

The CBIA i3dcore and i3dalgo libraries

Contributed by David Svoboda
Last Updated Wednesday, 06 January 2016

Introduction

The i3dcore is a 3D image representation library while the i3dalgo is a 3D image algorithms library. The former unifies the temporal image representation in the computer memory via the class Image3d as well as it provides means to store 3D images permanently on hard drives. It is a core element for image processing and analysis. The latter library is a collection of image processing and analysis algorithms. Most of software packages in the CBIA directly depends on the i3dalgo which, in turn, directly depends on the i3dcore.

Both libraries are vital for the CBIA software development.

The libraries are written in C++ with intensive use of templates. They are maintained to compile and run well on Windows with recent Visual Studio and Linux with recent GNU compiler continually.

License

The i3dcore and i3dalgo are both property of the CBIA. We offer them for free under the GNU GPL license. However, we protect all our downloads from the CBIA web pages by login and password. If you do not have your password yet, please register .

Download

- source files of the current version of the i3dcore
- source files of the current version of the i3dcore, i3dalgo and examples
- binary and header files for MS Windows, release version (not yet available the the latest version)
- binary and header files for MS Windows, debug version (not yet available the the latest version)

We do not provide binary versions for Linuxes as there are too many configurations possible (.so files depend on certain version of glibc, version found on your system will probably differ from ours). We encourage anyone to compile libraries from sources. It is especially easy on Linux.

An installation manual can be found here. It is primarily targeted for situation when both libraries and examples are downloaded. However, it is very easy to follow it when only the i3dcore was downloaded.

Depending on a setup of your operating system, you may also need to download and install some of the 3rd party required libraries. These are property of their respective owners, not CBIA. Original and actual versions can be free downloaded from the sources. We have decided to provide a ZIP archive with a collection of binaries we use under MS

Windows environment.

- 3rd party MS Windows development and run-time required binaries

Changes

6 January 2016

Feature: aside to the i3dcore and i3dalgo libraries, we now provide an image convertor between the i3dcore image representation class and the OpenCV representation class Mat.

- source file of the convertor

12 October 2015

Features: support for the latest MSVC 2015; implementation of various texture features (Tamura, MPEG7, Gabor, LBP).

- source files i3dcore
- source files i3dcore, i3dalgo and examples

17 June 2013

Features: support for the latest MSVC 2012; MSVC 2005 and MSVC 2008 no longer supported; new implementation of linear and nonlinear diffusion filters; steerable filtering based on the eigendecomposition of Hessian matrix; CMake configuration files simplified; bug fixes

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows (32bit), release version
- binary and header files for MS Windows (32bit), debug version

18 April 2012

Note: The last supported version in the MSVC 2010!

Features: full (read/write) support for PNG, DICOM and HDF5 file formats; new implementations of watershed algorithm; new morphological reconstruction algorithms; ellipse fitting algorithm; bug fixes

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows (32bit), release version
- binary and header files for MS Windows (32bit), debug version

1 October 2010

Features: 2D and 3D Haralick features; new topology-preserving implementations of fast level set methods; the implementation of topological watershed; morphological reconstruction based on the hybrid reconstruction algorithm; minimal build of both i3dcore and i3dalgo libraries can be done without any dependencies; support for the latest MSVC 2010; default warning level set to 3 in MS Windows; Java wrappers of the Bio-Formats library excluded from the sources

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows, release version
- binary and header files for MS Windows, debug version

13 April 2010

Note: The last supported version in the MSVC 2008!

Features: troubles with memory leaks solved & better memory management implemented; new watershed implementations; the implementation of new voxel based registration methods; type 'BINARY' substituted for type 'bool'; the decomposition of structuring elements (math. morphology); bugs in Fourier transform wrapper removed; new implementation of fast level set methods; in Windows only shared version (DLL) is currently supported

- source files i3dcore
- source files i3dcore, i3dalgo and examples

- binary and header files for MS Windows, release version
- binary and header files for MS Windows, debug version
- 3rd party MS Windows development and run-time required binaries for MSVC 2008

1 December 2009

Features: extended information loading from image files; mean shift filtering implemented; median filtering using the moving histogram technique implemented; fast convolution reimplemented in order to manage huge image data

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows, release version
- binary and header files for MS Windows, debug version

20 May 2009

Features: fast 3D convolution included.

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows, release version
- binary and header files for MS Windows, debug version

24 March 2009

Features: the first official release under the GNU GPL.

- source files i3dcore
- source files i3dcore, i3dalgo and examples
- binary and header files for MS Windows, release version

- binary and header files for MS Windows, debug version