KbG - Azterketa - 2019ko urria

1. (1 puntu) (a) Kalkulatu ABC triangeluko P puntuaren α, β, γ koordenatu barizentrikoak t, s parametroetatik abiatuz (hartu AB ertzean Q etab). (b) Zein dira koordenatu barizentrikoen ezaugarriak? Zer da konbinazio konbexua? (c) Adierazi nola egiten den interpolazio barizentrikoa, triangeluko erpinetan funtzioaren balioak f_A, f_B, f_C ezagunak badira, eta triangelu barruko P puntuan funtzioaren balioa interpolatu nahi bada.

2. (1,5 puntu)

   a) Lau puntu emanik, eta ardatz kartesiarrak finkatuz, Bézier kurba definituko dugu $\gamma(t) = G \cdot M \cdot \mathbf{t}$ formulan $G = [P_0\, P_1\, P_2\, P_3]$ geometria matrizea eta M oinarri matrizea hartuz, $\mathbf{t} = [1\, t\, t^2\, t^3]^T$:

   $$M = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   Hasteko, kalkulatu $M \cdot \mathbf{t}$ eta idatzi $b_0(t), b_1(t), b_2(t), b_3(t)$ polinomioak. Kalkulatu polinomio bakoitzak hartzen dituen balioak t parametroa 0 denean eta 1 denean.

   b) Espazioan P_{ij} hamasei punturen sareta hartuz, Bézier adabakia da $S(u,v) = \sum_{i,j=0}^{3} P_{ij} b_i(u) b_j(v)$, non $u, v \in [0,1]$ parametroak diren. Frogatu adabakiaren $S(u,0), S(u,1), S(0,v), S(1,v)$ ertzak Bézier kurbak direla, saretaren ertzetako puntuen gainekoak.

3. (1 puntu) Izan bedi planoan transformazio proiektiboa, koordenatu kartesiarrak hartuz matrize hau duena (espazio proiektiboan):

   $$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

   Marraztu planoan $P(1,0)$ eta $Q(-4,3)$ puntuak, eta PQ zuzena (infinitua), ez idatzi zuzen parametrikoa. Kalkulatu P' eta Q' transformatuak, marraztu bi puntuak eta zuzen transformatua. Oro har, noiz da $R(x,y)$ puntuaren transformatua infinitua? Marraztu gutxi gorabehera

PQ zuzenaren gaineko R puntua eta zuzen transformatuan R'_∞ puntu infinitua alde batera eta bestera. Azkenik, PQ zuzeneko L_∞ puntu infinituaren transformatua, L' finitua, zuzen transformatuaren zein aldetan dago? (Puntuen ordena gordetzen da limitearen aurretik)

4. (2 puntu)

   a) Rodrigues-en formula egiaztatuko dugu. Dakigunez, $\mathbb{R}^3$ espazioan, $\mathbf{w}$ bektore unitarioa emanik, J_ω matrizearen bidez kalkula dezakegu biderkadura bektoriala, $J_\omega \mathbf{v} = \mathbf{w} \times \mathbf{v}$, gainera $J^2_\omega \mathbf{v} = \mathbf{w} \times (\mathbf{w} \times \mathbf{v})$. Bestalde, edozein bektore deskonposa dezakegu bi osagaitan, bat $\mathbf{w}$ bektoreari perpendikularra eta bestea $\mathbf{w}$-ri paraleloa, $\mathbf{v} = \mathbf{v}_\perp + \mathbf{v}_\parallel$. Gainera $\mathbf{w} \times (\mathbf{w} \times \mathbf{v}_\perp) = -\mathbf{v}_\perp$ dugu. Kalkulatu, θ angelua emanik, $(I + \sin\theta\, J_\omega + (1 - \cos\theta) J^2_{\omega)})\, \mathbf{v}$ bektore transformatua zein den eta interpretatu emaitza.

   b) Dakigunez biraketak adierazten ditugu koaternoien bidez, $\mathbf{q}(\theta, \mathbf{w}) = \cos(\theta/2)\, [1\ 0\ 0\ 0]^T + \sin(\theta/2)\, [0\ \omega_x\ \omega_y\ \omega_z]^T$. Marraztu $\mathbb{R}^4$ espazioaren eskema bat: U ardatza eta disko batean XYZ ardatzak. Marraztu $\mathbf{q}(0, \mathbf{w})$, $\mathbf{q}(\pi, \mathbf{w})$ eta $\mathbf{q}(\theta, \mathbf{w})$ koaternoiak. Zein da $\mathbf{q}(-\theta, -\mathbf{w})$ koaternoia? Eta zein da $\mathbf{q}(2\pi - \theta, -\mathbf{w})$ koaternoia?

   c) Azaldu *Slerp* interpolazioa zertarako erabiltzen den.

5. (3 puntu)

   a) Munduaren XYZ ardatzetan, Y ardatz bertikala izanik, ate koadro bat kokatuko dugu $x = 1/4$ planoan, atearen oinak $y = 0$ planoan daude, altuera 1 da eta zabalera ere 1 da. Gainera mutiko bat (segmentu bat) dago atearen aurrean $x = 1/2$ planoan eta bere altuera $1/2$ da. Atearen zentroa eta mutikoa $z = 0$ planoan daude. Idatzi atearen P_1, P_2, P_3, P_4 erpinen eta mutikoaren Q_1, Q_2 erpinen koordenatuak. Irudia egin: ardatzak, atea, mutikoa.

   b) Kamera kokatuko dugu $P(1, 1/2, 0)$ puntuan, atearen zentroari (eta mutikoaren burugainari) begira. Urruneko planoan $AB = AC = f = 1$ da, beraz neurriak ez dira aldatuko. Aurreko marrazkiaren gainean adierazi kameraren $X'Y'Z'$ ardatzak. Idatzi atearen eta mutikoaren erpinen koordenatuak kameraren ardatzetan.

   c) Idatzi kameraren $\mathbf{u}, \mathbf{v}, \mathbf{w}$ bektoreak honela: bakoitzaren osagaiak munduko sisteman. Idatzi M_{per} matrizea biderkadura bezala: eskalaketarik ez, biraketaren alderantzizkoa eta translazioaren aurkakoa, bi transformazio afin hauen matrizeen biderkadura.

d) Irudi bat egin kameraren $Y'Z'$ ardatzekin (Y' bertikala), plano honetan marraztu perspektibadun ikuspegi bolumenaren eta urruneko planoaren sekzioak (triangelu bat dugu kameratik $-Z'$ aldera), baita atea eta mutikoa ere (bi segmentu bertikal). Ikuspegia paralelizatzen badugu, zer ikusten dugu $X'Y'$ planoan? Kalkuluak egiteko n beharko genuke baina gutxi gorabehera marraztu.

6. (1,5 puntu)

a) Kameraren ikuspegi bolumen paraleloan ABC triangeluaren eta (x, y) zuzenaren arteko P ebakidura kalkulatzeko ekuazioak idatzi behar dira: $\alpha A + \beta B + \gamma C = P$ berdintzatik hiru ekuazio ditugu puntu bakoitzak hiru koordenatu kartesiar dituenez, gainera $\alpha + \beta + \gamma = 1$ betetzen da, beraz guztira lau ekuazio idatzi. Ezezagunak α, β, γ eta z dira, guztiak alde batera eraman, ezagunak diren gaiak bestaldera. Ekuazioak matrizialki idatzi, zein da M koefizienteen matrizea? Idatzi ekuazioen soluzioa M^{-1} erabiliz.

b) 3D eszenan triangeluak eta argiztapena emanik, 2D grafikoa renderizatzeko *rasterizazio* algoritmoaren Python programa idatzi behar da. Pentsatu *triang_distirak* zerrendan triangeluen distirak ditugula eta $Malderantzizkoak$ zerrendan gorde dela aurreko puntuan definitu den M matrizearen alderantzizkoa triangelu bakoitzerako. Gainera sortu da *piramide_irudia* pixel matrizea hasierako distirak esleituz, *zabal* da zabalera, *koadroa_xy(i,j)* funtzioa definitu da indizeetatik koordenatuak lortzeko.

Idatzi Python programa rasterizazio algoritmorako:

```
for each pixel position (x,y):
  closest[x,y] = infin
for each triangle T:
  for each pixel position (x,y):
    let R be the ray through (x,y) from the eye
    if P, the intersection of R and T, exists:
      if the distance to P is less than closest[x,y]:
        pixel[x,y] = light scattered at P to omega_o
        closest[x,y] = distance to P
```