Design document ‘Pathology Image Processing Libraries and Workstation’

This document describes the general design of the pathology libraries components, the design of the workstation and the detailed description of the classes and components.

Library Components

Core

The core library contains all the utility functionality (conversion of strings, handling of file paths, etc.) that is generally used by other libraries. The second part of this library is the presence of the base class for image sources (both readers and filters).

Content

- Stringconversion
  - Methods to parse/convert std::strings
- Filetools
  - Methods to (cross-platform) handle file system interactions (copy, delete, create directories, etc.)
- PathologyEnums
  - Header files which declares all the enumerations used in the pathology project
- ImageSource
  - Base class for modules that provides images. This class implements all the functionality to obtain information on the ImageSource related to its output (colortype, datatype, dimensions, samples per pixel)

Dependencies

The core library has dependencies on the boost libraries.

IO

The IO library contains the functionality for reading and writing multi-resolution pathology images. Some functionality is implemented ourselves (LIF, VSI), but most of it comes through wrapping of third-party libraries like OpenSlide

Content

- MultiResolutionImage
  - Base class for MultiResolutionImage’s, all vendor specific images are derived from this class. It implements a template function to obtain image data, regardless of the data type of the slide file itself. Finally, it contains a cache for loaded tiles.
- MultiResolutionImageReader/Writer
  - Classes to either read a MultiResolutionImage (instantiates the correct derived class for loading of the image), or write it to disk (as a multi-resolution, tiled TIFF)
- TileCache
  - A caching class which implements a tile cache to enable faster image loading when the same tiles are loaded repeatedly.
- JPEG2000Codec

    - Wrapper around libjasper to encode or decode individual tiles with the JPEG2000 compression algorithm
- VSIImage/TIFFImage/OpenSlideImage/LIFImage
    - Specific implementations of MultiResolutionImage for different file types, implements the initialize and readRegion member functions of MultiResolutionImage.

Dependencies

IO has a large number of third-party dependencies due to the fact that it needs to support several compression methods (ZIP through libz, JPEG2000 through libjasper, JPEG through libjpeg, DCMTK for lossless JPEG, LZW through libtiff). Furthermore, libtiff is used for both reading and writing of multi-resolution TIFF. Last, OpenSlide is used to add support for several proprietary fileformats (e.g. MRXS, SVS, SCN).

Image processing

MeVisLab

Contains MeVisLab-module version of code implemented in the other libraries. For example, it implements the color deconvolution methods and image IO in MeVisLab modules.

Content

- DIAGColorDeconvolution
    - Modules which functions as a wrapper around the color deconvolution filter implemented in imageprocessing
- DIAGMultiResolutionImageReader/Writer
    - Wrappers around the IO library to provide multi-resolution image access in MeVisLab

Dependencies

One extra dependency is introduces, the dependency on MeVisLab.

Executables

Contains commandline executables which use code from the pathology library components to perform specific tasks.

Content

- MultiResImageConverter
    - Implements conversion from proprietary formats to multi-resolution tiled TIFF
- PatholTestTunner
    - Runs the unittests for the pathology libraries

Dependencies

No extra dependencies imposed, dependencies of individual executables are fully determined by the pathology libraries they use.