

Space Photogrammetry HW4 Ground To Image function

Spring 2010

due Friday April 2

1/4

1. Make sure that your algorithm for HW3 is working correctly and is consistent with misclosure:  $dx = 1.95m$ ,  $dy = 0.38m$
2. Embed that algorithm into a matlab function:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ECF} = \text{img-xyz}(l, s, h, \text{adj})$$

with 2 additional features,

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

$$\text{adj} = [w_0; w_1; w_2; \phi_0; \phi_1; \phi_2; k_0; k_1; k_2]$$

which are used as,

$$dw = w_0 + w_1 l + w_2 l^2$$

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

$$dk = k_0 + k_1 l + k_2 l^2$$

These are then used to construct rotation matrix

$$M_A = M_3(dk) M_2(d\phi) M_1(dw)$$

which you use to multiply the view vector prior to the rotation into ECF. If  $\text{adj} = \text{zero vector}$ , 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 ,

2/4

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

don't confuse  $\phi$  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

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

misclosure given  $\phi, \lambda$       computed  $\phi, \lambda$  from  $l, s, h$

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

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

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

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  $l^0, s^0$

3/4

•  $F^0 = \begin{bmatrix} F_\phi \\ F_\lambda \end{bmatrix} = fpl(\phi, \lambda, h, l^0, s^0, adj)$  (mis closure vector)

• jacobian =

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

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

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

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

•  $l^0 = l^0 + \Delta l$

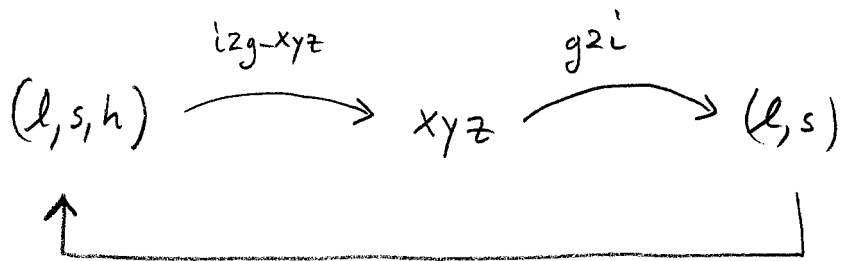
$s^0 = s^0 + \Delta s$

• check for small  $\Delta l, \Delta s$

repeat step (c) until they are small

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

6. confirm that your  $i2g-xyz$  and  $g2i$  functions are <sup>4/4</sup> consistent:



these should match, consistent with your convergence criteria

7. demonstrate this "round trip" consistency for

$$(l, s)_1 = (5000, 5000)$$

$$(l, s)_2 = (15,000, 15,000)$$

$$(l, s)_3 = (27,000, 27,000)$$

next adjustment using the adjustable parameters.

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

$\uparrow$   
 insert adjustable parameters

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