

## **4. PRAKTIKA: Errendatzea (rendering)**

Praktika honetan, 3D eszena batean kokatutako piramide bat renderizatu nahi dugu 2D-n kamera idealaren artifizioarekin eta argi direkzional bat gehituz eszenari. Hau egiteko, piramidea moduluan dauden metodoak erabiliko ditugu.

Lehenik, piramidea sortzeko erabiltzen diren triangeluak (3 erpinez definituak) eta distiren adierazpenak (Lambert legea aplikatuz) lortu beharko dira. Piramidearen azpiko tapa ez denez kontuan hartuko, funtzioari 24 aurpegi kopurua eta argi-iturriaren bektorea pasatu behar zaio lista batean.

```
triangeluZerrenda=piramidea_sortu(N, [1, 0.5, 0])
```

Ondoren, erpinen koordenatuak kameraren ikuspegi bolumen paralelo estandarrean dauden triangeluak kalkulatu ditugu piramidea\_paraleloan funtzioaren bidez. Honi, kameraren informazioa pasatuko zaio.

```
triangeluaParalelo=piramidea_paraleloan(triangeluZerrenda[0], pi/20, pi/4, [2, 0.5, 2], 5, 5, 1, 5)
```

Bi funtzio horiek erabili ondoren, renderizazio-algoritmoari jarraituko diogu. Hau pixelei koloreak esleitzeko koordenatu barizentrikoak eta kolore horietarako beharrezko matrizeak kalkulua. Gainera, piramidea sortuko den koadroa definitu behar dugu. Kode osoa hau lehenengo praktikatik har dezakegu, eta honako hau litzateke:

```
def koadroa_xy(i, j):
    koad_0_x=-1
    koad_0_y=1
    koad_l=2
    zatitz = float(m_zabal - 1)
    return koad_0_x + koad_l * j / zatitz, koad_0_y - koad_l * i / zatitz
```

```
Mald = []
for tri in triangeluaParalelo:
    column=np.column_stack((tri,[0,0,-1]))
    row = np.row_stack((column,[1,1,1,0]))
    inv = np.linalg.inv(row)
    Mald.append(inv)
```

```
m_zabal = 1000
```

Orain, lehenengo praktikan egin genuen bezala, irudiaren pixel matrizea sortuko dugu. Horretarako hau egingo dugu:

```
piramide = np.ones((m_zabal, m_zabal, 1))
```

Azkenik, pixel guztiak (eta horien izpiak) eta kameratik hurbilen dagoena aztertuko ditugu, pixel horiekin sortzeko piramidea ikusiko dugu.

```
for i in range(m_zabal):
    for j in range(m_zabal):
        hurbilena=float('+inf')
        for tri in range(len(triangeluZerrenda[0])):
            koord_bariz = Mald[tri].dot([koadroa_xy(i,j)[0], koadroa_xy(i,j)[1], 0, 1])

            if (koord_bariz[0]>= 0 and koord_bariz[1]>=0 and koord_bariz[2]>=0 and
koord_bariz[3]<hurbilena ):
                hurbilena = -koord_bariz[3]
                piramide[i,j] = triangeluZerrenda[1][tri]
```

Gure piramidea behin sortuta, irudia ikusteko, hurrengo kodea exekutatu dugu.

```
plt.imshow(piramide)
```

