. M. Mottaghy, and H. Herzog. “MRI for attenuation correction in PET: methods and challenges”. In: *Magn Reson Mater Phy* 26 (2013), pp. 99–113. DOI: 10.1007/s10334-012-0353-4.
- [14] N. Burgos, M. J. Cardoso, K. Thielemans, M. Modat, S. Pedemonte, J. Dickson, A. Barnes, R. Ahmed, C. J. Mahoney, J. M. Schott, J. S. Duncan, D. Atkinson, S. R. Arridge, B. F. Hutton, and S. Ourselin. “Attenuation Correction Synthesis for Hybrid PET-MR Scanners: Application to Brain Studies”. In: *IEEE Transactions on Medical Imaging* 33 (2014). DOI: 10.1109/tmi.2014.2340135.
- [15] T. Jones and D. Townsend. “History and future technical innovation in positron emission tomography”. In: *Journal of Medical Imaging* (4 2017). DOI: 10.1117/1.JMI.4.1.011013.
- [16] B. Bendriem and D. W. Twonsend. *The Theory and Practice of 3D PET*. Springer-Science+Business Media, B.V., 1998. DOI: 10.1007/978-94-017-3475-2.
- [17] F. H. Fahey. “Data Acquisition in PET Imaging”. In: *Journal of Nuclear Medicine Technology* 30 (2002), pp. 39–49.
- [18] P. J. Markiewicz, M. J. Ehrharde, K. Erlansson, P. J. Noonan, A. Barnes, J. M. Schott, D. Atkinson, S. R. Arridge, B. f. Hutton, and S. Ourselin. “NiftyPET:a High-throughput Software Platform for High Quantitative Accuracy and Precision PET Imaging and Analysis”. In: *Neuroinformatics* (16 2017), pp. 95–115. DOI: 10.1007/s12021-017-9352-y.
- [19] K. Thielemans, C. Tsoumpas, M. Sanida, T. Beisel, P. Aguiar, N. Diakaios, and M. W. Jacobson. “STIR: software for tomographic image reconstruction release 2”. In: *Physics in Medicine and Biology* (57 2012), pp. 867–883. DOI: 10.1088/0031-9155/57/4/867.
- [20] M. A. Viergever, O. D. Faugeras, J. J. Koendeink, P. S. M., S. Tsuji, and S. W. Zucker. *Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*. Springer Science+Business Media Dordrecht, 1996. DOI: 10.1007/978-94-015-8749-5.
- [21] F. Dellekart. *PET-Toolbox-Comparison*. <https://github.com/fdellekart/PET-Toolbox-Comparison>. Accessed: 2025-01-13.
- [22] G. Delso, S. Fürst, B. Jakoby, R. Ladebeck, C. Ganter, S. G. Nekolla, M. Schwaiger, and S. I. Ziegler. “Performance Measurements of the Siemens mMR Integrated Whole-Body PET/MR Scanner”. In: *Journal of Nuclear Medicine* 52 (2011), pp. 1914–1922. DOI: 10.2967/jnumed.111.092726. eprint: <https://jnm.snmjournals.org/content/52/12/1914.full.pdf>.
- [23] Siemens Medical Solutions USA Inc. *Biograph mMR*. <https://www.siemens-healthineers.com/en-us/magnetic-resonance-imaging/mr-pet-scanner/biograph-mmr>. Accessed: 2024-10-31.
- [24] G. A. Prenosil, H. Sari, M. Fürstner, A. Afshar-Oromieh, K. Shi, A. Rominger, and M. Hentschel. “Performance Characteristics of the Biograph Vision Quadra PET/CT System with a Long Axial Field of View Using the NEMA NU 2-2018 Standard”. In: *Journal of Nuclear Medicine* 63 (2022), pp. 476–484. DOI: 10.2967/jnumed.121.261972. eprint: <https://jnm.snmjournals.org/content/63/3/476.full.pdf>.

- [25] J. van Sluis, J. de Jong, J. Schaar, W. Noordzij, P. van Snick, R. Dierckx, R. Borra, A. Willemsen, and R. Boellaard. “Performance Characteristics of the Digital Biograph Vision PET/CT System”. In: *Journal of Nuclear Medicine* 60 (2019), pp. 1031–1036. DOI: 10.2967/jnumed.118.215418. eprint: <https://jnm.snmjournals.org/content/60/7/1031.full.pdf>.
- [26] Siemens Medical Solutions USA Inc. *Biograph Vision Quadra*. <https://www.siemens-healthineers.com/en-us/molecular-imaging/pet-ct/biograph-vision-quadra>. Accessed: 2025-02-05.
- [27] D. Merkel. “Docker: lightweight linux containers for consistent development and deployment”. In: *Linux journal* 2014 (2014), p. 2.
- [28] Siemens Medical Solutions USA Inc. *PETLINK - A proposed digital interconnect standard for data acquisition in nuclear medicine*. <https://www.siemens-healthineers.com/en-us/molecular-imaging/petlink-documents>. Accessed: 2024-10-31.
- [29] E. Ovtchinnikov, R. Brown, C. Kolbitsch, E. Pasca, C. da costa-Luis, A. G. Gillman, B. A. Thomas, N. Efthimiou, J. Mayer, P. Wdhwa, M. J. Ehrhardt, S. Ellis, J. S. Jorgensen, J. Matthews, C. Prieto, A. J. Reader, C. Tsoumpas, M. Turner, D. Atkonson, and K. Thielemans. “SIRF: Synergistic Image Reconstruction Framework”. In: *Computer Physics Communications* (249 2020). DOI: 10.1016/j.cpc.2019.107087.
- [30] K. Thielemans, C. Tsoumpas, D. Sauge, C. Labbe, C. Morel, M. Jacobson, A. Zverovich, T. Beisel, C. Falcon, R. Twyman, D. Deidda, and M. Strugari. *STIR - Software for Tomographic Image Reconstruction - User’s Guide Version 6.2*. <https://stir.sourceforge.net/documentation/STIR-UsersGuide.pdf>. Accessed: 2024-11-20.
- [31] B. Thomas, K. Thielemans, A. Gillman, R. Brown, and C. Badger. *UCL/pet-rd-tools: v2.0.1*. Version v2.0.1. 2020. DOI: 10.5281/zenodo.4136503.
- [32] M. Brett et al. *nipy/nibabel: 5.3.1*. Version 5.3.1. 2024. DOI: 10.5281/zenodo.13936989.
- [33] Nilearn contributors et al. *nilearn*. Version 0.10.4. 2024. DOI: 10.5281/zenodo.10948303.
- [34] M. Jenkinson, C. F. Beckmann, T. E. J. behrens, M. W. Woolrich, and S. M. Smith. “FSL”. In: *NeuroImage* (62 2011), pp. 782–790. DOI: 10.1016/j.neuroimage.2011.09.015.
- [35] T. Merlin, S. Stute, D. Benoit, J. Bert, T. Carlier, C. Comtat, M. Filipovic, F. Lamare, and D. Visvikis. “CASToR: a generic data organization and processing code framework for multi-modal and multi-dimensional tomographic reconstruction”. In: *Physics in Medicine and Biology* 63 (2018). DOI: 10.1088/1361-6560/aadac1.
- [36] Medisip, Ghent University. *PET image reconstruction software QETIR*. <https://www.ugent.be/ea/ibitech/en/research/medisip/software-lab/software-lab13.htm>. Accessed: 2024-11-4.
- [37] L. Polson, R. Fedrigo, C. Li, M. Sabouri, O. Dzikunu, S. Ahamed, A. Rahmim, and C. Uribe. *PyTomography: A Python Library for Quantitative Medical Image Reconstruction*. 2023. arXiv: 2309.01977 [physics.med-ph].
- [38] ETSI. *STIR2PETSIRD*. <https://github.com/ETSIhackers/STIR2PETSIRD>. Accessed: 2024-11-4.
- [39] S. Pedemonte, C. Catana, and K. Van Leemput. “An Inference Language for Imaging”. In: *Bayesian and graphical Models for Biomedical Imaging*. Springer, 2014, pp. 61–72.

- [40] V. V. Wettenhovi, M. Vauhkonen, and V. Kolehmainen. “OMEGA - open-source emission tomography software”. In: *Physics in Medicine & Biology* 66 (2021). DOI: 10.1088/1361-6560/abe65f.
- [41] T. Kösters, K. P. Schäfers, and F. Wübbeling. “EMRecon: An Expectation Maximization Based Image Reconstruction Framework for Emission Tomography Data”. In: *IEEE Nuclear Science Symposium Conference Record* (2011). DOI: 10.1109/NSSMIC.2011.6153840.
- [42] J. J. Scheins and H. Herzog. “PET REconstruction Software TOOLkit - PRESTO A Novel, Universal C++ Library for Fast, Iterative, Fully 3D PET Image Reconstruction using Highly Compressed, Memory-Resident System Matrices”. In: *IEEE Nuclear Science Symposium Conference Record* (2008). DOI: 10.1109/NSSMIC.2008.4774193.
- [43] NIPET contributors and F. Dellekart. *NIPET - fork with timing instrumentation*. <https://github.com/fdellekart/NIPET/tree/measure-timing>. Accessed: 2024-11-21.
- [44] *Get-Counter cmdlet documentation*. <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.diagnostics/get-counter?view=powershell-7.4>. Accessed: 2024-10-31.
- [45] A. C. Evans, A. L. Janke, D. L. Collins, and S. Baillet. “Brain templates and atlases”. In: *NeuroImage* (2012). DOI: 10.1016/j.neuroimage.2012.01.024.
- [46] E. T. Rolls, C.-C. Huang, C.-P. Lin, and J. Feng. “Automated anatomical labelling atlas 3”. In: *NeuroImage* (2020). DOI: 10.1016/j.neuroimage.2019.116189.
- [47] M. T. Gandia-Ferrero, I. Torres-Espallardo, B. Martinez-Sanchis, C. Morera-Ballester, E. Munoz, P. Sopena-Navales, G. Gonzalez-Pavon, and L. Marti-Nonmati. “Objective Image Quality Comparison Between Brain-Dedicated PET and PET/CT Scanners”. In: *Journal of Medical Systems* 47 (2023). DOI: 10.1007/s10916-023-01984-7.
- [48] Wellcome Centre for Human Neuroimaging. *Statistical Parametric Mapping - SPM 12*.
- [49] Siemens Medical Solutions USA Inc. *Biograph Vision Quadra System Specification*. https://cdn0.scrvt.com/39b415fb07de4d9656c7b516d8e2d907/3df06be092e5de00/6fdc8b779b1b/MI_Biograph_Vision_Quadra_spec_sheet_MI-5055_H00D05162003139323.pdf. Accessed: 2025-02-07.
- [50] R. L. Siddon. “Prism representation: a 3D ray-tracing algorithm for radiotherapy applications”. In: *Physics in Medicine and Biology* 30 (1985). DOI: 10.1088/0031-9155/30/8/005.

Appendix

Appendix A

To extract the number of prompt and delayed coincidences per frame from the list mode data a program was implemented in C++. It parses the coincidence events stored in a 32-bit PETLINK [28] format. The main program shown in Listing 1 relies on the PETLINKStream class, the declaration and implementation of which are shown in Listings 2 and 3. The PETLINKStream could also be useful for other applications which require reading the list mode data such as movement correction.

Listing 1: Source code of main program to count true and delayed coincidences in list mode data

```
#include <vector>

#include "PETLINKStream.h"

struct FrameResult {
    unsigned long n_prompts = 0;
    unsigned long n_delayeds = 0;
};

int main(int argc, char **argv) {
    auto filename = "/path/to/listmode/data.bf";
    auto outfile = "/path/to/output/events_per_frame.csv";
    PETLINKStream input_stream(filename);
    if (!input_stream.good()) {
        std::cerr << "File␣'" << filename << "␣not␣found.";
        exit(EXIT_FAILURE);
    }

    unsigned long long prompts = 0;
    unsigned long long delayeds = 0;
    int current_frame = 0;
    std::vector<FrameResult> results;
    results.resize(200);

    for (EventOrTag next : input_stream) {
        if (next.tag.is_timetag) {
            if (
                next.tag.elapsed_millis / 1000
                > (current_frame + 1) * 30)
            {
                std::cout << "Frame␣" << current_frame
                    << "␣finished␣with␣" << prompts
                    << "␣prompts␣and␣" << delayeds
                    << "␣delayeds." << std::endl;
            }
        }
    }
}
```

```

        results[current_frame].n_prompts = prompts;
        results[current_frame].n_delayeds = delayed;
        prompts = 0;
        delayed = 0;
        current_frame++;
    };
    continue;
}

if (next.event.is_prompt) {
    prompts++;
} else {
    delayed++;
}
}

std::ofstream result_file(outfile);
result_file << "frame, □prompts, □delayeds" << "\n";
current_frame = 0;

for (FrameResult result : results) {
    result_file << current_frame << "," << result.n_prompts << ","
        << result.n_delayeds << "\n";
    current_frame++;
}

result_file.close();
return 0;
}

```

Listing 2: Declaration of the PETLINKStream

```

#include <cstdint>
#include <filesystem>
#include <fstream>
#include <iostream>

/// @brief Coincidence Event
/// Can be either a prompt or delayed event
/// Position is indicated via the bin_address
/// Haven't yet found out how this relates to actual
/// positions with certainty, source code of niftyjet
/// and stir could tell
class Event {
public:
    Event() { Event(0); };

```

```

    Event(uint32_t word);
    uint32_t word;
    bool is_prompt;
    bool is_delayed;
    uint32_t bin_address;
};

/// @brief Tags store different metadata within the stream
/// For example timetags store the elapsed time
class Tag {
public:
    Tag() : Tag(0xffffffff) {};
    Tag(uint32_t word);
    uint32_t word;
    bool is_timetag;
    uint32_t elapsed_millis;
};

/// @brief Wrapper to return either event or tag
/// Can have either tag or event set but not both
struct EventOrTag {
    bool is_event;
    Event event;
    Tag tag;
};

/// @brief Input filestream to read listmode files
/// Supports only 32 bit PETLINK format
class PETLINKStream : public std::ifstream {
public:
    PETLINKStream() {};
    PETLINKStream(const char *listmode_file);
    PETLINKStream(std::filesystem::path listmode_file)
        : PETLINKStream(listmode_file.c_str()) {};
    ~PETLINKStream();

    struct iterator {
        PETLINKStream *stream;
        EventOrTag current_element;
    };

public:
    iterator(
        PETLINKStream *s = nullptr,
        bool is_end = false) : stream(s)
    {
        if (stream && !is_end && *stream) {

```

```

        current_element = stream->get_next();
        if (stream->eof())
            stream = nullptr;
    } else {
        stream = nullptr;
    }
}

EventOrTag operator*() const {
    return current_element;
};

iterator &operator++() {
    if (stream) {
        current_element = stream->get_next();
        if (stream->eof())
            stream = nullptr;
    }
    return *this;
};

iterator operator++(int) {
    iterator temp = *this;
    ++(*this);
    return temp;
};

bool operator==(const iterator &other) const {
    return stream == other.stream;
};

bool operator!=(iterator other) const {
    return !(*this == other);
};

iterator begin() { return iterator(this); }

iterator end() { return iterator(this, true); }

/// @brief Get the next element in the stream
/// @return Tag or event, depending on the next 32 bits type
std::shared_ptr<EventOrTag> get_next();

protected:
    const char *listmode_file;

```

```
private:
    uint32_t current_word;
};
```

Listing 3: Implementation of the PETLINKStream

```
#include "PETLINKStream.h"
#include <cassert>
#include <fstream>
#include <iostream>
#include <memory>

PETLINKStream::PETLINKStream(const char *listmode_file)
    : listmode_file(listmode_file) {
    open(listmode_file, std::ifstream::in | std::ifstream::binary);
};

PETLINKStream::~PETLINKStream() { close(); };

EventOrTag PETLINKStream::get_next() {
    read(
        reinterpret_cast<char *>(&current_word),
        sizeof(current_word)
    );
    EventOrTag result;
    if (current_word >> 31) {
        result.is_event = false;
        result.tag = Tag(current_word);
    } else {
        result.is_event = true;
        result.event = Event(current_word);
    }
    return result;
};

Event::Event(uint32_t word) : word(word) {
    is_prompt = static_cast<bool>(word >> 30);
    is_delayed = !is_prompt;
    bin_address = word & 0x3fffffff;
};

Tag::Tag(uint32_t word) : word(word) {
    is_timetag = (word >> 29) == 0x4;
    elapsed_millis = word & 0x01fffffff;
};
```