FDIRC SIMULATION TOOLS: PREPARING FOR THE PROTOTYPE

Doug Roberts
University of Maryland

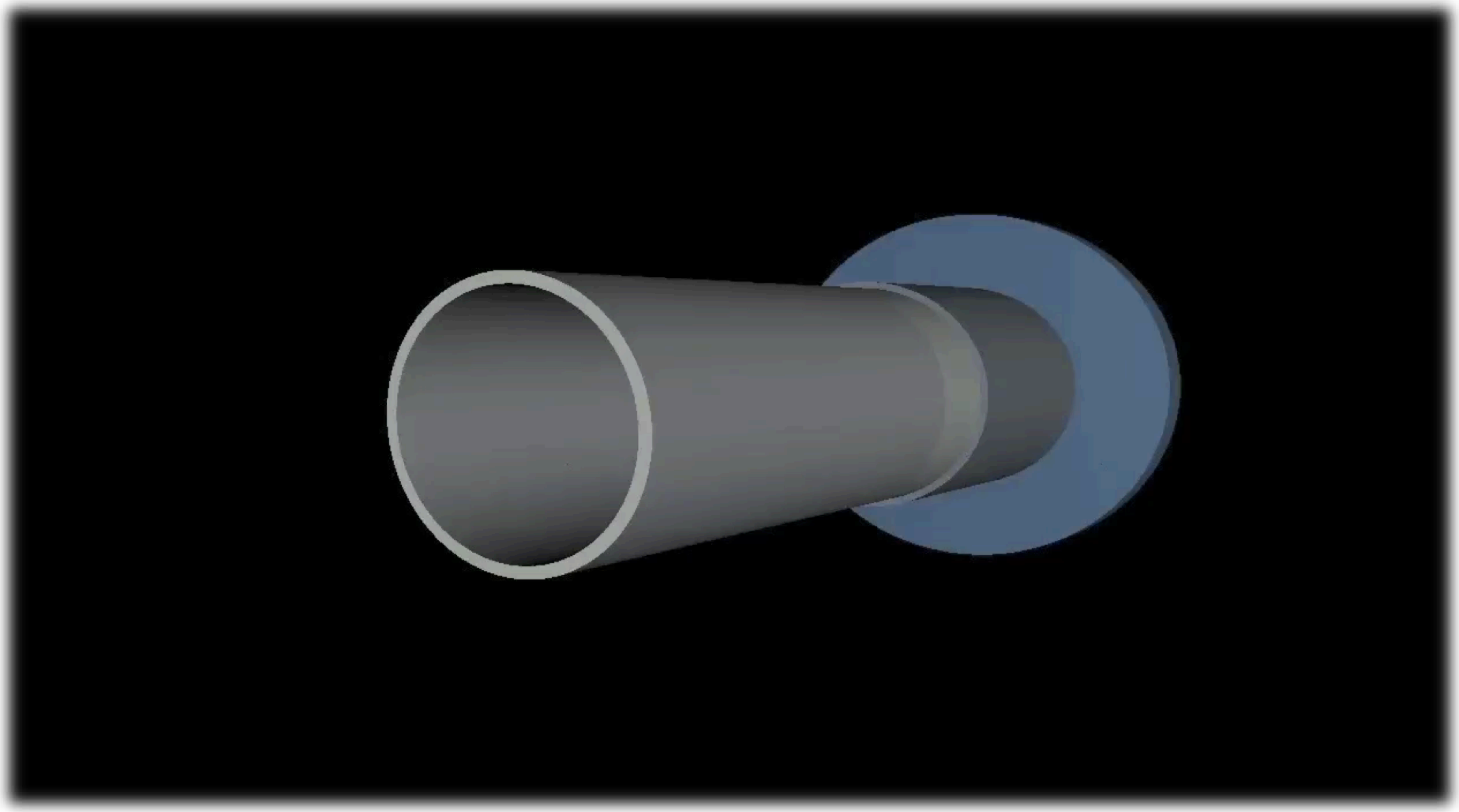
Overview

- This presentation is just meant as a summary and status of the current FDIRC Geant4-based simulation and to begin thinking about what can be used for the full prototype and what still needs to be done
- Simulation Status
  - How to get it
  - Options
  - Suitability for Prototype
- Analysis Tools
  - Current Method
  - What should we be doing, and how to organize

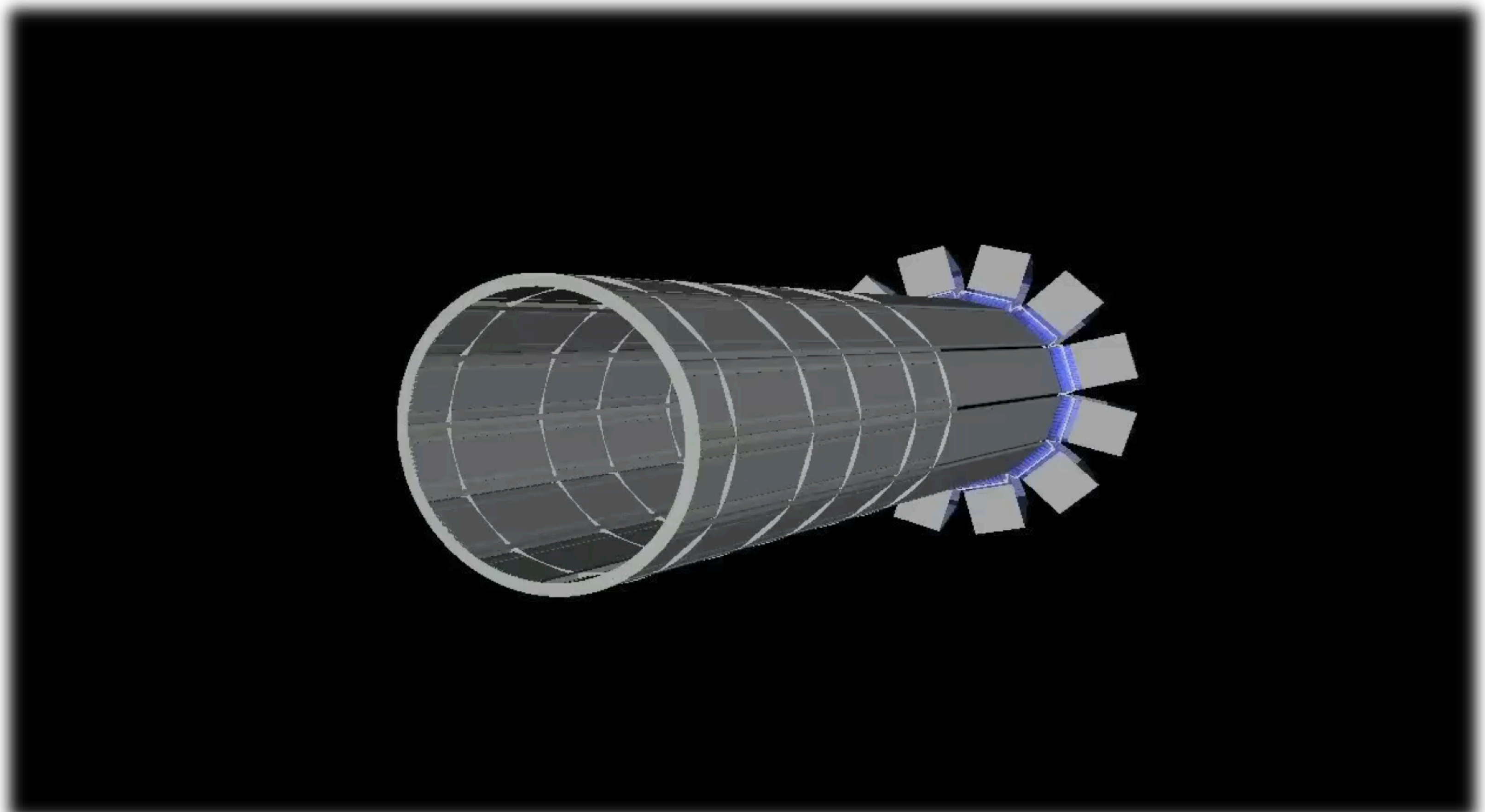
Simulation Status

- We have a Geant4-base standalone simulation of the FDIRC
- It is in the SuperB svn system:
  - Repository: FDIRC
  - Package: fDircG4
- This package has been used to study resolution vs. geometry and other options
- Has also been used to create the gdml file for BRUNO
- The model allows for creation and tracking of optical (Cerenkov) photons through the active materials of the detector
- Also includes all of the support structure for the bar boxes.
  - Support around FBLOCK itself is not really correct or very detailed

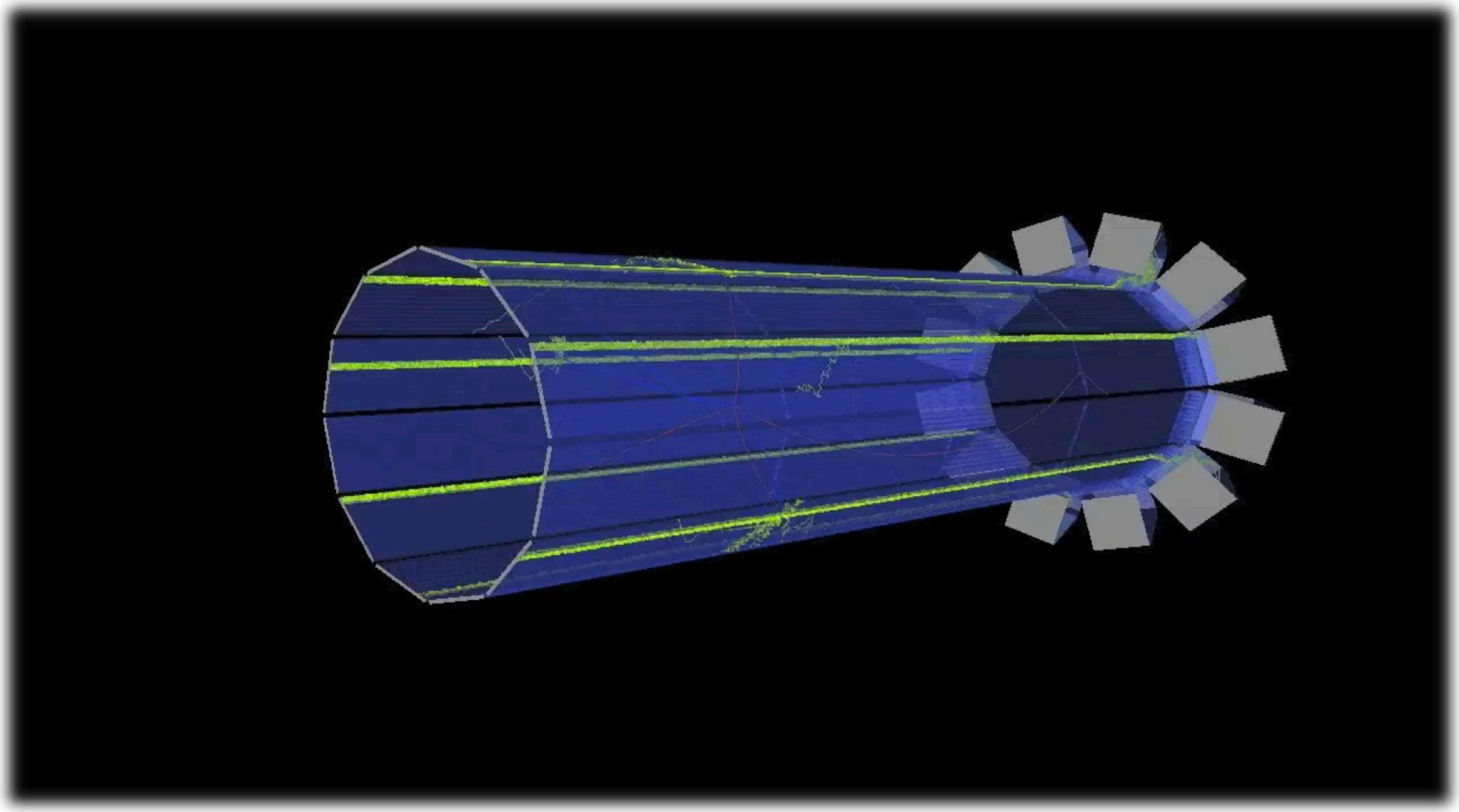
FDIRC in Bruno (all support)



No Outer Skins



An Event



Options

- Can be run either as full detector (all 12 sectors) or single sector
  - Single sector is what prototype will be
- Magnetic field on or off
- Single particle, di-muon-like distribution, sample of events, single photon
  - Like the di-muon, we could generate muons from a sample distribution for the CRT
- All geometry constants are contained in a single file
- For prototype, we can use a single sector, keep bar box structure but turn off other support, turn off magnetic field
- Bottom Line: I think this will be very easy to adapt to the CRT prototype

Simulation Output

- Simulation is (mostly) decoupled from specific photodetector choice or layout
  - One caveat: simulation can use QE of tube to speed up simulation.
    - Not just QE, but QE * charge collection efficiency
  - If we have mixed assortment of detectors on prototype, we will have to be more clever
- Output is basically a ROOT file with lots of information, most of which we won’t need
- What we do need:
  - Generated event info (track p, direction, position, time, and same info as track enters quartz bar…)
  - Generated optical photon info (wavelength, time, direction…)
  - Detected photon info (wavelength, time, position on focal plane…)

Analysis Tools

- Currently, all of my analysis has been done within ROOT
  - Not sure if this is the future plan?
- All of the ROOT scripts are in the fDircG4 package ("macros" directory), but there is little if any documentation at the moment
- Multi-step process
  - Simulate Events
  - Generate single-photon dictionary
  - Process single-photon dictionary
    - Automatically done the first time you run the resolution calculation macro if the photon dictionary root file doesn't exist yet
  - Correlate simulated events with dictionary
  - Analysis

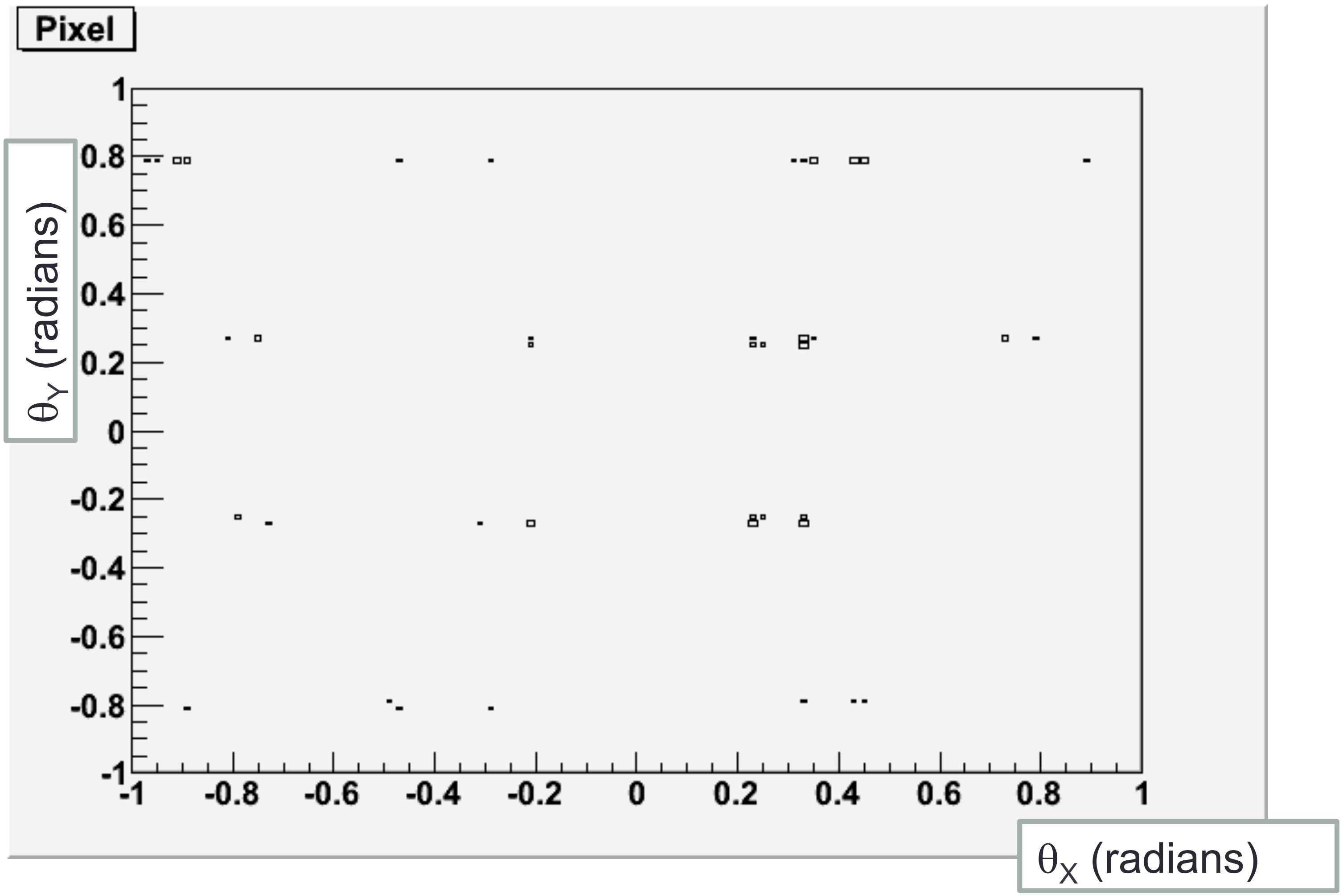
Analysis Process

- Simulate Events
  - Basic output is time and position of photon hits at the focal plane of the FBLOCK.
  - Position is in local coordinates of the focal plane
    - Be careful of a funny sign flip! (Do we have a coordinate sign convention, number convention…?)
  - Also keep original track information
- Single Photon Simulation
  - Fixed wavelength (410 nm)
  - Isotropic angular distribution
  - Randomized over x, y, z in bar
  - Randomized over 12 bars in a bar box
  - Same output as other events
  - Typically using a sample of 20,000,000 generated photons
    - Need new sample each time we change the geometry

Process Single Photon Dictionary

- Single Photon Dictionary is used to create a “map” from PMT pixel number to photon angles at the exit of the last quartz bar. Map also includes expected time propagation.
- Multi-valued map. Depends on:
  - Bar hit (use track information)
  - Path photon takes
    - Lots of different paths in FBLOCK can map from a bar to a given pixel
- Processing involves clustering dictionary:
  - Within a given pixel, try to reduce the number of dictionary entries by performing a nearest-neighbor clustering algorithm
  - Works in (θ_x, θ_y, t) space with some nominal resolution in these variables.
  - Recursive algorithm with a cutoff based on “χ^2” in (θ_x, θ_y, t) space.
  - Persist clustered dictionary
  - Uses a “PixelMapper” function to go from focal plane coordinates to PMT pixel number
    - Includes packing efficiency, dead space around edge of tube
    - Would have to make a version for prototype

Photon Dictionary, Edge Pixel, Bar 6



Pre-Clustering. Note near symmetry in $|\theta_x|$ and $|\theta_y|$

Calculate Possible θ_C Values

- For each photon hit, we calculate possible θ_C values with respect to the track
- Several sources of ambiguity to try to go back to photon's original direction:
  - Left-Right
  - Up-Down
  - Forward-Backward
  - Path in FBLOCK
    - Actually, path after leaving last quartz bar. Includes bounces in wedges.
- Therefore, each photon can have many different θ_C "solutions"
  - Time resolves forward-backward in most cases
  - Many solutions are non-physical and are discarded
  - But still left with multiple solutions
- Even with multiple solutions, idea is that correct one will be common to all photons from the track

Running in ROOT

- From ROOT in “macros” directory:
  - run “LoadMacros.C” (root[0] .x LoadMacros.C)
  - Execute “run” method of DoRes.C
    - This needs some work to be generalized. Ideas welcome!
  - This will load the relevant SinglePhoton dictionary or create it if it doesn’t exist.
  - Determines possible θ_C solutions for each photon
  - Output will be a new ROOT file that contains all solutions for each photon in TTree “ResCalcTree”
- Macro “AngleRes.C” has some examples of making plots from this TTree

- I make no guarantees that this will just run OOTB!

What Next?

- This code was primarily written for my use, and therefore not very user-friendly, or clean, or documented.
- I have no attachment to this code, and would be happy to see it all scrapped. If I had to do it all over again, I would probably do some things differently based on what I've learned.
- Would be nice to have some organized plan to build some code and structure that would be useful and easy to use for the collaboration.
- I'd be more than happy to work on this with others.