OTB Users Days 2016

Integrating S1 geometry in Orfeo ToolBox

OTB Team

07/06/2016

Introduction

- S1 geometry in OTB requested by many users
- Ortho-rectification of S1 images needed
- Lots of discussion during last OTB Users Days : politics vs. technical
- Will hopefully be available in OTB 5.6

Current sensor modelling in OTB

What it is

- A "generic" sensor model and several specializations (alos, radarsat, envisat, csk, tsx . . .)
- Available in ossimPlugins
- One-shot funding by CNES more than 10 years ago

Why not using it ?

Because noone can maintain this code :

- Responsibility is shared between 28 poorly documented classes
- It makes coverity static code analysis burst in tears
- It makes my eyes bleed
- It is responsible for lots of failing tests on dashboard (are sensor model supposed to return NaN ?)
- It suffers from chaotic behaviour

How bad is it ?

A frequent pattern found by coverity :

```
MyFunction()
{
  MyType * t = new MyType();
  [hundreds of lines of code]
  if(condition)
   {
   // Error, get out of the function
    return 1;
   }
  [hundreds of lines of code]
  delete t;
  return 0;
}
```

(During first runs, most of critrical issues reported by coverity where localted in ossimPlugins)

How bad is it ?

Do you know what is this piece of code doing ?

```
for (int i = 0 ; i < nRoot ; i++)
   {
     /* Computation of the "distance" between roots images and equation values */
     u = x[i] - posX ;
     v = y[i] - posY ;
     w = z[i] - posZ ;
     r[i] = fabs ((u * u + v * v + w * w) / (dist1 * dist1) - 1.0 ) ;
     u = u * speedX ;
     v = v * speedY ;
     w = w * speedZ ;
     if (fabs (fDop) > EPSILON)
       r[i] = r[i] + fabs (1.0 + 2.0 * (u + v + w) / (lambda * dist1 * fDop1));
     else
       r[i] = r[i] + fabs (2.0 * (u + v + w) / (lambda * dist1)) ;
     u = x[i] / he ;
     v = y[i] / he ;
     w = z[i] / hp ;
     r[i] = r[i] + fabs (u * u + v * v + w * w - 1.0) ;
     indice[i] = i ;
   }
```

- It is obviously solving SAR direct location somehow
- But where are the documentation, the references, the equations that are solved ?

Conclusion on current implementation

- Some of those derived sensor models work
- Other don't
- We can not extend this code
- We can barely maintain it (appart from coverity alert fixes)

What about OSSIM models ?

ossimRS1SarModel

- Directly written for RadarSat1
- Does not support all modes
- Only implements forward sensor model
- Again, code is difficult to understand

ossimSARModel

- Generic SAR implementation ...
- ... with no practical implementation available
- Based on a SAR sensor model from some book (seems more adapted to airborne SAR)
- Almost no class documentation
- Not easy to understand what the parameters are from their name (ORP ? Oipr ? ARP ?)

So . . . I wrote our own model

- A single, well documented class for generic SAR sensor modelling
- Using equations from ESA document "Guide to ASAR geocoding" (ref RSL-ASAR-GC-AD)
- Using by-contract programming
- Implementing both forward and inverse location
- One subclass by sensor (currently S1 and TSX)
- The S1 model supports all S1 modes and product types
- S1 inverse model precision is bellow one 10th of pixel (wrt GCPs)
- TSX model also available and working
- And to be sure I asked Luc to review the new code

Autotest output

```
GCP #425
Azimuth time: ref=2016-Jan-17 20:42:13.875636, predicted: 2016-Jan-17 20:42:13.875615, res=-00:00:00.000021
Slant range time: ref=0.005180835301345, predicted: 0.005180835301355, res=0.000000000000009
Image point: ref=( 3352.000000000000000, 15987.000000000000000 ), predicted=( 3352.262004643973341, 15986.920482901527066 ), res=( 0.2

GCP #426
Azimuth time: ref=2016-Jan-17 20:42:13.875667, predicted: 2016-Jan-17 20:42:13.875647, res=-00:00:00.000020
Slant range time: ref=0.005243347466786, predicted: 0.005243347466793, res=0.000000000000006
Image point: ref=( 4190.000000000000000, 15987.000000000000000 ), predicted=( 4189.967328529995029, 15986.929009914703784 ), res=( -0.

GCP #427
Azimuth time: ref=2016-Jan-17 20:42:13.875700, predicted: 2016-Jan-17 20:42:13.875680, res=-00:00:00.000020
Slant range time: ref=0.005309205259236, predicted: 0.005309205259240, res=0.000000000000004
Image point: ref=( 5028.000000000000000, 15987.000000000000000 ), predicted=( 5027.767536782606840, 15986.937803397040625 ), res=( -0.

GCP #428
Azimuth time: ref=2016-Jan-17 20:42:13.875734, predicted: 2016-Jan-17 20:42:13.875716, res=-00:00:00.000018
Slant range time: ref=0.005378285083704, predicted: 0.005378285083704, res=0.000000000000001
Image point: ref=( 5866.000000000000000, 15987.000000000000000 ), predicted=( 5865.804389267414081, 15986.947396286863295 ), res=( -0.

GCP #429
Azimuth time: ref=2016-Jan-17 20:42:13.875770, predicted: 2016-Jan-17 20:42:13.875752, res=-00:00:00.000018
Slant range time: ref=0.005450463706746, predicted: 0.005450463706743, res=0.000000000000002
Image point: ref=( 6704.000000000000000, 15987.000000000000000 ), predicted=( 6704.005285616486617, 15986.956989176687784 ), res=( 0.0

GCP #430
Azimuth time: ref=2016-Jan-17 20:42:13.875808, predicted: 2016-Jan-17 20:42:13.875791, res=-00:00:00.000017
Slant range time: ref=0.005525618936514, predicted: 0.005525618936508, res=0.000000000000005
Image point: ref=( 7542.000000000000000, 15987.000000000000000 ), predicted=( 7542.188331271720926, 15986.967381473996284 ), res=( 0.1

GCP #431
Azimuth time: ref=2016-Jan-17 20:42:13.875847, predicted: 2016-Jan-17 20:42:13.875830, res=-00:00:00.000017
Slant range time: ref=0.005603630217246, predicted: 0.005603630217238, res=0.000000000000008
Image point: ref=( 8380.000000000000000, 15987.000000000000000 ), predicted=( 8380.210509368893327, 15986.977773771304783 ), res=( 0.2
```

What we plan to do

For OTB 5.6 :

- Merge with current S1 model (which only reads parameters for radiometric calibration)
- Also use the new SAR sensor model instead of ossimPlugins code for TSX model

For later :

- Change all SAR sensor models to use the new code
- Get rid of the old crappy code

In the mean time :

- Situation is not perfect since we will have a mix of implementations
- Better than nothing