CWI

Universiteit Antwerpen

The ASTRA Tomography Toolbox

5 April 2016

Introduction

CWI

Universiteit Antwerpen

Today

- **Introduction to ASTRA**
- Exercises
- More on ASTRA usage
- Exercises
- Extra topics
- Hands-on, questions, discussion

About Me

Willem Jan Palenstijn

Researcher / postdoc

Computational Imaging group

Center for Mathematics & Computer Science (CWI), Amsterdam

Lead developer on ASTRA Toolbox

History

- 2007: Toolbox started at U. Antwerp (Vision Lab)
  - Initial goal: reduced implementation work for internal PhD projects
- 2012: First open source release
- 2014: Now developed jointly by CWI and Vision Lab
- Dec 2015: Release 1.7

CWI

Universiteit Antwerpen

What, exactly, is ASTRA?

ASTRA is a modular toolbox for tomographic reconstruction, with a focus on research.

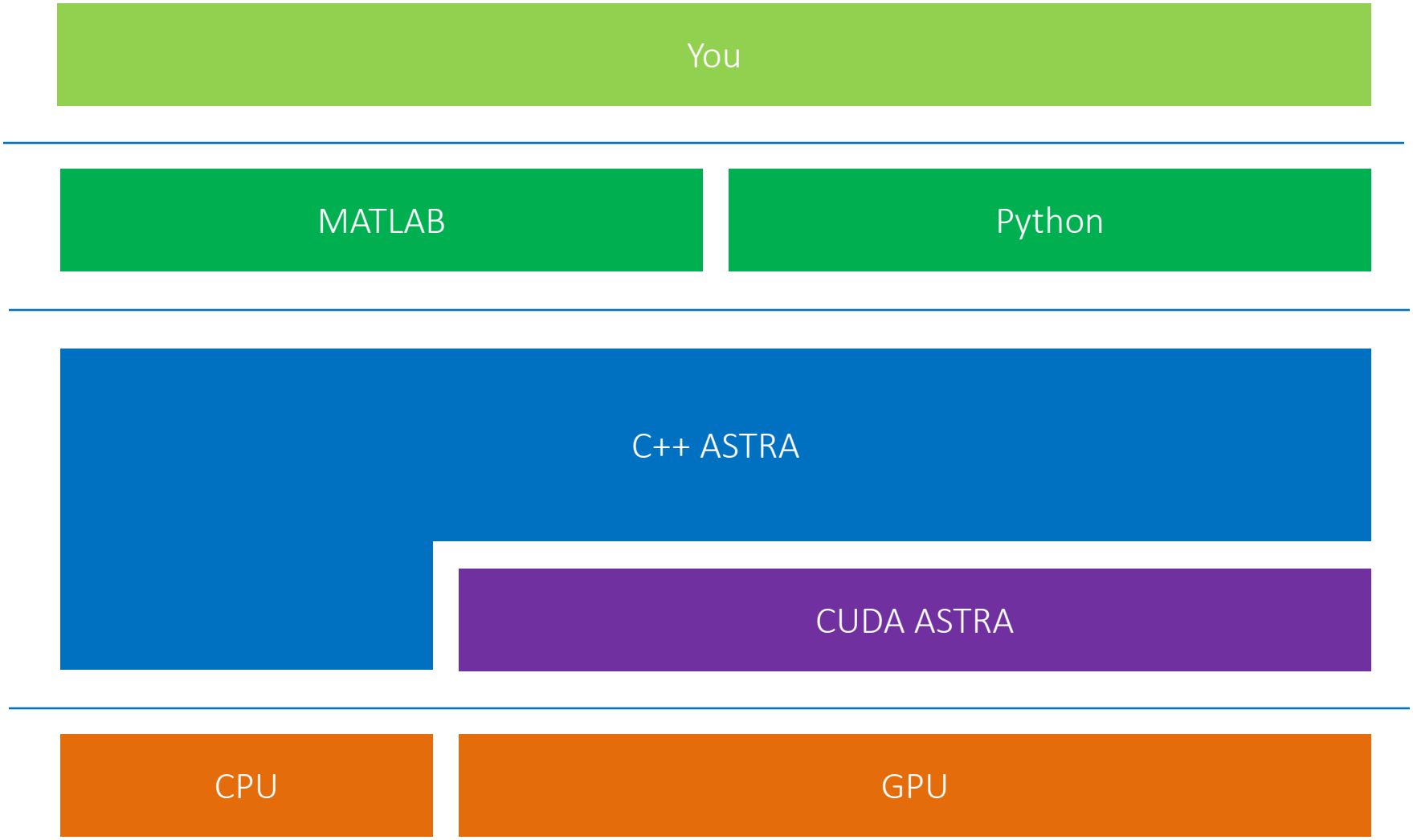
Broad support
algorithms and geometries

Powerful
C++ and CUDA

Easy to use
Interface to Matlab and Python

Flexible
Building block for custom algorithms

Toolbox organization



CWI

Universiteit Antwerpen

Scope of today

ASTRA can be used for:

- 2D tomography using CPU
- 2D tomography using GPU
- 3D tomography using GPU
- All from Matlab and Python

The exercise sessions today will focus on Matlab+2D/CPU, but most topics will generalize readily to 3D and/or GPU usage.

The last lecture today will also explicitly cover 3D and GPU topics.

Toolbox concepts

volume geometry

- volume size
- number of pixels/voxels

options: 2D and 3D

limitation: pixel size should be 1x1, volume centred around origin

---

volume geometry

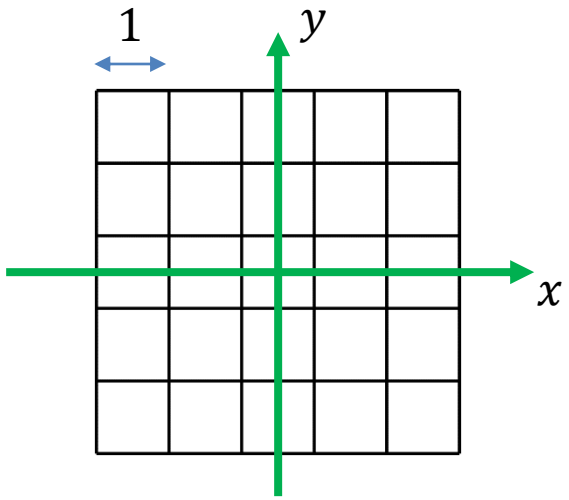
Toolbox concepts

volume geometry (2D)

number of columns (x)

number of rows (y)

```
vol_geom = astra_create_vol_geom(5, 5);
```



volume geometry

Toolbox concepts

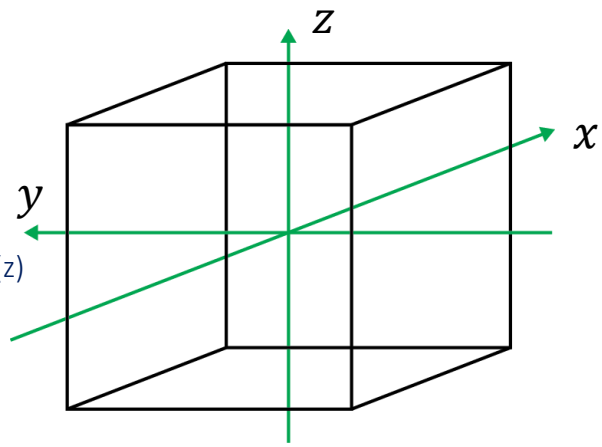
volume geometry (3D)

number of rows (y)

number of columns (x)

number of slices (z)

```
vol_geom = astra_create_vol_geom(5, 5, 5);
```

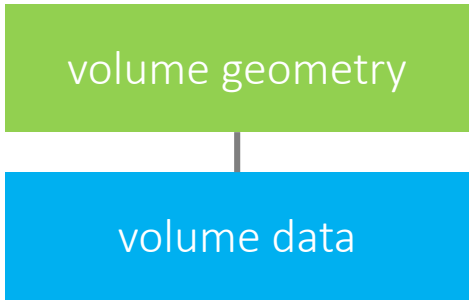


volume geometry

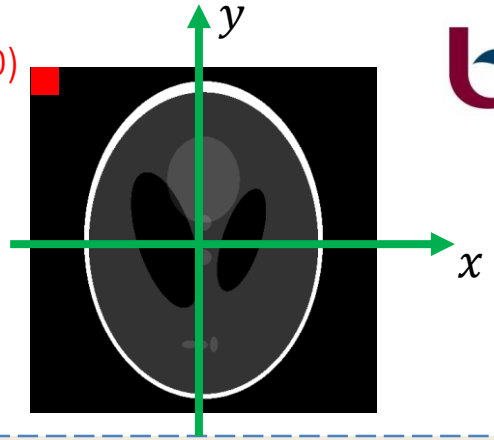
Toolbox concepts

volume data

- place to store data (e.g. reconstruction)
  - In C++: float*
  - In MATLAB/Python: id
- links to volume geometry



Toolbox concepts



row: x-axis

column: inverse y-axis

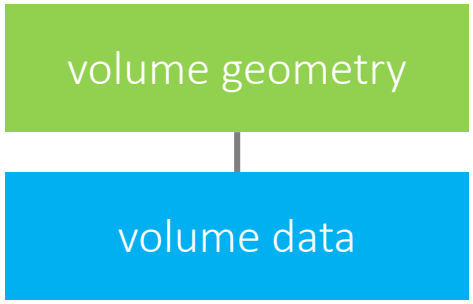
volume data (2D)

```
reconstruction_id = astra_mex_data2d('create', '-vol', vol_geom);
reconstruction_id = astra_mex_data2d('create', '-vol', vol_geom, 0);
reconstruction_id = astra_mex_data2d('create', '-vol', vol_geom, V);
```

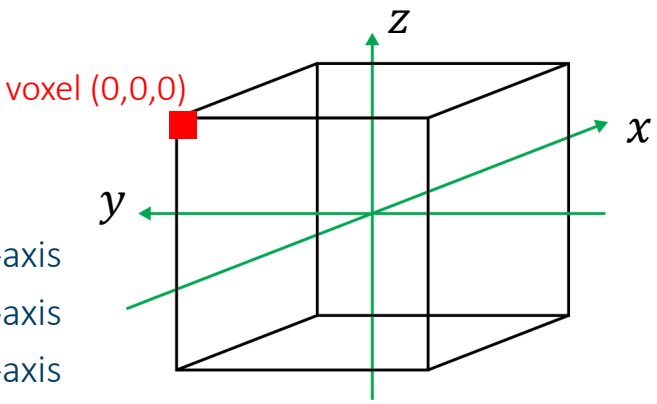
```
astra_mex_data2d('store', reconstruction_id , 0);
astra_mex_data2d('store', reconstruction_id , V);
```

```
reconstruction = astra_mex_data2d('get', reconstruction_id );
```

```
astra_mex_data2d('delete', reconstruction_id);
```



Toolbox concepts



row: x-axis

column: inverse y-axis

Slice: inverse z-axis

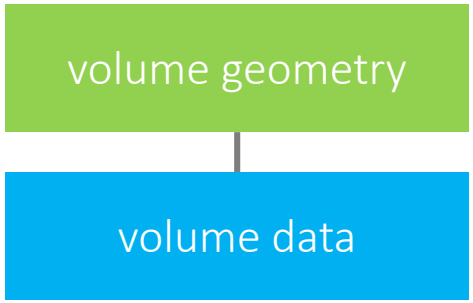
volume data (3D)

```
reconstruction_id = astra_mex_data3d('create', '-vol', vol_geom);
reconstruction_id = astra_mex_data3d('create', '-vol', vol_geom, 0);
reconstruction_id = astra_mex_data3d('create', '-vol', vol_geom, V);
```

```
astra_mex_data3d('store', reconstruction_id , 0);
astra_mex_data3d('store', reconstruction_id , V);
```

```
reconstruction = astra_mex_data3d('get', reconstruction_id );
```

```
astra_mex_data3d('delete', reconstruction_id);
```



Toolbox concepts

projection geometry

- trajectory of source and detector plane
- number of detectors
- detector size

options: 2D: parallel-beam, fan-beam; 3D: parallel-beam, cone-beam; vector based geometries

limitations: detectors must be square and central detector must cast ray through origin

---

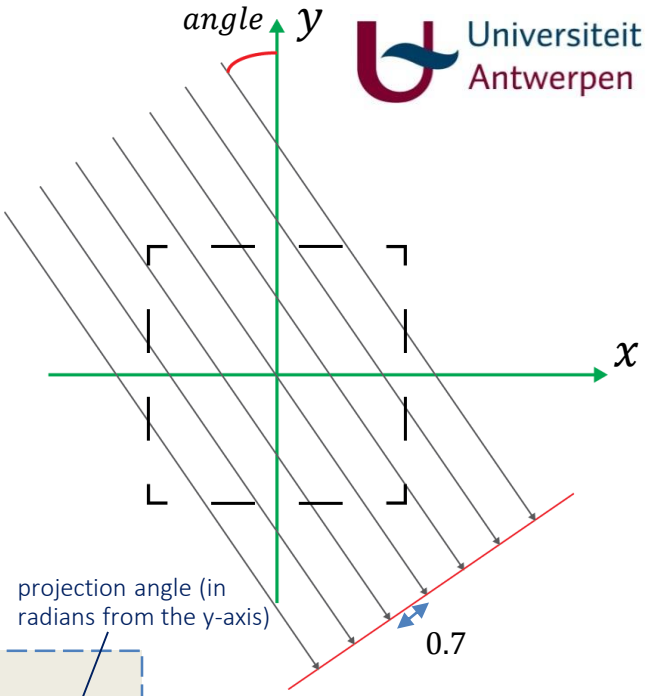
projection geometry

volume geometry

volume data

Toolbox concepts

projection geometry (2D parallel)



detector width

detector count (per projection)

projection angle (in radians from the y-axis)

```
angles = linspace2(0,pi,180);
proj_geom = astra_create_proj_geom('parallel', 0.7, 7, angles);
```

projection geometry

volume geometry

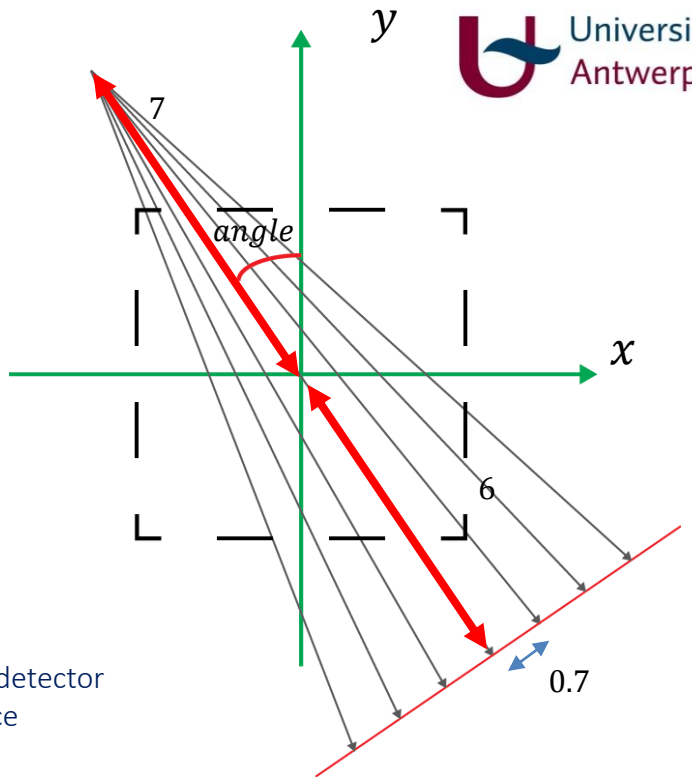
volume data

Toolbox concepts

projection geometry (2D fan-beam)

```
angles = linspace2(0,pi,180);
proj_geom = astra_create_proj_geom(
  'fanflat', 0.7, 7, angles, 7, 6);
```

- detector width
- detector count (per projection)
- projection angle (in radians from the y-axis)
- source-origin distance
- origin-detector distance



projection geometry

volume geometry

volume data

Toolbox concepts

projection geometry (3D cone-beam)

```
angles = linspace2(0,pi,180);
proj_geom = astra_create_proj_geom('cone', 0.7, 0.7, 7, 7, angles, 7, 6);
```

detector width

detector height

projection angle (in radians from the y-axis)

origin-detector distance

detector row count (per projection)

detector col count (per projection)

source-origin distance

---

projection geometry

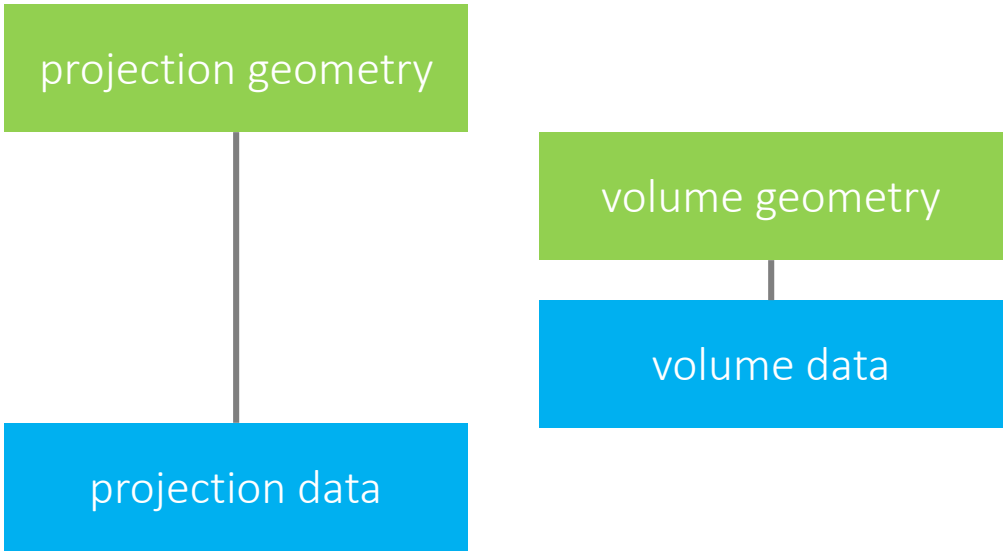
volume geometry

volume data

Toolbox concepts

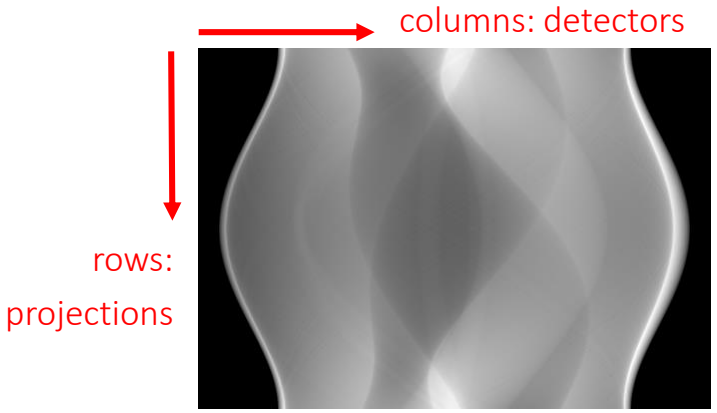
projection data

- place to store projection (e.g. sinogram)
  - similar to volume data
- links to projection geometry

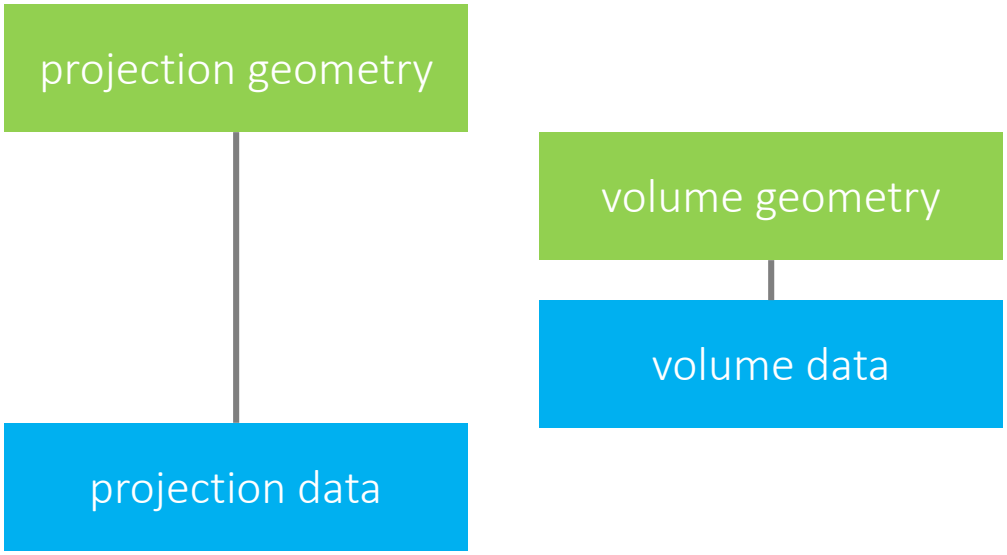


Toolbox concepts

projection data (2D)

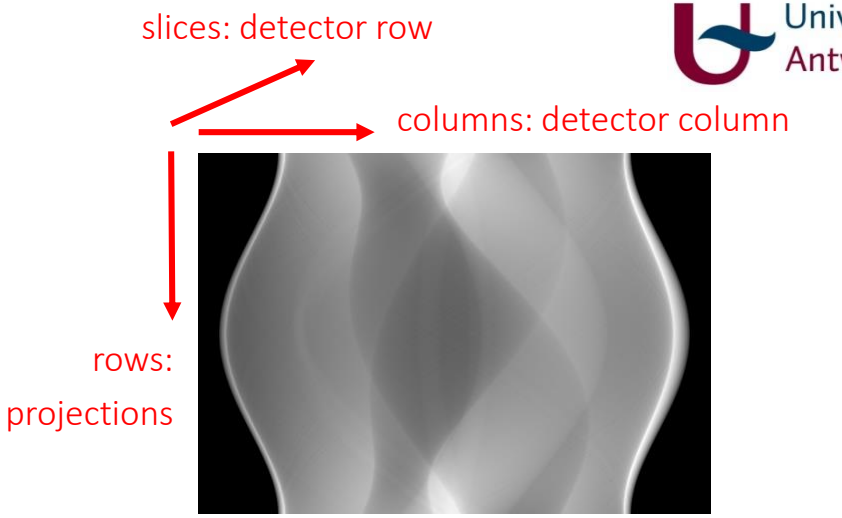



```
projection_id = astra_mex_data2d('create', '-sino', proj_geom);
projection_id = astra_mex_data2d('create', '-sino', proj_geom, 0);
projection_id = astra_mex_data2d('create', '-sino', proj_geom, V);
```

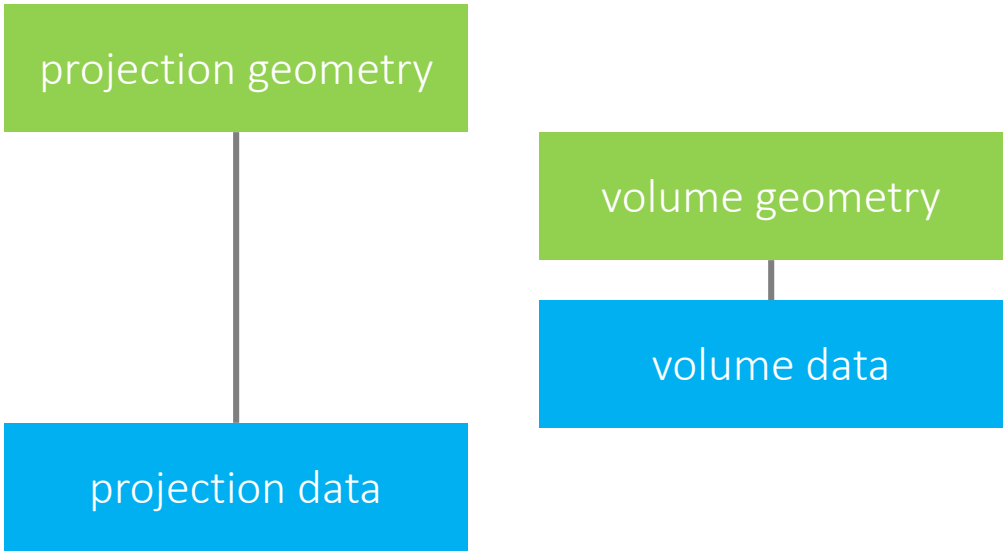


Toolbox concepts

projection data (3D)



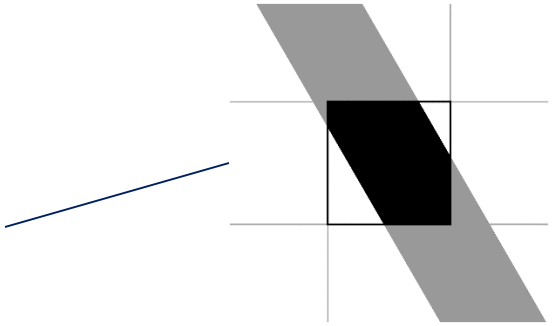

```
projection_id = astra_mex_data3d('create', '-sino', proj_geom);
projection_id = astra_mex_data3d('create', '-sino', proj_geom, 0);
projection_id = astra_mex_data3d('create', '-sino', proj_geom, V);
```



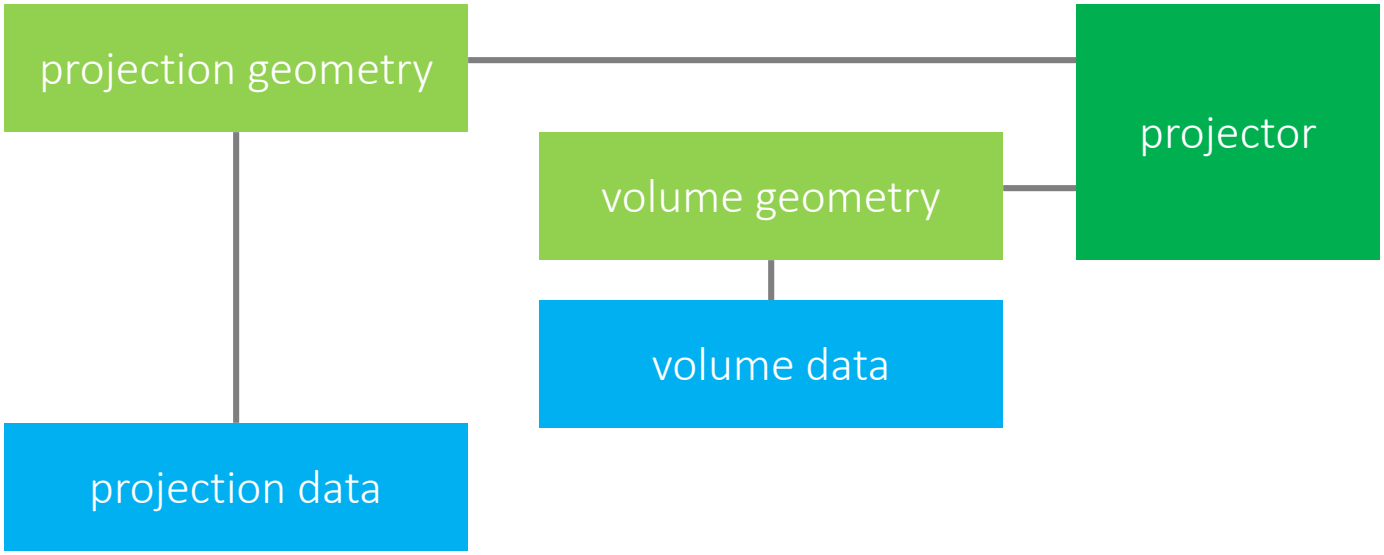
Toolbox concepts

Projector

- links volume geometry to projection geometry
- only for CPU algorithms
- in matlab/python: stored by id



options 2D parallel-beam: line, josephs kernel ('linear'), strip
2D fan-beam: line, strip
3D cone-beam: josephs kernel ('linear')

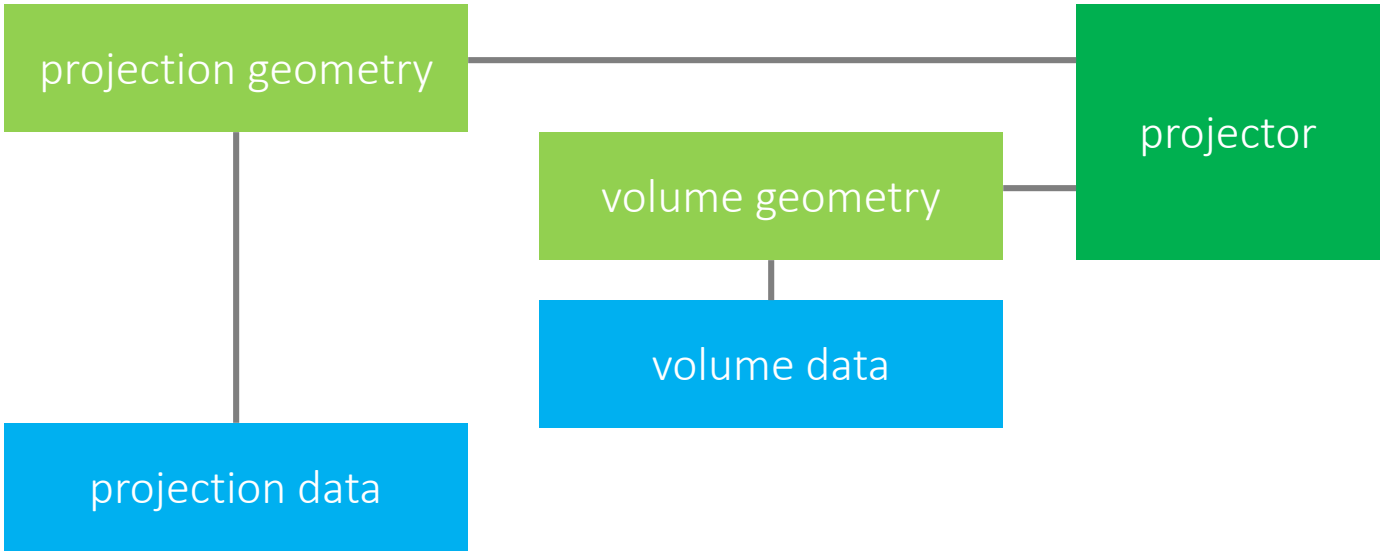


Toolbox concepts

Projector

```
projector_id = astra_create_projector('strip', proj_geom, vol_geom);
```

```
astra_mex_projector('delete', projector_id );
```



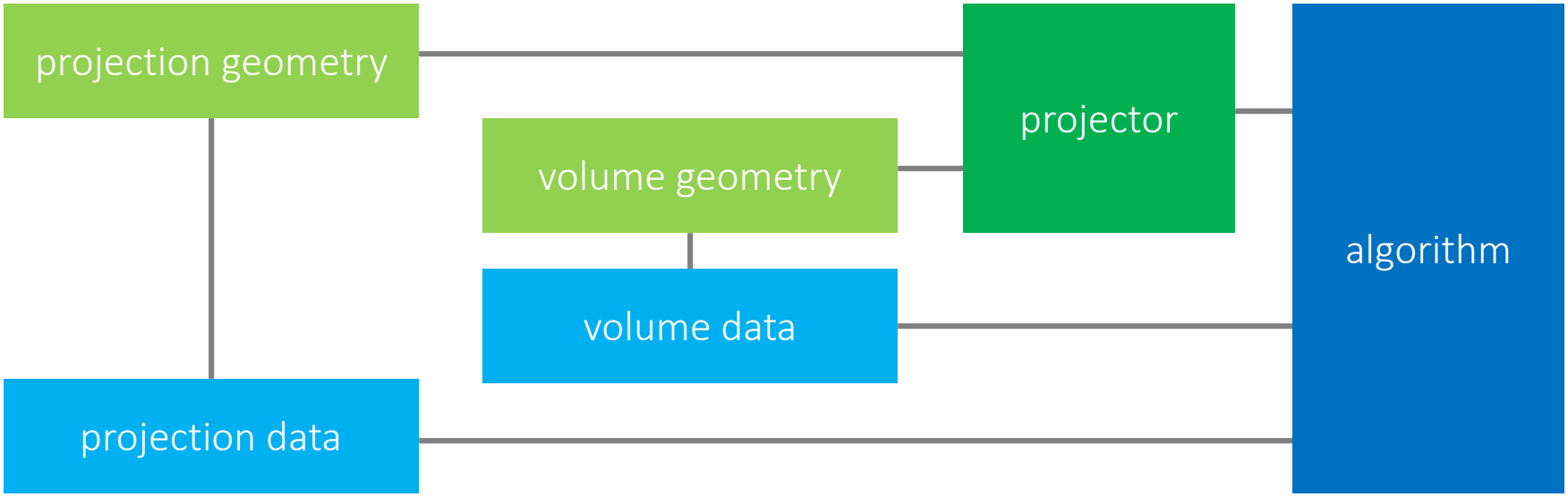
Toolbox concepts

Algorithm

- Where the computations happen
- Configured with matlab structs / python dicts
- Stored by id

Options, CPU: FBP, ART, SART, SIRT, CGLS, FP, BP

GPU: FBP, FDK, SIRT, CGLS, FP, BP



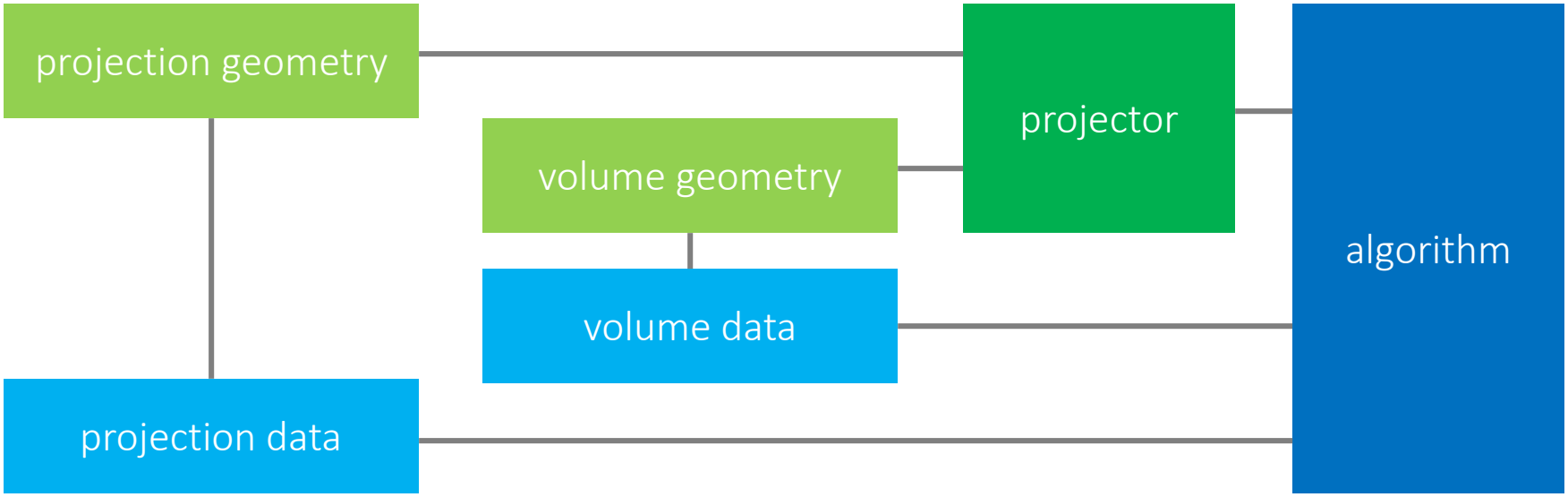
Toolbox concepts

Algorithm (CPU)

```
cfg = astra_struct('SIRT');
cfg.ProjectionDataId = projection_id;
cfg.ReconstructionDataId = reconstruction_id;
cfg.ProjectorId = projector_id;
alg_id = astra_mex_algorithm('create', cfg);
```

```
astra_mex_algorithm('iterate', alg_id, 50);
```

```
astra_mex_algorithm('delete', alg_id);
```



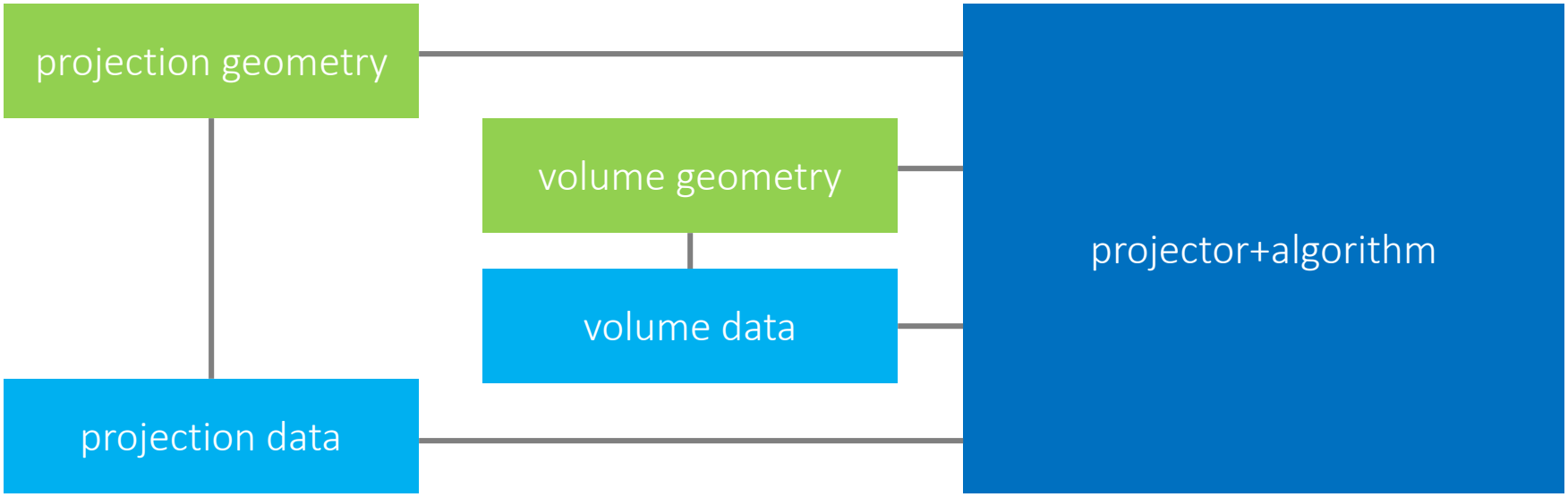
Toolbox concepts

Algorithm (GPU)

```
cfg = astra_struct('SIRT_CUDA');
cfg.ProjectionDataId = projection_id;
cfg.ReconstructionDataId = reconstruction_id;
alg_id = astra_mex_algorithm('create', cfg);
```

```
astra_mex_algorithm('iterate', alg_id, 50);
```

```
astra_mex_algorithm('delete', alg_id);
```



CWI

Universiteit Antwerpen

Intermediate results

With most iterative algorithms, it is possible to continue iterating after a set of iterations.

```
for i = 1:10
    astra_mex_algorithm('iterate', alg_id, 10);
    r = astra_mex_data2d('get', rec_id);
    % (process or display r)
end
```

Note: with CGLS the output will be different than doing 100 iterations at once, since that resets with each run. For SART, SIRT the output will be the same.

CWI

Universiteit Antwerpen

Tools of the toolbox

- Create a forward projection

```
[fp_id, fp] = astra_create_sino(volume, projector_id);
```

- Create a backprojection

```
[bp_id, bp] = astra_create_backprojection(fp, projector_id);
```

- Delete all astra objects

```
astra_clear;
```

CWI

Universiteit Antwerpen

More information

Where to find information?

- These slides (see workshop webpage for PDF)
- Docs: http://www.astra-toolbox.com/
- Samples: <astra_root>/samples/

CWI

Universiteit Antwerpen

Today

- Introduction to ASTRA
- **Exercises (see workshop webpage for PDF, part 1)**
- More on ASTRA usage
- Exercises
- Extra topics
- Hands-on, questions, discussion

Practical issues

For the exercises, you need:

- Either, have ASTRA installed in Matlab on your laptop.
  Later today, you'll also need SPOT:
  http://www.cs.ubc.ca/labs/scl/spot

- Or, use thinlinc to access the DTU server.
  Start Matlab using Applications->DTU->Mathematics->ASTRA
  Add ASTRA to your matlab path:

```
addpath('/appl/astra/1.7.1beta/matlab/mex');
addpath('/appl/astra/1.7.1beta/matlab/tools');
```

  Samples are in /appl/astra/1.7.1beta/samples/matlab