

PoissonMVCs (by Xianying Li)

This is the C++ implementation for the paper “Poisson Coordinates” on the CPU.

Interface

Function: `void poissonMVCs(const vector<Point2D> &poly, const Point2D &p, vector<double> &coords, BaseCircle c = BaseCircle());`

This function computes the Poisson coordinates of a given point with respect to a 2D polygon; this polygon should be without holes.

Input: a 2D polygon *poly* (without holes), a 2D point *p*, and a projection circle *c*.

Output: the Poisson coordinates *coords*.

Notice:

1. *c* is **NOT** required to be a unit circle.
2. *p* is required to be within the interior of *c*.
3. To obtain a ‘regular placement’, we can use a shared base circle *c* for all points *p*.
4. Setting *c*’s radius to zero means to use a unit circle centered at $(p.x+c.cx, p.y+c.cy)$.
5. Using default *c* mean to computes the mean value coordinates.

Function: `void poissonMVCs(const vector<Point2D> &poly, const vector<int> &edge, const Point2D &p, vector<double> &coords, BaseCircle c = BaseCircle());`

This function computes the Poisson coordinates of a given point with respect to a 2D region; this region could have one or more holes.

poly stores the position information of all the vertices of the region.

edge gives the edge information:

edge[2*i] and *edge*[2*i+1] indicate the indices of the 2 endpoints of the i-th edge in *poly*.

Function: `void minCircle(const vector<Point2D> &poly, BaseCircle &c);`

This function computes the minimal circle that covers *poly*.

Use this function to get the base circle of the ‘basic regular placement’.