Satellite Photogrammetry Homework 4
assigned 6 April 2016, due Friday 15 April

1. Create a function to generate partial derivatives of our 2 condition equations:

$$\begin{bmatrix} \partial F_1/\partial l & \partial F_1/\partial s & \partial F_1/\partial \phi & \partial F_1/\partial \lambda & \partial F_1/\partial h & \partial F_1/\partial q_i & \partial F_1/\partial q_j & \partial F_1/\partial q_k \\ \partial F_2/\partial l & \partial F_2/\partial s & \partial F_2/\partial \phi & \partial F_2/\partial \lambda & \partial F_2/\partial h & \partial F_2/\partial q_i & \partial F_2/\partial q_j & \partial F_2/\partial q_k \end{bmatrix} = \text{fi2g\_pl\_0\_part}(im\#, l, s, h, \phi, \lambda, dp)$$

where $dp = \left[dq_i \; dq_j \; dq_k\right]^T$

Inside fi2g_pl_0_part, compute the partial derivatives of the function fi2g_pl_0 by numerical approximation (just like in fg2i!)

2. Measure 9 GCP's and at least 7 tie/pass points on both images colo1.tif and colo2.tif. You may use imeas2.m located at ftp://ftp.ecn.purdue.edu/bethel. Read the comments at the top for instructions.

3. Modify or make a new version of fi2g which correctly uses the im# argument, and also uses the dp argument as follows: interpolate q_i, q_j, q_k, q_s as usual. Normalize the vector. Add $q_i = q_i + dq_i$, $q_j = q_j + dq_j$, $q_k = q_k + dq_k$, then normalize again. Make sure all functions pass the im# and dp arguments.

4. Create a main script to do the 2 image BBA. For all global support data variables add an additional array index to choose between image 1 & 2. Read in from 2 sets of support data files. See sample code! Create initial approximations for $\phi \lambda h$ of the tie points using fi2g_pl function (guess the h).

5. <u>Baseline solution</u>: use 8 GCP's and 7 tie points as in the diagram. Develop 2 linearized condition equations for each point on each image.

$$\begin{bmatrix} \partial F_1/\partial \ell & \partial F_1/\partial s \\ \partial F_2/\partial \ell & \partial F_2/\partial s \end{bmatrix} \begin{bmatrix} v_\ell \\ v_s \end{bmatrix} + \begin{bmatrix} \partial F_1/\partial \phi & \partial F_1/\partial \lambda & \partial F_1/\partial h & \partial F_1/\partial q_i & \partial F_1/\partial q_j & \partial F_1/\partial q_k \\ \partial F_2/\partial \phi & \partial F_2/\partial \lambda & \partial F_2/\partial h & \partial F_2/\partial q_i & \partial F_2/\partial q_j & \partial F_2/\partial q_k \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\lambda \\ \Delta h \\ \Delta q_i \\ \Delta q_j \\ \Delta q_k \end{bmatrix} = \begin{bmatrix} -F_1^0 \\ -F_2^0 \end{bmatrix} - A(\ell - \ell^0)$$

$$A \quad v \quad + \quad B \quad \Delta \quad = \quad f$$

Solve, update, and iterate, as shown on the condition equation layout. Do either fixed number of iterations or variable number, but confirm convergence of parameter correction vector.

<u>Enhanced Solution</u> (optional) use same or larger number of tie points. Enforce uncertainty of attitude data using unified LS. Details coming. Use the 9^{th} GCP as a "check point" and compare computed vs. actual value. OR, run 9 times with a different GCP witheld each time to get 9 checkpoints. Summarize discrepancies with RMS_e, RMS_n, RMS_u. This is known as LOO = Leave one out strategy.

6. Show residuals, coordinates of tie points, and final corrections to the 3 attitude parameters. Show evidence of convergence.

```
global ephdata attdata
global eph_start_time first_line_time avg_line_rate dtl dte
global x0 y0 z0 det_pitch PD

ephdata=zeros(13,1011,2);
attdata=zeros(15,1011,2);
eph_start_time=zeros(2,1);
first_line_time=zeros(2,1);
avg_line_rate=zeros(2,1);
dtl=zeros(2,1);
dte=zeros(2,1);
x0=zeros(2,1);
y0=zeros(2,1);
z0=zeros(2,1);
PD=zeros(2,1);
det_pitch=zeros(2,1);

fid=fopen('colo1\\colo1.eph','rt');
for i=1:10
  lin=fgets(fid);
  end
temp=fscanf(fid,'( %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f,
%f),\n',[13,1011]);
ephdata(:,:,1)=temp;
fclose(fid);

fid=fopen('colo2\\colo2.eph','rt');
for i=1:10
  lin=fgets(fid);
  end
temp=fscanf(fid,'( %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f,
%f),\n',[13,969]);
ephdata(:,1:969,2)=temp;
fclose(fid);

fid=fopen('colo1\\colo1.att','rt');
for i=1:10
  lin=fgets(fid);
  end
temp=fscanf(fid,'( %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f,
%f),\n',[15,1011]);
attdata(:,:,1)=temp;
fclose(fid);

fid=fopen('colo2\\colo2.att','rt');
for i=1:10
  lin=fgets(fid);
  end
temp=fscanf(fid,'( %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f,
%f),\n',[15,969]);
attdata(:,1:969,2)=temp;
fclose(fid);


% ok now some constants read (manually) from support file headers
% make 2 element arrays to hold values for image 1 & image 2

eph_start_time=[9.545140;55.144961]; % seconds
first_line_time=[16.876479;62.178479]; % seconds
avg_line_rate=[24000.0;24000.0]; % Hz
dtl=[1/avg_line_rate(1);1/avg_line_rate(2)]; % seconds
dte=[0.02;0.02];
```

```
x0= [   0.053720;0.053720];
y0= [ 140.711930;140.711930];
z0= [7949.165;7949.165];
PD=z0;
det_pitch= [0.008000;0.008000];
```