accumulate not a part of std

Asked 1 year, 8 months ago Modified 8 months ago Viewed 6k times

5

I'm new to C++ (and OpenCV). Found some code and trying to compile it but get an error. What does it mean?

First of all `accumulate` was `std::accumulate()` but the compiler returned an error telling it was not a member of `std`. Then I changed it to `cv::accumulate()` but now I get a new error? Maybe it should be a memeber of `std` but I'm missing to include a header file?

```
In file included from txtbin.hpp:11,
                 from txtbin.cpp:12:
dewarp.hpp: In member function ‘std::vector<cv::Point_<double> >
Dewarp::keypoints_from_samples(const std::vector<std::vector<cv::Point_<double> >
>&, std::vector<double, std::allocator<double> >&,
std::vector<std::vector<double, std::allocator<double> > >&)’:
dewarp.hpp:96:47: error: invalid initialization of reference of type
‘cv::InputArray’ {aka ‘const cv::_InputArray&’} from expression of type
‘std::vector<double, std::allocator<double> >::iterator’
   96 |   double mean = cv::accumulate(py_coords.begin(), py_coords.end(), 0.0) /
py_coords.size();
```

code

```
std::vector<double> px_coords, py_coords;

for(const std::vector<cv::Point2d>& points : span_points){
        std::vector<double> py_coords, px_coords;
        for(const cv::Point2d& point : points){
            py_coords.push_back(point.dot(y_dir));
            px_coords.push_back(point.dot(x_dir) - px0);
        }
        double mean = cv::accumulate(py_coords.begin(), py_coords.end(), 0.0) /
py_coords.size();
        y_coords.push_back(mean - py0);
        x_coords.push_back(px_coords);
    }
```

c++ opencv

Share Improve this question Follow

asked Apr 23, 2022 at 18:35

clarkk

27.3k 72 206 353

Most likely you are missing a proper #include. Leave it up to you what should you #include. – 273K Apr

12 `std::accumulate` is part of `<numeric>` . Make sure that is included in your include-file set.
– WhozCraig Apr 23, 2022 at 19:05

@WhozCraig it works.. create an answer :) – clarkk Apr 23, 2022 at 19:12

1 Answer

Sorted by: Highest score (default)

4

Converting WhozCraig's comment to an answer:

```
#include <numeric>
```

to use `std::accumulate` .

Share Improve this answer Follow

answered May 2, 2023 at 3:52
ggorlen
48.4k 7 83 121