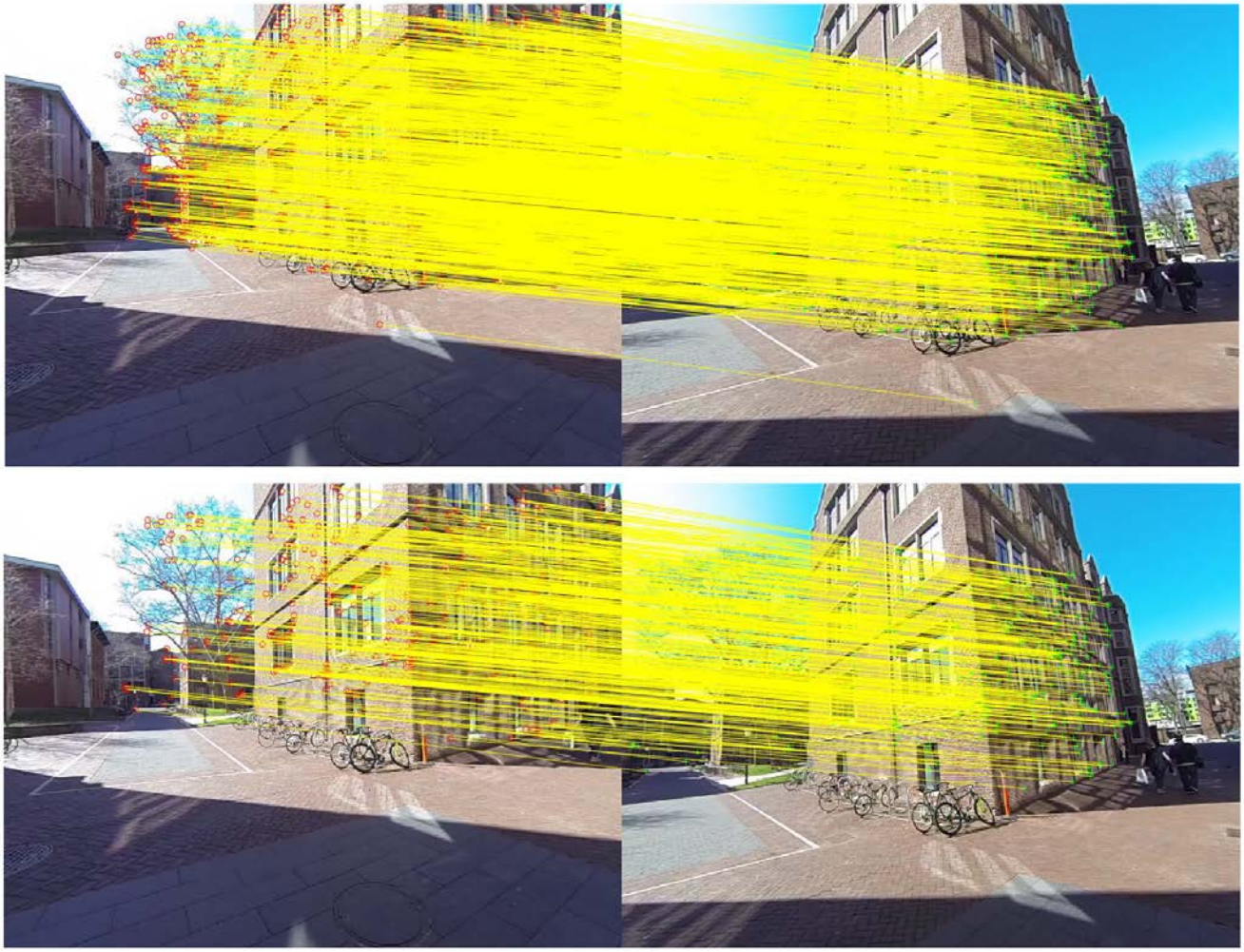
CIS580 Machine Perception Project2 Milestone3

Submitted by Wudao Ling

All the Visualization and Result are attached in file.

1. InlinersRANSAC

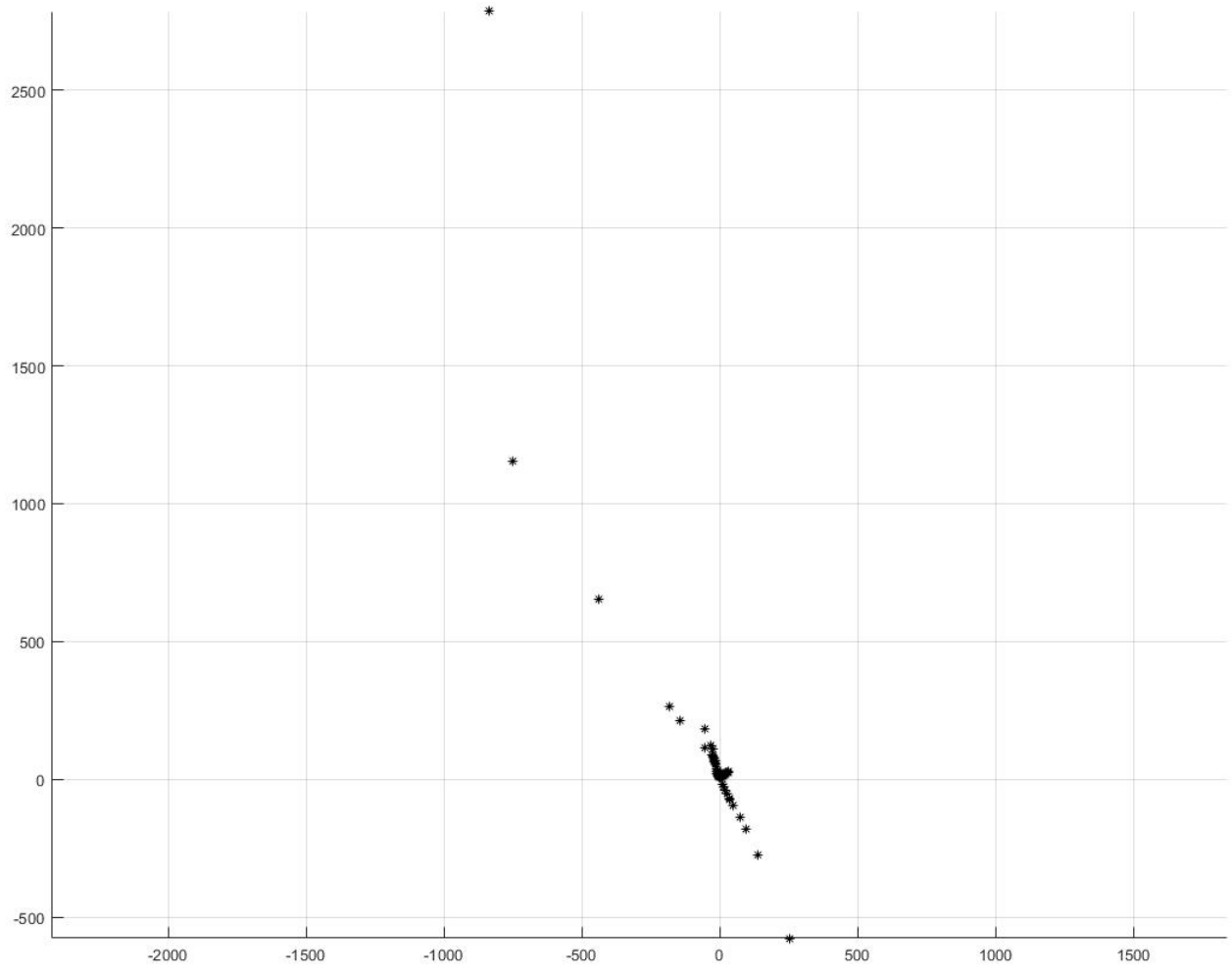
Threshold 0.5; Iteration log(1-0.99)/log(1-0.4^8); 500 inliners



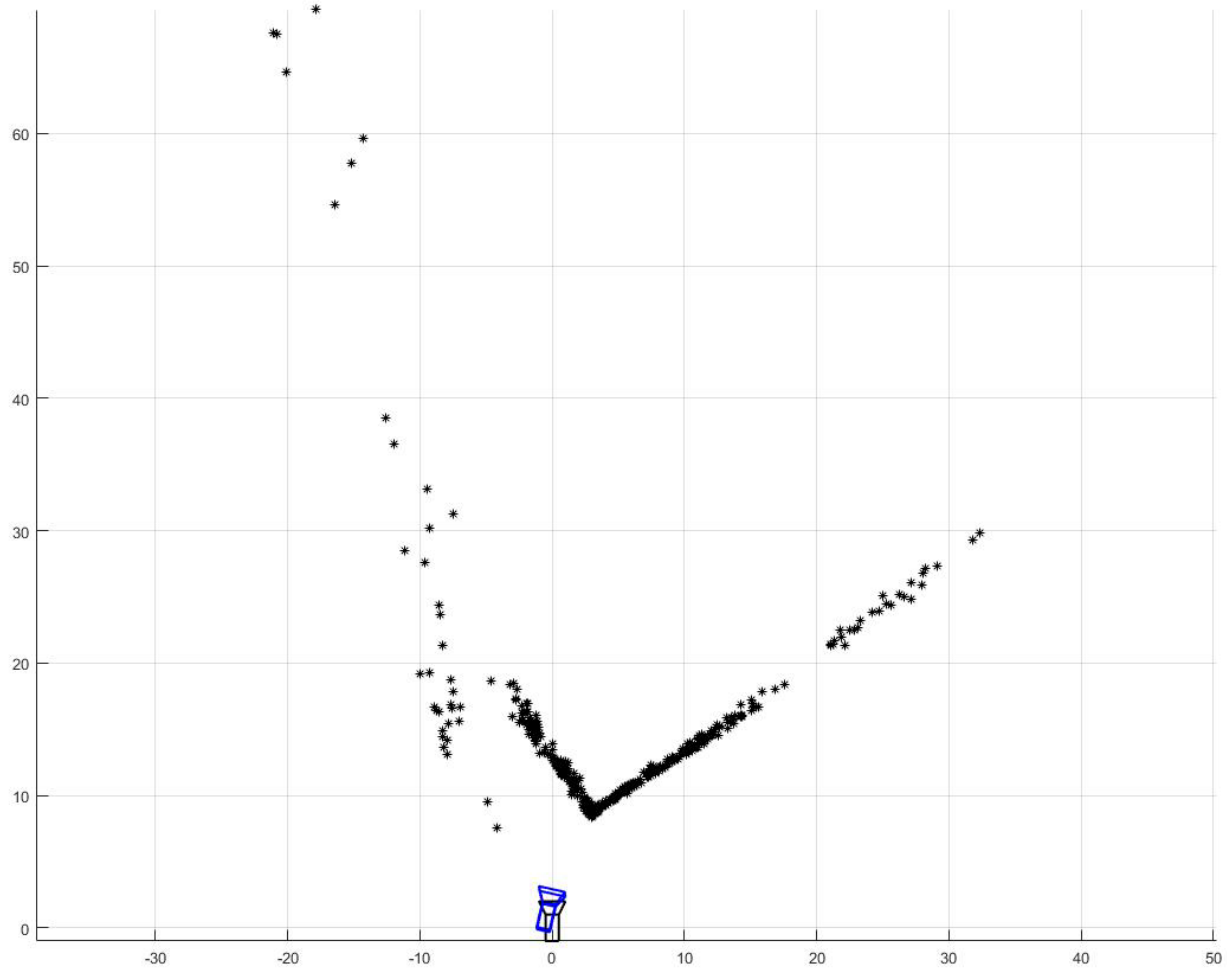
The image sequence I used is 123456, and image pair above is image 1 and 2.

2. Linear/Nonlinear triangulation

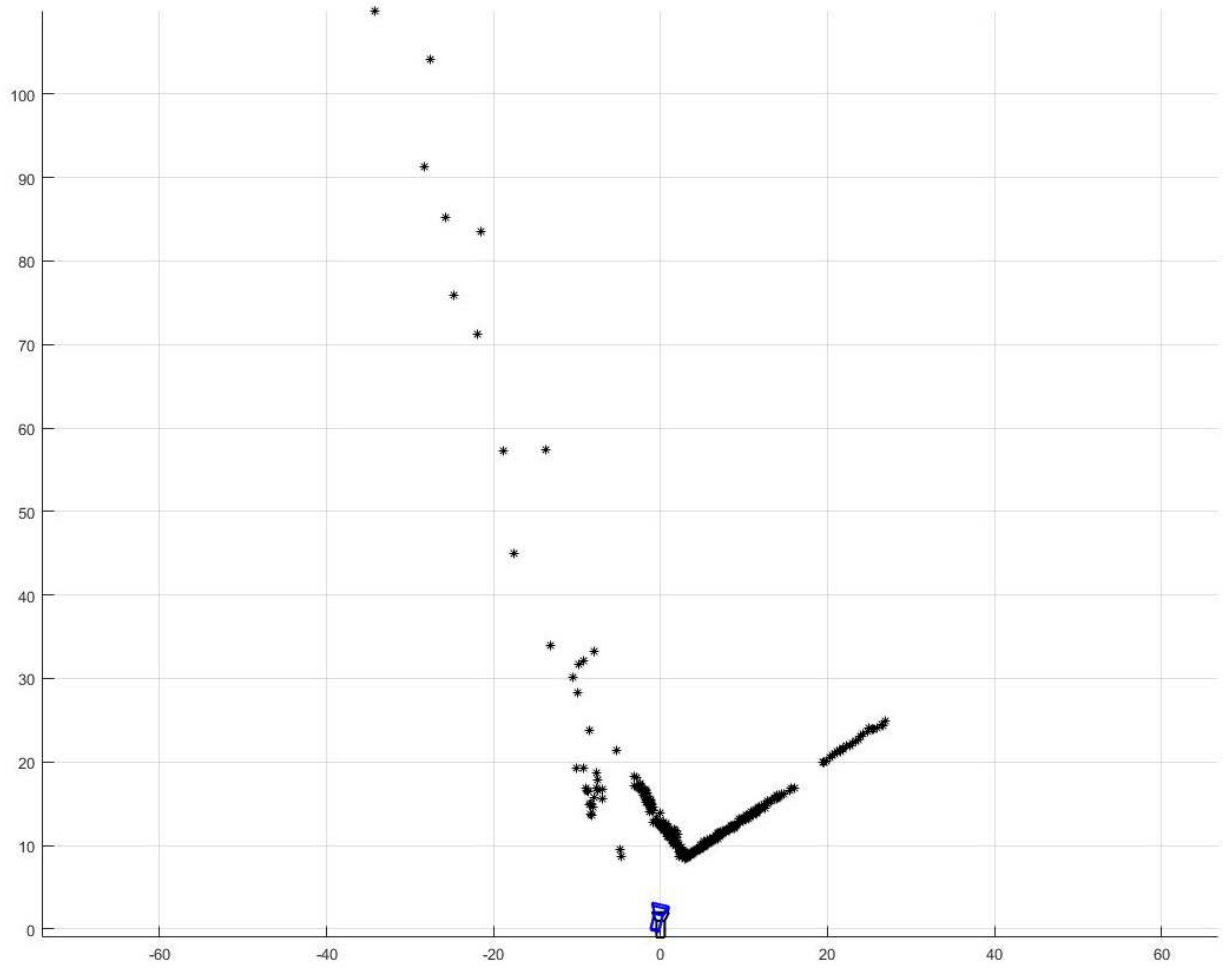
Linear Triangulation
Reprojection error: 0.738178014984072
(Matching features between image 1 and 2, reprojected points in red, real point in green)

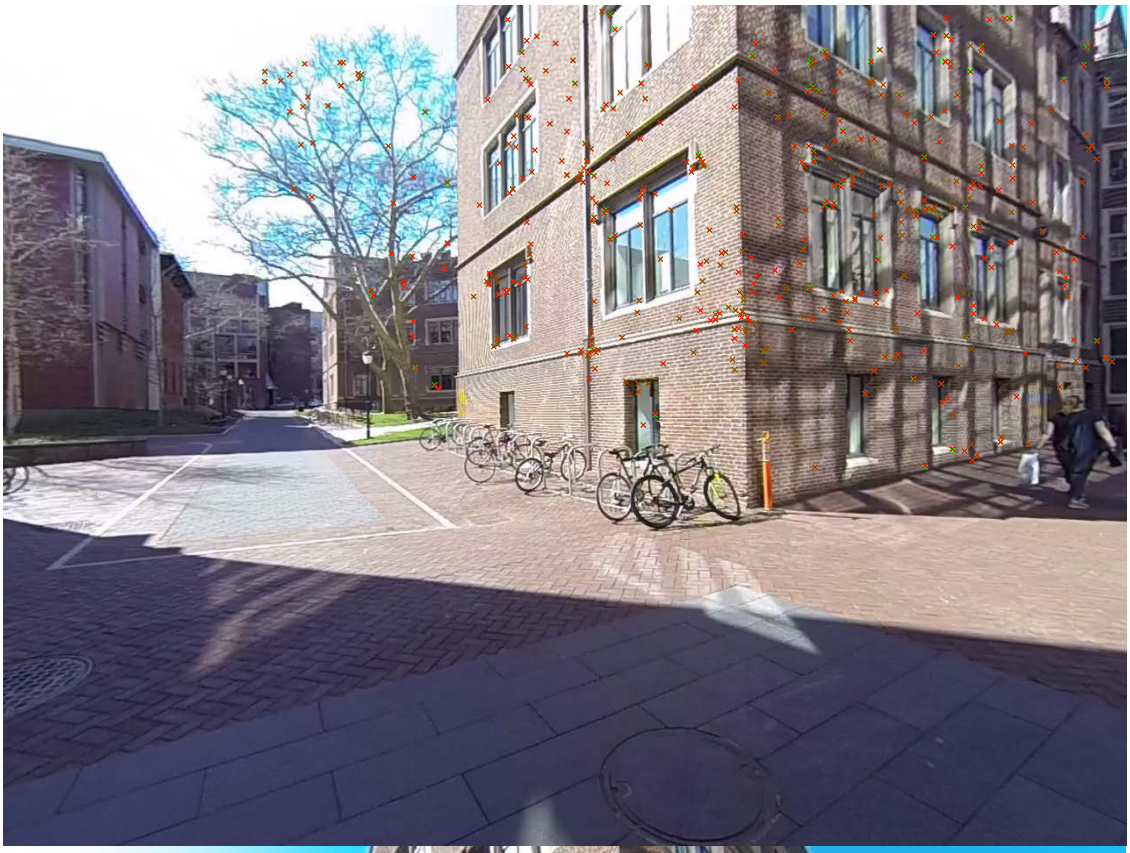


Linear Triangulation: after removing outliers with Z < 0 or distance > 80
Reprojection error: 0.773308691884858
(Matching features between image 1 and 2, reprojected points in red, real point in green)

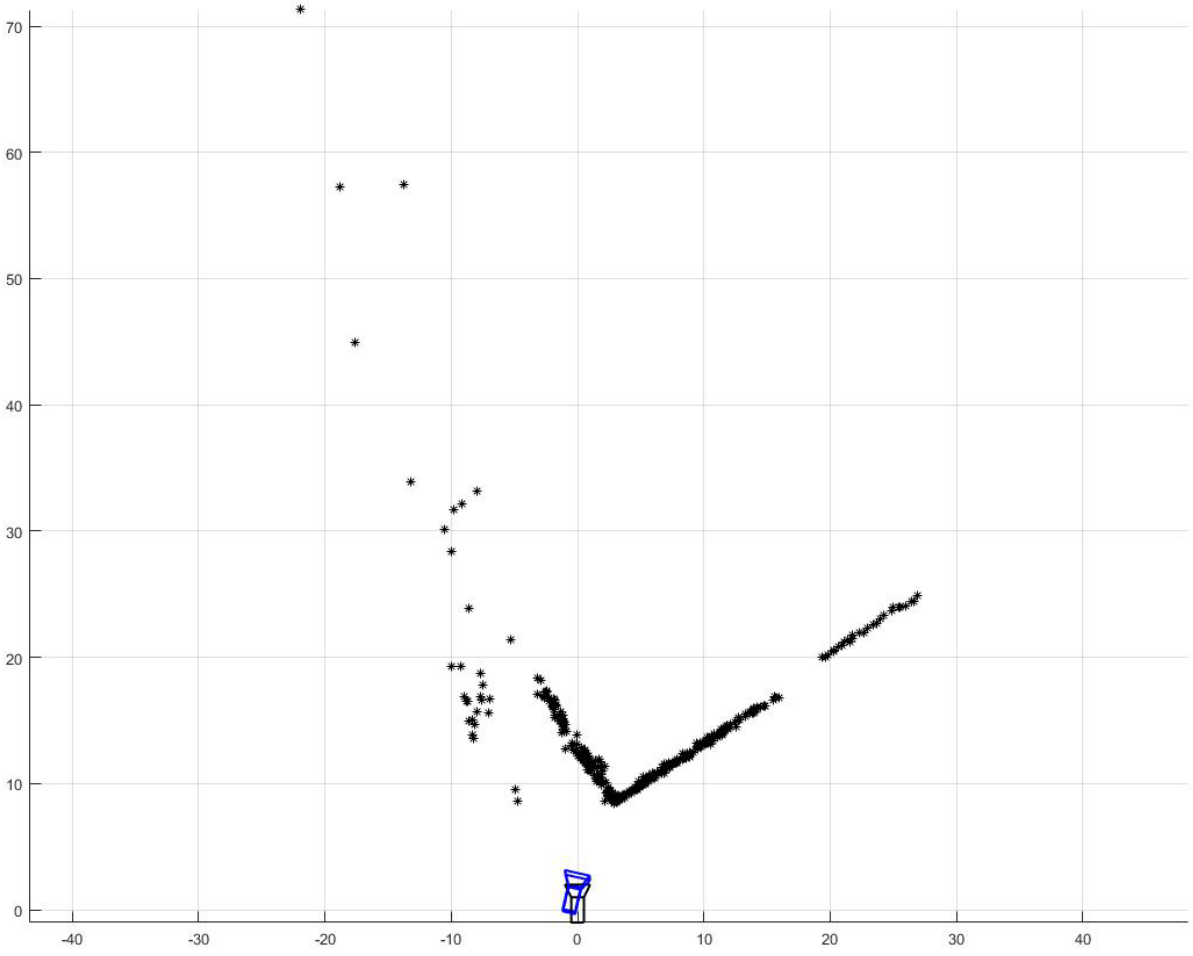


Nonlinear Triangulation
Reprojection error: 0.093986003890904
(Matching features between image 1 and 2, reprojected points in red, real point in green)





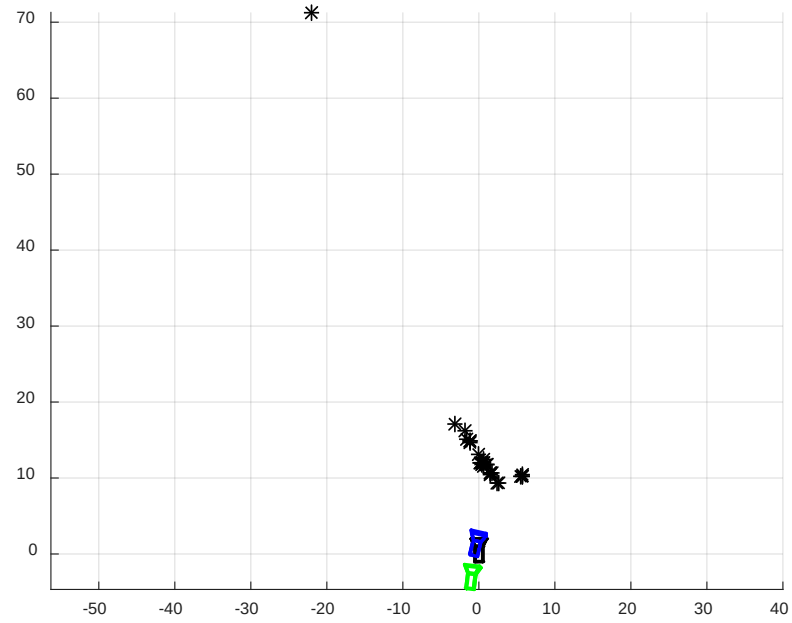
Nonlinear Triangulation (outliers removed): Z < 0 or distance > 80
Reprojection error: 0.093746406693864
(Matching features between image 1 and 2, reprojected points in red, real point in green)



3. PnP

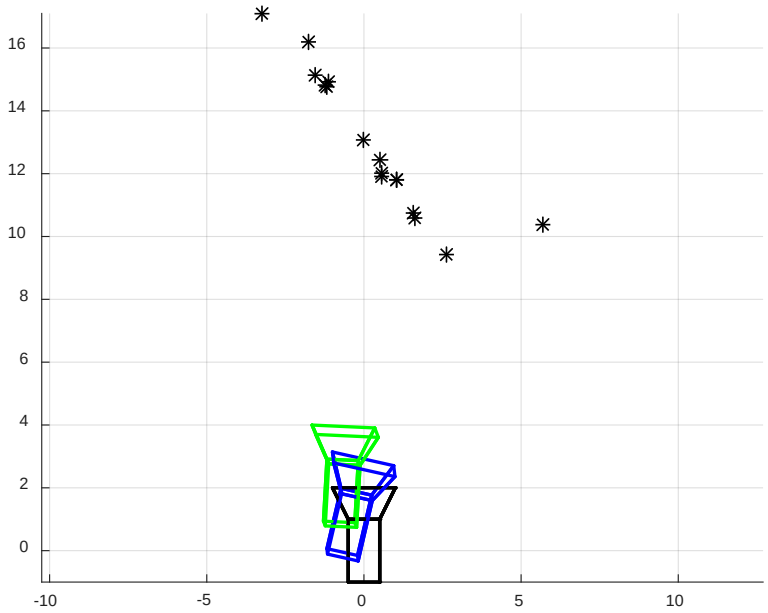
LinearPnP (before PnP RANSAC) Reprojection error:1.598022698606238e+04

Use all the 463 correspondences camera 3 has with previous cameras to calculate camera 3's position and orientation. (reprojected correspondences in red, real correspondences in green)



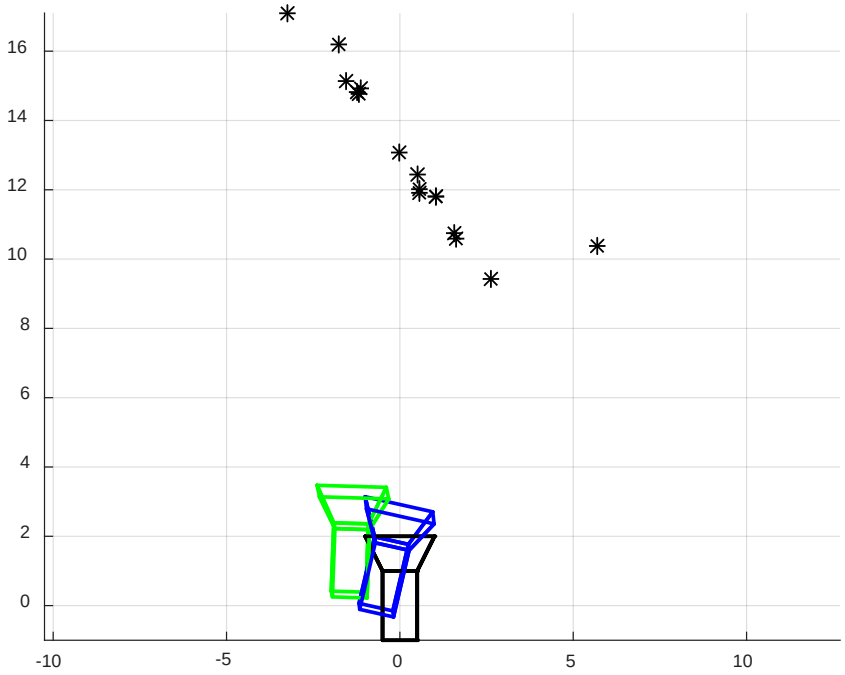
LinearPnP (After PnP RANSAC) Reprojection error: 1.989144568565782e+03

Threshold 10; Iteration log(1-0.99)/log(1-0.25^6); 16 inliners
(reprojected inliners correspondences in red, real inliners correspondences in green)



NonlinearPnP (Removed outliers in RANSAC) Reprojection error: 0.486728097675991

(reprojected inliners correspondences in red, real inliners correspondences in green)

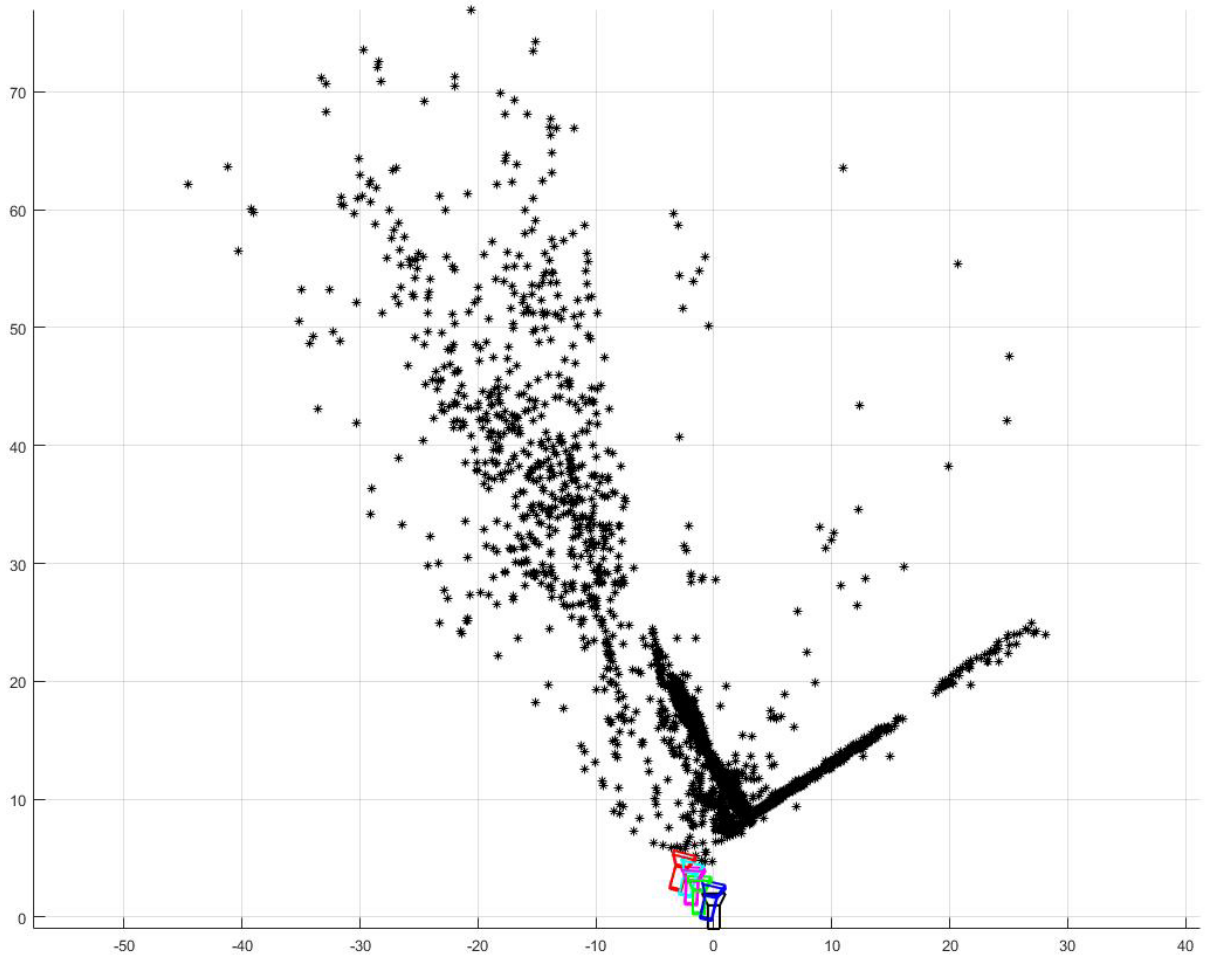


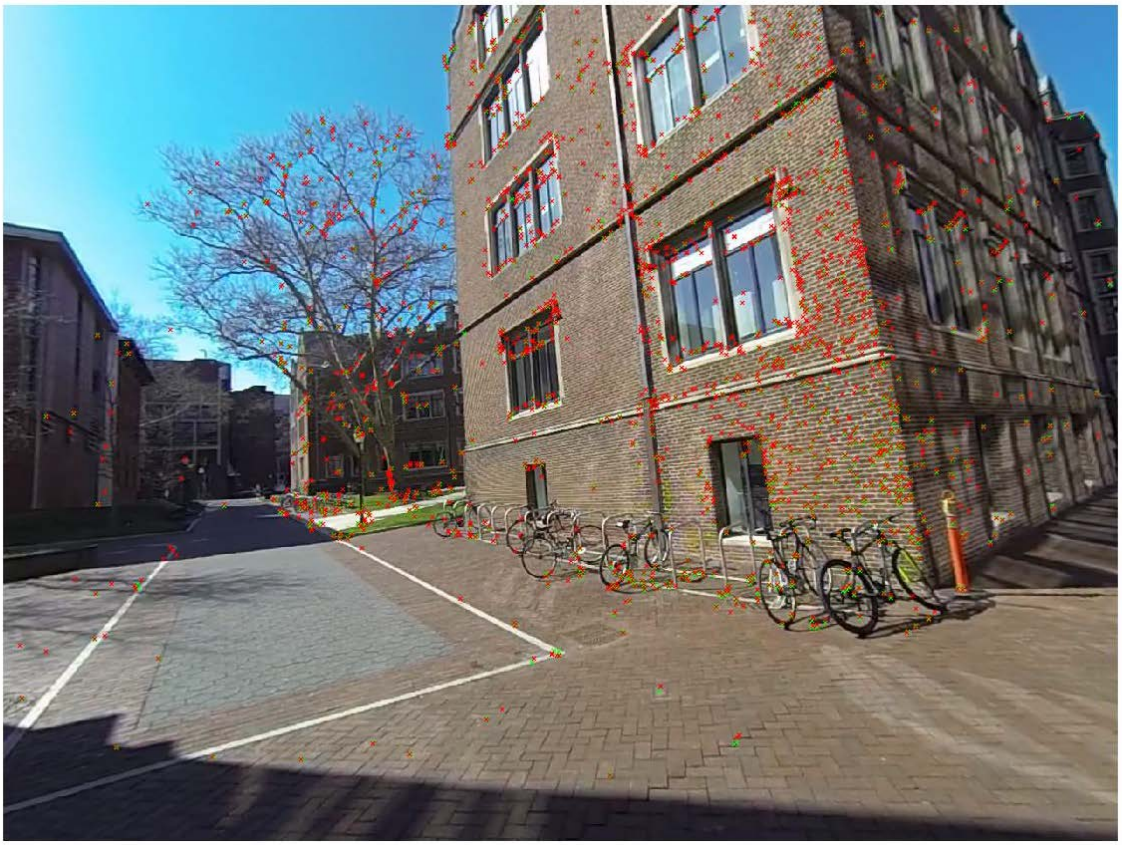
4. Bundle Adjustment with Jacobian

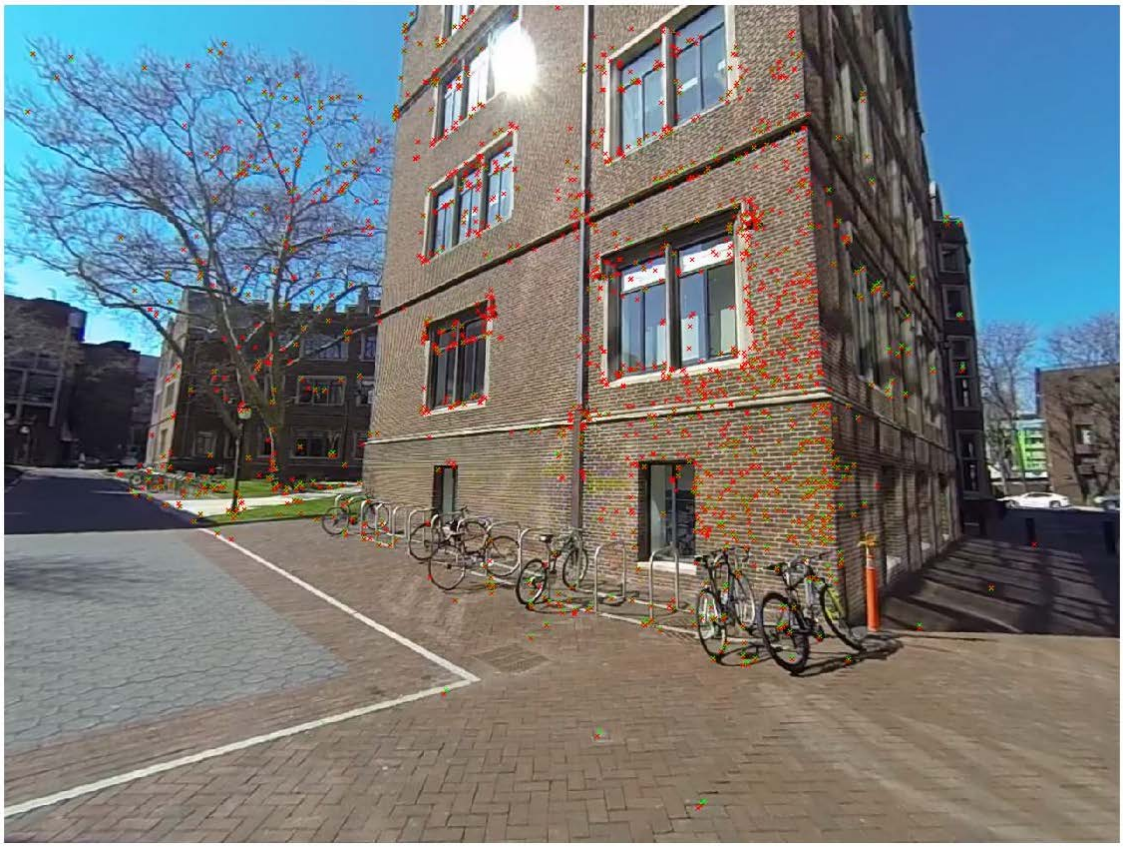
4755 reconstructed points and 6 cameras

Before Reprojection error: 0.740724873398653

(in 2D image, real points in green, reprojected points in red)

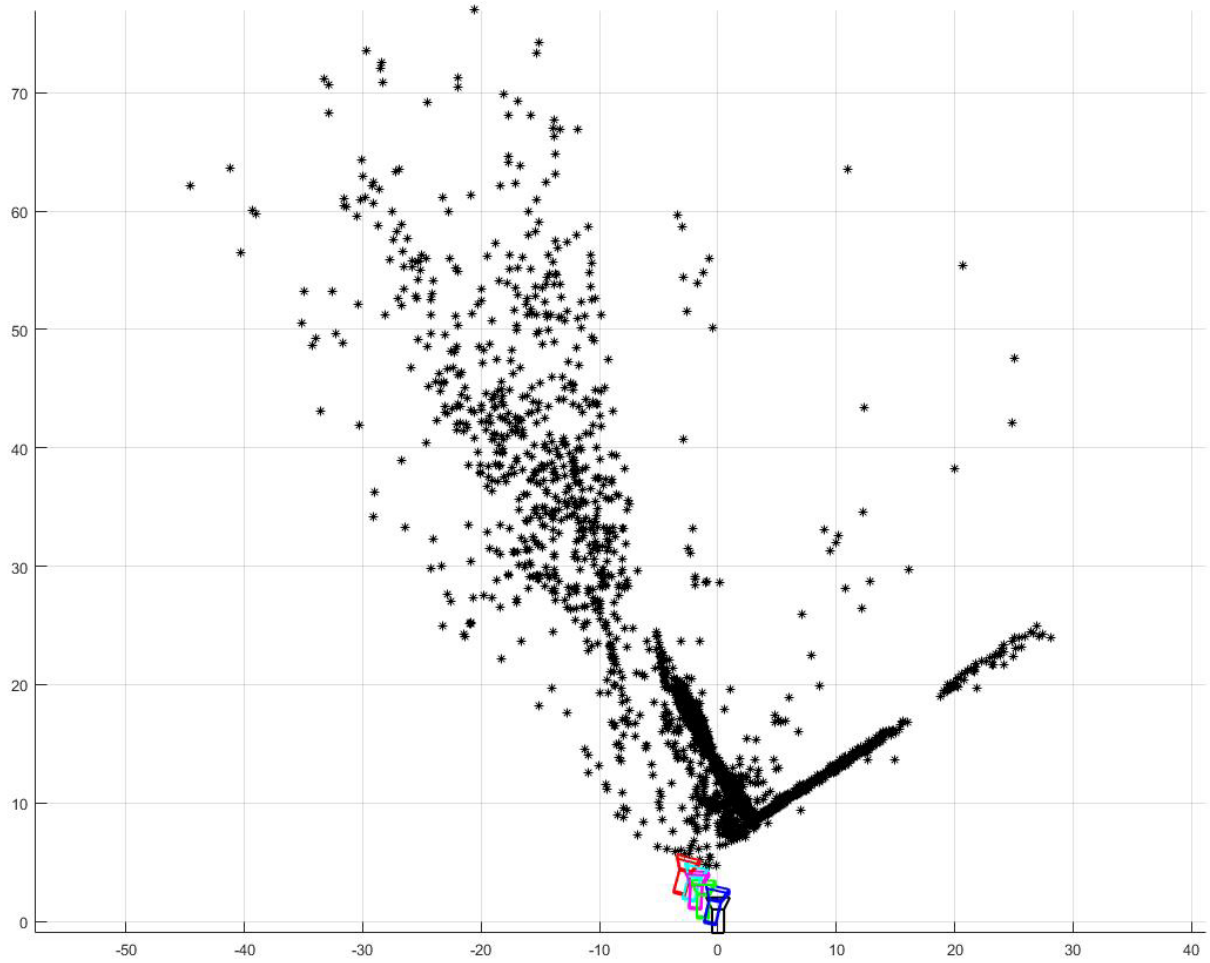


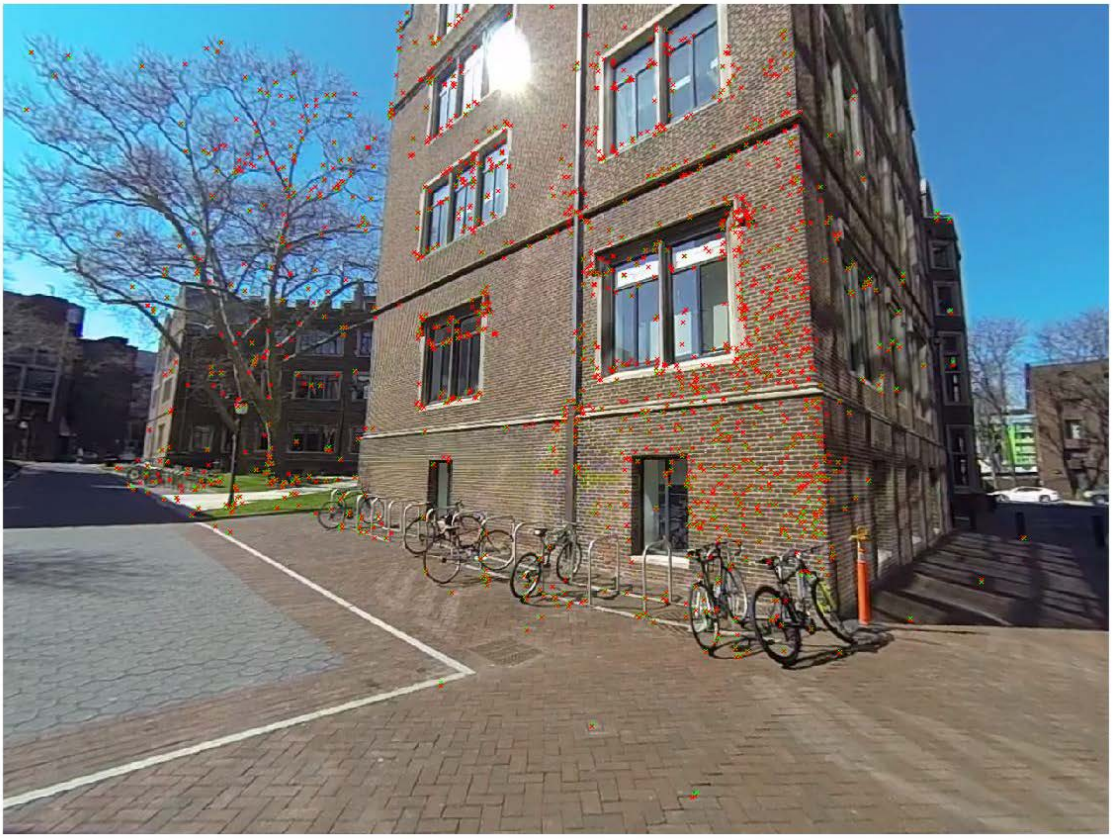




After Reprojection error: 0.566114001760971

(in 2D image, real points in green, reprojected points in red)





Appendix

Jacobian in Bundle Adjustment

```
function [error,jacobian] = BundleError(K,x,y,V,I,J,params)
% output error and jacobian matrix for bundle adjustment
% input x,y matching 2D point in image
% input V Visibility Matrix J*I
% input I num of images
% input J num of matching points
% input params (7*I+3*j)*1 cameras parameters and points for adjustment

[Cset, Rset, X] = decodeParams(params, I, J);
error = zeros(I*J*2,1);
jacobian = zeros(I*J*2,7*I+3*J);
for i = 1:I
    P = K*Rset{i}*[eye(3),-Cset{i}];
    for j = 1:J
        Xt = X(j,:)';
        error(((i-1)*J+j-1)*2+1:((i-1)*J+j-1)*2+2) = [V(j,i)*(x(j,i) -
(P(1,:)*[Xt;1])/(P(3,:)*[Xt;1]));...
                                                      V(j,i)*(y(j,i) -
(P(2,:)*[Xt;1])/(P(3,:)*[Xt;1]))];

        if nargout > 1
            Rt = Rset{i};
            Ct = Cset{i};
            qt = R2q(Rt);
            u = P(1,:)*[Xt;1];
            v = P(2,:)*[Xt;1];
            w = P(3,:)*[Xt;1];

            ujacobianC = -[K(1,1)*Rt(1,1)+K(1,3)*Rt(3,1) ,
K(1,1)*Rt(1,2)+K(1,3)*Rt(3,2) , K(1,1)*Rt(1,3)+K(1,3)*Rt(3,3)];
            vjacobianC = -[K(2,2)*Rt(2,1)+K(2,3)*Rt(3,1) ,
K(2,2)*Rt(2,2)+K(2,3)*Rt(3,2) , K(2,2)*Rt(2,3)+K(2,3)*Rt(3,3)];
            wjacobianC = -Rt(3,:);
            fjacobianC = [V(j,i)*(w*ujacobianC-u*wjacobianC)/(w^2) ;
V(j,i)*(w*vjacobianC-v*wjacobianC)/(w^2)];

            ujacobianX = [K(1,1)*Rt(1,1)+K(1,3)*Rt(3,1) ,
K(1,1)*Rt(1,2)+K(1,3)*Rt(3,2) , K(1,1)*Rt(1,3)+K(1,3)*Rt(3,3)];
            vjacobianX = [K(2,2)*Rt(2,1)+K(2,3)*Rt(3,1) ,
K(2,2)*Rt(2,2)+K(2,3)*Rt(3,2) , K(2,2)*Rt(2,3)+K(2,3)*Rt(3,3)];
            wjacobianX = Rt(3,:);
            fjacobianX = [V(j,i)*(w*ujacobianX-u*wjacobianX)/(w^2) ;
V(j,i)*(w*vjacobianX-v*wjacobianX)/(w^2)];

            ujacobianR = [K(1,1)*(Xt-Ct)' zeros(1,3) K(1,3)*(Xt-Ct)'];
            vjacobianR = [zeros(1,3) K(2,2)*(Xt-Ct)'  K(2,3)*(Xt-Ct)'];
            wjacobianR = [zeros(1,3) zeros(1,3) (Xt-Ct)'];
            fjacobianR = [V(j,i)*(w*ujacobianR-u*wjacobianR)/(w^2) ;
V(j,i)*(w*vjacobianR-v*wjacobianR)/(w^2)];
            Rjacobianq = [ 0 0 -4*qt(3) -4*qt(4);...
                          -2*qt(4) 2*qt(3) 2*qt(2) -2*qt(1) ;...
                           2*qt(3) 2*qt(4) 2*qt(1) 2*qt(2) ;...
```

```
                            2*qt(4) 2*qt(3) 2*qt(2) 2*qt(1) ;...
                            0 -4*qt(2) 0 -4*qt(4) ;...
                           -2*qt(2) -2*qt(1) 2*qt(4) 2*qt(3);...
                           -2*qt(3) 2*qt(4) -2*qt(1) 2*qt(2) ;...
                            2*qt(2) 2*qt(1) 2*qt(4) 2*qt(3) ;...
                            0 -4*qt(2) -4*qt(3) 0];
            fjacobianq = fjacobianR * Rjacobianq;

% negative jacobian since we are minimizing b-f
            jacobian(((i-1)*J+j-1)*2+1:((i-1)*J+j-1)*2+2,(i-1)*7+1:(i-
1)*7+7)= -[fjacobianq , fjacobianC];
            jacobian(((i-1)*J+j-1)*2+1:((i-1)*J+j-1)*2+2,7*I+(j-
1)*3+1:7*I+(j-1)*3+3)= -fjacobianX;

        end
    end
end
jacobian = sparse(jacobian);
end
```