

# Erweiterung der Funktionalität von Raytracing-Software

Bernhard Michel

Ein herzliches Dankeschön an:

B. Berneker, R. Hermann, Ch. Horneber, M. Jahny, M. Kroneberger, I. Zupevc  
(fx64)

Hembach **Photonik**

# Übersicht

- Raytracing-Software
- Erweiterung der Funktionalität bestehender Software (Nutzung vorhandener Schnittstellen)
- Erweiterung der Kernfunktionalität von Raytracern – Entwicklung neuer Software

# Raytracing-Software

Sequentielles und nicht-sequentielles Raytracing, Funktionsumfang,  
Kommerzielle Raytracer im Überblick

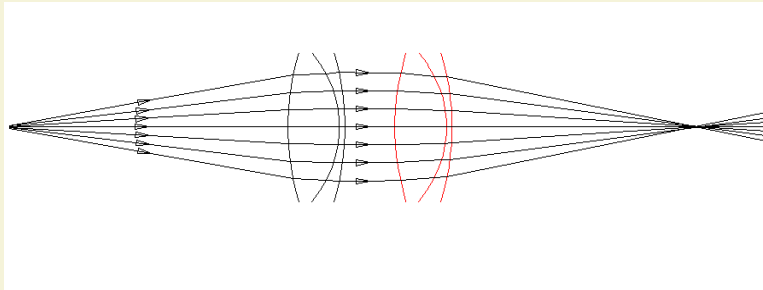
# Sequentielle und nicht-sequentielle Raytracer

- Standard-Werkzeug für Optikdesign- und Analyse sind Strahlverfolgungsprogramme („Raytracer“)
- Raytracing: Lichtausbreitung in einem optischen System durch Strahlen

## **Sequentielle Raytracer:**

Strahlen treffen Objekte in vordefinierter Reihenfolge

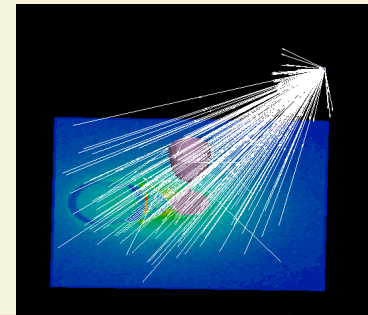
- Linsendesign
- Abbildungsoptik



## **Nicht-Sequentielle Raytracer:**

Strahlen treffen jeweils nächstes in ihrem Weg liegendes Objekt

- Allgemeinere Aufgaben (Beleuchtungsoptik, Störlichtanalyse, **Radiometrie**)

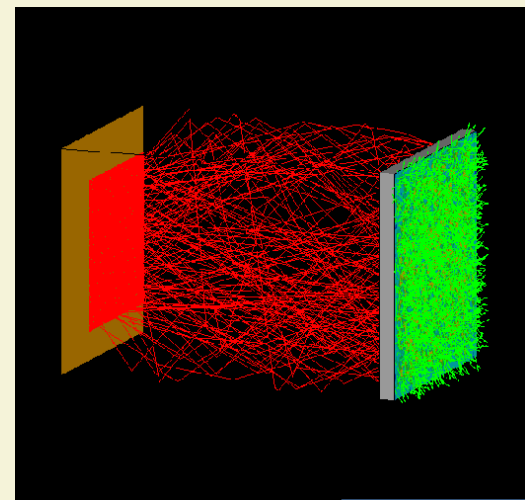
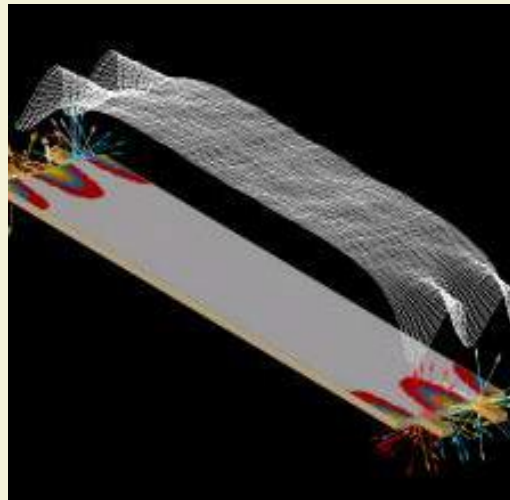
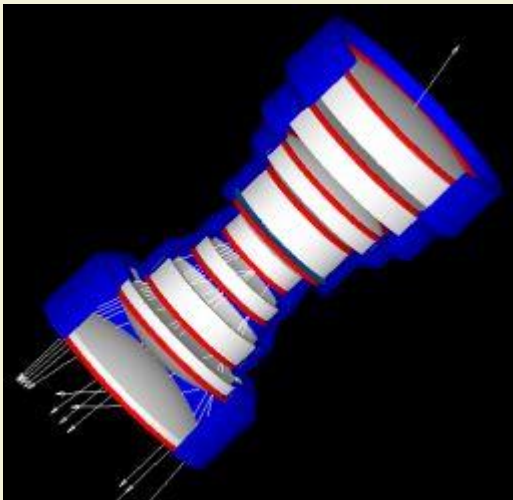


**Dieser Vortrag:** *nicht-sequentielle Raytracer*, insbesondere ASAP, ZEMAX und LambdaSpect/eigene Software

Hembach **Photonik**

# Beispiel ASAP

- Beliebige 3-dimensionale optische/mechanische Systeme; Raytracing und Analyse
- Grafik für Visualisierung der Geometrie, Strahlengänge und Analyseergebnisse
- Lichtstreuung, Beugung, Reflexion, Brechung, Absorption, Polarisation
- Nicht-sequentielles Raytracing, Gaußstrahl-Propagation, Finite-Difference Beam Propagation



**Aufgabe:** Quantitative Analyse, genaue Voraussage, Erklärung der Eigenschaften eines optischen Systems.

Quelle:  
Breault Research

Hembach **Photonik**

# Kommerzielle Raytracer im Überblick

<b>Software</b>	<b>Hersteller</b>	<b>Bemerkung</b>
ASAP	Breault Research Organization	
APEX	Breault Research Organization	Integriert in SolidWorks
Code V	Synopsis	Linsendesign
FRED	Photon Engineering	
LambdaSpect	fx64	integriert in Inventor
LightTools	Synopsis	
LucidShape	Brandenburg	
OpTaLix	Optenso	Sequenziell/nichtsequenziell
OptiCAD	OptiCAD Corporation	
OptisWorks	Optis	integriert in SolidWorks
OSLO	Sinclair Optics	Linsendesign
Photopia	LTI Optics	
Simulux	Infotec	
SPEOS	Optis	
SPEOS CAA V5 based	Optis	integriert in Catia V5
Spray W.Theiss	Hard- and Software	
TracePro	Lambda Research	
ZEMAX	Radiant ZEMAX	Sequentiell/nichtsequenziell

Überblick unvollständig!

Violett dargestellte Raytracer werden genauer besprochen



# Erweiterung der Funktionalität bestehender Software

Stärken und Schwächen von Raytracern, Ankopplung externer Software

# Stärken und Schwächen von Raytracern

## Stärken

- Nahezu beliebige Systemgeometrien und optische Eigenschaften
- Meist gute Anbindung an CAD-Umgebung
- Ergebnisse für viele Zwecke ausreichend genau und zuverlässig
- Erhöhung der Genauigkeit durch Erhöhung der Strahlenzahl
- Objektiv – in Ergebnisse gehen (idealerweise) keine a-priori Erwartungen ein

## Schwächen

- Wellenoptik nur eingeschränkt möglich
- Viele Aufgabenstellungen erfordern Nicht-Standard-Lösungen: in kommerziellen Raytracern nicht implementiert
- Genauigkeit für viele Aufgaben nicht ausreichend
- Zeitaufwändig: Rechenzeit als Show-Stopper
- Statistischer Fehler in radiometrischen (photometrischen) Ergebnissen skaliert typischerweise mit  $\frac{1}{\sqrt{N}}$  (N: Zahl der Strahlen)  
=> extrem langsame Konvergenz



# Was tun, wenn die Funktionen des Raytracers nicht ausreichen?

- Hersteller fragen: „Feature request“  
meist nur dann erfolgreich, wenn es sich um ein noch fehlendes „Standardfeature“ der Software handelt
- Verwenden externer Software zur Input-Erzeugung und zur Analyse der Ergebnisse (Pre- und Post-Processing)
- „Fernsteuern“ des Raytracers mittels externer Software, zum Beispiel zur Optimierung
- „Add-ons“ – zusätzliche Funktionen, die vom Raytracer verwendet werden, zum Beispiel DLLs (dynamische Linkbibliotheken)

# Entsprechender Zeitschriftenartikel aus der Zeitschrift Photonik (Mai 2012)

Optiksoftware

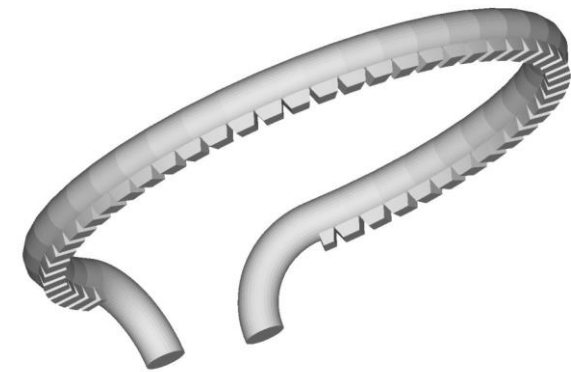
## Kopplung von Raytracern mit Spezialsoftware

Bernhard Michel, Monika Kroneberger, Robert Hermann,  
Hembach Photonik GmbH, Rednitzhembach

Die Entwicklung optischer Systeme basiert heute oft auf Strahlverfolgung mit kommerziellen Raytracing-Programmen. Über ihre grafischen Benutzeroberflächen können Standardaufgaben einfach und effizient gelöst werden. Bei komplexeren Aufgabestellungen kann der Nutzer auf integrierte Skriptsprachen zurückgreifen. Maximale Flexibilität für kreative Raytracing-Lösungen sowie für die Automatisierung von Design und Analyse erhält man jedoch, indem man die Software über die dafür vorgesehenen Schnittstellen mit spezialisierten Programmen, Skript-Routinen und dynamischen Link-Bibliotheken koppelt.

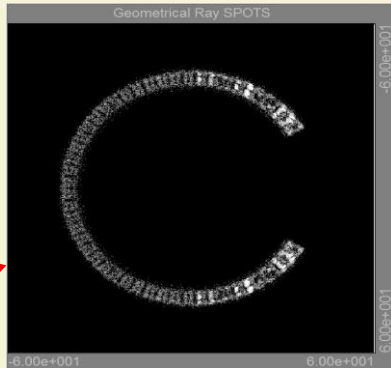
Hembach **Photonik**

# Beispiel: Lichtleiteroptimierung mit Python und ASAP

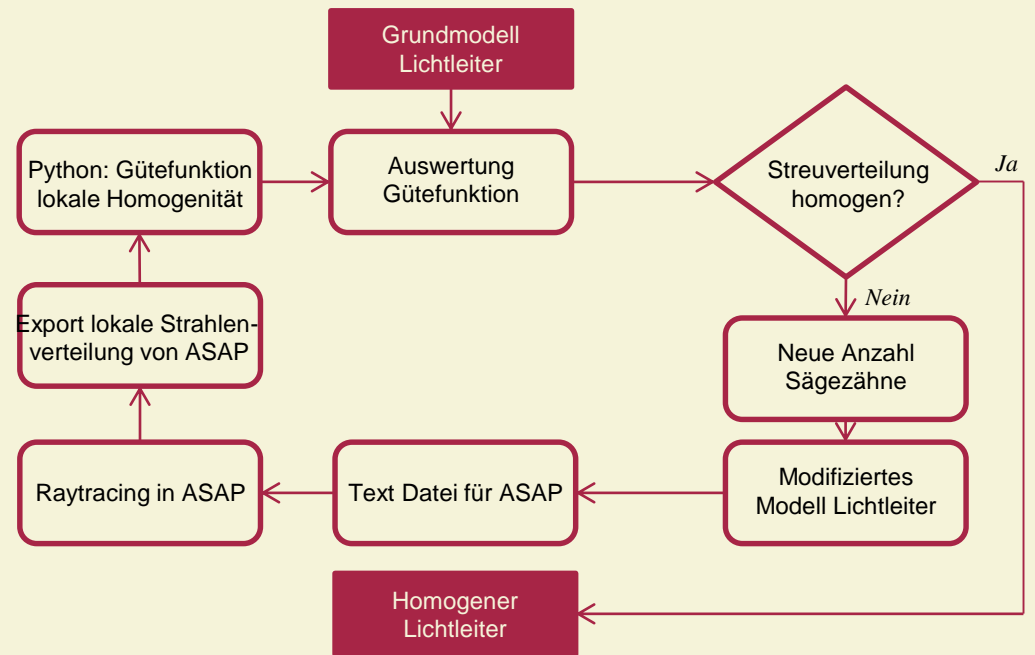
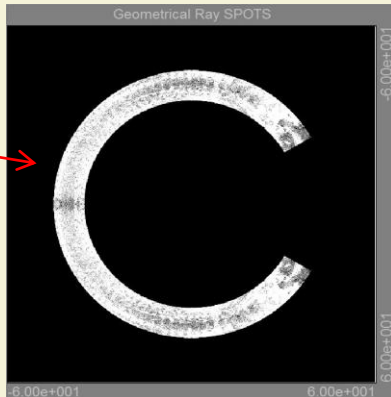


Python zur Fernsteuerung von ASAP

vorher



nachher:  
deutlich heller  
und homogener!



Optimierungsalgorithmen: Particle Swarm, Simulated Annealing, Genetische Algorithmen etc.



# Beispiel: Stavroudis-Baffle

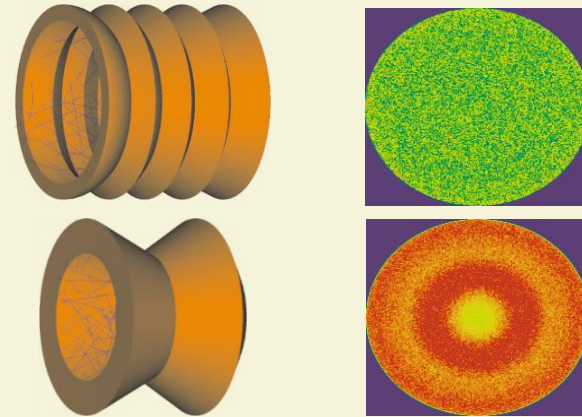
## Auslegung mit Python und ZEMAX

### Stavroudis Baffle:

- Hochreflektierende Blende
- Alles Licht, das auf die Blende trifft, wird wieder nach außen zurückreflektiert
- Hervorragend für MID-IR-Optik (thermische Strahlung)
- Hohe Anforderungen an Form und Materialqualität

### Warum Python?

- Große numerische Bibliothek
- Einfache Programmiersprache
- Freeware



### Wie funktioniert die Ansteuerung?

- Schnittstelle: DDE
- Python agiert als Server
- ZEMAX ist Client
- Vollständiger Zugriff auf Befehlsbibliothek von Zemax
- Geometrieerstellung in Python, Raytracing und Analyse in ZEMAX

# Erweiterte Funktionalität durch benutzerdefinierte DLLs

## **ASAP:**

- Eine einzige Datei (Userprog.dll) kann vom Nutzer überschrieben werden
- Definiere neue Streumodelle, benutzerdefinierte Oberfläche etc.
- Programmierung nur mit FORTRAN einfach

## **ZEMAX:**

- Ganze Bibliothek benutzerdefinierter DLLs vorhanden
- Beispiel: Mie-Streuung
- Sehr flexibel
- Erfahrung nötig zum Programmieren

# Zusammenfassung

- Kopplung kommerzieller Raytracer mit externer Software wurde am Beispiel von ASAP und ZEMAX dargestellt
- Ähnliche Methodik auch bei anderen Software-Produkten möglich
- Wesentliche Erweiterung des Funktionsumfangs möglich
- Alternative: falls die Software über ausreichend hoch entwickelte, standardisierte Skriptsprache verfügt, können viele Aufgaben direkt im Raytracer erledigt werden

# Erweiterung der Kernfunktionalität von Raytracern – Entwicklung neuer Software

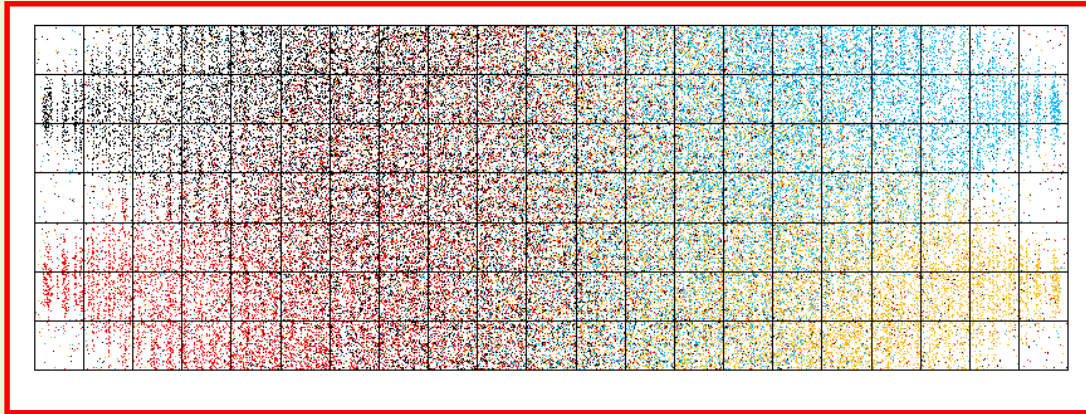
Probleme konventioneller Raytracer, Differentielles Raytracing, Einsatz der GPU

# Probleme konventioneller Raytracer

- Für viele Aufgaben sind Raytracer (z. T. methodenbedingt) zu langsam:
  - Leuchtdichteberechnung
  - Optimierung mit zum Teil hunderten oder Tausenden Iterationsschleifen
  - Störlichtanalyse
- ... oder zu ungenau:
  - Berechnung der Gütefunktion bei Optimierung (Rauschen!)
  - Anwendungen in der Messtechnik
- Lösungsansätze:
  - Differenzielles Raytracing
  - Raytracing auf GPU (Grafikkarte)



# Radiometrie/Photometrie in „normalen“ Raytracern



Grundproblem aller „konventionellen“ Raytracer:  
**Rauschen!**

Faustregel: Relativer Fehler gemäß Poisson-Statistik

$$\frac{1}{\sqrt{N}}$$

(N: Zahl der Strahlen im Pixel)  
100 mal mehr Strahlen  
=> 10 mal geringerer Fehler  
=> extrem ineffizient bei hohen Genauigkeitsanforderungen

- Häufig: Strahlen werden mit Monte-Carlo-Verfahren durch das optische System verfolgt
- Strahlen werden auf einer Detektorfläche in Pixel einsortiert
- Gesamtfluss aller Strahlen in einem Pixel pro Pixelfläche ergibt Bestrahlungs-/Beleuchtungsstärke
- Ähnliches Vorgehen bei anderen radiometrischen/photometrischen Größen

# Differenzielles Raytracing

- Statt einem Strahl wird ein (evtl. infinitesimales) Strahlenbündel durch System verfolgt
- Bündel „transportiert“ keinen Lichtstrom, sondern eine Lichtstrom*dichte* (Strahldichte, Strahlstärke, Bestrahlungsstärke)
- Grundlegende Idee stammt aus dem 19. Jahrhundert
- Im Idealfall reicht bereits ein Strahl, um die Bestrahlungsstärke auf einem Detektor exakt zu berechnen.
- ~~Existierende Implementierungen: Raytrace, CodeV, m.E. auch ASAP~~
  - Im Folgenden wird ein neuartiger Weg zur Implementierung vorgestellt, der viele der bekannten Probleme umgeht.
  - Aus unserer Sicht hat differenzielles Raytracing ein großes Potential und macht einige Aufgabenstellungen überhaupt erst praktisch lösbar.



# Berechnung der Bestrahlungsstärke $E(x,y)$

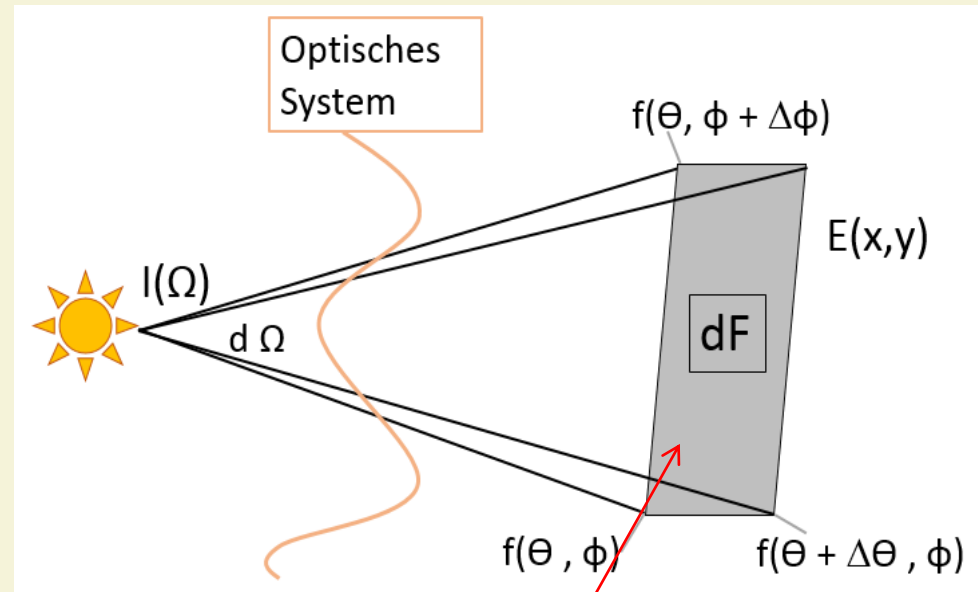
- Strahl: Funktion der Winkel  $\theta, \varphi$
- $I(\Omega)$ : Strahlstärke
- $P$ : Lichtleistung der Quelle
- $E$ : Bestrahlungsstärke auf Detektor

$$E = dP / dF$$

- $dP = I(\Omega) d\Omega$
- $dF = JF d\theta d\varphi$
- $JF = \text{Jakobideterminante}$

$$E = I(\Omega) \frac{\sin \theta}{JF}$$

**Ergebnis exakt im sequentiellen Fall!**



Strahlenbündel leuchtet Parallelogramm in Detektorebene aus!

Hembach **Photonik**

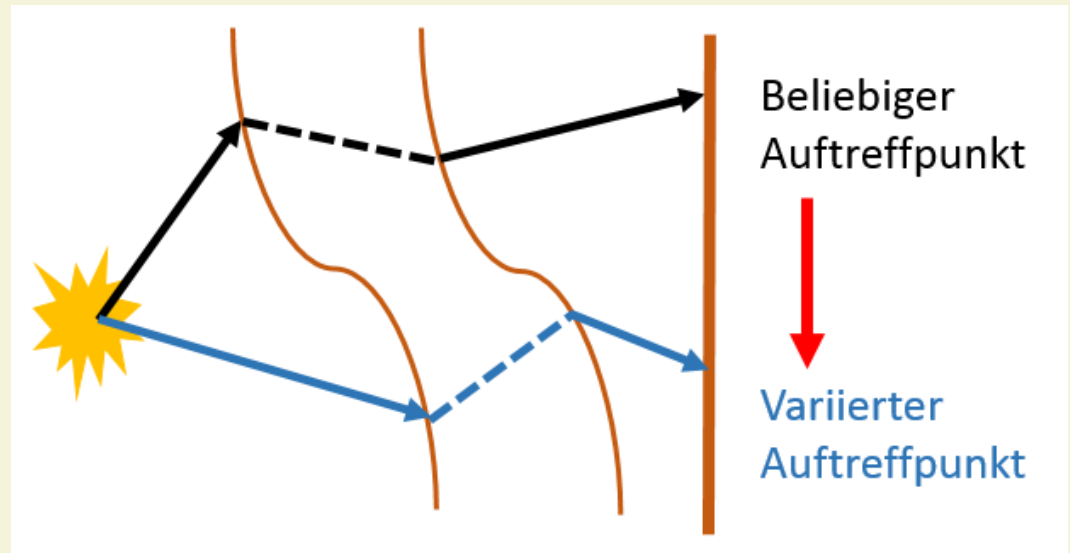
# Strahlvariation

- Strahl trifft auf irgendeinen Ort auf Detektor

• Variation des Strahls zu *gewünschtem* Auftreffpunkt

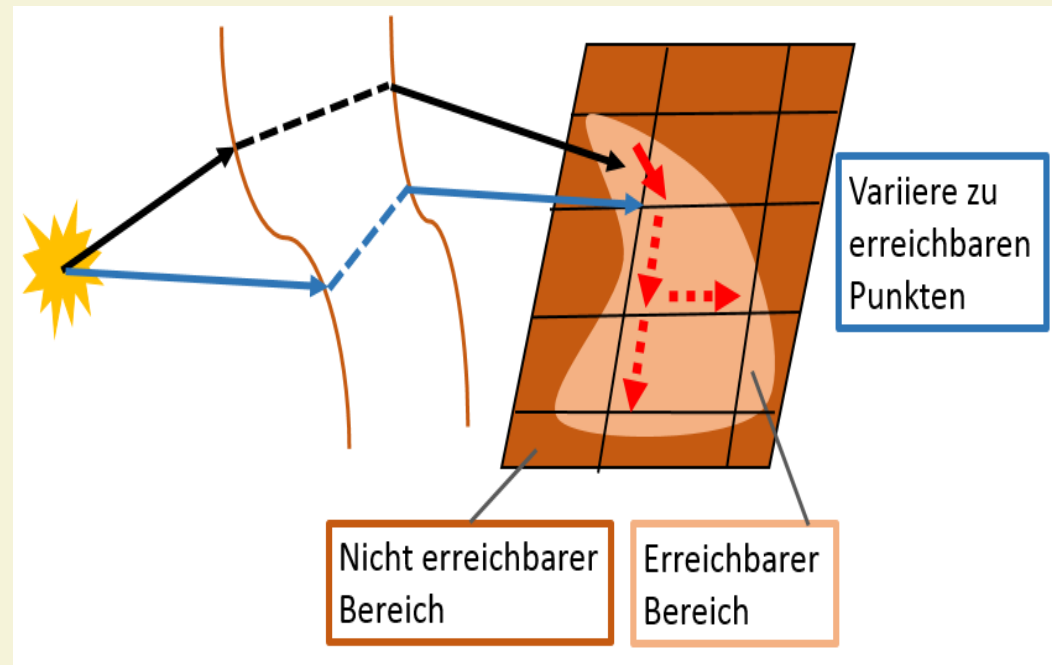
- Äquivalenzklassen für Strahlen (Richtung, Position, Hits...)

• Pro Äquivalenzklasse nur ein Strahl nötig



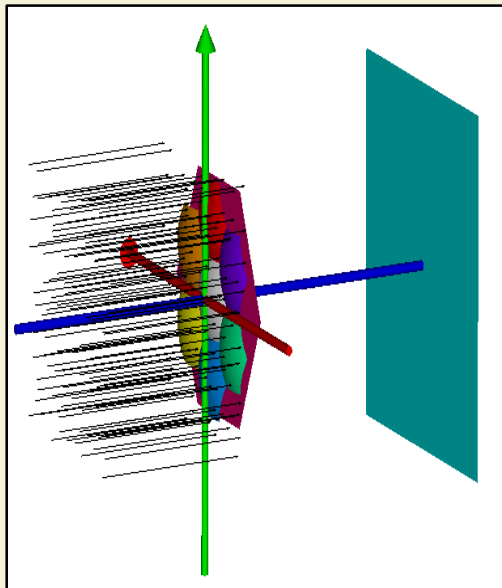
# Vorteile des differenziellen Raytracings mit Strahlvariation

- Signifikante Reduktion der benötigten Strahlzahl
- Alle erreichbaren Punkte am Detektor werden berücksichtigt
- Exakte Berechnung der Bestrahlungsstärke an allen Punkten

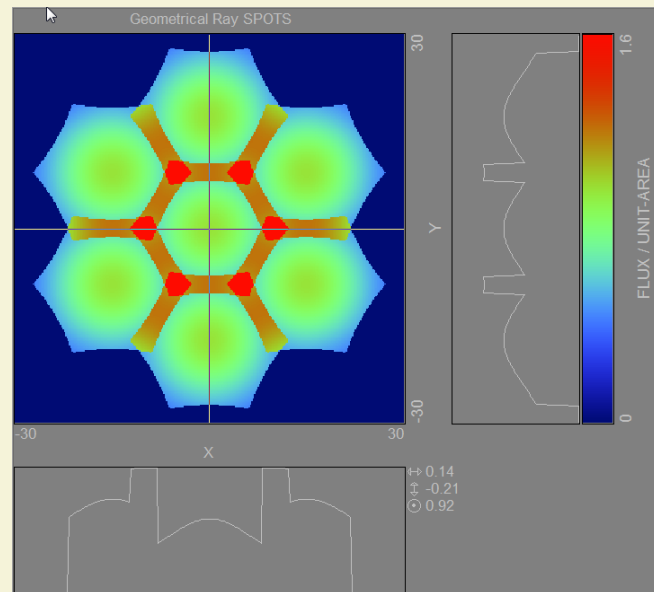


# Anwendungsbeispiel

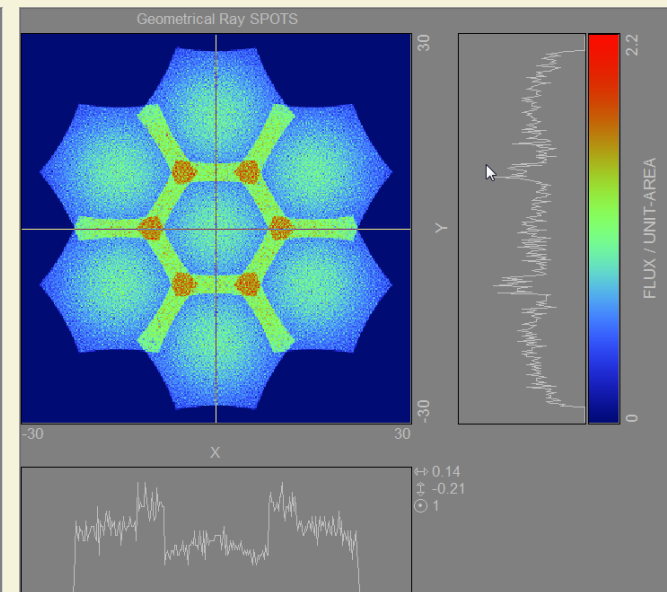
Berechnung der Bestrahlungsstärke für hexagonales Linsenarray  
- hier müssen mehrere Strahlenpfade berücksichtigt werden! -



Systemaufbau



Ergebnis differenzielles  
Raytracing:  
**exakt**



Ergebnis ASAP

# Zusammenfassung differenzielles Raytracing/Bemerkungen

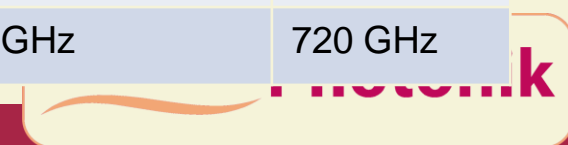
- Exakte Radiometrie für Punktquellen und kollimierte Quellen
- Je nach System oft nur wenige Startstrahlen nötig; der Rest wird durch Strahlvariation erledigt
- Anwendung für ausgedehnte Lichtquellen möglich; Algorithmus dann langsamer, aber immer noch genau.
- Großes Potential Strahlvariation – viele (noch unbekannt) Anwendungen

# Raytracing auf GPU – LambdaSpect (fx64)

- Hochparallelisierte OpenCL Implementierung von klassischen Raytracing
- Nicht sequentieller Hochleistungs-Raytracer
- Mittels FX64 LambdaSpect direkte Verbindung mit dem CAD System
- Test an simplen System (Lichtquelle, Glaskugel und Detektor):

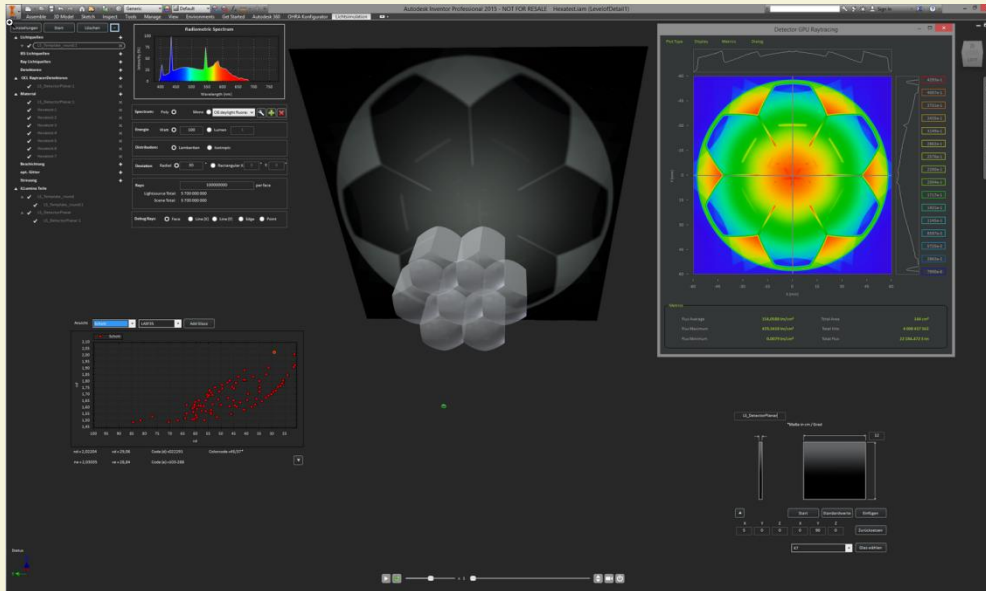
Programm	Zeit 50Mio Strahlen [s]	Strahlerzeugung [s]	Tracing [s]	Grafik [s]
ASAP	506	62	431	13
CPU - Implementierung	657	353	304	-
OpenCL GPU	~3	~0	~1	~2

Programm	Hardware	Anzahl Kerne	Kerne Frequenz	Gesamt
ASAP	CPU i7 4770	4	3,5 GHz	14 GHz
CPU - Implementierung	CPU Core2Duo	2	3,3 GHz	6,6 GHz
OpenCL GPU	nVidia GTX570	480	1,5 GHz	720 GHz





# LambdaSpect (fx64)



- Dramatische Erhöhung der Raytracing-Performance durch GPU Raytracing
- Differenzielles Raytracing zur weiteren Erhöhung der Geschwindigkeit und Genauigkeit

- Bisher: CPU-basiert
- Direkt implementiert in Autodesk Inventor
- Verbindung zwischen OpenCL Kernel und CAD
- Ab März 2015 Standalone Produkt bestehend aus LambdaSpect, Autodesk Technologie, und OpenCL GPU Kernel;  
Betaversion ab Januar 2015
- GPU Raytracing in Zusammenarbeit mit Hembach Photonik
- In Arbeit: Integration von differenziellem Raytracing
- Gefördert durch AiF GmbH (Kooperationsprojekt ZIM-Förderung)

# Vielen Dank für Ihre Aufmerksamkeit!

## *Kontakt:*

Hembach Photonik GmbH  
Dr. Bernhard Michel  
Finkenstr. 1-3  
91126 Rednitzhembach  
Tel: 0 91 22 – 88 99 49 1  
[bm@hembach-photonik.de](mailto:bm@hembach-photonik.de)