HW 3 Solutions

Problem 1 solution:

A) Recover the three dominant vanishing points.

Vanishing points in pixel coordinates (with (1,1) top left of image)

vp1=[1.0946e+03 -2.0780e+04 1]

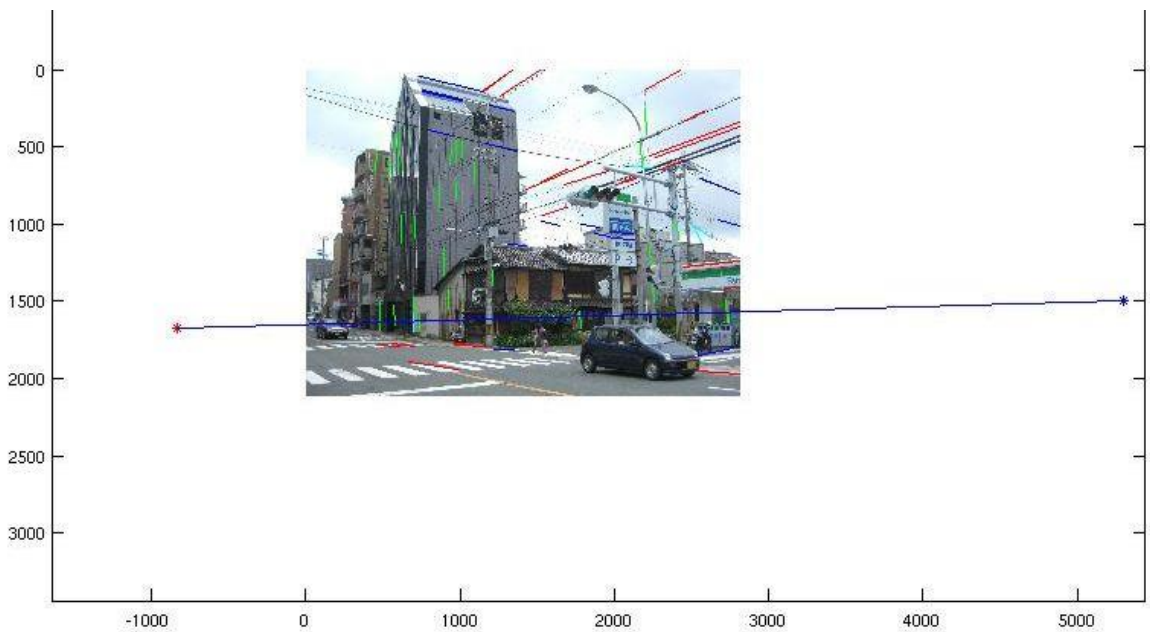
vp2=[5.3031e+03 1.4922e+03 1]

vp3=[-836.7876 1.6697e+03 1]

Horizon is a line passing through vanishing points of lines parallel to the ground plane, which is vp2 and vp3 in this case.

horizon=vp2 x vp3

horizon =[-177.5794 -6.1398e+03 1.0103e+07]



B) Recover f, u0, v0 from 3 orthogonal vanishing points.

A solution is given by equations 4 and 22 in Caprile and Torre, "Using Vanishing Points for Camera Calibration", IJCV 1990.

Using orthogonality

vp1' inv(K)’ inv(K)vp2=0

vp2' inv(K)’ inv(K)vp3=0

vp3' inv(K)’ inv(K)vp1=0

```
x1 = vp1(1); x2 = vp2(1); x3 = vp3(1);
y1 = vp1(2); y2 = vp2(2); y3 = vp3(2);

u0 = (y1*y2^2 - y1^2*y2 - y1*y3^2 + y1^2*y3 + y2*y3^2 - y2^2*y3 - x1*x2*y1 + ...
   x1*x2*y2 + x1*x3*y1 - x1*x3*y3 - x2*x3*y2 + x2*x3*y3)/...
   (x1*y2 - x2*y1 - x1*y3 + x3*y1 + x2*y3 - x3*y2);
v0 = (x2*(x1 - x3) - u0*(x1 - x3) + y2*(y1 - y3))/(y1 - y3);
f = sqrt(-(vp1(1)-u0)*(vp2(1)-u0) - (vp1(2)-v0)*(vp2(2)-v0));
```

f=3003.7

u0=1729.9 v0=1184.8

35 mm equi focal length =36*f/w=38.39 mm

C) Recover the rotation matrix R

Each vanishing point can be used to recover a column of **R**.

$$w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_i$$

where (uw, vw, w) is a vanishing point in homogeneous coordinates. To solve for w, you can use the property of rotation matrices that each column has a unit vector length. You can check that you have a proper rotation matrix by if R'*R gives you the identity matrix.

Using this approach, and the vanishing points and intrinsic parameters estimated above,

$$\mathbf{R} = \begin{bmatrix} 0.764 & -0.029 & -0.645 \\ 0.066 & -0.990 & 0.122 \\ 0.642 & 0.136 & 0.755 \end{bmatrix}$$

D) Estimating height from vanishing points.

Sign: ~1.65m (135 px tall)

Building: ~16.5m (~1110px)

Camera: ~12m (bottom of sign to horizon ~= 1000px)

Tractor: ~2.2m (~212px)

2. Mission Possible?

a) Yes, because the camera cannot translate, you need to project the view seen from the camera's position onto the screen. This requires estimating the camera's 3D position and the 3D geometry and texture of the scene behind the screen. The person monitoring the camera is limited to the viewpoint of the camera itself. You don't need to know camera rotation, just the points on the screen and scene that each ray of the camera passes through.

b) Yes, you would need to track the 3D position of the guard's eye. The same scene geometry and texture would also be needed as part (a).

c) No, because the projection can only simulate a single viewpoint at any given time. The view seen from the camera and the guard's eye would be different.

3. Epipolar Geometry

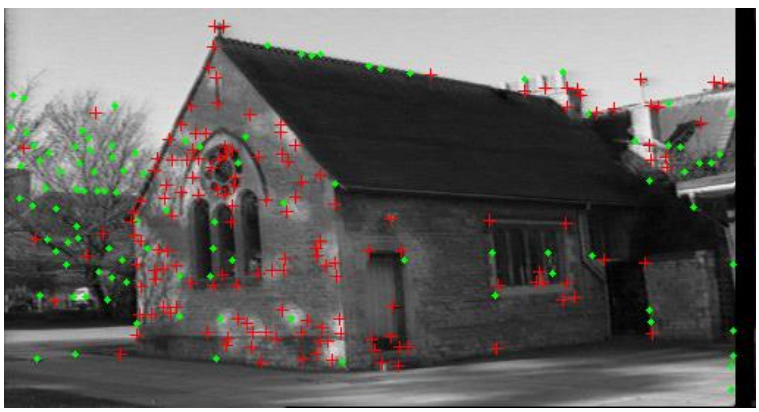
a) See attached code.

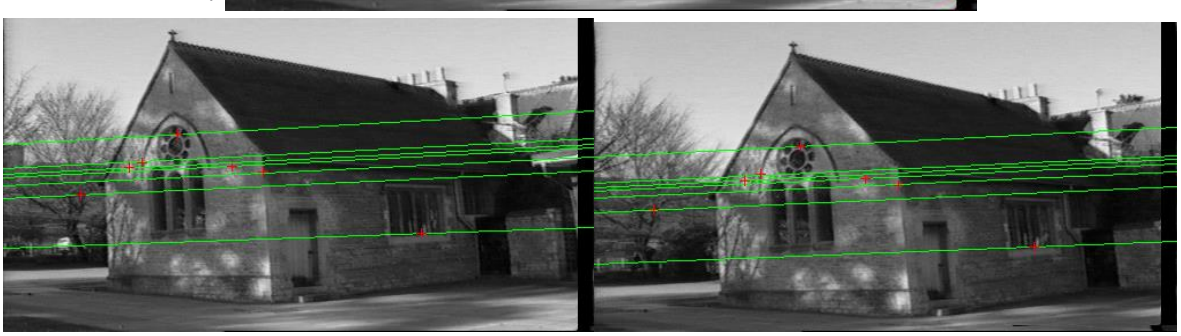
* The inlier/outlier test is as described in class (point within 1.5 pixels of corresponding epipolar line)

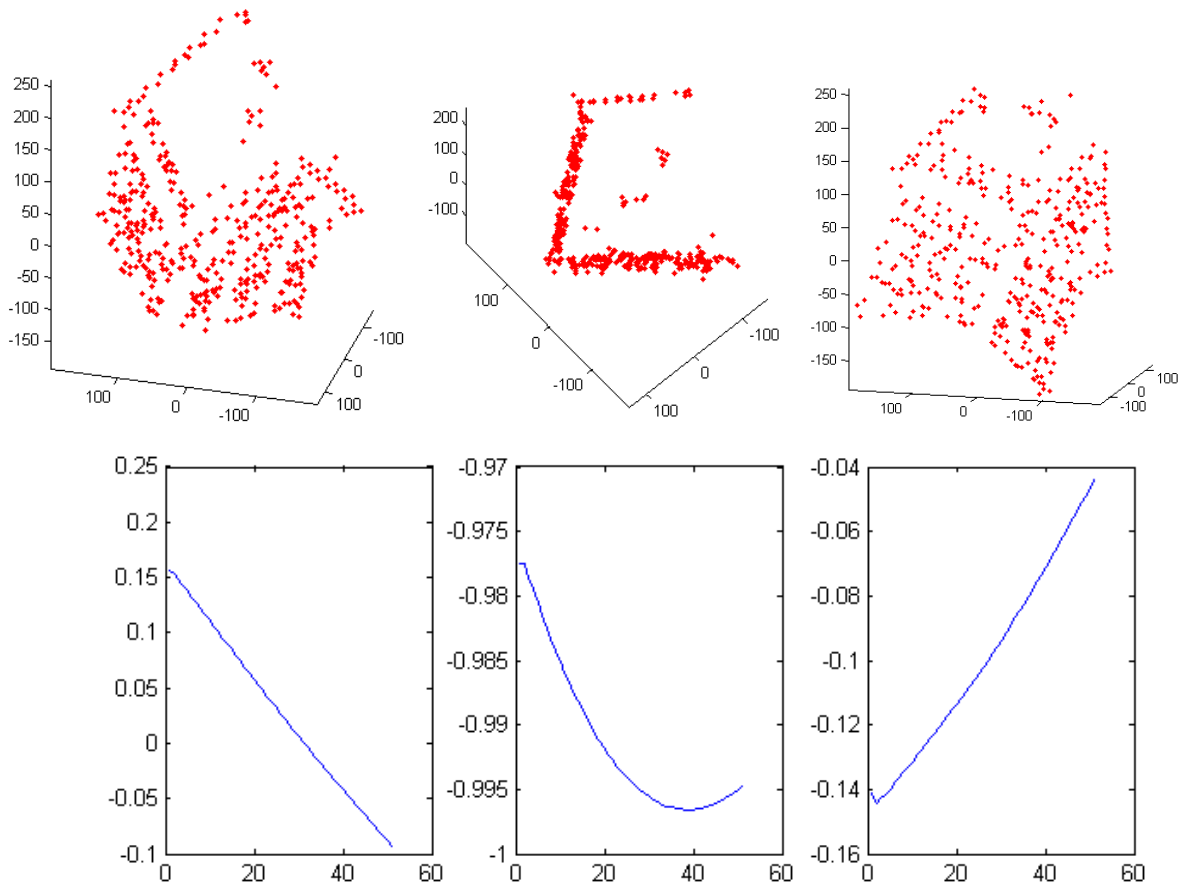
* See recovered F below:

$$F = \begin{bmatrix} -0.0000 & -0.0000 & 0.0121 \\ 0.0000 & 0.0000 & 0.1473 \\ -0.0096 & -0.1447 & 0.9783 \end{bmatrix}$$

*





4. Affine Structure from Motion

See the attached code for interest point detection, tracking, and SFM.