

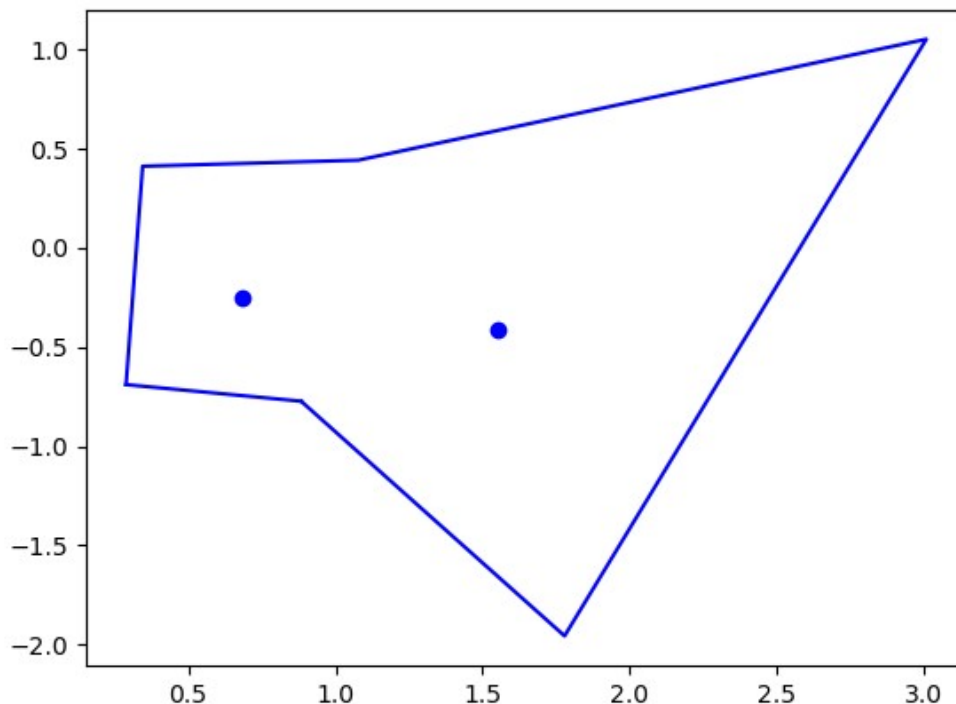
1.Praktika: Ehundura mapaketa

Lehengo praktikaren helburua, espazioan dagoen bi zatidun biobo bat sortzea da.

Hau egiteko, lehenengo pausua, bionboaren egitura definitzen duten puntuak lortzea da. Ehundura mapaketa puntu hauek mugatzen duten poligonoaren gainean egin behar da. Horretarako, hurrengo kodea dugu:

```
# Kameraren bidez XYZ espazioan dagoen bionboaren muturrak X'Y' planoko puntu bezala ikusiko
# ditugu, puntu horiek mugatzen duten poligonoaren gainean egin behar da ehundura mapaketa,
# bi atalen zentroak ere hartuko ditugu triangelu sarea definitzeko. Vxy matrizea lortuko
# dugu, honek bi lerro ditu puntuen x, y koordenatuekin eta bere zutabeak dira poligonoaren
# erpinak V0, V1, ... , V5 eta bi zatien zentroak V6, V7:
Vxy = km.kamera_ikuspegia(-pi/20, 5*pi/9, [1.75, 1.6, 1.25], 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()
```

Kode honek, hurrengo poligonoa sortzen du:



Kameraren idealaren bidez, XYZ planoan dauden puntuak X'Y' planoan ikusten ditugu. Hori dela eta, X'Y' planoko triangelu sarea definitu dezakegu, hurrengo kodea erabiliz:

```
# X'Y' planoko triangelu sarea
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. Koadro hau zehazteko, koadro honen alde eta haren goiko ezkerreko erpinaren koordinatuak artuko ditugu eta hurrengo funtzioa erabiliko dugu:

```
# Mapaketa irudiko matrizearen pixel baten indizeetatik X'Y' planoan dagokion puntuaren
# koordinatu kartesiarrek lortzeko funtzioa:
def koadroa_xy(i, j):
    koad_0_x = 0.25
    koad_0_y = 1.10
    koad_l = 3
    zatitz = float(m_zabal - 1)
    return koad_0_x + koad_l * j / zatitz, koad_0_y - koad_l * i / zatitz
```

Ehundura mapaketarekin hasteko, gure X'Y' planoko poligonoen koordinatuak, UV planoko triangelu sare bezala idatzi 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]))]
```

Ondoren, koordinatu barizentrikoak kalkulatzeko erabiliko digun, matrizea eta haren alderantzizkoa definitu ditugu:

```
Malderantzizkoak = []
for triangelua in triangeluak_xy:
    M = np.row_stack((triangelua,[1,1,1]))
    Malderantzizkoak.append(np.linalg.inv(M))
```

Hau egin ondoren, mapaketa irudiaren pixel guztiak iteratuko ditugu. Pixel bakoitzeko, hari dagokion (x, y) puntuaren triangeluarekiko koordinatu bariozentrikoak kalkulatu ditugu. Puntua triangeluaren barruan egotekotan, hari dagozkion mapako (u, v) koordinatuak interpolazio bariozentrikoaren bidez lortuko ditugu. Hau izanda eta lehen aipatutako funtzioa erabilita, pixel matrizeko (i, j) indizeak lortzen ditugu. Indize hauek ditugula, Hikone irudiko koloreak lortu ditzazkegu, eta horrekin, indize horiek ehundura mapaketan duten puntua koloreztatuko ditugu. Azkenin, irudia bistaratu baino ez dugu egin behar.

Honen kodea hurrengoa da:

```

for i in range(0,m_zabal):
    for j in range(0,m_zabal):
        for t in range(0,8):
            xy = koadroa_xy(i,j)
            koord_bariz = Malderantzizkoak[t].dot([xy[0], xy[1],1])
            if np.all(koord_bariz >= 0):
                koord_uv = triangeluak_uv[t].dot(koord_bariz)
                pixels = pixel_indizeak(koord_uv[0], koord_uv[1])
                kolorea = hikone_irudia[pixels[0], pixels[1]]
                mapaketa_irudia[i][j] = kolorea
plt.imshow(mapaketa_irudia)
plt.show()

```

Hau da, guztia exekutatu ostean lortzen dugun emaitza:

