

ISIS - Recommendation #866

Look at blend program to see why it is so slow on large image lists and see how much time it would take to add improvements

2012-05-30 09:03 AM - Janet Barrett

Status: Closed	
Priority: Normal	
Assignee: Janet Barrett	
Category: Applications	
Target version: N/A	
Impact:	Software Version:
Description The blend program can take upwards of 36 hours to work on a list of over a 1000 Themis images. Need to determine exactly how blend is doing its work and see if adding other capabilities to it (such as using footprints/findimageoverlaps to determine overlaps more quickly or to have an option where only adjacent images in the input list are blended) will speed up the program. <i>Steps to reproduce:</i> I need to obtain some data from Robin or Lynn.	
Related issues: Related to ISIS - Bug #991: Optimize 'Blend'	Assigned

History

#1 - 2012-05-30 10:06 AM - Janet Barrett

I obtained some data from Robin to do testing which is located at /work/users/rfergason/carto/blend/test_Dec2011/100_images. I spoke to Stuart and he agrees that it would be a good idea to run some timing tests.

#2 - 2012-05-30 10:26 AM - Tammy Becker

The initial part of this post (diagnosing the issue) will be covered by Isis Support. Depending on the outcome of the evaluation and decided resolution, the effort could be partially covered by another project (lgaddis/rferguson).

#3 - 2012-06-13 10:41 AM - Janet Barrett

I have run timing tests on the blend program and have come up with the following statistics after running blend on a list of 47 Themis files:

OverlapStatistics: 102.580002831 seconds = 4.50857505648%
File 1 Chip: 807.690002441 seconds = 35.4994238436%
File 2 Chip: 783.960021973 seconds = 34.4564486527%
Linear regression: 342.700012207 seconds = 15.062280018%
Perform blend: 238.289993286 seconds = 10.4732724731%
Total time: 2275.22003174 seconds

Very little time is going into the call to OverlapStatistics which first determines if 2 files overlap. As a result, it may not be prudent at this time to try and add footprint capability to the program.

The majority of the time is going into the creation of the Chips for each file's overlap area. I talked to Stuart about this and he is concerned that there may be some extra work being done by the Chip class that is not required. Basically, the Chip class is trying to geom one image to another to come up with the overlap areas that are then used to do statistics on. I am going to do a little more investigation to see if this is the case. If we can get rid of some of these unnecessary calls during the Chip creation process, then we will see a big time savings.

#4 - 2012-06-14 10:58 AM - Janet Barrett

I have now narrowed down the time synch for the blend program. As you recall, the setting up of the chip for each individual image file takes roughly 35% of the total processing time. For each blend that happens between 2 images this adds up to 70% of the time (35% per image). We thought that the interpolation and geom were taking up most of the time, but it turns out that there is no geom being done on these images.

Almost 100% of the time that the Chip is taking goes into the Read() method of the Chip class. I did some timing tests on the Read() method and came up with the following breakdown:

```
SetChipPosition((double)samp, (double)line); 12.8961284232%
```

```
if((CubeSample() < 0.5) || (CubeLine() < 0.5) ||  
(CubeSample() > cube.getSampleCount() + 0.5) ||  
(CubeLine() > cube.getLineCount() + 0.5)) {
```

```
p_buf[line-1][samp-1] = Isis::NULL8; .0204245435778%
```

```
} else if(p_clipPolygon == NULL)  
port.SetPosition(CubeSample(), CubeLine(), band); 12.7764478941%
```

```
cube.read(port); 60.834324297%
```

```
p_buf[line-1][samp-1] = interp.Interpolate(CubeSample(), CubeLine(), port.DoubleBuffer()); 13.4726748421%  
}
```

This set of code is gone through for every pixel in the Chip. As you can see, the cube.read is taking up 60% of the time. A port is read for every pixel in the Chip. I changed the interpolation type from CubicConvolution to Bilinear, but that only cut the time down slightly. The read is very expensive, so if we could cut down on the number of reads, then the program would run a lot faster. Maybe the entire Chip or larger chunks of the Chip could be read at once and the interpolation done on the data in a buffer in memory.

I will discuss these findings with Stuart and figure out where we should go from here.

#5 - 2012-07-06 10:59 AM - Janet Barrett

I have talked to Stuart about the findings on the time usage of the blend program. We think it would probably take at least 2 weeks to modify the code that reads the Chip data so that it takes less time. Stuart does not have the money to pay for this modification, so another project would need to pay for this modification to be made so that the blend program is faster. I will leave it up to the interested parties to decide if this is something that we want to pursue.

#6 - 2012-08-28 02:21 PM - Janet Barrett

This issue was resolved and the actual work on blend is being done through Mantis post [#991](#).

#7 - 2013-08-15 01:07 PM - Anonymous

- *Target version changed from 150 to N/A*