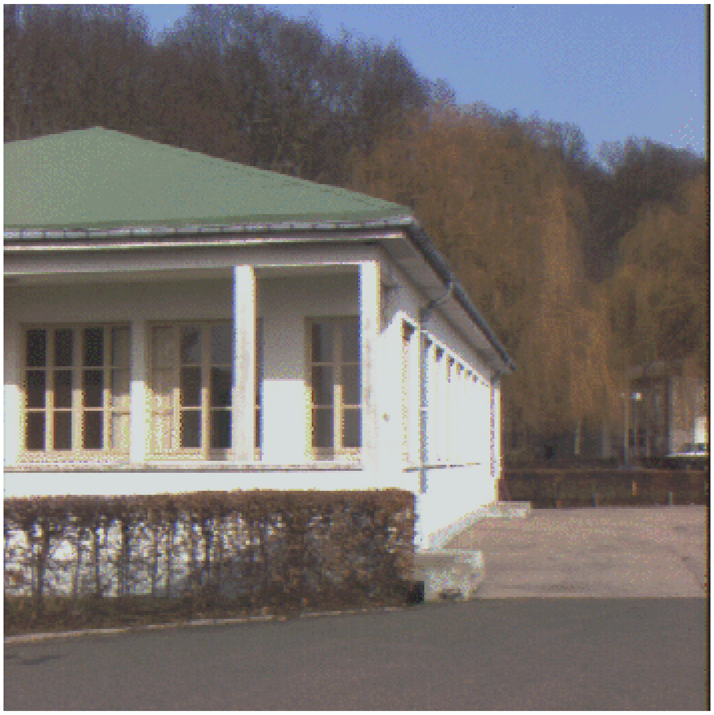
Eight Point Algorithm on 2 Views

The implemenation gives us

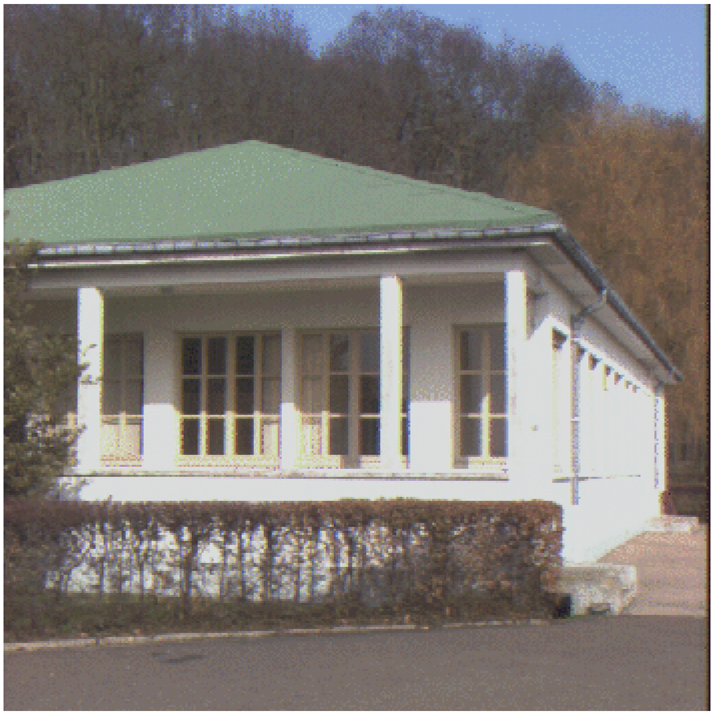
1. **Rotation Matrix (how much camera rotated)**
2. **Translation Matrix (how much camera translated)**
3. **3D points**

```
I1=imread('batinria0.tif');
imshow(I1)
```



```
subplot()
```

```
I2=imread('batinria1.tif');
imshow(I2)
```

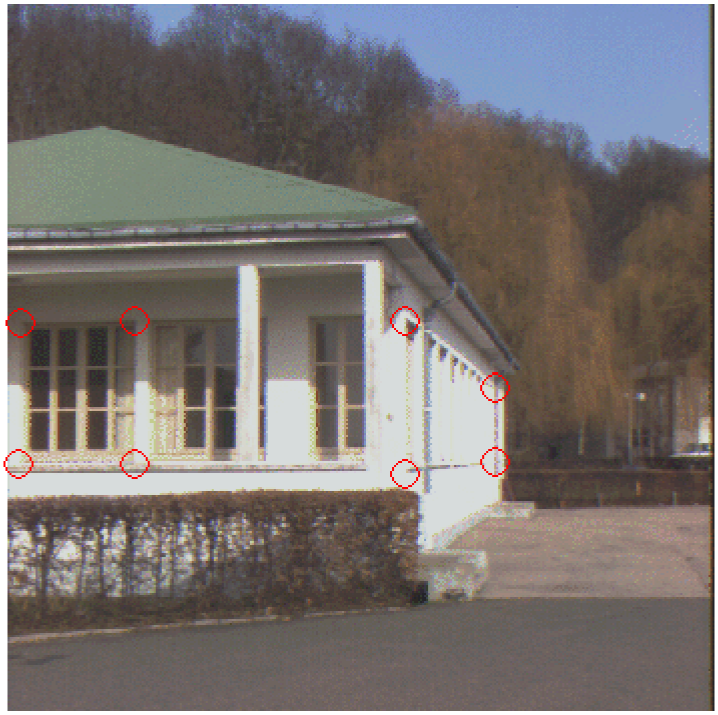


```
x1 = [
   10.0000
   92.0000
    8.0000
   92.0000
  289.0000
  354.0000
  289.0000
  353.0000
  ];
y1 = [
  232.0000
  230.0000
  334.0000
  333.0000
  230.0000
  278.0000
  340.0000
  332.0000
  ];
x2 = [
  123.0000
  203.0000
  123.0000
  202.0000
  397.0000
```

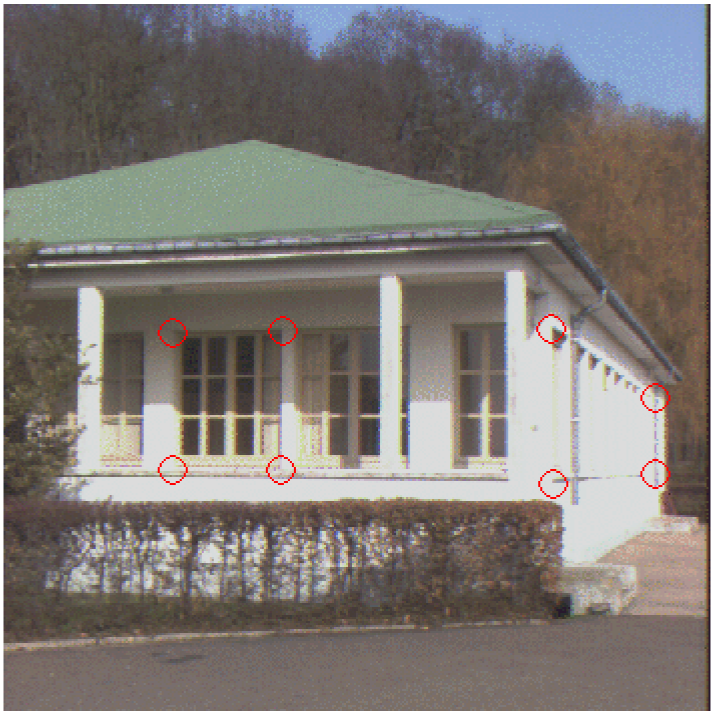
```
    472.0000
    398.0000
    472.0000
    ];
y2 = [
    239.0000
    237.0000
    338.0000
    338.0000
    236.0000
    286.0000
    348.0000
    341.0000
    ];
```

1. Mark Corressponding Points

```
imshow(I1)
hold on
plot(x1,y1,'ro','Markersize',10)
```



```
figure()
imshow(I2)
hold on
plot(x2,y2,'ro','Markersize',10)
```



2. Convert from Pixel Coordinates to Image Coordinates

```
Pixcord_1=[x1.';y1.';ones(1,8)]
```

```
Pixcord_1 = 3×8
    10    92     8    92   289   354   289   353
   232   230   334   333   230   278   340   332
     1     1     1     1     1     1     1     1
```

```
Pixcord_2=[x2.';y2.';ones(1,8)]
```

```
Pixcord_2 = 3×8
   123   203   123   202   397   472   398   472
   239   237   338   338   236   286   348   341
     1     1     1     1     1     1     1     1
```

```
K1=[844.310547 0 243.413315;
    0 1202.508301 281.529236;
    0 0 1]
```

```
K1 = 3×3
10^3 ×
    0.8443         0    0.2434
         0    1.2025    0.2815
         0         0    0.0010
```

```
K2=[852.721008 0 252.021805;
    0 1215.657349 288.587189;
    0 0 1]
```

```
K2 = 3×3
10^3  ×
    0.8527         0    0.2520
         0    1.2157    0.2886
         0         0    0.0010
```

```
Imcord_1=K1\Pixcord_1
```

```
Imcord_1 = 3×8
   -0.2765   -0.1793   -0.2788   -0.1793    0.0540    0.1310    0.0540    0.1298
   -0.0412   -0.0429    0.0436    0.0428   -0.0429   -0.0029    0.0486    0.0420
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

```
Imcord_2=K2\Pixcord_2
```

```
Imcord_2 = 3×8
   -0.1513   -0.0575   -0.1513   -0.0587    0.1700    0.2580    0.1712    0.2580
   -0.0408   -0.0424    0.0406    0.0406   -0.0433   -0.0021    0.0489    0.0431
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

3. Calucate chi Matrix (Kronecker Product)

```
a = zeros(9,8);
for i = 1:1:8
    chi(:,i)=kron(Imcord_1(:,i),Imcord_2(:,i));
end
```

4. Compute SVD of chi, Take 9th column of Vx Unstack to E Matrix (3X3)

```
[Ux Sx Vx]=svd(chi.');
E=reshape(Vx(:,9),3,3)
```

```
E = 3×3
   -0.0039   -0.5630    0.0265
    0.6168    0.0524   -0.3527
   -0.0244    0.4172   -0.0033
```

5. Compute SVD of E

```
[U S V]=svd(E);
```

6. Project E to Essential Space

```
S_new=[1 0 0; 0 1 0 ; 0 0 0];
Essential_Matrix=U*S_new*V.'
```

```
Essential_Matrix = 3×3
    0.0210   -0.8016    0.0220
    0.8676    0.0359   -0.4946
   -0.0546    0.5956    0.0058
```

7. Calculate Rotation and Translation Matrices

```
Rz=[0 -1 0; 1 0 0; 0 0 1];
T1_hat = U*Rz*S_new*U.';
R1 = U*Rz.'*V.';

T2_hat = U*Rz.'*S_new*U.';
R2 = U*Rz*V.';

T1 = [ -T1_hat(2,3); T1_hat(1,3); -T1_hat(1,2) ];
T2 = [ -T2_hat(2,3); T2_hat(1,3); -T2_hat(1,2) ];
```

8. Structure Reconstruction to find 3D location

```
M1 = zeros(3*8,8+1);
for i = 1:1:8
    M1(3*i-2:3*i,i)= hat(Imcord_2(:,i)).'*R1*Imcord_1(:,i);
    M1(3*i-2:3*i,9)=hat(Imcord_2(:,i))*T1;

end

M2 = zeros(3*8,8+1);
for i = 1:1:8
    M2(3*i-2:3*i,i)= hat(Imcord_2(:,i)).'*R2*Imcord_1(:,i);
    M2(3*i-2:3*i,9)=hat(Imcord_2(:,i))*T2;

end
```

9. Plot 3D location 1

```
M1
```

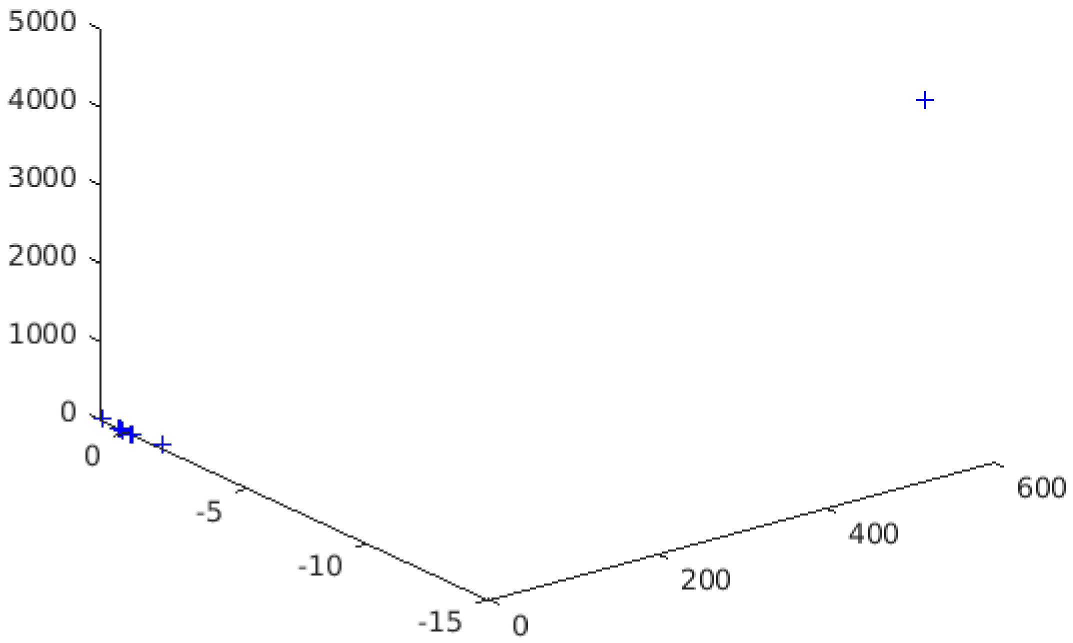
```
M1 = 24×9
    0.0265         0         0         0         0         0         0         0 ···
   -0.0021         0         0         0         0         0         0         0
    0.0039         0         0         0         0         0         0         0
         0    0.0229         0         0         0         0         0         0
         0   -0.0013         0         0         0         0         0         0
         0    0.0013         0         0         0         0         0         0
         0         0    0.0278         0         0         0         0         0
         0         0   -0.0020         0         0         0         0         0
         0         0    0.0043         0         0         0         0         0
         0         0         0    0.0243         0         0         0         0
      ⋮
```

```
[EIGVEV,DIAGMAT]=eig(M1.'*M1)
```

```
EIGVEV = 9×9
   -0.0013   -0.0004    0.0135   -0.0103    0.0157   -0.0365    0.9987    0.0255 ···
   -0.0015   -0.0004    0.0180   -0.0150    0.9989    0.0364   -0.0150    0.0052
   -0.0006   -0.0002    0.0059   -0.0044    0.0058   -0.0112    0.0249   -0.9996
   -0.0016   -0.0005    0.0182   -0.0144    0.0352   -0.9983   -0.0377    0.0106
   -0.0075   -0.0024    0.9899    0.1390   -0.0165    0.0159   -0.0113    0.0047
   -1.0000    0.0016   -0.0082    0.0040   -0.0013    0.0014   -0.0011    0.0005
   -0.0050   -0.0016    0.1383   -0.9900   -0.0182    0.0166   -0.0113    0.0047
   -0.0016   -1.0000   -0.0027    0.0012   -0.0004    0.0004   -0.0003    0.0001
   -0.0002   -0.0001    0.0019   -0.0013    0.0009   -0.0012    0.0012   -0.0006
DIAGMAT = 9×9
    0.0002         0         0         0         0         0         0         0···
         0    0.0002         0         0         0         0         0         0
         0         0    0.0003         0         0         0         0         0
         0         0         0    0.0003         0         0         0         0
         0         0         0         0    0.0005         0         0         0
         0         0         0         0         0    0.0006         0         0
         0         0         0         0         0         0    0.0007         0
         0         0         0         0         0         0         0    0.0008
         0         0         0         0         0         0         0         0
```

```
lambda1 = EIGVEV(1:8, 1);
gamma  = EIGVEV(8 + 1, 1);
if gamma < 0
    gamma = -gamma;
    lambda1 = -lambda1;
end
lambda1 = lambda1 / gamma;
X1 = Imcord_1 .* repmat(lambda1', 3, 1);
X2 = R1 * X1 + repmat(T1, 1, 8);

figure
plot3(X1(1,:), X1(2,:), X1(3,:), 'b+')
```



10. Plot 3D location 2

```
M2
```

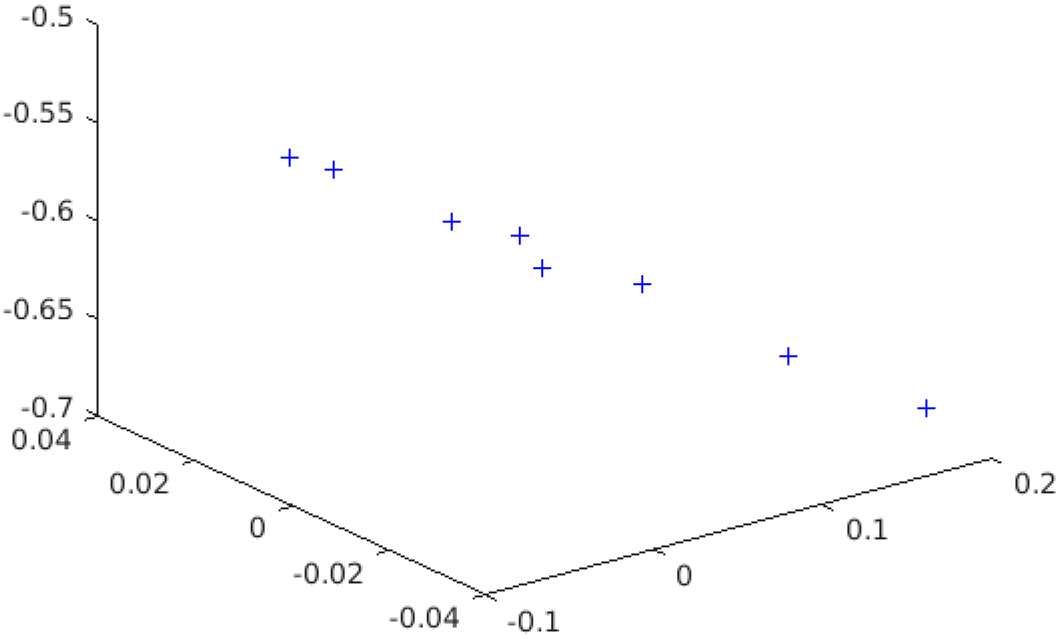
```
M2 = 24×9
    0.0738         0         0         0         0         0         0         0 ⋯
   -1.0478         0         0         0         0         0         0         0
   -0.0316         0         0         0         0         0         0         0
         0    0.0860         0         0         0         0         0         0
         0   -1.0003         0         0         0         0         0         0
         0   -0.0375         0         0         0         0         0         0
         0         0   -0.0229         0         0         0         0         0
         0         0   -1.0507         0         0         0         0         0
         0         0    0.0392         0         0         0         0         0
         0         0         0   -0.0190         0         0         0         0
     ⋮
```

```
[EIGVEV2,DIAGMAT2]=eig(M2.'*M2)
```

```
EIGVEV2 = 9×9
    0.3443   -0.0028    0.1084   -0.0221    0.2412   -0.0129   -0.5012   -0.7001 ⋯
    0.3230   -0.0029    0.1129   -0.0244    0.2830   -0.6935    0.5233   -0.0053
    0.3427   -0.0028    0.1077   -0.0220    0.2392   -0.0126   -0.4840    0.7140
    0.3222   -0.0029    0.1131   -0.0245    0.2847    0.7202    0.4862   -0.0051
    0.2836   -0.0054    0.2580   -0.6610   -0.6337    0.0020    0.0361   -0.0007
    0.2691   -0.6912   -0.6483    0.0254   -0.1484    0.0010    0.0191   -0.0004
    0.2822   -0.0058    0.2856    0.7480   -0.5155    0.0018    0.0339   -0.0007
    0.2685    0.7226   -0.6139    0.0250   -0.1464    0.0010    0.0189   -0.0004
   -0.5013    0.0021   -0.0766    0.0125   -0.1123    0.0016    0.0345   -0.0008
DIAGMAT2 = 9×9
    0.0004         0         0         0         0         0         0         0 ⋯
         0    0.5336         0         0         0         0         0         0
         0         0    0.5683         0         0         0         0         0
         0         0         0    0.6756         0         0         0         0
         0         0         0         0    0.7514         0         0         0
         0         0         0         0         0    1.0078         0         0
         0         0         0         0         0         0    1.0522         0
         0         0         0         0         0         0         0    1.1053
         0         0         0         0         0         0         0         0
```

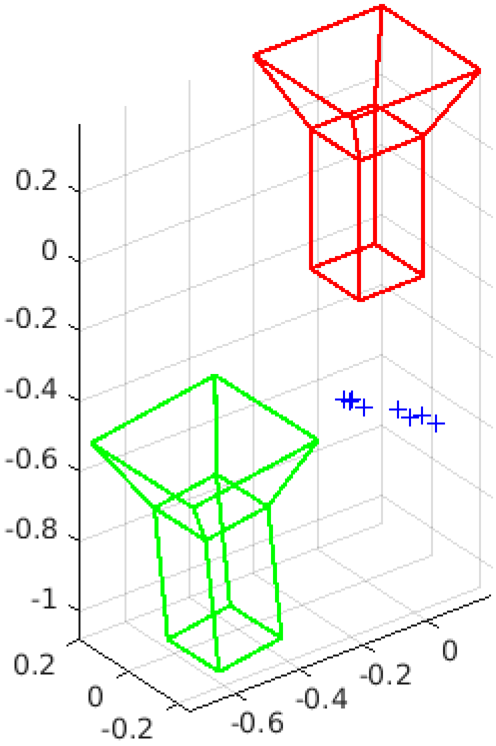
```
lambda2 = EIGVEV2(1:8, 1);
gamma2  = EIGVEV2(8 + 1, 1);
if gamma2 < 0
    gamma2 = -gamma2;
    lambda2 = -lambda2;
end
lambda2 = lambda2 / gamma2;
X12 = Imcord_1 .* repmat(lambda2', 3, 1);
X22 = R2 * X12 + repmat(T2, 1, 8);
```

```
figure
plot3(X1(1,:), X1(2,:), X1(3,:), 'b+')
```

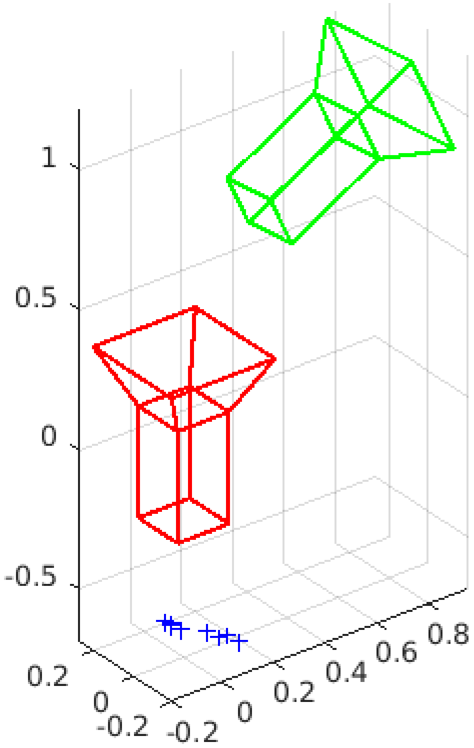


11. Plot camera to Visualise the seen

```
figure
plot3(X1(1,:), X1(2,:), X1(3,:), 'b+')
hold on
plotCamera('Location',[0 0 0],'Orientation',eye(3),'Opacity',0, 'Size', 0.2, 'Color', [
plotCamera('Location', -R1'*T1,'Orientation',R1,'Opacity',0, 'Size', 0.2, 'Color', [0 1
axis equal
grid on
```



```
figure
plot3(X12(1,:), X12(2,:), X12(3,:), 'b+')
hold on
plotCamera('Location',[0 0 0],'Orientation',eye(3),'Opacity',0, 'Size', 0.2, 'Color', [
plotCamera('Location', -R2'*T2,'Orientation',R2,'Opacity',0, 'Size', 0.2, 'Color', [0 1
axis equal
grid on
```



"" Helper Function ""

```
function hatmat = hat(vec)
    hatmat = [0 -vec(3) vec(2); vec(3) 0 -vec(1); -vec(2) vec(1) 0];
end
```