Registration and Resampling in SIRF

Richard Brown

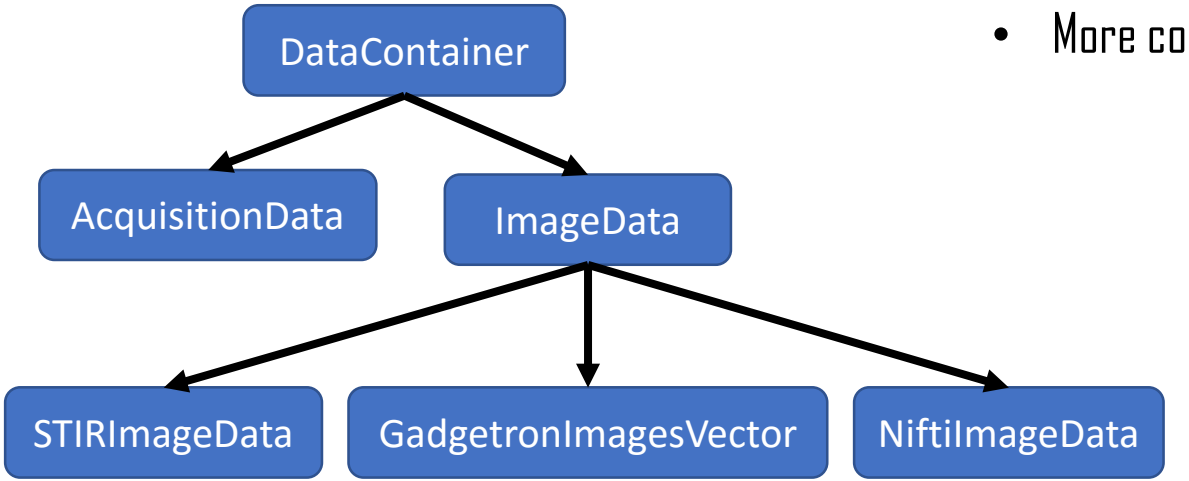
15/02/19

Reg

- Wrapper around NIfTI and NiftyReg
- Available in C++, Python, Matlab
- Merged in master, enabled by default
- Resampling and registration (rigid, affine, non-rigid)

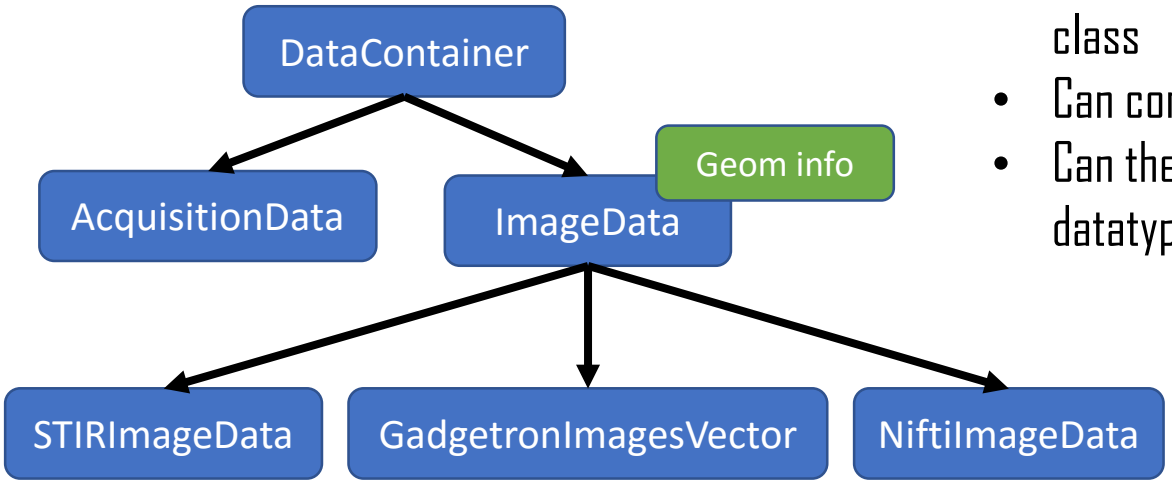
- After restructure of image hierarchy, can register resample different image types

Data hierarchy



- Necessary for any kind of synergy
- More common methods (e.g., clone, maths)

Data hierarchy



- Images construct a common "geometrical info" class
- Can construct NiftiImageData from this
- Can therefore register/resample different datatypes

PYTHON REGISTRATION

Can change

Any image type

Convert to NIfTI (if necessary)

Clone ref. with flo. data

```
import pReg
import pSTIR as eng_ref
import pGadgetron as eng_flo

ref = eng_ref.ImageData("PET_im.hv")
flo = eng_flo.ImageData("MR_im.h5")

registration = pReg.NiftyAladinSym()
registration.set_reference_image(ref)
registration.set_floating_image(flo)
# set any parameters
registration.process()

output = registration.get_output()
output.write("MR_as_PET.hs")
```

MATLAB ALIASES

- Aliases are tricky in Matlab (no "import A as B")
- Instead, create struct (eng_ref) with handles to all classes in library (mSTIR)
- After that, functionality is the same

```
set_up_engine('Reg');
eng_ref = set_up_engine('mSTIR');
eng_flo = set_up_engine('mGadgetron');

ref = eng_ref.ImageData("PET_im.hv");
flo = eng_flo.ImageData("MR_im.h5");

registration = mReg.NiftyAladinSym();
registration.set_reference_image(ref);
registration.set_floating_image(flo);
% set any parameters
registration.process();

output = registration.get_output();
output.write("MR_as_PET.hs");
```

Conclusion

- Better image hierarchy
- Convert images to NIfTI
- Register/resample different image types
- Aliases in Matlab

Caveats

- Can't quite do Gadgetron->NIfTI yet
- Can't do motion estimation (joint recon.)