Digital Photogrammetric Systems Homework 4

Relative Orientation assigned 7 Nov 08, due 17 Nov 08

1. Select a static scene (no moving vehicles or people, etc.) and take 2 digital images with $B/H \approx 0.15$ (use the camera for which you determined pixel size)

2. Select 8–20 well defined points visible in both images (use convergence & $\approx 100\%$ overlap) and measure these points (ℓ, s), etc.

3. convert (ℓ, s) or (r, c) into (x, y)

4. use the accompanying notes to write 2 matlab functions and a main program to compute the relative orientation. (residuals should be 1–2 pixels)

5. turn in code (your own!) and numerical results. Fix $b_x = 100.0$.

Relative Orientation

make a function to evaluate coplanarity condition:

$$\boxed{F_{cp} = coplan(io, p, l)}$$

$io = [x0; y0; f]$

$p = [by; bz; \omega; \phi; k]$

$l = [x_l; y_l; x_r; y_r]$

$bx = 100$

$M_1 = I_3$

$M_2 = M_3(k) M_2(\phi) M_1(\omega)$

$$q_1 = M_1^T \begin{bmatrix} x_l - x_0 \\ y_l - y_0 \\ -f \end{bmatrix}$$

$$q_2 = M_2^T \begin{bmatrix} x_r - x_0 \\ y_r - y_0 \\ -f \end{bmatrix}$$

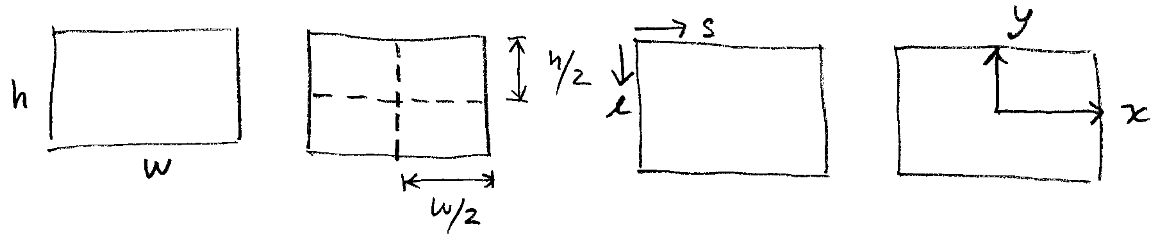
$$F_{cp} = \begin{vmatrix} bx & by & bz \\ q_1(1) & q_1(2) & q_1(3) \\ q_2(1) & q_2(2) & q_2(3) \end{vmatrix}$$

Make a function to linearize the coplanarity condition

```
[Fcp, a, b] = lin_coplan( io, p, l )
 io, p, l  as before
 dl= 0.001
 dp = [.00001 .00001 .00000001 .00000001 .00000001]
  F = coplan(io, p, l)
  for i = 1: 4
      l2 = l
      l2(i) = l2(i) + dl
      F2 = coplan(io, p, l2)
      a(i) = (F2-F)/dl
      end
   for i = 1:5
      p2 = p
      p2(i) = p2(i) + dp(i)
      F2 = coplan( io, p2, l )
      b(i) = (F2-F) /dp(i)
      end

   Fcp = F
```

Using these functions write a program to do relative orientation



$x = s - w/2$

$y = -(l - h/2)$

$l = (x_l, y_l, x_r, y_r)$

$l\phi = l$ (at the beginning)

$n =$ # points $\times$ 4

$\mu = 5$

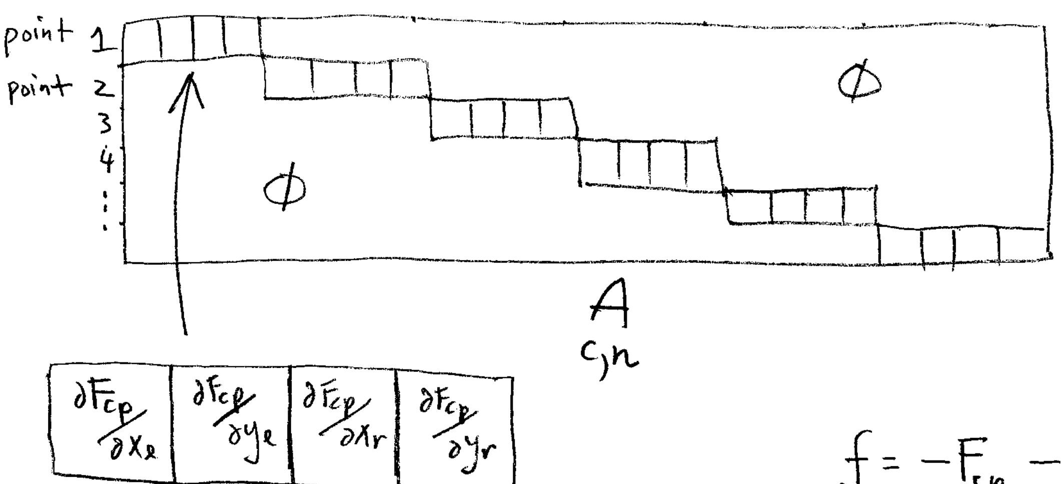
$n\phi = 5 + 3 * npnt$

$r = n - n\phi$

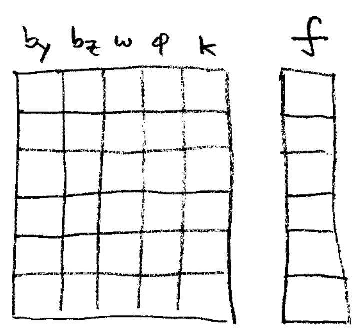
$c = r + \mu$

$p = [by ; bz ; \omega ; \phi ; k]$ initially all zeros

$\underset{c,n}{A}$, $\underset{c,\mu}{B}$, $\underset{c,1}{f}$, $W = I_n$



(a)



$\underset{c,\mu}{B}$

$f = -F_{cp} - A(l - l^0)$

$f = -F_{cp} - a(l - l^0)$

$Q = W^{-1}$

$Q_e = A Q A^T$

$W_e = Q_e^{-1}$

$N = B^T W_e B$

$t = B^T W_e f$

$\Delta = N^{-1} t$

$p = p + \Delta$ (iterate until Δ's are small)

$k = W_e (f - B\Delta)$

$V = Q A^T k$

$l^0 = l + V$

$$l = \begin{bmatrix} x_{l_1} \\ y_{l_1} \\ x_{r_1} \\ y_{r_1} \\ \hdashline x_{l_2} \\ y_{l_2} \\ x_{r_2} \\ y_{r_2} \\ \hdashline x_{l_3} \\ y_{l_3} \\ x_{r_3} \\ y_{r_3} \\ \hdashline \vdots \end{bmatrix}$$

since A is block diagonal, you can work only with the 4x1 sub-vectors of l and l^0