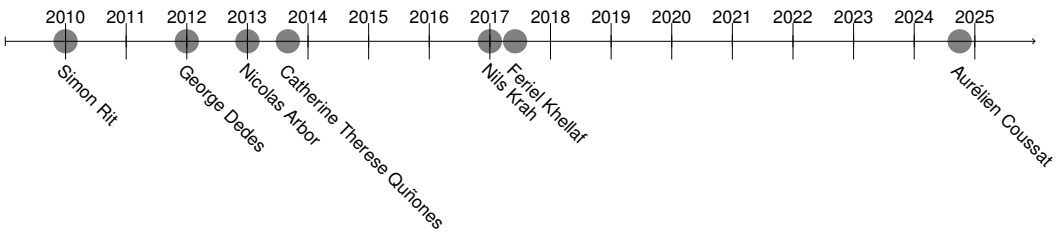
PCT, an Insight Toolkit (ITK) module for proton computed tomography

Simon Rit, Nicolas Arbor, Aurélien Coussat, George Dedes, Axel Garcia, Lucas Gandel, Sébastien Goll, Feriel Khellaf, Nils Krah, Catherine Therese Quiñones

October 21, 2025

PCT started with distance driven binning[1]



+ Other LMU students: Jannis Dickmann and Stefanie Götz
+ Software developers: Lucas Gandel and Sébastien Goll (Kitware), Axel Garcia.

[1] S. Rit, G. Dedes, N. Freud, D. Sarrut, and J.M. Létang. "Filtered backprojection proton CT reconstruction along most likely paths". In: *Med Phys* 40.3, 031103 (2013), p. 031103. DOI: 10.1118/1.4789589.

Used by several groups on a variety of datasets

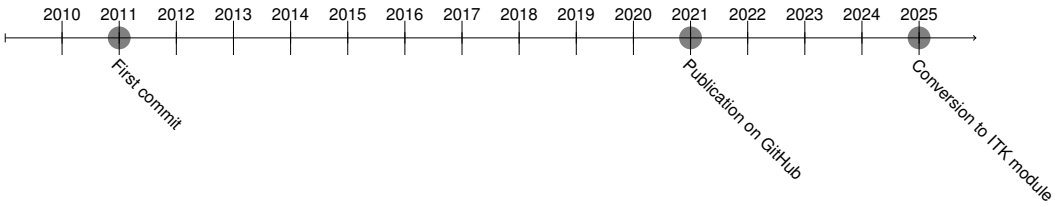
- Simulated data
- Real data of the LLU and ProtonVDA prototypes[2]
- Real data of the INFN prototype[3]

[2] G. Dedes, H. Drosten, S. Götz, J. Dickmann, C. Sarosiek, M. Pankuch, N. Krah, S. Rit, V. Bashkirov, R.W. Schulte, R.P. Johnson, K. Parodi, E. DeJongh, and G. Landry. "Comparative accuracy and resolution assessment of two prototype proton computed tomography scanners". In: *Med Phys* 49.7 (2022), pp. 4671–4681. DOI: 10.1002/mp.15657, frame.

[3] M. Scaringella, M. Bruzzi, P. Farace, E. Fogazzi, R. Righetto, S. Rit, F. Tommasino, E. Verroi, and C. Civinini. "The INFN proton computed tomography system for relative stopping power measurements: calibration and verification". In: *Phys Med Biol* 68.15 (2023), p. 154001. DOI: 10.1088/1361-6560/ace2a8, frame.

Repository history

https://github.com/RTKConsortium/PCT



Copyright and License

https://github.com/RTKConsortium/PCT/blob/hooks/LICENSE

- Apache 2.0 license approved by the Open Source Initiative (OSI)
- Same as ITK
- A non-contaminating license (BSD-like)

$\Rightarrow$ Appealing to companies

PCT one year ago

- Configured using CMake but compiled on Linux only
- Mainly C++ (only one python script)
- A few command-line tools using Gengetopt for parsing the command line
- A few dependencies
  - Insight toolkit (ITK) with the reconstruction toolkit (RTK)
  - ROOT
  - Geant4 (optional)

$\Rightarrow$ Complex and mandatory compilation

PCT redesign goals

- Make PCT a (remote) module of ITK, as RTK[4]
- Remove other dependencies (Geant4 and ROOT), at least in C++

⇒ Simpler compilation for developers, ITK compilation only

- Translate command line tools to Python (but keep C++ and Python versions)
- Distribute Python packages for Linux, Windows and MacOS

⇒ No compilation required for users, just `pip install itk-pct`

- Documentation https://proton-ct.readthedocs.io/

[4] S. Rit, S. Brousmiche, J. Finet, G.C. Sharp, and P. Steininger. "Reconstruction Toolkit (RTK) v2, an Insight Toolkit (ITK) module for tomographic reconstruction". In: *Proceedings of the XXth International Conference on the use of Computers in Radiation therapy (ICCR)*. Lyon, France, 2024.

Insight ToolKit (ITK)

https://www.itk.org/

”ITK is an open-source, cross-platform library (...) for processing, segmentation, and registration of scientific images in two, three, or more dimensions.“

Insight ToolKit (ITK)

Basic requirements for a C++ image processing library for sciences:

- Pixel values can be of many types $\Longrightarrow$ templates
  PCT: `float`, `itk::Vector<float, 3>`.
- Images can have any dimension $\Longrightarrow$ templates and iterators
  PCT: 2D, 3D and 4D.
- "Scientific" + "any dimension" $\Longrightarrow$ huge images

$\Rightarrow$ Piece-by-piece processing (streaming), multi-threading

ITK's modular architecture

- ITK splits its code in modules

  https://github.com/InsightSoftwareConsortium/ITK/tree/master/Modules

- Dependency graph based on CMake options (see `itk-module.cmake`)
- External modules: `External` or `Remote`
- RTKConsortium currently maintains
  - 2 remote modules: `CudaCommon` and `RTK`
  - 1 external (to be remote) module: `PCT`

Python wrappings

- Each ITK module has its own Python package
- In each module, `.wrap` files define a list of pre-compiled templates
  See e.g. `pctFDKDDConeBeamReconstructionFilter.wrap`
- Only the required packages are loaded
- Up-to-date packages are compiled by the CI as GitHub artifacts

PCT code organization

Available at `https://github.com/RTKConsortium/PCT`

Organization of the subdirectories

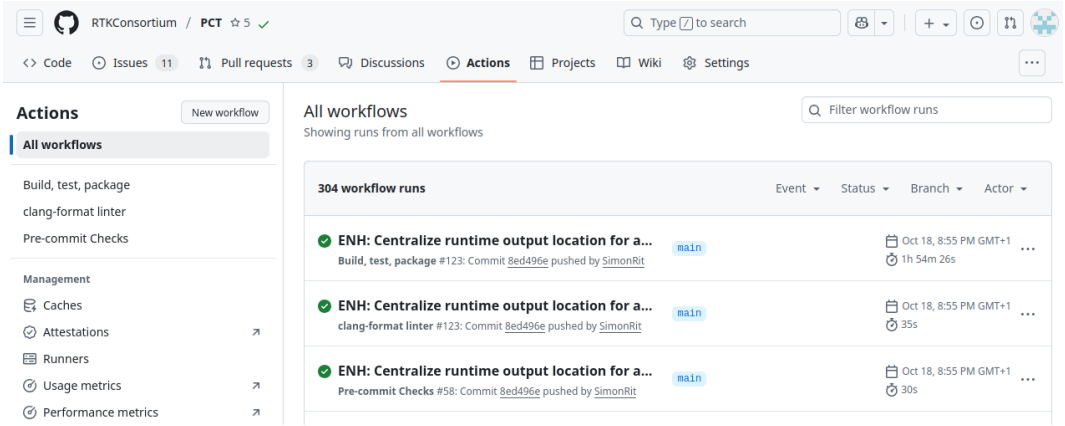
- `include`: PCT `.h{xx}` headers,
- `src`: PCT `.cxx` (non-templated) source files,
- `test`
- `wrapping`: CMake wrapping macros,
- `cmake`: CMake modules,
- `utilities`: third party code, e.g., `gengetopt`
- `applications`: command line tools,
- `documentation`, `gate`, `scripts`

Command line tools

- One application per subdirectory in the `applications` folder
- First C++, on-going translation to Python except `pctpairprotons` and `pctweplfit` which are Python only
- `--help` for the complete list of options from the command line, `help(application)` in Python
- 3 versions in Python (in the future release / CI packages)
  - Command line tool (i.e. reloading the PCT package which is slow):
    ```
    pctfdk -g geometry.xml -p . -r proj.*.mha -o recon.mha
    ```
  - Python function with a single string:
    ```
    pct.pctfdk('-g geometry.xml -p . -r proj.*.mha -o recon.mha')
    ```
  - Python function with keyword arguments:
    ```
    pct.pctfdk(geometry='geometry.xml', path='.',
               regexp='-r proj.*.mha', output='recon.mha')
    ```

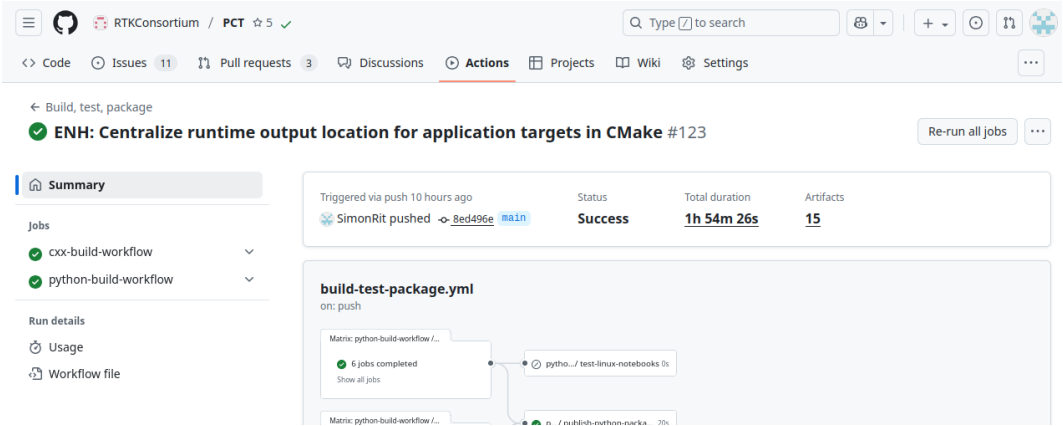
Testing

Continuous integration with GitHub actions:
https://github.com/RTKConsortium/PCT/actions



Testing

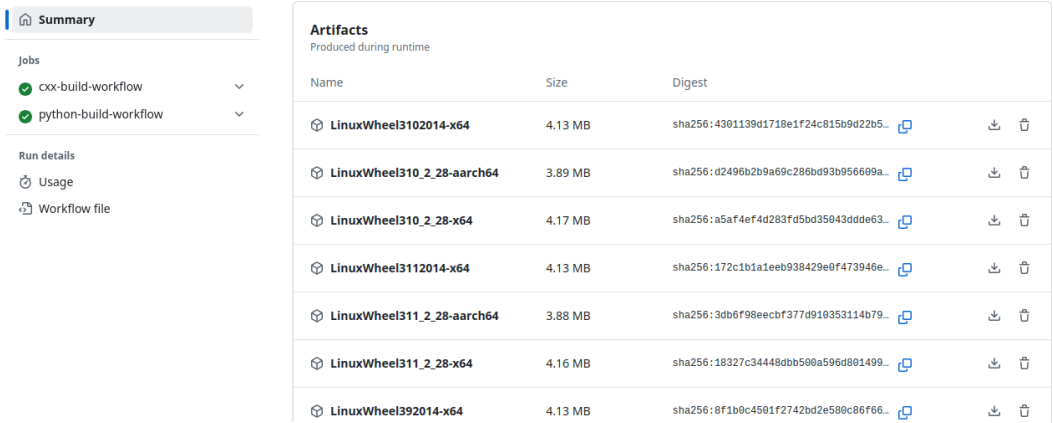
Continuous integration with GitHub actions:
https://github.com/RTKConsortium/PCT/actions



Testing

Continuous integration with GitHub actions:

https://github.com/RTKConsortium/PCT/actions



Python wrappings

C++ code of `pctfdk.cxx`

```
using ImageType = itk::Image<float, 3>;
auto feldkamp = pct::FDKDDConeBeamReconstructionFilter<ImageType>::New();
feldkamp->SetInput(0, constantImageSource->GetOutput());
feldkamp->SetProjectionStack(pssf->GetOutput());
feldkamp->SetGeometry(mygeo);
```

Python code `pctfdk.py`

```
ImageType = itk.Image[itk.F, 3]
feldkamp = pct.FDKDDConeBeamReconstructionFilter[ImageType].New()
feldkamp.SetInput(0, constantImageSource.GetOutput())
feldkamp.SetProjectionStack(pssf.GetOutput())
feldkamp.SetGeometry(mygeo)
```

or using the Pythonic interface

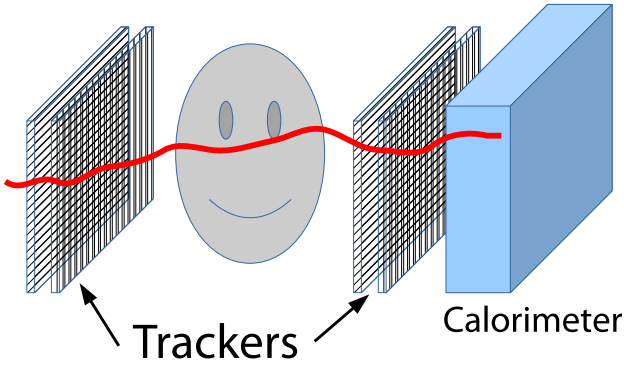
```
pct.fdkdd_cone_beam_reconstruction_filter(constantImageSource.GetOutput(),
                                          projection_stack=pssf.GetOutput(),
                                          geometry=mygeo)
```

Full simulation and energy-loss reconstruction example

```
python protonct.py --output gate_simulation --verbose

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

Outline

List mode ion CT data

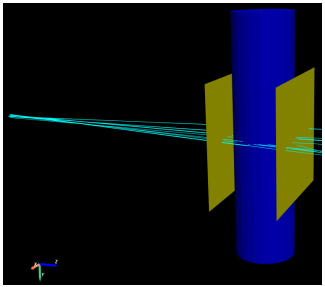


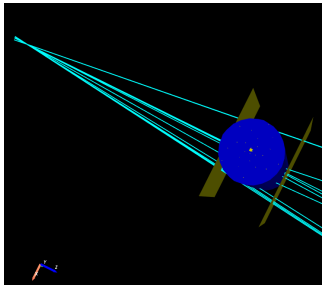
Known: E_{in}

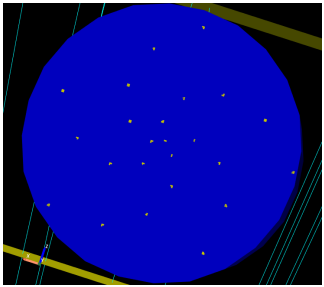
Records: E_{out}, $\boldsymbol{x}_{\text{in}}$, $\boldsymbol{x}_{\text{out}}$, $\frac{\mathrm{d}\boldsymbol{x}_{\text{in}}}{\mathrm{d}t}$, $\frac{\mathrm{d}\boldsymbol{x}_{\text{out}}}{\mathrm{d}t}$

Gate v10 simulation[5]

```
pip install opengate
python protonct.py --output gate_simulation --verbose --visu
```







Outputs: 2 ROOT files with the position, direction, energy, time of particle and Geant4 IDs in each (yellow) plane

[5] https://ionimaging.org/assets/talks/ws2024vienna-nils-krah.pdf

Gate simulation code

```
import math
import argparse

import opengate as gate
from scipy.spatial.transform import Rotation

def protonct(
    output,
    projections=720,
    protons_per_projection=1000,
    seed=None,
    visu=False,
    verbose=False,
):

    # Units
    nm = gate.g4_units.nm
    mm = gate.g4_units.mm
    cm = gate.g4_units.cm
    m = gate.g4_units.m
    sec = gate.g4_units.second
    MeV = gate.g4_units.MeV
    Bq = gate.g4_units.Bq
```

Gate simulation code

```
# Simulation
sim = gate.Simulation()

sim.random_engine = "MersenneTwister"
sim.random_seed = "auto" if seed is None else seed
sim.check_volumes_overlap = False
sim.visu = visu
sim.visu_type = "vrml"
sim.g4_verbose = False
sim.progress_bar = verbose
sim.number_of_threads = 1

sim.run_timing_intervals = [[j * sec, (j + 1) * sec] for j in range(projections)]

# Misc
yellow = [1, 1, 0, 0.5]
blue = [0, 0, 1, 0.5]

# Geometry
sim.volume_manager.add_material_database(
    gate.utility.get_contrib_path() / "GateMaterials.db"
)
sim.world.material = "Vacuum"
sim.world.size = [4 * m, 4 * m, 4 * m]
sim.world.color = [0, 0, 0, 0]
```

Gate simulation code

```
# Phantom

def add_spiral(sim):
    # Mother of all
    spiral = sim.add_volume("Tubs", name="Spiral")
    spiral.rmin = 0 * cm
    spiral.rmax = 10 * cm
    spiral.dz = 40 * cm
    spiral.material = "Water"
    spiral.color = blue
    spiral.rotation = Rotation.from_euler("yz", [90, 90], degrees=True).as_matrix()

    # Spiral rotation
    tr, rot = gate.geometry.utility.volume_orbiting_transform(
        "y", 0, 360, projections, spiral.translation, spiral.rotation
    )
    spiral.add_dynamic_parametrisation(translation=tr, rotation=rot)

    # Spiral inserts
    sradius = 4
    radius = list(range(0, 100 - sradius // 2, sradius))
    sangle = 139
    angles = [math.radians(a) for a in range(0, sangle * len(radius), sangle)]
    posx = [radius[i] * math.cos(angles[i]) for i in range(len(radius))]
    posy = [radius[i] * math.sin(angles[i]) for i in range(len(radius))]
```

Gate simulation code

```
def add_spiral_insert(
    sim,
    mother,
    name,
    rmin=0 * mm,
    rmax=1 * mm,
    dz=40 * cm,
    material="Aluminium",
    translation=None,
    color=None,
):

    spiral_insert = sim.add_volume("Tubs", name=name)
    spiral_insert.mother = mother.name
    spiral_insert.rmin = rmin
    spiral_insert.rmax = rmax
    spiral_insert.dz = dz
    spiral_insert.material = material
    spiral_insert.translation = translation
    spiral_insert.color = color

for i in range(len(radius)):
    add_spiral_insert(
        sim,
        spiral,
        f"SpiralInsert{i:02d}",
        translation=[posx[i] * mm, posy[i] * mm, 0],
    )
```

Gate simulation code

```
# Beam
source = sim.add_source("GenericSource", "mybeam")
source.particle = "proton"
source.energy.mono = 200 * MeV
source.energy.type = "mono"
source.position.type = "box"
source.position.size = [16 * mm, 1 * nm, 1 * nm]
source.position.translation = [0, 0, -1060 * mm]
source.direction.type = "focused"
source.direction.focus_point = [0, 0, -1000 * mm]

if sim.visu:
    # For visualisation speed, the number of particle is decreased
    source.activity = 10 * Bq
else:
    source.activity = protons_per_projection * Bq

# Physics list
sim.physics_manager.physics_list_name = "QGSP_BIC_EMZ"
```

Gate simulation code

```
# Phase spaces
def add_detector(sim, name, translation):
    plane = sim.add_volume("Box", "PlanePhaseSpace" + name)
    plane.size = [400 * mm, 400 * mm, 1 * nm]
    plane.translation = translation
    plane.material = "Vacuum"

    phase_space = sim.add_actor("PhaseSpaceActor", "PhaseSpace" + name)
    phase_space.attached_to = plane.name
    phase_space.attributes = [
        "RunID",
        "EventID",
        "TrackID",
        "KineticEnergy",
        "PreGlobalTime",
        "Position",
        "Direction",
    ]
    phase_space.output_filename = f"{output}/PhaseSpace{name}.root"
    ps_filter = sim.add_filter("ParticleFilter", "Filter" + name)
    ps_filter.particle = "proton"
    phase_space.filters.append(ps_filter)

add_detector(sim, "In", [0, 0, -110 * mm])
add_detector(sim, "Out", [0, 0, 110 * mm])

sim.run()
```

Outline

Full simulation and energy-loss reconstruction example

```
python protonct.py --output gate_simulation --verbose

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

Proton track reconstruction

Pair data of the same proton before and after the scanned object

Previously implemented in C++, hence the ROOT dependency

Now implemented in Python only using the package `uproot`

`pctpairprotons` produces one file per gantry position with perfect “track reconstruction” using the Geant4 IDs

Welcome to use your own track reconstruction algorithm and produce the list mode data in the PCT format
⇒ ITK to NumPy bridge, see `itk.array_from_image` and `itk.image_from_array`

List mode ion CT data format

- 2D image of 3D vectors: `itk::Image<itk::Vector<float, 3>, 2>`
- Each line corresponds to one proton
- 5 columns:
  - 2 vectors for the entrance and exit positions,
  - 2 vectors for the entrance and exit directions,
  - 1 vector for the entrance and exit energies and a user-defined scalar or 1 vector with 0., WEPL, and a user-defined scalar.

pctpairprotons --help

```
Pair corresponding protons from GATE ROOT files

options:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit
  -i INPUT_IN, --input-in INPUT_IN
                        Root phase space file of particles before object (default: None)
  -j INPUT_OUT, --input-out INPUT_OUT
                        Root phase space file of particles after object (default: None)
  -o OUTPUT, --output OUTPUT
                        Output file name (default: None)
  --plane-in PLANE_IN   Plane position of incoming protons (default: None)
  --plane-out PLANE_OUT
                        Plane position of outgoing protons (default: None)
  --min-run MIN_RUN     Minimum run (inclusive) (default: 0)
  --max-run MAX_RUN     Maximum run (exclusive) (default: 1000000.0)
  --no-nuclear          Remove inelastic nuclear collisions (default: False)
  --fit FIT             Fit file used to convert from energy loss or TOF to WEPL (default: None)
  --fit-kind {tof,energy}
                        Whether to convert to WEPL using energy loss or TOF (default: None)
  --store-time          Store time instead of energy in the output list-mode (default: False)
  --verbose, -v         Verbose execution (default: False)
  --psin PSIN           Name of tree in input phase space (default: PhaseSpace)
  --psout PSOUT         Name of tree in output phase space (default: PhaseSpace)
```

pctpairprotons --help

```
Pair corresponding protons from GATE ROOT files

options:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit
  -i INPUT_IN, --input-in INPUT_IN
                        Root phase space file of particles before object (default: None)
  -j INPUT_OUT, --input-out INPUT_OUT
                        Root phase space file of particles after object (default: None)
```

Input phase spaces

```
  -o OUTPUT, --output OUTPUT
                        Output file name (default: None)
  --plane-in PLANE_IN   Plane position of incoming protons (default: None)
  --plane-out PLANE_OUT
                        Plane position of outgoing protons (default: None)
  --min-run MIN_RUN     Minimum run (inclusive) (default: 0)
  --max-run MAX_RUN     Maximum run (exclusive) (default: 1000000.0)
  --no-nuclear          Remove inelastic nuclear collisions (default: False)
  --fit FIT             Fit file used to convert from energy loss or TOF to WEPL (default: None)
  --fit-kind {tof,energy}
                        Whether to convert to WEPL using energy loss or TOF (default: None)
  --store-time          Store time instead of energy in the output list-mode (default: False)
  --verbose, -v         Verbose execution (default: False)
  --psin PSIN           Name of tree in input phase space (default: PhaseSpace)
  --psout PSOUT         Name of tree in output phase space (default: PhaseSpace)
```

pctpairprotons --help

```
Pair corresponding protons from GATE ROOT files

options:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit
  -i INPUT_IN, --input-in INPUT_IN
                        Root phase space file of particles before object (default: None)
  -j INPUT_OUT, --input-out INPUT_OUT
                        Root phase space file of particles after object (default: None)
  -o OUTPUT, --output OUTPUT
                        Output file name (default: None)
  --plane-in PLANE_IN   Plane position of incoming protons (default: None)
  --plane-out PLANE_OUT
                        Plane position of outgoing protons (default: None)
  --min-run MIN_RUN     Minimum run (inclusive) (default: 0)
  --max-run MAX_RUN     Maximum run (exclusive) (default: 1000000.0)
  --no-nuclear          Remove inelastic nuclear collisions (default: False)
```

Artificially filter out unwanted nuclear events

```
  --fit FIT             Fit file used to convert from energy loss or TOF to WEPL (default: None)
  --fit-kind {tof,energy}
                        Whether to convert to WEPL using energy loss or TOF (default: None)
  --store-time          Store time instead of energy in the output list-mode (default: False)
  --verbose, -v         Verbose execution (default: False)
  --psin PSIN           Name of tree in input phase space (default: PhaseSpace)
  --psout PSOUT         Name of tree in output phase space (default: PhaseSpace)
```

pctpairprotons --help

```
Pair corresponding protons from GATE ROOT files

options:
  -h, --help            show this help message and exit
  -V, --version         show program's version number and exit
  -i INPUT_IN, --input-in INPUT_IN
                        Root phase space file of particles before object (default: None)
  -j INPUT_OUT, --input-out INPUT_OUT
                        Root phase space file of particles after object (default: None)
  -o OUTPUT, --output OUTPUT
                        Output file name (default: None)
  --plane-in PLANE_IN   Plane position of incoming protons (default: None)
  --plane-out PLANE_OUT
                        Plane position of outgoing protons (default: None)
  --min-run MIN_RUN     Minimum run (inclusive) (default: 0)
  --max-run MAX_RUN     Maximum run (exclusive) (default: 1000000.0)
  --no-nuclear          Remove inelastic nuclear collisions (default: False)
  --fit FIT             Fit file used to convert from energy loss or TOF to WEPL (default: None)
  --fit-kind {tof,energy}
                        Whether to convert to WEPL using energy loss or TOF (default: None)
```

Use fitting data produced by Gate simulations

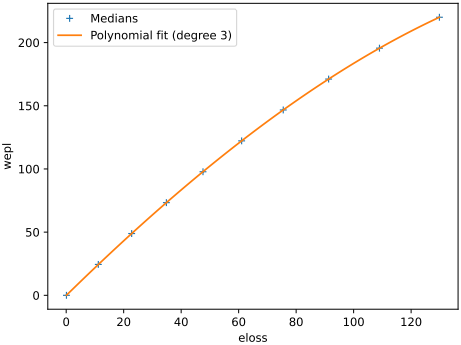
```
  --store-time          Store time instead of energy in the output list-mode (default: False)
  --verbose, -v         Verbose execution (default: False)
  --psin PSIN           Name of tree in input phase space (default: PhaseSpace)
  --psout PSOUT         Name of tree in output phase space (default: PhaseSpace)
```

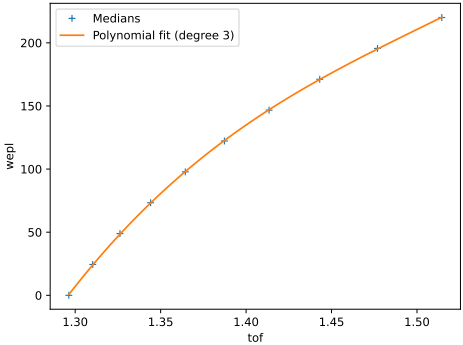
Outline

Full simulation with pctweplfit

```
python protonct.py --output gate_simulation --verbose

pctweplfit -o weplfit --savefig -v

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --fit weplfit/eloss_to_wepl_fit_deg3.json \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

pctwelpfit output





pctweplfit --help

```
-h, --help            show this help message and exit
-V, --version         show program's version number and exit
-o OUTPUT, --output OUTPUT
                      Path of outputs (default: pctweplfit)
-n NUMBER_OF_PARTICLES, --number-of-particles NUMBER_OF_PARTICLES
                      Number of generated particles (default: 10000)
--path-type {phantom_length,simple,realistic}
                      How to compute proton path (default: simple)
--phantom-length-samples PHANTOM_LENGTH_SAMPLES
                      Number of phantom length samples (default: 10)
-w PHANTOM_WIDTH, --phantom-width PHANTOM_WIDTH
                      Phantom width (default: 40.0)
-d DETECTOR_DISTANCE, --detector-distance DETECTOR_DISTANCE
                      Distance between detectors (default: 220.0)
-e INITIAL_ENERGY, --initial-energy INITIAL_ENERGY
                      Initial energy of the protons (in MeV) (default: 200.0)
--number-of-detectors NUMBER_OF_DETECTORS
                      Number of detectors in the phantom (default: 10)
--polydeg-min POLYDEG_MIN
                      Minimum polynom degree (default: 3)
--polydeg-max POLYDEG_MAX
                      Maximum polynom degree (default: 3)
--visu                Visualize Monte Carlo simulation (default: False)
--display             Display polynomial fit plot (default: False)
--savefig             Write polynomial fit plot to disk (default: False)
--verbose, -v         Verbose execution (default: False)
```

Outline

Full simulation and energy-loss reconstruction example

```
python protonct.py --output gate_simulation --verbose

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

Nuclear interactions

Protons which have undergone nuclear interactions are a nuisance and must be filtered out before reconstruction.

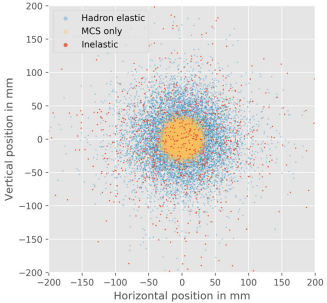


Figure from N. Krah, F. Khellaf, J.M. Letang, S. Rit, and I. Rinaldi. “A comprehensive theoretical comparison of proton imaging set-ups in terms of spatial resolution.”. In: *Phys Med Biol* 63.13 (13 2018), p. 135013. DOI: 10.1088/1361-6560/aaca1f

Nuclear interactions

Protons which have undergone nuclear interactions are a nuisance and must be filtered out before reconstruction.

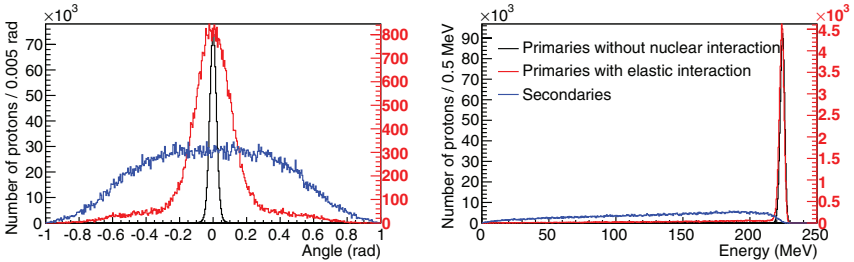


Figure from C.T. Quiñones, J.M. Létang, and S. Rit. “Filtered back-projection reconstruction for attenuation proton CT along most likely paths”. In: *Phys Med Biol* 61.9 (2016), pp. 3258–3278. DOI: 10.1088/0031-9155/61/9/3258

pctpaircuts --help

```
  -i, --input=STRING       Input file name containing the proton pairs
  -o, --output=STRING      Output file name
  -s, --source=DOUBLE      Source position  (default='0.')
      --anglecut=DOUBLE    Cut parameter on the SD of proton angle. (default='3.')
      --energycut=DOUBLE   Cut parameter on the SD of proton energy. (default='3.')
  -r, --robust             Use robust estimation using 50/19.1 %ile.  (default=off)
      --robustopt=INT      Use newer options for robust cut.  (default='0')
      --primaries          Consider only primary protons  (default=off)
      --nonuclear          Consider only primary protons without nuclear interactions  (default=off)
      --wet                Write WET instead of initial energy  (default=off)

Projections parameters:
      --origin=DOUBLE      Origin (default=centered)
      --dimension=INT      Dimension(Deprecated) Use --size instead.  (default='256')
      --size=INT           Size  (default='256')
      --spacing=DOUBLE     Spacing  (default='1')
      --direction=DOUBLE   Direction
      --like=STRING        Copy information from this image (origin, size, spacing, direction)

Output cut images:
      --menergy=STRING     Mean energy file name
      --senergy=STRING     Sigma energy file name
      --sangle=STRING      Sigma angle file name
      --count=STRING       Count file name
```

pctpaircuts --help

```
  -i, --input=STRING       Input file name containing the proton pairs
  -o, --output=STRING      Output file name
  -s, --source=DOUBLE      Source position  (default='0.')
      --anglecut=DOUBLE    Cut parameter on the SD of proton angle. (default='3.')
      --energycut=DOUBLE   Cut parameter on the SD of proton energy. (default='3.')
  -r, --robust             Use robust estimation using 50/19.1 %ile.  (default=off)
      --robustopt=INT      Use newer options for robust cut.  (default='0')
      --primaries          Consider only primary protons  (default=off)
      --nonuclear          Consider only primary protons without nuclear interactions  (default=off)
      --wet                Write WET instead of initial energy  (default=off)

Projections parameters:
      --origin=DOUBLE      Origin (default=centered)
      --dimension=INT      Dimension(Deprecated) Use --size instead.  (default='256')
      --size=INT           Size  (default='256')
      --spacing=DOUBLE     Spacing  (default='1')
      --direction=DOUBLE   Direction
      --like=STRING        Copy information from this image (origin, size, spacing, direction)
```

Define a grid of pixels on the entrance detector plane

```
Output cut images:
      --menergy=STRING     Mean energy file name
      --senergy=STRING     Sigma energy file name
      --sangle=STRING      Sigma angle file name
      --count=STRING       Count file name
```

pctpaircuts --help

```
  -i, --input=STRING       Input file name containing the proton pairs
  -o, --output=STRING      Output file name
  -s, --source=DOUBLE      Source position  (default='0.')
      --anglecut=DOUBLE    Cut parameter on the SD of proton angle. (default='3.')
      --energycut=DOUBLE   Cut parameter on the SD of proton energy. (default='3.')
  -r, --robust             Use robust estimation using 50/19.1 %ile.  (default=off)
      --robustopt=INT      Use newer options for robust cut.  (default='0')
      --primaries          Consider only primary protons  (default=off)
      --nonuclear          Consider only primary protons without nuclear interactions  (default=off)
      --wet                Write WET instead of initial energy  (default=off)

Projections parameters:
      --origin=DOUBLE      Origin (default=centered)
      --dimension=INT      Dimension(Deprecated) Use --size instead.  (default='256')
      --size=INT           Size  (default='256')
      --spacing=DOUBLE     Spacing  (default='1')
      --direction=DOUBLE   Direction
      --like=STRING        Copy information from this image (origin, size, spacing, direction)

Output cut images:
      --menergy=STRING     Mean energy file name
      --senergy=STRING     Sigma energy file name
      --sangle=STRING      Sigma angle file name
      --count=STRING       Count file name
```

Bin protons which hit the same pixel of the entrance detector
Compute the energy and angle deviation means and sigmas

pctpaircuts --help

```
  -i, --input=STRING       Input file name containing the proton pairs
  -o, --output=STRING      Output file name
  -s, --source=DOUBLE      Source position  (default='0.')
      --anglecut=DOUBLE    Cut parameter on the SD of proton angle. (default='3.')
      --energycut=DOUBLE   Cut parameter on the SD of proton energy. (default='3.')
```

Select those within `--anglecut` / `--energycut` σ around the mean.

```
  -r, --robust             Use robust estimation using 50/19.1 %ile.  (default=off)
      --robustopt=INT      Use newer options for robust cut.  (default='0')
      --primaries          Consider only primary protons  (default=off)
      --nonuclear          Consider only primary protons without nuclear interactions  (default=off)
      --wet                Write WET instead of initial energy  (default=off)

Projections parameters:
      --origin=DOUBLE      Origin (default=centered)
      --dimension=INT      Dimension(Deprecated) Use --size instead.  (default='256')
      --size=INT           Size  (default='256')
      --spacing=DOUBLE     Spacing  (default='1')
      --direction=DOUBLE   Direction
      --like=STRING        Copy information from this image (origin, size, spacing, direction)

Output cut images:
      --menergy=STRING     Mean energy file name
      --senergy=STRING     Sigma energy file name
      --sangle=STRING      Sigma angle file name
      --count=STRING       Count file name
```

pctpaircuts --help

```
  -i, --input=STRING       Input file name containing the proton pairs
  -o, --output=STRING      Output file name
  -s, --source=DOUBLE      Source position  (default='0.')
      --anglecut=DOUBLE    Cut parameter on the SD of proton angle. (default='3.')
      --energycut=DOUBLE   Cut parameter on the SD of proton energy. (default='3.')
```

Select those within `--anglecut` / `--energycut` σ around the mean.

```
  -r, --robust             Use robust estimation using 50/19.1 %ile.  (default=off)
```

Median and centiles are used to computes the means and σ.

```
      --robustopt=INT      Use newer options for robust cut.  (default='0')
      --primaries          Consider only primary protons  (default=off)
      --nonuclear          Consider only primary protons without nuclear interactions  (default=off)
      --wet                Write WET instead of initial energy  (default=off)

Projections parameters:
      --origin=DOUBLE      Origin (default=centered)
      --dimension=INT      Dimension(Deprecated) Use --size instead.  (default='256')
      --size=INT           Size  (default='256')
      --spacing=DOUBLE     Spacing  (default='1')
      --direction=DOUBLE   Direction
      --like=STRING        Copy information from this image (origin, size, spacing, direction)

Output cut images:
      --menergy=STRING     Mean energy file name
      --senergy=STRING     Sigma energy file name
      --sangle=STRING      Sigma angle file name
```

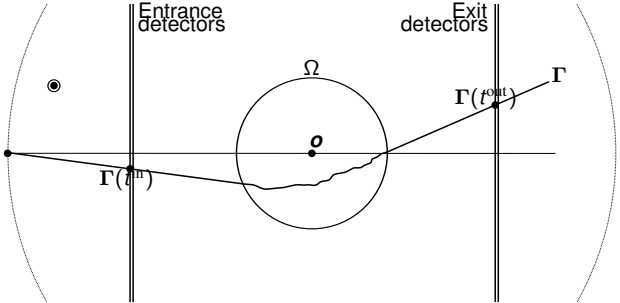
Outline

Full simulation and energy-loss reconstruction example

```
python protonct.py --output gate_simulation --verbose

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

Most likely path

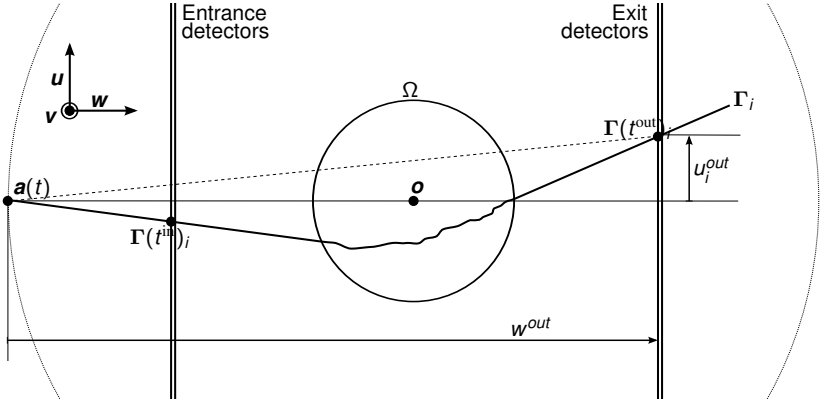
Estimates the proton path from available measurements, i.e., the positions $\mathbf{\Gamma}(t^{\text{in}})$ and $\mathbf{\Gamma}(t^{\text{out}})$, the directions $\dot{\mathbf{\Gamma}}(t^{\text{in}})$ and $\dot{\mathbf{\Gamma}}(t^{\text{out}})$, the energies E^{in} and E^{out}, the patient enveloppe Ω.



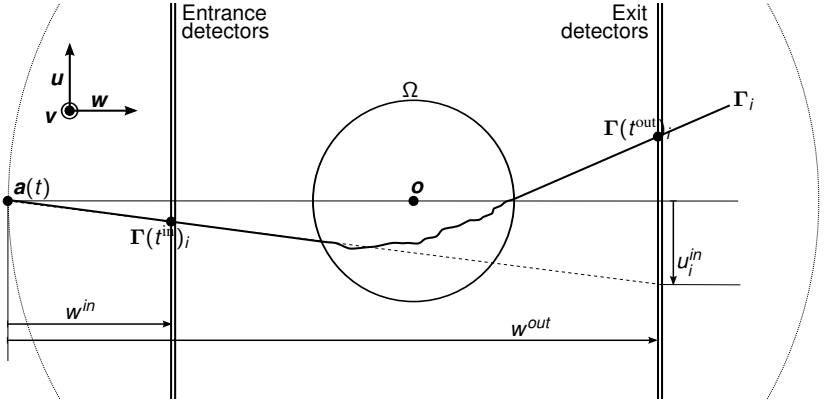
Most likely paths in PCT

- Third order polynomial
  T. Li, Z. Liang, J.V. Singanallur, T.J. Satogata, D.C. Williams, and R.W. Schulte. “Reconstruction for proton computed tomography by tracing proton trajectories: a Monte Carlo study”. In: *Med Phys* 33.3 (2006), pp. 699–706
- Bayesian approach assuming water
  R. W. Schulte, S. N. Penfold, J. T. Tafas, and K. E. Schubert. “A maximum likelihood proton path formalism for application in proton computed tomography.”. In: *Med Phys* 35.11 (2008), pp. 4849–4856. DOI: 10.1118/1.2986139
- Polynomial formulation according to the energy loss polynomial
  N. Krah, J.-M. Létang, and S. Rit. “Polynomial modelling of proton trajectories in homogeneous media for fast most likely path estimation and trajectory simulation.”. In: *Phys Med Biol* 64.19 (19 2019), p. 195014. DOI: 10.1088/1361-6560/ab3d0b
- Energy adaptive
  N. Krah, D. Dauvergne, J.M. Létang, S. Rit, and E. Testa. “Energy-adaptive calculation of the most likely path in proton CT”. In: *Phys Med Biol* 66.20NT02 (2021), pp. 1–6. DOI: 10.1088/1361-6560/ac2999

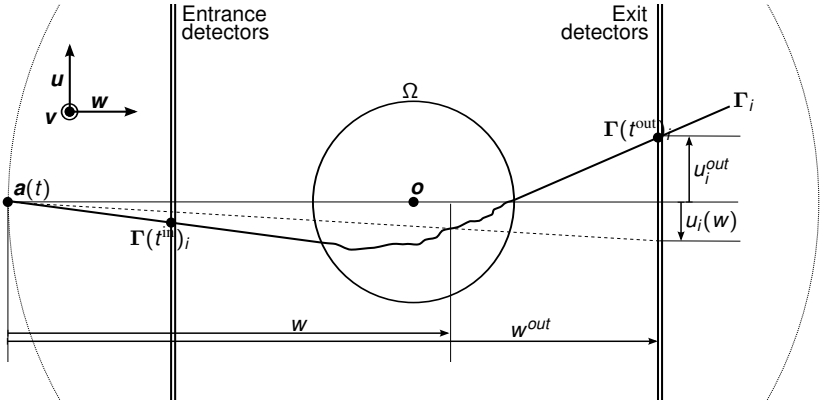
“Natural” binning



Binning driven by entrance position



Distance-driven binning



Distance-driven binning

- Name following

  B. De Man and S. Basu. “Distance-driven projection and backprojection in three dimensions”. In: *Phys Med Biol* 49.11 (2004), pp. 2463–2475

- In practice, binning is computed for several distances between the entrance and the exit detector.

$\Rightarrow$ 4D sinogram instead of a standard 3D sinogram

Illustration on one projection only

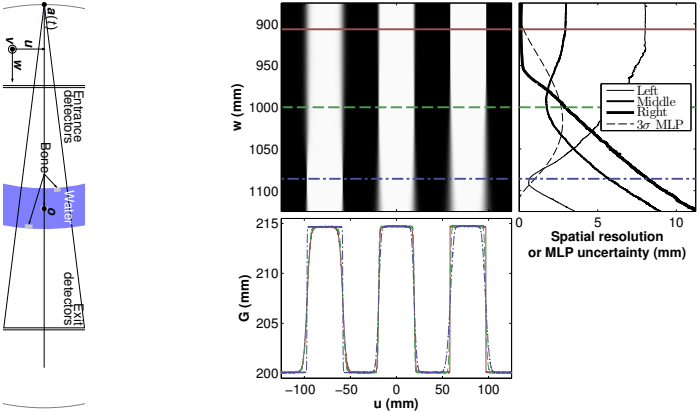


Figure from S. Rit, G. Dedes, N. Freud, D. Sarrut, and J.M. Létang. “Filtered backprojection proton CT reconstruction along most likely paths”. In: *Med Phys* 40.3, 031103 (2013), p. 031103. DOI: 10.1118/1.4789589

pctbinning --help

```
    --elosswepl=STRING        Output file name (alias for --output)
-c, --count=STRING            Image of count of proton pairs per pixel
    --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
    --noise=STRING            Image of WEPL variance per pixel
    --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                              Tracker resolution in mm
    --trackerspacing=DOUBLE   Tracker pair spacing in mm
    --materialbudget=DOUBLE   Material budget x/X0 of tracker
    --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING          Output file name (alias for --output)
```

S. Rit, G. Dedes, N. Freud, D. Sarrut, and J.M. Létang. “Filtered backprojection proton CT reconstruction along most likely paths”. In: *Med Phys* 40.3, 031103 (2013), p. 031103. DOI: 10.1118/1.4789589

```
-c, --count=STRING              Image of count of proton pairs per pixel
    --scatwepl=STRING           Image of scattering WEPL of proton pairs per pixel
    --noise=STRING              Image of WEPL variance per pixel
    --quadricIn=DOUBLE          Parameters of the entrance quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE         Parameters of the exit quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING            Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert          Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT            Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE             Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                                Tracker resolution in mm
    --trackerspacing=DOUBLE     Tracker pair spacing in mm
    --materialbudget=DOUBLE     Material budget x/X0 of tracker
    --fill                      Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING        Output file name (alias for --output)
-c, --count=STRING            Image of count of proton pairs per pixel
```

C.T. Quiñones, J.M. Létang, and S. Rit. "Filtered back-projection reconstruction for attenuation proton CT along most likely paths". In: *Phys Med Biol* 61.9 (2016), pp. 3258–3278. DOI: 10.1088/0031-9155/61/9/3258

```
    --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
    --noise=STRING            Image of WEPL variance per pixel
    --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                              Tracker resolution in mm
    --trackerspacing=DOUBLE   Tracker pair spacing in mm
    --materialbudget=DOUBLE   Material budget x/X0 of tracker
    --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING          Output file name (alias for --output)
-c, --count=STRING              Image of count of proton pairs per pixel
    --scatwepl=STRING           Image of scattering WEPL of proton pairs per pixel
```

N. Krah, C.T. Quiñones, J.M. Letang, and S. Rit. "Scattering proton CT". In: *Phys Med Biol* 65.22 (2020), p. 225015. DOI: 10.1088/1361-6560/abbd18

```
    --noise=STRING              Image of WEPL variance per pixel
    --quadricIn=DOUBLE          Parameters of the entrance quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE         Parameters of the exit quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING            Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert          Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT            Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE             Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                                Tracker resolution in mm
    --trackerspacing=DOUBLE     Tracker pair spacing in mm
    --materialbudget=DOUBLE     Material budget x/X0 of tracker
    --fill                      Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
      --elosswepl=STRING        Output file name (alias for --output)
  -c, --count=STRING            Image of count of proton pairs per pixel
      --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
      --noise=STRING            Image of WEPL variance per pixel
```

J. Dickmann, P. Wesp, M. Rädler, S. Rit, M. Pankuch, R.P. Johnson, V.A. Bashkirov, R.W. Schulte, K. Parodi, G. Landry, and G. Dedes. "Prediction of image noise contributions in proton computed tomography and comparison to measurements". In: *Phys Med Biol* 64.14 (2019), p. 145016. DOI: 10.1088/1361-6560/ab2474

```
      --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
      --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
      --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
      --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

      --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
      --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
      --trackerresolution=DOUBLE
                                Tracker resolution in mm
      --trackerspacing=DOUBLE   Tracker pair spacing in mm
      --materialbudget=DOUBLE   Material budget x/X0 of tracker
      --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING        Output file name (alias for --output)
-c, --count=STRING            Image of count of proton pairs per pixel
    --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
    --noise=STRING            Image of WEPL variance per pixel
    --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
```

R. W. Schulte, S. N. Penfold, J. T. Tafas, and K. E. Schubert. "A maximum likelihood proton path formalism for application in proton computed tomography.". In: *Med Phys* 35.11 (2008), pp. 4849–4856. DOI: 10.1118/1.2986139

```
    --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                              Tracker resolution in mm
    --trackerspacing=DOUBLE   Tracker pair spacing in mm
    --materialbudget=DOUBLE   Material budget x/X0 of tracker
    --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING          Output file name (alias for --output)
-c, --count=STRING              Image of count of proton pairs per pixel
    --scatwepl=STRING           Image of scattering WEPL of proton pairs per pixel
    --noise=STRING              Image of WEPL variance per pixel
    --quadricIn=DOUBLE          Parameters of the entrance quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE         Parameters of the exit quadric support function, see
                                  http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING            Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert          Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT            Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
```

N. Krah, J.-M. Létang, and S. Rit. "Polynomial modelling of proton trajectories in homogeneous media for fast most likely path estimation and trajectory simulation.". In: *Phys Med Biol* 64.19 (19 2019), p. 195014. DOI: 10.1088/1361-6560/ab3d0b

```
    --ionpot=DOUBLE             Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                                Tracker resolution in mm
    --trackerspacing=DOUBLE     Tracker pair spacing in mm
    --materialbudget=DOUBLE     Material budget x/X0 of tracker
    --fill                      Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING        Output file name (alias for --output)
-c, --count=STRING            Image of count of proton pairs per pixel
    --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
    --noise=STRING            Image of WEPL variance per pixel
    --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                              Tracker resolution in mm
    --trackerspacing=DOUBLE   Tracker pair spacing in mm
    --materialbudget=DOUBLE   Material budget x/X0 of tracker
```

N. Krah, F. Khellaf, J.M. Letang, S. Rit, and I. Rinaldi. "A comprehensive theoretical comparison of proton imaging set-ups in terms of spatial resolution.". In: *Phys Med Biol* 63.13 (13 2018), p. 135013. DOI: 10.1088/1361-6560/aaca1f

```
    --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

pctbinning --help

```
    --elosswepl=STRING        Output file name (alias for --output)
-c, --count=STRING            Image of count of proton pairs per pixel
    --scatwepl=STRING         Image of scattering WEPL of proton pairs per pixel
    --noise=STRING            Image of WEPL variance per pixel
    --quadricIn=DOUBLE        Parameters of the entrance quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --quadricOut=DOUBLE       Parameters of the exit quadric support function, see
                                http://education.siggraph.org/static/HyperGraph/raytrace/rtinter4.htm
    --mlptype=STRING          Type of most likely path (schulte, polynomial, or krah)  (default='schulte')
    --mlptrackeruncert        Consider tracker uncertainties in MLP [Krah 2018, PMB]  (default=off)

    --mlppolydeg=INT          Degree of the polynomial to approximate 1/beta^2p^2  (default='5')
    --ionpot=DOUBLE           Ionization potential used in the reconstruction in eV  (default='68.9984')
    --trackerresolution=DOUBLE
                              Tracker resolution in mm
    --trackerspacing=DOUBLE   Tracker pair spacing in mm
    --materialbudget=DOUBLE   Material budget x/X0 of tracker
    --fill                    Fill holes, i.e. pixels that were not hit by protons  (default=off)
```

D.C. Hansen, T. Sangild Sorensen, and S. Rit. "Fast reconstruction of low dose proton CT by sinogram interpolation". In: *Phys Med Biol* 61.15 (2016), pp. 5868–5882. DOI: 10.1088/0031-9155/61/15/5868

Outline

Full simulation and energy-loss reconstruction example

```
python protonct.py --output gate_simulation --verbose

pctpairprotons --input-in gate_simulation/PhaseSpaceIn.root --input-out gate_simulation/PhaseSpaceOut.root \
  --psin PhaseSpaceIn --psout PhaseSpaceOut --plane-in -110 --plane-out 110 --verbose \
  --output pairs.mhd

for i in {0000..0719}; do
  pctpaircuts --input pairs${i}.mha --output pairs_cut${i}.mha --spacing 2,50 --size 120,1 --robust
  pctbinning --input pairs_cut${i}.mha --output proj${i}.mha --spacing=1,50,1 --size=220,1,220 --source -1000.
done

rtksimulatedgeometry --output geometry.xml --nproj 720 --sid 1000 --sdd 1100

pctfdk --geometry geometry.xml --path . --regexp proj....\\.mha --output recon.mha --dimension=210,1,210 -v
```

Reconstruction algorithm

Adaptation of the FDK algorithm in
`pct::FDKDDConeBeamReconstructionFilter`

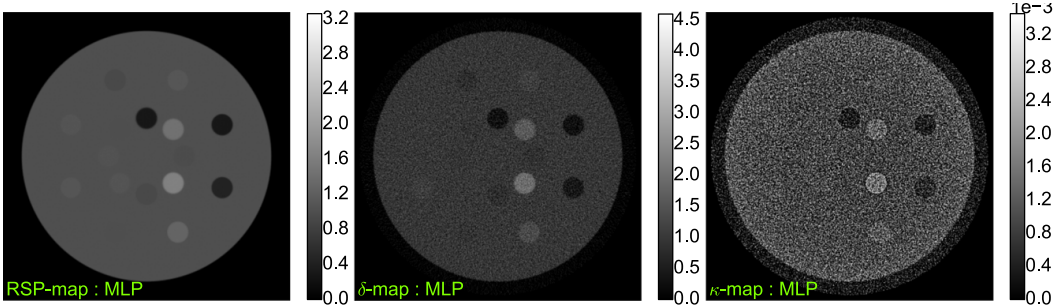
- Same 2D processing (weighting and filtering) at each distance
  `pct::FDKDDWeightProjectionFilter` and
  `rtk::FFTRampImageFilter`

- Rotate and add (+account for divergence if not parallel)

$\Rightarrow$ Each proton information is backprojected along its most likely path
`pct::FDKDDBackProjectionImageFilter`

Results

N. Krah, C.T. Quiñones, J.M. Letang, and S. Rit. “Scattering proton CT”. In: *Phys Med Biol* 65.22 (2020), p. 225015. DOI: 10.1088/1361-6560/abbd18



pctfdk --help

```
  -h, --help              Print help and exit
  -V, --version           Print version and exit
  -v, --verbose           Verbose execution  (default=off)
      --config=STRING     Config file
  -g, --geometry=STRING   XML geometry file name
  -p, --path=STRING       Path containing projections
  -r, --regexp=STRING     Regular expression to select projection files in path
  -o, --output=STRING     Output file name
  -l, --lowmem            Load only one projection per thread in memory
                            (default=off)
      --wpc=DOUBLE        Water precorrection coefficients (default is no
                            correction)

Ramp filter:
      --pad=DOUBLE        Data padding parameter to correct for truncation
                            (default='0.0')
      --hann=DOUBLE       Cut frequency for hann window in ]0,1] (0.0 disables
                            it)  (default='0.0')
     --hannY=DOUBLE      Cut frequency for hann window in ]0,1] (0.0 disables
                            it)  (default='0.0')

Volume properties:
      --origin=DOUBLE     Origin (default=centered)
      --dimension=INT     Dimension(Deprecated) Use --size instead.
                            (default='256')
      --size=INT          Size  (default='256')
      --spacing=DOUBLE    Spacing  (default='1')
      --direction=DOUBLE  Direction
      --like=STRING       Copy information from this image (origin, size,
```

pctfdk --help

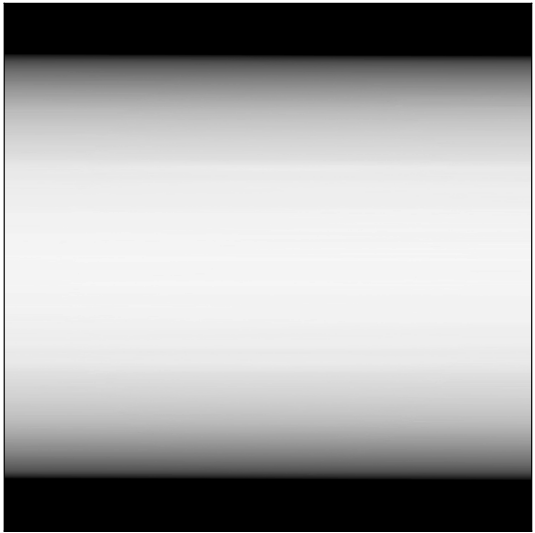
```
  -h, --help                Print help and exit
  -V, --version             Print version and exit
  -v, --verbose             Verbose execution  (default=off)
      --config=STRING       Config file
  -g, --geometry=STRING     XML geometry file name
  -p, --path=STRING         Path containing projections
  -r, --regexp=STRING       Regular expression to select projection files in path
  -o, --output=STRING       Output file name
  -l, --lowmem              Load only one projection per thread in memory
                              (default=off)
      --wpc=DOUBLE          Water precorrection coefficients (default is no
                              correction)

Ramp filter:
      --pad=DOUBLE          Data padding parameter to correct for truncation
                              (default='0.0')
      --hann=DOUBLE         Cut frequency for hann window in ]0,1] (0.0 disables
                              it)  (default='0.0')
```

E. Fogazzi, D. Trevisan, P. Farace, R. Righetto, S. Rit, M. Scaringella, M. Bruzzi, F. Tommasino, and C. Civinini. "Characterization of the INFN proton CT scanner for cross-calibration of x-ray CT". In: *Phys Med Biol* 68.12 (2023), p. 124001. DOI: 10.1088/1361-6560/acd6d3

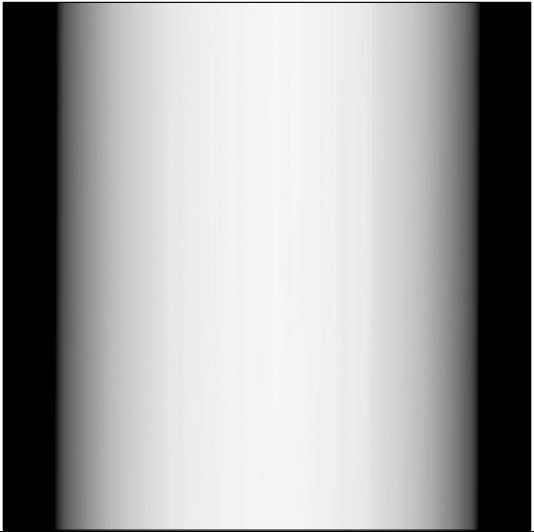
```
      --hannY=DOUBLE        Cut frequency for hann window in ]0,1] (0.0 disables
                              it)  (default='0.0')

Volume properties:
      --origin=DOUBLE       Origin (default=centered)
      --dimension=INT       Dimension(Deprecated) Use --size instead.
                              (default='256')
```

Outline

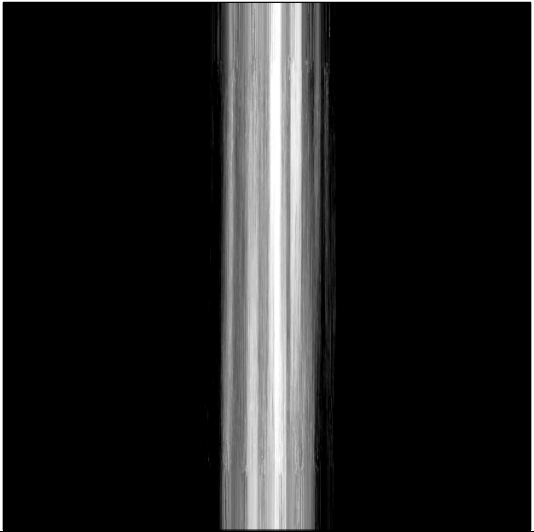
Pixel-specific backprojection



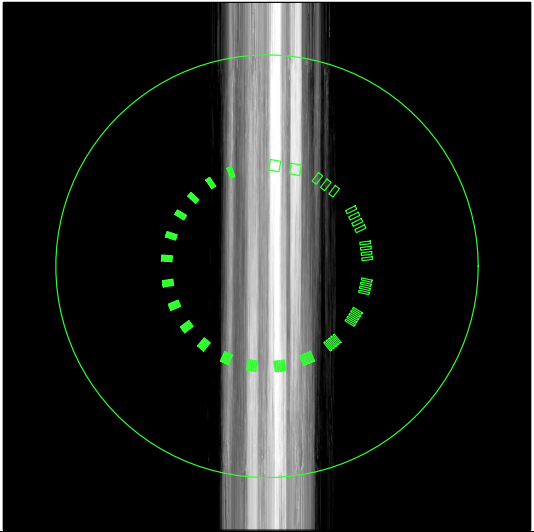
Pixel-specific backprojection



Pixel-specific backprojection

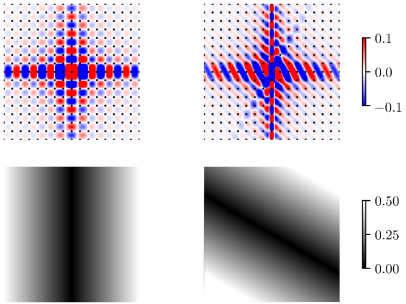


Pixel-specific backprojection



Directional ramp filter

F. Khellaf, N. Krah, J.M. Létang, and S. Rit. "2D directional ramp filter". In: *Phys Med Biol* 65.8 (2020), 08NT01.
DOI: 10.1088/1361-6560/ab7875



Code not in PCT (yet)...

Outline

Backprojection then filtering

S. Rit, R Clackdoyle, J. Hoskovec, and J.M. Létang. "List-mode proton CT reconstruction using their most likely paths via the finite Hilbert transform of the derivative of the backprojection". In: *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*. Newport, USA, 2015, pp. 324–327

- Based on Noo's two-step Hilbert transform method and Zeng's adaptation to backproject first

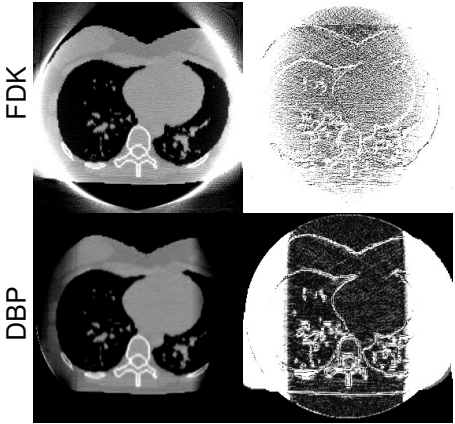
  F. Noo, R. Clackdoyle, and J.D. Pack. "A two-step Hilbert transform method for 2D image reconstruction". In: *Phys Med Biol* 49.17 (2004), pp. 3903–3923. DOI: 10.1088/0031-9155/49/17/006

  G.L. Zeng. "Image reconstruction via the finite Hilbert transform of the derivative of the backprojection.". In: *Med Phys* 34.7 (2007), pp. 2837–2843

- Bonus: region-of-interest reconstruction.

The second step, the inverse Hilbert transform, is not in PCT yet.

Backprojection then filtering



[6] S. Rit, R Clackdoyle, J. Hoskovec, and J.M. Létang. "List-mode proton CT reconstruction using their most likely paths via the finite Hilbert transform of the derivative of the backprojection". In: *Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*. Newport, USA, 2015, pp. 324–327.

Conclusions

- Overwiew of our open-source reconstruction package PCT
- Already available but

  - `pctpaircuts` must be translated to Python
  - Finalize and document full attenuation, scatter pipelines
  - Idem for variance and He CT in collaboration with LMU
- Will advertise the release on users@ionimaging.org
- We welcome any feedback and contribution
  - What data format are you using? What does it store?
  - Any alternative open source software available?