Technische Universität München

Audi
Vorsprung durch Technik

Real-time Electromagnetic Wave Propagation using OptiX for Simulation of Car-to-Car-Communication

NVIDIA GTC 2014, San Jose

Manuel Schiller [1]

Andreas Kern [2]

Alois Knoll [1]

[1] Robotics and Embedded Systems
Department of Informatics
Technische Universität München

[2] Audi Electronics Venture GmbH

Agenda

- Motivation and Goals
- Simulation of Radio Wave Propagation
- Implementation in OptiX and CUDA
- Optimization of Ray Tracing Performance
- Outlook

Audi
Vorsprung durch Technik

Motivation

- Vehicular Adhoc Networks (VANETs) will improve traffic safety, driving comfort and efficiency
- Advanced Driver Assistance Systems (ADAS) based on VANETs need to be tested thoroughly before deployment
- Real world testbeds
  - Are costly
  - Are not yet available
  - Do not deliver reproducible results
- Virtual test drives are already used to validate ADAS (see GTC 2012) and shall be enhanced for VANET simulation

Audi
Vorsprung durch Technik

Signal Propagation in VANETs

- Signal propagation has **major impact** on ADAS **performance**
- Various wireless channel models exist:
  - Simple models (e.g. free space propagation)
  - Statistical models
  - Deterministic models
- Only deterministic models allow for a **site** and **situation specific** simulation of the wireless communication channel

Audi
Vorsprung durch Technik

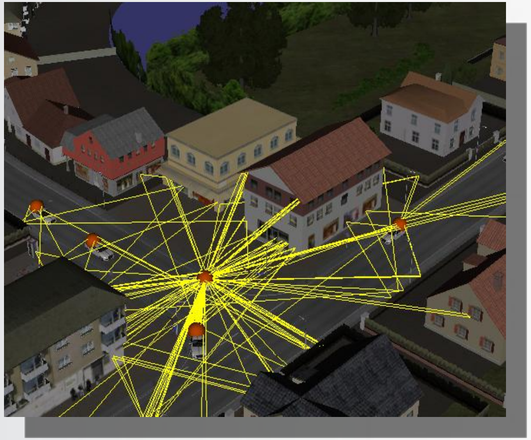
Requirements and Goals

Simulate the wireless communication channel

- For **dynamic** and **detailed** scenes,
- Highly **accurate,**
- In **real time** (necessary for hardware-in-the-loop simulation)

in order to allow realistic testing of ADAS based on VANETs in virtual reality.

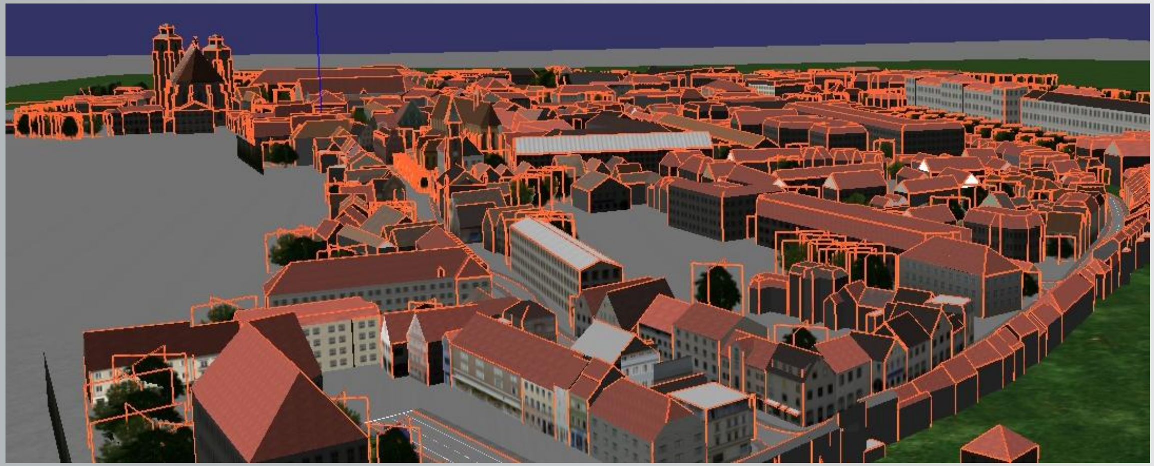
Radio Wave Propagation using Ray Tracing

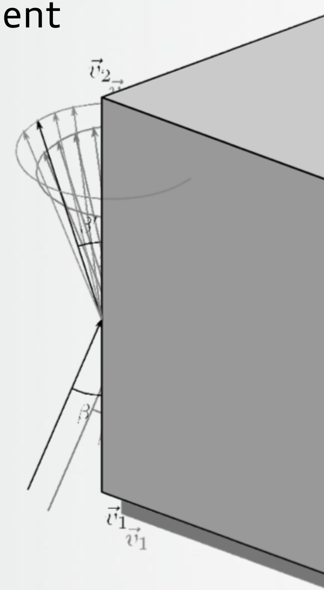


- Radio waves can be modeled as rays using
  - **G**eometrical **O**ptics and
  - the **U**nified **T**heory of **D**iffraction
- Ray tracing can be applied to find propagation paths of radio waves
- Brute-Force approach: Higher ray count = higher accuracy
- **Static** (e.g. buildings) and **dynamic** obstacles (e.g. cars) block wave / ray propagation and cause **reflection** and **diffraction** of waves
- Accurate 3D models of environment and cars are necessary

Diffraction

- Diffraction at edges (e.g. of buildings) allows signal to be received even if there is no line-of-sight
- Important propagation phenomenon at intersections especially in urban environment
- Diffraction edges are detected in an automated offline preprocessing step





Audi
Vorsprung durch Technik

Comparison with Image Rendering

- Multiple „lights“ = transmitting antennas
- Multiple „cameras“ = receiving antennas
- No image plane, rather a 360 degree field of view
- No approximations (like GI algorithms), we need the exact ray interactions for accurate calculation of amplitude and phase of the electromagnetic field
- Moving obstacles, therefore shadow regions cannot be precomputed

Audi
Vorsprung durch Technik

Implementation

- We use **NVIDIA OptiX** for GPU Raytracing to find the propagation paths between transmitter and receiver
- High quality acceleration structures enable us to simulate highly **detailed** and **dynamic** scenes (carefully selecting the appropriate ones is crucial!)
- We employ different custom geometry types:
  - triangle meshes for 3D models
  - spheres for antennas
  - cylinders for diffraction edges
- OptiX allows us to concentrate on the actual wave propagation rather than on low-level ray tracing optimization
- Some (high-level) optimizations are still necessary

Memory Management

- Every valid propagation path needs to be stored
- Worst case memory allocation for each ray: $M = N_D^{L_D} \cdot L_R \cdot M_I$

N_D	New rays at diffraction (e.g.100)
L_D	Maximum diffractions (e.g. 2)
L_R	Maximum reflections (e.g. 5)
M_I	Memory needed per interaction (e.g. 32 byte)

$$\Rightarrow M = 1.6 \text{ Megabyte}$$

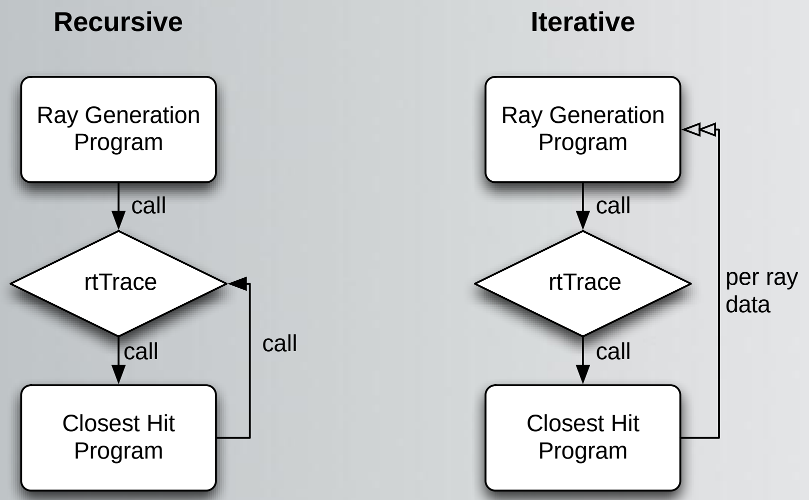
4 GB GPU memory:
Only 2500 rays, but we rather need millions!

- Dynamic memory management:
  - Allocate a global buffer for all threads
  - When a path needs to be stored, atomic operations ensure serialized buffer access

Audi
Vorsprung durch Technik

Improving Ray Tracing Performance
Recursion vs. Iteration

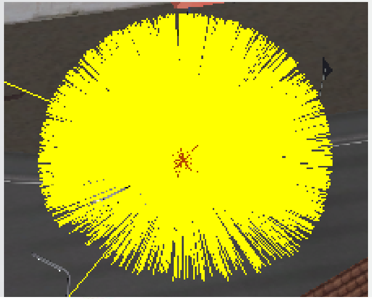
- Max. reflections / diffractions can lead to very high recursion depths
- Iterative Ray Tracing is **up to 10 % faster** than recursive approach

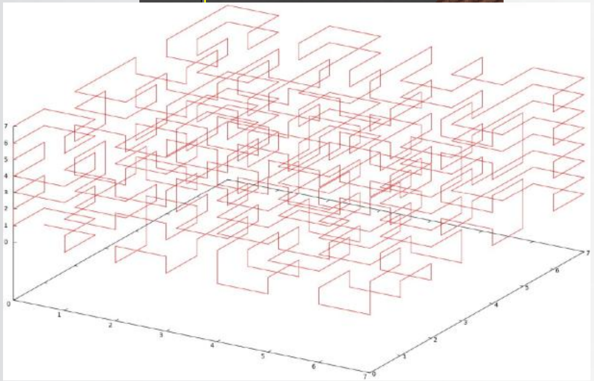


Audi Vorsprung durch Technik

Improving Ray Tracing Performance
Ray reordering

- Naive approach:
  - Sample N random directions on sphere surface, then trace them immediately
  - works, but bad ray coherence (memory access and divergence)
- Better:
  - Sort random directions before tracing using space filling curve (Hilbert curve, Z-curve)
  - Outperforms naive approach by **up to 100 %**

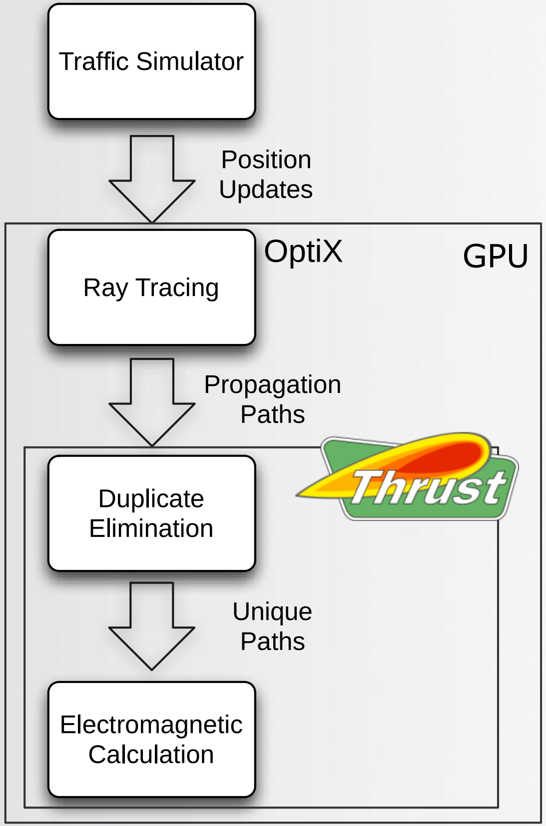




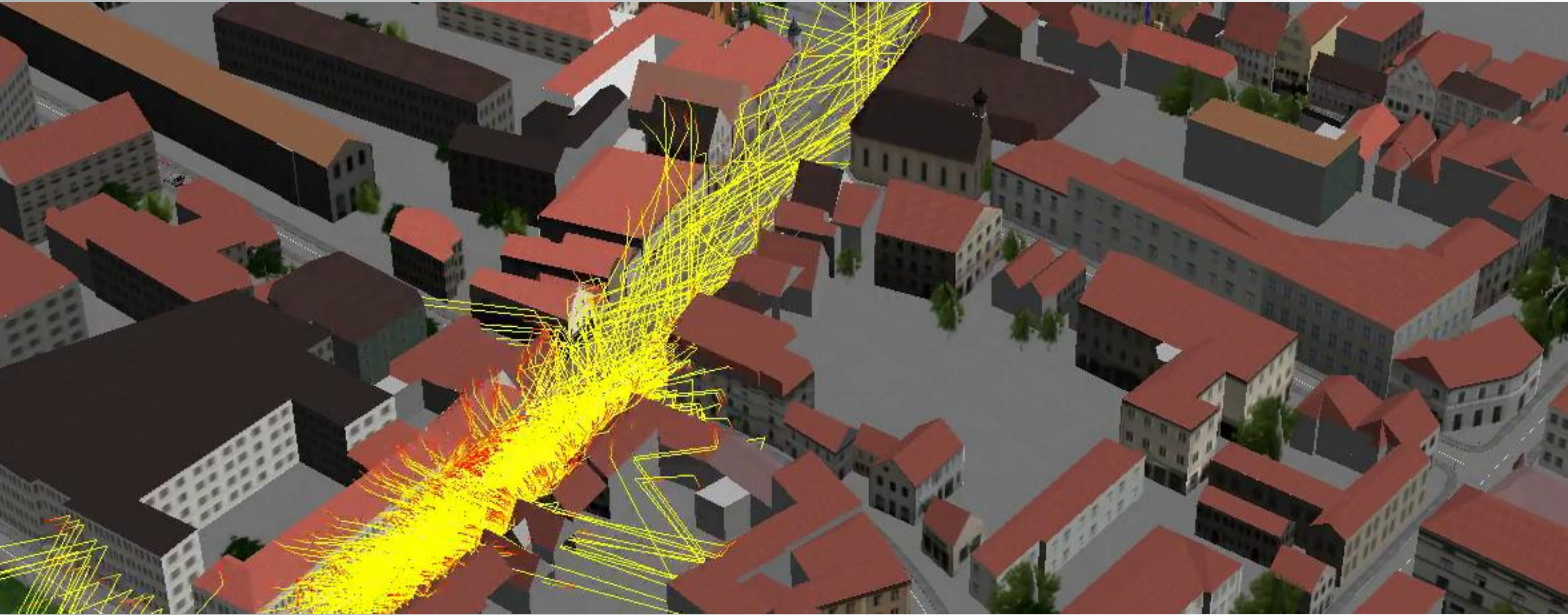
Audi
Vorsprung durch Technik

Wave Propagation Simulation Pipeline

- No on-the-fly „shading“ of rays, because only few rays arrive
- Only geometric information of propagation path is stored
- Electromagnetic field calculation is applied in a postprocessing step for each detected propagation path
- Postprocessing is also done on GPU using Thrust:
  - No memory copying needed thanks to OptiX-CUDA interop
  - Parallel iterating over propagation paths
  - Parallel reduce-by-key to sum up contribution of different propagation paths per receiving antenna



Video



Outlook

- Coupling of ray tracing results with network simulator
- Simulation of MIMO antenna systems
- Exploration of Multi-GPU performance
- Exploitation of frame coherence

Audi
Vorsprung durch Technik

Technische Universität München

Audi
Vorsprung durch Technik

Thank you very much.

Contact:

Manuel Schiller, Technische Universität München

manuel.schiller@in.tum.de