Scientific Ray-Tracing with OptiX
#23136

Timo Stich, Niklas Mevenkamp (CRT-CA)

14.10.2017
### Agenda

<table>
<thead>
<tr>
<th></th>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Motivation</td>
</tr>
<tr>
<td>3</td>
<td>Simulation</td>
</tr>
<tr>
<td>4</td>
<td>Using OptiX</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
</tr>
</tbody>
</table>
## Agenda

1. Introduction
2. Motivation
3. Simulation
4. Using OptiX
5. Results
**corporate algorithms**. **locations**

Jena, Oberkochen, Kaiserslautern & Munich

**Jena**
13 co-workers

**Oberkochen**
15 co-workers

**Kaiserslautern**
3 co-workers

**Munich**
2 co-workers (hiring)
The equation
4 strategic directions = 4 platform topics / the bottom-up perspective

\[ y = \hat{A}x + \nu \]

- Virtual prototyping
- Computational imaging
- Machine learning
- Streaming, hardware & software implementations

Best in class full system simulation
Revolutionary digital reconstruction
Pioneering role in learning algorithms
Tailored ways to efficiently run algorithms
## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Motivation</td>
</tr>
<tr>
<td>3</td>
<td>Simulation</td>
</tr>
<tr>
<td>4</td>
<td>Using OptiX</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
</tr>
</tbody>
</table>
Scanning Electron Microscope
ZEISS MultiSEM

• Scanning Electron Microscope (SEM) yields high-resolution images of solid materials (< 10 nm)

• Up to 91 electron beams working in parallel with unprecedented imaging speed

• QUESTION:
  Can we simulate the scattering of electrons close to real-time to push the resolution even further?

• -> NVIDIA OptiX for Scientific Ray-Tracing!
SEM image formation process

single electron beam x dwell time = one pixel intensity

- relevant scattering effects:
  - elastic scattering
    - no energy loss
    - change in direction
  - inelastic scattering
    - energy loss
    - change in direction
    - atom ionization
      - SE generation (~0-50eV)
      - Auger electrons (AE)
      - characteristic X-rays

---

ZEISS, Timo Stich, CRT-CA
SEM image formation process

image acquisition:

- beam is scanned across the sample
- measurement repeated at each position
Physical model

- Assumption: only **binary electron – atom collisions** are relevant

- Model: conservation of particles $\rightarrow$ Boltzmann equation

$$\begin{align*}
\Omega \cdot \nabla \psi(x, \epsilon, \Omega) &= N_V(x) \int_{\epsilon}^{\infty} \int_{S^2} \sigma_{\text{in}}(x, \epsilon', \epsilon, \Omega' \cdot \Omega) \psi(x, \epsilon', \Omega') \, d\Omega' \, d\epsilon' \\
&+ N_V(x) \int_{S^2} \sigma_{\text{el}}(x, \epsilon, \Omega' \cdot \Omega) \psi(x, \epsilon, \Omega') \, d\Omega' \\
&- N_V(x) \sigma_{\text{in}}^{\text{tot}}(x, \epsilon) \psi(x, \epsilon, \Omega) \\
&- N_V(x) \sigma_{\text{el}}^{\text{tot}}(x, \epsilon) \psi(x, \epsilon, \Omega).
\end{align*}$$

- $\mathcal{X}$ = position
- $\epsilon$ = energy
- $\Omega$ = direction

$\psi(x, \epsilon, \Omega) = \text{density of electrons (per phase space el.)}$

$N_V(x) = \text{number of atoms per unit volume}$

$S^2 = \text{unit sphere}$

**DCS** = Differential scattering Cross Section

**TCS** = Total scattering Cross Section

$$\begin{align*}
\sigma_{\text{in}}(x, \epsilon', \epsilon, \Omega' \cdot \Omega) &= \text{inelastic DCS} \\
\sigma_{\text{el}}(x, \epsilon, \Omega' \cdot \Omega) &= \text{elastic DCS} \\
\sigma_{\text{in}}^{\text{tot}}(x, \epsilon) &= \text{elastic / inelastic TCS} \\
\sigma_{\text{el}}^{\text{tot}}(x, \epsilon) &= \text{elastic} / \text{inelastic TCS}
\end{align*}$$
Forward simulation
= solving the Boltzmann equation

Three major approaches:

• Lattice Boltzmann method
  • discretization of space (→ discretization of velocities)
  • collision and transfer of electron density at macroscopic scale (histograms)
  • boundary conditions non-trivial

• Monte Carlo method
  • random sampling of electron trajectories
  • investigation of phase space at microscopic scale
  • dominant method in materials science and metrology

• Method of moments + Finite Differences
  • much faster than the above two methods
  • still object of basic research

Source: Duclous et al. ‘09
# Agenda

1. Introduction
2. Motivation
3. Simulation
4. Using OptiX
5. Results
Software
Existing 3D Monte Carlo simulations for electrons

• **DTSA-II:**
  • PE / SE: CSDA / not available
  • geometry: intersections of half-spaces in 3D
  • API: Java / Jython scripting
  • license: public domain (binaries online, source code received from external contact)
  • language: Java
  • www: http://www.cstl.nist.gov/div837/837.02/epq/dtsa2/

• **CASINO:**
  • PE / SE: detailed / detailed
  • API: very limited scripting support, mostly GUI based
  • license: unknown
  • language: unknown
  • www: http://www.gel.usherbrooke.ca/casino/index.html

• **PENELOPE:**
  • PE / SE: detailed / not available
  • aimed at high energy particles ($\epsilon > 1\text{MeV}$)
  • www: [https://www.oecd-nea.org/tools/abstract/detail/nea-1525](https://www.oecd-nea.org/tools/abstract/detail/nea-1525)

• **JMonsel:**
  • aimed specifically at SE simulation
  • available as an extension for DTSA-II (source received from external contact)
  • no MWE available
Monte Carlo simulation of electrons
Continuous slowing-down approximation (CSDA)

• Assumptions:
  • changes in flight direction are mainly caused by elastic interactions
  • energy loss can be approximated as a continuous process

• Sampling trajectories:
  • move electrons in a straight line between elastic collisions
  • change direction according to elastic DCS
  • energy loss between elastic collisions based on energy loss rate

• Secondary electron yield:
  • modeled as a continuous process
  • energy is transferred from PE to SEs
  • SE yield reduced by absorption of SE within material (depth vs. free path)
Secondary electron yield

Option 1: simulate SE in the same fashion as PE

- cannot be done in CSDA, i.e. all inelastic collisions have to be simulated in detail
- inelastic interactions of SE are more complicated than those of PE
  - plasmon excitation
  - atom bonding
  - charging
- significantly increases computational cost

Option 2: approximate SE yield with analytical function

- ODE: \( I_{SE}(0) = 0, \quad \frac{dI_{SE}}{ds} = \frac{|S(\epsilon(s))|}{\epsilon_s} \exp\left(-\frac{\text{depth}(s)}{\lambda_s}\right) \)

  \(\epsilon_s = \text{effective SE generation energy}\)

  \(\lambda_s = \text{effective escape depth}\)

- forward Euler discretization: \( I_{SE}(s) \approx \frac{s|S(\epsilon^0)|}{\epsilon_s} \exp\left(-\frac{\text{depth}(0)}{\lambda_s}\right) \)

- fast and accurate for simple geometries (e.g. edge geometry with homogenous material)
- likely accurate enough for edge geometries with a few layers
- questionable whether accurate enough for complex geometries
Monte Carlo simulation of electrons
CSDA algorithm

Initialization: \[ x_0 = d_b \sqrt{-2 \log \xi} \ast (\cos \phi, \sin \phi, 1)^T, \phi = 2\pi \xi, \ \epsilon_{0.5} = E_b, \ \Omega_{0.5} = (0,0,-1)^T \]

Path tracing
or moving electrons from \((x_n, \epsilon_{n+0.5}, \Omega_{n+0.5})\) to \((x_{n+1}, \epsilon_{n+1.5}, \Omega_{n+1.5})\) in phase space

1. sample free path (disregarding surrounding geometry): \[ s_{\text{bulk}} = -\lambda_e (x_n, \epsilon_{n+0.5}) \ast \ln(\xi) \]
2. compute closest hit \(p\) with surrounding geometry (assuming infinite travel)
   1. if \(|p - x_n| < s_{\text{bulk}}\), set \(s = |p - x_n|\)
   2. else set \(s = s_{\text{bulk}}\)
3. update position: \[ x_{n+1} = x_n + S \ast \Omega_{n+0.5} \]
4. compute energy loss: \[ W = |s \ast S(x_n, \epsilon_{n+0.5})| \]
5. update energy: \[ \epsilon_{n+1.5} = \epsilon_{n+0.5} - W \]
6. accumulate SE yield: \[ I_{\text{SE}} = I_{\text{SE}} + \frac{W}{\epsilon_s} \ast \exp\left(-\frac{\text{depth}(x_n)}{\lambda_s}\right) \]
7. if \(s = s_{\text{bulk}}\) (i.e. no material interface was crossed)
   1. sample azimuthal scattering angle: \(\phi = 2 \pi \xi\)
   2. sample polar scattering angle: \(\theta = f(x_{n+1}, \epsilon_{n+1.5}, \xi)\)
   3. update direction: \(\Omega_{n+1.5} = \text{rotation}(\phi, \theta) \ast \Omega_{n+0.5}\)

use tabulated \(f\) or perform inverse transform sampling on DCS

Nvidia OptiX
use RNG to produce \(\xi \in U(0,1)\)
## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Motivation</td>
</tr>
<tr>
<td>3</td>
<td>Simulation</td>
</tr>
<tr>
<td>4</td>
<td>Using OptiX</td>
</tr>
<tr>
<td>5</td>
<td>Results</td>
</tr>
</tbody>
</table>
Engine structure
Objects & functions

- parallelization of multiple initial rays
- free lunch

350 million rays / second

- implement **desired result** of hit event
- hit event **emission** is free lunch

- to be implemented by developer

- implement desired result of hit event
- hit event emission is free lunch
Engine structure
Execution pipeline

Launch
- rtContextLaunch
- Ray Generation Program
- Exception Program

Traverse
- Node Graph Traversal
- Selector Visit Program
- Acceleration Traversal
- Intersection Program

Shade
- Miss Program
- Closest Hit Program
- Any Hit Program
Engine structure
Execution pipeline

Launch
- rtContextLaunch
  - Ray Generation Program
  - Exception Program

rtTrace

Traverse
- Node Graph Traversal
- Selector Visit Program
- Acceleration Traversal
- Intersection Program
- Any Hit Program

Evaluation of intersections
- Shade
- Miss Program
- Closest Hit Program
- Any Hit Program

Material properties
- Reflection
- Transmission
- Absorption
- Ray splitting
Closest-Hit Program
- Stores the Material Idx
- Trace in new direction
OptiX - Ray Program

Volume Scattering
- Trace without hit
- Update Energies
- Update Direction
Handling of Co-Planar Surfaces
- OptiX cannot distinguish between co-planar materials
- Solution: Trace a second ray type which ignores leaving rays in hit program (Normal * Ray > 0 )
## Agenda

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Introduction</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Motivation</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Simulation</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Using OptiX</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Results</td>
</tr>
</tbody>
</table>
Benchmark: SEMC & SEMtiX vs. DTSA-II
Electron trajectories

Silicon

Aluminum oxide

SEMC

100 random trajectories for Si @ 1keV (SEMC)

SEMC

100 random trajectories for Al2O3 @ 1keV (SEMC)

DTSA-II

100 random trajectories for Si @ 1keV (DTSAII)

DTSA-II

100 random trajectories for Al2O3 @ 1keV (DTSAII)

SEMiX

100 random trajectories for Si @ 1keV (SEMiX)

SEMiX

100 random trajectories for Al2O3 @ 1keV (SEMiX)

\[ \lambda \]

0.23nm @ 50eV
to
1.45nm @ 1keV

0.16nm @ 50eV
to
1.03nm @ 1keV
Benchmark: SEMC & SEMtiX vs. DTSA-II
BSE line profiles of different materials

Si @ 1keV line profile (w=10nm, h=10nm)

Al2O3 @ 1keV line profile (w=10nm, h=10nm)
Benchmark: SEMC & SEMtix vs. DTSA-II

BSE with varying beam energy

Si BSE with varying beam energy

Al2O3 BSE with varying beam energy
Benchmark: SEMC & SEMtiX vs. DTSA-II
BSE with varying line width

Si @ 1keV (varying line width)

Al2O3 @ 1keV (varying line width)
Benchmark: SE yield vs. published figures

**Aluminum oxide**

**Silicon dioxide**

**PMMA**

Dapor 2011

Results

SEM scans of 10nm and 50nm line profiles

Silicon line profile (w=10nm, h=10nm) @ 1keV

width = 10 nm

 Silicon

Aluminum oxide line profile (w=10nm, h=50nm) @ 1keV

width = 50 nm

Aluminum oxide
Results
SEM scans of a 25nm line profile with varying beam diameters (1-50nm)

General comment on running times:

- **SEMC** (Matlab): 40-70 seconds / 10,000 electrons, 1.1-1.7 hours / line scan (101 points)
- **DTSA-II** (Java): 8 seconds / 10,000 electrons, 0.24 hours / line scan (101 points)
- **SEMtiX** (CUDA / OptiX): 1 second / 1,700,000 electrons, 1 minute / image (101 x 101 points)
Results
SEM images simulated using SEMtiX

PMMA single line
(x,y,z)=(50nm, 1um, 50nm)
1 keV x 5nm
101 x 101 px (1px = 1nm x 1nm)
10000 electrons
55 seconds

10.000 electrons / px
101 x 101 px / image
60s / image

⇒ 1.7 million electrons / s

Silicon dioxide (SiO2) two lines
(x, y, z, dx)=(25nm, 1um, 50nm, 10nm)
1 keV x 1nm
101 x 101 px (1px = 1nm x 1nm)
1000 electrons
11 seconds

PMMA line on Aluminum oxide
(x, y, z)=(50nm, 1um/75nm, 25nm)
1 keV x 1nm
101 x 101 px (1px = 1nm x 1nm)
1000 electrons
10 seconds

y=1um
y=75nm
What to know more? Let us know!