DaLI Descriptor Library

Generated by Doxygen 1.8.6

Wed Jan 7 2015 12:05:28

Contents

Chapter 1

DaLI Descriptor Library

Author

Edgar Simo-Serra esimo@iri.upc.edu

Version

1.0

Date

January 2015

1.1 License

1.2 Overview

This library implements the Deformation and Illumination Invariant Feature Point Descriptor or DaLI for short. The object is to provide a robust and fast feature point descriptor that has both deformation and illumination invariant properties. This descriptor is meant to be used in computer vision problems dealing with deformable objects.

1.3 Dependencies

- ceigs

  - ARPACK
  - LAPACK
  - BLAS
  - gfortran
  - CXSparse (suitesparse)
  - UMFPACK
- FFTW3

1.4 Changelog

- Version 0.1 (internal release), January 2012
  - Initial implementation as per original paper.

1.5 References

- Edgar Simo-Serra, Carme Torras, Francesc Moreno-Noguer. DaLI: Deformation and Light Invariant Descriptor International Journal of Computer Vision (IJCV), 2015.
- F. Moreno-Noguer. Deformation and Illumination Invariant Feature Point Descriptor. Conference in Computer Vision and Pattern Recognition (CVPR), 2011.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 dali_info_t Struct Reference

Gets information from the execution time.

`#include <dali.h>`

Data Fields

- int nodes_real
- int nodes_total
- int faces
- double time_meshing
- double time_laplacebeltrami
- double time_eigenvectors
- double time_hks
- double time_hks_si
- double time_elapsed

3.1.1 Detailed Description

Gets information from the execution time.

3.1.2 Field Documentation

3.1.2.1 int dali_info_t::faces

Number of faces for calculations.

3.1.2.2 int dali_info_t::nodes_real

Number of real nodes (pixels).

3.1.2.3 int dali_info_t::nodes_total

Total number of nodes for calculations.

3.1.2.4 double dali_info_t::time_eigenvectors

Time spent calculating eigenvectors.

3.1.2.5 double dali_info_t::time_elapsed

Sum of all elapsed time.

3.1.2.6 double dali_info_t::time_hks

Time spent calculating the Heat Kernel Signature.

3.1.2.7 double dali_info_t::time_hks_si

Time spent making scale-invariant.

3.1.2.8 double dali_info_t::time_laplacebeltrami

Time spent on calculating Laplace-Beltrami.

3.1.2.9 double dali_info_t::time_meshing

Time spent on meshing.

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

- dali.h

3.2 dali_params_t Struct Reference

The descriptor parameters.

`#include <dali.h>`

Data Fields

- dali_mesh_type_t mtype
- int Sz
- int Sz_coarse
- double beta
- int ncomp
- int ntime
- int wmax
- double **mesh_K**
- double **mesh_sigma**
- int threads
- int lanczos
- int eigs_iter
- int use_si
- int verbose

3.2.1 Detailed Description

The descriptor parameters.

Should be generally set by dali_optsDefault.

See Also

dali_optsDefault

3.2.2 Field Documentation

3.2.2.1 double dali_params_t::beta

Magnitude of the embedding, generally 1000<=beta<=2000. This represents the value to scale the intensity by. Defaults to 2000.

3.2.2.2 int dali_params_t::eigs_iter

Number of iterations to perform maximum for the ARNOLDI iteration algorithm used to calculate the Heat Kernel Signature. Defaults to 300.

3.2.2.3 int dali_params_t::lanczos

Number of Lanczos vectors to use in the Arnoldi iteration algorithm. More vectors generally mean more memory usage and faster convergence. A value of 0 attempts to set a good number of Lanczos vectors based on the value of ncomp. The value should be at least ncomp+1. Usually the value range is between 2∗ncomp to 4∗ncomp. Defaults to 0.

3.2.2.4 dali_mesh_type_t dali_params_t::mtype

Type of the mesh. Defaults to DALI_MESH_TYPE_CIRCLE_VARIABLE.

3.2.2.5 int dali_params_t::ncomp

Number of eigenvalues/vectors to compute for the Laplace-Beltrami operator. This is currently the slowest part of the algorithm and has the greatest impact on performance. Defaults to 200.

3.2.2.6 int dali_params_t::ntime

Number of time slices to use. This is used when performing the DFT. It must be less than or equal to ncomp. Defaults to 100.

3.2.2.7 int dali_params_t::Sz

Half of the patch size, generally 20<=S<=30. Defaults to 30.

3.2.2.8 int dali_params_t::Sz_coarse

Radius to use for denser meshing when using a variable type mesh. Defaults to 15.

3.2.2.9 int dali_params_t::threads

Number of threads to use. This will not work without a thread-safe version of ARPACK. Defaults to 1.

3.2.2.10 int dali_params_t::use_si

Whether or not to perform the final scale invariance stage (i.e. calculate HKS instead of HKS-SI Defaults to 1

3.2.2.11 int dali_params_t::verbose

Verbosity level of the algorithm. 0 is completely silent, while other values will display progress through stdout. Defaults to 0.

3.2.2.12 int dali_params_t::wmax

Dimension in the frequency domain, generally 20. This is only to save storage space. It does not actually speed up calculations. It must be equal or less to ntime. Defaults to 20.

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

- dali.h

3.3 dali_t Struct Reference

The actual DaLI descriptor itself.

`#include <dali.h>`

Data Fields

- int n
- int nodes
- int ulen
- int vlen
- int wlen
- int sz
- int len
- double * desc
- char * mask
- int * shape
- double * gauss
- double * sigma
- double * sgauss

3.3.1 Detailed Description

The actual DaLI descriptor itself.

To access the descriptor the following code can be used:

```
// Accesses slice at position (u,v,w) within the descriptor
// Only valid for u in [desc->uoff, desc->uoff+desc->ueff)
// Only valid for v in [desc->voff, desc->voff+desc->veff)
// Only valid for w in [0,desc->wlen)
double d = desc->desc[ ((u-desc->uoff)*desc->vlen + (v-desc->vlen) *
      desc->wlen + w ];
```

See Also

dali_compute

3.3.2 Field Documentation

3.3.2.1 double* dali_t::desc

Descriptor data of length len*wlen.

3.3.2.2 double* dali_t::gauss

Gaussian information (only stores x^2+y^2) of length len. Shared by all descriptors.

3.3.2.3 int dali_t::len

Descriptor length.

3.3.2.4 char* dali_t::mask

Mask data of length len.

3.3.2.5 int dali_t::n

Number of descriptors. This is used internally for freeing and shouldn't be modified.

3.3.2.6 int dali_t::nodes

Number of nodes.

3.3.2.7 double* dali_t::sgauss

Actual Gaussian information of length len. Shared by all descriptors.

3.3.2.8 int* dali_t::shape

Buffer that keeps track of the patch shape. Shared by all descriptors.

3.3.2.9 double* dali_t::sigma

Sigma used to calculate sgauss.

3.3.2.10 int dali_t::sz

Original half patch size.

3.3.2.11 int dali_t::ulen

The pitch of the width of the descriptor.

3.3.2.12 int dali_t::vlen

The pitch of the height of the descriptor.

3.3.2.13 int dali_t::wlen

Frequency information of the descriptor.

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

- dali.h

Index