Space Photogrammetry HW4 GroundToImage function

Spring 2010 due Friday April 2

1. Make sure that your algorithm for HW3 is working correctly and is consistent with misclosure: $dx = 1.95\,m$, $dy = 0.38\,m$

2. Embed that algorithm into a matlab function:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ECF} = i2g\_xyz(\ell, s, h, adj)$$

with 2 additional features,

(a) an extra argument "adj" will contain the adjustable parameter vector,

$$adj = [\omega_0; \omega_1; \omega_2; \phi_0; \phi_1; \phi_2; \kappa_0; \kappa_1; \kappa_2]$$

which are used as,

$$d\omega = \omega_0 + \omega_1 \ell + \omega_2 \ell^2$$

$$d\phi = \phi_0 + \phi_1 \ell + \phi_2 \ell^2$$

$$d\kappa = \kappa_0 + \kappa_1 \ell + \kappa_2 \ell^2$$

these are then used to construct rotation matrix

$$M_A = M_3(d\kappa)\, M_2(d\phi)\, M_1(d\omega)$$

which you use to multiply the view vector <u>prior</u> to the rotation into ECF. If adj = zero vector, then this addition has no influence on the projection, $M_A = I_3$.

(b) include global variables "eph" and "att" which you will fill from outside the function by reading the WV1 support files (see demo code).

3. Call prior function i2g-xyz in,

$$\begin{bmatrix} \phi \\ \lambda \end{bmatrix} = i2g\text{-}pl(l, s, h, adj)$$

don't confuse ϕ latitude with $d\phi$ rotation!

4. Call prior function i2g-pl in

$$\begin{bmatrix} F_\phi \\ F_\lambda \end{bmatrix} = fpl(\phi, \lambda, h, l, s, adj)$$

this function implements the following

$$\underbrace{\begin{bmatrix} F_\phi \\ F_\lambda \end{bmatrix}}_{\text{misclosure}} = \underbrace{\begin{bmatrix} \phi \\ \lambda \end{bmatrix}}_{\text{given } \phi, \lambda} - \underbrace{i2g\text{-}pl(l, s, h, adj)}_{\text{computed } \phi, \lambda \text{ from } l, s, h}$$

this represents 2 equations in 2 unknowns (l, s) - nonlinear - which we will solve iteratively. Recall Newton Raphson method,

$$F \approx F^0 + \frac{\partial F}{\partial x} \cdot \Delta x = 0 \quad \left(\begin{array}{l} F \text{ written to be zero when solved,} \\ \text{i.e. we are finding a root} \end{array} \right)$$

$$\Delta x = \left[\frac{\partial F}{\partial x}\right]^{-1} (-F^0) \quad , \quad \Delta x = J^{-1}(-F^0)$$

5. Make a function $\begin{bmatrix} l \\ s \end{bmatrix} = g2i(x, y, z, adj)$

which implements the above iterative solution for (l, s).

(a) transform $XYZ \rightarrow \phi\lambda h$

(b) initialize unknowns l^0, s^0 to middle of image.

(c) perform an iterative refinement of l^0, s^0

- $F^0 = \begin{bmatrix} F_\varphi \\ F_\lambda \end{bmatrix} = fpl(\varphi, \lambda, h, l^0, s^0, adj)$ (misclosure vector)

- jacobian :

$$\begin{bmatrix} \frac{\partial F_\varphi}{\partial l} \\ \frac{\partial F_\lambda}{\partial l} \end{bmatrix} = \frac{fpl(\varphi, \lambda, h, l^0+\Delta l, s^0, adj) - fpl(\varphi, \lambda, h, l^0, s^0, adj)}{\Delta l}$$

$$\begin{bmatrix} \frac{\partial F_\varphi}{\partial s} \\ \frac{\partial F_\lambda}{\partial s} \end{bmatrix} = \frac{fpl(\varphi, \lambda, h, l^0, s^0+\Delta s, adj) - fpl(\varphi, \lambda, h, l^0, s^0, adj)}{\Delta s}$$

$$J = \begin{bmatrix} \frac{\partial F_\varphi}{\partial l} & \frac{\partial F_\varphi}{\partial s} \\ \frac{\partial F_\lambda}{\partial l} & \frac{\partial F_\lambda}{\partial s} \end{bmatrix}$$

- $\begin{bmatrix} \Delta l \\ \Delta s \end{bmatrix} = J^{-1} \begin{bmatrix} -F_\varphi \\ -F_\lambda \end{bmatrix}$

- $l^0 = l^0 + \Delta l$
  $s^0 = s^0 + \Delta s$

- check for small $\Delta l, \Delta s$
  repeat step (c) until they are small

(d) return refined l, s as results of function g2i

6. confirm that your i2g-xyz and g2i functions are consistent:

$$
\begin{tikzcd}
(\ell, s, h) \arrow[r, "\text{i2g-xyz}"] & xyz \arrow[r, "\text{g2i}"] & (\ell, s) \arrow[ll, bend left]
\end{tikzcd}
$$

these should match, consistent with your <u>convergence</u> <u>criterion</u>

7. demonstrate this "round trip" consistency for

$$(\ell, s)_1 = (5000, 5000)$$

$$(\ell, s)_2 = (15{,}000, 15{,}000)$$

$$(\ell, s)_3 = (27{,}000, 27{,}000)$$

---

next <u>adjustment</u> using the adjustable parameters.

$$V_{ECF} = M_{ES} M_{SC} V_{camera}$$

insert adjustable parameters

$$V_{ECF} = \underbrace{M_{ES} M_A}_{\text{modified rotation to ECF}} M_{SC} V_{camera}$$