Unified data format

From pCT wiki

To facilitate sharing of data from different sources (simulation, experimental), we propose a **unified data format** for proton histories to be used by all reconstruction software.

Contents

Version specifications

Since the first header field is a version identifier, the data format can be defined in different ways and this header field will tell the data parser how to handle it. This section will serve as a log of all format versions.

Version 0

Implemented in svn/sim/branches/ford_ideal and svn/recon/branches/ford_gen2

Contains the following headers:

- Magic number identifier: "PCTD" (4-byte string)
- Format version identifier (integer)
- Number of events in file (integer)
- Projection angle (float | degrees)
- Beam energy (float | MeV)
- Acquisition/generation date (integer | Unix time (http://en.wikipedia.org/wiki/Unix_time))
- Pre-process date (integer | Unix time)
- Phantom name or description (variable length string)
- Data source (variable length string)
- Prepared by (variable length string)

** Note on variable length strings: each variable length string should be preceded with an integer containing the number of characters in the string.*

Event data:

Data is be stored with all of one type in a consecutive row, meaning the first entries will be N t0 values, where N is the number of events in the file. Next will be N t1 values, etc. This more closely matches the data structure in memory.

- Detector coordinates in mm relative to a phantom center, given in the detector coordinate system:
    - t0 (float * N)
    - t1 (float * N)
    - t2 (float * N)
    - t3 (float * N)
    - v0 (float * N)
    - v1 (float * N)
    - v2 (float * N)
    - v3 (float * N)
    - u0 (float * N)
    - u1 (float * N)
    - u2 (float * N)
    - u3 (float * N)
- WEPL in mm (float * N)

Version 1

To be implemented in 2014 MC and preprocessing software as input for Blake's reconstruction code.

Contains the following headers:

- Magic number identifier: "PCTD" (4-byte string)
- Format version identifier (integer, 1)
- Run number (integer)
- Number of events in file N (integer)
- Projection angle (float | degrees)
- U coordinates of tracker planes (4 * float | mm)
- Beam energy (float | MeV)
- Acquisition/generation date (integer | Unix time)
- Pre-process date (integer | Unix time)
- Phantom name or description (variable length string)
- Data source (variable length string)
- Prepared by (variable length string)

* *Note on variable length strings: each variable length string should be preceded with an integer containing the number of characters in the string. Strings should be ASCII (1 byte per character, 0x00 - 0x7F)*

Event data:

Data is to be stored with all of one type in a consecutive row, meaning the first entries will be N t0 values, where N is the number of events in the file. Next will be N t1 values, etc. This more closely matches the data structure in memory. U coordinates are constant for the run and stored in the header as well as projection angle.

- Detector coordinates in multiples of 10um relative to a phantom center (–327.68mm to 327.67mm

  range), given in the detector coordinate system:
    - event number (_int32 * N)
    - t0 (_int16 * N)
    - t1 (_int16 * N)
    - t2 (_int16 * N)
    - t3 (_int16 * N)
    - v0 (_int16 * N)
    - v1 (_int16 * N)
    - v2 (_int16 * N)
    - v3 (_int16 * N)
- WEPL in 10um (_int16 * N)

General Notes and Ideas

Text files are unnecessarily large but inherently more simple. Adding compression will reduce size, but add processing time to file creation and reading. Is there a simple, portable binary format that would work well? ROOT trees?

Headers

All data files should contain metadata with the following information:

- Format version identifier
- Phantom name or description of simulation
- Acquisition/generation date
- Pre-process date
- Source (e.g., 'geant4' or 'phase1 scanner')
- Prepared by (e.g., 'Ford Hurley')
- Optional: source code used to generate the data

Ideally, these headers would be plain text and human readable even if the event data is in a binary format.

Previous event formats

In general, an event should be 4 detector coordinate triplets (x,y,z), a WEPL or energy value, and a rotation angle.

Geant4 text file example:

```
x1      y1      x2      y2      x3     y3    x4     y4   total_E  angle
1.432 -0.0114 4.8064 -1.2882 1.6144 0.1026 1.5916 0.2166 152.971   12
```

For a 5-stage detector this could be:

```
x1      y1      x2      y2      x3     y3    x4     y4   e[0]   e[1]    e[2]   e[3]  e[4]   angle
1.432 -0.0114 4.8064 -1.2882 1.6144 0.1026 1.5916 0.2166 22.3   22.8     18    43    23      12
```

LLU text example:

```
x1        y1         z1 x2        y2         z2 x3        y3         z3 x4        y4         z4 WEPL       angle
12.066303 -43.179974 0  13.572552 -45.119671 2  16.704826 -53.285725 4  18.318504 -58.259033 6  125.922112 0
```

LLU binary example:

```
struct s_event_data {
   float x_pos[4];
   float y_pos[4];
   char  z_pos[4];
   float wepl;
   float rotation;
};
```

Where for the LLU examples the z-coordinates are indices for a look up table. All coordinates are relative to a point on the rotation access, and in mm. WEPL is in mm, and angle is in degrees.

One idea to avoid projection angle issues is to generalize track coordinates to the phantom coordinate system, and do not report rotation angle at all. This would future-proof the format for more general detector and patient positioning. For example, for potentially changing z-coordinates (to minimize detector to patient distance as a function of angle), or for a non-rigid but optically tracked detector system which flexes slightly as it rotates around the gantry.

Other types of data

The most important type of data to put into a unified format are proton histories for reconstruction. There are other data types that may be worthwhile coming up with some specifications for data format.

1. Raw (detector) data. By definition, these will not be "unified", since there are several different types of detectors, including simulated ones, but we can define a common data wrapper for this type.
2. Pre-processed data. This is currently what is fed into reconstruction, and is the most important type to get "unified."
3. Pre-processed data in A matrix format. Instead of track hits in 4 locations, this data included the proton path through the imaging subject.
4. Reconstructed image data.

Retrieved from "http://scipp2.ucsc.edu/wiki/index.php/Unified_data_format"

- This page was last modified on 29 January 2014, at 15:15.
- This page has been accessed 189 times.