LAB 2 - Report

VARUN
AGGARWAL

IMAGE 09

$$Total\ Iterations: 5$$

$$\widehat{\sigma_0^2} = 2.167e - 04$$

$$Threshold = 10^{-8}$$

Summary

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	1.6000e+00	3.2000e+00	3.5000e+00	0.0000e+00	0.0000e+00	0.0000e+00
Final EOP Value	1.8897e+00	3.0359e+00	3.7350e+00	-1.9184e+01	-4.3450e+00	2.0501e+00
SD for EOPs	9.8035e-03	8.8936e-03	3.9152e-03	2.1493e-03	2.2192e-03	9.0243e-04
Delta EOP from BASC	1.9622e-08	1.5206e-08	5.9034e-09	1.8372e-07	1.9285e-07	6.1227e-08

Dispersion Matrix

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
X0	9.6109e-05	-2.7239e-06	-1.6399e-06	9.6521e-07	2.1529e-05	5.0135e-06
Y0	-2.7239e-06	7.9096e-05	-2.2975e-05	-1.8865e-05	-5.9816e-07	5.4154e-07
Z0	-1.6399e-06	-2.2975e-05	1.5329e-05	5.8205e-06	-5.0113e-07	-3.3482e-07
Omega (deg)	9.6521e-07	-1.8865e-05	5.8205e-06	4.6197e-06	2.0314e-07	-6.5548e-08
Phi (deg)	2.1529e-05	-5.9816e-07	-5.0113e-07	2.0314e-07	4.9247e-06	1.0929e-06
Kappa (deg)	5.0135e-06	5.4154e-07	-3.3482e-07	-6.5548e-08	1.0929e-06	8.1437e-07

EOPs over iterations

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	1.6000e+00	3.2000e+00	3.5000e+00	0.0000e+00	0.0000e+00	0.0000e+00
Iteration 1	1.9589e+00	2.5775e+00	3.4054e+00	-2.2255e-01	-4.0115e-02	3.0624e-02
Iteration 2	1.9190e+00	2.9273e+00	3.7355e+00	-3.1565e-01	-6.9540e-02	3.6687e-02
Iteration 3	1.8905e+00	3.0351e+00	3.7370e+00	-3.3466e-01	-7.5685e-02	3.5860e-02
Iteration 4	1.8897e+00	3.0359e+00	3.7350e+00	-3.3483e-01	-7.5834e-02	3.5782e-02
Iteration 5	1.8897e+00	3.0359e+00	3.7350e+00	-3.3483e-01	-7.5834e-02	3.5782e-02

IMAGE 09

Residuals X	Residuals Y
-9.1259e-03	-7.2668e-03
-5.5392e-03	2.4577e-04
-8.9980e-04	8.7198e-04
7.6428e-03	2.9308e-03
2.1732e-02	6.8278e-03
1.7264e-02	-6.9440e-03
1.5714e-03	-4.2503e-03
-1.0076e-03	-7.3490e-03
-6.8217e-04	-2.2903e-03
5.5982e-03	4.6218e-03
2.2853e-02	5.0299e-03
2.0849e-03	5.6196e-03
-5.1236e-03	-1.0008e-03
-5.5518e-03	3.1556e-03
-1.5575e-02	1.2531e-02
-9.6309e-04	3.4986e-03
1.1395e-03	8.9576e-03
-6.4612e-04	1.7559e-03
-6.3297e-04	7.7563e-04
-1.4522e-02	1.6327e-02
-1.8579e-02	-1.0490e-02
-4.9213e-03	-5.4186e-03
2.6750e-03	-1.9738e-02
2.5594e-03	-1.3976e-02
-1.0906e-02	5.3875e-03

Measured Image Co-ordinates (mm) x_a	Measured Image Co-ordinates (mm) y_a
-5.3877e+00	4.9374e+00
-2.9219e+00	4.8443e+00
-4.9439e-01	4.6817e+00
1.7983e+00	4.4987e+00
3.9337e+00	4.2987e+00
-5.0479e+00	2.4249e+00
-2.7741e+00	2.3434e+00
-5.5643e-01	2.2389e+00
1.5572e+00	2.1432e+00
3.5525e+00	2.0444e+00
-4.7123e+00	3.0738e-01
-2.6387e+00	2.4322e-01
-6.1275e-01	1.7794e-01
1.3400e+00	1.3305e-01
3.1989e+00	9.2280e-02
-4.4315e+00	-1.4901e+00
-2.5313e+00	-1.5383e+00
-6.6436e-01	-1.5756e+00
1.1514e+00	-1.5975e+00
2.8854e+00	-1.5967e+00
-4.1956e+00	-3.0278e+00
-2.4407e+00	-3.0661e+00
-7.0131e-01	-3.0987e+00
9.9217e-01	-3.0956e+00
2.6228e+00	-3.0774e+00

IMAGE 10

$$Total\ Iterations: 4$$

$$\widehat{\sigma_0^2} = 4.191e-05$$

$$Threshold = 10^{-8}$$

Summary

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	1.6361e+00	2.1841e+00	3.7271e+00	-1.8415e-02	-9.1752e-02	1.6060e+00
Final EOP Value	1.6805e+00	2.1659e+00	3.5086e+00	-4.1494e-01	-4.9369e+00	9.2477e+01
SD for EOPs	3.6917e-03	3.6434e-03	1.1079e-03	9.0229e-04	9.1565e-04	3.0840e-04
Delta EOP from BASC	2.6532e-05	2.9775e-06	4.9556e-06	8.3212e-05	3.1282e-04	4.9575e-05

Dispersion Matrix

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
X0	1.3628e-05	-1.9510e-06	2.2989e-07	5.0942e-07	3.3402e-06	-9.4792e-08
Y0	-1.9510e-06	1.3275e-05	-8.9664e-07	-3.2459e-06	-5.1142e-07	6.2440e-08
Z0	2.2989e-07	-8.9664e-07	1.2274e-06	2.4118e-07	4.8197e-08	-5.3369e-09
Omega (deg)	5.0942e-07	-3.2459e-06	2.4118e-07	8.1412e-07	1.3356e-07	-8.6030e-09
Phi (deg)	3.3402e-06	-5.1142e-07	4.8197e-08	1.3356e-07	8.3842e-07	-1.9525e-08
Kappa (deg)	-9.4792e-08	6.2440e-08	-5.3369e-09	-8.6030e-09	-1.9525e-08	9.5113e-08

EOPs over iterations

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	1.6361e+00	2.1841e+00	3.7271e+00	-1.8415e-02	-9.1752e-02	1.6060e+00
Iteration 1	1.6849e+00	2.1596e+00	3.4951e+00	-5.8678e-03	-8.5812e-02	1.6146e+00
Iteration 2	1.6805e+00	2.1658e+00	3.5085e+00	-7.2273e-03	-8.6160e-02	1.6140e+00
Iteration 3	1.6805e+00	2.1659e+00	3.5086e+00	-7.2420e-03	-8.6166e-02	1.6140e+00
Iteration 4	1.6805e+00	2.1659e+00	3.5086e+00	-7.2420e-03	-8.6166e-02	1.6140e+00

IMAGE 10

Residuals	
X	Y
-6.6692e-03	-2.5122e-03
4.1185e-04	1.4938e-03
2.8644e-03	5.6031e-04
5.0626e-03	-2.5870e-03
3.0542e-03	4.3497e-04
-4.5266e-04	-1.6212e-03
-4.7744e-03	-8.5173e-04
1.8819e-03	-1.3400e-03
2.1246e-03	-1.0629e-03
1.9392e-03	-2.2475e-03
-3.0407e-03	-3.9211e-03
-1.5160e-03	-7.0483e-03
2.4444e-03	-5.4390e-03
1.3557e-03	-4.9984e-03
-1.6863e-04	-1.5924e-03
-3.9452e-03	5.6887e-03
-5.3150e-04	3.7276e-03
-1.2248e-03	1.1518e-03
-9.9875e-04	-5.0982e-04
5.8577e-03	1.4762e-02
6.6226e-03	5.0771e-03
-2.2598e-03	5.0348e-04
-7.7244e-03	-6.6451e-05
1.3230e-03	2.8813e-03

Measured Image Co-ordinates (mm)	
x_a	y_a
4.2568e+00	4.6788e+00
4.0854e+00	2.3093e+00
3.8851e+00	4.1800e-02
3.6800e+00	-2.0823e+00
3.4790e+00	-4.0521e+00
1.9221e+00	4.7589e+00
1.7977e+00	2.3888e+00
1.6796e+00	1.1750e-01
1.5550e+00	-2.0216e+00
1.4339e+00	-4.0200e+00
-3.9670e-01	4.8045e+00
-4.7240e-01	2.4500e+00
-5.3340e-01	1.9210e-01
-5.9150e-01	-1.9492e+00
-6.4540e-01	-3.9707e+00
-2.6632e+00	4.8258e+00
-2.7100e+00	2.5321e+00
-2.7285e+00	2.9180e-01
-2.7338e+00	-1.8528e+00
-2.7224e+00	-3.8693e+00
-4.8100e+00	4.8076e+00
-4.8779e+00	2.5851e+00
-4.8782e+00	3.6970e-01
-4.8346e+00	-1.7626e+00

IMAGE 14

$$Total\ Iterations: 4$$

$$\widehat{\sigma_0^2} = 7.2144e-05$$

$$Threshold = 10^{-8}$$

Summary

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	9.3801e-01	2.4928e+00	2.9883e+00	-1.4337e-01	-3.0529e-01	1.5814e+00
Final EOP Value	1.0095e+00	2.5332e+00	2.8064e+00	-8.9398e+00	-1.7052e+01	9.1181e+01
SD for EOPs	3.8604e-03	3.7749e-03	2.7233e-03	1.1152e-03	1.2504e-03	5.4805e-04
Delta EOP from BASC	7.6935e-06	1.7155e-05	7.1523e-06	3.2238e-04	1.9592e-04	6.1085e-05

Dispersion Matrix

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
X0	1.4903e-05	6.8329e-07	7.2055e-06	2.3765e-07	4.6886e-06	2.0271e-07
Y0	6.8329e-07	1.4250e-05	-3.4952e-06	-4.0880e-06	-2.2047e-08	-1.1044e-06
Z0	7.2055e-06	-3.4952e-06	7.4166e-06	1.4050e-06	2.6646e-06	4.5797e-07
Omega (deg)	2.3765e-07	-4.0880e-06	1.4050e-06	1.2437e-06	1.6500e-07	3.2188e-07
Phi (deg)	4.6886e-06	-2.2047e-08	2.6646e-06	1.6500e-07	1.5636e-06	1.0213e-07
Kappa (deg)	2.0271e-07	-1.1044e-06	4.5797e-07	3.2188e-07	1.0213e-07	3.0036e-07

EOPs over iterations

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	9.3801e-01	2.4928e+00	2.9883e+00	-1.4337e-01	-3.0529e-01	1.5814e+00
Iteration 1	1.0147e+00	2.5411e+00	2.7957e+00	-1.5778e-01	-2.9739e-01	1.5918e+00
Iteration 2	1.0095e+00	2.5332e+00	2.8063e+00	-1.5604e-01	-2.9761e-01	1.5914e+00
Iteration 3	1.0095e+00	2.5332e+00	2.8064e+00	-1.5603e-01	-2.9761e-01	1.5914e+00
Iteration 4	1.0095e+00	2.5332e+00	2.8064e+00	-1.5603e-01	-2.9761e-01	1.5914e+00

IMAGE 14

Residuals

X	Y
-9.1494e-03	7.1838e-03
-7.1017e-03	1.0709e-05
6.2206e-05	-5.1577e-03
2.1545e-03	-5.2476e-04
3.7755e-03	2.8729e-03
5.4394e-04	-6.8885e-04
2.6088e-03	-1.0071e-02
-3.2874e-03	-6.9874e-03
8.8390e-03	-2.9649e-03
3.5038e-03	-1.4876e-03
7.0983e-04	-4.2513e-03
-1.2045e-02	2.3366e-03
5.4908e-03	2.6080e-04
6.5868e-03	4.6639e-03
2.4318e-04	3.8286e-03
-7.2762e-03	6.7173e-03
5.8241e-03	1.4691e-03
3.4599e-03	1.2703e-03
-1.0075e-03	3.3365e-04

Measured Image Co-ordinates (mm)

x_a	y_a
5.0408e+00	-3.3540e-01
4.5063e+00	-2.7179e+00
4.0613e+00	-4.6482e+00
2.4057e+00	2.6935e+00
2.1457e+00	-1.7670e-01
1.9236e+00	-2.5023e+00
1.7394e+00	-4.4140e+00
-5.4660e-01	2.7038e+00
-5.0720e-01	-2.9400e-02
-4.8000e-01	-2.2763e+00
-4.5620e-01	-4.1595e+00
-3.1775e+00	2.7274e+00
-2.9114e+00	1.3230e-01
-2.6867e+00	-2.0413e+00
-2.4950e+00	-3.8797e+00
-5.4743e+00	2.7125e+00
-5.0707e+00	2.5380e-01
-4.6992e+00	-1.8410e+00
-4.3740e+00	-3.6312e+00

IMAGE 18

$$Total\ Iterations: 4$$

$$\widehat{\sigma_0^2} = 1.3664e - 04$$

$$Threshold = 10^{-8}$$

Summary

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	1.9813e+00	1.1559e+00	3.9338e+00	2.2549e-01	2.7793e-03	1.5747e+00
Final EOP Value	2.1016e+00	1.1947e+00	3.6996e+00	1.3367e+01	1.4655e+00	9.0231e+01
SD for EOPs	7.2693e-03	7.0928e-03	2.5152e-03	1.6975e-03	1.6901e-03	6.4748e-04
Delta EOP from BASC	1.2351e-05	3.6947e-06	4.9635e-06	1.0053e-05	1.3206e-04	1.0902e-05

Dispersion Matrix

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
X0	5.2843e-05	-1.1007e-06	-4.0257e-06	3.5920e-07	1.2145e-05	-2.2374e-06
Y0	-1.1007e-06	5.0308e-05	8.4483e-06	-1.1905e-05	-2.2545e-07	5.2030e-07
Z0	-4.0257e-06	8.4483e-06	6.3260e-06	-2.0936e-06	-9.9653e-07	2.6862e-07
Omega (deg)	3.5920e-07	-1.1905e-05	-2.0936e-06	2.8815e-06	7.8427e-08	-1.1986e-07
Phi (deg)	1.2145e-05	-2.2545e-07	-9.9653e-07	7.8427e-08	2.8563e-06	-4.9485e-07
Kappa (deg)	-2.2374e-06	5.2030e-07	2.6862e-07	-1.1986e-07	-4.9485e-07	4.1923e-07

EOPs over iterations

	X0	Y0	Z0	Omega (deg)	Phi (deg)	Kappa (deg)
Initial Approximation	9.3801e-01	2.4928e+00	2.9883e+00	-1.4337e-01	-3.0529e-01	1.5814e+00
Iteration 1	1.0147e+00	2.5411e+00	2.7957e+00	-1.5778e-01	-2.9739e-01	1.5918e+00
Iteration 2	1.0095e+00	2.5332e+00	2.8063e+00	-1.5604e-01	-2.9761e-01	1.5914e+00
Iteration 3	1.0095e+00	2.5332e+00	2.8064e+00	-1.5603e-01	-2.9761e-01	1.5914e+00
Iteration 4	1.0095e+00	2.5332e+00	2.8064e+00	-1.5603e-01	-2.9761e-01	1.5914e+00

IMAGE 18

Residuals X	Residuals Y
-4.5648e-03	-1.8875e-03
-2.4678e-03	-2.3467e-03
-1.1436e-03	-1.7240e-05
-4.1860e-04	3.4880e-03
-8.8269e-03	5.6334e-03
2.0109e-04	-1.1697e-03
1.1524e-03	-5.4694e-04
5.6302e-03	2.1803e-03
5.2687e-03	1.2950e-03
2.0063e-03	-3.0703e-03
6.5752e-03	3.3518e-03
6.0511e-03	3.7739e-03
8.8111e-03	2.0679e-03
8.4652e-03	-4.4021e-03
1.1348e-03	-1.2135e-02
5.8902e-03	7.1434e-04
2.4951e-03	4.4472e-03
4.3101e-04	5.0749e-03
-1.4891e-03	-3.8245e-03
-1.5643e-03	-6.0432e-03
3.0723e-03	-1.6205e-02
-1.4727e-02	-5.0023e-03
-2.0604e-02	1.1902e-03
-1.2807e-02	1.2664e-03
1.4222e-02	2.1839e-02

Measured Image Co-ordinates (mm) x_a	Measured Image Co-ordinates (mm) y_a
3.2893e+00	3.8357e+00
3.2950e+00	2.0059e+00
3.2772e+00	1.6290e-01
3.2471e+00	-1.6697e+00
3.2149e+00	-3.4760e+00
1.5691e+00	4.0453e+00
1.5740e+00	2.1277e+00
1.5795e+00	1.8440e-01
1.5697e+00	-1.7649e+00
1.5570e+00	-3.6961e+00
-3.3780e-01	4.2673e+00
-3.4410e-01	2.2563e+00
-3.3750e-01	2.0580e-01
-3.3810e-01	-1.8635e+00
-3.4120e-01	-3.9356e+00
-2.4459e+00	4.4805e+00
-2.4810e+00	2.4023e+00
-2.4964e+00	2.4660e-01
-2.5036e+00	-1.9536e+00
-2.4987e+00	-4.1554e+00
-4.7066e+00	4.6783e+00
-4.8269e+00	2.5441e+00
-4.8928e+00	2.7650e-01
-4.9179e+00	-2.0524e+00
-4.8917e+00	-4.3690e+00

Issues and Explanation

- The EOPs for all the images, when compared to output of BASC, the difference is of the order of $10^{-4}\ to\ 10^{-7}$
- The residuals from this lab exercise are also very close to residuals from BASC
- It took around 4 to 5 iterations for the solution to converge with a threshold for change in $\Delta\widehat{\sigma_0^2} = 10^{-8}$
- $\widehat{\sigma_0^2}$ is of the order of $10^{-4} and\ 10^{-5}$
- IOPs were used from Lab 1
- One issue encountered during this project was with the linearization of Co-linearity equations
- There were multiple terms with very similar variable name hence to completely understand the meaning of each variable, derivation had to be done from scratch
- While, coding in the mathematical equations, a misplaced plus or minus within the code let to diverging output
- Debugging required a lot of focus because of the number of equations and variables in the model

Computer Pseudo Code

main.m

Declare common parameters -> IOPs, Ground co-ordinates of targets

For each image:
    Get image co-ordinates for each image

    Call **LSA.m function**

    Generate various results by calling **results.m** function

LSA.m

Define threshold for convergence

Start loop (max 100 iterations):
    Calculate A matrix by calling **calc_A.m**
    Calculate y matrix by calling **calc_y.m**
    Calculate x_hat
    Update EOPs -> old_EOPs + x_hat
    Calculate e vector and $\widehat{\sigma_0^2}$

    If $\left|\widehat{\sigma_{0ld}^2} - \widehat{\sigma_{new}^2}\right| < threshold$
    Then Calculate D and break out of loop

Computer Pseudo Code

Calc_A.m

```
Calculate rotation matrix by calling rotation.m
Calculate N_x, N_y and D
For i=1 to num_of_targets
    Calculate a_i , b_i  → i = 1,2,3,4,5,6
    add to A matrix
End loop
```

Calculate rotation matrix by calling **rotation.m**
Calculate N_x, N_y and D
For i=1 to num_of_targets
    Calculate a_i , b_i $\rightarrow i = 1,2,3,4,5,6$
    add to A matrix
End loop

Calc_y.m

Calculate $\bar{x}, \bar{y}, r$
Calculate $\Delta x_r, \Delta \mathrm{y_r}$ (radial distortions)
Calculate $\Delta x_d, \Delta y_d$ (de-centering distortion)
Remove distortions from image co-ordinate (x_{corr}, y_{corr})
Calculate y vector $(x_{corr} - x_p + c\ \frac{N_x}{D}, y_{corr} - y_p + c\ \frac{N_y}{D})$

rotation.m

Calculate rotation ω
Calculate rotation ϕ
Calculate rotation κ

Multiply the rotation matrices to get final rotation matrix

Code on Github:
https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – main.py

```
%% Common Parameters

% IOPs
 IOP = [6.7451660984e-2,-1.1709829919e-1,8.1671200690];
dist = [-2.9350008918e-4, 9.2190322166e-6,-2.2562559450e-7,6.1878890685e-5,-7.2907688047e-5,0];

% Ground co-ordinates of targets
XA = [-0.044038221723;0.89829592555;1.8428648416;2.7928032424;3.731135328;-0.036927916146;0.90108606018;1.8511589932;2.8010339963;3.7422959221;-0.037704361429;0.90892725171;1.8588605461;2.8062554595;3.7520655984;-0.024722332105;0.91454137562;1.8596674595;2.8078118825;3.7544680979;0;0.9263914178;1.8642980884;2.8111736654;3.7628755375];

YA =
[3.751130948;3.7594793917;3.7736433303;3.7823238561;3.7801397656;2.8057675589;2.8180218203;2.8257925207;2.8332144474;2.8378518827;1.8634346772;1.8678901457;1.8758730363;1.8838128654;1.8922797004;0.91840024749;0.92066047041;0.92760553219;0.93553829291;0.94557055946;0;-0.00026178267339;-0.0021390199929;-0.0019752230297;0];

ZA = [0.072640804133;0.048121990404;0;-0.047649472361;-0.070838477818;0.054102199693;0.026008316122;0.0059282538012;-0.023295585379;-0.044574905144;0.016369939657;0.010208246675;0.0071618170061;-0.0096698096332;-0.020910207448;-0.0091692255168;0.003877584251;0.003049069041;-0.0030509411619;-0.0098127723745;0;0.017342094476;0.016047411487;0.006582607018;0];

%% Image 1 - image_09

[EOP_i,EOP_basc,xa,ya] = data(9); % get data for image 9
[EOP_9,e_9,sigma_hat_9,iter_9,D_9,EOP_history_9] = LSA(xa,ya,XA,YA,ZA,IOP,dist,EOP_i); % run LSA
[table_9_summary,table_9_D,table_9_hist] =
results(EOP_i,EOP_9,D_9,EOP_basc,EOP_history_9,iter_9); % complie results

%% Image 1 - image_10

[EOP_i,EOP_basc,xa,ya] = data(10); % get data for image 10
[EOP_10,e_10,sigma_hat_10,iter_10,D_10,EOP_history_10] =
LSA(xa,ya,XA(1:24),YA(1:24),ZA(1:24),IOP,dist,EOP_i); % run LSA
[table_10_summary,table_10_D,table_10_hist] =
results(EOP_i,EOP_10,D_10,EOP_basc,EOP_history_10,iter_10); % complie results

%% Image 1 - image_14

[EOP_i,EOP_basc,xa,ya] = data(14); % get data for image 14
[EOP_14,e_14,sigma_hat_14,iter_14,D_14,EOP_history_14] =
LSA(xa,ya,XA([3:5,7:10,12:15,17:20,22:25]),YA([3:5,7:10,12:15,17:20,22:25]),ZA([3:5,7:10,12:15,17:20,22:25]),IOP,dist,EOP_i); % run LSA
[table_14_summary,table_14_D,table_14_hist] =
results(EOP_i,EOP_14,D_14,EOP_basc,EOP_history_14,iter_14); % complie results

%% Image 1 - image_18

[EOP_i,EOP_basc,xa,ya] = data(18); % get data for image 14
[EOP_18,e_18,sigma_hat_18,iter_18,D_18,EOP_history_18] = LSA(xa,ya,XA,YA,ZA,IOP,dist,EOP_i);
% run LSA
[table_18_summary,table_18_D,table_18_hist] =
results(EOP_i,EOP_18,D_18,EOP_basc,EOP_history_14,iter_14); % complie results
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – LSA.py

```
function [EOP,e,sigma_hat,iter,D,EOP_history] = LSA(xa,ya,XA,YA,ZA,IOP,dist,EOP)

%% parameters
max_iter = 10000;
thres = 1e-8;
sigma_hat = 0;
num_GCP = length(xa);

% Unpack EOPs and IOPs
[X0,Y0,Z0,omega,phi,kappa] = update_EOP(EOP, zeros(6,1));
[xp,yp,c] = assign_IOP(IOP);
EOP_history = EOP;
%% LSA iterations
for iter=1:max_iter

    % update sigma_hat_prev
    sigma_hat_prev = sigma_hat;

    % OAX, OAY, OAZ
    OAX = XA - X0;
    OAY = YA - Y0;
    OAZ = ZA - Z0;

    % calculate A_50x6 (num_GCP,c,OAX,OAY,OAZ,omega,phi,kappa)
    [A,Nx,Ny,D] = calc_A(num_GCP, c, OAX, OAY, OAZ, omega, phi, kappa);

    % calculate y_50x1 (dist,xa,ya,xp,yp,c,Nx_mat,Ny_mat,D_mat)
    y = calc_y(dist,xa,ya,xp,yp,c,Nx,Ny,D);

    % calculate x_hat
    P = eye(num_GCP*2);
    x_hat = inv(A'*P*A)*A'*P*y;

    % update EOPs
    [X0,Y0,Z0,omega,phi,kappa,EOP] = update_EOP(EOP, x_hat);
    EOP_history = [EOP_history;EOP]

    % calculate e and sigma_hat
    e = y - A*x_hat;
    sigma_hat = (e'*P*e)/(num_GCP - length(x_hat));

    % break out of loop is threshold is reached
    if abs(sigma_hat - sigma_hat_prev) < thres
          D = sigma_hat*inv(A'*P*A);
          e = [e(1:2:end),e(2:2:end)];
        EOP = [X0,Y0,Z0,rad2deg(omega),rad2deg(phi),rad2deg(kappa)];
        break
    end

end
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – calc_A.py

```
function [A,Nx_mat,Ny_mat,D_mat] = calc_A(num_GCP,c,OAX,OAY,OAZ,omega,phi,kappa)

%% calculate rotation
r = rotation(omega,phi,kappa);

%% calculate N and D (OAX,OAY,OAZ,rot)
Nx_mat = r(1,1)*OAX + r(2,1)*OAY + r(3,1)*OAZ;
Ny_mat = r(1,2)*OAX + r(2,2)*OAY + r(3,2)*OAZ;
 D_mat = r(1,3)*OAX + r(2,3)*OAY + r(3,3)*OAZ;

%% calculate aij, bij

% create an empty A matrix
A = zeros(num_GCP*2,6);

% loop over all ground control points
for i=1:num_GCP

    % Nx, Ny, D for ith target
    Nx = Nx_mat(i);
    Ny = Ny_mat(i);
     D =  D_mat(i);

    % aij for ith GCP
    a1 = r(1,1)*D - r(1,3)*Nx;
    a2 = r(2,1)*D - r(2,3)*Nx;
    a3 = r(3,1)*D - r(3,3)*Nx;
    a4 = D*(r(3,1)*OAY(i) - r(2,1)*OAZ(i)) + Nx*(r(2,3)*OAZ(i) - r(3,3)*OAY(i));
    a5 = D*D*cos(kappa) - Nx*(-Nx*cos(kappa) + Ny*sin(kappa));
    a6 = -1 * Ny*D;

    % bij for ith GCP
    b1 = r(1,2)*D - r(1,3)*Ny;
    b2 = r(2,2)*D - r(2,3)*Ny;
    b3 = r(3,2)*D - r(3,3)*Ny;
    b4 = D*(r(3,2)*OAY(i) - r(2,2)*OAZ(i)) + Ny*(r(2,3)*OAZ(i) - r(3,3)*OAY(i));
    b5 = -1*D*D*sin(kappa) - Ny*(-Nx*cos(kappa) + Ny*sin(kappa));
    b6 = Nx*D;

    % multiplying by c/D^2 before appending into A
    A(2*i-1,:) = [a1,a2,a3,a4,a5,a6]*c/D^2;
      A(2*i,:) = [b1,b2,b3,b4,b5,b6]*c/D^2;

end
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – calc_y.py

```
function [y] = calc_y(dist,xa,ya,xp,yp,c,Nx,Ny,D)

%% calculate x_corr and y_corr

% unpack distortion parameters
k1 = dist(1);
k2 = dist(2);
k3 = dist(3);
p1 = dist(4);
p2 = dist(5);
p3 = dist(6);

% calculate x_bar, y_bar, r
x_bar = xa - xp;
y_bar = ya - yp;
r = sqrt(x_bar.^2 + y_bar.^2);

% calculate delta_xr, delta_yr - radial distortion
delta_xr =  x_bar .* (k1*r.^2 + k2*r.^4 + k3*r.^6);
delta_yr =  y_bar .* (k1*r.^2 + k2*r.^4 + k3*r.^6);

% calculate delta_xd, delta_yd - de-centering lens distortion
delta_xd = (1+p3*r.^2).*(p1*(r.^2 + 2*x_bar.^2) +
2*p2*x_bar.*y_bar);
delta_yd = (1+p3*r.^2).*(2*p1*x_bar.*y_bar + p2*(r.^2 +
2*y_bar.^2));

% cremove distortions
x_corr = xa - delta_xr - delta_xd;
y_corr = ya - delta_yr - delta_yd;

%% calcualte y

x_corr = x_corr - (xp - c*Nx./D);
y_corr = y_corr - (yp - c*Ny./D);

% alternate merge in order : x_corr and y_corr
y = [x_corr y_corr]';
y = y(:);

end
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – rotation.py

```
function [rot]=rotation(omega, phi, kappa)

rot_o = [1 0 0; 0 cos(omega) -sin(omega); 0
sin(omega) cos(omega)];

rot_p = [cos(phi) 0 sin(phi); 0 1 0; -sin(phi) 0
cos(phi)];

rot_k = [cos(kappa) -sin(kappa) 0; sin(kappa)
cos(kappa) 0; 0 0 1];

rot = rot_o * rot_p * rot_k;

end
```

Compute Code – update_EOP.py

```
function
[X0,Y0,Z0,omega,phi,kappa,EOP]=update_EOP(EOP, x_hat)

       X0 = EOP(1) + x_hat(1);
       Y0 = EOP(2) + x_hat(2);
       Z0 = EOP(3) + x_hat(3);
    omega = EOP(4) + x_hat(4);
      phi = EOP(5) + x_hat(5);
    kappa = EOP(6) + x_hat(6);
    EOP = [X0,Y0,Z0,omega,phi,kappa];

end
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – results.py

```
function [summary,D,hist,e]=results(EOP_i, EOP_f, D, EOP_basc, hist,
iter)
    out = [EOP_i;EOP_f;transpose(diag(D).^0.5);abs(EOP_basc - EOP_f)];

      row=["Initial Approximation"];
    for i=1:iter
        row=[row,sprintf("Iteration %d",i)];
    end

    summary = array2table(out,'VariableNames',["X0","Y0","Z0","Omega
(deg)","Phi (deg)","Kappa (deg)"],'RowNames',["Initial
Approximation","Final EOP Value","SD for EOPs","Delta EOP from BASC"]);

    D = array2table(D,'VariableNames',["X0","Y0","Z0","Omega (deg)","Phi
(deg)","Kappa (deg)"],'RowNames',["X0","Y0","Z0","Omega (deg)","Phi
(deg)","Kappa (deg)"]);

    hist = array2table(hist,'VariableNames',["X0","Y0","Z0","Omega
(deg)","Phi (deg)","Kappa (deg)"],'RowNames',row);
end
```

Compute Code – assign_IOP.py

```
function [xp,yp,c]=assign_IOP(IOP)
     xp = IOP(1);
     yp = IOP(2);
      c = IOP(3);
end
```

Code on Github:

https://github.com/Salazar-Prime/photogrammetry/tree/master/Project%202

Compute Code – data.py

```
function [EOP_i,EOP_basc,xa,ya]=data(image_no)

    switch image_no
        case 9
            EOP_i = [1.60,3.20,3.5,0,0,0];
            EOP_basc = [1.88968674,3.03586621,3.73500036,-19.1841784,-
4.34497759,2.05014903];
            xa = [-5.3877;-2.92187;-0.49439;1.79834;3.93369;-5.04787;-
2.77407;-0.55643;1.5572;3.55246;-4.71235;-2.63868;-
0.61275;1.34004;3.19895;-4.43154;-2.5313;-0.66436;1.1514;2.88537;-4.19557;-
2.4407;-0.70131;0.99217;2.62277];
            ya =
[4.93742;4.84433;4.68172;4.49867;4.29866;2.42494;2.3434;2.23887;2.1432;2.04
441;0.30738;0.24322;0.17794;0.13305;0.09228;-1.49013;-1.53833;-1.5756;-
1.59753;-1.59671;-3.02777;-3.06606;-3.09871;-3.09563;-3.07743];
        case 10
            EOP_i = [1.636114 2.184056 3.727135 deg2rad(-1.055093)
deg2rad(-5.256979) deg2rad(92.014892)];
            EOP_basc = [1.68045678e+000   2.16587520e+000   3.50855335e+000
-4.14855263e-001  -4.93725230e+000   9.24769326e+001];
            xa =
[4.2568;4.0854;3.8851;3.68;3.479;1.9221;1.7977;1.6796;1.555;1.4339;-
0.3967;-0.4724;-0.5334;-0.5915;-0.6454;-2.6632;-2.71;-2.7285;-2.7338;-
2.7224;-4.81;-4.8779;-4.8782;-4.8346];
            ya= [4.6788;2.3093;0.0418;-2.0823;-
4.0521;4.7589;2.3888;0.1175;-2.0216;-4.02;4.8045;2.45;0.1921;-1.9492;-
3.9707;4.8258;2.5321;0.2918;-1.8528;-3.8693;4.8076;2.5851;0.3697;-1.7626];
        case 14
            EOP_i = [0.938012 2.492823 2.988345 deg2rad(-8.214459)
deg2rad(-17.492066) deg2rad(90.607889)];
            EOP_basc = [1.00952899e+000   2.53318361e+000   2.80635782e+000
-8.94008150e+000  -1.70515538e+001   9.11810993e+001];
            xa = [5.0408;4.5063;4.0613;2.4057;2.1457;1.9236;1.7394;-
0.5466;-0.5072;-0.48;-0.4562;-3.1775;-2.9114;-2.6867;-2.495;-5.4743;-
5.0707;-4.6992;-4.374];
            ya = [-0.3354;-2.7179;-4.6482;2.6935;-0.1767;-2.5023;-
4.414;2.7038;-0.0294;-2.2763;-4.1595;2.7274;0.1323;-2.0413;-
3.8797;2.7125;0.2538;-1.841;-3.6312];
        case 18
            EOP_i = [1.981303 1.155948 3.933779 deg2rad(12.919527)
deg2rad(0.159243) deg2rad(90.221535)];
            EOP_basc = [2.10162316e+000   1.19465520e+000   3.69958966e+000
1.33667830e+001   1.46559054e+000   9.02308070e+001];
            xa =
[3.2893;3.295;3.2772;3.2471;3.2149;1.5691;1.574;1.5795;1.5697;1.557;-
0.3378;-0.3441;-0.3375;-0.3381;-0.3412;-2.4459;-2.481;-2.4964;-2.5036;-
2.4987;-4.7066;-4.8269;-4.8928;-4.9179;-4.8917];
            ya = [3.8357;2.0059;0.1629;-1.6697;-
3.476;4.0453;2.1277;0.1844;-1.7649;-3.6961;4.2673;2.2563;0.2058;-1.8635;-
3.9356;4.4805;2.4023;0.2466;-1.9536;-4.1554;4.6783;2.5441;0.2765;-2.0524;-
4.369];

    end
end
```