

Status Report

Shota Izumiyama

11 Oct. 2019

mPMT-Japan meeting

Simulation condition (remind)

- Generated electron events by WCSim
- Detector: Hyper-K with 20" PMTs and mPMTs
 - Photo coverage: 20inch PMT = 20 %, mPMT = 10 %
- Particle gun:
 - Particle generator: gps
 - 10,000 events of 10 MeV e-
 - Uniform in detector
 -
- Reconstruction: Benjamin-san's code
 - Not Bonsai

New

- Turned on masking mPMT
 - Benjamin-san suggested that vertex resolution using only B&L PMTs got better than one with mPMT in my output files.
- Changed calculation of vertex resolution
 - Used events whose distance between rec. and mc. is < 300 cm
 - Vertex res. $\rightarrow$ about 70 cm
- Goal
 - Reproduce Benjamin-san's result to validate the fitting code and mc behaviour

Reconstruction w/o mPMTs

- Turned on masking mPMT to check performance w/o mPMTs
 - `maskPMT = true`
 - `pmtTypeMasked=1;`
 - `maskedProbability=1.1` // all mPMT are masked
- Benjamin-san suggested that vertex resolution using only B&L PMTs got better than one with mPMT in my output files.

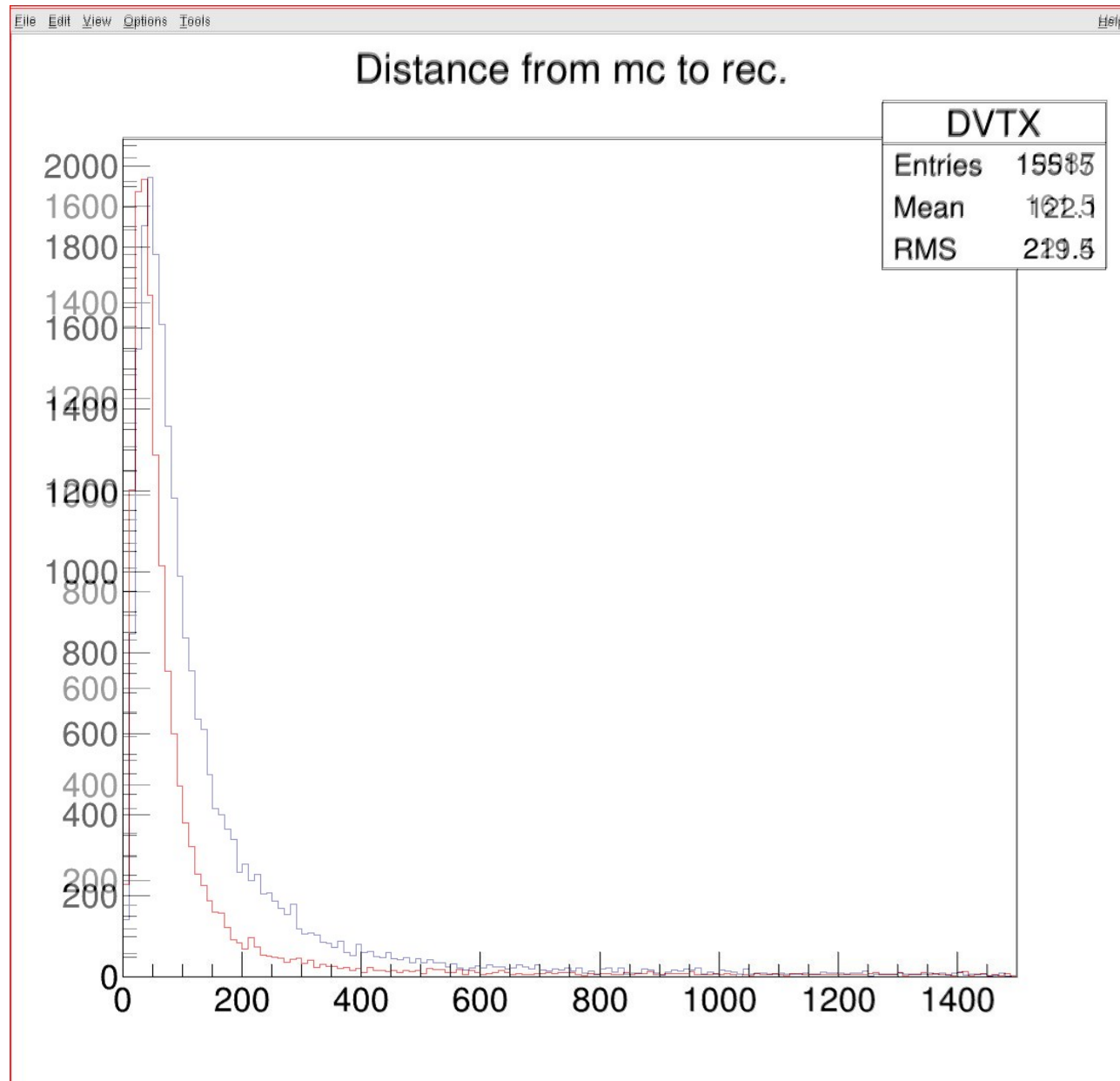
Reconstruction w/o mPMTs (cont.)

Red: w/ mPMT

Blue: w/o mPMTs

Red has better
vertex res.

At last meeting,
Benjamin-san said
w/o mPMT get
better. This result
looks strange.



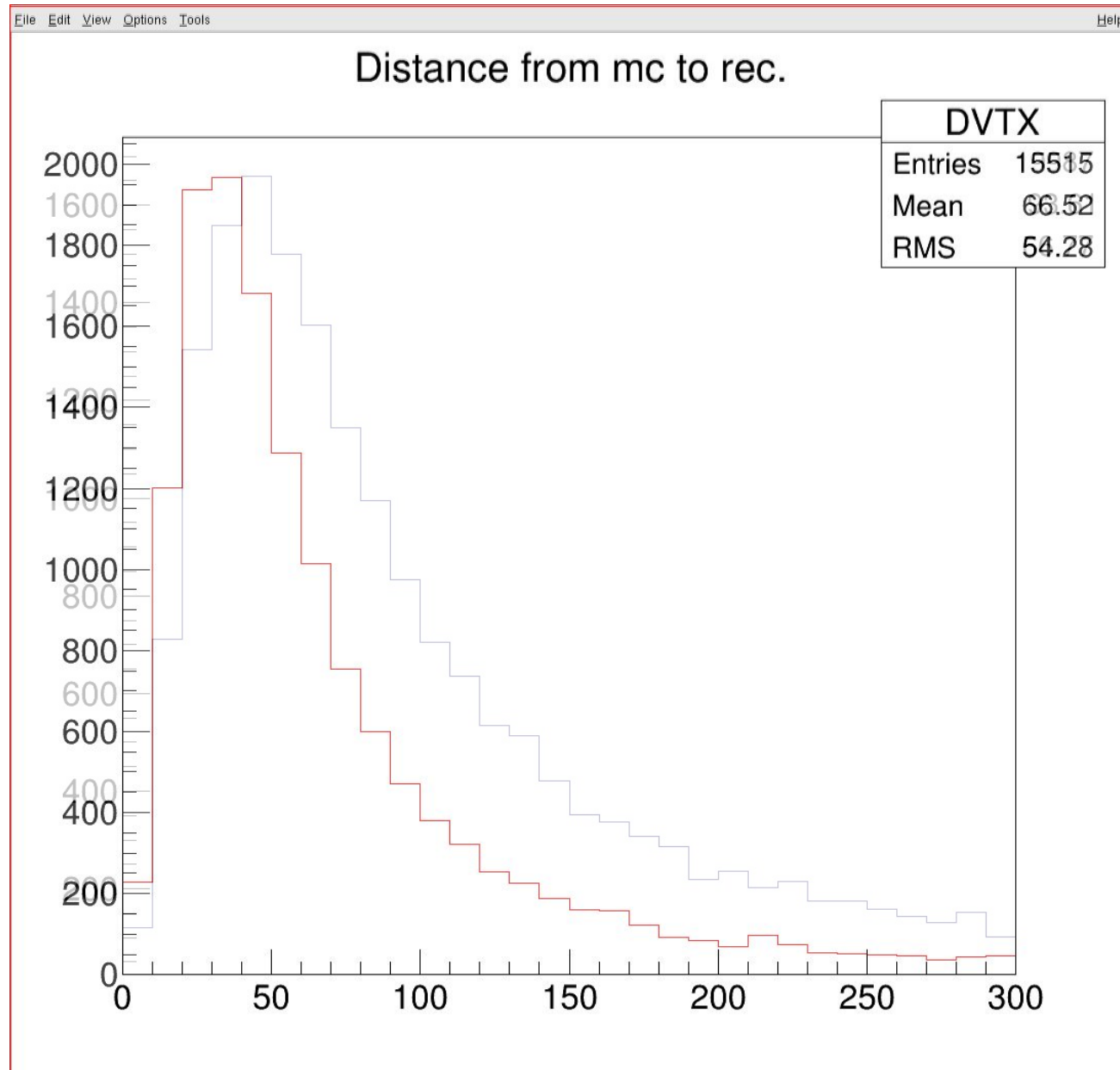
Reconstruction w/o mPMTs (cont.)

Red: w/ mPMT

Blue: w/o mPMTs

Red has better
vertex res.

At last meeting,
Benjamin-san said
w/o mPMT get
better. This result
looks strange.



Reconstruction w/o mPMTs (cont.)

Red: w/ mPMT

Blue: w/o mPMTs

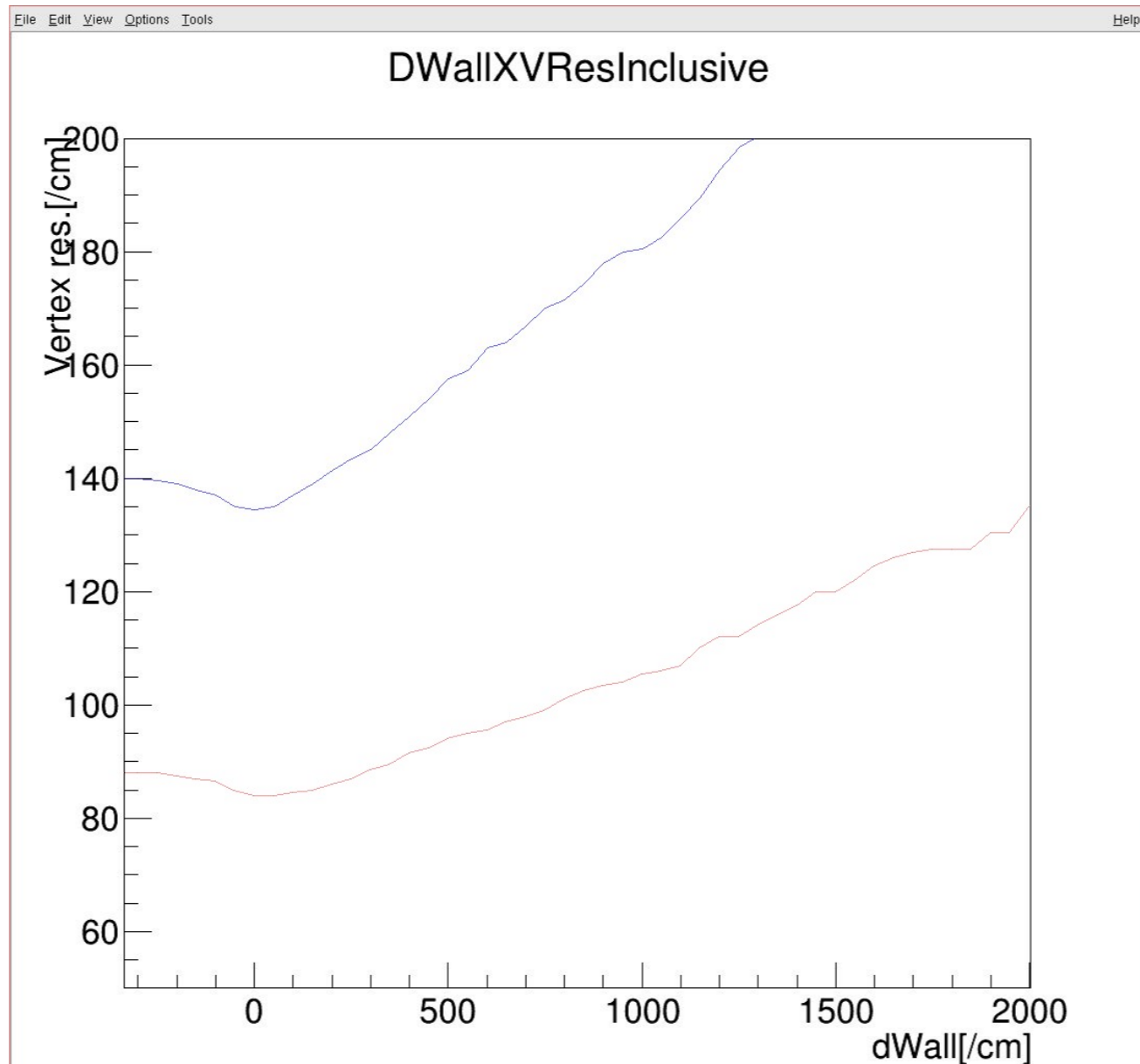
Vertex res.

Red: ~ 90 cm

Blue: ~ 140 cm

Vertex res. w/o
mPMTs got worse.

timePDF for
mPMTs are
working
correctly..?

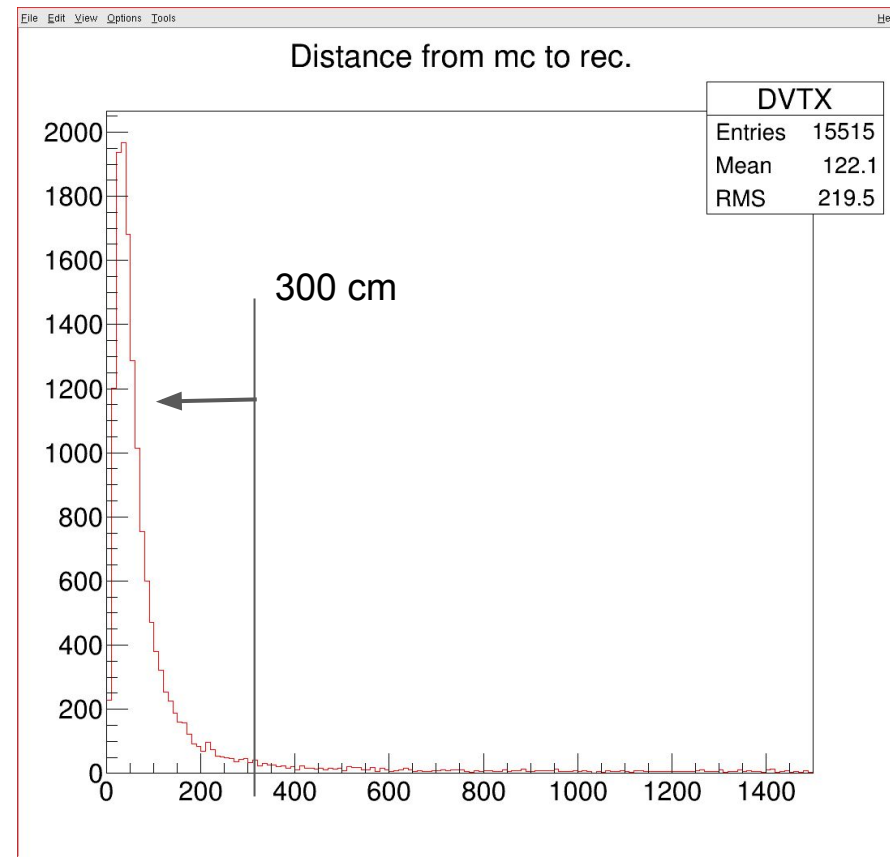


Changed vertex resolution

- Changed vertex resolution

- Previous: vertex resolution is defined as distance between rec. and mc. which has 68.2% of all reconstructed events
- → Today: defined as the distance which has 68.2% of reconstructed events having the distance < 300 cm

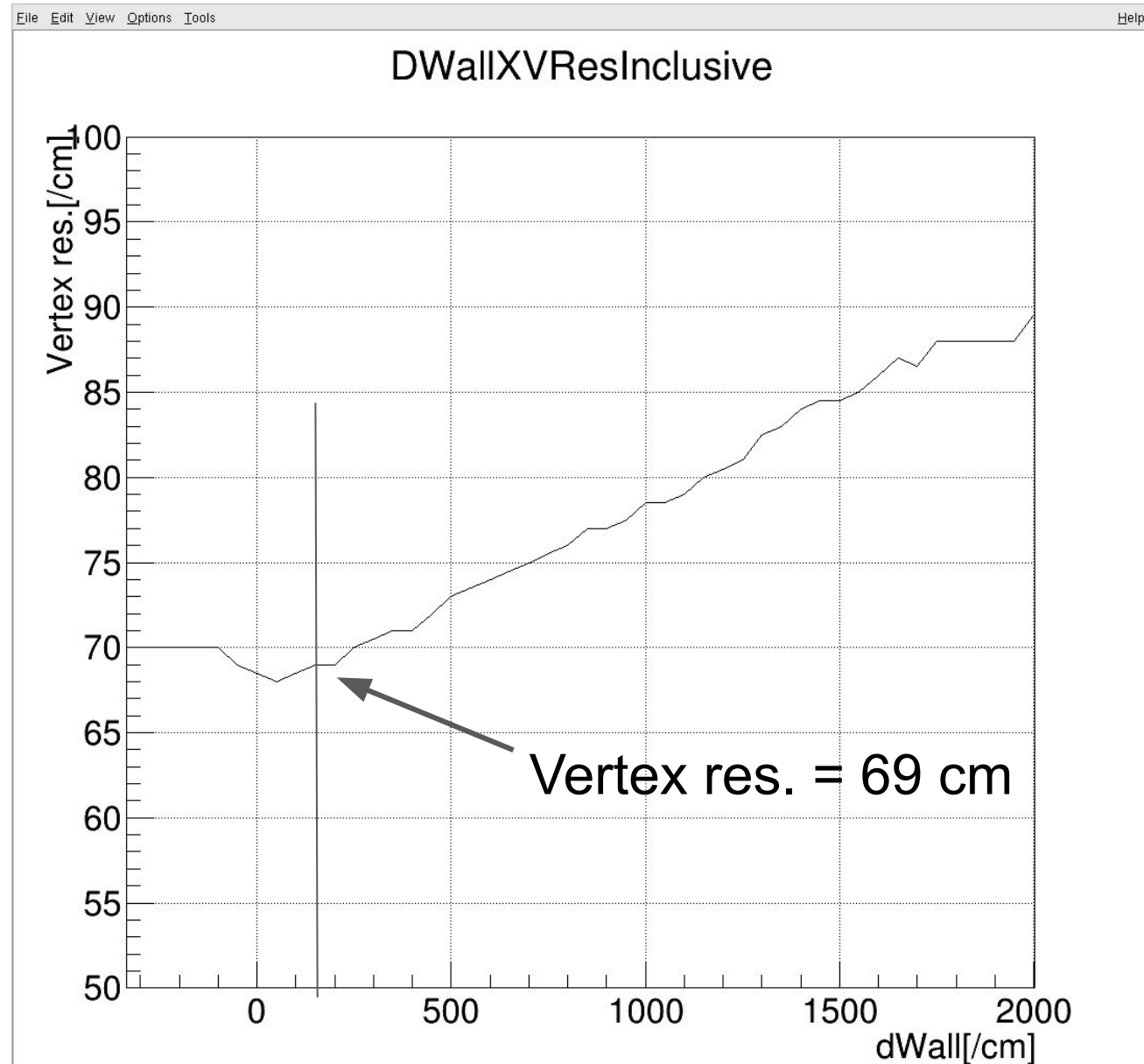
-



Changed vertex resolution (cont.)

Vertex res. is
calculated with
events of $> d_{Wall}$

Vertex resolution
for $d_{Wall} > 150$
cm is about 69 cm

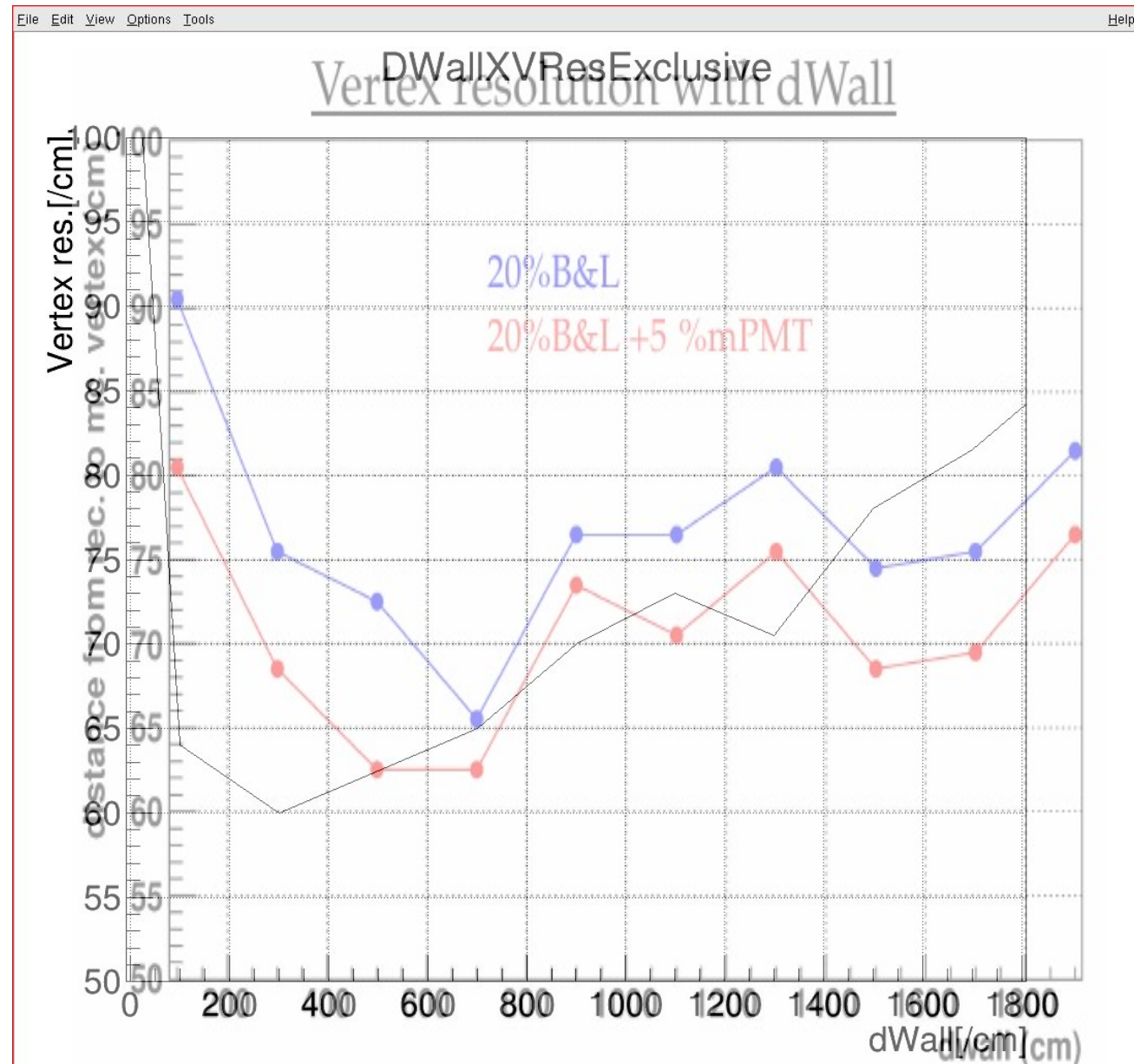


Changed vertex resolution (cont.)

Vertex res. is
calculated with
events of dWall in
 ± 100 cm

Overlaid
Benjamin-san's
plot

It looks
compatible?



Fraction of miss-reconstruction

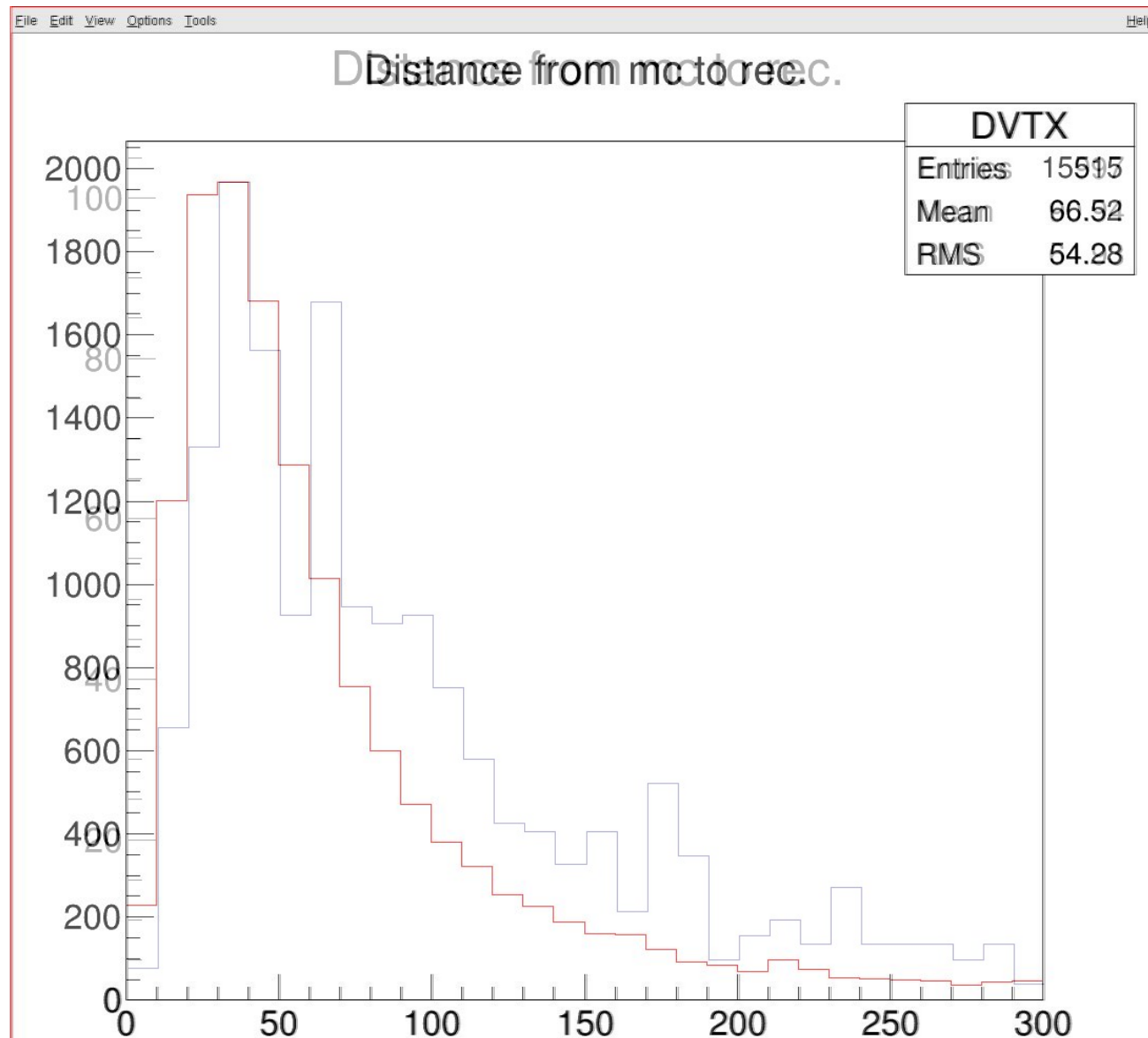
- Calculated fraction f
 - $f = (\text{\# of events reconstructed inner, reconstructed dWall} > 1000 \text{ cm}) / (\text{\# of events generated near edge, } 0 \text{ cm} < \text{true dWall} < 500 \text{ cm})$
- For this MC (see page 2), $f \sim 0.02$

Benjamin-san's output of WCSim

I reconstructed
using
Benjamin-san's
WCSim output file

Blue:
Benjamin-san
Red: me (same
with red of p.6)

Reconstruction
code seems to
have problem
also?



WCSim PMT TTS

- I checked “WCSimPMTObject.cc”
 - B&L PMT: TTS sigma ~ 0.955 ns (for $Q=1$)
 - 3inch R14374: TTS sigma ~ 0.6 ns

```

1767 float BoxandLine20inchHQE::HitTimeSmearing(float Q) {
1768     G4float sig_param[4]={0.6314,0.06260,0.5711,23.96};
1769     G4float lambda_param[2]={0.4113,0.07827};
1770     G4float sigma_lowcharge = sig_param[0]*(exp(-sig_param[1]*Q)+sig_param[2]);
1771
1772     G4float highcharge_param[2];
1773     highcharge_param[0]=2*sig_param[0]*sig_param[1]*sig_param[3]*sqrt(sig_param[3])*exp(-sig_param[1]*sig_param[3])
;
1774     highcharge_param[1]=sig_param[0]*((1-2*sig_param[1]*sig_param[3])*exp(-sig_param[1]*sig_param[3])+sig_param[2])
;
1775     G4float sigma_highcharge = highcharge_param[0]/sqrt(Q)+highcharge_param[1];
1776
1777     G4float sigma = sigma_lowcharge*(Q<sig_param[3])+sigma_highcharge*(Q>sig_param[3]);
1778     G4float lambda = lambda_param[0]+lambda_param[1]*Q;
1779     G4float Smearing_factor = G4RandGauss::shoot(-0.2,sigma)-1/lambda*log(1-G4UniformRand());
1780     return Smearing_factor;
1781 }

```

```

float PMT3inchR14374::HitTimeSmearing(float Q) {

```

```

//float timingConstant = 0.75;//B. Quilain, to match the TTS = 1.4ns (sigma at 0.6) measured at 1 p.e.
float timingResolution = 0.6;//B. Quilain, to match the TTS = 1.4ns (sigma at 0.6) measured at 1 p.e.
//0.5*(0.33 + sqrt(timingConstant/Q)); //factor 0.5 for expected improvement and required TTS
// looking at SK's jitter function for 20" tubes
//if (timingResolution < 0.58) timingResolution=0.58;
float Smearing_factor = G4RandGauss::shoot(0.0,timingResolution);
return Smearing_factor;
}

```

Plan

- Validation of Benjamin-san's reconstruction code → OK (?)
 - However memory leak problem remains for reconstruction code
- n-tagging
 - Generate MC of 2.2 MeV gamma
 - Reconstruct 2.2 MeV gamma with Benjamin-san's method
- Fraction of miss reconstruction
 - Generate MC of electrons
 - with some energies (3 MeV, 5 MeV, 10 MeV ?)
 - with some dark rate?
 - with # of mPMTs (masking mPMT)
 - Calculate fraction of miss reconstruction like today's method
 - How define regions, inner volume and near edge?
 -

Software issue

- WCSim had segmentation fault after I rebuilt.
 - I cannot generate MC
 - I put log on “/disk02/usr6/sizu/wcsimgen*.out”
- FitVertexLE memory leak
 - It allocated large memory and I cannot run
 - It used all of memory of sukap, 125 GB, and was killed
 - I tried fix memory leak but WCSim library seems to have issue
 - When I analyzed it with valgrind, it founds GetPMT() function of WCSimRootGeom class has leak
- They seems to be complicated for me
- → Do you have any idea? I need help to fix them