Cluster facade and PC-tree scopes

Brett Viren

February 11, 2025

Issue

*A point tree **scope** defines a selection on the PC-tree that determines **coordinate** arrays on which a k-d tree is built.*

Currently `Cluster` has these pre-defined `Tree::Scope` data members

`scope` `x/y/z` 3D coordinates

`scope_wire_index` `u/v/w` wire indices

`scope2ds[3]` 3 scopes `x/y` from PCs named like `2dp<n>`.

We want more than one 3D `scope` to be considered by `Cluster` methods

Eg, "raw" `x/y/z` but also `{x/y/z}_corr` after correcting for t_0.

Proposed solution - a new `Facade::Scope` class

1. Identify all `Cluster` methods that rely on **scope**.
2. Define a new `Scope` facade class that:
   - holds a 3D scope
   - provides the scope-dependent methods.

User calls:

```
Facade::Scope Cluster::scope_facade(Tree::Scope user_scope);
```

Proposed solution - factoring `Facade::Cluster`

Best (but more work): remove 3D scope-dependent methods from `Cluster`.

- User code now makes a `Facade::Scope` for default `x/y/z` and uses that.

Less good (but easier): `Cluster` scope-dependent methods as-is and either:

- Leave their implementations as-is **or**
- Give `Cluster` a `Facade::Scope` to which methods forward **or**
- Make `Cluster` inherit from `Scope` to supply the methods.

Scope methods

LLamma3.3:70b tells me:

> *Based on the provided context, the following methods call any of the previously listed methods (`kd_radius()`, `kd_knn()`, `get_closest_points()`, or `get_blob_indices()`):*

`Cluster::get_furthest_wcpoint()` This method calls `get_closest_points()`.

`Cluster::blob_merge(Cluster& other)` This method calls `get_blob_indices()`.

`Cluster::npoints_within(const Cluster& other, double distance) const` This method calls `kd_radius()`.

Actually, it **hallucinates many methods** until I instruct it to:

> *… and only list methods actually in the context.*