Satellite Photogrammetry HW2 20 Feb 2015
due Monday 2 March

(00) measure the 9 control points described in the document under "notes". measure each point on 2 images colo1 and colo2. Units must be pixels. For each measurement, estimate the uncertainty (½, 1, 2, ... 5... pixels). Before we use these ground coordinates will be provided.

Build a suite of matlab functions & a generic "main" program:

(0) main program: read and store support data for 2 images. This includes ephemeris, attitude, & constants, times, etc. so that you can access them by an additional array index.

suggestion:

```
ephdata = zeros(13, 1011, 2);   % colo1
temp* = fscanf( .... [13,1011]);
ephdata(:,:,1) = temp;
% next file colo2
temp* = fscanf( .... [13,969]);
ephdata(:, 1:969, 2) = temp;
```

(* read 10 header lines)

If you prefer to do this with classes etc., OK. We just need to work with multiple images at the same time

make all of this support data global so it can be accessed by any function which also declares it as global.

(1) function $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \text{FI2G}(im\#, l, s, h, \Delta p)$ This implements the image to ground projection done in HW1. The Δp vector will be:

$$\Delta p = \begin{bmatrix} \Delta q_i \\ \Delta q_j \\ \Delta q_k \\ \Delta q_s \\ \Delta X_s \\ \Delta Y_s \\ \Delta Z_s \end{bmatrix}$$

Note the nominal values for all elements of Δp are zero. If any are nonzero then you have changed the physical model (which we will do later)

for now im# will be either 1 or 2

show that you get same result as HW1

(2) function $\begin{bmatrix}\phi\\ \lambda\end{bmatrix}$ = FI2G_PL(im#, l, s, h, Δp) This is just a wrapper of FI2G (1) + a conversion from cartesian to geographic.

(3) function $\begin{bmatrix}\Delta\phi\\ \Delta\lambda\end{bmatrix}$ = FI2G_PL_0 (im#, l, s, h, ϕ, λ, Δp)
(Zero)

This is just a wrapper of FI2G_PL (2) + a subtraction. The output vector is just a <u>misclosure</u> between input ϕ, λ and the ϕ, λ produced from l, s, h. It is just

$$\begin{bmatrix}\Delta\phi\\ \Delta\lambda\end{bmatrix} = \begin{bmatrix}\phi\\ \lambda\end{bmatrix} - \text{FI2G\_PL}(\text{im\#}, l, s, h, \Delta p)$$

(4) using function (3) FI2G_PL_0 we next numerically solve for l, s as unknowns, given input ϕ, λ, h. In other words, we invert FI2G_PL. make function $\begin{bmatrix}l\\ s\end{bmatrix}$ = FG2I(im#, ϕ, λ, h, Δp) to accomplish this.

(a) start with $(l^\circ, s^\circ) = (0, 0)$, input $\phi, \lambda, h, \Delta p$

Iteration Loop

(b) compute $\begin{bmatrix}F_\phi\\ F_\lambda\end{bmatrix}$ = FI2G_PL_0(im#, l°, s°, h, ϕ, λ, Δp)

(c) compute partial derivatives $\partial F_\phi/\partial l$, $\partial F_\phi/\partial s$, $\partial F_\lambda/\partial l$, $\partial F_\lambda/\partial s$ numerically:

$$\begin{bmatrix}\partial F_\phi/\partial l\\ \partial F_\lambda/\partial l\end{bmatrix} \approx \frac{\text{FI2G\_PL\_0}(\text{im\#}, l^\circ+\Delta l, s^\circ, h, \phi, \lambda, \Delta p) - \text{FI2G\_PL\_0}(\text{im\#}, l^\circ, s^\circ, h, \phi, \lambda, \Delta p)}{\Delta l}$$

$$\begin{bmatrix}\partial F_\phi/\partial s\\ \partial F_\lambda/\partial s\end{bmatrix} \approx \frac{\text{FI2G\_PL\_0}(\text{im\#}, l^\circ, s^\circ+\Delta s, h, \phi, \lambda, \Delta p) - \text{FI2G\_PL\_0}(\text{im\#}, l^\circ, s^\circ, h, \phi, \lambda, \Delta p)}{\Delta s}$$

(d) solve newton iteration step for <u>corrections</u> to l°, s° :

$$\begin{bmatrix} \partial F_\phi/\partial l & \partial F_\phi/\partial s \\ \partial F_\lambda/\partial l & \partial F_\lambda/\partial s \end{bmatrix} \begin{bmatrix} \Delta l \\ \Delta s \end{bmatrix} = \begin{bmatrix} -F_\phi \\ -F_\lambda \end{bmatrix}$$

{ note: $\Delta l, \Delta s$ in step (c) are perturbations to estimate partials. Here $\Delta l, \Delta s$ are corrections to refine current estimate l°, s°. }

$$J \quad \Delta \quad = \quad -F$$

$$\Delta = -J^{-1} \cdot F$$

(e) update l°, s°

$$\begin{bmatrix} l^\circ_{new} \\ s^\circ_{new} \end{bmatrix} = \begin{bmatrix} l^\circ_{old} + \Delta l \\ s^\circ_{old} + \Delta s \end{bmatrix}$$

iterate until $\Delta l, \Delta s$ are very small

After you are done iterating, the refined l, s are consistent with ϕ, λ, h and you have solved the ground to image projection problem.

$$\begin{bmatrix} l \\ s \end{bmatrix} = FG2I(im\#, \phi, \lambda, h, op)$$

Note: equations in (d) are from Taylor Series Linearization

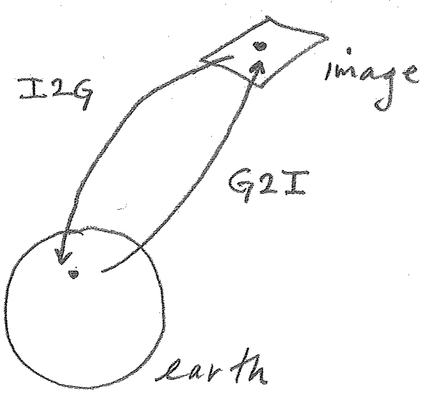
$$F_\phi(l, s) = 0 \quad , \quad F_\phi \approx F_\phi(l^\circ, s^\circ) + \partial F_\phi/\partial l \, \Delta l + \partial F_\phi/\partial s \, \Delta s = 0$$

$$F_\lambda(l, s) = 0 \quad , \quad F_\lambda \approx \underbrace{F_\lambda(l^\circ, s^\circ) + \partial F_\lambda/\partial l \, \Delta l + \partial F_\lambda/\partial s \, \Delta s = 0}_{}$$

put this in matrix form and you get

$$J \Delta = -F$$

(★) verify function (4) by input <u>any</u> ϕ, λ, h in the image, compute l, s. Then use function (2) to verify that you get back ϕ, λ, h. What is the difference? You could do this with a control point, but any point works.



this should be a <u>closed</u> <u>loop</u>.

Apply Δp in the function FI2G as follows:

- interpolate q_i, q_j, q_k, q_s based on line number and corresponding time
- unitize $\vec{q}$
- revise $q_i = q_i + \Delta q_i$
  $q_j = q_j + \Delta q_j$
  $q_k = q_k + \Delta q_k$
  $q_s = q_s + \Delta q_s$

recall $\vec{\Delta p} = \begin{bmatrix} \Delta q_i \\ \Delta q_j \\ \Delta q_k \\ \Delta q_s \\ \Delta X_s \\ \Delta Y_s \\ \Delta Z_s \end{bmatrix}$

- unitize $\vec{q}$
- interpolate X_s, Y_s, Z_s base on line number and corresponding time
- revise $X_s = X_s + \Delta X_s$
  $Y_s = Y_s + \Delta Y_s$
  $Z_s = Z_s + \Delta Z_s$

Functions to compute partial derivatives:

(5) for function $\begin{bmatrix}\phi \\ \lambda\end{bmatrix} = \text{FI2G\_PL}(\text{im\#}, l, s, h, \Delta p)$, make

$$\begin{bmatrix} \frac{\partial\phi}{\partial l} & \frac{\partial\phi}{\partial s} & \frac{\partial\phi}{\partial h} & \frac{\partial\phi}{\partial q_i} & \frac{\partial\phi}{\partial q_j} & \frac{\partial\phi}{\partial q_k} & \frac{\partial\phi}{\partial q_s} & \frac{\partial\phi}{\partial X_s} & \frac{\partial\phi}{\partial Y_s} & \frac{\partial\phi}{\partial Z_s} \\ \frac{\partial\lambda}{\partial l} & \frac{\partial\lambda}{\partial s} & \frac{\partial\lambda}{\partial h} & \frac{\partial\lambda}{\partial q_i} & \frac{\partial\lambda}{\partial q_j} & \frac{\partial\lambda}{\partial q_k} & \frac{\partial\lambda}{\partial q_s} & \frac{\partial\lambda}{\partial X_s} & \frac{\partial\lambda}{\partial Y_s} & \frac{\partial\lambda}{\partial Z_s} \end{bmatrix} \overset{(5)}{=} \text{FI2G\_PL\_PARTIALS}(\text{im\#}, l, s, h, \Delta p)$$

(6) for function $\begin{bmatrix}l \\ s\end{bmatrix} = \text{FG2I}(\text{im\#}, \phi, \lambda, h, \Delta p)$, make

$$\begin{bmatrix} \frac{\partial l}{\partial\phi} & \frac{\partial l}{\partial\lambda} & \frac{\partial l}{\partial h} & \frac{\partial l}{\partial q_i} & \frac{\partial l}{\partial q_j} & \frac{\partial l}{\partial q_k} & \frac{\partial l}{\partial q_s} & \frac{\partial l}{\partial X_s} & \frac{\partial l}{\partial Y_s} & \frac{\partial l}{\partial Z_s} \\ \frac{\partial s}{\partial\phi} & \frac{\partial s}{\partial\lambda} & \frac{\partial s}{\partial h} & \frac{\partial s}{\partial q_i} & \frac{\partial s}{\partial q_j} & \frac{\partial s}{\partial q_k} & \frac{\partial s}{\partial q_s} & \frac{\partial s}{\partial X_s} & \frac{\partial s}{\partial Y_s} & \frac{\partial s}{\partial Z_s} \end{bmatrix} \overset{(6)}{=} \text{FG2I\_PARTIALS}(\text{im\#}, \phi, \lambda, h, \Delta p)$$

---

for partials let $\Delta q_{i,j,k,s} = 1\times10^{-6}$

$\Delta X_s, Y_s, Z_s = 1\times10^{-4}$ (units meters)

$\Delta\phi, \lambda = 1\times10^{-8}$ (units radians)

$\Delta h = 1\times10^{-4}$ (units meters)

---

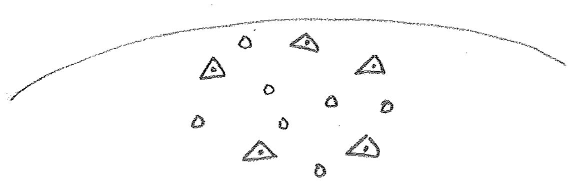
(7) $\begin{bmatrix}\Delta l \\ \Delta s\end{bmatrix} = \text{FG2I\_ls\_0}(\text{im\#}, \phi, \lambda, h, l, s, \Delta p)$ {like collinearity equations for frame photogrammetry}

$$\begin{bmatrix}\Delta l \\ \Delta s\end{bmatrix} = \begin{bmatrix}l \\ s\end{bmatrix} - \text{FG2I}(\text{im\#}, \phi, \lambda, h, \Delta p)$$

(8) partials of function (7)

$$\begin{bmatrix} \frac{\partial F_l}{\partial\phi} & \frac{\partial F_l}{\partial\lambda} & \frac{\partial F_l}{\partial h} & \frac{\partial F_l}{\partial q_i} & \frac{\partial F_l}{\partial q_j} & \frac{\partial F_l}{\partial q_k} & \frac{\partial F_l}{\partial q_s} & \frac{\partial F_l}{\partial X_s} & \frac{\partial F_l}{\partial Y_s} & \frac{\partial F_l}{\partial Z_s} \\ \frac{\partial F_s}{\partial\phi} & \frac{\partial F_s}{\partial\lambda} & \frac{\partial F_s}{\partial h} & \frac{\partial F_s}{\partial q_i} & \frac{\partial F_s}{\partial q_j} & \frac{\partial F_s}{\partial q_k} & \frac{\partial F_s}{\partial q_s} & \frac{\partial F_s}{\partial X_s} & \frac{\partial F_s}{\partial Y_s} & \frac{\partial F_s}{\partial Z_s} \end{bmatrix} \overset{(8)}{=} \text{FG2I\_ls\_0\_PARTIALS}(\text{im\#}, \phi, \lambda, h, l, s, \Delta p)$$

<u>In</u> a bundle block adjustment the <u>observations</u> are l, s on the images, <u>constants</u> are ϕ, λ, h for control points, plus any satellite parameter <u>NOT</u> adjusted, unknowns are ϕ, λ, h of pass points (tie points) and Δp's for any adjustable parameters on any images



○ = pass point

△ = control point

- hand in <u>measurements</u> (note we do not actually use them here !)
- hand in source code for 8 functions
- show result of function evaluated at control point of HW1
- show closed loop for FI2G-PL & FG2I