Introduction to OSSIM C++ Library

Disclaimer

- Though SSC Pacific makes every effort to perform quality assurance on its training materials, the material in this presentation may inadvertently include technical inaccuracies or other errors. We would be grateful if users notify us of any errors or inaccuracies they may find.
- The presentation contains references to links and to third-party websites. These are provided for the convenience and interest of users and this implies neither responsibility for, nor approval of, information contained in these websites on the part of the U.S. Government. The USG makes no warranty, either express or implied, as to the accuracy, availability or content of information, text, graphics in the links/third party websites. The USG has not tested any software located at these sites and does not make any representation as to the quality, safety, reliability or suitability of such software, nor does this presentation serve to endorse the use of such sites.

Overview of Talk

- Introduction to OSSIM Source Code (getting source code, building source code)
- OSSIM Source Code Examples:
  - Ossim TileToIplImage example – how to use OpenCV in an ossimFilter.
  - ossim-image-info – the image handler, geometry, and projection.
  - ossim-mosaic – mosaic one or more images
  - ossim-mask-filter for land masking
- Ossim Image Data to IplImage Example – issues without tiling
- Conclusions

Introduction to OSSIM Source Code

- Over 1 million lines of C++ code
- Perform remote sensing/image processing functions
- Handles very large images (> 6GB)
- Handles wide range of data types (nitf, tif, jpeg, HDF, .xml) as well as .shp files
- Capable of performing wide range of remote sensing tasks
- Can be extended to increase functionality (ie writing plugins, writing ossim Filters, etc)

OSSIM Source Code Overview

- Break very large images up into tiles (typically 512x512)
- Just as with ImageLinker, we have image chains in code. We always start with an image handler and often finish with an image writer
- Using OSSIM filters, we can perform land masking, image mosaicing, edge detection, and many other functions
- OSSIM source code can easily be extended as we will show during our next lesson – writing OSSIM filters

Systems Center PACIFIC

OSSIM Overview

OSSIM

(Open Source Software Image Map)

Supports many different file formats

Allows us to open satellite images!!

Ortho and Geo rectify images

Basic image processing techniques

Works with very large images

Mosaic/Merge

Tile images ≥ TBs

C/C++ Library

Image Linker

OSSIM Planet

OMAR

OSSIM Open Source Software Links

Open Source Software Image Map (OSSIM)

Download OSSIM Binary Installer: http://download.osgeo.org/ossim/ossimplanet-1.7.15-minimal.exe (download the minimal Windows installer)

OSSIM Project website: http://www.ossim.org
OSSIM Wiki: http://trac.osgeo.org/ossim/
OSSIM Project tutorials: http://download.osgeo.org/ossim/tutorials/
OSSIM API Documentation: http://trac.osgeo.org/ossim/doxygen/
OSSIM Mailing list: https://lists.sourceforge.net/lists/listinfo/ossim-developer

OSSIM Mailing List

OSSIM Mailing list: https://lists.sourceforge.net/lists/listinfo/ossim-developer

Importance of signing up for mailing list:

1) OSSIM is always under development and as such is always changing
2) Developers monitor the list and answer questions about developing using OSSIM

OSSIM Resources

List of the required support libraries for OSSIM

libjpeg v 7 - http://www.ijg.org/libgeotiff 1.2.5 - http://trac.osgeo.org/geotiff/gdal 1.7.0 - http://trac.osgeo.org/gdal/wiki/DownloadSourcetiff 3.8.2 - http://www.libtiff.org/OpenCV v 2.1.0 - http://opencv.willowgarage.com/wiki/OpenThreads 2.9.5 - http://www.openscenegraph.org/projects/osg - OpenThreads comes as part of OpenSceneGraph.-

OSSIM source code - http://www.ossim.org/OSSIM/OSSIM_Home.html - The components we build from OSSIM are:
- OSSIM
- ossim_plugins

OSSIM Dependencies

- **Windows:**
  - http://download.osgeo.org/ossim/dependencies/windows_vcexpress2008/ (FOR VISUAL STUDIO 2008 ONLY)
  - The ossim-3rd-party-vs2010 folder contains the Visual Studio edition of these files

- **Linux/Fedora**
  - yum install svn cmake gcc-c++ qt-devel opencv-devel libgeotiff-devel libjpeg-devel libtiff-devel OpenSceneGraph-devel gdal gdal-devel zlib-devel openmpi-devel minizip-devel libcurl-devel libcurl expat-devel expat yasm libtool postgis mapserver

Getting the OSSIM Source Code

- The OSSIM source code is hosted on an SVN server located at:
  - http://svn.osgeo.org/ossim/trunk
- Basics of SVN
  - http://subversion.apache.org/
  - http://tortoisesvn.net/downloads.html
  - Common commands:
    - Checkout code: svn co <source-location> <destination> [-r]
    - Update code: svn update [-r]
    - Difference code: svn diff
- To get OSSIM:
  - svn co http://svn.osgeo.org/ossim/trunk .

Building the OSSIM Source Code

- OSSIM is built using CMAKE
- Basics of CMAKE
  - http://www.cmake.org
  - Simple platform and compiler independent configuration files.
  - Generates native makefiles and workspaces that can be used in the compiler environment of your choice. (Microsoft Visual Studio/GNU Makefiles)
  - Out-of-source builds
- CMAKE Configuration Files:
  - Examples located at:
  - ossim_install\build_scripts:
    - build_ossim_vs10.bat
    - ossim-package-cmake-config-vs10-nmake-v1.bat

Building the OSSIM Source Code

- Let’s take a look at the CMAKE file we used
  - ../ossim_trunk/ossim_package_support/cmake/build_scripts/windows/ossim-package-cmake-config-vs10-nmake-v1.bat

Generator – can make Nmake Makefiles, or Unix Makefiles as well. See CMAKE documentation for more information.

- Interesting Lines
  - -G “Visual Studio 10” \
  - -DBUILD_OSSIM=ON^
  - -DBUILD_OSSIM_PLUGIN=ON ^
  - -DBUILD_OSSIMGDAL_PLUGIN=ON^

Turn ON or OFF the compilation of certain sections of code.

Building the OSSIM Source Code

- Steps for building OSSIM
  - Install OSSIM dependencies
  - Get source code using SVN
  - Create a build directory
  - Copy build scripts into the build directory (and modify)
  - Run build_ossim_vs10.bat (Produce the makefiles/Visual Studio solutions)
  - From a command prompt cd to your build directory and run make (or open Visual Studio solution)

Configuring OSSIM

- OSSIM is configured by a preferences file
  - This file is specified by the environment variable OSSIM_PREFS_FILE
    - An example file is provided in the ossim source code at:
    - ../ossim_trunk/ossim/etc/templates/ossim_preferences_template
  - This file tells OSSIM where to find:
    - Elevation data
    - Plugins
    - Projection database files
  - Sets other things like
    - Cache size
    - Number of processing cores to use
    - Tile size to use for processing

Systems Center PACIFIC

Configuring OSSIM

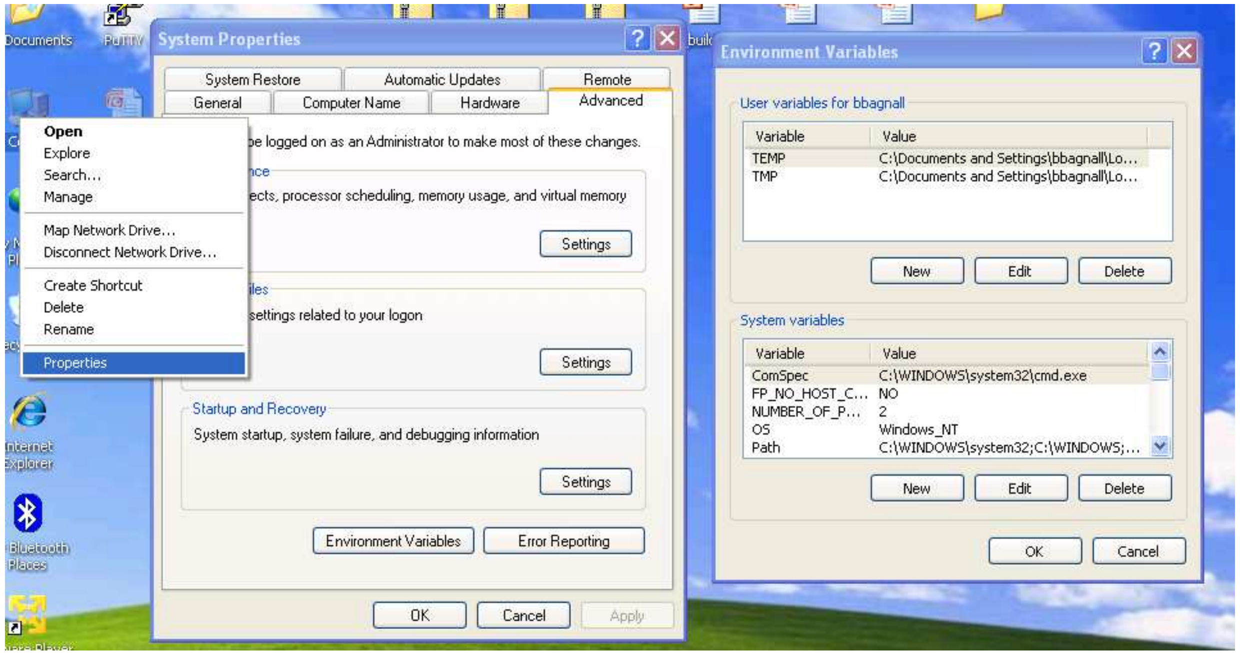
- The entries for the plugins look like:
  - plugin.file1: $(OSSIM_INSTALL)/bin/lossimkakadu_plugin.dll
  - plugin.file2: $(OSSIM_INSTALL)/bin/ossim_plugin.dll
  - plugin.file3: $(OSSIM_INSTALL)/bin/ossimgdal_plugin.dllProjection
- The entries for the elevation look like:
  - elevation_manager.elevation_source1.connection_string: $(OSSIM_DATA)/elevation/srtm1//elevation_manager.elevation_source1.type: srtm_directory
  - elevation_manager.elevation_source1.min_open_cells: 25
  - elevation_manager.elevation_source1.max_open_cells: 50
  - elevation_manager.elevation_source1.memory_map_cells: true
  - elevation_manager.elevation_source1.geoid.type: geoid1996

Configuring OSSIM

- OSSIM is configured by a preferences file
  - Can set the preferences file by using a bat file and typing:

set OSSIM_PREFS_FILE=c:\libraries\ossim\ossim_preferences

  - Or you can set an environment variable using the My Computer->Properties->Advanced->Environment Variables



be logged on as an Administrator to make most of these changes.

ects, processor scheduling, memory usage, and virtual memory

settings related to your logon

Startup and Recovery

System startup, system failure, and debugging information

User variables for bbagnall

Variable	Value
TEMP	C:\Documents and Settings\bbagnall\Lo...
TMP	C:\Documents and Settings\bbagnall\Lo...

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;...

OSSIM Basics

- OSSIM is separated into groups of related code
  - Base
    - Contains all basic OSSIM classes that other OSSIM functions use
  - Imaging
    - Contains classes for filtering images and using the image data
  - Projection
    - Contains classes for projecting satellite imagery to convert between pixel coordinates (pixel location) and geographic
  - Elevation
    - Contains classes for using elevation data with OSSIM

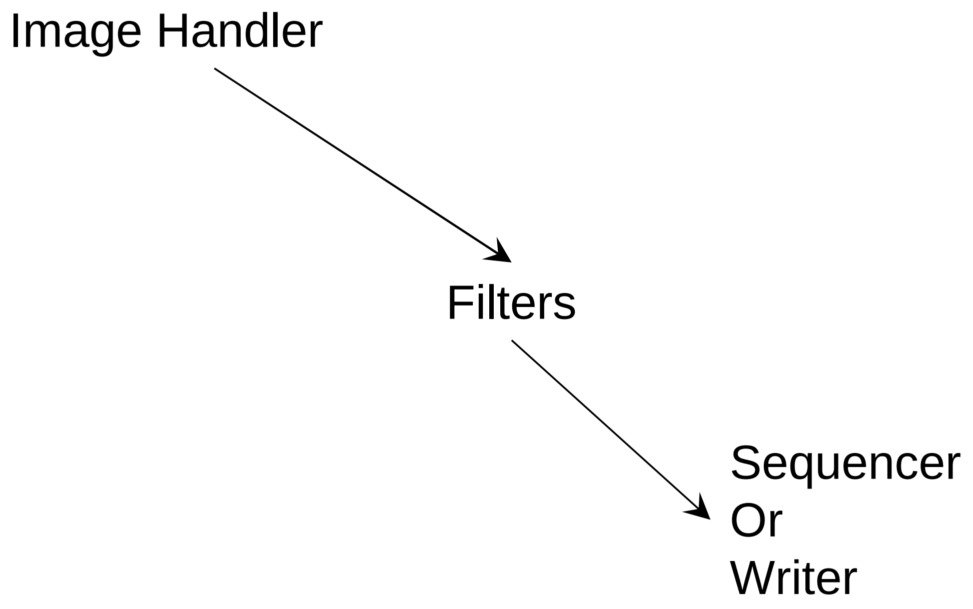
OSSIM Basics

- OSSIM is separated into groups of related code
  - Plugins
    - These extend the functionality of OSSIM by linking it with other libraries
      - Ossim_plugin
        - For reading SAR imagery
      - Osism_gdal_plugin
        - For reading many other types of data
      - Ossim_kakadu_plugin
        - For reading JPEG2000 imagery

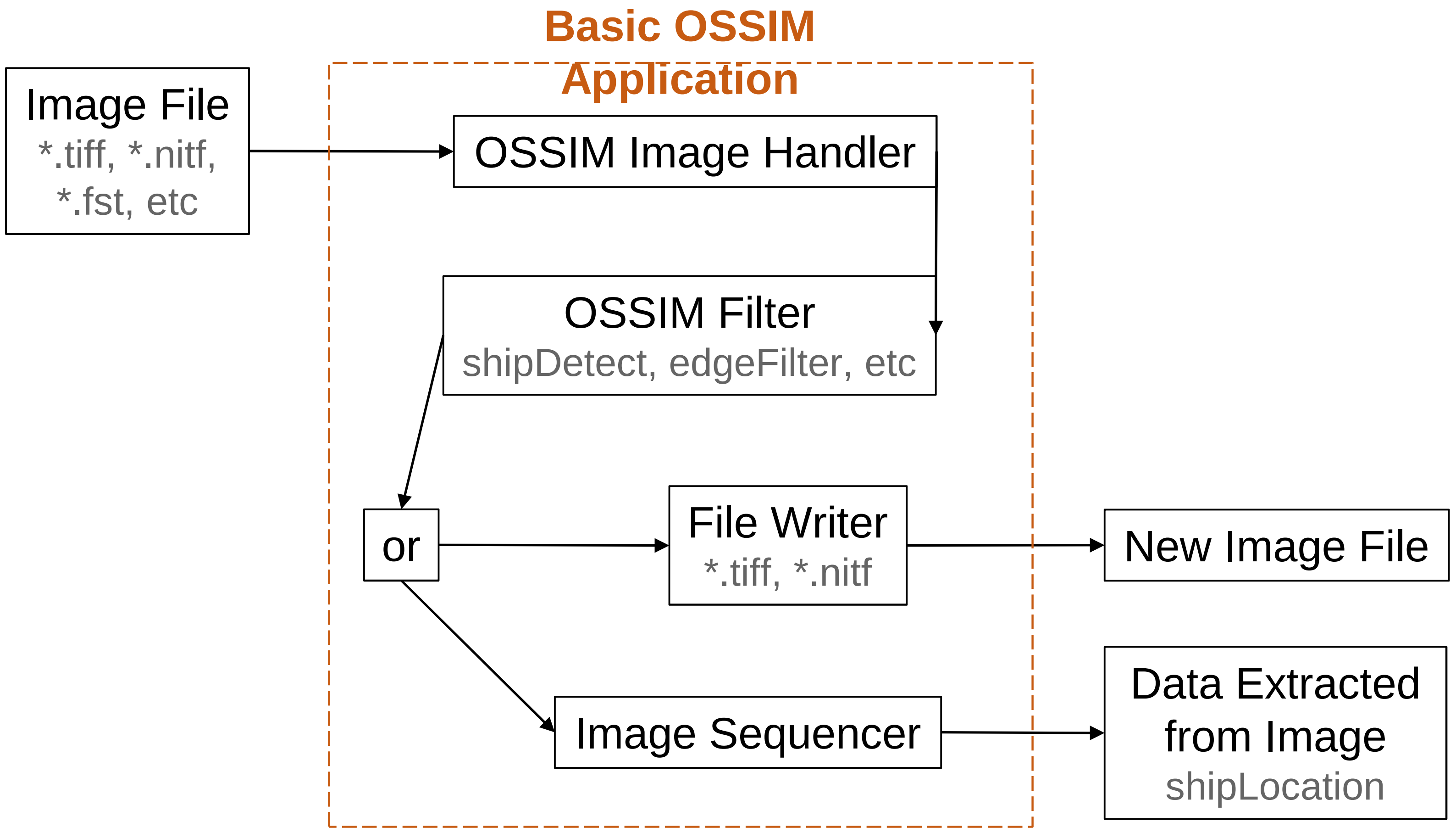
OSSIM Basics

- OSSIM comes with many basic data types to make your life easier
  - ossimString – With many helpers like toDouble
  - ossimGpt – Lattitude/Longitude
  - ossimDpt
  - ossimPolyon – With many helpers like isWithin
  - ossimQuaternion

OSSIM Basics



OSSIM Basics



Let’s look at some OSSIM code!

OSSIM Example-
ossim-info

ossim-info Example

- Navigate to the source code found at:
  - //opt/alpha/ossim/ossim_trunk/ossim/src/apps/ossim-info/ossim-info.cpp
    - ossimInit::instance()->initialize(); --- needed in every ossim application
    - ossimRefPtr<ossimInfo> --- used for garbage collection
    - ossimNotify(ossimNotifyLevel_WARN) --- used for displaying messages of varying severity
- Navigate to the binary found at:
  - //opt/alpha/ossim/ossim_planet_build/bin/ossim-info
- Run ossim-info on an image:
  - cd /opt/alpha/ossim/ossim_planet_build/bin
  - ./ossim-info /opt/alpha/Day\ 2\ Images/San\ Francisco\ Example/TerraColor_SanFrancisco_US_15m.tif

Other ossim example code and applications can be found in:

ossim_trunk/ossim/src/examples
ossim_trunk/ossim/src/apps
ossim_trunk/ossim/src/test

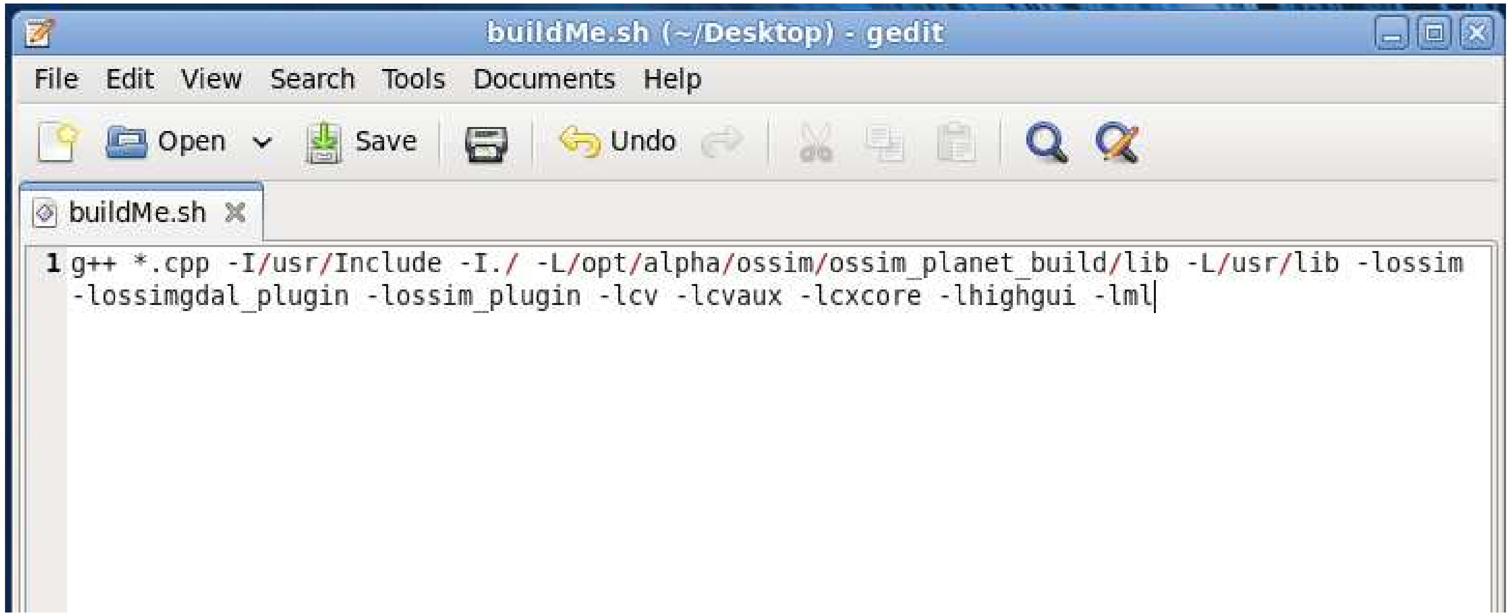
OSSIM Example-
ossim-mosaic

ossim-mosaic Example

- Navigate to the source code found at:
  - //opt/alpha/ossim/ossim_trunk/ossim/src/apps/ossim-mosaic/ossim-mosaic.cpp
- Navigate to the binary found at:
  - //opt/alpha/ossim/ossim_planet_build/bin/ossim-mosaic
- Run ossim-info on an image:

Adding Additional Include Directories

- Modify the `buildMe.sh` script that we wrote earlier
- Add -L/opt/alpha/ossim/ossim_planet_build/lib
- Add -lossim -lossimgdal_plugin -lossim_plugin –lcvaux -lml


buildMe.sh (~/Desktop) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
buildMe.sh

```
g++ *.cpp -I/usr/Include -I./ -L/opt/alpha/ossim/ossim_planet_build/lib -L/usr/lib -lossim
-lossimgdal_plugin -lossim_plugin -lcv -lcvaux -lcxcore -lhighgui -lml
```

Systems Center PACIFIC

OSSIM Example- ossim-mask-filter-test

ossim-mask-filter-test Example

- Source:
  - **//opt/alpha/ossim/ossim_trunk/ossim/src/test/ossim-mask-filter-test.cpp**
  - ossimTimer::instance()->setStartTick();
  - ossimImageHandlerRegistry::instance()->open(inputImgName);
- Binary:
  - //opt/alpha/ossim/ossim_planet_build/bin/ossim-mask-filter-test

- ./ossim-mask-filter-test 4 0 /mnt/hgfs/My\ Documents/chile/TerraColor_SanFrancisco_US_15m.tif /mnt/hgfs/My\ Documents/chile/TerraColor_SanFrancisco_US_15m.shp /home/alpha/work.tiff

Reading in images, converting from OSSIM data to OpenCV IplImages

Converting OSSIM ImageSource to an OpenCV IplImage

Next, we look at how to convert from an OSSIM image to an OpenCV IplImage without tiling the image using OSSIM.

The example code we'll discuss is found in /Lesson03_OpenCV_and_OpenCV_with_blobs/examples/OSSIM_To_OpenCV/OSSIMToOpenCVImage

Open the OSSIMtoOpenCVImage.cpp file

Converting OSSIM ImageSource to an OpenCV IplImage

Questions –

1. What are some possible limitations to this method?
2. Does this solve our problem of “very large images”.

OSSIM Image Tiling and converting to IplImages

Next, we look at how to tile an image in OSSIM, and to take each tile image and convert it to an OpenCV IplImage.

This is the basic premise behind virtually all filters that you can create.

/opt/alpha/OSSIM Lessons & Examples/Lesson04_Building_OSSIM_and_OSSIMc++lib/examples/ossim_filter_TileToIpl

./a.out /opt/alpha/Day 2 Images/San Francisco Example/TerraColorimage.tiff

If It can't find libossim.so
export LD_LIBRARY_PATH=/opt/alpha/ossim/ossim_planet_build/lib

build_ossim_vs10.bat

```
@echo off

set OLDPATH=%PATH%

call "C:\Program Files\Microsoft Visual Studio 10.0\VC\vcvarsall.bat"

set PATH="C:\Program Files\CMake
2.8\bin";C:\libraries\qt\v4_7_3\bin;%PATH%
set QTDIR=C:\libraries\qt\v4_7_3
set QMAKESPEC=C:\libraries\qt\v4_7_3\mkspecs\win32-msvc2010

call ossim-package-cmake-config-vs10-nmake-v1.bat

set PATH=%OLDPATH%
set OLDPATH=
```

ossim-package-cmake-config-vs10-nmake-v1.bat

```
set common_dir="c:/libraries/ossim-3rd-party-vs2010"
set common_lib=%common_dir%/lib/win32
set build_dir="c:/libraries/ossim/trunk"
set pkg_dir="c:/libraries/ossim/trunk"

cmake -G "Visual Studio 10"^
 -DWIN32_USE_MP=ON^
 -DBUILD_CSMAPI=OFF^
 -DBUILD_LIBRARY_DIR=lib^
 -DBUILD_OMS=OFF^
 -DBUILD_OSSIM=ON^
 -DBUILD_OSSIM_PLUGIN=ON^
 -DBUILD_OSSIMCONTRIB_PLUGIN=ON^
 -DBUILD_OSSIMCSM_PLUGIN=OFF^
 -DBUILD_OSSIMGDAL_PLUGIN=ON^
 -DBUILD_OSSIMKAKADU_PLUGIN=ON^
 -DBUILD_OSSIMLIBRAW_PLUGIN=OFF^
 -DBUILD_OSSIMMRSID_PLUGIN=OFF^
```

ossim-package-cmake-config-vs10-nmake-v1.bat Continued

```
-DBUILD_OSSIMNDF_PLUGIN=OFF^
-DBUILD_OSSIMNUI_PLUGIN=OFF^
-DBUILD_OSSIMPNG_PLUGIN=OFF^
-DBUILD_OSSIMREGISTRATION_PLUGIN=OFF^
-DBUILD_OSSIMQT4=ON^
-DBUILD_OSSIM_MPI_SUPPORT=0^
-DBUILD_OSSIMPLANET=OFF^
-DBUILD_OSSIMPLANETQT=OFF^
-DBUILD_OSSIMPREDATOR=OFF^
-DBUILD_OSSIM_TEST_APPS=1^
-DBUILD_RUNTIME_DIR=bin^
-DBUILD_SHARED_LIBS=ON^
-DBUILD_WMS=OFF^
-DCMAKE_BUILD_TYPE=Release^
-DCMAKE_INCLUDE_PATH=%common_dir%/include^
-DCMAKE_INSTALL_PREFIX=%common_dir%/local^
-DCMAKE_LIBRARY_PATH=%common_dir%/lib^
-DCMAKE_MODULE_PATH=%pkg_dir%/cmake/CMakeModules^
```

ossim-package-cmake-config-vs10-nmake-v1.bat Continued

```
-DKAKADU_ROOT_SRC=%common_dir%/src/kakadu_v6_4^
-DKAKADU_LIBRARY=%common_lib%/kdu_v64.lib^
-DKAKADU_AUX_LIBRARY=%common_lib%/kdu_a64.lib^
-
DGEOS_INCLUDE_DIR=%common_dir%/include/geos/capi;%common
_dir%/include/geos/source/headers^
-DGEOS_LIBRARY=%common_lib%/geos.lib^
-
DOPENTHREADS_LIBRARY=%common_lib%/OpenThreadsWin32.lib^
-DPNG_LIBRARY=%common_lib%/libpng.lib^
-DTIFF_LIBRARY=%common_lib%/libtiff_i.lib^
-DGEOTIFF_LIBRARY=%common_lib%/geotiff_i.lib^
-DFFTW3_LIBRARY=%common_lib%/libfftw3-3.lib^
-DOSSIM_COMPILE_WITH_FULL_WARNING=ON^
-DOSSIM_DEPENDENCIES=%common_dir%^
-DOSSIM_DEV_HOME=%build_dir%^
-DBUILD_SHARED_LIBS=ON^
-DGDAL_LIBRARY=%common_lib%/gdal_i.lib^
```

ossim-package-cmake-config-vs10-nmake-v1.bat Continued

```
-DGDAL_INCLUDE_DIR=%common_dir%/include/gdal^
 -DGEOTIFF_INCLUDE_DIR=%common_dir%/include/geotiff^
 -DJPEG_INCLUDE_DIR=%common_dir%/include/jpeg8a^
 -DJPEG_LIBRARY=%common_lib%/libjpeg.lib^
 -DOPENTHREADS_INCLUDE_DIR=%common_dir%/include^
 -DTIFF_INCLUDE_DIR=%common_dir%/include/tiff^
 -DZLIB_INCLUDE_DIR=%common_dir%/zlib^
 -DZLIB_LIBRARY=%common_lib%/zlib.lib^
 -DMINIZIP_INCLUDE_DIR=%common_dir%/include/minizip^
 -DMINIZIP_LIBRARY=%common_lib%/minizip.lib^
 -DOSSIM_PLUGINS_INCLUDE_DIR=%pkg_dir%/ossim_plugins^
 %pkg_dir%/ossim_package_support/cmake/
```