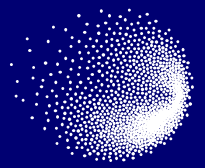


PSI

Fast GPU-friendly indexing (cell reorientation) algorithm for X-ray crystallography

Hans-Christian Stadler Kleeb (hans-christian.stadler@psi.ch)
NoBugs Conference 2024 at Grenoble

Agenda

1 Motivation
2 Paper on Algorithm, Implementations
3 Speed Potential
4 Indexing Quality
5 Algorithm
6 Questions

Motivation

New Jungfrau Detector 16M@2kHz

- High data rate – data reduction desirable.
- Fast feedback on measurement data – can a frame be indexed?
- Potential for data reduction - only save indexable frames.
- Potential for data reduction – only save relevant image areas.
- Just speed up indexing.

Project REDML

- Piero Gasparotto[1], Luis Barba[2], HC Stadler[1], Greta Assman[1], Henrique Mendonça[3], Alun Ashton[1], Markus Janousch[1], Filip Leonarski[1], Benjamín Béjar[2]
- PSI[1], SDSC[2], CSCS[3]

Paper/Algo/Implementations

Toro (paper/algo/implementation)

- Algorithm
- Pytorch based implementation in Python (Luis Barba)
- Quality and speed measurements
- https://journals.iucr.org/j/issues/2024/04/00/jo5098/index.html
- https://renkulab.io/projects/lfbarba/toro-indexer-for-serial-crystallography

Fast feedback indexer (algo/implementation)

- Same algorithm except for details
- CUDA based implementation in C++ (HC Stadler)
- Higher speed for online (frame by frame) processing
- https://github.com/paulscherrerinstitute/fast-feedback-indexer

Speed Potential

On my laptop:

```
$ ./simple-data-bulk-indexer --rep=1000 --quiet --method=ifssr --maxspots=150
--cells=16 --cands=24 --hsp=$((25*1024)) --triml=.01 --trimh=.3 --delta=.1 --
minpts=8 --contr=.8 --iter=8 -ths=7 --rblks=3 --ipg=5 --reducalc
../../data/simple/files/image*_local.txt

files: 10, operations: 10000
per operation average timings:
   clock time: 0.000456699s
 reading time: 4.59597e-07s
   index time: 0.00226413s
  refine time: 0.000205426s
```

→ 0.46ms per frame (10 frames indexed 1000 times) for this run
depends on parameters / HW / warmup / disturbances …

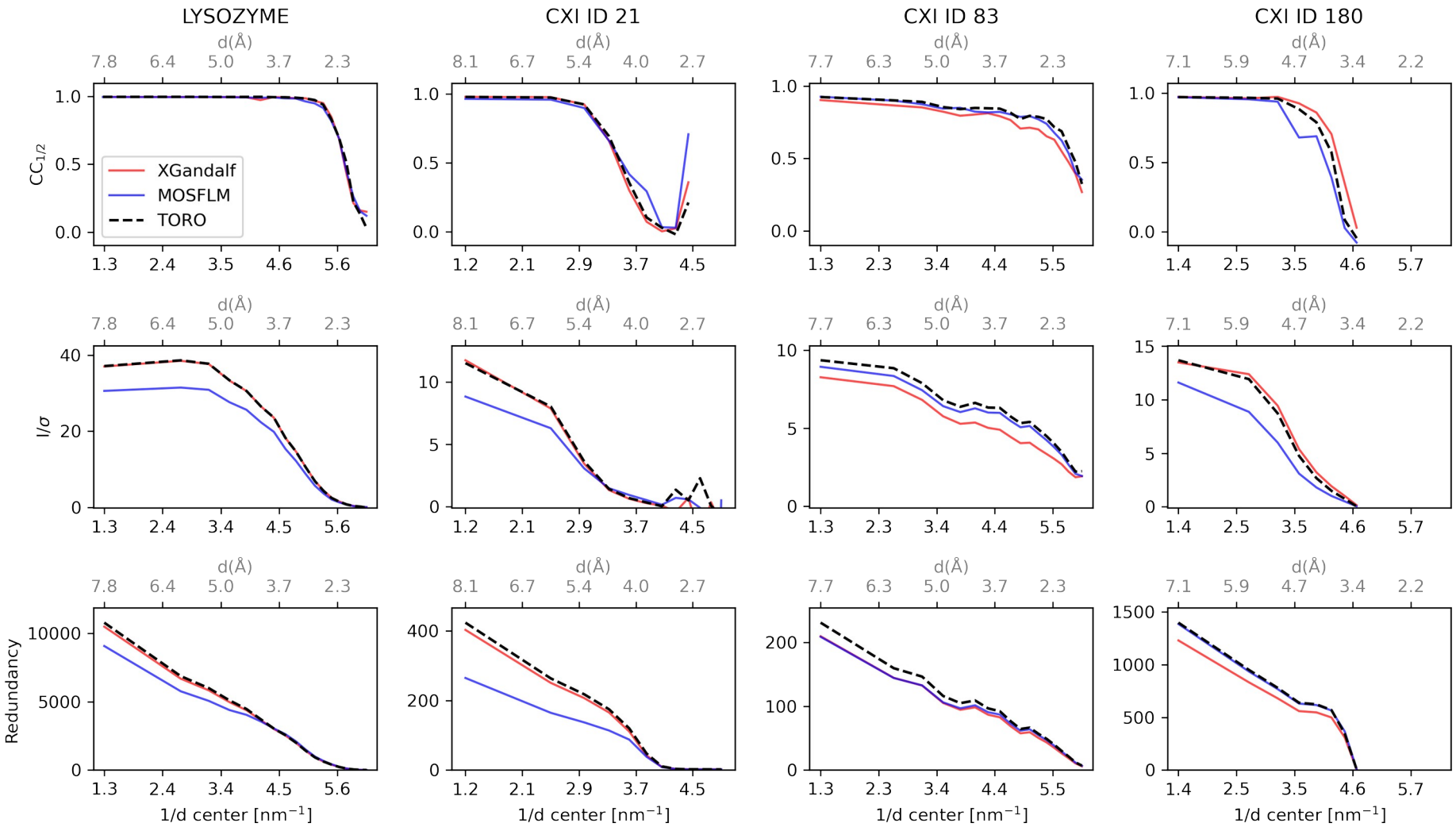
→ try it yourself: examples/cpp-simple-data-bulk-indexer

Pipeline: read → index start → index end → refine (→ index start)

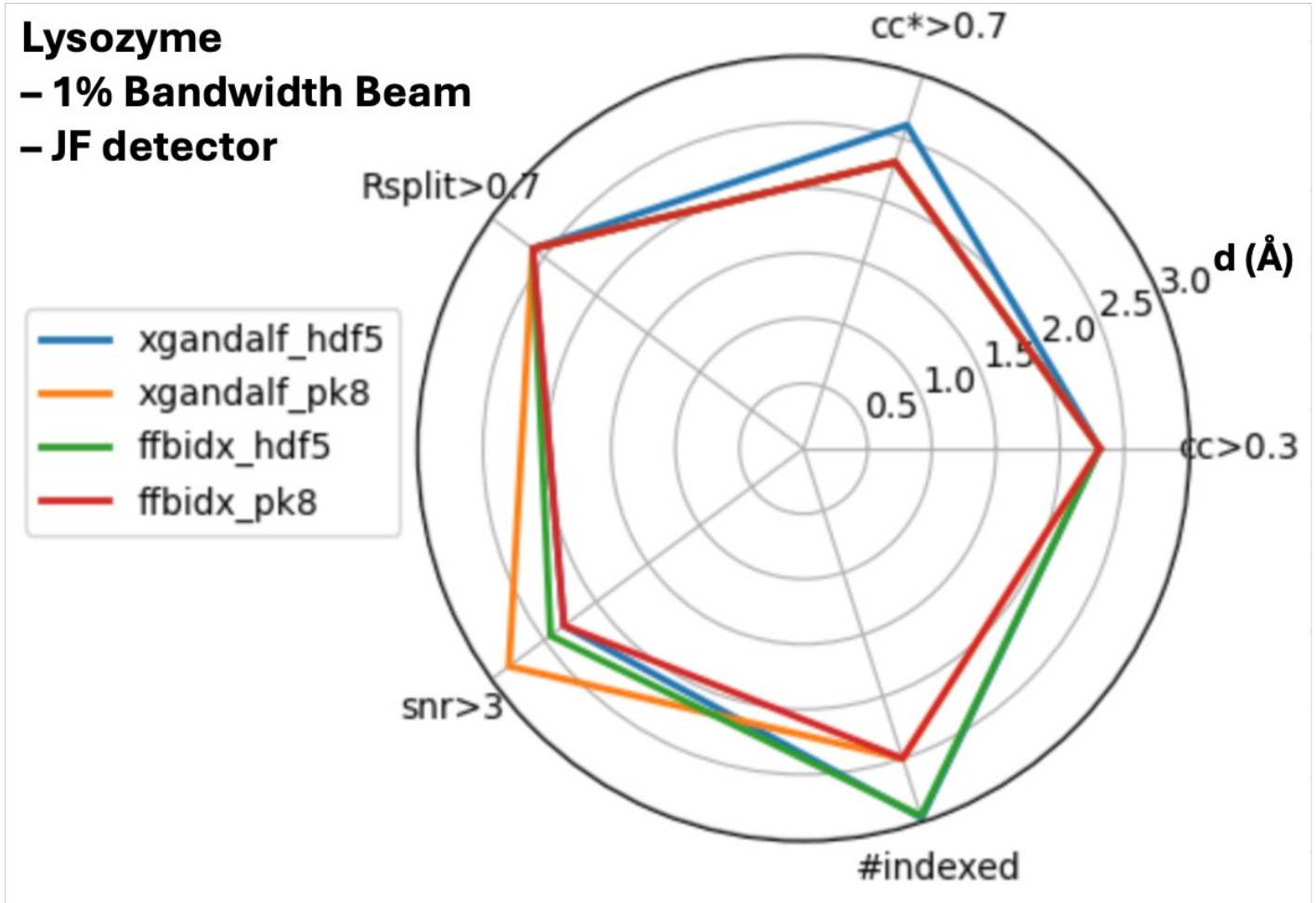
Timings are for input in memory → output in memory

Indexing Quality (Toro)

(see paper)

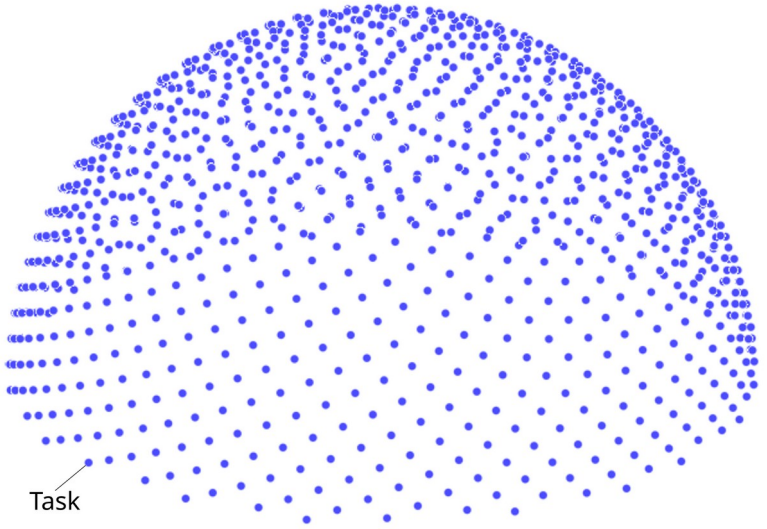


Indexing Quality (Fast Feedback Indexer)



Unpublished, measurements and visualization by Duan Jiaxin, PSI

Algorithm



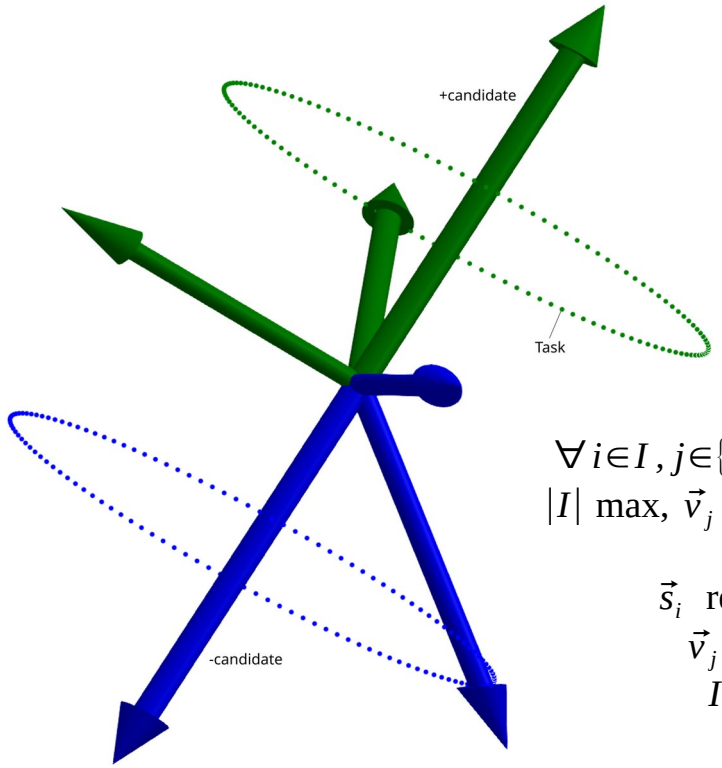
1 – Candidate vector sampling
2 – Refinement of best candidates
3 – Candidate cell sampling
4 – Refinement of best candidates
5 – Select solution

Iterative refinement by least squares fitting to spots that are considered inliers with candidate induced Miller indices.

→ Sampling is GPU friendly (trivially parallel)

Best candidate collection and refinement is not very GPU friendly (synchronization, low number)

→ 1-3 on GPU, 4+5 on CPU



$\forall i \in I, j \in \{1,2,3\} : \vec{v}_j \cdot \vec{s}_i$ is ~int
$|I|$ max, $\vec{v}_j$ with given geometry

$\vec{s}_i$ reciprocal spot
$\vec{v}_j$ cell vector
I inlier set