Errendatzea (rendering)

3D eszena batean kokatutako piramide bat renderizatu behar da 2D-n kamera idealaren artifizioarekin eta argi direkzional bat gehituz eszenari. Horretarako, *piramidea* moduluan bi funtzio ditugu. Hasteko, *piramidea_sortu* funtzioak piramidearen triangelu zerrenda emango digu non triangelu bakoitzaren erpinak matrize batean dauden zutabeka munduko koordenatuetan, gainera funtzio honek aurpegi bakoitzaren distira ere ematen digu, argi direkzional baterako Lambert-en legea aplikatuz. Piramidearen azpiko tapa ez dugu kontuan izango. Funtzioaren argumentuak dira aurpegi kopurua eta argi-iturriaren bektorea, 24 eta [1, 0.5, 0] adibidez. Bigarren funtzioa *piramidea_paraleloan* da eta honek kalkulatzen du triangelu zerrenda non erpinen koordenatuak kameraren *ikuspegi bolumen paralelo estandarrean* dauden. Funtzioaren argumentuak dira triangelu zerrenda (erpinak munduko koordenatuetan) eta kameraren zehaztapenak, adibidez: pi/20, pi/4, [2, 0.5, 2], 5, 5, 1, 5 (kamera orientatzeko bi angelu X-en eta Y-ren inguruan biratuz, ondoren posizioa eta gainerako parametroak).

Piramidearen 2D renderizazioa lortzeko *izpi-isurketaren* metodoa erabili behar da, teorian ikusi den bezala. Kalkulu guztiak **ikuspegi bolumen paralelo estandarrean** egin behar dira. XY planoan (kameraren ardatzak) koadro bat hartuko dugu eta gainean pixel matrize bat ezarriko diogu, hau izango da renderizatutako irudia pixel bakoitzaren distira edo kolorea esleituz.

Algoritmoa

Malderantzizkoak = []
FOR triangelu guztiak

> Kalkulatu eta erantsi zerrendari matrize honen alderantzizkoa
> (*np.column_stack*, *np.row_stack*, *np.linalg.inv*)
>
> $$\begin{vmatrix} A_x & B_x & C_x & 0 \\ A_y & B_y & C_y & 0 \\ A_z & B_z & C_z & -1 \\ 1 & 1 & 1 & 0 \end{vmatrix}$$
>
> gero erabiliko dugu matrize honen alderantzizkoa, iterazio bakoitzean ABC triangeluaren eta (x,y) izpiaren arteko ebakidura kalkulatzeko, ekuazio sistemaren ezezagunak dira ebakiduraren koordenatu barizentrikoak eta *z* koordenatua, alpha A + beta B + gamma C = P lehenengo hiru ekuazioak dira, z ezkerrera doa, alpha + beta + gamma = 1 laugarren ekuazioa da.

Sortu irudiaren pixel matrizea, *piramide_irudia* izenekoa adibidez, hasieran 0,5 balioa eman pixel guztiei (ingurunearen distira litzateke), zabalera 1000 hartu (edo 500 probak egiteko makal badoa). Pixel matrizearen indizeetatik koordenatu kartesiarrak lortzeko kopiatu *koadroa_xy(i, j)* funtzioa lehenengo praktikatik, orain koadroaren goiko ezkerreko erpinaren koordenatuak (-1,1) dira eta koadroaren aldea 2.

FOR pixel guztiak (eta dagozkien izpiak) zeharkatuz *i* eta *j* indizeekin

hurbilena aldagaian gordeko dugu izpi honek aztertutako triangeluekin dituen ebakiduren artean kameratik hurbilena den ebakiduraren distantzia, triangeluak aztertzen hasi aurretik *float('+inf')* balioa esleitu aldagaiari.

FOR triangelu guztiak

Kalkulatu non mozten duen triangelua pixelari dagokion *(x, y)* izpiak: goian kalkulatu den matrize alderantzizkoa biderkatu [*x*, *y*, 0, 1] bektorearekin, emaitza alpha, beta, gamma eta *z* dira.

Baldin alpha, beta, gamma >= 0 (ebakidura du izpiak triangeluarekin) eta kameratik ebakidurarako - *z* distantzia txikiagoa bada aurretik gordetako distantzia baino, orduan gorde distantzia berria *hurbilena* aldagaian eta triangelu horren distira esleitu *(i, j)* pixelari *piramide_irudia* matrizean. Kontuz, *z* negatiboa denez, distantzia - z da positiboa izateko.

Bestela, baldin alpha edo beta edo gamma < 0 (ez dago ebakidurarik) edo - *z* ez bada aurretik gordetako distantzia baino txikiagoa, ezer ez egin.

Amaitzeko, hartu pixel matrizea eta *plt.imshow* funtzioarekin sortu grafikoa. Koloreak erabiltzeko (adibidez ingurune urdina, piramidean gorri argiak/ilunak), matrizeko pixelak [*red*, *green*, *blue*] zerrendak izan behar dira, kolore bakoitza [0, 1] tartean (orain *p[:,:,2]*, *p[i,j,:]* erako esleipenak egin beharko dira).

Gerta daiteke piramidearen aurpegi bat izpiaren paraleloa izatea, orduan salbuespena (np.linalg.linalg.LinAlgError) dugu. Bi irtenbide: kamera pixka bat mugitu edo, aukera hobea, *try/exception* kodea sartu, aurpegiak ez du ebakidurarik eta aurrera jarraitu.