

# Development of a Parallel Computing Optimized Head Movement Correction Method in Positron Emission Tomography



Entwicklung einer parallelisierten Methode zur  
Bewegungskorrektur bei Hirnuntersuchungen in der  
Positronen-Emissions-Tomographie

---

Jens Langner

1. Einführung Positronen-Emissions-Tomographie
2. Bewegungskorrektur und ihre Implementation
3. Motivation und Ziele
4. Vorabanalysen und Implementation
5. Ergebnisse & Vergleich
6. Zusammenfassung und Ausblick

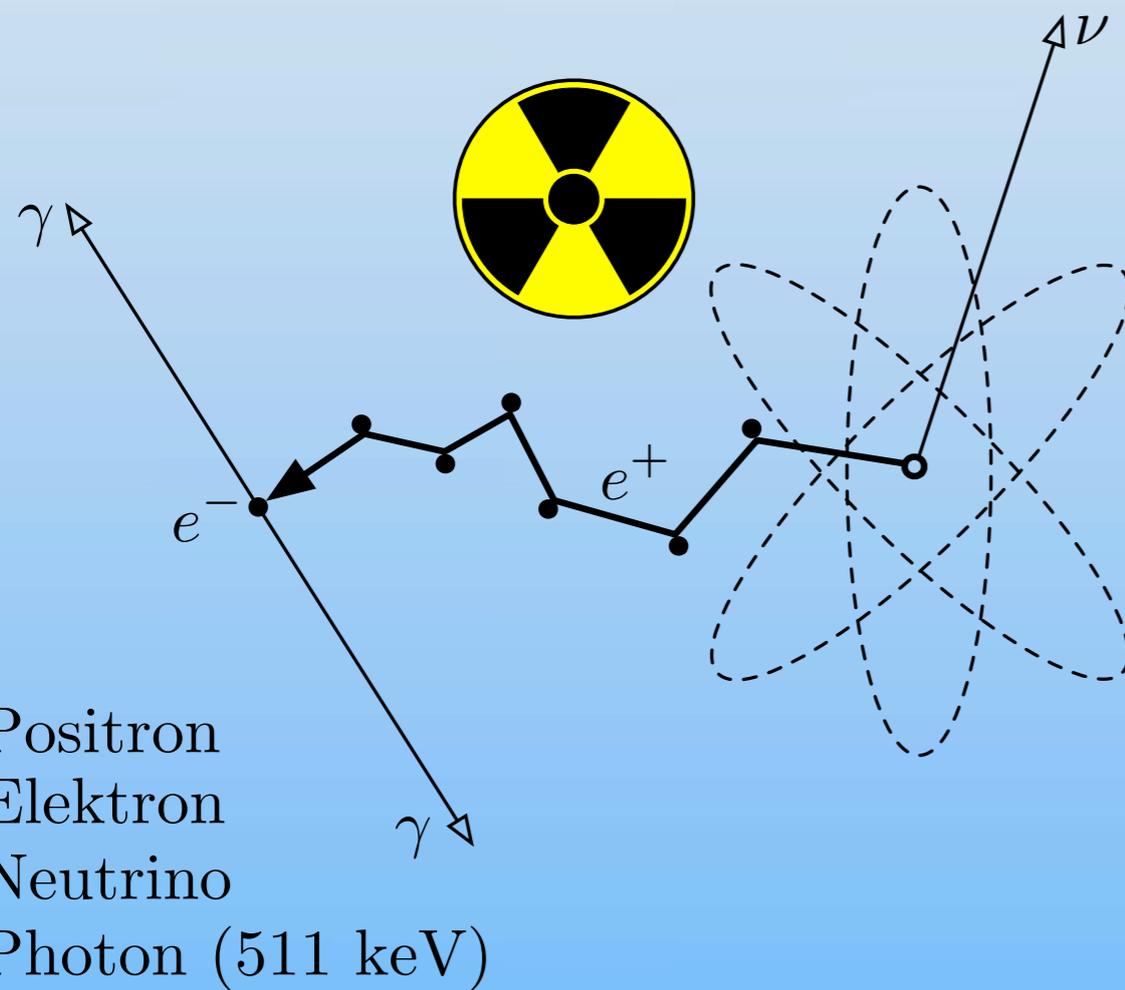


ECAT HR+ Scanner  
am PET Zentrum Rossendorf

## Positronen-Emissions-Tomographie (PET)

- nicht-invasives nuklearmedizinisches Bildgebungsverfahren
- Injizierung einer radioaktiv markierten Substanz (sog. Tracer)
- Anreicherung in stoffwechselreichen Regionen
- Registrierung der Zerfälle – Weiterleitung an eine Elektronik
- Berechnung der Aktivitätsverteilung durch Rekonstruktionsverfahren
- Nutzung zur medizinischen Diagnostik und Therapie (z.B. Tumordiagnostik, etc.)

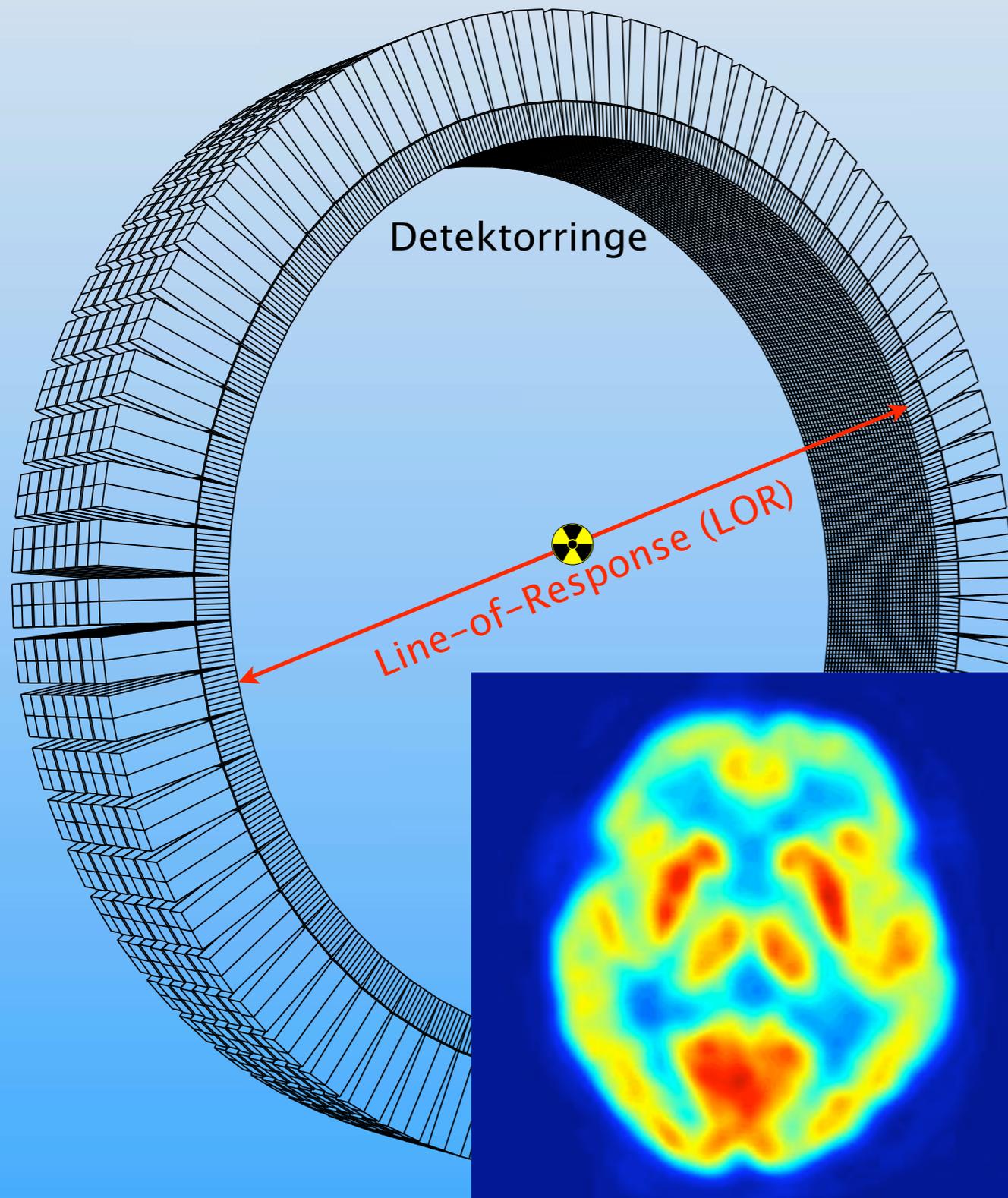
## PET Grundlagen



### Annihilationsprozess ( $\beta^+$ -Zerfall):

- Emittieren eines Positron und Neutrino durch Protonenzerfall
- Interaktion mit Elektronen der umgebenden Materie
- Vereinigung mit einem einzelnen Elektron (Annihilation)
- Umwandlung der Massen in radioaktive Strahlung (Gamma-Quanten)
- Quanten entfernen sich in entgegengesetzter Richtung

## PET Grundlagen

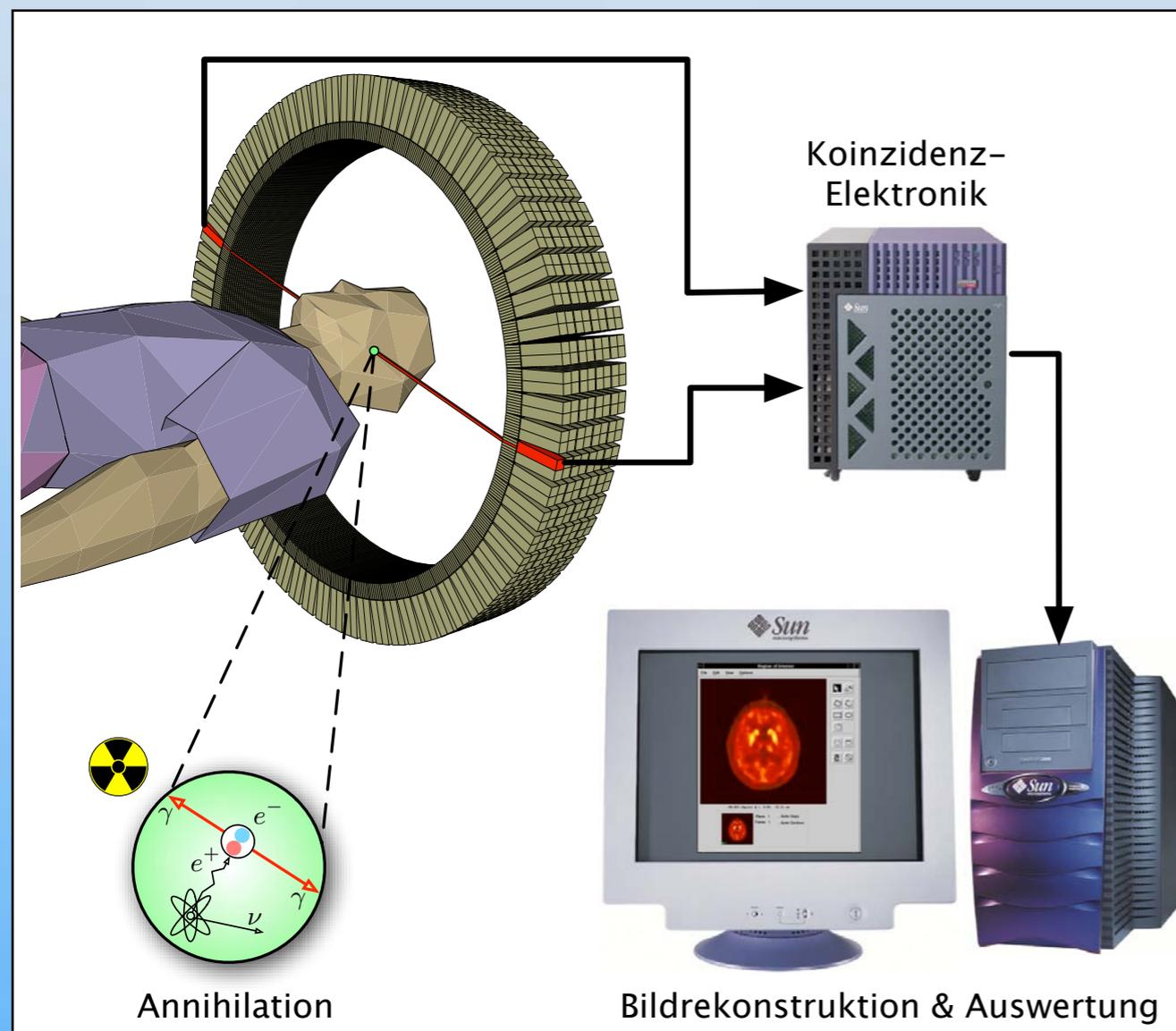


- ringförmige Anordnung gamma-sensitiver Detektoren
- Koinzidenzelektronik filtert die Ereignisse
- Gleichzeitigkeit entscheidet über Gültigkeit (Koinzidenz)
- Registrierung in diskreten Werten als sog. Line-of-Response (LOR)
- Einsortieren aller LORs in ein Histogramm (in PET: Sinogramm)
- Erzeugung von Volumendatensätzen durch computergestützte Rekonstruktionsverfahren

Qualitative Einflüsse & Genauigkeit:

- Limitierung der Auflösung durch physikalische Gegebenheiten auf 4–5mm
- externer Einfluss durch Organbewegungen (Herz/Lunge etc.)
- starker Einfluss durch ungewollte Patientenbewegungen
- besonders bei Regionen mit kleinen Strukturen (Hirnuntersuchungen) nicht vernachlässigbar

➔ Motivation: Bewegungskorrektur bei Hirnuntersuchungen mit Hilfe eines externen Trackingsystems

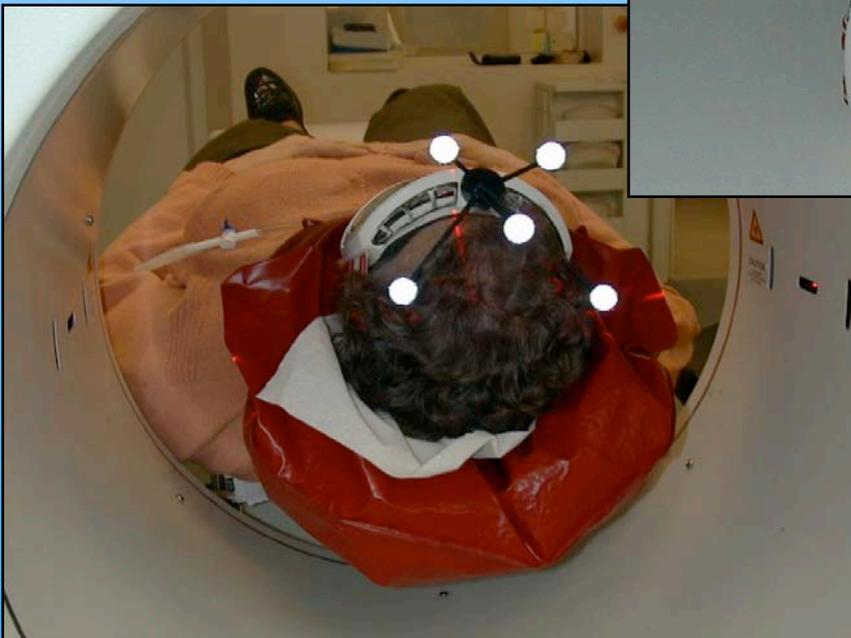


**PET Verarbeitungsschema**  
– von der Annihilation bis zur Auswertung –

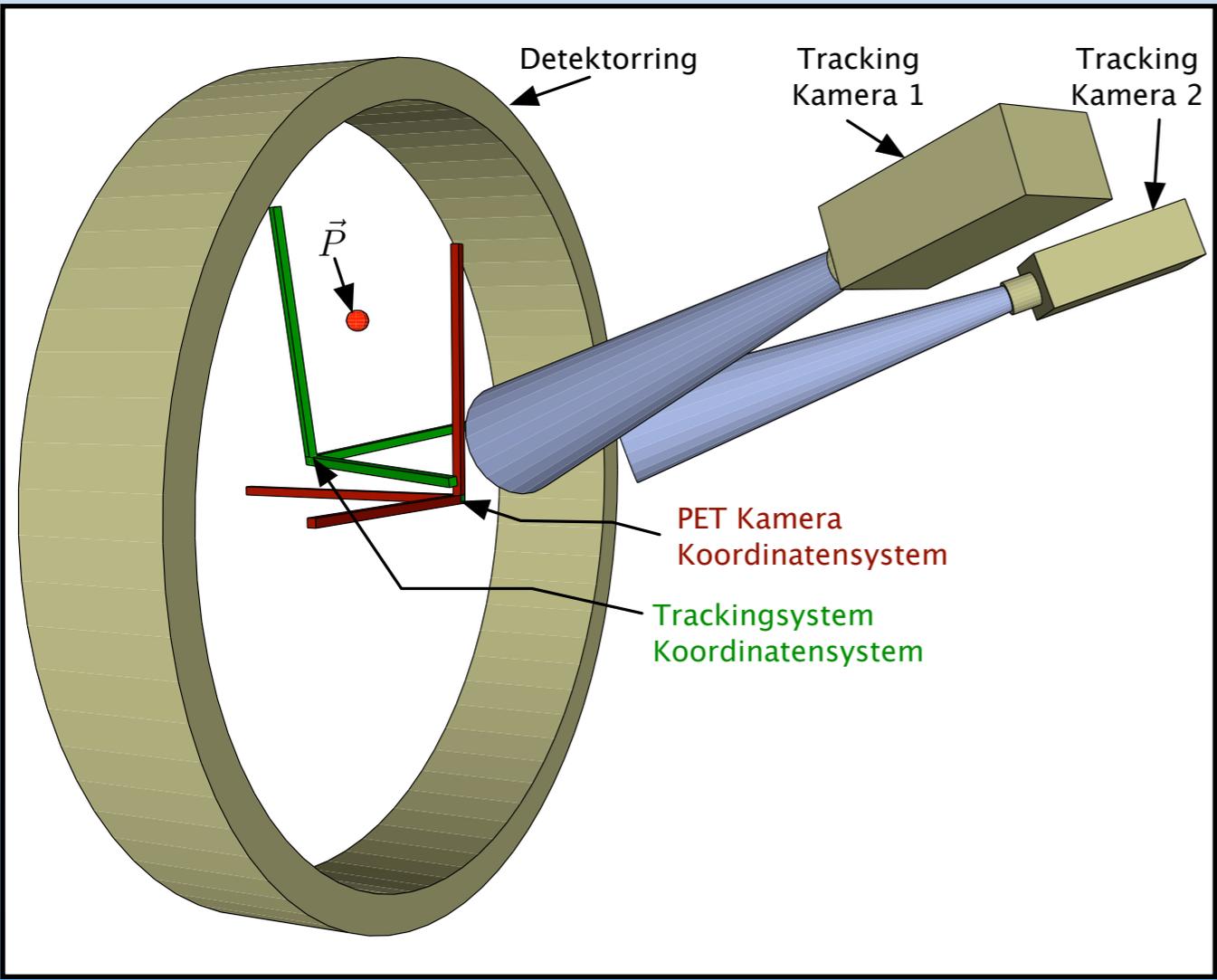
## Bewegungskorrektur

### Kameragestütztes Tracking:

- passives Bewegungstrackingsystem mit zwei Kameras
- integrierte Infrarotblitze leuchten die PET Kamera von hinten aus
- passive Markeranordnungen (Bodies) reflektieren das Licht an die mit einem CCD ausgerüsteten Kameras
- Bestimmung der 3D-Position durch stereoskopische Anordnung möglich
- Räumliche Auflösung  $< 1\text{mm}$ , maximale Abtastrate liegt bei 60Hz



# Bewegungskorrektur

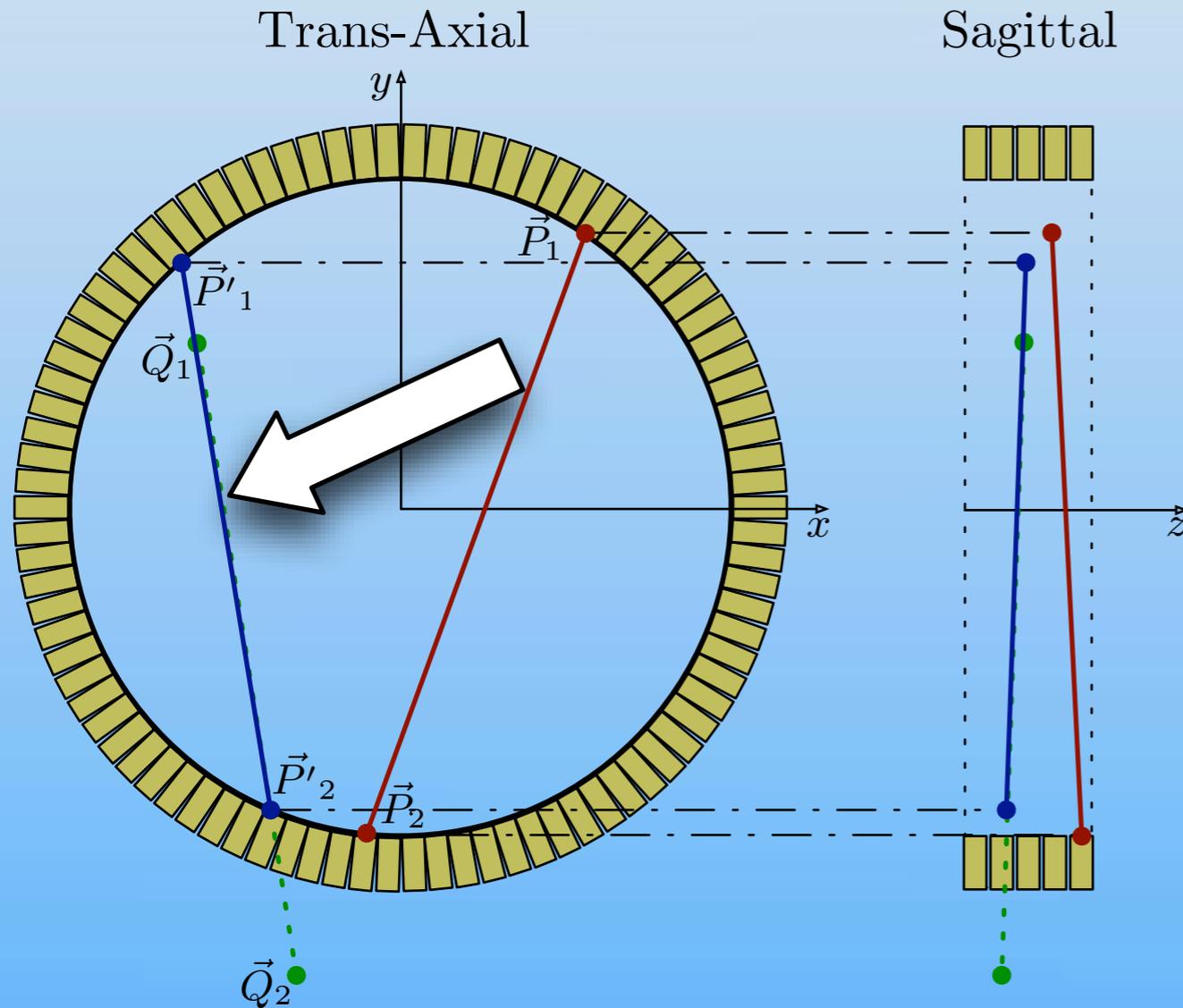


## Cross-Calibration:

- das Trackingsystem und der Tomograph besitzen unterschiedliche Koordinatensysteme
- Notwendigkeit der Transformation der Bewegungsinformationen in das PET System
- daher Cross-Calibration zwischen beiden Systemen
- nur in regelmäßigen Intervallen oder bei Repositionierung ist eine Neuberechnung notwendig

$$\begin{pmatrix} P'_x \\ P'_y \\ P'_z \\ 1 \end{pmatrix} = \begin{pmatrix} T_{cc1,1} & T_{cc1,2} & T_{cc1,3} & t_{ccx} \\ T_{cc2,1} & T_{cc2,2} & T_{cc2,3} & t_{ccy} \\ T_{cc3,1} & T_{cc3,2} & T_{cc3,3} & t_{ccz} \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

LOR basierte Bewegungskorrektur:



1. Umrechnung in kartesische Koordinaten:  
 $RE, AN, RI_{A,B} \rightarrow \vec{P}_{1,2}$
2. Räumliche Transformation:  
 $\vec{P}_{1,2} \rightarrow \vec{Q}_{1,2}$
3. Schnittpunktberechnung:  
 $\vec{Q}_{1,2} \rightarrow \vec{P}'_{1,2}$
4. Rückkonvertierung in kodierte Werte:  
 $\vec{P}'_{1,2} \rightarrow RE', AN', RI'_{A,B}$

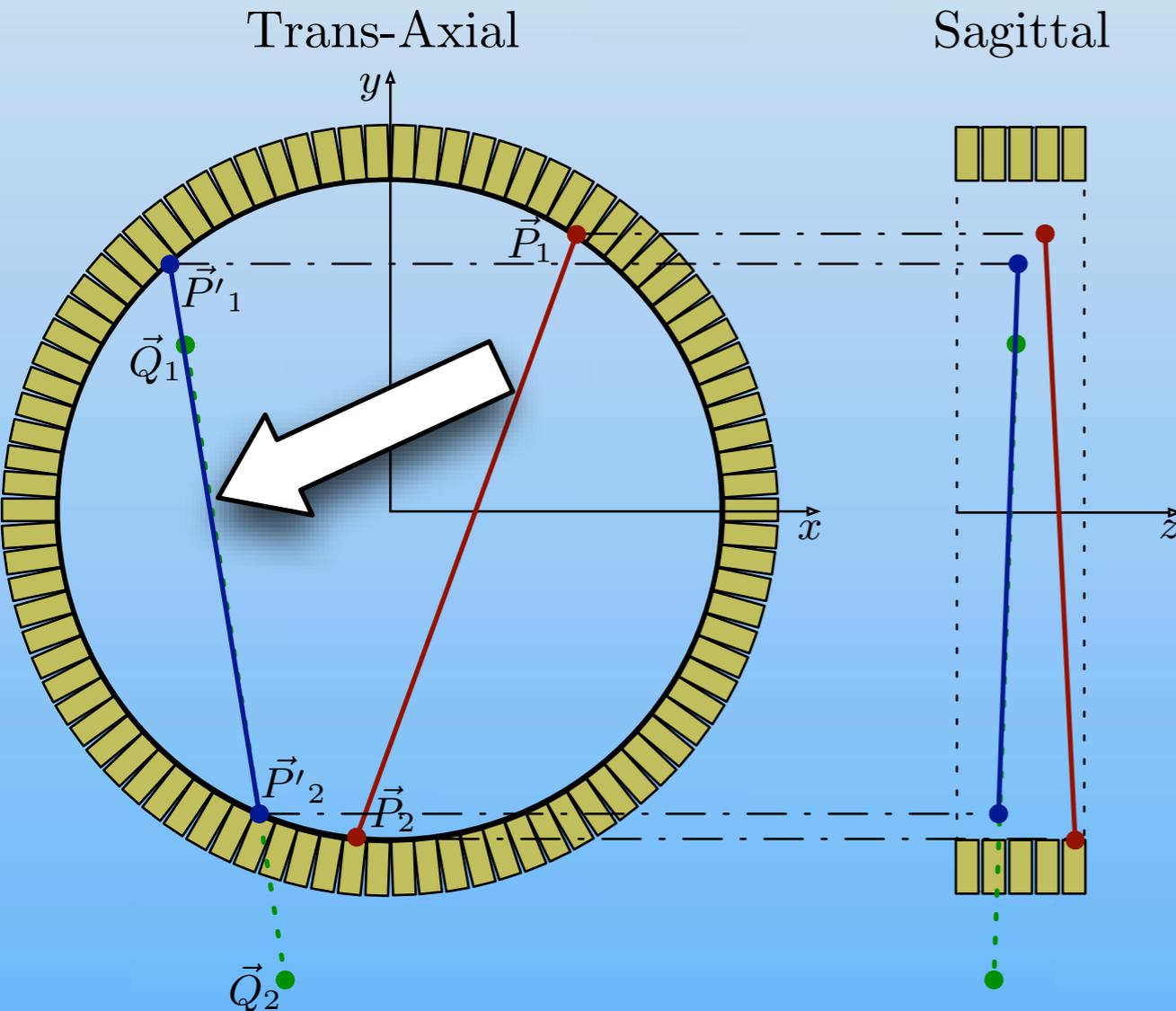
$RE, AN, RI_{A,B}$   
PET Scanner kodierte  
LOR Koordinaten

(1) P.Buehler et al. An accurate method for correction of movement in PET.  
PET research center Rossendorf, 2003.

Auftretende Probleme & Korrekturen:

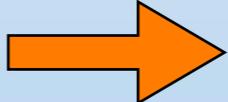
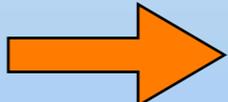
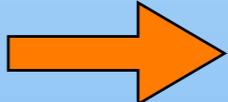
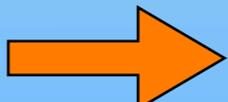
- Unterschiedliche Detektorsensivitäten  
führen zu Ringartefakten  
> **Normalisierungs-Korrektur**
- Durch Diskretisierung kommt es zu  
leeren und überfüllten Bereichen  
> **LOR Diskretisierung-Korrektur (LDC)**
- LORs können aus dem Field-of-View  
(FOV) transformiert werden  
> **Out-of-FOV Korrektur (OFC)**

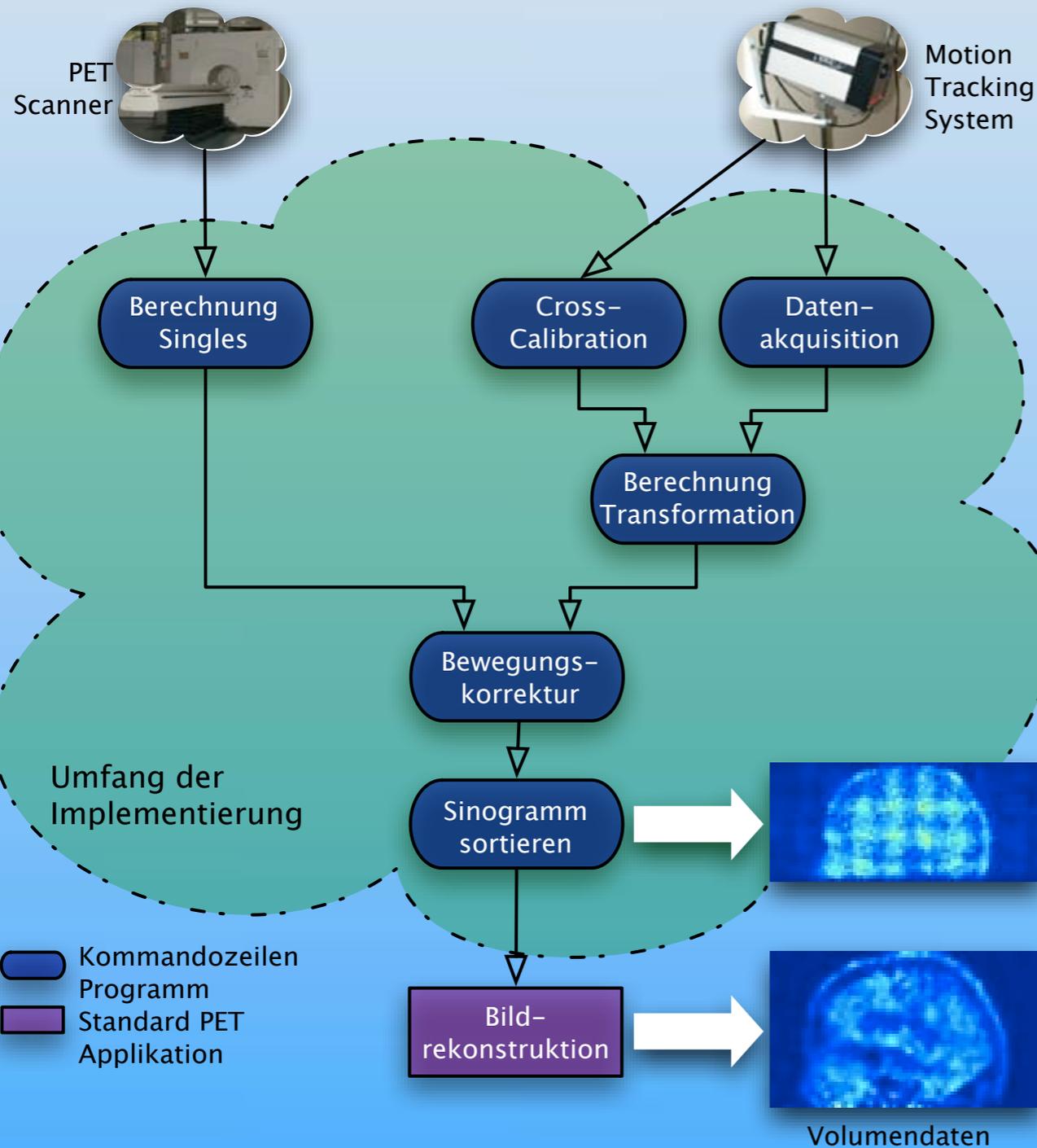
**➔ Hoher Berechnungsaufwand !**  
 ~ 5GB/Stunde Datenvolumen  
 ~ 20 Mio. LORs pro OFC  
 > pro Bewegungsinformation eine OFC



(1) P.Buehler et al. An accurate method for correction of movement in PET.  
PET research center Rossendorf, 2003.

## Bisherige Implementierungen & Motivation der Arbeit:

-  unoptimierte/sequentielle Berechnung  
schließt eine Routinenutzung aus  
> ~20 Stunden für eine 1 stündige PET Aufnahme  
 Optimierung der Berechnungen  
durch Parallelisierung der  
Algorithmen
-  keine Multiframefähigkeit  
> nicht für dynamische Studien einsetzbar  
 Implementierung der  
Multiframefähigkeit
-  Beschränkung der Nutzung auf  
experimentelle Zwecke  
> nur Kommandozeilen Nutzung, hohe Komplexität  
in der Bedienung  
 Programmierung einer  
graphischen Oberfläche  
mittels Qt
-  separate Handhabung der Cross-  
Calibration  
> zusätzliche Verwaltung  
 Direkte Integration
-  Nicht modulare Implementierung und  
Nutzung uneinheitlicher  
Programmierungsumgebungen  
> C, IDL, Shell-Skripte  
 Plattform-unabhängige und  
objekt-orientierte  
Programmierung in C++



## I. Work-Flow Analyse:

- Identifizierung bestehender Komponenten und Programmteile
- Datenflussanalyse und Bestimmung der zukünftigen Systemkomponenten
- Aufzeigen der Systemgrenzen  
> **interne/externe Schnittstellen**
- Bestimmen der Datenformate  
> **ECAT / Listmode Format**
- Aufteilen der Komponenten in einzelne Programmmodule

*Amdahl's Law (1967):*

$$S(N) = \frac{1}{f + \frac{1-f}{N}}$$

mit  $f > 0$

$$\lim_{N \rightarrow \infty} S(N) = 1/f$$

$S(N)$  - Zu erwartende Geschwindigkeitssteigerung

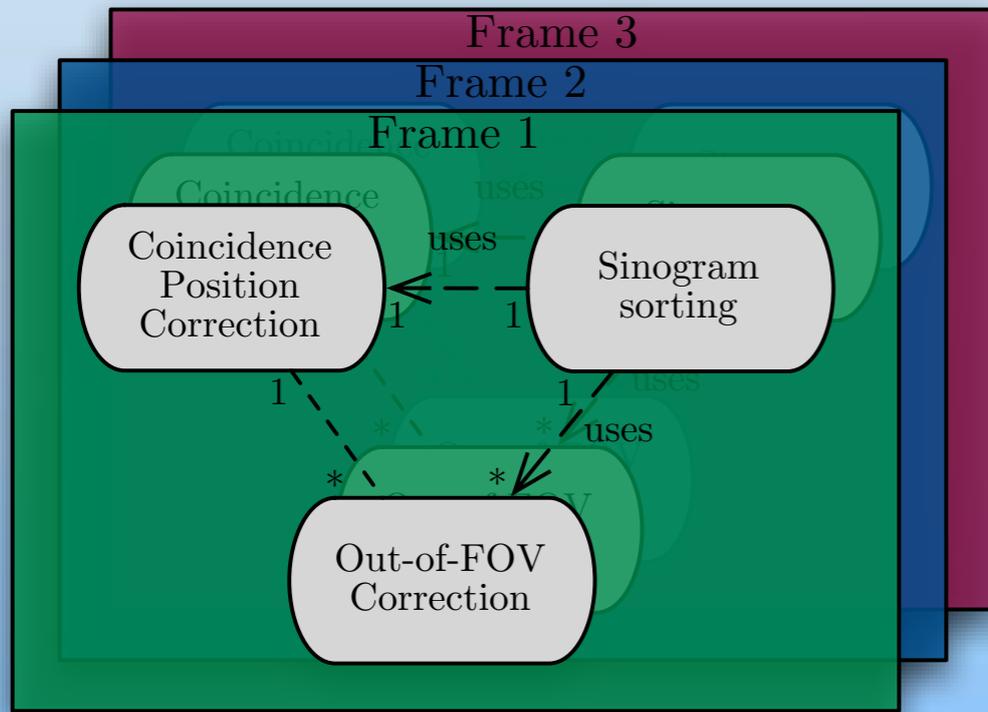
$f$  - Sequentieller Anteil

$N$  - Anzahl Prozessoren

## II. Parallelisierung:

- Der maximal zu erreichende SpeedUp skaliert laut Amdahl's Law nicht mit der Anzahl von Prozessoren
- Geschwindigkeitssteigerung ist vom sequentiellen Anteil abhängig
- Abhängigkeitsanalysen müssen parallelisierbare Bereiche aufzeigen
- Schrittweise Parallelisierung erhöht den Durchsatz

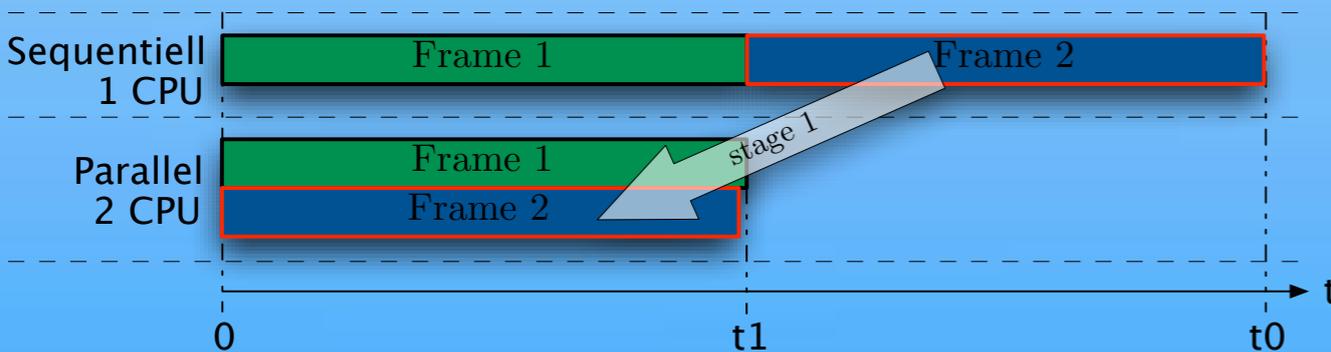
(1) G.M.Amdahl. Validity of single-processor approach to achieving large-scale computing capability. Proceedings of AFIPS Conference, Reston 1967.



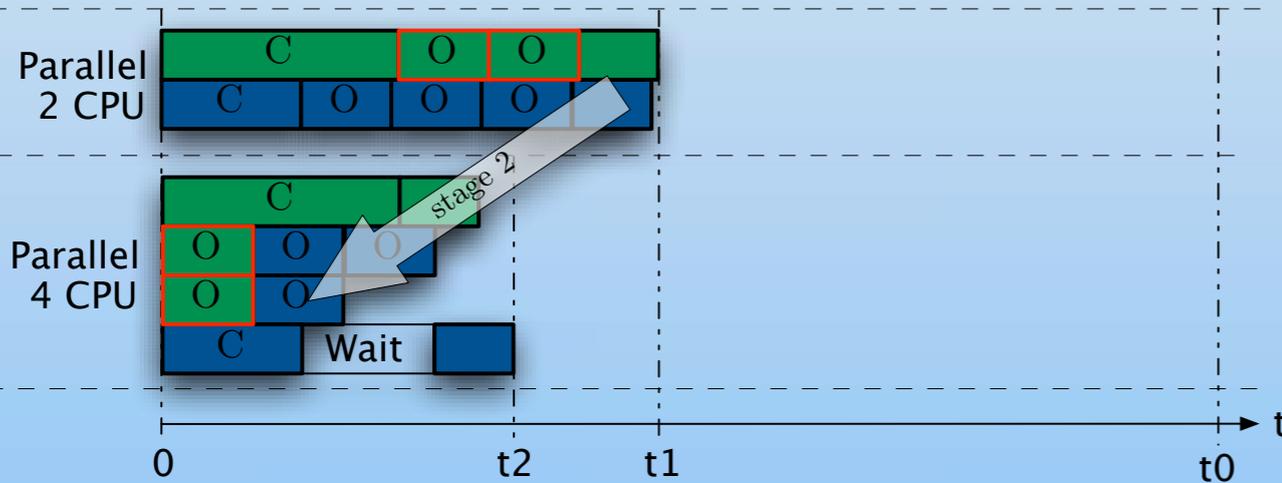
### III. Abhängigkeitsanalysen:

- Typische Untersuchungen haben um die 20 Frames
- Jeder Frame stellt eine unabhängige Einheit dar
- Out-of-FOV Korrektur ist stark parallelisierbar
- Schrittweise Optimierung, **Schritt1 -> Schritt2 -> Schritt3**
- Schritt 1:  
Parallelisierung der Frames ( $t_0 \rightarrow t_1$ )

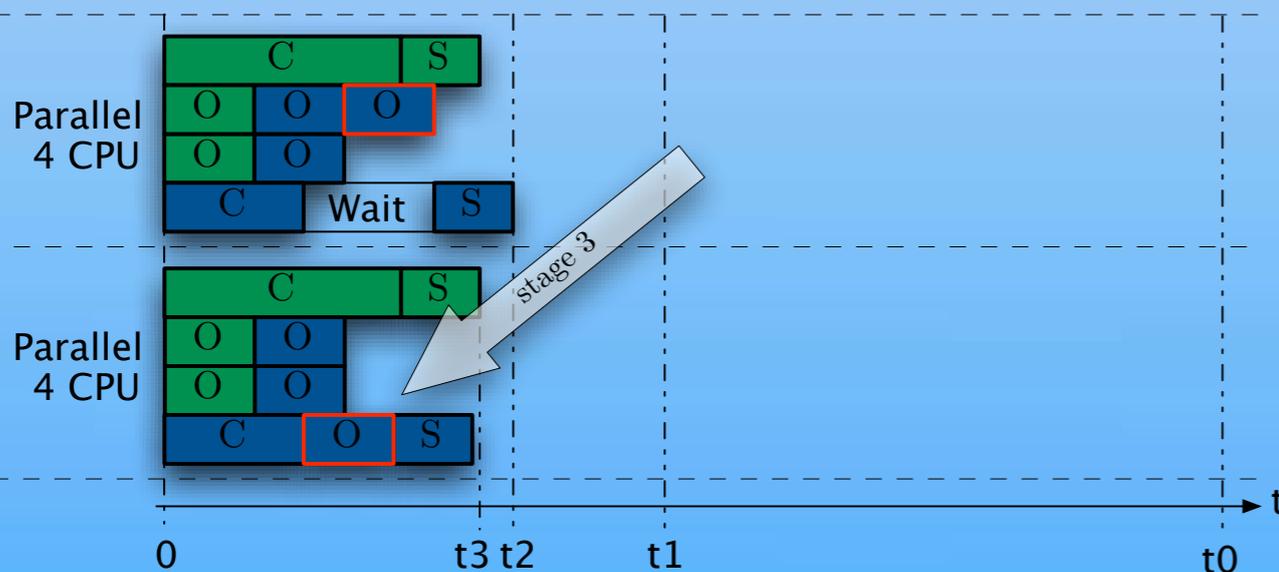
#### Schritt 1:



Schritt 2:

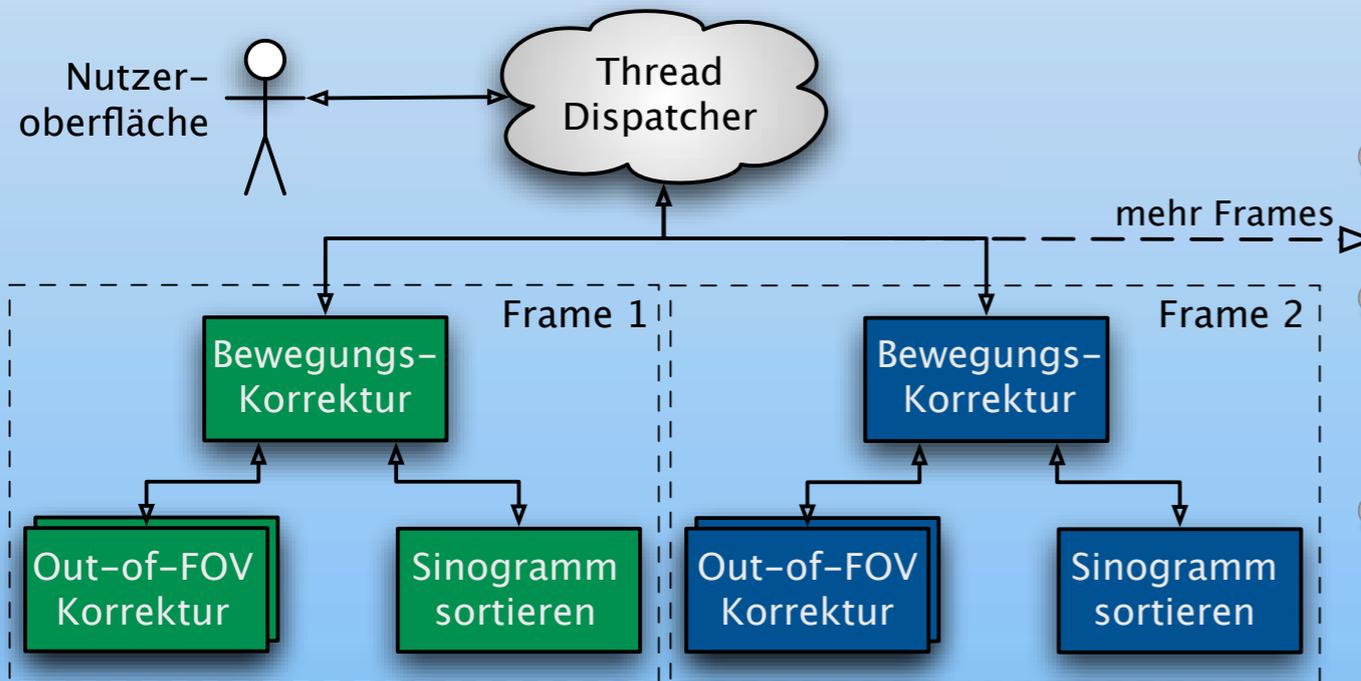


Schritt 3:



### III. Abhängigkeitsanalysen:

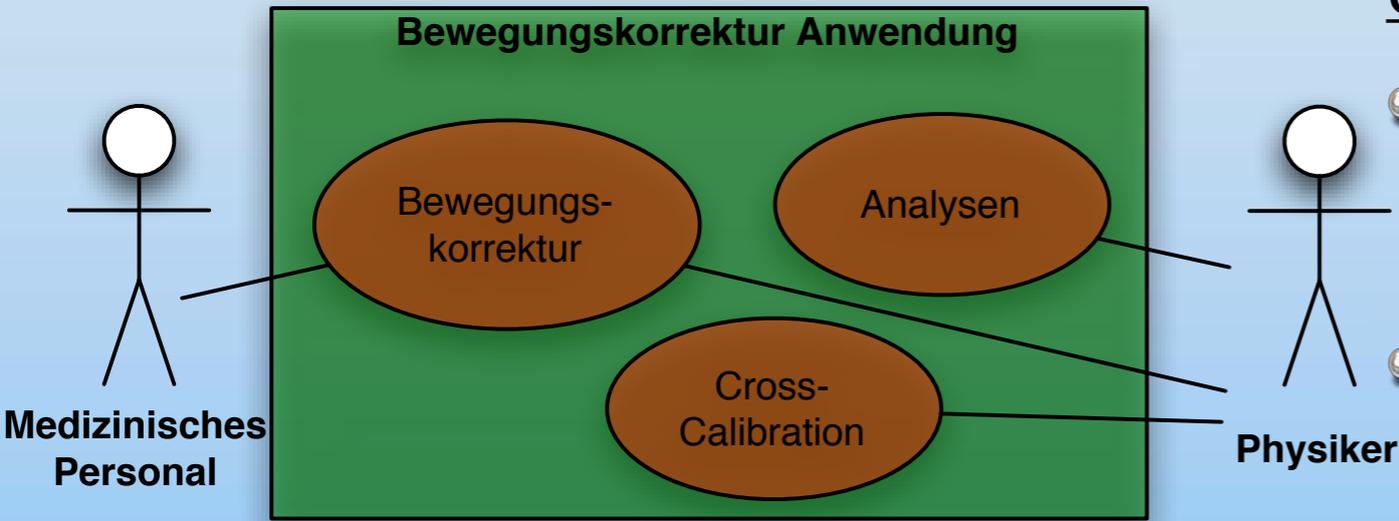
- Schritt 2:  
Parallelisierung der Out-of-FOV Korrekturen ( $t_1 \rightarrow t_2$ )
- Abhängigkeit zwischen Korrektur und Sortierung führt zu einer sog. "WaitCondition"
- Schritt 3:  
Auflösen der WaitCondition ( $t_2 \rightarrow t_3$ )
- Sukzessive Verminderung bei mehreren Frames
- Resultierende Verminderung von  $t_0$  zu  $t_3$ .



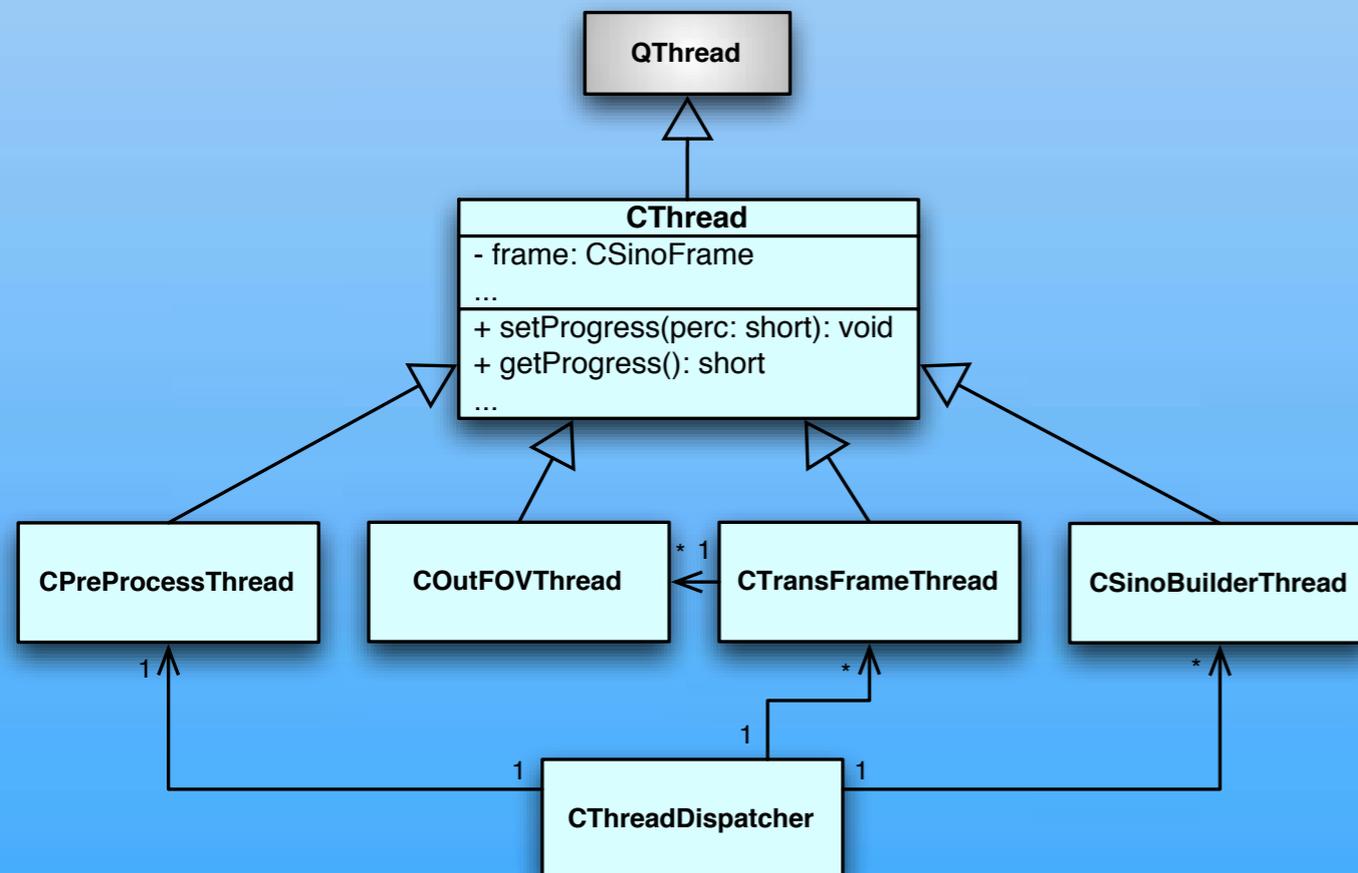
### Multithreading Implementierung:

- Implementation durch Qt-eigene Threads (POSIX orientiert)
- Einführung einer Thread-Hierarchie
- Nutzung von Semaphoren und Mutual-Exclusives (Mutex)
- Keine direkte Kommunikation mit der Nutzeroberfläche möglich
- "Thread Dispatcher" fungiert als Kommunikationszentrale
- Kommunikation zum Dispatcher erfolgt über Eventhandler
- Weiterreichen der Events per direktem Methodenaufruf

Anwendungsfalldiagramm:



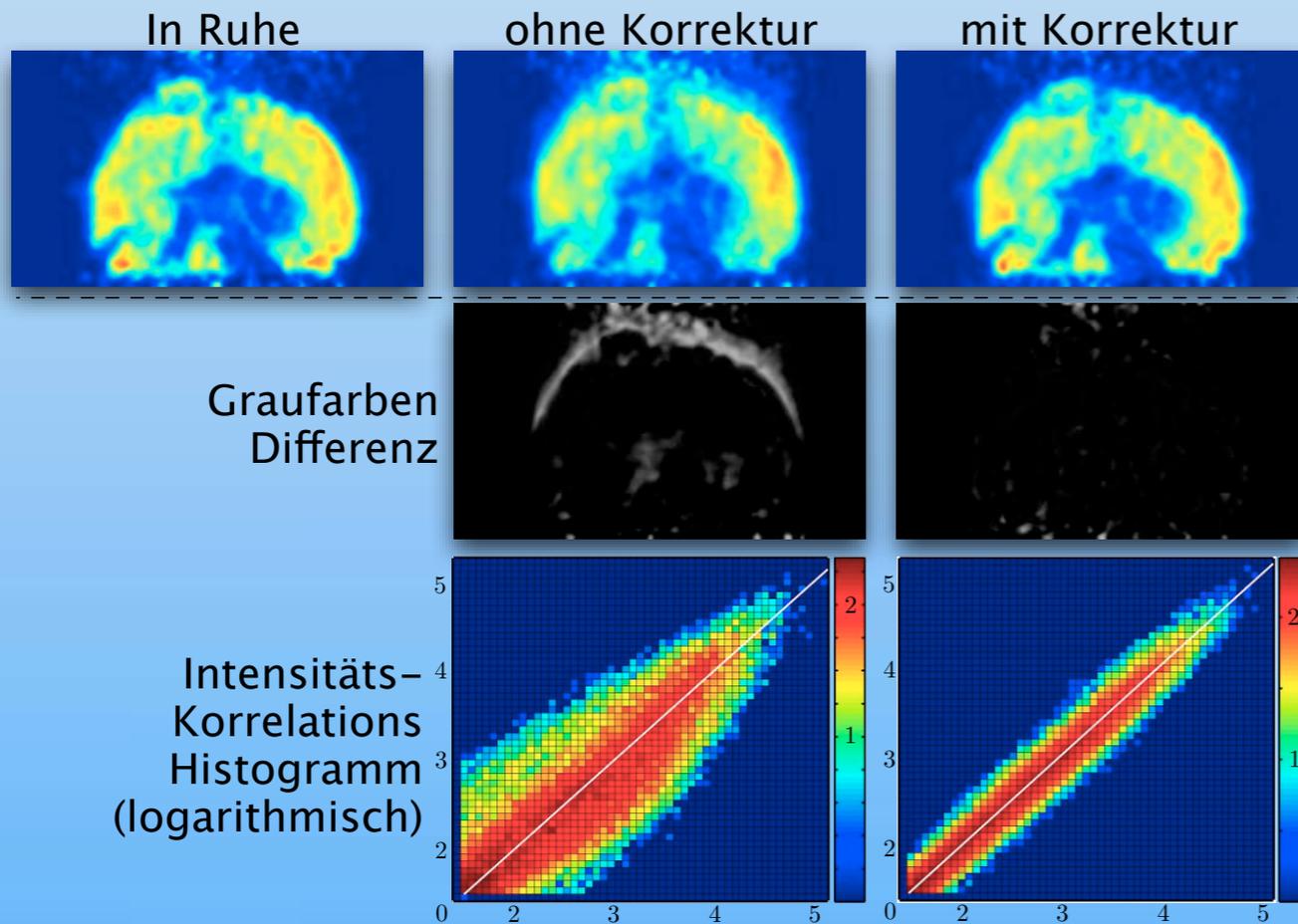
Klassendiagramm:



Umsetzung mit UML:

- Spezifikation des Systems mit Hilfe der Unified Modeling Language (UML)
- Einteilung der Anwendungsfälle in **Bewegungskorrektur, Cross-Calibration und Analysen**
- Nutzung von Klassendiagrammen z.B. Überführung der Thread-Hierarchie
- Darstellen der Abhängigkeiten sowie Ableitungshierarchien

Beispiel einer Phantommessung  
(Bewegung in Z Richtung)

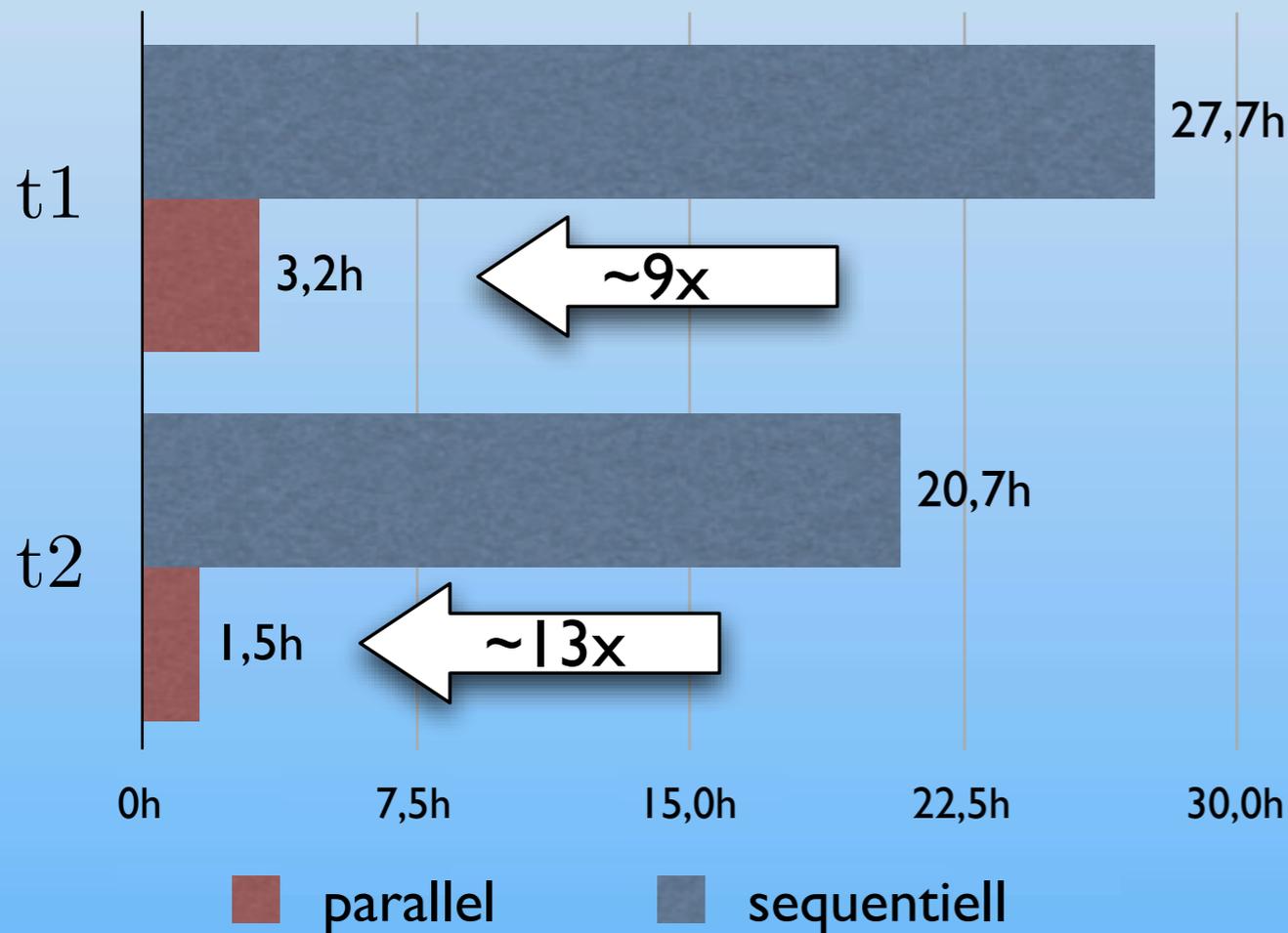


Sagittale Ansichten der rekonstruierten Daten einer Phantommessung, bei der das Phantom schrittweise in axialer Richtung (z) bewegt wurde.

Qualitative und quantitative Bewertung:

- Durchführung von drei Phantommessungen a 10 Minuten
- schrittweise Bewegung in den letzten 5 Minuten
- sichtbare qualitative Verbesserung durch Bewegungskorrektur
- quantitative Verbesserung durch Auswertung von 2D Korrelationshistogrammen registrierbar
- Bestätigung dieser Ergebnisse nach Durchführung einer "In-vivo" Studie

Geschwindigkeitsvergleich  
alte (sequentielle) und neue (parallele) Berechnung:



t1: Vergleich Single-Frame Studie – 1h PET Akquisition  
 t2: Vergleich Multi-Frame Studie – 1h PET Akquisition  
 (da beim Multiframe Vergleich für die sequentielle Implementierung kein multiframe support existiert, wurde hierfür eine Singleframe-Studie mit gleichen Optionen/Komplexität verwendet)

Performancevergleiche:

- Vergleich der Geschwindigkeit in 16 unterschiedlichen Tests
- 9-fache Beschleunigung bei Durchführung in der höchsten Komplexität (Single-Frame)
- Praxisnähere Korrektur einer Multi-Frame Studie ergab 13-fache Beschleunigung
- dies wurde durch frame-basierte Optimierung erreicht

**1:1,5 Verhältnis**  
anstatt 1:20

4x900MHz UltraSPARC-III mit  
16GB RAM unter Solaris 2.9

The screenshot displays the LMMC (ListMode Movement Correction) software interface, version 0.13 (11.11.2003). The window is titled "LMMC - ListMode Movement Correction v0.13 (11.11.2003)" and features a menu bar with "File", "Settings", "Window", and "Help".

The main interface is divided into several sections:

- Movement Correction Tab:**
  - Tracking data (Correction):** Method: Use precalculated data. Precalculated data section includes: Tracking data source: Tracking data file (\*.trk); Cross calibration set: No Room Calibration found; Tracking data file (\*.trk): [empty]; Listmode acquisition start: 12:10:3; Tracked body number: 1. Transformation data status: not available. An "Export data" button is present.
  - Scanner data (Input):** Listmode files section with a table for adding files (Filename, Add, Remove). Frame specification section with a table for adding frames (#, Start, End, Duration, Add, Remove, Import, Details). Frame: 0 - 1 s.
  - Normalization data files:** Singles rate file (\*.sgl) [empty]; Use Normalization file [checked] [empty].
  - Output format settings:** Output file format: 3D ECAT7-Auto Sinogram (\*.S); Filename: [empty]; Patient data: n/a. A "Modify" button is present.
  - Movement Correction processing:** Out-of-FOV correction [checked] (4, 4, 1); LOR discretization correction [checked] (Enhanced 3D). A "Start" button is present.

## Zusammenfassung & Ausblick:

- Durch schrittweise Parallelisierung konnte die Bewegungskorrektur effektiv beschleunigt werden und kann somit in Zukunft bei Routineuntersuchungen eingesetzt werden
- Eine graphische Nutzeroberfläche erlaubt die übersichtliche Nutzung und Integration in den normalen Arbeitsablauf
- Durch Implementation der Multiframefähigkeit kann die Bewegungskorrektur bei dynamischen Studien angewandt werden
- Die direkte Integration der Cross-Calibration erleichtert und beschleunigt die Administration der Bewegungskorrektur
- Portabilität und Erweiterbarkeit sind durch die Programmiersprache C++ und das Benutzeroberflächen Framework Qt gewährleistet
- Durch Kapselung einzelner Komponenten können diese in zukünftigen Projekten am FZR direkt verwendet werden
- Import- und Exportmodule nutzen gängige Standards wie XML und ECAT und tragen so zur Integration in die bestehenden Systeme bei