STIR and Tensorflow

Philipp Windischhofer

July 27, 2017

Table of Contents

Where to start

How to do ray tracing on a CPU?

How to do ray tracing on a GPU?

Results

The big picture

- STIR
- iterative reconstruction algorithms
- find the optimum of a *cost function*

$$P(\text{global LOR response}|\text{image}) = \prod_{\text{all LORs}} P(\text{LOR response}|\text{image})$$

$$P(\text{LOR response}|\text{image}) = \int_{\text{LOR}} \text{image density}$$

- in the following: how to compute / approximate this integral?

empty

How to do ray tracing on a CPU?

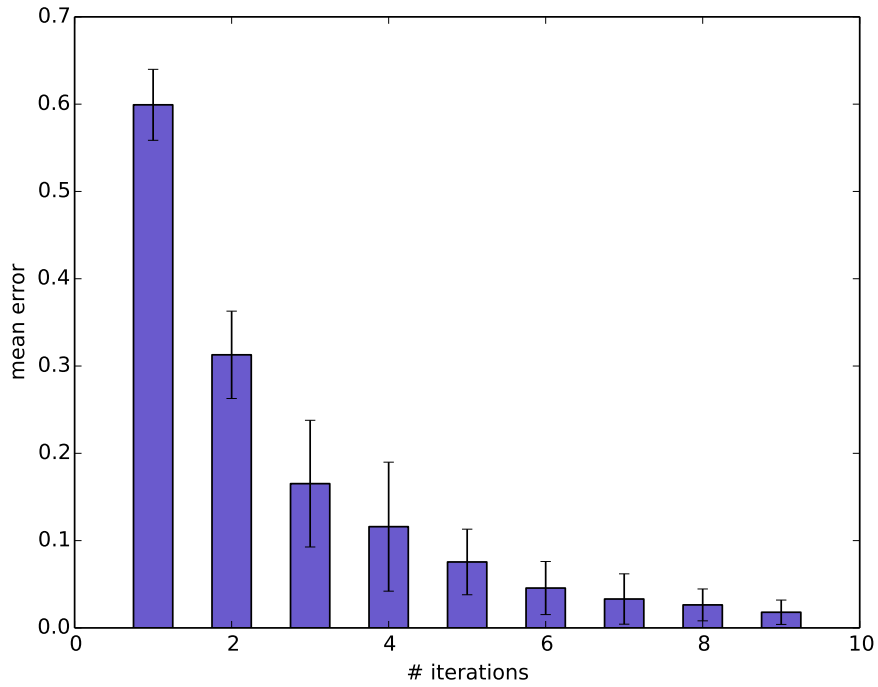
Siddon's algorithm

How to do ray tracing on a GPU?

Ray marching: an iterative algorithm

Accuracy

- assume: have "enough" points along the LOR



- with 6 iterations, are already at $\sim 5\%$ level!

Ray marching: an iterative algorithm

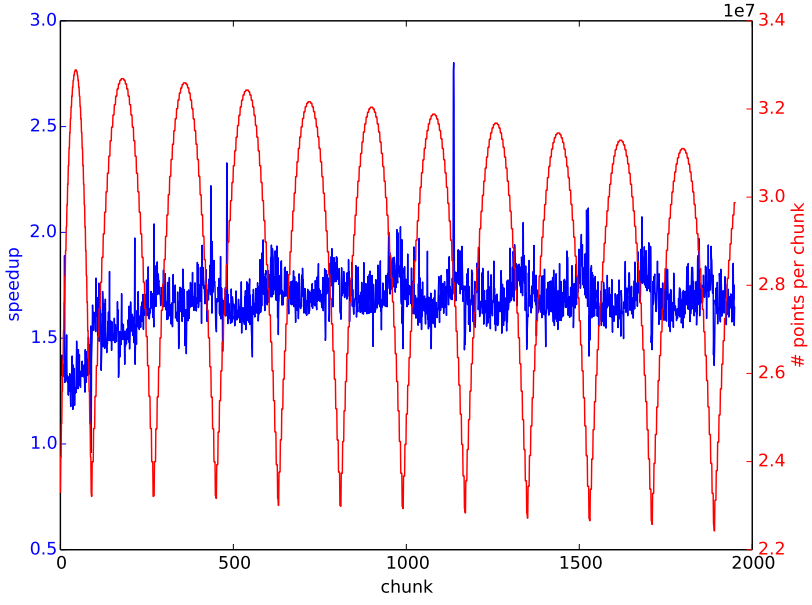
Accuracy

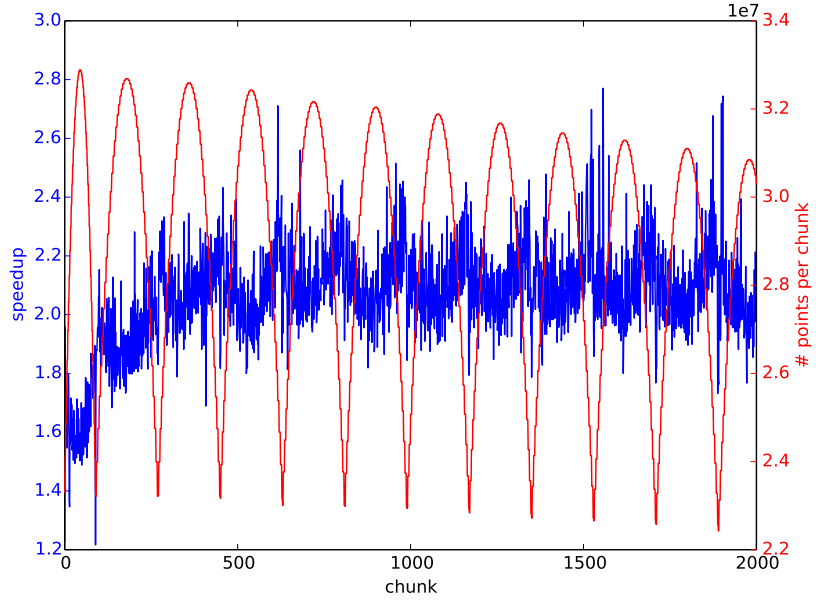
- are there ever enough points?
- does not matter for STIR!

Bringing together STIR and Tensorflow

graph generation in python and graph utilization in c++

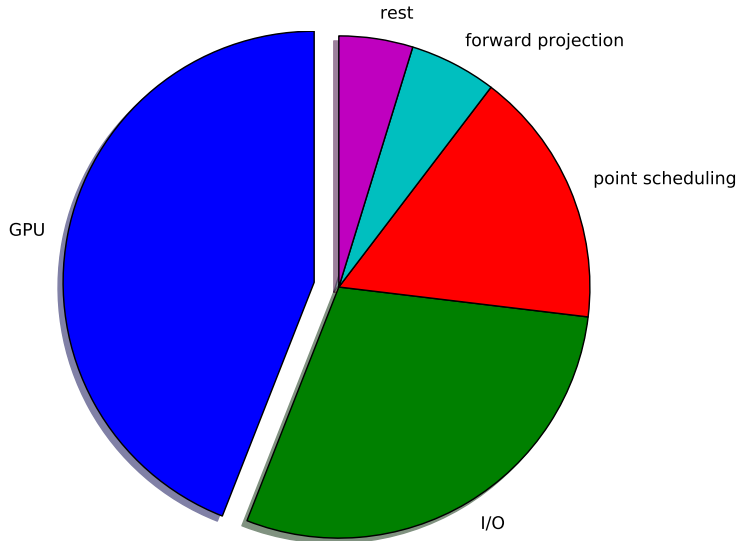
Speedup without caching

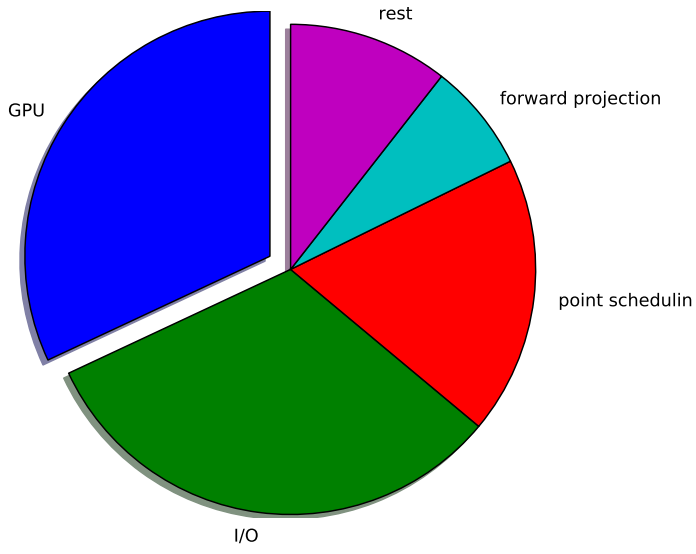




left: 6 iterations, **right**: 2 iterations

- average speedup is similar

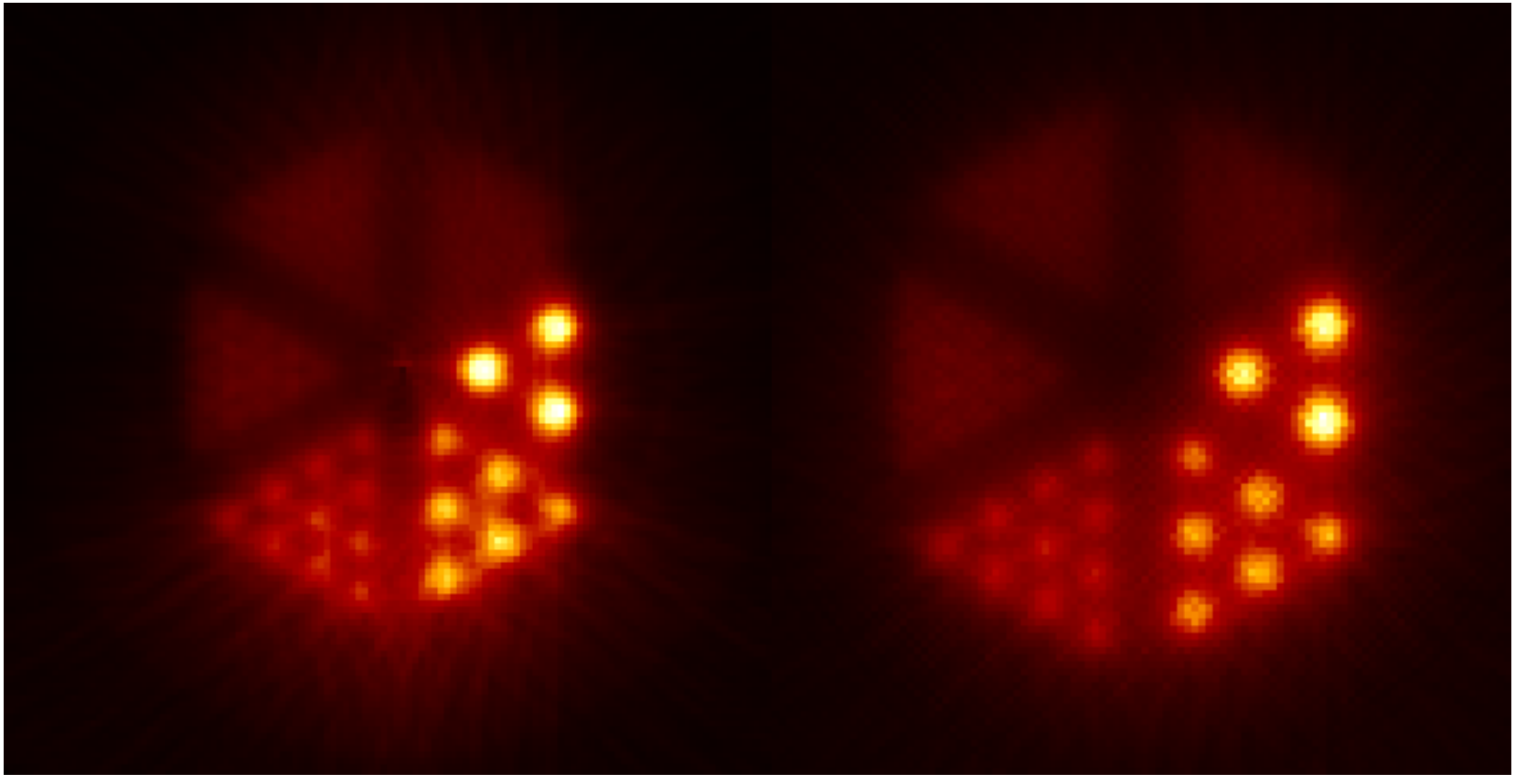




left: 6 iterations, **right**: 2 iterations

- *I/O*: converting from `ProjMatrixElemsForOneBin` to `Tensor` and back
- *point scheduling*: choose points to sample the TOR / LOR

Images



left: 2 iterations, 20LORs per matrix element, **right**: original STIR

- some artefacts: too few points

How to proceed?

- whole toolchain is in place

Where to find the code?

- STIR-TF: `https://github.com/philippwindischhofer/STIR/tree/stir-tf`
- ray tracing scripts: `https://gitlab.phys.ethz.ch/luster/tf-raytracing`

Any comments and contributions are welcome!