Heat Diffusion Library

Generated by Doxygen 1.8.6

Sat Apr 2 2016 00:26:15

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 HeatDiffusion::Distance::BiharmonicDistance Class Reference

Compute the Biharmonic distance between vertices in a 3D mesh.

```
#include <BiharmonicDistance.h>
```

Public Member Functions

- BiharmonicDistance ()

  Constructor.
- BiharmonicDistance (float ∗eval, float ∗evec, unsigned int nV, unsigned int nE)

  Constructor.
- void setEigendecomposition (float ∗eval, float ∗evecs)

  Setter.
- void setNumberVertices (unsigned int nV)

  Setter method.
- void setNumberEigenvalues (unsigned int nE)

  Setter method.
- float computeSquaredDistance (unsigned int v1, unsigned int v2)

  Computation of distance.

Protected Attributes

- float ∗ **evals**
- float ∗ **evecs**
- unsigned int **num_vertex**
- unsigned int **num_eigenvalues**

2.1.1 Detailed Description

Compute the Biharmonic distance between vertices in a 3D mesh.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 HeatDiffusion::Distance::BiharmonicDistance::BiharmonicDistance ( )

Constructor.

Default constructor.

2.1.2.2 HeatDiffusion::Distance::BiharmonicDistance::BiharmonicDistance (float * *eval,* float * *evec,* unsigned int *nV,* unsigned int *nE*)

Constructor.

It creates an object with the provided data

Parameters

eval	the buffer of eigenvalues
evecs	the buffer of eigenvectors
nV	the number of vertices
nE	the number of eigenvalues

2.1.3 Member Function Documentation

2.1.3.1 float HeatDiffusion::Distance::BiharmonicDistance::computeSquaredDistance (unsigned int *v1,* unsigned int *v2*)

Computation of distance.

Computes the squared conmute-time distance between two vertices

Parameters

v1	the index of the first vertex
v2	the index of the second vertex

2.1.3.2 void HeatDiffusion::Distance::BiharmonicDistance::setEigendecomposition (float * *eval,* float * *evecs*)

Setter.

Sets the eigenvalues and eigenvectors of the Laplacian matrix

2.1.3.3 void HeatDiffusion::Distance::BiharmonicDistance::setNumberEigenvalues (unsigned int *nE*)

Setter method.

Sets the number of eigen-pairs

Parameters

nE	the number of eigen-pairs

2.1.3.4 void HeatDiffusion::Distance::BiharmonicDistance::setNumberVertices (unsigned int *nV*)

Setter method.

Sets the number of vertices

Parameters

nV	the number of vertices

The documentation for this class was generated from the following file:

- include/Distance/BiharmonicDistance.h

2.2 HeatDiffusion::Distance::ConmuteTimeDistance Class Reference

Compute the Conmute-time distance between vertices in a 3D mesh.

```
#include <ConmuteTimeDistance.h>
```

Public Member Functions

- ConmuteTimeDistance ()

  Constructor.
- ConmuteTimeDistance (float ∗eval, float ∗evec, unsigned int nV, unsigned int nE)

  Constructor.
- void setEigendecomposition (float ∗eval, float ∗evecs)

  Setter.
- void setNumberVertices (unsigned int nV)

  Setter method.
- void setNumberEigenvalues (unsigned int nE)

  Setter method.
- float computeSquaredDistance (unsigned int v1, unsigned int v2)

  Computation of distance.

Protected Attributes

- float ∗ **evals**
- float ∗ **evecs**
- unsigned int **num_vertex**
- unsigned int **num_eigenvalues**

2.2.1 Detailed Description

Compute the Conmute-time distance between vertices in a 3D mesh.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 HeatDiffusion::Distance::ConmuteTimeDistance::ConmuteTimeDistance ( )

Constructor.

Default constructor.

2.2.2.2 HeatDiffusion::Distance::ConmuteTimeDistance::ConmuteTimeDistance (float ∗ *eval*, float ∗ *evec*, unsigned int *nV*, unsigned int *nE*)

Constructor.

It creates an object with the provided data

Parameters

eval	the buffer of eigenvalues

evecs	the buffer of eigenvectors
nV	the number of vertices
nE	the number of eigenvalues

2.2.3 Member Function Documentation

2.2.3.1 float HeatDiffusion::Distance::ConmuteTimeDistance::computeSquaredDistance (unsigned int *v1*, unsigned int *v2*)

Computation of distance.

Computes the squared conmute-time distance between two vertices

Parameters

v1	the index of the first vertex
v2	the index of the second vertex

2.2.3.2 void HeatDiffusion::Distance::ConmuteTimeDistance::setEigendecomposition (float ∗ *eval*, float ∗ *evecs*)

Setter.

Sets the eigenvalues and eigenvectors of the Laplacian matrix

2.2.3.3 void HeatDiffusion::Distance::ConmuteTimeDistance::setNumberEigenvalues (unsigned int *nE*)

Setter method.

Sets the number of eigen-pairs

Parameters

nE	the number of eigen-pairs

2.2.3.4 void HeatDiffusion::Distance::ConmuteTimeDistance::setNumberVertices (unsigned int *nV*)

Setter method.

Sets the number of vertices

Parameters

nV	the number of vertices

The documentation for this class was generated from the following file:

- include/Distance/ConmuteTimeDistance.h

2.3 HeatDiffusion::Distance::DiffusionDistance Class Reference

Compute the Diffusion distance between vertices in a 3D mesh.

```
#include <DiffusionDistance.h>
```

Public Member Functions

- DiffusionDistance ()

Constructor.

- DiffusionDistance (float ∗eval, float ∗evec, unsigned int nV, unsigned int nE)

  Constructor.
- void setEigendecomposition (float ∗eval, float ∗evecs)

  Setter.
- void setNumberVertices (unsigned int nV)

  Setter method.
- void setNumberEigenvalues (unsigned int nE)

  Setter method.
- float computeSquaredDistance (unsigned int v1, unsigned int v2, float t)

  Computation of distance.

Protected Attributes

- float ∗ **evals**
- float ∗ **evecs**
- unsigned int **num_vertex**
- unsigned int **num_eigenvalues**

2.3.1 Detailed Description

Compute the Diffusion distance between vertices in a 3D mesh.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 HeatDiffusion::Distance::DiffusionDistance::DiffusionDistance ( )

Constructor.

Default constructor.

2.3.2.2 HeatDiffusion::Distance::DiffusionDistance::DiffusionDistance (float ∗ *eval*, float ∗ *evec*, unsigned int *nV*, unsigned int *nE*)

Constructor.

It creates an object with the provided data

Parameters

eval	the buffer of eigenvalues
evecs	the buffer of eigenvectors
nV	the number of vertices
nE	the number of eigenvalues

2.3.3 Member Function Documentation

2.3.3.1 float HeatDiffusion::Distance::DiffusionDistance::computeSquaredDistance (unsigned int *v1*, unsigned int *v2*, float *t*)

Computation of distance.

Computes the squared diffusion distance between two vertices given a value of time

Parameters

v1	the index of the first vertex
v2	the index of the second vertex
t	the value of time

2.3.3.2 void HeatDiffusion::Distance::DiffusionDistance::setEigendecomposition (float ∗ *eval,* float ∗ *evecs*)

Setter.

Sets the eigenvalues and eigenvectors of the Laplacian matrix

2.3.3.3 void HeatDiffusion::Distance::DiffusionDistance::setNumberEigenvalues (unsigned int *nE*)

Setter method.

Sets the number of eigen-pairs

Parameters

nE	the number of eigen-pairs

2.3.3.4 void HeatDiffusion::Distance::DiffusionDistance::setNumberVertices (unsigned int *nV*)

Setter method.

Sets the number of vertices

Parameters

nV	the number of vertices

The documentation for this class was generated from the following file:

- include/Distance/DiffusionDistance.h

2.4 HeatDiffusion::Descriptor::HeatKernelSignature Class Reference

HeatKernelSignature class.

```
#include <HeatKernelSignature.h>
```

Public Member Functions

- HeatKernelSignature ()

  A default constructor.
- HeatKernelSignature (float ∗eval, float ∗evec, float ∗area, unsigned int nV, unsigned int nE, unsigned int dim)

  A constructor.
- void setEigendecomposition (float ∗eval, float ∗evec, float ∗area)

  Setter method.
- void setNumberVertices (unsigned int nV)

  Setter method.
- void setNumberEigenvalues (unsigned int nE)

  Setter method.
- void setDescriptorDimension (unsigned int dim)

Setter method.

- void computeDescriptor ()

  Computation method.

- float ∗ getDescriptors ()

  Access method.

Protected Attributes

- float ∗ **evals**
- float ∗ **evecs**
- float ∗ **areas**
- unsigned int **num_vertex**
- unsigned int **num_eigenvalues**
- unsigned int **dimension**
- float ∗ **hks**

2.4.1 Detailed Description

HeatKernelSignature class.

A class to compute Heat Kernel Signatures

2.4.2 Constructor & Destructor Documentation

2.4.2.1 HeatDiffusion::Descriptor::HeatKernelSignature::HeatKernelSignature ( )

A default constructor.

It creates an empty HeatKernelSignature object. You must use the setter methods to provide the required information.

See Also

setEigendecomposition(), setNumberVertices(), setNumberEigenvalues() and setDescriptorDimension().

2.4.2.2 HeatDiffusion::Descriptor::HeatKernelSignature::HeatKernelSignature (float ∗ *eval,* float ∗ *evec,* float ∗ *area,* unsigned int *nV,* unsigned int *nE,* unsigned int *dim*)

A constructor.

It creates a HeatKernelSignature object with the provided information.

Parameters

eval	the buffer of eigenvalues
evec	the buffer of eigenvectors
area	the buffer of areas
nV	the number of vertices
nE	the number of eigen-pairs
dim	the dimension of the output descriptors

2.4.3 Member Function Documentation

2.4.3.1 void HeatDiffusion::Descriptor::HeatKernelSignature::computeDescriptor ( )

Computation method.

It performs the computation of the HKS descriptors

2.4.3.2 float∗ HeatDiffusion::Descriptor::HeatKernelSignature::getDescriptors ( )

Access method.

Gets the buffer of descriptors. Descriptor matrix (num_vertex x dimension) are stored in row-wise format

2.4.3.3 void HeatDiffusion::Descriptor::HeatKernelSignature::setDescriptorDimension (unsigned int *dim*)

Setter method.

Sets the dimension of the output descriptors

Parameters

dim	the dimension of the output descriptors

2.4.3.4 void HeatDiffusion::Descriptor::HeatKernelSignature::setEigendecomposition (float ∗ *eval,* float ∗ *evec,* float ∗ *area*)

Setter method.

Sets the information of the Eigendecomposition of the Laplacian

Parameters

eval	the buffer of eigenvalues
evec	the buffer of eigenvectors
area	the buffer of areas

2.4.3.5 void HeatDiffusion::Descriptor::HeatKernelSignature::setNumberEigenvalues (unsigned int *nE*)

Setter method.

Sets the number of eigen-pairs

Parameters

nE	the number of eigen-pairs

2.4.3.6 void HeatDiffusion::Descriptor::HeatKernelSignature::setNumberVertices (unsigned int *nV*)

Setter method.

Sets the number of vertices

Parameters

nV	the number of vertices

The documentation for this class was generated from the following file:

- include/Descriptor/HeatKernelSignature.h

2.5 HeatDiffusion::Laplacian Class Reference

Laplacian class.

```
#include <Laplacian.h>
```

Public Member Functions

- Laplacian ()

  Default constructor.
- Laplacian (int t, unsigned int numEig)

  A constructor.
- ∼Laplacian ()

  A destructor.
- void compute ()

  Method compute.
- void setGeometryData (int num_vertex, float ∗vertexBuffer, int num_triangles, unsigned int ∗indexBuffer)

  Method to set the geometry.
- void setType (int t)
- void setNumberEigenvalues (int numEig)
- float ∗ getAreas ()

  Access method.
- float ∗ getEvals ()

  Access method.
- float ∗ getEvecs ()

  Access method.
- unsigned int getNumEigenvalues ()

  Access method.

2.5.1 Detailed Description

Laplacian class.

A class to compute the Laplace-Beltrami operator of a 3D triangular mesh and its Eigendecomposition.

2.5.2 Constructor & Destructor Documentation

2.5.2.1 HeatDiffusion::Laplacian::Laplacian ( )

Default constructor.

It creates an empty object. You must use setType and setNumberEigenvalues to provide the required information for the computation.

See Also

setType() and setNumberEigenvalues()

2.5.2.2 HeatDiffusion::Laplacian::Laplacian (int *t*, unsigned int *numEig*) [inline]

A constructor.

It creates a Laplacian object with the given type and number of eigenvalues.

Parameters

t	the type of Laplacian={BELKIN, COTAN, GRAPH}
numEig	the number of eigen-pairs in the Eigendecomposition

2.5.2.3 HeatDiffusion::Laplacian::∼Laplacian ( )

A destructor.

Destroy the memory buffers for eigenvalues, eigenvectors and areas

2.5.3 Member Function Documentation

2.5.3.1 void HeatDiffusion::Laplacian::compute ( )

Method compute.

Method that actually performs the computation of the Laplacian matrix and the Eigendecomposition

2.5.3.2 float∗ HeatDiffusion::Laplacian::getAreas ( ) `[inline]`

Access method.

Get the buffer with area elements for vertices. Hint: the buffer is destroyed when Laplacian object is done. Make a copy if needed.

2.5.3.3 float∗ HeatDiffusion::Laplacian::getEvals ( ) `[inline]`

Access method.

Get the buffer with eigenvalues. Hint: the buffer is destroyed when Laplacian object is done. Make a copy if needed.

2.5.3.4 float∗ HeatDiffusion::Laplacian::getEvecs ( ) `[inline]`

Access method.

Get the buffer with eigenvectors. The eigenvectors are stored column-wise. Hint: the buffer is destroyed when Laplacian object is done. Make a copy if needed.

2.5.3.5 unsigned int HeatDiffusion::Laplacian::getNumEigenvalues ( ) `[inline]`

Access method.

Get the number of eigenvalues.

2.5.3.6 void HeatDiffusion::Laplacian::setGeometryData (int *num_vertex,* float ∗ *vertexBuffer,* int *num_triangles,* unsigned int ∗ *indexBuffer*)

Method to set the geometry.

It sets the required geometry.

Parameters

num_vertex	the number of vertices in the 3D mesh.
vertexBuffer	the vertex buffer (three values for each vertex).
num_triangles	the number of triangles in the 3D mesh.
indexBuffer	the index buffer (three values for each face).

2.5.3.7 void HeatDiffusion::Laplacian::setNumberEigenvalues (int *numEig*) [inline]

It sets the number of eigen-pairs to compute.

Parameters

numEig	the number of eigen-pairs required.

2.5.3.8 void HeatDiffusion::Laplacian::setType (int *t*) [inline]

It sets the type of Laplacian to compute.

Parameters

t	the type of Laplacian = {BELKIN, COTAN, GRAPH}

The documentation for this class was generated from the following file:

- include/Laplacian.h

2.6 HeatDiffusion::Vector Class Reference

Public Member Functions

- **Vector** (float x, float y, float z)
- float **x** ()
- float **y** ()
- float **z** ()
- float **dot** (Vector b)

The documentation for this class was generated from the following file:

- include/Laplacian.h

2.7 HeatDiffusion::Descriptor::WaveKernelSignature Class Reference

WaveKernelSignature class.

```
#include <WaveKernelSignature.h>
```

Public Member Functions

- WaveKernelSignature ()

  A default constructor.
- WaveKernelSignature (float ∗eval, float ∗evec, float ∗area, unsigned int nV, unsigned int nE, unsigned int dim, float var)

A constructor.

- void setEigendecomposition (float ∗eval, float ∗evec, float ∗area)

  Setter method.
- void setNumberVertices (unsigned int nV)

  Setter method.
- void setNumberEigenvalues (unsigned int nE)

  Setter method.
- void setDescriptorDimension (unsigned int dim)

  Setter method.
- void computeDescriptor ()

  Computation method.
- float ∗ getDescriptors ()

  Access method.

Protected Attributes

- float ∗ **evals**
- float ∗ **evecs**
- float ∗ **areas**
- unsigned int **num_vertex**
- unsigned int **num_eigenvalues**
- float **variance**
- unsigned int **dimension**
- float ∗ **wks**

2.7.1 Detailed Description

WaveKernelSignature class.

A class to compute Wave Kernel Signatures

2.7.2 Constructor & Destructor Documentation

2.7.2.1 HeatDiffusion::Descriptor::WaveKernelSignature::WaveKernelSignature ( )

A default constructor.

It creates an empty WaveKernelSignature object. You must use the setter methods to provide the required information.

See Also

> setEigendecomposition(), setNumberVertices(), setNumberEigenvalues() and setDescriptorDimension().

2.7.2.2 HeatDiffusion::Descriptor::WaveKernelSignature::WaveKernelSignature (float ∗ *eval,* float ∗ *evec,* float ∗ *area,* unsigned int *nV,* unsigned int *nE,* unsigned int *dim,* float *var*)

A constructor.

It creates a WaveKernelSignature object with the provided information.

Parameters

eval	the buffer of eigenvalues
evec	the buffer of eigenvectors
area	the buffer of areas
nV	the number of vertices
nE	the number of eigen-pairs
dim	the dimension of the output descriptors
var	the variance of energy functions

2.7.3 Member Function Documentation

2.7.3.1 void HeatDiffusion::Descriptor::WaveKernelSignature::computeDescriptor ( )

Computation method.

It performs the computation of the HKS descriptors

2.7.3.2 float∗ HeatDiffusion::Descriptor::WaveKernelSignature::getDescriptors ( )

Access method.

Gets the buffer of descriptors. Descriptor matrix (num_vertex x dimension) are stored in row-wise format

2.7.3.3 void HeatDiffusion::Descriptor::WaveKernelSignature::setDescriptorDimension (unsigned int *dim*)

Setter method.

Sets the dimension of the output descriptors

Parameters

dim	the dimension of the output descriptors

2.7.3.4 void HeatDiffusion::Descriptor::WaveKernelSignature::setEigendecomposition (float ∗ *eval,* float ∗ *evec,* float ∗ *area*)

Setter method.

Sets the information of the Eigendecomposition of the Laplacian

Parameters

eval	the buffer of eigenvalues
evec	the buffer of eigenvectors
area	the buffer of areas

2.7.3.5 void HeatDiffusion::Descriptor::WaveKernelSignature::setNumberEigenvalues (unsigned int *nE*)

Setter method.

Sets the number of eigen-pairs

Parameters

nE	the number of eigen-pairs

2.7.3.6 void HeatDiffusion::Descriptor::WaveKernelSignature::setNumberVertices (unsigned int *nV*)

Setter method.

Sets the number of vertices

Parameters

nV	the number of vertices

The documentation for this class was generated from the following file:

- include/Descriptor/WaveKernelSignature.h

Index