# imglib2 network transfer of multiple images

Vladimír Ulman

CSBD and MPI-CBG

19th Oct, 2017
DAIS wp 1.3

The document is valid as of commit TBA (19th Oct, 2017)

# The Goal of DAIS wp 1.3 (recap)

- Direct transmission of the imglib2 data between two peers
- Peers are assumed to be two independent processes
- Processes may:
  - live on the same machine
  - live on machines connected over TCP/IP network
  - any app, e.g., FIJI, KNIME, wrapped C/C++/Python library
- Inspired by the BlueTooth principles
- Uses proprietary/own protocol:
  - text header and binary voxel data
  - *similar to ICS (Image Cytometry Standard)*
  - default TCP port 54545
  - *no collision in /etc/services*
- Networking is done via ZeroMQ package.

# Multiple images extension

- This document describes an extension of the single image tranfer to support sequential transfer of multiple images

- The single image transfer is initiated with "v1" header
- Multiple image transfer is encapsulating the original single-image protocol
- Multiple image transfer is recognized with optional "v0" header

- The "v0" header is a service part of the transfer protocol
- The "v0" header is single-message transfer

- The "v1" header is a data-transfer part of the transfer protocol
- The "v1" header is followed by multi-messages transfer

# Multiple images extension

A sends two images to B, protocol view:

- A: sends "v0 expect 2 images"
- A: sends "v1 dimNumber...." and follows single image protocol for the 1st image
- B: initially it waits for "v0 expect..." and populates ImgTransfer.expectedNumberOfImages, else complains
- B: waits for "v1 dimNumber" and follows single image protocol for the 1st image
- B: immediately waits for any message from sender
- A: sends "v0 expect 2 images"
- B: got message from sender, updates ImgTransfer.allTransfered = false
- A: "v1 dimNumber...." and follows single image protocol for the 2nd image
- B: waits for "v1 dimNumber" and follows single image protocol for the 2nd image
- B: immediately waits for any message from sender
- A: sends "v0 hangup"
- B: got message from sender, updates ImgTransfer.allTransfered = true

# Multiple images extension

A sends two images to B, programming view:

- A sends one image in one function call
- Call ends once the image was transfered
  - A: sends "v0 expect 2 images"
  - A: sends "v1 dimNumber...." and follows single image protocol for the 1st image
  - A: (waits until it hears "done" from B)
- Must call hangUpAndClose() to allow last image to be received!

- B receives one image in one function call
- Call ends after either "hangup" was received,
  or transfer of a next image has begun
- Call returns always with an image (or with an exception)
  - B: initially it waits for "v0 expect..." and populates ImgTransfer.expectedNumberOfImages, else complains
  - B: (this above is not done for second and later function call)
  - B: waits for "v1 dimNumber" and follows single image protocol for the 1st image
  - B: (sends "done")
  - B: immediately waits for any message from sender

# The API of DAIS wp 1.3

How to use it from Java:

```
import de.mpicbg.ulman.imgtransfer.ImgTransfer;
[ import de.mpicbg.ulman.imgtransfer.ProgressCallback; ]

ImgPlus<?> imgP = ...;
int imgsNo = 2;

A_sendsTo_B:
A: ImgTransfer A = new ImgTransfer("tcp://"+remoteURL, imgsNo, timeoutTime[, logger]);
A: A.sendImage((ImgPlus) imgP);
A: A.sendImage((ImgPlus) imgP);
A: A.hangUpAndClose();

B: ImgTransfer B = new ImgTransfer(portNo, timeoutTime[, logger]);
B: while (B.isThereNextImage()) imgP = B.receiveImage();
B:
B: after first image: imgsNo = B.getExpectedNumberOfImages();
B: after the while(): the connection is automatically closed
```

– or –

```
A_downloadsFrom_B:
//similarily, symetrically...
```

# The API of DAIS wp 1.3

- Multi-image is no longer using static methods
- Objects must be created
- No connection is initiated at all until the first transfer

- There are 4 constructors, one for one transfer mode
- Methods cannot be mixed, there's 1:1 enforced between available methods and used constructors

- Logger is optional
- Logger must implement ProgressCallback.info(String) and ProgressCallback.setProgress(0≤float≤1)

# The Control Flow of API of DAIS wp 1.3

- All 4 (state-aware, non-static) transfer functions should finish either with:
  - success after image was for sure sent or received
  - exception due to timeout (no functional connection was established)
  - exception due to other error (e.g. socket issues, protocol error)
- Currently supported:
  - what ever is supported with inside "v1 header", that is:
  - what ever is supported for single image transfer