OMERO-2-OpenCV & OMERO-2-ITK

Emil Rozbicki

OME Meeting - 08.07.2014

University of Dundee
Wellcome Trust Centre for
Gene Regulation & Expression
College of Life Sciences, University of Dundee
Dundee, Scotland, UK

GLENCOE SOFTWARE
Seattle, WA, USA

OpenCV

What is OpenCV?

- designed with a strong focus on real-time applications
- written in optimized C/C++, can take advantage of multi-core processing
- enabled with CUDA and OpenCL
- C++, C, Python and Java interfaces
- free, cross-platform, released under a BSD license

Who uses OpenCV:

- Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota
- many many startups

Use cases:

- stitching streetview images
- detecting intrusions
- monitoring mine equipment
- helping robots navigate and pick up objects
- running interactive art
- checking runways for debris
- inspecting labels on products in factories
- rapid face detection

47K people of user community, 7 million downloads (200K/month)

What is OpenCV?

Modules:

- **core** - basic data structures and basic functions used by all other modules
- **imgproc** - linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on
- **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction
- **video** - motion estimation, background subtraction, and object tracking algorithms
- **features2d** - salient feature detectors, descriptors, and descriptor matchers
- **objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on)
- **highgui** - interface to video capturing, image and video codecs, as well as simple UI capabilities
- **gpu** - GPU-accelerated algorithms from different OpenCV modules

itk

Kitware

What is Insight Toolkit (ITK)?

- implemented in C++ using templated code (generic programming)
- open-source software toolkit for performing registration and segmentation (Apache 2.0 license)
- primarily develop for medical instrumentation as CT, MRI or ultrasound scanners
- cross-platform, using the CMake build environment
- supports multiple language bindings, including Python and Java
- multi-threaded (shared memory) with parallel processing
- does not address visualization or graphical user interface

Known users:

- National Library of Medicine (NLM)
- National Institute of Dental and Craniofacial Research (NIDCR)
- National Science Foundation (NSF)
- National Eye Institute (NEI)
- National Institute of Neurological Disorders and Stroke (NINDS)
- National Institute of Mental Health (NIMH)
- National Institute on Deafness and Other Communication Disorders (NIDCD)
- National Cancer Institute (NCI)

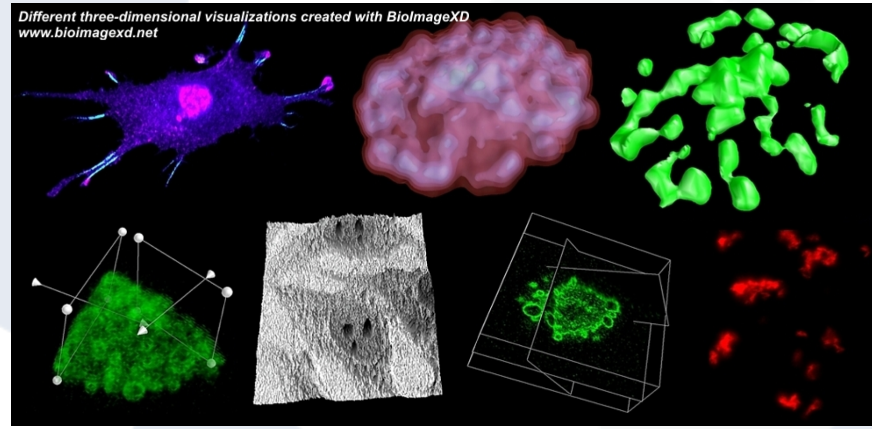
12,600 web visits each month.

What is Insight Toolkit (ITK)?

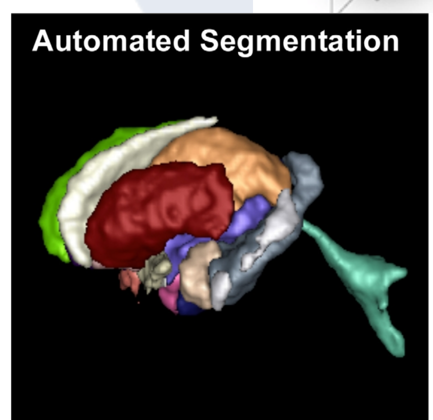
VTK - Visualisation Toolkit

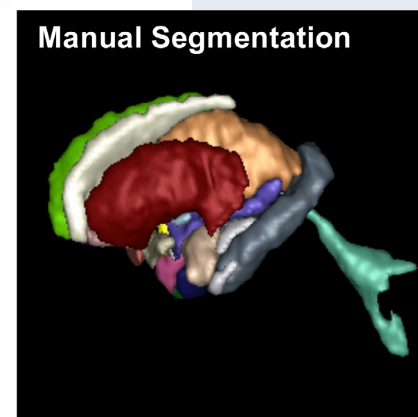
- 3D computer graphics
- 3D modeling
- 3D image processing
- volume rendering
- scientific visualization
- information visualization
- free and open source
- wrapping of the C++ core into Python and Java

BioImageXD



3DSlicer





Known users:

- Los Alamos National Lab (LANL)
- National Library of Medicine (NLM)
- National Alliance for Medical Image Computing (NA-MIC)
- Department of Energy (DOE)
- ASC Program Sandia National Laboratories
- Army Research Laboratory (ARL)

ITK Examples, VTK Examples

OMERO-2-ITK

&

OMERO-2-OpenCV

OMERO-2-OpenCV

Class List

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

[detail level 1 2]

N sha1	
▼ N simple_omero	
C logger	
C connector	Simple connection to OMERO server
C image	Simple image access

Class List

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

[detail level 1 2]

▼ N omero2cv	
C type_converter	OMERO to OpenCV Image type converter
C plane_store	Basic building block for data storage in OpenCV format. Stores 3D stack as cv::Mat vector
C channel_store	
C image_store	
C image	Read/Write images to and from OMERO in OpenCV format

SimpleOMERO

Connector Class

```
simple_omero::connector *Omero = new simple_omero::connector();
Omero->connect("server", "port", "user", "password");
```

Image Class

Public Member Functions

image (const omero::api::ServiceFactoryPrx &session, const int &image_id)
SimpleOMERO image contructor for Retriving image from OMERO. More...

image (const omero::api::ServiceFactoryPrx &session, const int &dataset_id, const omero::model::PixelsTypePtr &**pixel_type**, const int &width, const int &height, const int &depth, const int &**number_of_channels**, const int &**number_of_timepoints**, const std::string &**name**, const std::string &**description**, const double &**pixel_size_x**, const double &**pixel_size_y**, const double &**pixel_size_z**)
SimpleOMERO image contructor for Creating image in OMERO. More...

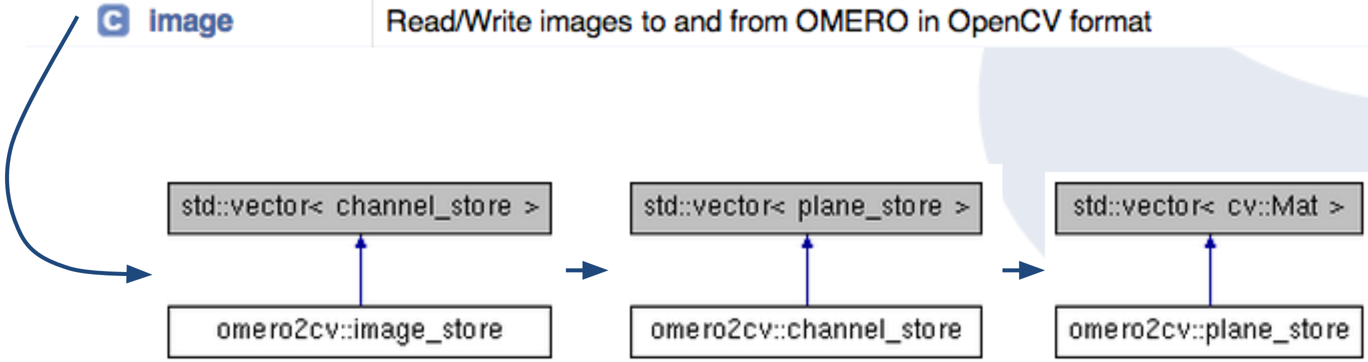
```
int image_id = 1;
simple_omero::connector *Omero = new simple_omero::connector();
Omero->connect("server", "port", "user", "password");
simple_omero::image *image = new simple_omero::image(
    Omero->get_session(), image_id
);
```

OMERO2CV & OMERO2ITK

Class List

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

▼ N omero2cv	
C type_converter	OMERO to OpenCV Image type converter
C plane_store	Basic building block for data storage in OpenCV format. Stores 3D stack as cv::Mat vector
C channel_store	
C image_store	
C image	Read/Write images to and from OMERO in OpenCV format




```
// Display the image using the OpenCV HighGui class.
int plane_index_to_show = 0;
cv::namedWindow( "Display window", CV_WINDOW_NORMAL | CV_WINDOW_KEEPRATIO);
cv::imshow("Display window", image->pixel_store.t(0).c(0).z(plane_index_to_show));
cv::waitKey(0);
```

OMERO2CV & OMERO2ITK

```
// Connect to an OMERO server to Read and Write Images.
simple_omero::connector *Omero =
    new simple_omero::connector("localhost", "4064", "user", "pass");
int image_id = 1;

/// Intitialise a new omero2cv::image object to read the data.
omero2cv::image *image =
    new omero2cv::image(Omero->get_session(), image_id);

// Allocate the Pixel Store to read only a single 3D stack.
int timepoint = 4; // Choose a time point to read from the time laps data.
int channel = 2; // Choose a channel to read from the multi-channel data.
std::vector<int> timepoint_list;
timepoint_list.push_back(timepoint);

std::vector<int> channel_list;
channel_list.push_back(channel);

std::vector<int> plane_list;
for (int z = 0; z < image->size_z; z++) { // Read the whole stack or a subset.
    plane_list.push_back(z);
}

image->allocate_pixel_store(timepoint_list, channel_list, plane_list);

// Read the image from the server to the memory.
image->read_image();

// Do some operations.
...

// Delete the object.
delete image;
```

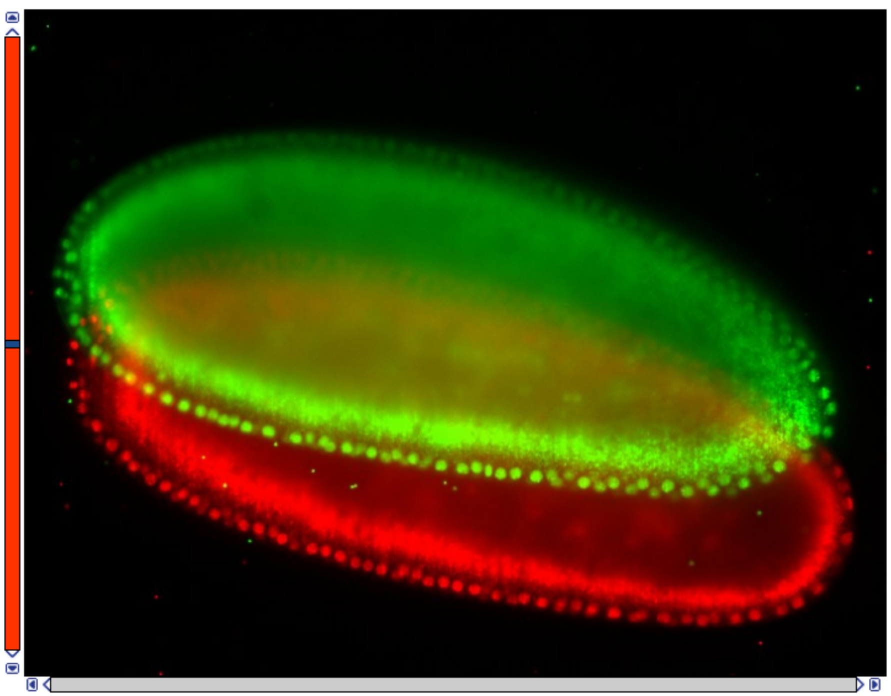
Registration of multi-view light-sheet (SPIM) data with OMERO and OpenCV

Registration of multiview light-sheet (SPIM) data

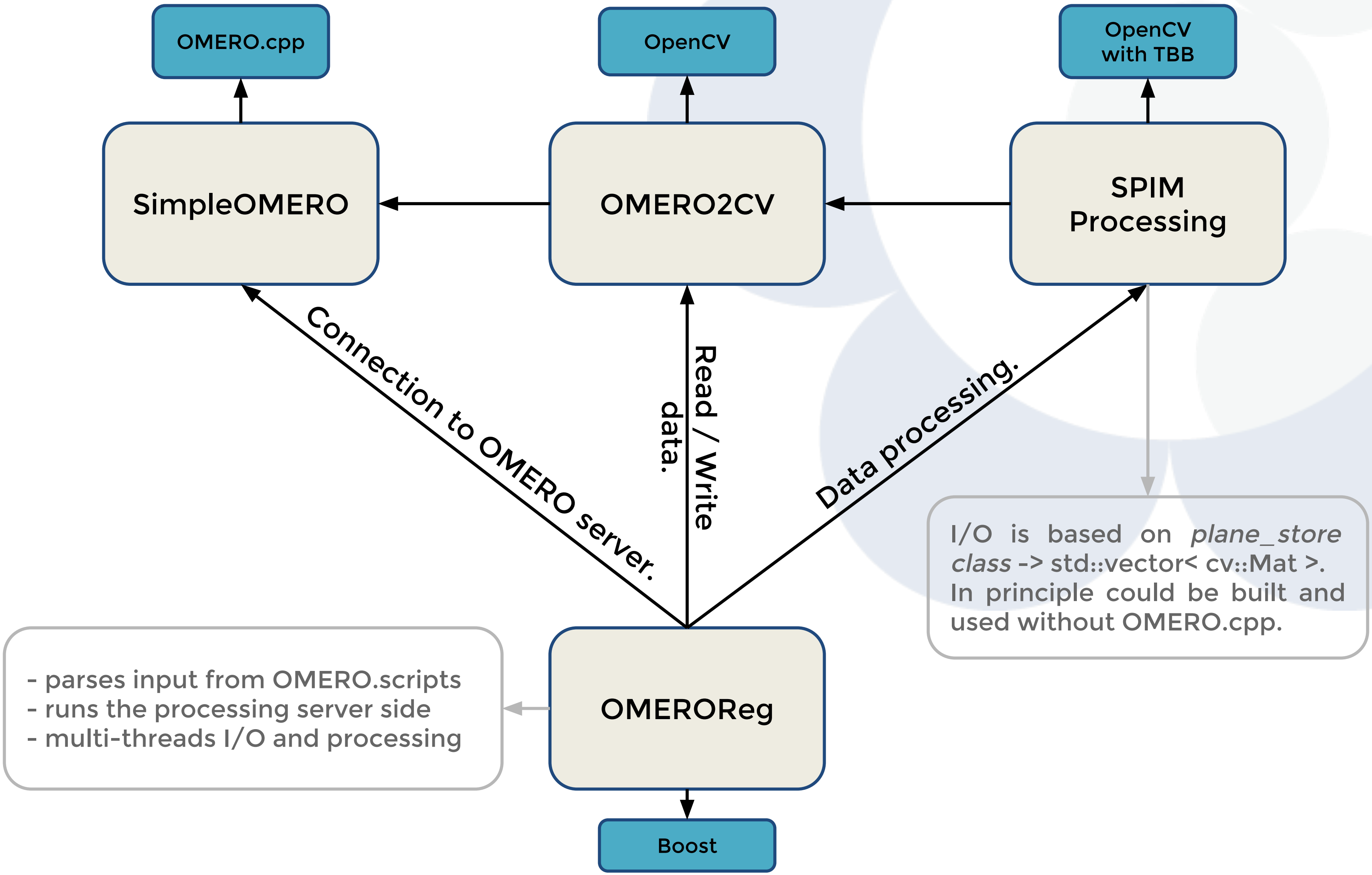
- **S. Preibisch, S. Saalfeld, J. Schindelin and P. Tomancak (2010)** "Software for bead-based registration of selective plane illumination microscopy data", *Nature Methods*, **7**(6):418-419.

Processing Steps:

1. Filter data to remove "fake beads" (Difference of Means based on 3D integral image).
2. Find Maxima (beads) in 3D.
3. Construct local bead descriptors. (kd-tree search).
4. Find corresponding beads in different views. (kd-tree search).
5. Estimate affine transform between the views. (RANSAC).
6. Aligning views by remapping pixels according to transform from point 5. (fast pixel access).



Registration of multiview light-sheet (SPIM) data



THANK YOU!