SphericalInterpolate

SphericalInterpolate *triangulationDataWave*, *dataPointsWave*, *newLocationsWave*

The SphericalInterpolate operation works in conjunction with the SphericalTriangulate operation to calculate interpolated values on a surface of a sphere. Given a set of $\{x_i, y_i, z_i\}$ points on the surface of a sphere with their associated values $\{v_i\}$, the SphericalTriangulate operation performs the Delaunay triangulation and creates an output that is used by the SphericalInterpolate operation to calculate values at any other point on the surface of a sphere. The interpolation calculation uses Voronoi polygons to weigh the contribution of the nearest neighbors to any given location on the sphere.

Parameters

triangulationDataWave is a 13 column wave that was created by the SphericalTriangulate operation.

dataPoints is a 4 column wave. The first 3 columns are the $\{x_i, y_i, z_i\}$ locations that were used to create the triangulation, and the last column corresponds to the $\{v_i\}$ values at the triangulation locations.

newLocationsWave is a 3 column wave that specifies the x, y, z locations on the sphere at which the interpolated values are calculated. Note that internally, each triplet is normalized to a point on the unit sphere before it is used in the interpolation.

Details

You will always need to use the SphericalTriangulate operation first to generate the *triangulationDataWave* input for this operation.

The result of the operation are put in the wave W_SphericalInterpolation.

See Also

SphericalTriangulate, **Triangulate3D**, **ImageInterpolate** with keyword Voronoi

Demo

Choose File→Example Experiments→Analysis→SphericalTriangulationDemo.

SphericalTriangulate

SphericalTriangulate [**/Z**] ***tripletWaveName***

The SphericalTriangulate operation triangulates an arbitrary XYZ triplet wave on a surface of a sphere.

It starts by normalizing the data to make sure that $\text{sqrt}(x^2+y^2+z^2)=1$, and then proceeds to calculate the Delaunay triangulation.

Flags

/Z No error reporting.

Details

The result of the triangulation is the wave M_SphericalTriangulation. This 13 column wave is used in SphericalInterpolate to obtain the interpolated values.

Example

```
// Generates output waves that can be used in Gizmo to display the triangulation.
// triangulationData is the M_TriangulationData output from SphericalTriangulation.
// tripletWave is the source wave input to SphericalTriangulation.
// Output wave sphereTrianglesPath can be used to display the triangulation as a path.
// Output wave sphereTrianglesSurf can be used to display the triangulation as a surface.
Function BuildTriangleWaves(triangulationData,tripletWave)
    Wave triangulationData, tripletWave

    // Extract 3 columns from triangulationData that contain the index of the row.
    Duplicate/O/FREE/r=[][1,3] triangulationData,triIndices
    Variable finalNumTriangles=dimSize(triIndices,0),i,j,k

    // Initialize both waves to NaN so any unassigned point would appear as a hole.
    Make/O/N=(5*finalNumTriangles,3) sphereTrianglesPath=NaN
    Make/O/N=(3*finalNumTriangles,3) sphereTrianglesSurf=NaN

    // Assign the values of the vertices to the two waves:
    Variable rowIndex,rowIndex0,outRowCount=0,outcount2=0
    for(i=1;i<finalNumTriangles;i+=1)
```