WeightedPoint Class

The Weighted Point class is used to unite clusters of points into a point that represents the center of mass for this specific cluster. The mathematical definition of the center of mass point is $p_{cm}=\frac{1}{n}\sum_{i=1}^{n} p_i$ since all the points have the same weight. In this manner, since we do not know how many points are in the cluster, we need to add point by point, until the distance between the center of mass and the next point is too large. During this sum, we need to count how many points were added, having a partial mean on each step.

Public Methods

- ## WeightedPoint( ) [1/2]

  WeightedPoint::WeightedPoint( )

  Default constructor that assigns default value to point and weight

- ## WeightedPoint( ) [2/2]

  ```
  WeightedPoint::WeightedPoint ( const double x,
                                 const double y,
                                 const int w = 1
                               )
  ```

  Constructor to assign value to x and y coordinates as well as point weight with default = 1

  Parameters

  x is the point's x coordinate
  y is the point's y coordinate
  w is the point's weight value

- ## getWeight( )

  ```
  int WeightedPoint::getWeight ( ) const
  ```

  Get point's weight value

- ## fusePoint( )

  ```
  bool WeightedPoint::fusePoint ( const WeightedPoint & wp,
                                  const double limitDist
                                )
  ```

  If the distance between the two points is bigger than the determined by *limitDist* the method returns false (fusion not made) else, it fuses the object with p calculating a weighted average between the two and returns true (fusion made)

Parameters

wp is the second weighted point to fuse with object
limitDist is the distance limit to fuse the two points

- operator == ( )

  bool WeightedPoint::operator == (const WeightedPoint & wp) const

  Checks if two weighted points both coordinates and weight are equal

  Parameters

  wp is the second weighted point to be compared

- operator != ( )

  bool WeightedPoint::operator != (const WeightedPoint & wp) const

  Checks if any of the two weighted points coordinates or weight are different

  Parameters

  wp is the second weighted point to be compared

- friend operator << ( )

  std::ostream & operator << (std::ostream & out, const WeightedPoint & wp)

  Print weighted point object in terminal in the form WeightedPoint: [x: ‘p.x’, y: ‘p.y’, Weight: ‘p.w’]

  Parameters

  out is where to print, normally terminal
  wp is the object to be printed