

# Optical Flow downloads

Contributed by VladimĀr Ulman  
Last Updated Wednesday, 15 February 2012

This page summarizes download pointers to libraries and tools related to the computation of optical flow .

## Overview

We maintain the Optical flow collection of libraries and tools in the CBIA. Every library and tool is licensed separately under the GNU GPL.

Technically, the collection is one repository in the Subversion with several directories and files in its root, namely dirs:

- libi4d with the i4dcore library,
- libfilters with the LinearFilters library,
- libof with the OpticalFlow library
- tools with single-purpose command-line programs including the Ground-truth generator,
- ofd with the GUI demo program (see screen shot),
- doc with documentation

and files:

- CMakeLists.txt -- overall CMake configuration file,
- Doxyfile -- Doxygen configuration file,
- LICENSE with the copy of GNU GPL license,
- ChangeLog file,
- Documentation.txt about how to document new functions.

The doc directory has some images referenced from the documentation in the source codes initially. After issuing the command make doc, a recent updated documentation will be created in the doc/html directory. The documentation is generated from source codes over all libraries and tools using the Doxygen . Hence the boundaries between libraries in the documentation are somewhat fuzzy.

A copy of doc/html directory is mirrored at [http://cbia.fi.muni.cz/user\\_dirs/of\\_doc/index.html](http://cbia.fi.muni.cz/user_dirs/of_doc/index.html).

Configuration of compile-time options and compiling and linking parameters is driven by the CMake, see the next page.

{mospagebreak title=Requirements and Options}

## Dependencies

The dependency tree visual summarization is here.

Basicaly, one needs to have the i3dcore and some image fileformats manipulating libraries, which are required by the i3dcore. This is enough for the i4dcore to compile and link.

The LinearFilters library is based on top of the i4dcore but it further requires the LAPACK library.

The OpticalFlow library is based on top of the LinearFilters but it further requires the i3dalgo library.

The tool programs and the Ground-truth generator gtgen are linked against the OpticalFlow library (and libraries pulled in by it).

The same holds for the GUI demo program ofd which requires OpenGL and wxWidgets in addition.

The i3dcore and i3dalgo are both CBIA libraries currently available upon request. Installation manual for these is available here. LAPACK and fileformats libraries can be freely download elsewhere. Many of them are part of major operating systems by default.

## Grouping

We provide separate CMakeFiles.txt file, a CMake configuration file, for every library, the GUI demo and the tools directory. However, they are expected to be called from the overall CMakeFiles.txt which is located in the root directory. It is the top-level configuration file and should be the only one the CMake program is pointed to.

This top-level CMakeLists.txt supports the following compilation configurations:

- Only the i4dcore and LinearFilters libraries will be compiled and built when WITH\_FILTERS\_ONLY is set to ON. Otherwise the OpticalFlow library is built too. This (without OpticalFlow) is the default and minimalistic configuration supported, it creates two libraries.
- The GUI demo program ofd will be compiled when WITH\_GRAPHIC\_UTILS is set to ON. This automatically changes WITH\_FILTERS\_ONLY to OFF since the OpticalFlow library is required by the program too.
- The tools with the Ground-truth generator gtgen will be compiled when WITH\_COMMAND\_UTILS is set to ON. This also changes WITH\_FILTERS\_ONLY to OFF.
- Selecting between Debug and Release variant with DEBUG\_VERSION. The Debug variant pushes compiler to include some extra information about the code useful for debuggers and disables code-generating optimizations. Often some extra lines of code from the source files are compiled. These mostly produce report lines in the console. The Release variant does the opposite (thus produces smaller and faster binaries).
- Additionally, one may disable the use of certain 3rd party library, such as FFTW or LAPACK. However, differing from the default setup may lead to inability to compile or link certain library.

In other words: One may compile only lib4d and LinearFilters. One may enlarge the compilation batch by adding the OpticalFlow library to it. To these three libraries one may further add all tools with gtgen and/or GUI demo program ofd.

Configuration, compilation and installation manual is part of the documentation.

{mospagebreak title=Sources and ChangeLog}

Current version

The exported SVN repository of this optical flow related collection of libraries and tools can be downloaded here. If one is interested only in, say, the LinearFilters library, he or she may change the configure options via CMake to build only the desired library. The configuration options were explained on the previous page.

A copy of current version's documentation is available here . A sample commented source code for the Heeger's optical flow method is available here .

Note that we don't provide any warranty or similar. It is licensed under the GNU GPL. The libraries and tools belong to

the CBIA.

History of changes  
2nd February 2010

This "release" brings the preliminary implementations of the Heeger's method for computation of optical flow as well as its advanced version. The latter should provide better flow fields on biomedical image data. More on this in the docs.

The arbitrary Gabor bank filtering was under further development. Besides staged versions (modulation, Gaussian and demodulation), direct filtering versions (via 1D IIR Gabor convolutions) were added. Some bugs were removed. Special attention was given to filtering near image boundary in all variant of Gaussian and Gabor filtering in the library.

The optimized bank filtering was finished and is now considered stable. One should call the `PrepareGaborBank()` to compute "the convolution recipes" for the given Gabor bank. The recipes provide information for convolution routines how to convolve very effectively. The recipes can be then processed with `ApplyGaborBank2_DC()`.

Since even-phase Gabor filtering is influenced with the DC offset present in the filtered image, functions to either remove DC from input image or to convolve and remove DC from outputs are made available.

We won't provide binary versions of this release until the Heeger's method is complete. Sorry for that.

6th August 2009

Changes were only in the `LinearFilters` library, mostly only Gabor filtering related stuff has changed.

The most important is the change of the `GaborFilter` structure which original content proved not be a lucky choice. The structure's semantics have changed. Now it should encode users wishes about a motion to be detected. A precise specification of general 3D and 4D Gabor filters are now carried out with `GaborFilter3D` and `GaborFilter4D` structures. A conversion routine exists.

As a result many functions had to be changed. Fortunately, mostly only their interface has changed because they now listen to the `GaborFilter3D` structure.

Some new functions were implemented, among these: "Gabors in the Fourier spectrum" related functions, the `Staged4GaborFiltering()` for effective simultaneous convolution with 4 Gabor filters and `ApplyGaborBank()` for fast convolution with bank of Gabor filters. More on this in the new Gabor Filters module .

The `ApplyBaseConvolution2()` was added. It corrects bug a in the `ApplyBaseConvolution()`. The latter was decided to be kept even with the bug for the backward compatibility.

23rd March 2009

The first official release of the Optical Flow collection under GNU GPL.

The `i4dcore`, an extension of `i3dcore` to uniquely store and manipulate 3D timelapse images ( $4D = 3D + \text{time}$ ), was considered complete and production ready. It also defined an internal CBIA standard to storing a sequence of 3D images on a hard drive through functions `Image4d.ReadSequence()` and `Image4d.SaveSequence()`. The related portion of documentation is this module .

The `LinearFilters` library implemented 1D convolution routines along  $x, y, z$  and time axes as well as along arbitrary  $(dx, dy, dz)$  axis. These were targeted for spatial image filtering using separable convolutions. Most of these functions were modified to conduct Gaussian filtering with IIR filters. Finally, 3D Gaussian and Gabor filtering separated into three to six 1D convolutions was implemented in functions `ApplyANIGaussL()`, `ApplyANIGaussU2()` and `StagedGaborFiltering()`.

The `OpticalFlow` library includes PDE-based methods based on full bidirectional multi grid framework . Some image statistics functions are included in the library as well.

The development of the Ground-truth generator and the GUI demo was stopped, considered as beta though not final.

{mospagebreak title=Current Binaries}

Binaries of current version of the optical flow collection are made available here. You may need to register prior downloading if you are here for the first time.

Windows systems

The command-line tools, the Ground-truth generator `of_gtgenb` and the GUI demo program `of_ofd` all in release variant are available in ZIP archive, 9.9MB. The archive contains also the dynamic libraries (DLL files) required for the executables (EXE files) to run. One may also copy the DLLs to her system paths or update her system paths to make them system-wide available.

The header and .LIB files, the devel package, necessary for your programs to compile and link against the library are in another ZIP archive, 2.1MB.

Note that the former archive contains i3dcore and i3dalgo DLLs while the latter has their header files with LIBs. Apart from the optical flow libraries one also receives both i3dcore and i3dalgo (without source files) in this way.

Microsoft Visual Studio 2008 project, a .SLN file, with a sample source file, 3.4MB. You will probably need to adjust Additional Include Directories in the C/C++ -> General tab and Additional Library Directories in the Linker -> General tab of the project's properties depending on where you placed the content of the devel ZIP archive. You will also need the devel package.

older binaries

- release version of the library and tools: 03/2009, 08/2009

- headers and the library: 03/2009, 08/2009

Linux systems

It is expected that basic image fileformat libraries such as libtiff and libjpeg be already installed in your system. Thus, these were excluded from the following downloads.

The command-line tools, the Ground-truth generator of \_gtgenb and the GUI demo program of \_ofd all in release variant are available in TAR.GZ archive, 0.75MB. The binaries are linked against dynamic .SO i3dcore, i3dalgo and libraries from this collection. These together with their header files are available in another TAR.GZ archive, 2.4MB.

The latter archive has headers and dynamic library files of i3dcore, i3dalgo, i4dcore, LinearFilters and OpticalFlow libraries -- just ready to compile and link with the example, TAR archive, 0.23MB. It is expected in the example that the headers be installed in /usr/local/include and libraries in /usr/local/lib. Otherwise one may need to change the Makefile.

Regarding the examples in both systems. One may need to adjust the project setting or Makefile according to his/her system. Moreover, the example uses only the LinearFilters library. If the OpticalFlow library is desired, changes must certainly be made.

older binaries

- release version of the library and tools: 03/2009

- headers and the library: 03/2009