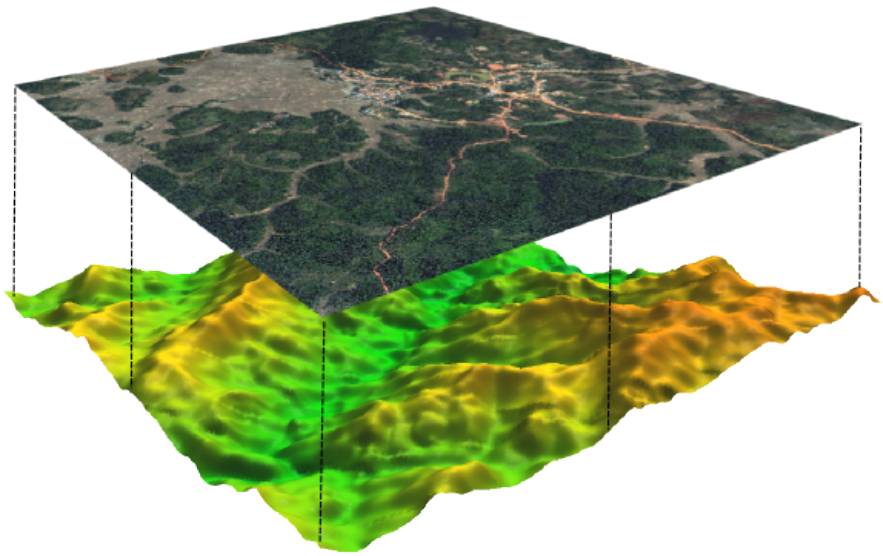
Method Reference:

Ortho-rectification of *QuickBird* imagery with OSSIM (*Open Source Software Image Map*)

Author: Stefan Sylla



October 16, 2011

Contents

1 Introduction

QuickBird imagery is increasingly used in different areas of land-related planning activities. However, by delivery the images are normally not ortho-rectified which means that, due to distortions, they cannot be used for mapping or spatial analysis, particularly if the image area covers mountaneous terrain.

Ortho-rectification is supposed to correct for the image distortions that are caused mainly by hilly terrain, but also by systemic deviations that occur during image-capturing at the sensor.

This tutorial leads through the ortho-rectification of QuickBird *Standard ortho-ready* satellite imagery (DigitalGlobe 2011) with *Open Source Software Image Map* ("OSSIM", s. `http://www.ossim.org/OSSIM/OSSIM.html`). *OSSIM* is an open source project for remote sensing and image processing that has been under active development since 1996. Once running, it provides for highly effective, stable and reliable image processing capability, including ortho-rectification for various sensor types.

The procedure described here was performed on Ubuntu 10.10 64bit with a compiled version of *OSSIM*.

2 Preconditions

To perform the ortho-rectification described here, you need a compiled installation of *OSSIM* (for the latest version s. `https://svn.osgeo.org/ossim/trunk`) and a digital elevation model (DEM) of the area that is covered by your imagery. The handling of QuickBird imagery requires a lot of processing resources due to its high resolution and the large size of the image-tiles. Therefore it is recommended to have >4GB of memory.

Getting *OSSIM* running is a task on its own, as there is no recent binaries available yet (those available are usually a slimmed version of the actual code). There is a general, though precise build instruction available (s. OSSIM 2010), which, with some effort of yours, will lead you through the installation process. You do not need to be a C++ programmer in order to get *OSSIM* running and using it. As long as you're open to using (or learning to use) a command-line and bring along some patience, there's nothing to worry about.

After a successful compilation, you need to setup the elevation and the preferences file so that *OSSIM* can use your DEM and geoid (standard geoid grids like egm96 should come with the installation) for the ortho-rectification process. Please refer to the tutorials on the *OSSIM*-wiki (`http://trac.osgeo.org/ossim/wiki/UserDocs`) for both the preferences-file- and the elevation-setup.

2.1 Elevation test / verify your installation

To make sure that your installation of *OSSIM*, including elevation setup, is correct, make a test using `ossim-info` with the `--height` parameter. Check any coordinate pair (lat long) that lies within the area of your DEM. If your

setup is running, you should get an output like this (in this case, a general_raster (*.ras) DEM was used):

```
$ ossim-info --height 17.58426 101.09839
Opened cell:  /.../ossimdata/elevation/general_raster/dem.ras
MSL to ellipsoid delta:  -33.7083039775783
Height above MSL: 530.158366938998
Height above ellipsoid:  496.45006296142
Geoid value:  -33.7083039775783
```

If you get something like this,

```
$ ossim-info --height 17.58426 101.09839
Did not find cell for point!
MSL to ellipsoid delta:  nan
Height above MSL: nan
Height above ellipsoid:  -30.913768826026
Geoid value:  -30.913768826026
```

... you have to check your elevation setup and/or your DEM-cells again. Otherwise, *OSSIM* will not be able to use your elevation data for the ortho-rectification process, which is crucial.

3 Ortho-rectification

3.1 Preparation

By delivery, Quickbird *Standard imagery* (s. DigitalGlobe 2011) is delivered in tiles. Normally you get them on several DVDs or on a mobile HD. For the ortho-rectification with *OSSIM*, every tile of one image should be in one folder (preferably on your local machine), which, in the case of QB *Standard imagery* includes at least the following:

- *.TIF-files (image-tiles)
- *.TIL-file (tiling-information)
- *.RPB-file (rational polynomial coefficients)

Here is an example on how your folder could look like:

```
$.../quickbird_imagery/ls
10MAY09035320-S2AS-052233437180_01_P002-BROWSE.JPG
10MAY09035320-S2AS-052233437180_01_P002.IMD
10MAY09035320-S2AS-052233437180_01_P002_README.TXT
10MAY09035320-S2AS-052233437180_01_P002.RPB
10MAY09035320-S2AS-052233437180_01_P002.TIL
10MAY09035320-S2AS-052233437180_01_P002.XML
```

```
10MAY09035320-S2AS_R1C1-052233437180_01_P002.TIF
10MAY09035320-S2AS_R2C1-052233437180_01_P002.TIF
10MAY09035320-S2AS_R3C1-052233437180_01_P002.TIF
10MAY09035320-S2AS_R4C1-052233437180_01_P002.TIF
10MAY09035320-S2AS_R5C1-052233437180_01_P002.TIF
BASE.TXT
```

Before you start the ortho-rectification, make sure that all the tiles that are listet in the *.TIL-file are in your image folder and that their naming corresponds to the one in the *.TIL-file. Furthermore, you should check if *OSSIM* understands your QB-imagery and the auxiliary files properly. You can check again by `ossim-info`:

```
$ ossim-info 10MAY09035320-S2AS-052233437180_01_P002.TIL
...
image0.band0.max_value:  2047
image0.band0.min_value:  1
image0.band0.null_value:  0
image0.band1.max_value:  2047
...
image0.overview_handler:  ossimTiffTileSource
...
image0.projection.line_den_coeff_00:  1
image0.projection.line_den_coeff_01:  -2.346772e-06
image0.projection.line_den_coeff_02:  0.0004076766
...
image0.projection.type:  ossimQuickbirdRpcModel
...
```

This is just an excerpt of the long list you should get after using `ossim-info` on your *.TIL-file. But it is important that you find the lines about the image-handler ('`ossimTiffTileSource`'), where *OSSIM* shows that it understands the *.TIL-file. Most important, the line '`image0.projection.type: ossimQuickbirdRpcModel`' has to come up, which indicates that the *.RPB-file (rational polynomial coefficients) is being recognized and will be used for the ortho-rectification.

That's it, finally! Now everything should be ready and *OSSIM* has all the information needed to perform the task.

3.2 `ossim-orthoigen`

For the ortho-rectification, we use `ossim-orthoigen` to produce the ortho-image in one run, using the example from above (assuming, we are within the imagery-folder). Before we start the time-consuming ortho-rectification itself, we check if our image-data is fine by creating a thumbnail-image with the `--t` option (the value depending on the size that you want to have for your thumb; the thumbnail

creation can be strongly accelerated by building pyramids of the images, using `ossim-img2rr` on the *.TIL-file):

```
ossim-orthoigen -t 8000 10MAY09035320-S2AS-052233437180_01_P0
02.TIL ortho-thumb-test.tif
```

Open the thumbnail that we just created in a GIS software and have a look on the visual appearance and projection etc. Alternatively, you could also use `gdalinfo` for an independent check of the newly created ortho-thumbnail-image. If you're happy with the results, start the "real" ortho-rectfication, which can take some minutes to hours, depending on the size of your image:

```
ossim-orthoigen 10MAY09035320-S2AS-052233437180_01_P002.TIL
10MAY09035320-S2AS-052233437180_01_P002.TIF
```

As you see, `ossim-orthoigen` provides for ortho-rectification in one run, you do not need to mosaic the tiles before, the information in the *.TIL-file is sufficient and is understood by *OSSIM*.

There are many options for `ossim-orthoigen` where you can adjust various parameters like resampling-type, output-projection, output filetype, color-balancing or rescaling, to mention just a few. Try it out and discover the capabilities of `ossim-orthoigen`. However, for a start the default setup which we used above should give you a ready-to-use ortho-rectified image.

Of course, if you do not want to use the *.TIL file and instead tell *OSSIM* manually what tiles have to be included into the ortho-rectification, you can also go for that:

```
ossim-orthoigen 10MAY09035320-S2AS_R1C1-052233437180_01_P002.T
IF 10MAY09035320-S2AS_R2C1-052233437180_01_P002.TIF ...  (list
all the TIF-tiles here) ...  10MAY09035320-S2AS-052233437180_
01_P002.TIF
```

Or, if you're having a huge list of tiles for your image, working on shell you could simplify this e.g. through:

```
ls *.TIF | tr '\n' ' ';
while read FILE do
ossim-orthoigen -t 8000 $FILE 10MAY09035320-S2AS-052233437180_
01_P002.TIF
done
```

Then, shell will generate a list of all the TIF-tiles in your folder and put them into the `ossim-orthoigen` codeline.

4 Remarks

The ortho-rectification described here did not include the use of GCP's that are normally necessary to get a higher accurate metric result for the ortho-image.

OSSIM is able to implement GCP's, but so far I did not get through the process (see my latest entry at `http://osgeo-org.1803224.n2.nabble.com/Improving-RPCs-with-GCPs-td6010420.html`). As a workaround, I loaded the resulting ortho-image into the geo-referencer of Quantum GIS (`www.qgis.org`) and implemented my GCPs there, after the actual ortho-rectification with `ossim-orthoigen`. From a photogrammetrical point of view this is not the way to do it, however, my results where as accurate as the ones I got from conventional processing in other software, where the GCPs are implemented within the ortho-rectification process itself.

References

DigitalGlobe (2011): DigitalGlobe Core Imagery Products Guide, `http://www.digitalglobe.com/digitalglobe2/file.php/811/DigitalGlobe\_Core\_Imagery\_Products\_Guide.pdf` (last accessed: Aug. 29, 2011).

OSSIM (2010): Ossim Build Instructions version 1.4, `http://download.osgeo.org/ossim/docs/pdfs/OssimBuildInstructions\_v1.4.pdf` (last accessed: Sept. 23, 2011).