# Update for cluster deghosting

Xuyang Ning

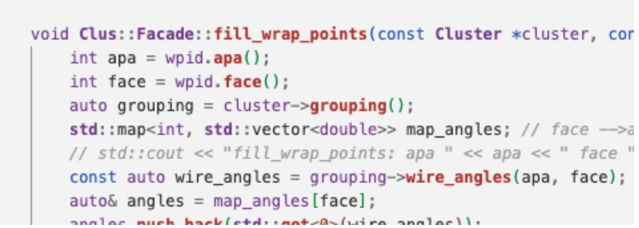
# Problems:

**In DynamicPointCloud**

For APA1 and 3:
dpc_points. wpid_2d. name = a3f0pA
dpc_points. wpid. name = a3f1pA

**make_points_cluster()**

**fill_wrap_points()**



```
void Clus::Facade::fill_wrap_points(const Cluster *cluster, cor
    int apa = wpid.apa();
    int face = wpid.face();
    auto grouping = cluster->grouping();
    std::map<int, std::vector<double>> map_angles; // face -->a
    // std::cout << "fill_wrap_points: apa " << apa << " face "
    const auto wire_angles = grouping->wire_angles(apa, face);
    auto& angles = map_angles[face];
    angles.push_back(std::get<0>(wire_angles));
```

**add_points():**

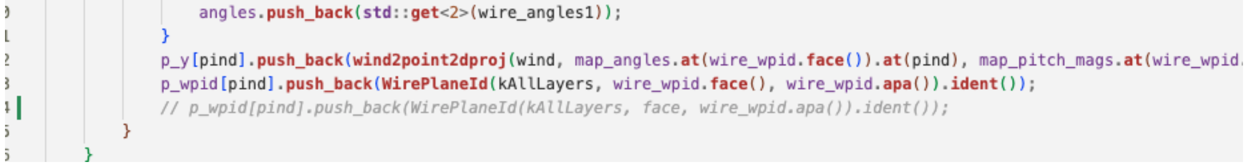
- All 2D points will be assigned to face 0,
- While points in cluster to be tested are all in face1

reference

target

**get_closest_2d_point_info()**

- Found nothing for APA3 w



```
                angles.push_back(std::get<2>(wire_angles1));
            }
            p_y[pind].push_back(wind2point2dproj(wind, map_angles.at(wire_wpid.face()).at(pind), map_pitch_mags.at(wire_wpid.
            p_wpid[pind].push_back(WirePlaneId(kAllLayers, wire_wpid.face(), wire_wpid.apa()).ident());
            // p_wpid[pind].push_back(WirePlaneId(kAllLayers, face, wire_wpid.apa()).ident());
        }
    }
```

Here if I change wire_wpid.face() to face, everything works well for my deghosting2

https://github.com/WireCell/wire-cell-toolkit/blob/apply-pointcloud/clus/src/DynamicPointCloud.cxx#L746

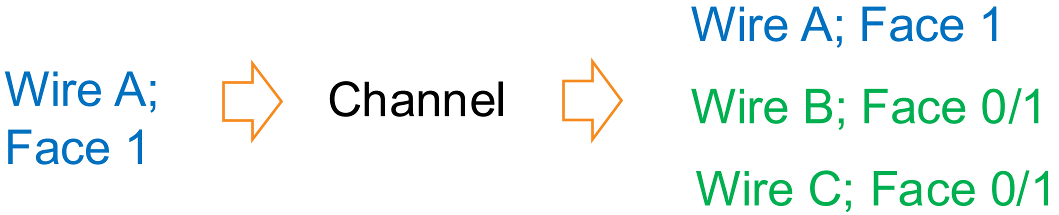
# fill_wrap_points()

- void Clus::Facade::fill_wrap_points()
- Project a 3D point (APA, face) to wire planes
  - First project to its real APA and face, get the wire index (wire A, face1)

  - Since the wires are wrapped, what we actually measured is on the channel.
  - One channel will corresponds to several wires index. (wire A B C)

- For the deghosting purpose, it means the ghost can actually happened on the other face in same APA.
  - Can't simply force wire plane face to be the same as point.

Expected:



Wire A; Face 1

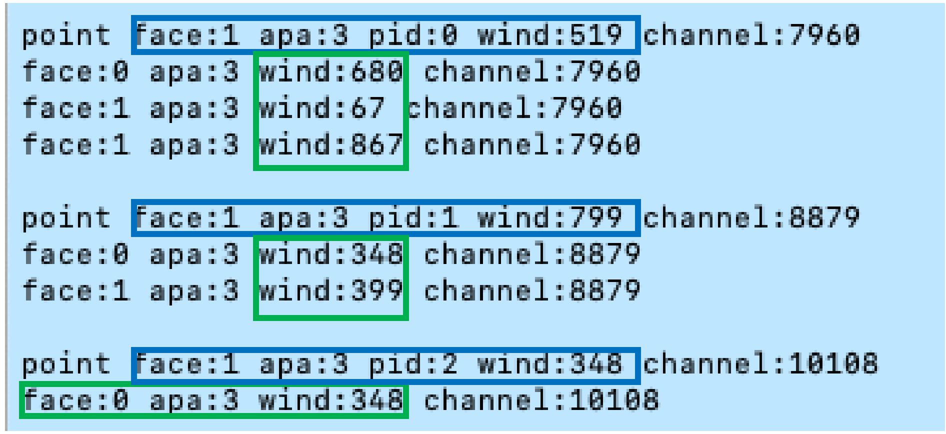
Channel

Wire A; Face 1

Wire B; Face 0/1

Wire C; Face 0/1

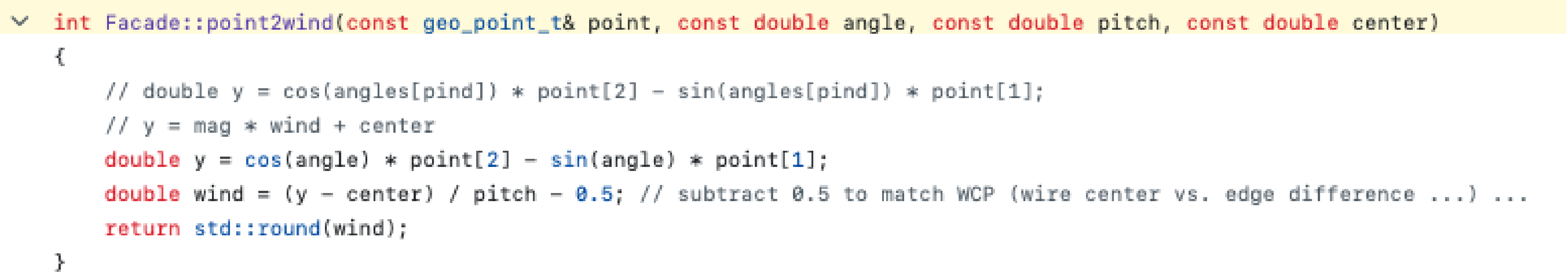
Actual:



```
point face:1 apa:3 pid:0 wind:519 channel:7960
face:0 apa:3 wind:680 channel:7960
face:1 apa:3 wind:67 channel:7960
face:1 apa:3 wind:867 channel:7960

point face:1 apa:3 pid:1 wind:799 channel:8879
face:0 apa:3 wind:348 channel:8879
face:1 apa:3 wind:399 channel:8879

point face:1 apa:3 pid:2 wind:348 channel:10108
face:0 apa:3 wind:348 channel:10108
```

# Project point to 2D plane

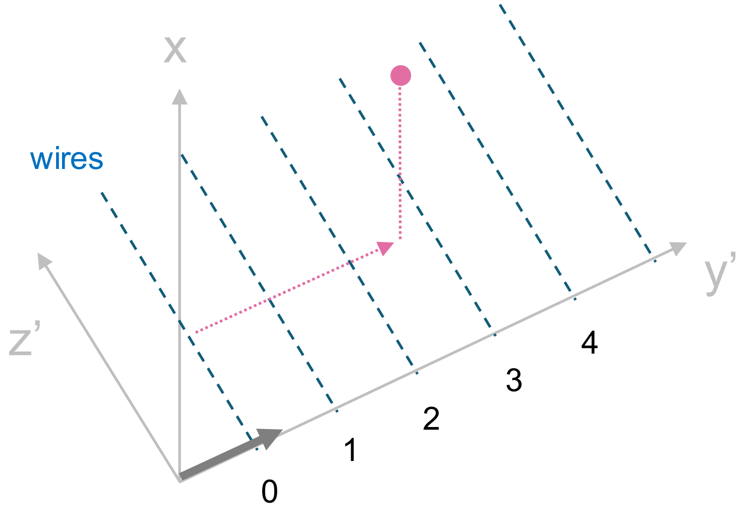
[clus/src/Facade_Util.cxx#L524]



```
int Facade::point2wind(const geo_point_t& point, const double angle, const double pitch, const double center)
{
    // double y = cos(angles[pind]) * point[2] - sin(angles[pind]) * point[1];
    // y = mag * wind + center
    double y = cos(angle) * point[2] - sin(angle) * point[1];
    double wind = (y - center) / pitch - 0.5; // subtract 0.5 to match WCP (wire center vs. edge difference ...) ...
    return std::round(wind);
}
```

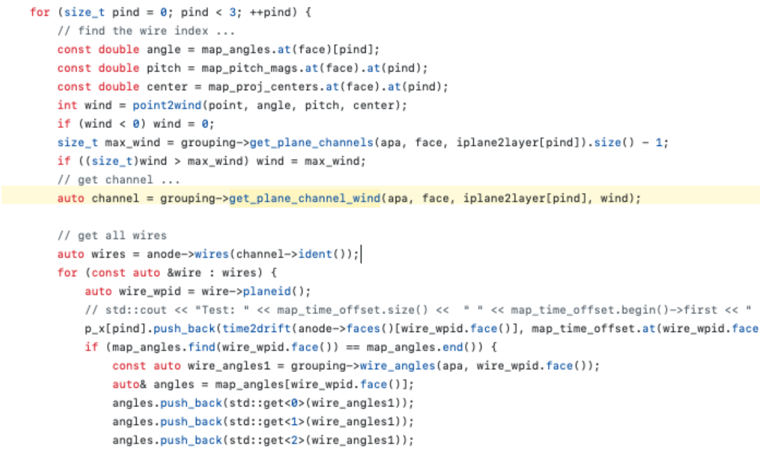
- Wire index are 1D coordinate
- In a certain face, wire index are defined directly from geometry.
  - can also calculated directly from geometry.
- wire index got from points are correct.

- Some facts:
  - In each face, wire index start from 0 in order.
  - For connected channels, the order may not be from small to large.
  - Order of channels may be more complicated when mixing with different faces.



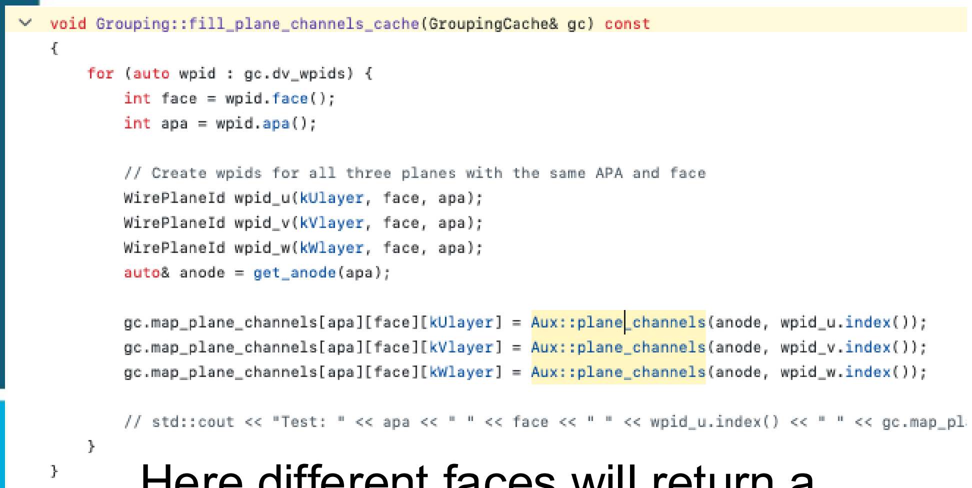
# From w_ind to channel:

clus/src/DynamicPointCloud.cxx#L730



```cpp
for (size_t pind = 0; pind < 3; ++pind) {
    // find the wire index ...
    const double angle = map_angles.at(face)[pind];
    const double pitch = map_pitch_mags.at(face).at(pind);
    const double center = map_proj_centers.at(face).at(pind);
    int wind = point2wind(point, angle, pitch, center);
    if (wind < 0) wind = 0;
    size_t max_wind = grouping->get_plane_channels(apa, face, iplane2layer[pind]).size() - 1;
    if ((size_t)wind > max_wind) wind = max_wind;
    // get channel ...
    auto channel = grouping->get_plane_channel_wind(apa, face, iplane2layer[pind], wind);

    // get all wires
    auto wires = anode->wires(channel->ident());
    for (const auto &wire : wires) {
        auto wire_wpid = wire->planeid();
        // std::cout << "Test: " << map_time_offset.size() <<  " " << map_time_offset.begin()->first << "
        p_x[pind].push_back(time2drift(anode->faces()[wire_wpid.face()], map_time_offset.at(wire_wpid.face
        if (map_angles.find(wire_wpid.face()) == map_angles.end()) {
            const auto wire_angles1 = grouping->wire_angles(apa, wire_wpid.face());
            auto& angles = map_angles[wire_wpid.face()];
            angles.push_back(std::get<0>(wire_angles1));
            angles.push_back(std::get<1>(wire_angles1));
            angles.push_back(std::get<2>(wire_angles1));
```

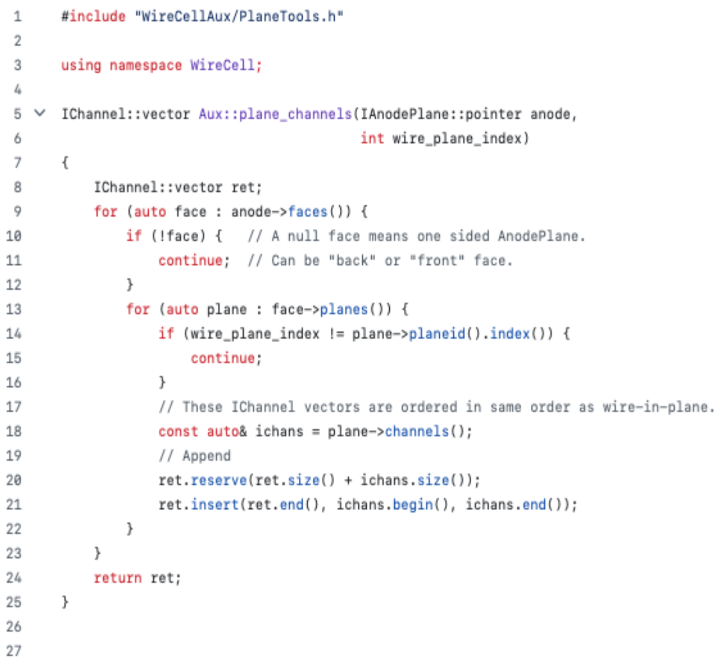
clus/src/Facade_Grouping.cxx#L214



```cpp
void Grouping::fill_plane_channels_cache(GroupingCache& gc) const
{
    for (auto wpid : gc.dv_wpids) {
        int face = wpid.face();
        int apa = wpid.apa();

        // Create wpids for all three planes with the same APA and face
        WirePlaneId wpid_u(kUlayer, face, apa);
        WirePlaneId wpid_v(kVlayer, face, apa);
        WirePlaneId wpid_w(kWlayer, face, apa);
        auto& anode = get_anode(apa);

        gc.map_plane_channels[apa][face][kUlayer] = Aux::plane_channels(anode, wpid_u.index());
        gc.map_plane_channels[apa][face][kVlayer] = Aux::plane_channels(anode, wpid_v.index());
        gc.map_plane_channels[apa][face][kWlayer] = Aux::plane_channels(anode, wpid_w.index());

        // std::cout << "Test: " << apa << " " << face << " " << wpid_u.index() << " " << gc.map_pl
    }
}
```

Here different faces will return a same Aux::plane_channels

aux/src/PlaneTools.cxx#L5



```cpp
#include "WireCellAux/PlaneTools.h"

using namespace WireCell;

IChannel::vector Aux::plane_channels(IAnodePlane::pointer anode,
                                     int wire_plane_index)
{
    IChannel::vector ret;
    for (auto face : anode->faces()) {
        if (!face) {   // A null face means one sided AnodePlane.
            continue;  // Can be "back" or "front" face.
        }
        for (auto plane : face->planes()) {
            if (wire_plane_index != plane->planeid().index()) {
                continue;
            }
            // These IChannel vectors are ordered in same order as wire-in-plane.
            const auto& ichans = plane->channels();
            // Append
            ret.reserve(ret.size() + ichans.size());
            ret.insert(ret.end(), ichans.begin(), ichans.end());
        }
    }
    return ret;
}
```

- Mix face 0 and 1
- Fill face 0 then face 1

For APA3:
Aux::plane_channels(anode, wpid_w.index())
= [9760, 9761, ..., 9280, 9281, ...]

Face 0 Face 1

- It uses wire index as the index of **plane_channels(anode, wpid_w.index())** to get corresponding channel number.
  - If search APA3, face1, w plane, wire_index=1,
  - will return channel=9761, which is a face 0 channel.

- For u and v plane, since wires are wrapped across 2 faces, things got even worse.
  - 1147 wires, more than 400 channels per face.
- Not one-one correspondence. Can't search by simple using wire-index as the vector<channel> index.

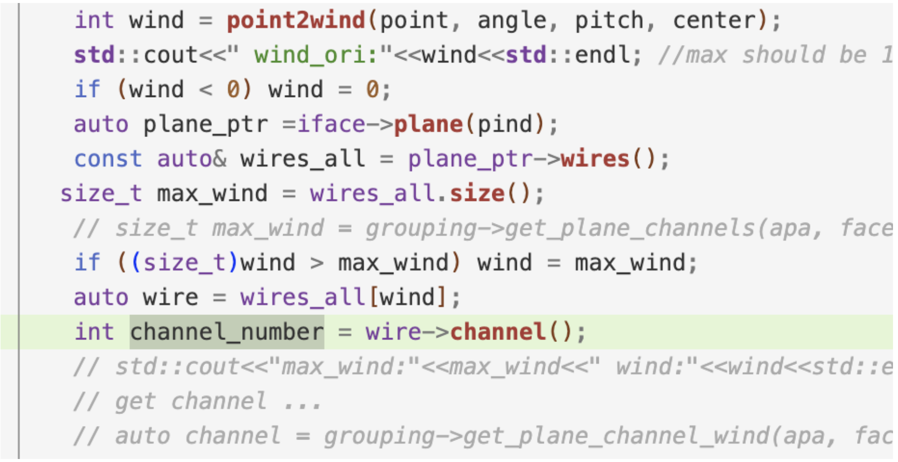
# Possible solution?

Aux::plane_channels(anode, wpid_w.index())

$$= [\underbrace{9760, 9761, \ldots,}_{\text{Face 0}}\ \underbrace{9280, 9281, \ldots}_{\text{Face 1}}]$$

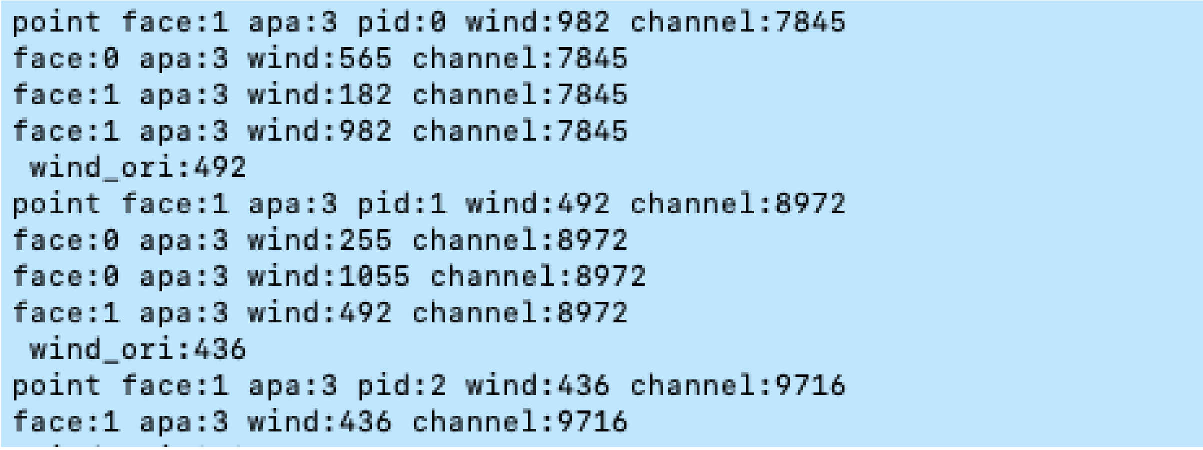
- Where this channel order come from?
  - Read from geometry file? Face 0 filled first, then face 1.
- Possible way to fix it?
  - Split channels in separated faces, and reorder it according to wire_index
  - Need to think about u and v plane.
  - We can discuss more….

- Now for my own use, I totally discard get_plane_channel_wind()



```
    int wind = point2wind(point, angle, pitch, center);
    std::cout<<" wind_ori:"<<wind<<std::endl; //max should be 1
    if (wind < 0) wind = 0;
    auto plane_ptr =iface->plane(pind);
    const auto& wires_all = plane_ptr->wires();
   size_t max_wind = wires_all.size();
    // size_t max_wind = grouping->get_plane_channels(apa, face
    if ((size_t)wind > max_wind) wind = max_wind;
    auto wire = wires_all[wind];
    int channel_number = wire->channel();
    // std::cout<<"max_wind:"<<max_wind<<" wind:"<<wind<<std::e
    // get channel ...
    // auto channel = grouping->get_plane_channel_wind(apa, fac
```

- Simple check is right.



```
point face:1 apa:3 pid:0 wind:982 channel:7845
face:0 apa:3 wind:565 channel:7845
face:1 apa:3 wind:182 channel:7845
face:1 apa:3 wind:982 channel:7845
 wind_ori:492
point face:1 apa:3 pid:1 wind:492 channel:8972
face:0 apa:3 wind:255 channel:8972
face:0 apa:3 wind:1055 channel:8972
face:1 apa:3 wind:492 channel:8972
 wind_ori:436
point face:1 apa:3 pid:2 wind:436 channel:9716
face:1 apa:3 wind:436 channel:9716
```