2.Single-View Geometry

Steps:

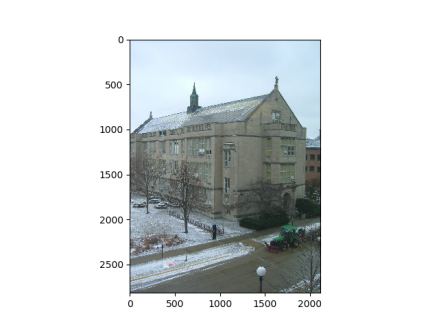
1.Read the image and pad the image.

image=ndimage.imread("img.jpg",flatten=False)

padding=5000

image\_padded\_list=[[[0,0,0] for j in range(5500+len(image[0]))] for i in range(len(image))]

img = mpimg.imread('img.jpg')



print(len(img),len(img[0]),img[0][0][0])

print(len(image\_padded\_list),len(image\_padded\_list[0]))

rows=len(img)

cols=len(img[0])

a=0

b=0

c=0

for i in range(rows):

b=0

for j in range(3500,cols+3500):

c=0

for k in range(3):

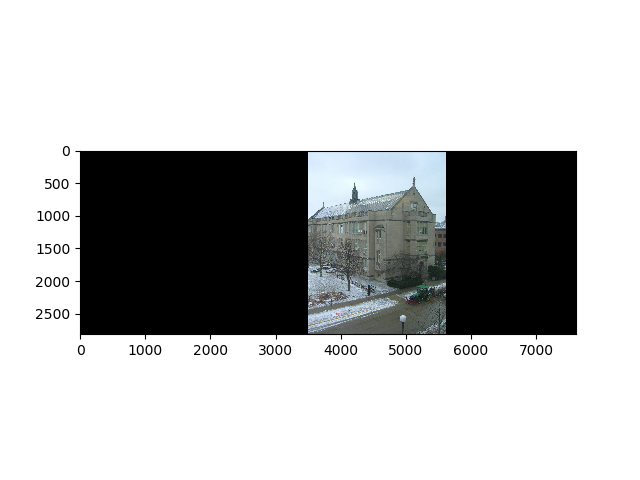
image\_padded\_list[i][j][k]=img[a][b][c]

c=c+1

b=b+1

a=a+1

image\_padded=np.array(image\_padded\_list)



2.Consider 8 points of 4 parellel lines manually.

points=[[3686,1048,1],[3890,1018,1],[3837,874,1],[3981,845,1],[3528,2438,1],[3762,2392,1],[3543,2528,1],[3747,2483,1]]

3.Find the lines using those selected points.Cross product of 2 points gives a line.

def lineofpoints(points):

lines=[]

j=0

for i in range(0,int(len(points)/2)):

line=np.cross(points[j],points[j+1])

j=j+2

lines.append(line/line[2])

return lines

4.Find the point of intersection of those 2 pairs of lines.These points of intersection are called **Vanishing points.**

def intersection\_ps(lines):

intersection\_p=[]

j=0

for i in range(0,int(len(lines)/2)):

ip=np.cross(lines[j],lines[j+1])

j=j+2

intersection\_p.append(ip/ip[2])

return intersection\_p

5.The line joining those 2 points is called **Vanishing line**.

vanishingline=np.cross(intersection\_p[0],intersection\_p[1])

vanishingline=vanishingline/vanishingline[2]

6.To find the tractor height,use the vanishing line we got is used here with known sign height.

def tractorheight(signline,signbase,br,vanishingline):

tractorbase=[5332,2309,1]

tractortop=[5332,2113,1]

tbsb=np.cross(signbase,tractorbase)

tbsb=tbsb/tbsb[2]

p\_vanline=np.cross(tbsb,vanishingline)

p\_vanline=p\_vanline/p\_vanline[2]

lineback=np.cross(p\_vanline,tractortop)

lineback=lineback/lineback[2]

t=np.cross(lineback,signline)

t=t/t[2]

bt=sqrt(((signbase[0]-t[0])\*\*2)+((signbase[1]-t[1])\*\*2))

tractorheight=round((bt/br)\*(1.65),2)

print("Tractor Height:")

print(tractorheight)

\*First draw a line with points of tractor base and sign base.Then take the intersection of the vanishing line and the sign base tractor base line.Then draw a line from that intersection point to tractor top.Point of intersection of the before line and the signline gives you the t point.The distance between t point and signbase is bt and the difference between signbase and signtop is br.Knowing the sign height,the height of the tractor is (bt/br)\*sign\_height.

7.To find the building height,use the vanishing line we got is used here with known sign height.

def buildingheight(signline,signbase,br,vanishingline):

buildingtop=[4438,921,1]

buildingbase=[4438,1986,1]

bbsb=np.cross(signbase,buildingbase)

bbsb=bbsb/bbsb[2]

p\_vanline=np.cross(bbsb,vanishingline)

p\_vanline=p\_vanline/p\_vanline[2]

lineback=np.cross(p\_vanline,buildingtop)

lineback=lineback/lineback[2]

t=np.cross(lineback,signline)

t=t/t[2]

bt=sqrt(((signbase[0]-t[0])\*\*2)+((signbase[1]-t[1])\*\*2))

buildingheight=round((bt/br)\*(1.65),2)

print("Building Height:")

print(buildingheight)

\*First draw a line with points of buiding base and sign base.Then take the intersection of the vanishing line and the sign base building base line.Then draw a line from that intersection point to building top.Point of intersection of the before line and the signline gives you the t point.The distance between t point and signbase is bt and the difference between signbase and signtop is br.Knowing the sign height,the height of the building is (bt/br)\*sign\_height.

8.Finding the camera height.

Take a point on vanishing line and and find the distance between signbase and signtop(stsb) and find the distance between signbase and vanishing point taken(sbpv2) and the camera height is (stsb/sbpv2)\*signheight.

def cameraheight(signbase,signtop):

p\_vanline2=[4547,1577,1]

stsb=round(sqrt(((signbase[0]-signtop[0])\*\*2)+((signbase[1]-signtop[1])\*\*2)),2)

sbpv2=round(sqrt(((signbase[0]-p\_vanline2[0])\*\*2)+((signbase[1]-p\_vanline2[1])\*\*2)),2)

cameraheight=round((sbpv2/stsb)\*1.65,2)

print("Camera Height:")

print(cameraheight)

9.Now plot all the parallel lines we took and the vanishing line.

