Sierra Nevada Corporation

GeoImage Library

User Manual

Author:
Marvin Smith

May 15, 2012

Introduction

The GeoImage Library is designed to allow for the loading and manipulation of NITF (National Imagery Transmission Format) images in OpenCV and C++ in general.

Usage

First, it is necessary to include the library. This is done with a single include statement. This proceedure is shown in Figure **??**.

```
#include <GeoImage>

int main( int argc, char* argv[] ){

    GEO::GeoImage img;
    return 0;
}
```

Figure 1: Process of importing the library and creating a basic image.

When you compile your program, it is necessary to link the library. This is shown in figure **??**. Note that the build system will default to `/opt/local` for the path. You may change this in the Makefile.

```
g++ hello_world.cpp -I/opt/local/include -L/opt/local/lib -lgeoimage
```

Figure 2: Compilation example.

API Tutorial

```
GEO::GeoImage( const string& image_filename = "_NO_IMAGE_SELECTED_",
               const bool& initialize = false );
```

Figure 3: Constructor for the GeoImage Object

This is the basic constructor for the GeoImage class. This constructor requires two arguments, the name of the file which you are loading and a flag to tell the image if it is to load the data. Note that certain images may be large, thus it is advantageous to load them only when they are required. True will load it immediately, and false will wait until the initialization flag is set at a later time. To set the flag and load the image, you may call the `set_init()` flag. Also, every structure in the GeoImage library resides in the GEO namespace.

You may also change the image filename PRIOR to loading the image.

Writing Images

In order to write images, you must possess certain information. First you must give the api the desired filename, the OpenCV Images, and finally the header information required to write the image.

```
GeoImage img( "U_1001A.NTF", false); //initialize image
img.set_init(true); //image is now loaded
```

Figure 4: Example of loading image into memory.

```
GEO::GeoImage imgin( "U_1001A.NTF", true);
Mat cvimg = imgin.get_image();

/* Some Operations are applied here */

string output_filename = "result.ntf";
GEO::GeoHeader_Info* header_data = imgin.get_header();

GEO::GDAL_Data::write( output_filename, cvimg, header_data);
```

Figure 5: General Process of Writing NITF Images