

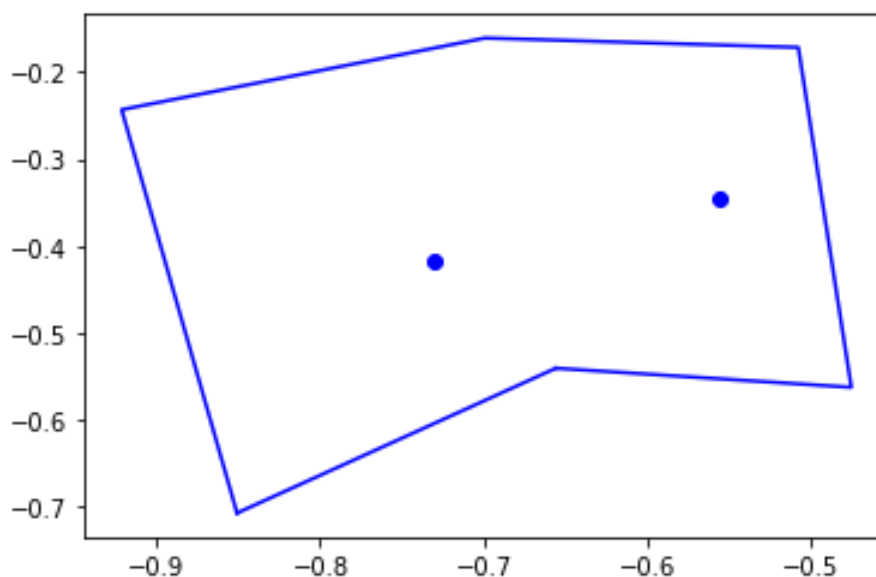
1. Praktika: Koordinatu barizentrikoak eta ehundura mapaketa

Praktika honen helburua espazioan dagoen bi zatidun bionbo bat sortzea da, zeinak Hikone pantailaren eskuineko aldean ehundura mapa bezala duen.

Hau egin ahal izateko, emandako lehenengo pausua bionboaren silueta definitzen duten puntuak kamera idealaren bidez lortzea izan da. Puntu oso garrantzitsuak dira, zeren horiek mugatzen duten poligonoaren gainean egin behar da ehundura mapaketa. Hurrengo kodea erabili dugu horretarako:

```
Vxy = km.kamera_ikuspegia(-pi/20, pi/20, [4, 3.5, 4], 1, 1, 0.1, 1)
# Marraztu V0, V1, ..., V5 poligonoa:
plt.plot(Vxy[0,0:6], Vxy[1,0:6], 'b-')
# Itxi poligonoa:
plt.plot([Vxy[0,0], Vxy[0,5]], [Vxy[1,0], Vxy[1,5]], 'b-')
# Bi zentroak ere:
plt.plot(Vxy[0,6:8], Vxy[1,6:8], 'bo')
# Ikus marrazkia. Parametro hauek aldatu pixka bat: bi angeluak, posizioa; besteak ez ukitu.
plt.show()
```

Honek irudi hau sortzen du:



Kameraren idealaren bidez, XYZ planoan dauden puntuak X'Y' planoan ikusten ditugu, eta horrela X'Y' planoko triangelu sareta definitu dezakegu:

```
triangeluak_xy = [
    np.column_stack((Vxy[:,6], Vxy[:,0], Vxy[:,1])), np.column_stack((Vxy[:,6], Vxy[:,1], Vxy[:,2])),
    np.column_stack((Vxy[:,6], Vxy[:,2], Vxy[:,3])), np.column_stack((Vxy[:,6], Vxy[:,3], Vxy[:,0])),
    np.column_stack((Vxy[:,7], Vxy[:,0], Vxy[:,3])), np.column_stack((Vxy[:,7], Vxy[:,3], Vxy[:,4])),
    np.column_stack((Vxy[:,7], Vxy[:,4], Vxy[:,5])), np.column_stack((Vxy[:,7], Vxy[:,5], Vxy[:,0]))]
```

X'Y' planoan dugun poligonoa barruan hartuko duen koadroa finkatu behar da, koadroa zehazteko goiko ezkerreko erpinaren koordinatuak hartuko ditugu koadroa_xy funtzioaren bidez eta koadroaren aldea ere.

```
def koadroa_xy(i, j):
    → koad_0_y=-0.1
    → koad_0_x=-1.0
    → koad_l=0.7
    → zatitz=float(m_zabal-1)
    → return koad_0_x+koad_l*j/zatitz, koad_0_y+koad_l*i/zatitz
```

Ehundurarekin hasteko, X'Y'-n dagoen poligonoen koordinatuak UV planoko triangelu sare gista idatzi ditugu *np.array* bat erabiliz, eta geroago koordinatu barizentrikoak kalkulatzeko beharko ditugun matrizea eta bere alderantzizkoa definitu ditugu:

```
triangeluak_uv=[
np.column_stack(([0.5,1],[0,0],[1,0])),np.column_stack(([0.5,1],[1,0],[1,2])),
np.column_stack(([0.5,1],[1,2],[0,2])),np.column_stack(([0.5,1],[0,2],[0,0])),
np.column_stack([[-0.5,1],[0,0],[0,2])),np.column_stack([[-0.5,1],[0,2],[-1,2])),
np.column_stack([[-0.5,1],[-1,2],[-1,0])),np.column_stack([[-0.5,1],[-1,0],[0,0]])]
```

```
M=[]
Mald=[]
for triangelu in triangeluak_xy:
    → M=np.row_stack((triangelu,[1,1,1]))
    → Mald.append(np.linalg.inv(M))
```

Hau eginda, bi *for* erabiliko ditugu mapaketa irudiaren pixel guztietatik pasatzeko. Pixel bakoitzean, honi dagokion (x,y) puntuaren triangeluarekiko koordinatu bariozentrikoak kalkulatu ditugu. Puntu hori triangeluaren barruan egotekotan puntuari dagozkion mapako (u,v) koordinatuak interpolazio barizentrikoaren bidez lortzen ditugu, eta (u,v) koordinatu hauek lehen definitutako funtzioaren bidez, koordinatuetatik pixel matrizeko (i,j) indizeak lortzen ditugu. Azkenik indize hauek ematen diguten Hikone irudiko puntuko koloreaz, ehundura mapaketako indize hoiek dituen puntua koloreztatzen dugu. Kodea hau da:

```
for i in range(m_zabal):
    → for j in range(m_zabal):
    → → for elem in range(len(Mald)):
    → → → koord_bariz=Mald[elem].dot([koadroa_xy(i,j)[0], koadroa_xy(i,j)[1], 1])
    → → → if np.all(koord_bariz>=0):
    → → → → u,v=triangeluak_uv[elem].dot(koord_bariz)
    → → → → p=pixel_indizeak(u,v)
    → → → → mapaketa_irudia[i,j]=hikone_irudia[p[0],p[1]]
```

Programaren exekuzioa bukatzen denean, hurrengo emaitza pantailatzen da:

