

CubicMVCs (by Xianying Li)

This is the C++ implementation for the paper “Cubic Mean Value Coordinates” on the CPU.

Interface

Function: `void cubicMVCs(const vector<Point2D> &poly, const Point2D &p, vector<double> &vCoords, vector<double> &gnCoords, vector<double> >Coords);`

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

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

Output: the cubic mean value coordinates **vCoords**, **gnCoords**, **gtCoords**.

vCoords[*i*] is the coordinate for boundary value at vertex **poly**[*i*];

gnCoords[2**i*] is the coordinate for (normal) gradient at vertex **poly**[*i*] in the left handside normal direction of [**poly**[*i*], **poly**[*i*+1]];

gnCoords[2**i*+1] is the coordinate for (normal) gradient at vertex **poly**[*i*+1] in the left handside normal direction of [**poly**[*i*], **poly**[*i*+1]];

gtCoords[2**i*] is the coordinate for (tangent) gradient at vertex **poly**[*i*] in the direction of [**poly**[*i*], **poly**[*i*+1]];

gtCoords[2**i*+1] is the coordinate for (tangent) gradient at vertex **poly**[*i*+1] in the direction of [**poly**[*i*+1], **poly**[*i*]];

Function: `void poissonMVCs(const vector<Point2D> &poly, const vector<int> &edge, const Point2D &p, vector<double> &vCoords, vector<double> &gnCoords, vector<double> >Coords);`

This function computes the cubic mean value 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**.