TECHNISCHE UNIVERSITÄT DRESDEN

Institut für Technische Informatik, Professur für Rechnerarchitektur – TU Dresden

Parallelisierung des Wellenfrontrekonstruktionsalgorithmus auf Multicore-Prozessoren

Jonas Schenke

12. April 2017

Betreuender HSL: Prof. Dr. Wolfgang E. Nagel
Betreuer: Dr. Elena-Ruxandra Cojocaru, Dr. Michael Bussmann, Matthias Werner
E-Mail: jonas.schenke@tu-dresden.de

ZIH
Center for Information Services & High Performance Computing

Aufgabenstellung

- Evaluierung und Performance-Analyse des derzeit fast durchgängig seriellen Wellenfrontrekonstruktionsalgorithmus
- Parallelisierung der kritischen Pfade für Mehrkernarchitekturen
- Performance-Messungen der parallelen Implementation
- Auswertung sowie Validierung der Ergebnisse

Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

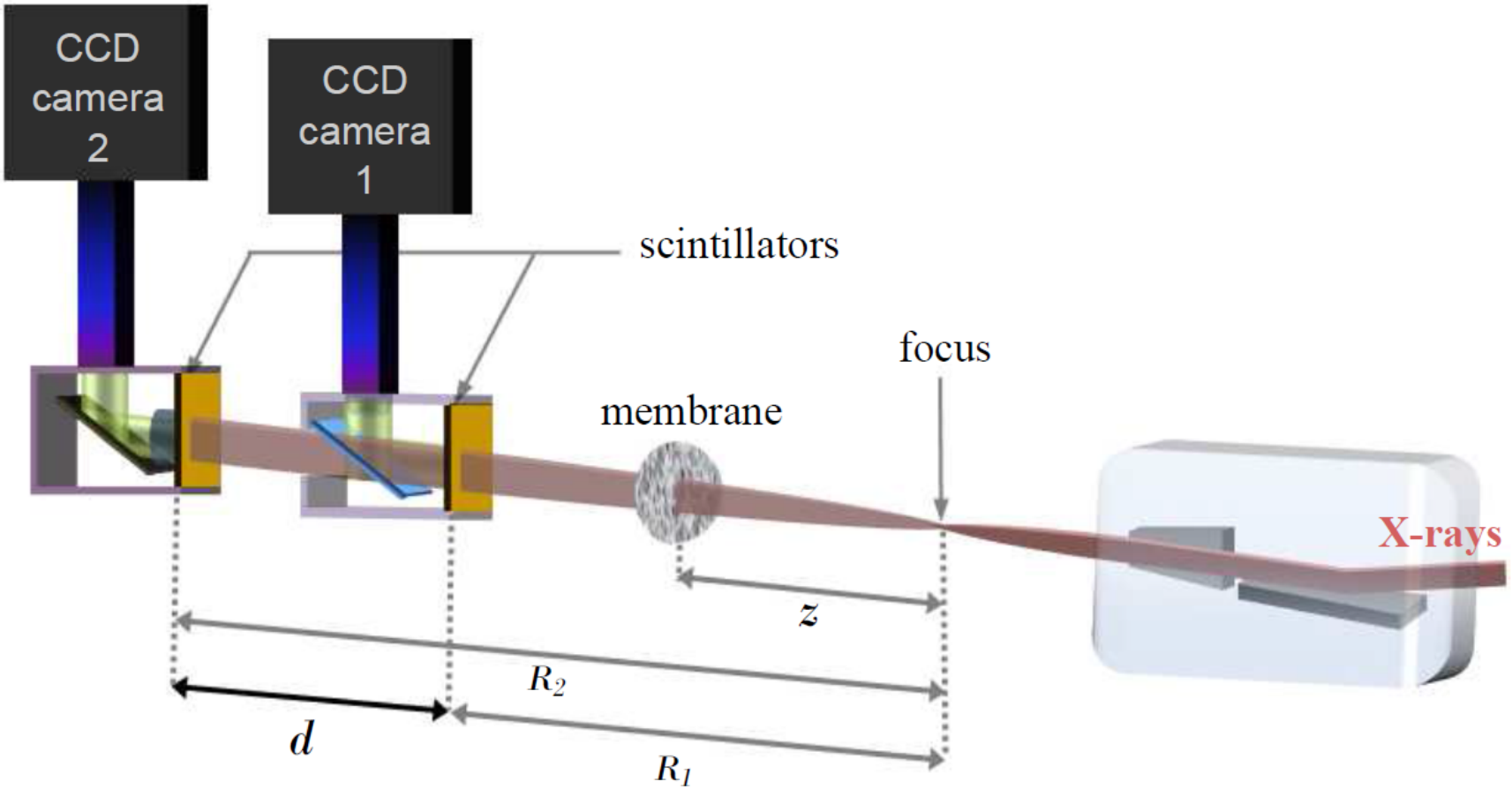
- Teil des **Eu**ropean **C**luster of **A**dvanced **L**aser **L**ight Sources (EUCALL)-Projektes →Überschneidung von **U**ltra**f**ast **D**ata **Ac**quisition (UFDAC; WP5) und **Pu**lse **C**haracterisation and **C**ontrol (PUCCA; WP7)
- Zusammenarbeit des **H**elmholtz-**Z**entrum **D**resden-**R**ossendorf e.V. (HZDR) und **E**uropean **S**ynchrotron **R**adiation **F**acillity (ESRF)
- **Algorithmus:** Sébastien Bérujon
- **Code:** Elena-Ruxandra Cojocaru
- **Daten:** Beamline BM05, ESRF, Grenoble, Frankreich

	Experiment 6	**Lenses**
Aufnahmedatum	Ruxandra Cojocaru, Sébastien Bérujon, Eric Ziegler	Thomas Rothand, Raymond Barett, Sébastien Bérujon, Rafael Celeste
Aufgenommen von	24. September 2017	10. April 2017

Gliederung

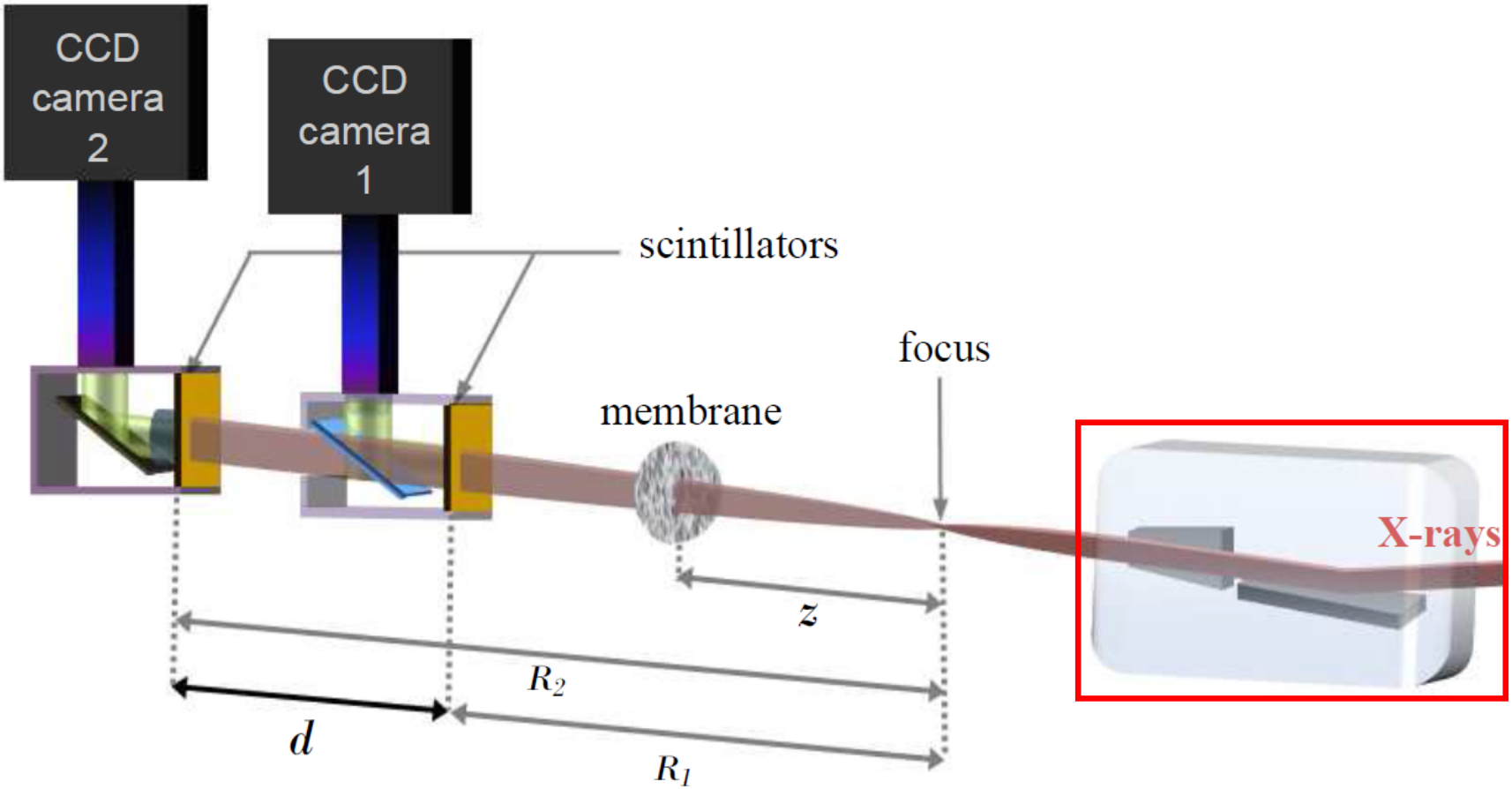
1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Versuchsaufbau



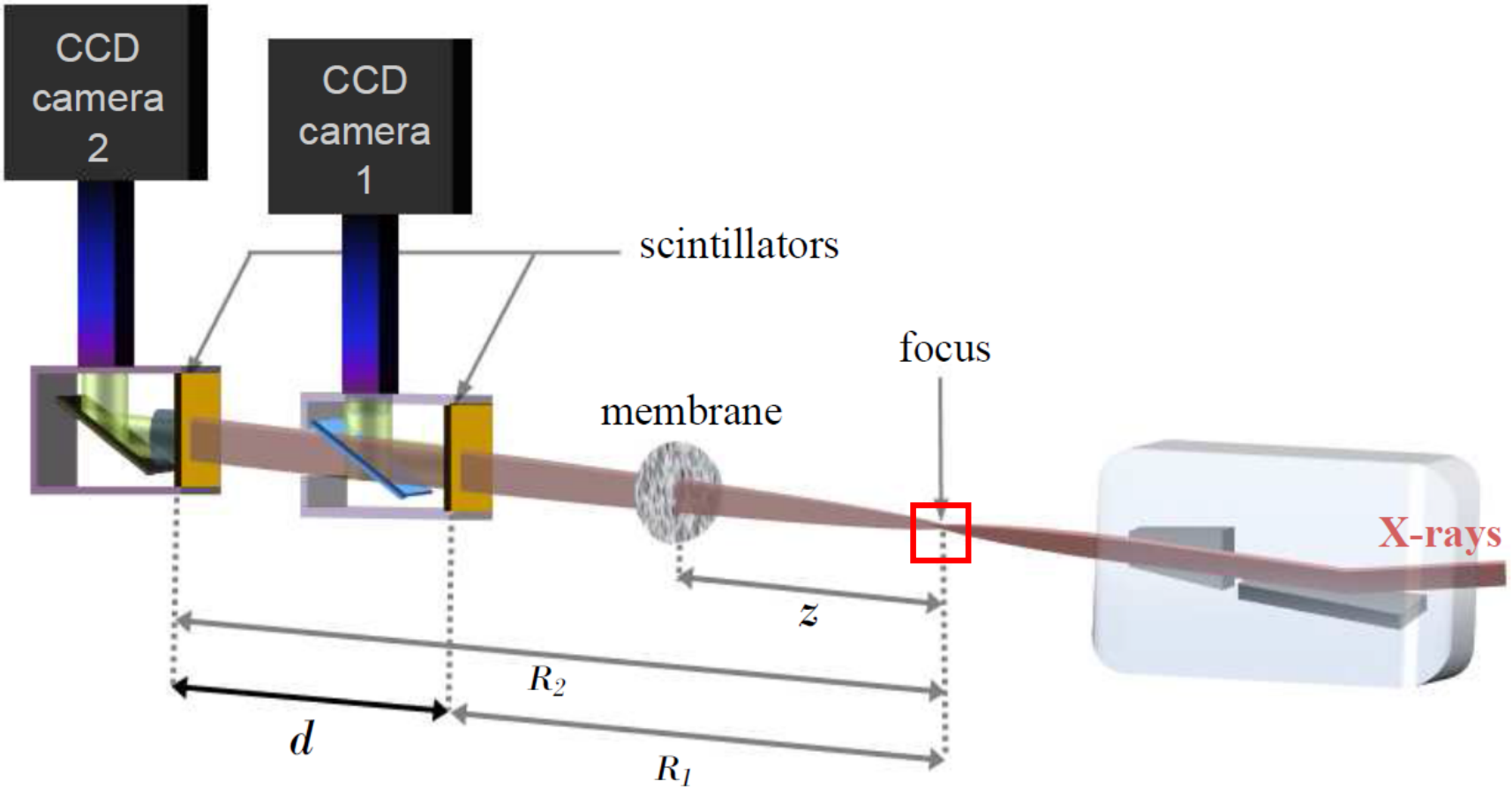
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



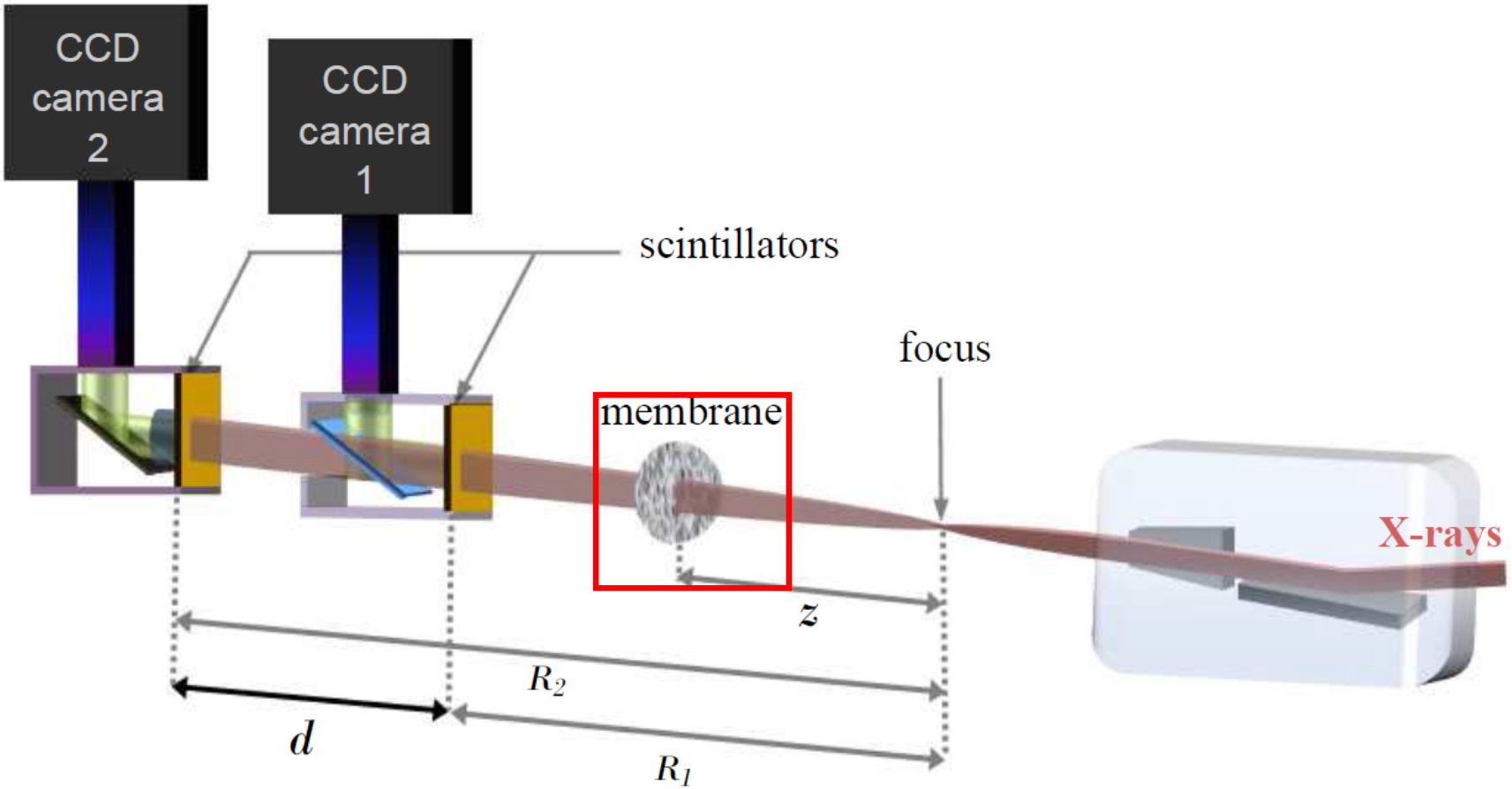
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



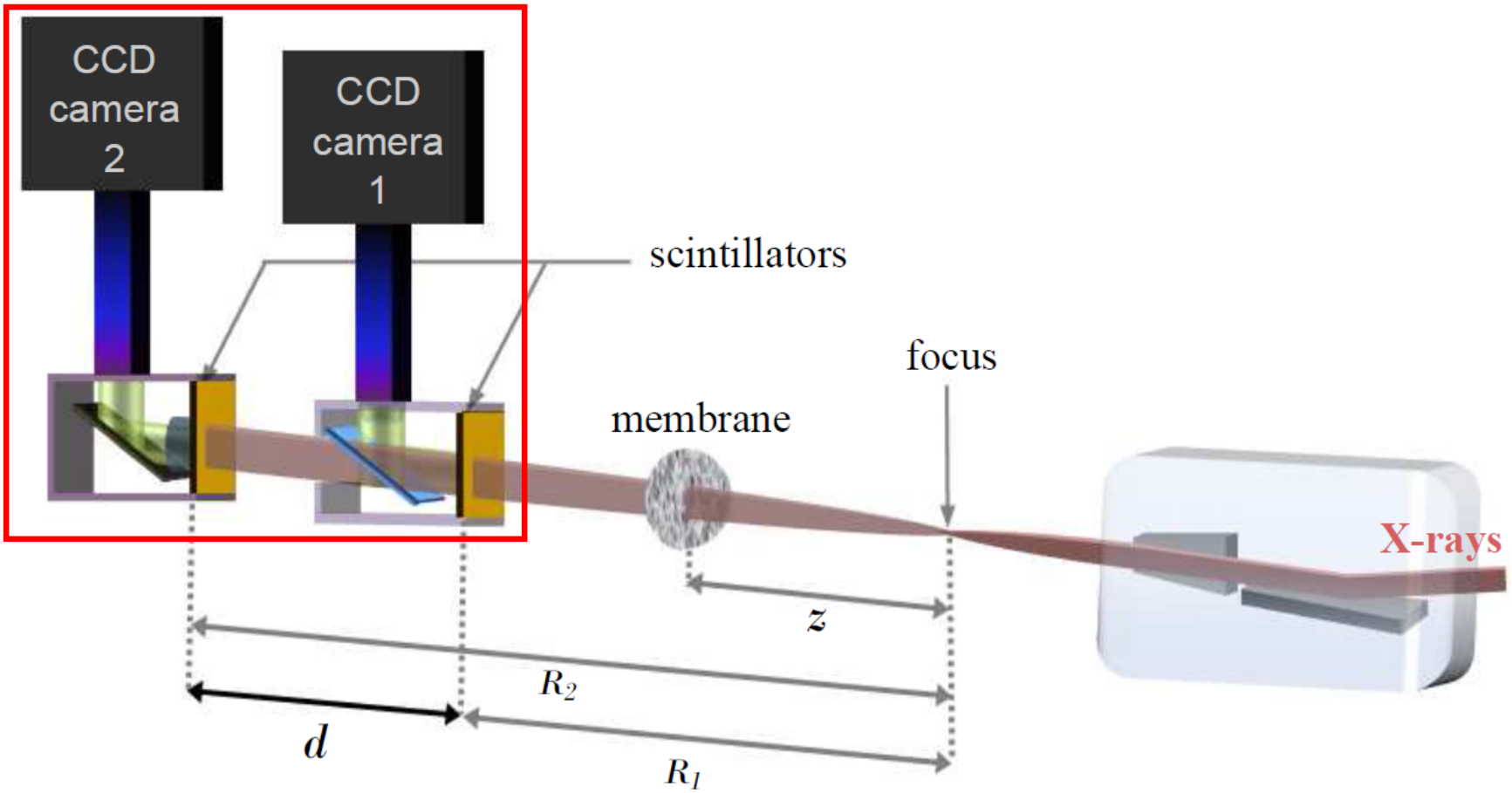
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



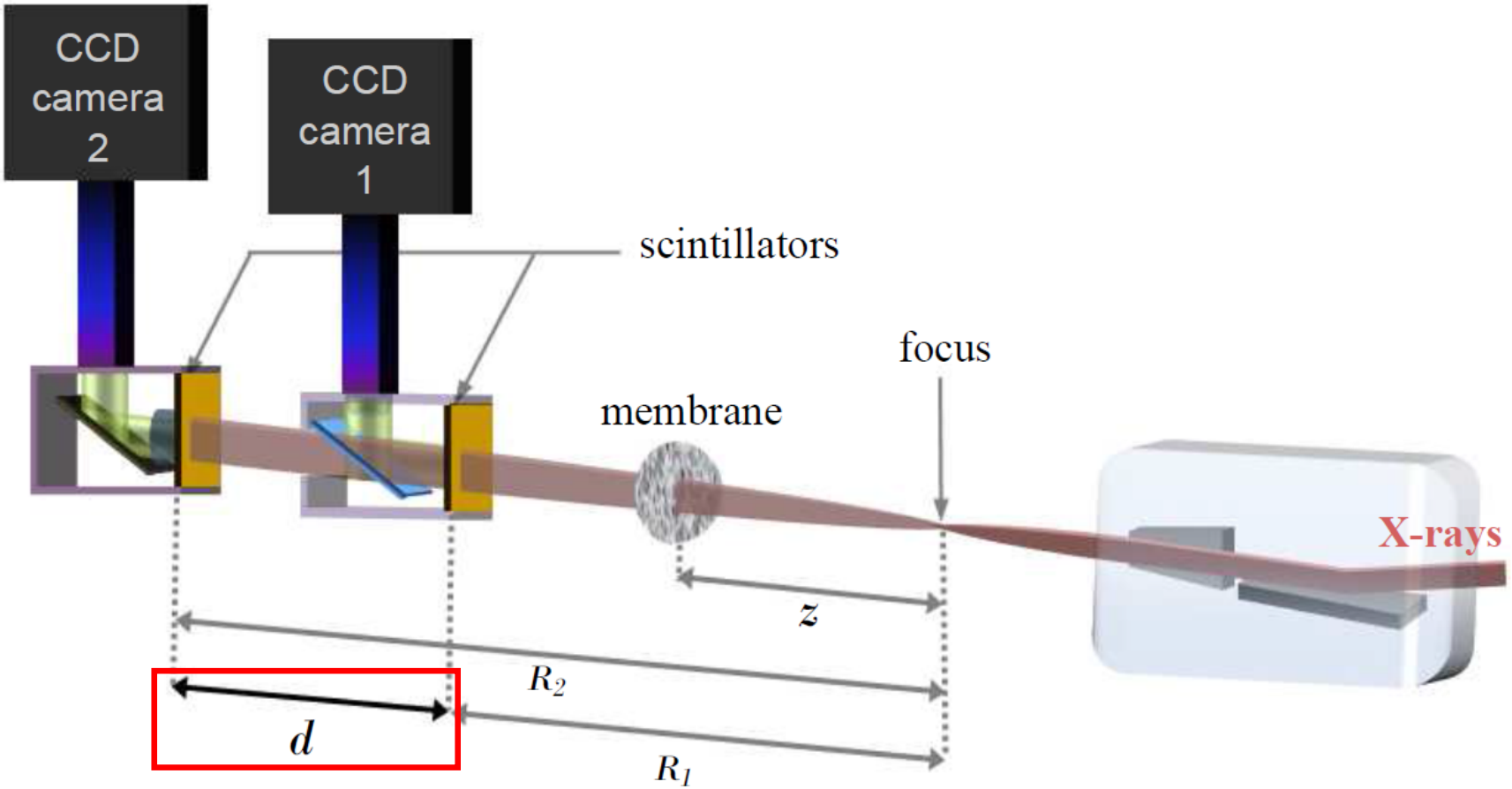
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



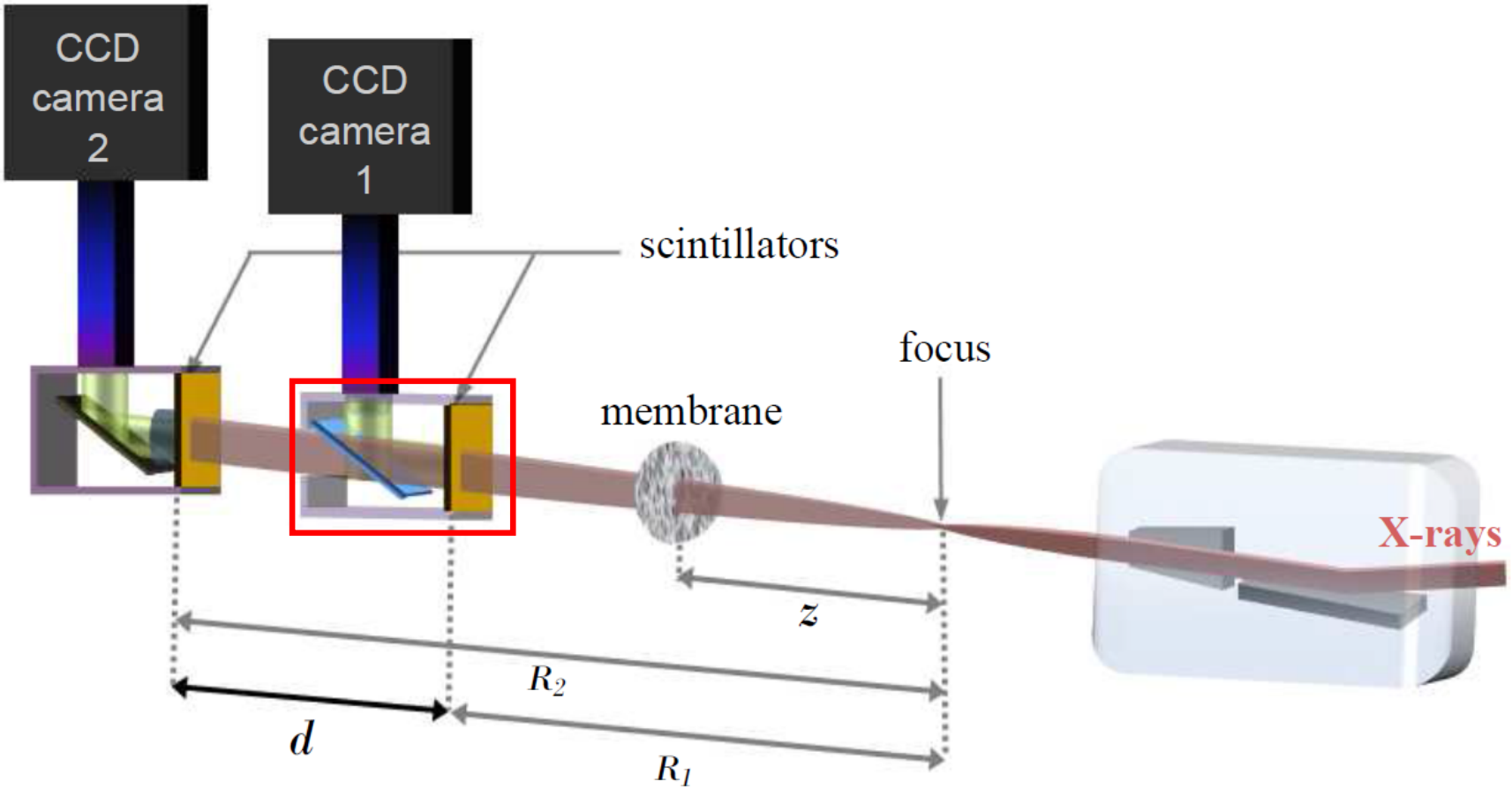
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



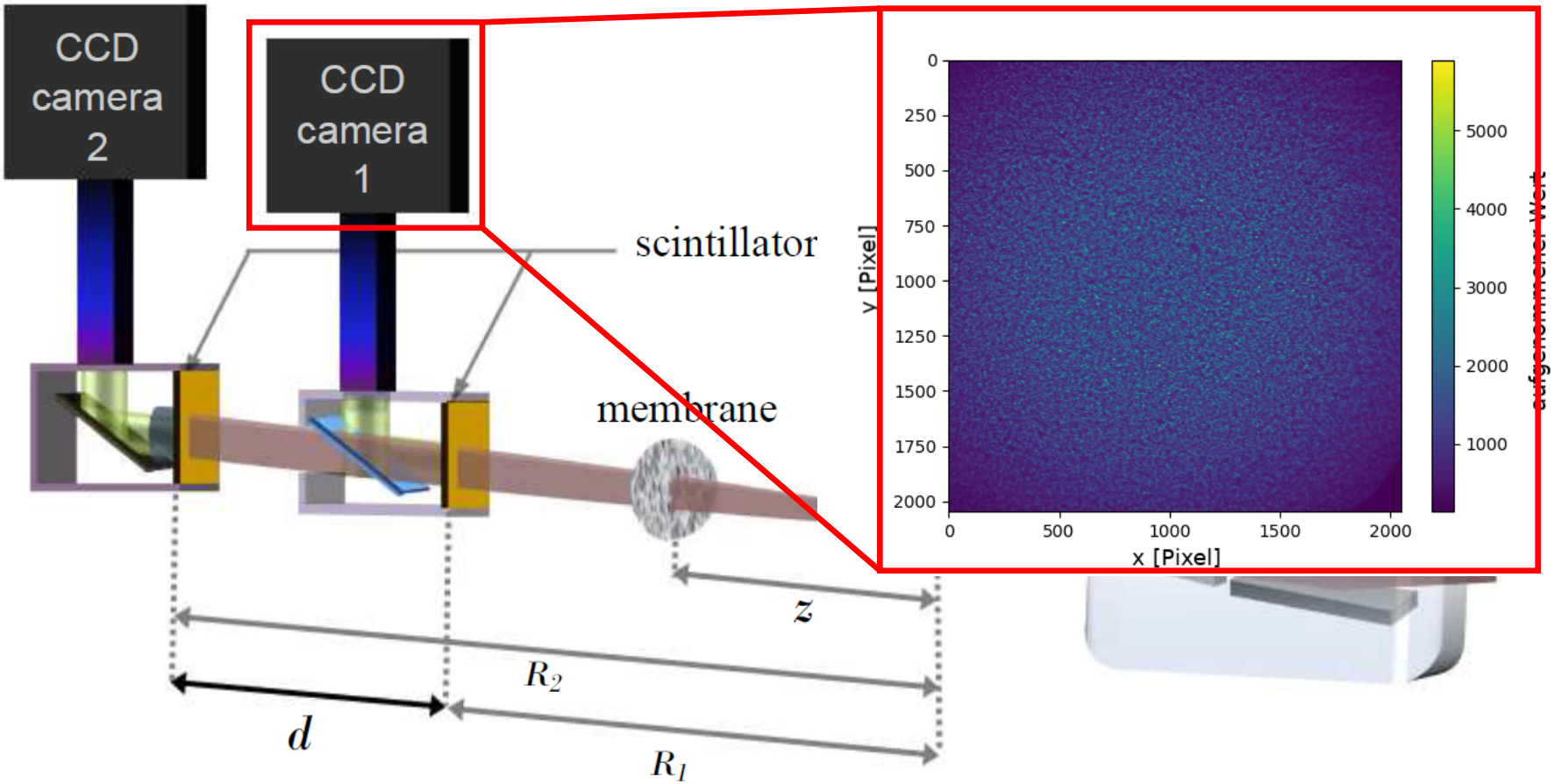
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



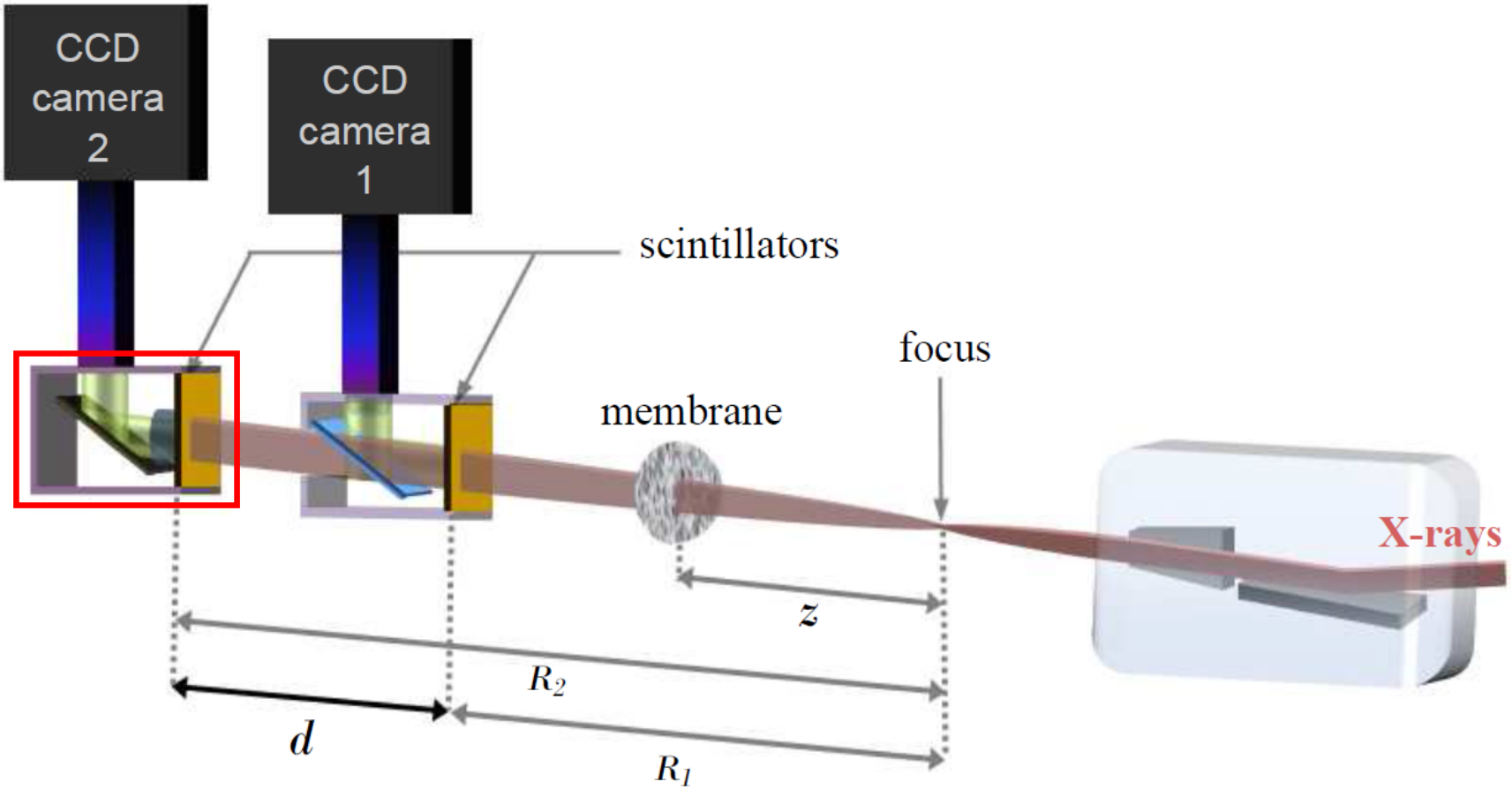
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



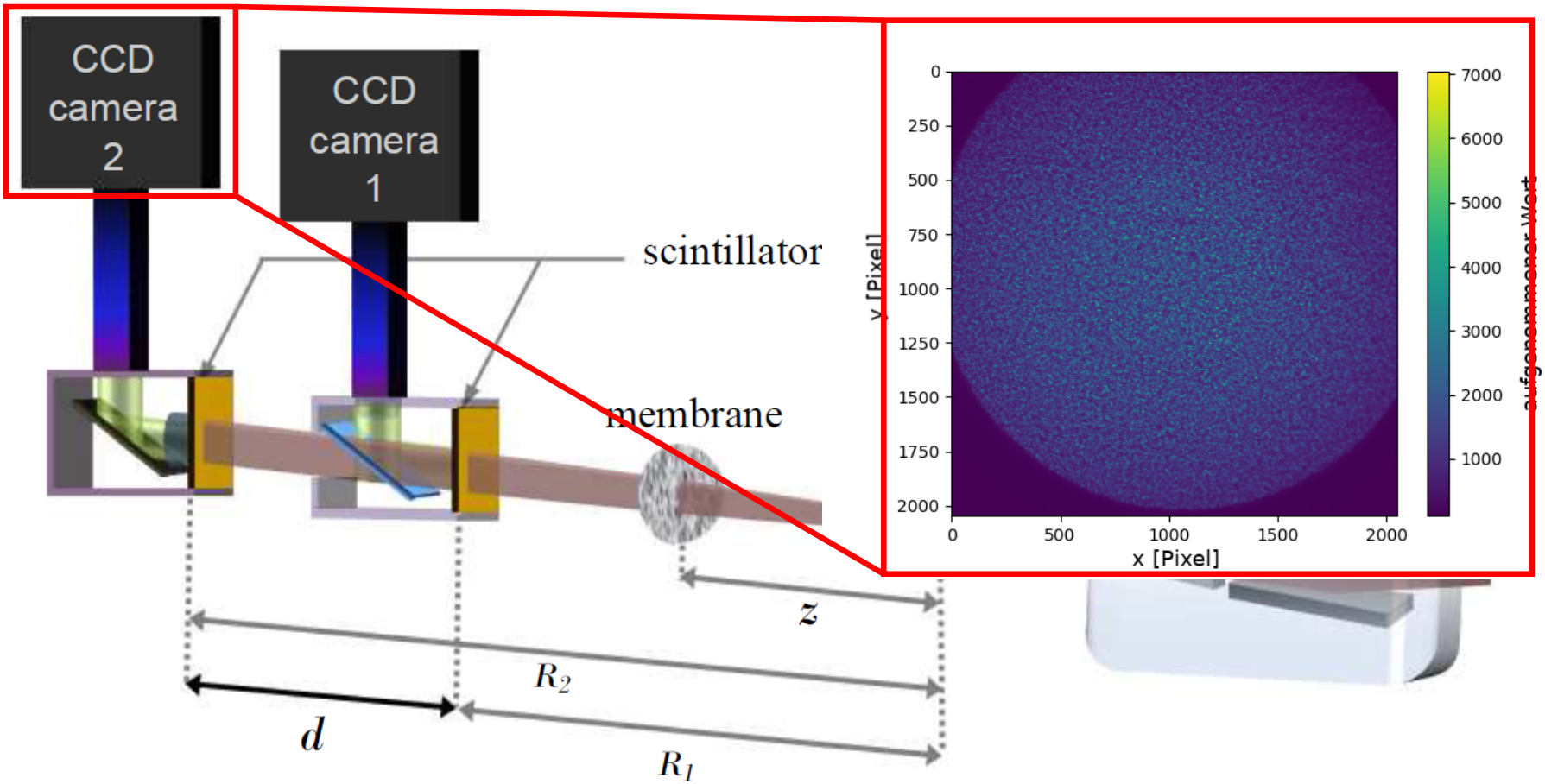
"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Versuchsaufbau



"X-ray pulse wavefront metrology using speckle tracking", Bérujon, Ziegler und Cloetens 2015, *Journal of Synchrotron Radiation*

Überblick

Zwei Phasen:

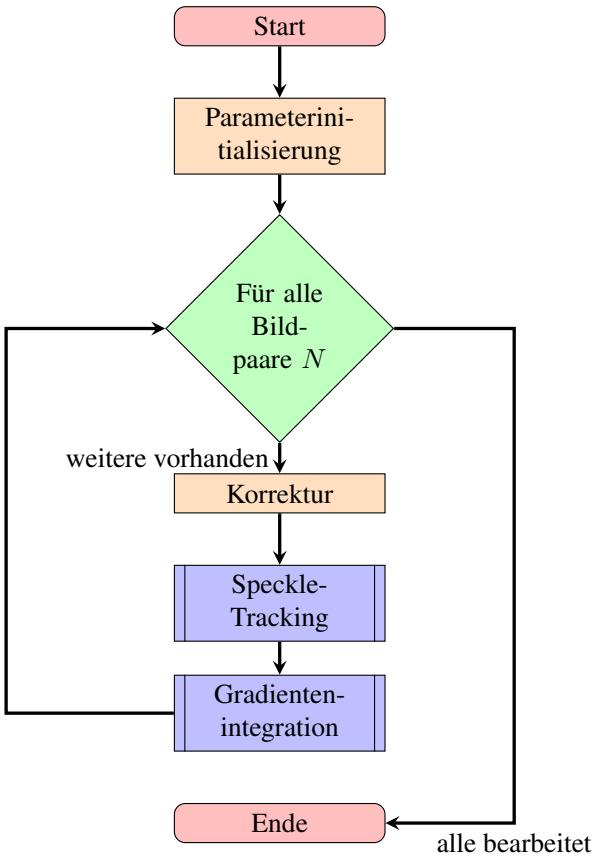
- Initialisierungsphase
  - Ermitteln des Kamerafehlers
  - Bestimmen Ablenkung in verschiedenen Positionen
    $\Rightarrow$ Grundlage für Hauptroutine
- Hauptroutine
  - Korrigieren von Kamerafehlern ($\rightarrow$ mit Ausgabe der Initialisierung)
  - Ablenkung nachverfolgen
  - Wellenfront rekonstruieren

Überblick

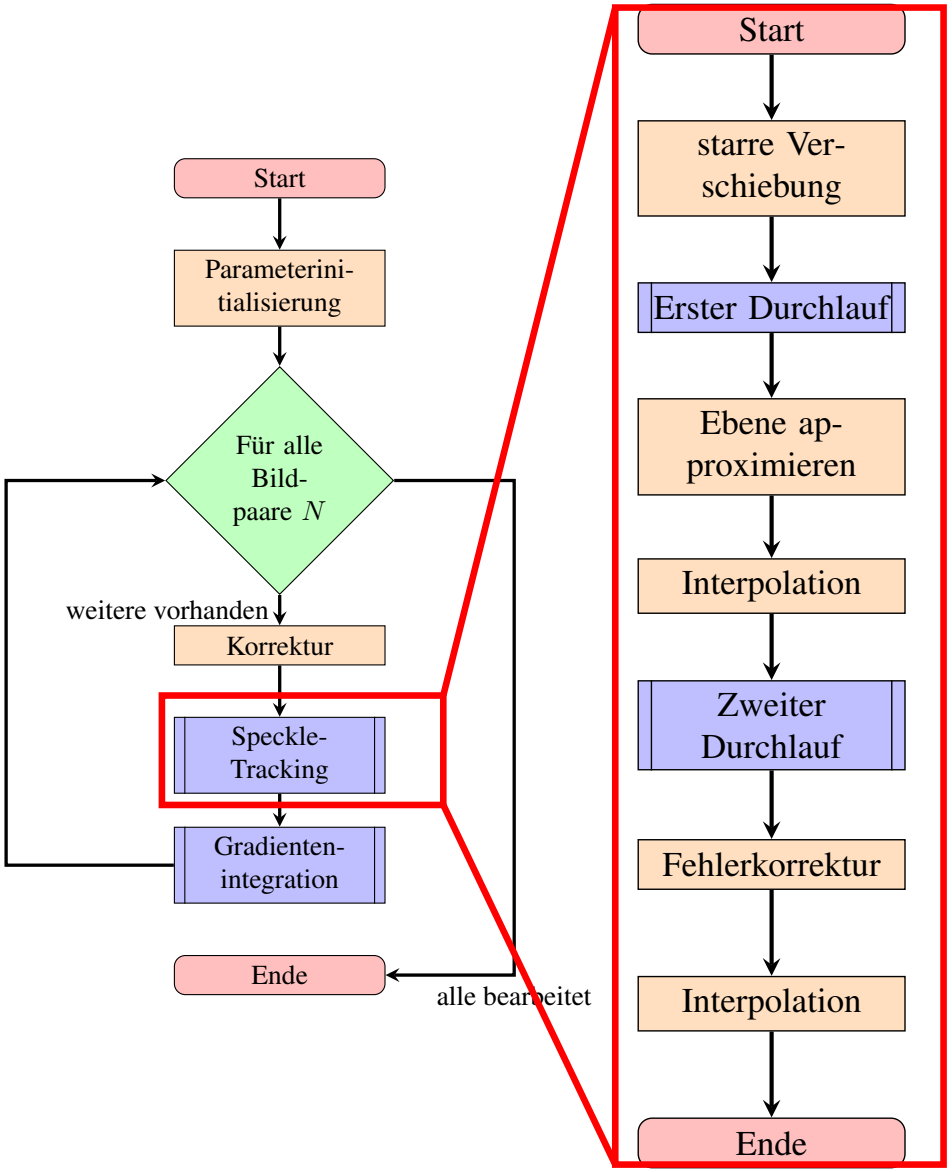
Zwei Phasen:

- Initialisierungsphase
  - Ermitteln des Kamerafehlers
  - Bestimmen Ablenkung in verschiedenen Positionen
    $\Rightarrow$ Grundlage für Hauptroutine
- Hauptroutine
  - Korrigieren von Kamerafehlern ($\rightarrow$ mit Ausgabe der Initialisierung)
  - Ablenkung nachverfolgen
  - Wellenfront rekonstruieren

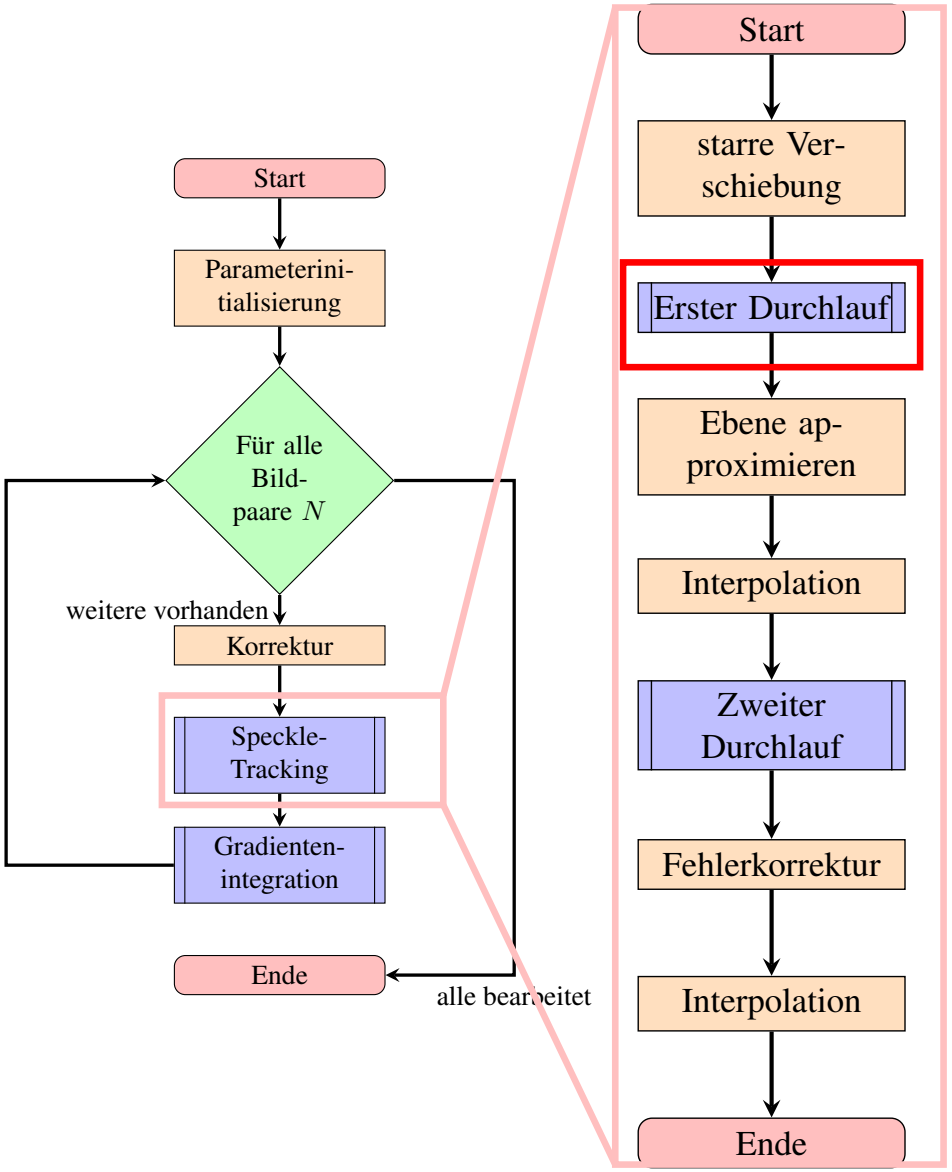
Hauptroutine



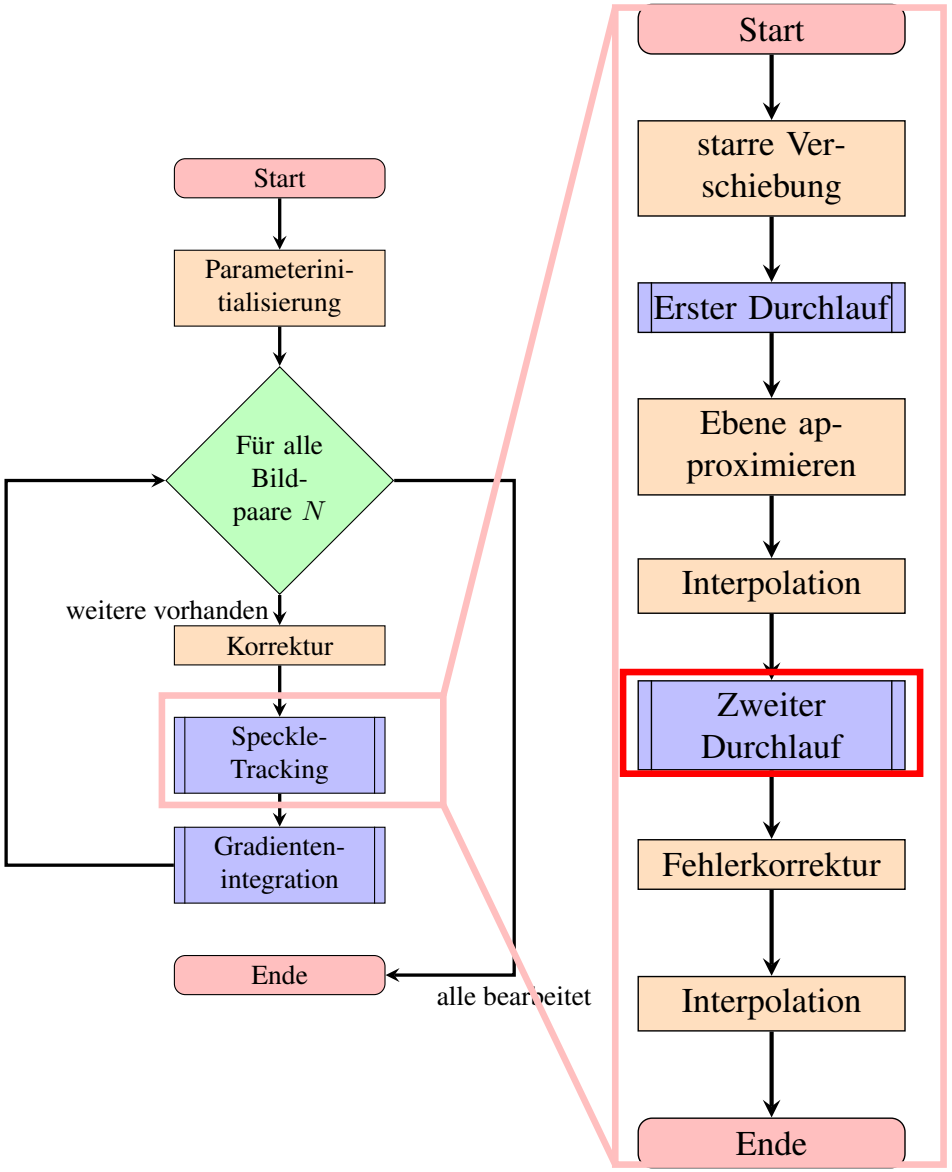
Hauptroutine



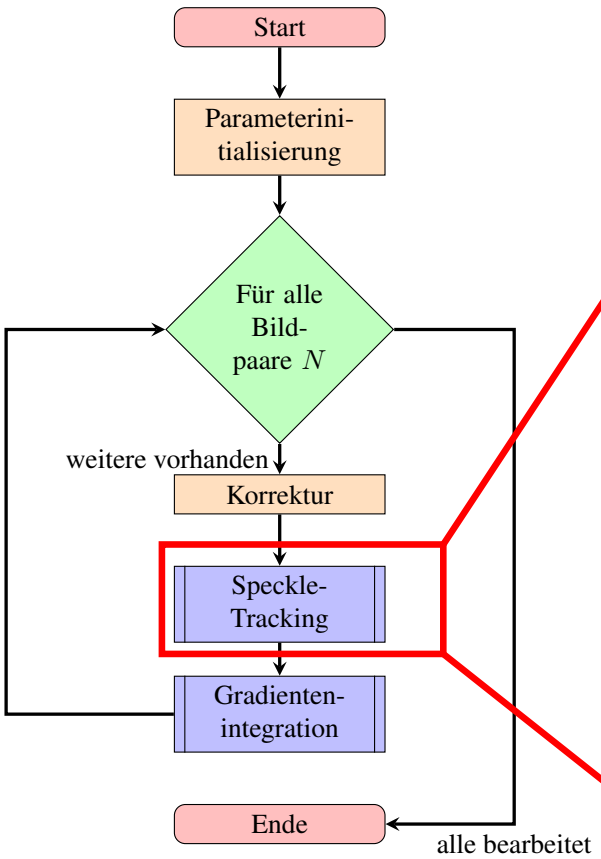
Hauptroutine

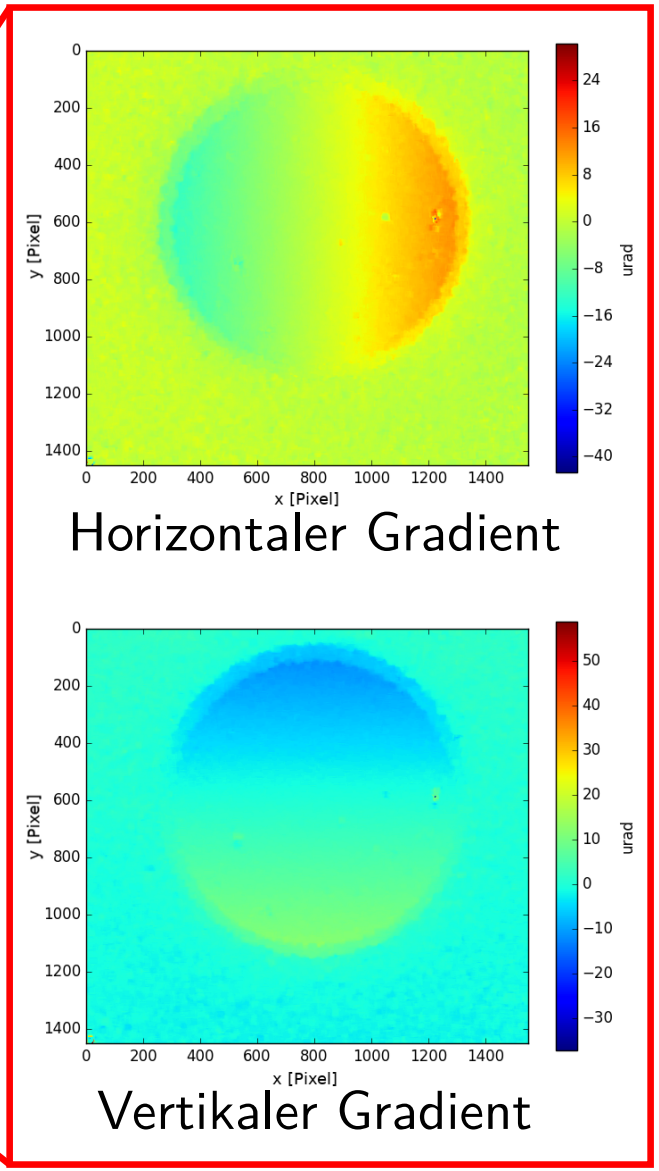


Hauptroutine

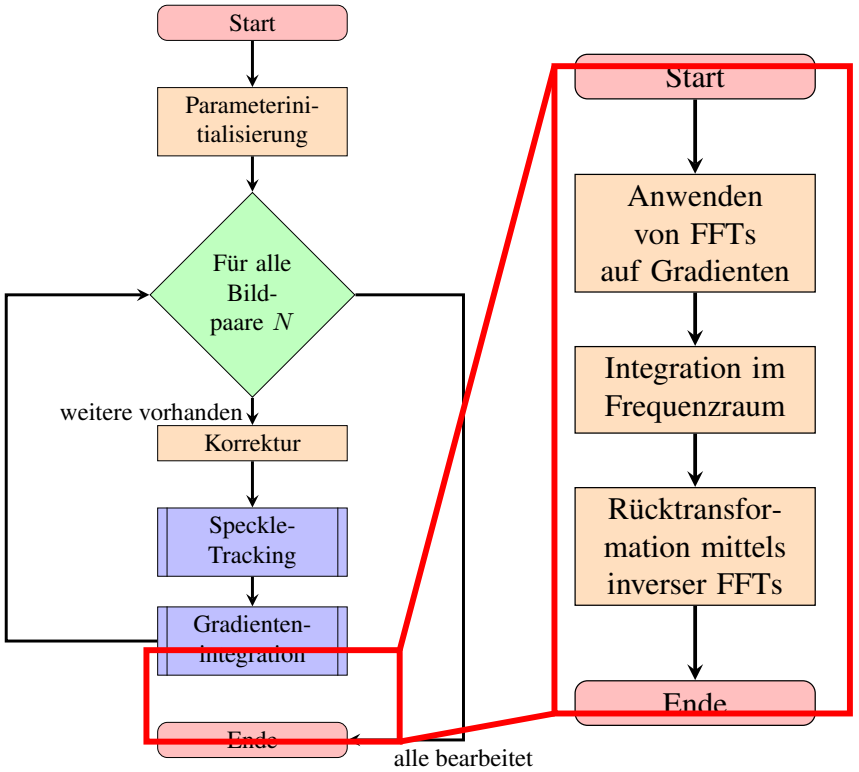


Hauptroutine





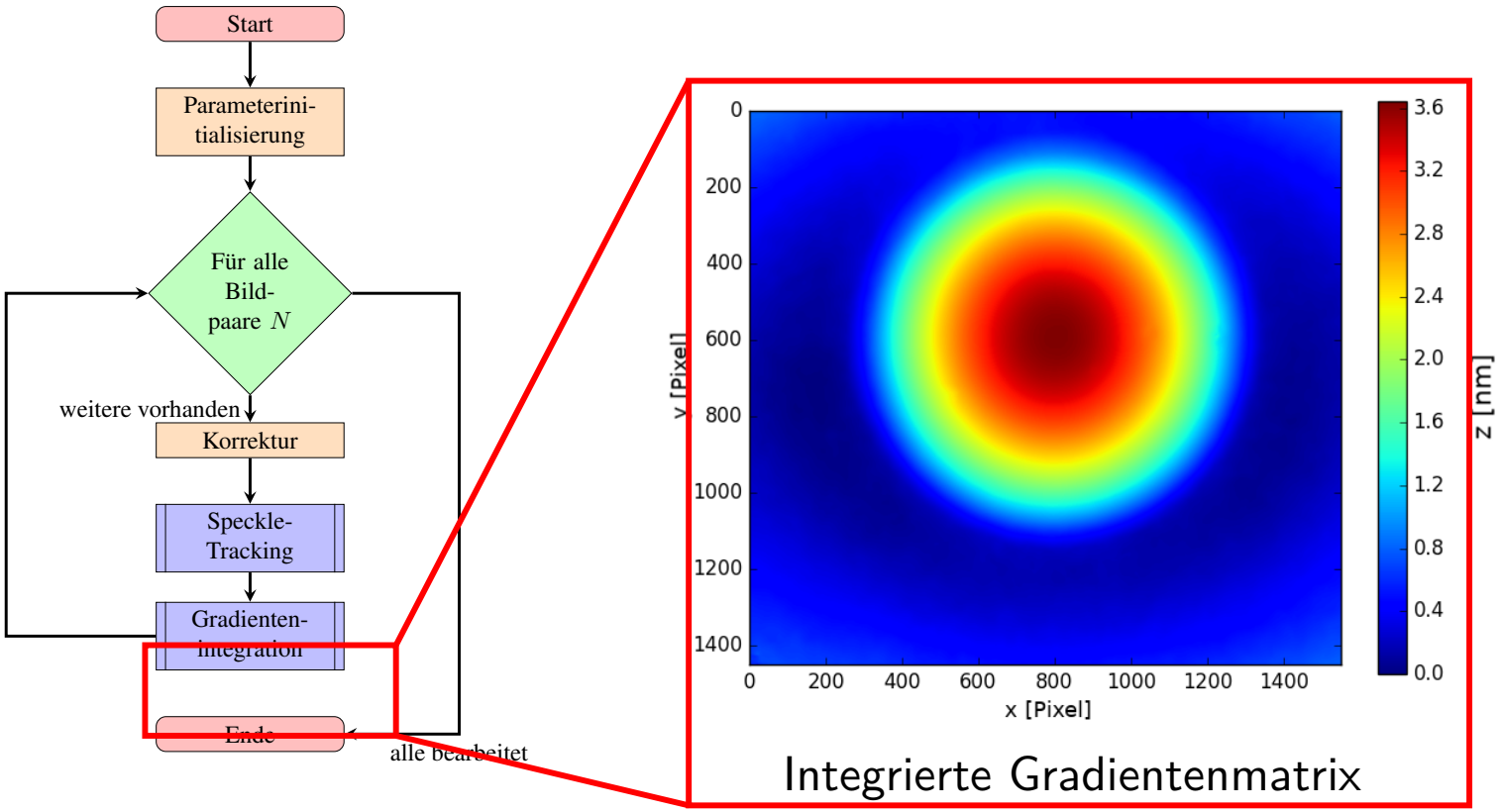
Hauptroutine



1

[1] "A method for enforcing integrability in shape from shading algorithms", Frankot und Chellappa 1988, *IEEE Transactions on Pattern Analysis and Machine Intelligence*

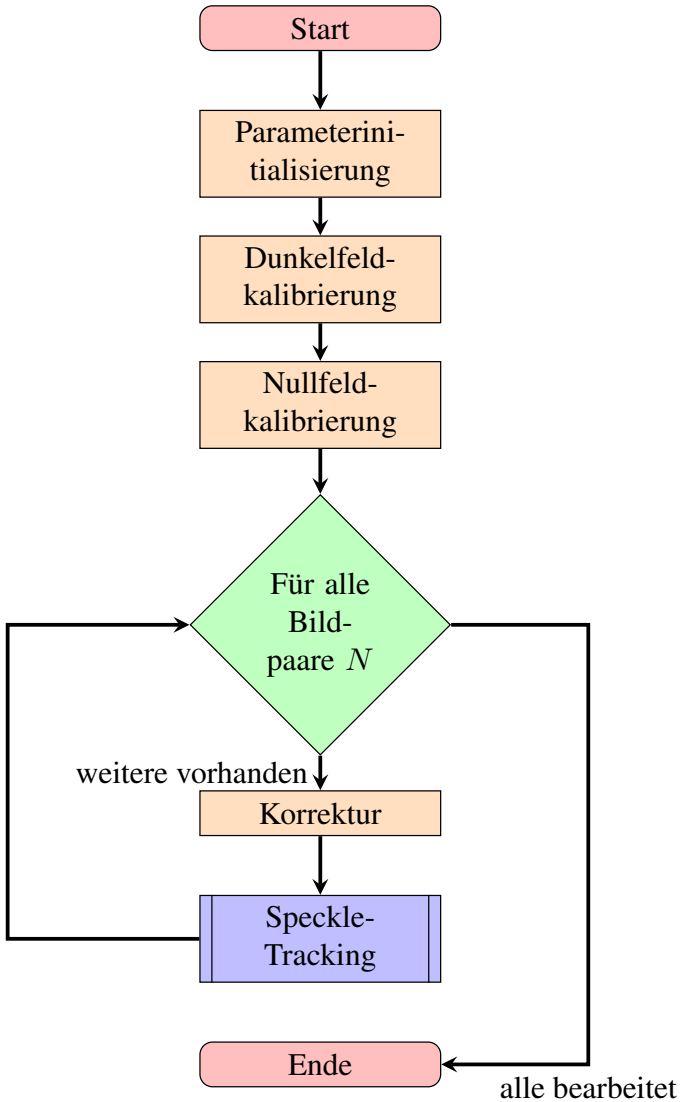
Hauptroutine



[1]

[1] "A method for enforcing integrability in shape from shading algorithms", Frankot und Chellappa 1988, *IEEE Transactions on Pattern Analysis and Machine Intelligence*

Initialisierung



Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Datensets

Initialisierung:

- Detector Distortion
  - ROI Größe: 1848x1848
  - Gitterauflösung: 4
  - Korrelationsgröße: 41

Hauptroutine:

- Experiment 6
  - ROI Größe: 1450x1450 (Bild), 550x550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 91
  - unterschiedliche Pixelgröße
- Lenses
  - ROI Größe: 1450x1550 (Bild), 1450x1550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 41
  - gleiche Pixelgröße

Datensets

Initialisierung:

- Detector Distortion
  - ROI Größe: 1848x1848
  - Gitterauflösung: 4
  - Korrelationsgröße: 41

Hauptroutine:

- Experiment 6
  - ROI Größe: 1450x1450 (Bild), 550x550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 91
  - unterschiedliche Pixelgröße
- Lenses
  - ROI Größe: 1450x1550 (Bild), 1450x1550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 41
  - gleiche Pixelgröße

Datensets

Initialisierung:

- Detector Distortion
  - ROI Größe: 1848x1848
  - Gitterauflösung: 4
  - Korrelationsgröße: 41

Hauptroutine:

- Experiment 6
  - ROI Größe: 1450x1450 (Bild), 550x550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 91
  - unterschiedliche Pixelgröße
- Lenses
  - ROI Größe: 1450x1550 (Bild), 1450x1550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 41
  - gleiche Pixelgröße

Datensets

Initialisierung:

- Detector Distortion
  - ROI Größe: 1848x1848
  - Gitterauflösung: 4
  - Korrelationsgröße: 41

Hauptroutine:

- Experiment 6
  - ROI Größe: 1450x1450 (Bild), 550x550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 91
  - unterschiedliche Pixelgröße
- Lenses
  - ROI Größe: 1450x1550 (Bild), 1450x1550 (Template)
  - Gitterauflösung: 1
  - Korrelationsgröße: 41
  - gleiche Pixelgröße

Gesamtlaufzeit I

Python 2.7, 2x Intel(R) Xeon(R) E5-2680 v3 (12 Kerne) @ 2.50GHz, kein MultiThreading

Gesamtlaufzeit II

Python 2.7, 2x Intel(R) Xeon(R) E5-2680 v3 (12 Kerne) @ 2.50GHz, kein MultiThreading

Gesamtlaufzeit III

Python 2.7, 2x Intel(R) Xeon(R) E5-2680 v3 (12 Kerne) @ 2.50GHz, kein MultiThreading

Profiling I

Profiling II

$\Rightarrow$ **über 95%**

Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Verarbeitung einzelner Bildpaare

content...

Innerhalb der Verarbeitung einzelner Bildpaare

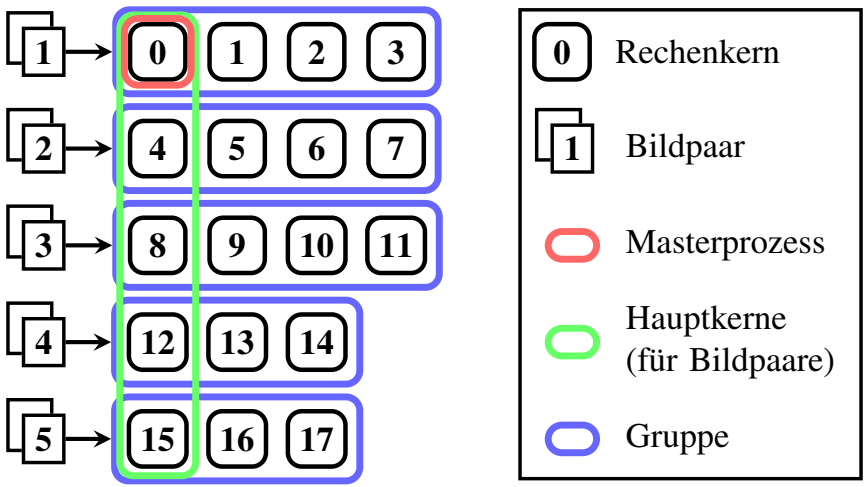


Abbildung: Verteilung von fünf Bildpaaren auf 18 Rechenknoten

Nutzen bereits optimierter Funktionen I

content...

Kompilieren - Gesamtes Programm I

asdf

Kompilieren - numba I

asdf

Kompilieren - Cython I

asdf

Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Möglichkeiten

is doch eh alles falsch

- Kompilieren **aber:** meist kein Speedup
- Parallelisierung **trivial möglich**
- Optimieren einzelner kritischer Pfade
  → C/C++ Portierung, optimierte Bibliotheken (numpy, numba, etc.), ...

Möglichkeiten

is doch eh alles falsch

- Kompilieren **aber:** meist kein Speedup
- Parallelisierung trivial möglich
- Optimieren einzelner kritischer Pfade
  → C/C++ Portierung, optimierte Bibliotheken (numpy, numba, etc.), ...

Möglichkeiten

is doch eh alles falsch

- Kompilieren **aber:** meist kein Speedup
- Parallelisierung **trivial möglich**
- Optimieren einzelner kritischer Pfade
  → C/C++ Portierung, optimierte Bibliotheken (numpy, numba, etc.), ...

Möglichkeiten

is doch eh alles falsch

- Kompilieren **aber:** meist kein Speedup
- Parallelisierung **trivial möglich**
- Optimieren einzelner kritischer Pfade
  → C/C++ Portierung, optimierte Bibliotheken (numpy, numba, etc.), ...

Möglichkeiten

is doch eh alles falsch

- Kompilieren **aber:** meist kein Speedup
- Parallelisierung **trivial möglich**
- Optimieren einzelner kritischer Pfade
  $\rightarrow$ C/C++ Portierung, optimierte Bibliotheken (numpy, numba, etc.), ...

Gliederung

1. Hinweise
2. Evaluierung des Wellenfrontrekonstruktionsalgorithmus
3. Performance-Analyse des derzeit fast durchgängig seriellen Codes
4. Parallelisierung der kritischen Abschnitte
5. Parallelisierung der kritischen Pfade für Vielkernarchitekturen
6. Stand der Arbeit

Jetziger Stand

is doch eh alles falsch **Bereits erledigt:**

- Evaluierung
- Performance-Analyse
- Kompilieren

In Bearbeitung:

- Parallelisieren
- Optimieren von reinem Python-Code

Jetziger Stand

is doch eh alles falsch **Bereits erledigt:**

- Evaluierung
- Performance-Analyse
- Kompilieren

In Bearbeitung:

- Parallelisieren
- Optimieren von reinem Python-Code

Weiterer Plan

Noch geplant:

- Optimierung kritischer Pfade
  - vermehrte Verwendung optimierter Bibliotheken
  - C/C++ Portierung einzelner Funktionen
- Validierung der parallelen Implementierung
- Performance-Messung der parallelen Implementierung

Probleme

- Bugs in Referenzimplementierung
- Python Multithreading
  - viele Profiler für Python, wenige mit Multithreading-Support
  - Forken der kompletten Python-Umgebung nötig
- schlechte Kompilierergebnisse

Probleme

- Bugs in Referenzimplementierung
- Python Multithreading
  - viele Profiler für Python, wenige mit Multithreading-Support
  - Forken der kompletten Python-Umgebung nötig
- schlechte Kompilierergebnisse

Probleme

- Bugs in Referenzimplementierung
- Python Multithreading
  - viele Profiler für Python, wenige mit Multithreading-Support
  - Forken der kompletten Python-Umgebung nötig
- schlechte Kompilierergebnisse

Probleme

- Bugs in Referenzimplementierung
- Python Multithreading
  - viele Profiler für Python, wenige mit Multithreading-Support
  - Forken der kompletten Python-Umgebung nötig
- schlechte Kompilierergebnisse

Probleme

- Bugs in Referenzimplementierung
- Python Multithreading
  - viele Profiler für Python, wenige mit Multithreading-Support
  - Forken der kompletten Python-Umgebung nötig
- schlechte Kompilierergebnisse

Weiterführende Literatur I

- Bérujon, Sébastien (2013). "Métrologie en ligne de faisceaux et d'optiques X de synchrotrons". Thèse de doctorat dirigée par Ziegler, Eric et Sawhney, Kawal Physique Grenoble 2013. Diss. Université de Grenoble. URL: http://www.theses.fr/2013GRENY010.
- Bérujon, Sébastien, Eric Ziegler, Roberto Cerbino u. a. (Apr. 2012). "Two-Dimensional X-Ray Beam Phase Sensing". In: *Phys. Rev. Lett.* 108 (15), S. 158102. DOI: 10.1103/PhysRevLett.108.158102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.108.158102.
- Bérujon, Sébastien, Eric Ziegler und Peter Cloetens (2015). "X-ray pulse wavefront metrology using speckle tracking". In: *Journal of Synchrotron Radiation* 22.4, S. 886–894. ISSN: 1600-5775. DOI: 10.1107/S1600577515005433. URL: http://dx.doi.org/10.1107/S1600577515005433.

Weiterführende Literatur II

Frankot, R. T. und R. Chellappa (Juli 1988). “A method for enforcing integrability in shape from shading algorithms”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.4, S. 439–451. ISSN: 0162-8828. DOI: 10.1109/34.3909.